Probabilistic Model Checking and Strategy Synthesis

Dave Parker

University of Birmingham

UC Berkeley, October 2014
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, Klaus Draeger, Mateusz Ujma, Bruno Lacerda, Nick Hawes
Overview

• **Probabilistic model checking**
  – verification vs. strategy synthesis
  – Markov decision processes (MDPs)
  – example: robot navigation

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

• **Stochastic (multi-player) games**
  – example: energy management

• **Permissive controller synthesis**
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, …
  – not just verification: correctness by construction
Probabilistic model checking

Model description

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

Probabilistic model checker

* e.g. PRISM

Probabilistic temporal logic specification

* e.g. PCTL, CSL, LTL

P ≤ 0.1 [ F fail ]

Quantitative results

Counter-example, strategy

Result
Probabilistic model checking

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

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

- Many types of probabilistic models supported
- Wide range of quantitative properties, expressible in temporal logic (probabilities, timing, costs, rewards, …)
- Often focus on numerical results (probabilities etc.)
  - analyse trends, look for system flaws, anomalies
- Provides "exact" numerical results/guarantees
  - compared to, for example, simulation
- Combines numerical & exhaustive analysis
  - especially useful for nondeterministic models
- Fully automated, tools available, widely applicable
  - network/communication protocols, security, biology, robotics & planning, power management, …
Markov decision processes (MDPs)

- Markov decision processes (MDPs)
  - widely used also in: AI, planning, optimal control, ...
  - model **nondeterministic** as well as **probabilistic** behaviour

Nondeterminism for:
- **control**: decisions made by a controller or scheduler
- **adversarial** behaviour of the environment
- **concurrency/scheduling**: interleavings of parallel components
- **abstraction**, or under-specification, of unknown behaviour
• A strategy (or “policy” or “adversary”)
  – is a resolution of nondeterminism, based on history
  – i.e. a mapping from finite paths to (distributions over) actions
  – induces an (infinite-state) discrete–time Markov chain

• Classes of strategies:
  – memory: memoryless, finite–memory, or infinite–memory
  – randomisation: deterministic or randomised
Verification vs. Strategy synthesis

• 1. Verification
  – quantify over all possible strategies (i.e. best/worst-case)
  – $P_{\leq 0.1} \left[ F \text{ err} \right]$: “the probability of an error occurring is $\leq 0.1$ for all strategies”
  – applications: randomised communication protocols, randomised distributed algorithms, security, …

• 2. Strategy synthesis
  – generation of "correct-by-construction" controllers
  – $P_{\leq 0.1} \left[ F \text{ err} \right]$: "does there exist a strategy for which the probability of an error occurring is $\leq 0.1$?"
  – 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: $P_{\leq 0.6} \left[ F \operatorname{goal}_1 \right]$

or

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

⇓

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

Optimal strategies: memoryless and deterministic

Computation:
graph analysis + numerical soln.
(linear programming, value iteration, policy iteration)
Example – Reachability

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: $P_{\text{max}} = 0.5$

Optimal strategies: memoryless and deterministic

Computation:
graph analysis + numerical soln.
(linear programming, value iteration, policy iteration)

Optimal strategy:
$s_0 : \text{east}$
$s_1 : \text{south}$
$s_2 : -$  
$s_3 : -$  
$s_4 : \text{east}$
$s_5 : -$
MDPs – Other properties

- **Costs and rewards** (expected, accumulated values)
  - e.g. $R_{min} = \mathbb{E}\left[ F \text{goal}_2 \right]$ – "what is the minimum expected number of moves needed to reach goal$_2$?"
  - optimal strategies: memoryless and deterministic
  - similar computation to probabilistic reachability

- **Probabilistic LTL** (multiple temporal operators)
  - e.g. $P_{max} = \mathbb{P}\left[ (G\neg \text{hazard}) \land (GF \text{goal}_1) \right]$ – "maximum probability of avoiding hazard and visiting goal$_1$ infinitely often?"
  - optimal strategies: finite-memory and deterministic
  - build product MDP, graph analysis, probabilistic reachability

- **Expected cost/reward** to satisfy (co-safe) LTL formula
  - e.g. $R_{min} = \mathbb{E}\left[ \neg \text{zone}_3 U (\text{zone}_1 \land (F \text{zone}_4)) \right]$ – "minimise exp. time to patrol zones 1 then 4, without passing through 3".
Application: Robot navigation

- **Navigation planning for a service robot** [IROS'14]
  - MetraLabs Scitos A5 robot + PRISM–based ROS module

- **MDP–based strategy synthesis**
  - MDP navigation map learnt with transition probabilities/times
  - high–level planning using MDPs + multiple co–safe LTL tasks
  - finite–memory strategies used to construct controllers
  - low–level navigation with separate continuous–space planner

![Image of robot and navigation graph](image-url)
Overview

• Probabilistic model checking
  – verification vs. strategy synthesis
  – Markov decision processes (MDPs)
  – example: robot navigation

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

• Stochastic (multi-player) games
  – example: energy management

• Permissive controller synthesis
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 is $> 10$ hrs?”

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

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

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

- **Optimal strategies:**
  - usually *finite-memory* (e.g. when using LTL formulae)
  - may also need to be *randomised*

- **Computation:**
  - construct a product MDP (with several automata),
    then reduces to linear programming [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 $\psi_2$)
    - (e.g. for assume–guarantee reasoning)
  - time–bounded (finite–horizon) properties
• **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 \( P_{\geq 0.7} [ G \neg \text{hazard} ] \) ? \( \sim 0.2278 \)

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

Strategy 1 (deterministic)
- $s_0 : east$
- $s_1 : south$
- $s_2 : -$  
- $s_3 : -$  
- $s_4 : east$
- $s_5 : west$

$\psi_1 = G \neg \text{hazard}$
$\psi_2 = GF \text{ goal}_1$
Example – Multi-objective

Strategy 2
(deterministic)

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

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

Optimal strategy:
(randomised)

\[ s_0 : 0.3226 : \text{east} \]
\[ 0.6774 : \text{south} \]

\[ s_1 : 1.0 : \text{south} \]

\[ s_2 : - \]

\[ s_3 : - \]

\[ s_4 : 1.0 : \text{east} \]

\[ s_5 : 1.0 : \text{west} \]
Multi-objective: Applications

Synthesis of controllers for dynamic power management [TACAS'11]

IBM TravelStar VP disk drive
- switches between power modes:
  - active/idle/idlelp/stby/sleep

MDP model in PRISM:
- power manager
- disk requests
- request queue
- power usage

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

Pareto curve:
\[ x = \text{"probability of completing task 1"}; \]
\[ y = \text{"probability of completing task 2"}; \]
\[ z = \text{"expected size of successful team"} \]
Overview

• Probabilistic model checking
  – verification vs. strategy synthesis
  – Markov decision processes (MDPs)
  – example: robot navigation

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

• Stochastic (multi-player) games
  – example: energy management

• Permissive controller synthesis
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: $\langle\langle C \rangle\rangle \psi : \text{do there exist} \text{ strategies for players in } C \text{ such that, for all strategies of other players, property } \psi \text{ holds?}$

- **Model checking [TACAS'12,FMSD'13]**
  - zero sum properties: analysis reduces to 2-player games
  - PRISM-games: 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→$s_4$ is in $[p,q] = [0.5-\Delta, 0.5+\Delta]$

![Stochastic Game Diagram]

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

Optimal strategies: memoryless and deterministic

Computation: graph analysis & numerical approximation

$S_i$ Player 1  $S_j$ Player 2
Example – Stochastic games

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

- **Energy management protocol for Microgrid**
  - 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
  - exposes protocol weakness (incentive for clients to act selfishly)
  - propose/verify simple fix using penalties
Overview

• Probabilistic model checking
  – verification vs. strategy synthesis
  – Markov decision processes (MDPs)
  – example: robot navigation

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

• Stochastic (multi-player) games
  – example: energy management

• Permissive controller synthesis
Permissive controller synthesis

- **Multi-strategy synthesis** [TACAS'14]
  - for Markov decision processes and stochastic games
  - choose *sets* of actions to take in each state
  - controller is free to choose any action at runtime
  - flexible/robust (e.g. actions become unavailable or goals change)

- **Example**

```plaintext
Multi-strategy:
- s₀: east or south
- s₁: south
- s₂: -
- s₃: -
- s₄: east
- s₅: west
```

```
0.4 east
0.1 south
0.8 south
0.1 stuck
0.6 stuck
0.1 west
0.4 west
0.5 south
0.9 stuck
0.1 north
```

```
S₀ -> S₁: 0.6 (east)
S₁ -> S₂: 0.5 (south)
S₂ -> S₁: 0.9 (stuck)
S₁ -> S₀: 0.1 (stuck)
S₀ -> S₃: 0.1 (west)
S₃ -> S₄: 0.4 (west)
S₄ -> S₅: 0.1 (north)
S₅ -> S₄: 0.9 (east)
S₄ -> S₃: 0.1 (goal₁)
S₃ -> S₀: 0.1 (goal₂)
S₀ -> S₅: 0.6 (goal₂)
S₅ -> S₀: 0.4 (goal₁)
```
Permissive controller synthesis

- Multi-strategies and temporal logic
  - multi-strategy $\Theta$ satisfies a property $P_{>p}[F \text{ goal }]$ iff any strategy $\sigma$ that adheres to $\Theta$ satisfies $P_{>p}[F \text{ goal }]

- We quantify the permissivity of multi-strategies
  - by assigning penalties to each action in each state
  - a multi-strategy is penalised for every action it blocks
  - static and dynamic (expected) penalty schemes

- Permissive controller synthesis
  - $\exists$ a multi-strategy satisfying $P_{\leq 0.6}[F \text{ goal}_1]$ with penalty $< c$?
  - what is the multi-strategy with optimum permissivity?
  - reduction to mixed-integer LP problems
  - applications: energy management, cloud scheduling, ...
Conclusion

- **Probabilistic model checking**
  - verification vs. controller synthesis
  - Markov decision processes, temporal logic, PRISM

- **Recent directions and extensions**
  - multi-objective probabilistic model checking
  - model checking for stochastic games
  - permissive controller synthesis

- **Challenges**
  - stochastic games: multi-objective, equilibria, richer logics
  - partial information/observability
  - probabilistic models with continuous time (or space)
  - scalability, e.g. symbolic methods, abstraction

www.prismmodelchecker.org