Probabilistic Model Checking and Strategy Synthesis

Dave Parker

University of Birmingham

NASA Ames, October 2013
Probabilistic Model Checking and Strategy Synthesis

Dave Parker

University of Birmingham

Joint work with: Marta Kwiatkowska, Vojtěch Forejt, Gethin Norman, Hongyang Qu, Aistis Simaitis, Taolue Chen, ...
Overview

• Probabilistic model checking & PRISM
  – example: Bluetooth

• Verification vs. strategy synthesis
  – Markov decision processes (MDPs)
  – example: robot controller

• Multi–objective probabilistic model checking
  – examples: team–formation/power management/…

• Model checking stochastic games
  – example: energy management
Motivation

• Verifying probabilistic systems…
  – unreliable or unpredictable behaviour
    • failures of physical components
    • message loss in wireless communication
    • unreliable sensors/actuators
  – randomisation in algorithms/protocols
    • random back-off in communication protocols
    • random routing to reduce flooding or provide anonymity

• We need to verify quantitative system properties
  – “the probability of the airbag failing to deploy within 0.02 seconds of being triggered is at most 0.001”
  – not just correctness: reliability, timeliness, performance, …
Probabilistic model checking

Model description

```plaintext
module A
    a : [0..N] init N;
    ab : [0..N] init 0;
    [r1] a>0 → k₁ᵃ : (a'=a-1)&(ab'=ab+1);
    [r2] ab>0 → k₂ⁿab : (a'=a+1)&(ab'=ab-1);
    [r3] a>0 → k₃ⁿᵃ : (a'=a-1);
endmodule
```

System requirements

Probabilistic temporal logic specification

- PCTL
- CSL
- LTL

Probabilistic model

- e.g. Markov chain

Quantitative results

Counter-example, strategy

Result

P ≤ 0.1 [ F fail ]
Probabilistic model checking

- Various types of probabilistic models supported

**PRISM models**

- discrete-time Markov chains (DTMCs)
- continuous-time Markov chains (CTMCs)
- Markov decision processes (MDPs)
- probabilistic automata (PAs)
- probabilistic timed automata (PTAs)
- stochastic multi-player games (SMGs)
Probabilistic model checking

- Various types of probabilistic models supported
- Wide range of quantitative properties, expressible in temporal logics (probabilities, timing, costs, rewards, ...)

Example PRISM properties

| PCTL (reachability) | • $P_{\leq 0.1} [\ F \ fail \ ]$ – “the probability of a failure is at most 0.1”
| CSL | • $S_{>0.999} [\ up \ ]$ – “the long-run probability of availability is >0.999”
| costs & rewards | • $R_{\{time\}<100} [\ F \ done \ ]$ – “the expected termination time is at most 100 seconds”
| probabilistic LTL | • $P_{\geq 0.75} [\ (G \ \neg hazard) \ \land \ (GF \ goal) \ ]$ – “the probability of avoiding the hazard visiting the goal infinitely often is $\geq 0.75$”

Probabilistic model checking

- Various types of probabilistic models supported
- Wide range of quantitative properties, expressible in temporal logics (probabilities, timing, costs, rewards, ...)
- Often focus on numerical results (probabilities etc.)
  - analyse trends, look for system flaws, anomalies

- $P_{\leq 0.1} [ F \text{ fail} ]$ – “the probability of a failure occurring is at most 0.1”
- $P_{=?} [ F \text{ fail} ]$ – “what is the probability of a failure occurring?”
Probabilistic model checking

- Various types of probabilistic models supported
- Wide range of quantitative properties, expressible in temporal logics (probabilities, timing, costs, rewards, …)
- Often focus on numerical results (probabilities etc.)
  - analyse trends, look for system flaws, anomalies
- Provides "exact" numerical results
  - compared to, for example, simulation
Probabilistic model checking

• Various types of probabilistic models supported

• Wide range of quantitative properties, expressible in temporal logics (probabilities, timing, costs, rewards, ...)

• Often focus on numerical results (probabilities etc.)
  – analyse trends, look for system flaws, anomalies

• Provides "exact" numerical results
  – compared to, for example, simulation

• Combines numerical & exhaustive analysis
  – especially useful for nondeterministic models

- $P=? \ [ F \ fail \ \{ trigger \} \{ max \} ]$
- $P_{\text{max}}=? \ [ F \ fail ]$
Probabilistic model checking

- Various types of probabilistic models supported
- Wide range of quantitative properties, expressible in temporal logics (probabilities, timing, costs, rewards, ...)
- Often focus on numerical results (probabilities etc.)
  - analyse trends, look for system flaws, anomalies
- Provides "exact" numerical results
  - compared to, for example, simulation
- Combines numerical & exhaustive analysis
  - especially useful for nondeterministic models
- Flexible, fully automated & widely applicable
  - network/communication protocols, security, robotics & planning, power management, nanotechnology, biology...
Case study: Bluetooth

- **Device discovery between pair of Bluetooth devices**
  - performance essential for this phase

- **Complex discovery process**
  - two asynchronous 28-bit clocks
  - pseudo-random hopping between 32 frequencies
  - random waiting scheme to avoid collisions
  - 17,179,869,184 initial configurations (too many to sample effectively)

- **Probabilistic model checking (PRISM)**
  - e.g. “worst-case expected discovery time is at most 5.17s”
  - e.g. “probability discovery time exceeds 6s is always < 0.001”
  - shows weaknesses in simplistic analysis
Overview

• Probabilistic model checking & PRISM
  – example: Bluetooth

• **Verification vs. strategy synthesis**
  – Markov decision processes (MDPs)
  – example: robot controller

• Multi-objective probabilistic model checking
  – examples: team-formation/power management/…

• Model checking stochastic games
  – example: energy management
Markov decision processes (MDPs)

- Markov decision processes (MDPs)
  - model nondeterministic as well as probabilistic behaviour
  - widely used also in planning, optimal control, ...
  - nondeterministic choice between probability distributions

Nondeterminism for:
- concurrency/scheduling: interleavings of parallel components
- abstraction, or under-specification, of unknown behaviour
- adversarial behaviour of the environment, or control
• **A strategy** (or “policy” or “adversary”)
  – is a resolution of nondeterminism, based on history
  – is (formally) a mapping $\sigma$ from finite paths to distributions
  – induces an (infinite-state) discrete-time Markov chain

• **Classes of strategies:**
  – **randomisation**: deterministic or randomised
  – **memory**: memoryless, finite-memory, or infinite-memory
Example strategy

- Strategy $\sigma$ which picks $b$ then $c$ in $s_1$
  - $\sigma$ is finite-memory and deterministic

- Fragment of induced Markov chain:
Verification vs. Strategy synthesis

1. Verification
   - quantify over all possible strategies (i.e. best/worst-case)
   - $P_{\leq 0.01} [ F \text{ err} ]$: “the probability of an error occurring is $\leq 0.01$ for all strategies”
   - applications: randomised communication protocols, randomised distributed algorithms, security, …

2. Strategy synthesis
   - generation of "correct-by-construction" controllers
   - $P_{\leq 0.01} [ F \text{ err} ]$: "does there exist a strategy for which the probability of an error occurring is $\leq 0.01$?"
   - applications: robotics, power management, security, …

Two dual problems; same underlying computation:
   - compute optimal (minimum or maximum) values
Running example

- Example MDP
  - robot moving through terrain divided into a 3 x 2 grid
Example – Reachability

Verify: \( P \leq 0.6 \ [ F \text{ goal}_1 ] \)

or

Synthesise for: \( P \geq 0.4 \ [ F \text{ goal}_1 ] \)

\( \Downarrow \)

Compute: \( P_{\text{max}} = ? \ [ F \text{ goal}_1 ] \)

Optimal strategies: memoryless and deterministic

Computation: graph analysis & linear programming problem
Example – Reachability

Verify: $P_{\leq 0.6} [ F \text{ goal}_1 ]$

or

Synthesise for: $P_{\geq 0.4} [ F \text{ goal}_1 ]$

\[ \Downarrow \]

Compute: $P_{\text{max}}=? [ F \text{ goal}_1 ] = 0.5$

Optimal strategies: memoryless and deterministic

Computation: graph analysis & linear programming problem

$x_0 \geq x_1$
(east)

$x_1 \geq 0.5$
(south)
Example – Reachability

Optimal strategy:
- $s_0$: east
- $s_1$: south
- $s_2$: –
- $s_3$: –
- $s_4$: east
- $s_5$: –

Verify: $P_{\leq 0.6} \left[ F \, \text{goal}_1 \right]$ or

Synthesise for: $P_{\geq 0.4} \left[ F \, \text{goal}_1 \right]$

⇓

Compute: $P_{\text{max}} = ? \left[ F \, \text{goal}_1 \right] = 0.5$

Optimal strategies: memoryless and deterministic

Computation: graph analysis & linear programming problem
Example – Costs/rewards

R_{\text{min}} = [ F \text{ goal}_2 ]

"what is the minimum expected number of moves needed to reach \text{goal}_2?"

Optimal strategies: memoryless and deterministic

Computation: graph analysis & linear programming problem
Example – Costs/rewards

Optimal strategy:

- $s_0$: south
- $s_1$: east
- $s_2$: -
- $s_3$: -
- $s_4$: west
- $s_5$: north

$R_{min} = [F \text{ goal}_2] = 19/15$

"what is the minimum expected number of moves needed to reach $\text{goal}_2$?"

Optimal strategies: memoryless and deterministic

Computation: graph analysis & linear programming problem
Example – LTL

P_{max} = \mathbb{P} \left[ (G \neg \text{hazard}) \land (GF \text{ goal}_1) \right]

"what is the maximum probability of avoiding hazard and visiting goal$_1$ infinitely often?"

Optimal strategies: finite-memory and deterministic

Computation:

construct product of MDP and a deterministic $\omega$–automaton;
then probabilistic reachability
Example – LTL

In this instance, memoryless (not usually)

Optimal strategy:
\[ s_0 : \text{south} \]
\[ s_1 : - \]
\[ s_2 : - \]
\[ s_3 : - \]
\[ s_4 : \text{east} \]
\[ s_5 : \text{west} \]

\[ P_{\text{max}} =? \ [ (G \neg \text{hazard}) \land (GF \text{ goal}_1) ] \]

"what is the maximum probability of avoiding hazard and visiting goal\(_1\) infinitely often?" = 0.1

Optimal strategies: finite-memory and deterministic

Computation:
construct product of MDP and a deterministic \(\omega\)-automaton; then probabilistic reachability
Overview

• Probabilistic model checking & PRISM
  – example: Bluetooth

• Verification vs. strategy synthesis
  – Markov decision processes (MDPs)
  – example: robot controller

• Multi-objective probabilistic model checking
  – examples: team-formation/power management/…

• Model checking stochastic games
  – example: energy management
Multi-objective model checking

- **Multi-objective probabilistic model checking**
  - investigate trade-offs between conflicting objectives
  - in PRISM, objectives are probabilistic LTL or expected rewards

- **Achievability queries**: multi($P_{>0.95} [ \text{F send} ], R_{time>10} [ C ]$)
  - e.g. “is there a strategy such that the probability of message transmission is $> 0.95$ and expected battery life $> 10$ hrs?”

- **Numerical queries**: multi($P_{max=?} [ \text{F send} ], R_{time>10} [ C ]$)
  - e.g. “maximum probability of message transmission, assuming expected battery life-time is $> 10$ hrs?”

- **Pareto queries**:
  - multi($P_{max=?} [ \text{F send} ], R_{time_{max=?}} [ C ]$)
  - e.g. "Pareto curve for maximising probability of transmission and expected battery life-time"
Multi-objective model checking

- Multi-objective probabilistic model checking
  - investigate trade-offs between conflicting objectives
  - in PRISM, objectives are probabilistic LTL or expected rewards

- Achievability queries: \( \text{multi}(P_{>0.95} [ F send ], R_{\text{time}>10} [ C ]) \)
  - e.g. “is there a strategy such that the probability of message transmission is > 0.95 and expected battery life > 10 hrs?”

- Numerical queries: \( \text{multi}(P_{\text{max=?}} [ F send ], R_{\text{time}>10} [ C ]) \)
  - e.g. “maximum probability of message transmission, assuming expected battery life-time > 10 hrs?”

- Pareto queries:
  - \( \text{multi}(P_{\text{max=?}} [ F send ], R_{\text{time max=?}} [ C ]) \)
  - e.g. "Pareto curve for maximising probability of transmission and expected battery life-time"
Multi-objective model checking

• Optimal strategies for multiple objectives
  – may be randomised
  – and finite-memory (when using LTL formulae)

• Multi-objective probabilistic model checking
  – reduces to linear programming,
    on an MDP–automata product [TACAS'07,TACAS'11]
  – can be approximated using iterative numerical methods,
    via approximation of the Pareto curve [ATVA'12]

• Extensions [ATVA'12]
  – arbitrary Boolean combinations of objectives
    • e.g. $\psi_1 \Rightarrow \psi_2$ (all strategies satisfying $\psi_1$ also satisfy $\psi_2$)
  – time–bounded (finite–horizon) properties
Example – Multi-objective

- Achievability query
  - $P_{\geq 0.7} [G \neg \text{hazard}] \land P_{\geq 0.2} [GF \text{ goal}_1]$? True (achievable)

- Numerical query
  - $P_{\text{max}=?} [GF \text{ goal}_1]$ such that $P_{\geq 0.7} [G \neg \text{hazard}]$? $\approx 0.2278$

- Pareto query
  - for $P_{\text{max}=?} [G \neg \text{hazard}] \land P_{\text{max}=?} [GF \text{ goal}_1]$?
Example – Multi-objective

Strategy 1
(deterministic)

\[
\begin{align*}
\psi_1 &= G \neg \text{hazard} \\
\psi_2 &= GF \text{ goal}_1
\end{align*}
\]

\[
\begin{align*}
\text{s}_0 &: \text{ east} \\
\text{s}_1 &: \text{ south} \\
\text{s}_2 &: \text{-} \\
\text{s}_3 &: \text{-} \\
\text{s}_4 &: \text{ east} \\
\text{s}_5 &: \text{ west}
\end{align*}
\]
Example – Multi-objective

Strategy 2
(deterministic)

\[
\begin{align*}
\psi_1 &= G \neg \text{hazard} \\
\psi_2 &= GF \text{ goal}_1
\end{align*}
\]
**Example – Multi-objective**

Optimal strategy: (randomised)

- $s_0: 0.3226 : \text{east}$
- $0.6774 : \text{south}$
- $s_1: 1.0 : \text{south}$
- $s_2: -$ (stuck)
- $s_3: -$ (stuck)
- $s_4: 1.0 : \text{east}$
- $s_5: 1.0 : \text{west}$

\[
\begin{align*}
\psi_1 &= G \neg \text{hazard} \\
\psi_2 &= GF \text{ goal}_1
\end{align*}
\]
Multi-objective: Applications

Synthesis of team formation strategies [CLIMA'11, ATVA'12]

Synthesis of dynamic power management controllers [TACAS'11]

"minimise energy consumption, subject to constraints on:
(i) expected job queue size;
(ii) expected number of lost jobs

Probabilistic assume-guarantee framework [TACAS'10, TACAS'11, Info&Comp'13]

Assume-guarantee query: "does component M₂ satisfy guarantee G, provided that assumption A always holds?"
reduces to...
"is there an adversary (strategy) of M₂ satisfying A but not G?"
Overview

• Probabilistic model checking & PRISM
  – example: Bluetooth

• Verification vs. strategy synthesis
  – Markov decision processes (MDPs)
  – example: robot controller

• Multi-objective probabilistic model checking
  – examples: team-formation/power management/…

• Model checking stochastic games
  – example: energy management
Stochastic multi-player games (SMGs)

- **Stochastic multi-player games**
  - players control states; choose actions
  - models competitive/collaborative behaviour
  - applications: security (system vs. attacker), controller synthesis (controller vs. environment), distributed algorithms/protocols, ...

- **Property specifications: rPATL**
  - \( \langle\langle\{1,2\}\rangle\rangle P_{\geq 0.95} [ F_{\leq 45} \text{ done} ] \) : "can nodes 1,2 collaborate so that the probability of the protocol terminating within 45 seconds is at least 0.95, whatever nodes 3,4 do?"
  - formally: \( \langle\langle C\rangle\rangle \psi : \text{do there exist} \) strategies for players in \( C \) such that, for all strategies of other players, property \( \psi \) holds?

- **Model checking [TACAS'12,FMSD'13]**
  - zero sum properties: analysis reduces to 2-player games
  - PRISM-games: [www.prismmodelchecker.org/games](http://www.prismmodelchecker.org/games)
Example – Stochastic games

• Two players: 1 (robot controller), 2 (environment)
  – Probability of $s_1$–south→$s_4$ is in $[p, q] = [0.5 - \Delta, 0.5 + \Delta]$
Example – Stochastic games

- **Two players: 1 (robot controller), 2 (environment)**
  - probability of $s_1$–south$\rightarrow s_4$ is in $[p,q] = [0.5-\Delta, 0.5+\Delta]$

```
Example:
```

```
s_0 -> s_1 (0.8, 0.1) -> s_2 (0.4, 0.6)
```

```
s_1 -> s_2 (0.6, 0.1) -> s_3 (0.1, 0.9)
```

```
s_2 -> s_3 (0.1) -> s_4 (0.4)
```

```
s_1 -> s_3 (0.1) -> s_4 (0.4)
```

```
s_2 -> s_4 (0.1) -> s_5 (0.6)
```

```
s_3 -> s_4 (0.1) -> s_5 (0.6)
```

```
s_4 -> s_6 (0.1) -> s_1 (0.1)
```

```
s_5 -> s_6 (0.1) -> s_1 (0.1)
```

```
s_6 -> s_1 (0.1) -> s_3 (0.1)
```

```
s_6 -> s_2 (0.1) -> s_4 (0.4)
```

```
s_6 -> s_5 (0.1) -> s_4 (0.4)
```

```
s_6 -> s_6 (0.1)
```

```
rPATL: $\langle\langle 1 \rangle\rangle P_{max=?} [ F \text{goal}_1 ]$
```

**Optimal strategies:** memoryless and deterministic

**Computation:** graph analysis & numerical approximation
Example – Stochastic games

- Two players: 1 (robot controller), 2 (environment)
  - probability of $s_1$–south→$s_4$ is in $[p,q] = [0.5-\Delta, 0.5+\Delta]$
Example: Energy management

- **Energy management protocol for Microgrid**
  - Microgrid: local energy management
  - randomised demand management protocol
  - random back-off when demand is high

- **Original analysis [Hildmann/Saffre'11]**
  - protocol increases "value" for clients
  - simulation-based, clients are honest

- **Our analysis**
  - stochastic multi-player game model
  - clients can cheat (and cooperate)
  - model checking: PRISM-games
Example: Energy management

- Exposes protocol weakness
  - incentive for clients to act selfishly

- We propose a simple fix (and verify it)
  - clients can be punished

Value per client

Value per client, with fix
Conclusion

• **Probabilistic model checking & PRISM**
  – Markov decision processes (MDPs)
  – PCTL, probabilistic LTL, expected costs/rewards
  – verification vs. controller synthesis

• **Multi-objective probabilistic model checking**
  – trade-offs between conflicting objectives
  – achievability queries, numerical queries, Pareto curves

• **Model checking for stochastic multi-player games**
  – competitive/collaborative behaviour
  – rPATL model checking

  – **Challenges**
    – stochastic games: multiple objectives, richer temporal logics
    – partial information/observability