Probabilistic Model Checking
and Strategy Synthesis
for Robot Navigation

Dave Parker
University of Birmingham
(joint work with Bruno Lacerda, Nick Hawes)

Overview

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

• **Application: Robot navigation**
  – probabilistic model checking + MDPs + LTL

• **Strategy synthesis techniques**
  – multi-objective probabilistic model checking
  – partially satisfiable task specifications
  – uncertainty + stochastic games
  – permissive controller synthesis
Quantitative verification

- Formal verification + quantitative aspects
- Probability
  - component failures, lossy communication, unreliable sensors/actuators, randomisation in algorithms/protocols
- Time: delays, time-outs, failure rates, ...
- Costs & rewards
  - energy consumption, resource usage, ...
- Not just about correctness...
  - reliability, timeliness, performance, efficiency, ...
  - “the probability of an airbag failing to deploy within 0.02 seconds of being triggered is at most 0.001”
  - “the expected energy consumption of the sensor is...”
Probabilistic model checking

- Construction and analysis of probabilistic models
  - state-transition systems labelled with probabilities (e.g. Markov chains, Markov decision processes)
  - from a description in a high-level modelling language

- Properties expressed in temporal logic, e.g. PCTL:
  - trigger $\rightarrow P_{\geq 0.999} [ F_{\leq 20} \text{ deploy} ]$
  - “the probability of the airbag deploying within 20ms of being triggered is at least 0.999”
  - properties checked against models using exhaustive search and numerical computation
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

- \( P_{\leq 0.1}[ F \text{ fail} ] \) – “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

- 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/heuristics
  - combines numerical & exhaustive analysis
- Fully automated, tools available, widely applicable
  - network/communication protocols, security, biology, robotics & planning, power management, …
- Key challenge: scalability
Markov decision processes (MDPs)

- Markov decision processes (MDPs)
  - also widely used also in: AI, planning, optimal control, ...

- A strategy (or “policy” or “adversary”)
  - resolves choices in an MDP based on its history so far

- Used to model:
  - control: decisions made by a controller or scheduler
  - adversarial behaviour of the environment
  - concurrency/scheduling: interleavings of parallel components

- 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)
   - $P_{\leq 0.1} [F_{err}]$: “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
   - $P_{\leq 0.1} [F_{err}]$: "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
Applications

• **Examples of PRISM–based strategy synthesis**

  Synthesis of dynamic power management controllers [TACAS'11]

  Motion planning for a service robot using LTL [IROS'14]

  Team formation strategy synthesis [CLIMA'11, ATVA'12]

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

**Pareto curve:**

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

- Example MDP
  - robot moving through terrain divided in to 3 x 2 grid
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: $P_{\max}=? \left[ F \text{ goal}_1 \right]$

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

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

Optimal strategies: memoryless and deterministic

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

Optimal strategy:  
$s_0 : \text{east}$  
$s_1 : \text{south}$  
$s_2 : -$  
$s_3 : -$  
$s_4 : \text{east}$  
$s_5 : -$
Linear temporal logic (LTL)

- **Probabilistic LTL** (multiple temporal operators)
  - e.g. $P_{\text{max}} = \text{?} \left[ (G\neg \text{hazard}) \land (GF \text{ goal}_1) \right]$ - "maximum probability of avoiding hazard and visiting goal$_1$ infinitely often?"
  - e.g. $P_{\text{max}} = \text{?} \left[ \neg \text{zone}_3 \lor (\text{zone}_1 \land (F \text{ zone}_4)) \right]$ - "max. probability of patrolling zones 1 then 4, without passing through 3".

- **Probabilistic model checking**
  - convert LTL formula $\psi$ to deterministic automaton $A_\psi$ (Buchi, Rabin, finite, …)
  - build/solve product MDP $M \otimes A_\psi$
  - reduction to simpler problem
  - optimal strategies are:
    - deterministic
    - finite-memory

\[
\text{Det. Buchi automaton } A_\psi \\
\text{for } \psi = G \neg h \land GF g_1
\]
Example: Product MDP construction

\[
M \otimes A_\psi
\]
Example: Product MDP construction

\[ M \otimes A_\psi \]

\[ \psi = G \neg h \land GF g \]
Co-safe LTL (and expected cost)

- Often focus on tasks completed in finite time
  - can restrict to co-safe fragment(s) of LTL
  - (any satisfying execution has a "good prefix")
  - e.g. $P_{\text{max}}= \neg \text{zone}_3 \up U (\text{zone}_1 \land (F \text{zone}_4))$
  - for simplicity, can restrict to syntactically co-safe LTL

- Expected cost/reward to satisfy (co-safe) LTL formula
  - e.g. $R_{\text{min}}= \neg \text{zone}_3 \up U (\text{zone}_1 \land (F \text{zone}_4))$ - "minimise exp. time to patrol zones 1 then 4, without passing through 3".

- Solution:
  - product of MDP and DFA
  - expected cost to reach accepting states in product
Overview

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

• Application: Robot navigation
  – probabilistic model checking + MDPs + LTL

• Strategy synthesis techniques
  – multi-objective probabilistic model checking
  – partially satisfiable task specifications
  – uncertainty + stochastic games
  – permissive controller synthesis
Application: Robot navigation

- **Navigation planning:**
  - **MDP** models navigation through an uncertain environment
  - **LTL** used to formally specify tasks to be executed
  - synthesise finite-memory strategies to construct plans/controllers

![Diagram of robot navigation](Image)
Application: Robot navigation

- **Navigation planning MDPs**
  - expected timed on edges + probabilities
  - learnt using data from previous explorations

- **LTL-based task specification**
  - expected time to satisfy (one or more) co-safe LTL formulas

- **Benefits of the approach**
  - LTL: flexible, unambiguous property specification
  - efficient, fully-automated techniques
    - LTL-to-automaton conversion, MDP solution
  - c.f. ad-hoc reward structures, e.g. with discounting
  - meaningful properties: probabilities, time, energy,…
  - guarantees on performance ("correct by construction")
Implementation & deployment

- **Implementation**
  - MetraLabs Scitos A5 robot
  - ROS module based on PRISM
  - with extensions:
    - co-safe LTL expectation
    - efficient re-planning [IROS'14]

- **Example deployment:**
  
  G4S Technology, Tewkesbury (STRANDES)
Probabilistic model checking

• Further use of probabilistic model checking…
  – (various probabilistic models, query languages)

• Nested queries
  – e.g. $R_{\text{min=?}} [ \text{safe} \ U (\text{zone}_1 \land (F \text{zone}_4)) ]$ – "minimise exp. time to patrol zones 1 then 4, passing only through safe".
  – where safe denotes states satisfying $\langle \langle \text{ctrl} \rangle \rangle R_{<2} [ F \text{base} ]$ – "there is a strategy to return to base with expected time < 2"

• Analysis of generated controllers
  – expected power consumption to complete tasks?
  – conditional expectation, e.g. expected time to complete task, assuming it is completed successfully?
  – more detailed timing information (not just mean time)
Overview

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

• Application: Robot navigation
  – probabilistic model checking + MDPs + LTL

• Strategy synthesis techniques
  – multi–objective probabilistic model checking
  – partially satisfiable task specifications
  – uncertainty + stochastic games
  – 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 costs

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

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

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

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

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

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

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

- **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
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 $P_{\geq 0.7} [ G \neg \text{hazard} ]$ ? $\sim 0.2278$

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

Strategy 1 (deterministic)

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

$\psi_1 = \text{GF} \text{goal}_1$

$\psi_2 = \text{G} \neg\text{hazard}$
Example – Multi-objective

\[ \psi_1 = G \neg \text{hazard} \]
\[ \psi_2 = GF \text{ goal}_1 \]

Strategy 2
(deterministic)
\[ s_0 : \text{south} \]
\[ s_1 : \text{south} \]
\[ s_2 : \text{--} \]
\[ s_3 : \text{--} \]
\[ s_4 : \text{east} \]
\[ s_5 : \text{west} \]
Example – Multi-objective

Optimal strategy: (randomised)

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

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

\[ s_2 : - \]

\[ s_3 : - \]

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

\[ s_5 : 1.0 : \text{west} \]
Application: Partially satisfiable tasks

- Partially satisfiable task specifications
  - via multi-objective probabilistic model checking [IJCAI'15]
  - e.g. $P_{\text{max}=?} [ \neg \text{zone}_3 U (\text{room}_1 \land (F \text{room}_4 \land F \text{room}_5)) ] < 1$

- Synthesise strategies that, in decreasing order of priority:
  - maximise the probability of finishing the task;
  - maximise progress towards completion, if this is not possible;
  - minimise the expected time (or cost) required

- Progress metric constructed from DFA
  - (distance to accepting states, reward for decreasing distance)

- Encode prioritisation using multi-objective queries:
  - $p = P_{\text{max}=?} [ \text{task} ]$
  - $r = \text{multi}(R_{\text{max}=?}^{\text{prog}} [ C ], P_{\geq p} [ \text{task} ])$
  - $\text{multi}(R_{\text{min}=?}^{\text{time}} [ C ], P_{\geq p} [ \text{task} ] \land R_{\geq r}^{\text{prog}} [ C ])$. 
Overview

- Probabilistic model checking
  - verification vs. strategy synthesis
  - Markov decision processes (MDPs)

- Application: Robot navigation
  - probabilistic model checking + MDPs + LTL

- Strategy synthesis techniques
  - multi-objective probabilistic model checking
  - partially satisfiable task specifications
  - uncertainty + stochastic games
  - permissive controller synthesis
MDPs + uncertainty

• Modelling uncertainty
  – e.g., transitions probabilities (or costs) specified as intervals

• Worst-case analysis
  – i.e. adversarial choice of probability values
  – stochastic game: controller vs. environment
  – "min–max" analysis
MDPs + uncertainty

- **Modelling uncertainty**
  - e.g., transitions probabilities (or costs) specified as intervals

- **Worst-case analysis**
  - i.e. adversarial choice of probability values
  - stochastic game: controller vs. environment
  - "min–max" analysis

- **PRISM-games** [FMSD'13]
  - stochastic multi-player games
  - temporal logic queries (rPATL)
  - e.g. $\langle\langle \text{ctrl} \rangle \rangle P_{\max} = ? [ F \text{goal}_1 ]$
  - reduces to solving 2-player game

\[ [p,q] = [0.5-\Delta, 0.5+\Delta] \]
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**

  ![Diagram]

  Multi-strategy:
  - $s_0$: east or south
  - $s_1$: south
  - $s_2$: 
  - $s_3$: 
  - $s_4$: east
  - $s_5$: west
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 $P_{\leq0.6}[F\text{ goal}_1]$ with penalty $< c$?
  - what is the multi-strategy with optimum permissivity?
  - reduction to mixed-integer LP problems
  - other applications: energy management, cloud scheduling, …
Conclusion

- **Probabilistic model checking & strategy synthesis**
  - Markov decision processes, temporal logic, PRISM
- **Robot navigation using MDPs & LTL**
  - PRISM extension embedded in ROS navigation stack
- **Recent extensions**
  - multi-objective probabilistic model checking
  - uncertainty & stochastic games, permissive controller synthesis
- **Challenges & directions**
  - partial information/observability, e.g. POMDPs
  - probabilistic models with continuous time (or space)
  - scalability, e.g. symbolic methods, abstraction

[www.prismmodelchecker.org](http://www.prismmodelchecker.org)