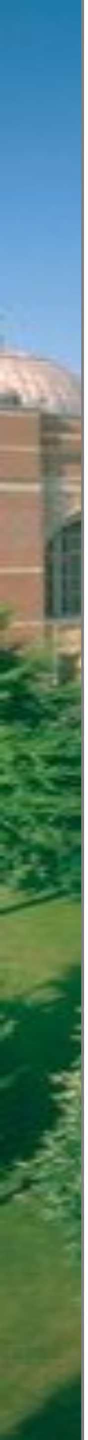
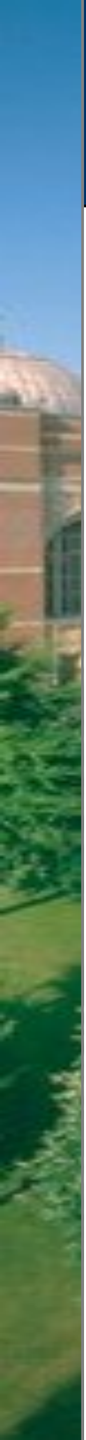


Probabilistic Model Checking: Advances and Applications

Dave Parker

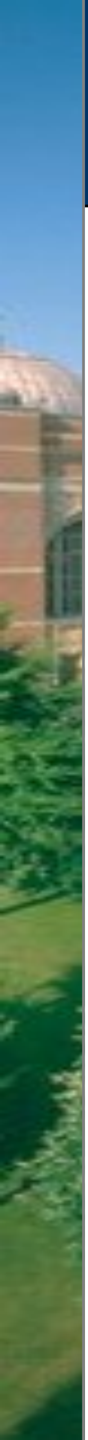
University of Birmingham

Highlights'18, Berlin, September 2018

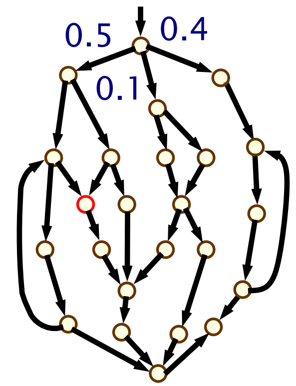


Overview

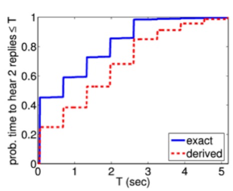
- Probabilistic model checking & PRISM
  - Markov decision processes (MDPs)
- Multi-objective probabilistic model checking
  - examples: robot navigation; task scheduling
- Partially observable models
  - POMDPs + real-time variants
  - examples: robot navigation; wireless scheduling
- Stochastic (multi-player) games
  - turn-based & concurrent games
  - examples: energy management, investor models

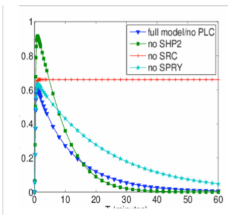


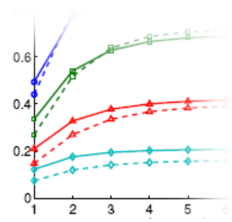
Probabilistic model checking

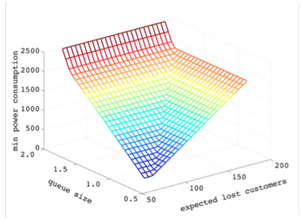


- Probabilistic model checking
  - formal construction/analysis of probabilistic models
  - "correctness" properties expressed in temporal logic
  - e.g. trigger → $P_{\geq 0.999}$ [$F^{\leq 20}$ deploy]
  - mix of exhaustive & numerical/quantitative reasoning









- Trends and advances
  - improvement in scalability to larger models
  - increasingly expressive/powerful model classes
  - from verification problems to control problems
  - ever widening range of application domains

PRISM (and extensions)

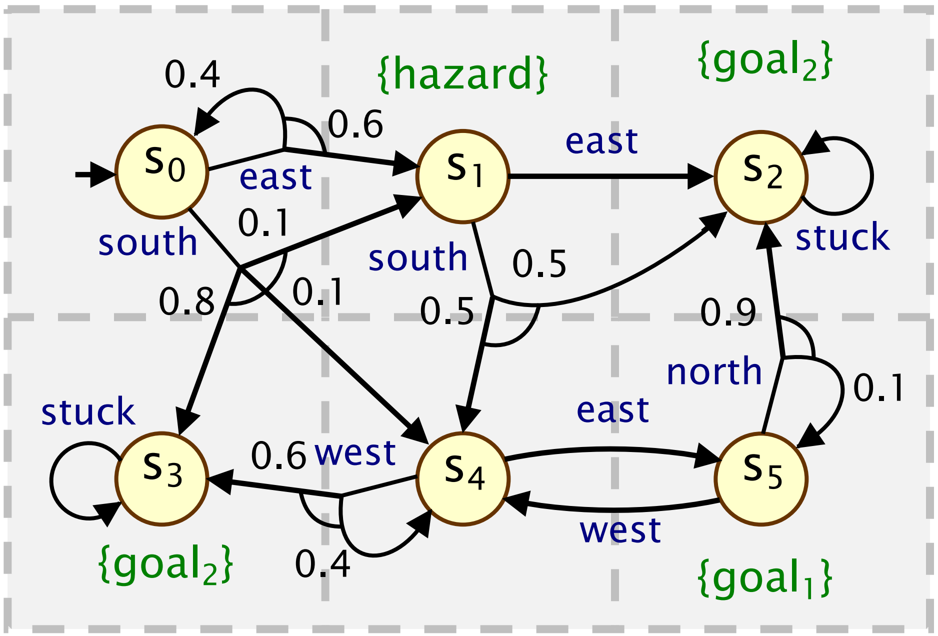
- PRISM model checker: www.prismmodelchecker.org
- Wide range of probabilistic models
  - discrete states & probabilities: Markov chains
  - + nondeterminism: Markov decision processes (MDPs)
  - + real-time clocks: probabilistic timed automata (PTAs)
  - + partial observability: POMDPs and POPTAs
  - + multiple players: (turn-based) stochastic games
  - + concurrency: concurrent stochastic games
- Expressive property specification language
  - PCTL/CSL, LTL, costs/rewards, multi-objective, strategies, …
- Tool features
  - modelling language, simulator, GUI, graph plotting, …

PRISM (and extensions)

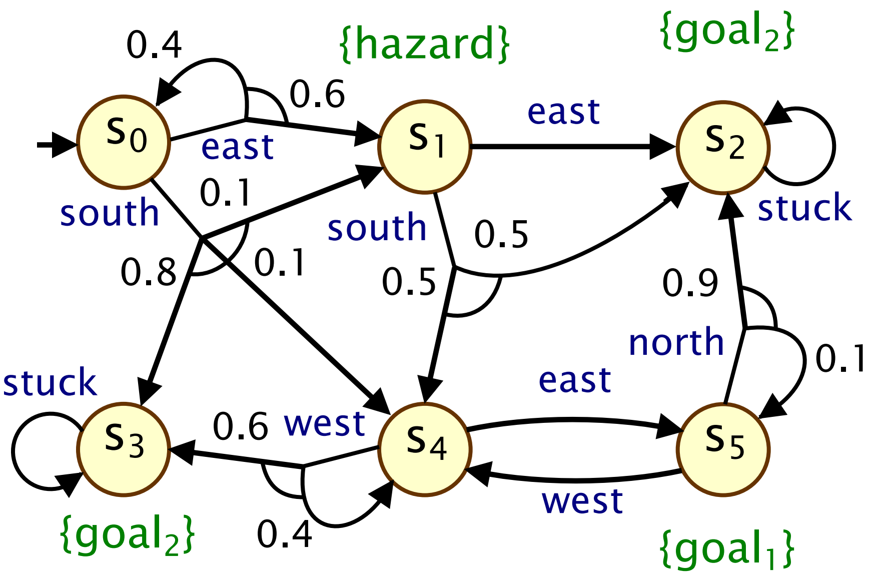
- Various verification engines
  - symbolic/explicit/hybrid, exact, parametric, statistical model checking, abstraction refinement, …
- Open source development
  - github.com/prismmodelchecker/prism
  - incl. benchmark & testing suites
- Interfaces & connections
  - Java API
  - ModelGenerator interface: programmatic model construction
  - HOAF support for automata import/export

Markov decision processes

- Example Markov decision processes (MDP)
  - robot moving through terrain divided in to 3 x 2 grid
  - strategies represent possible ways to navigate grid



Example – Reachability



Synthesise strategy satisfying:
$P_{\geq 0.4}$ [F goal$_1$]

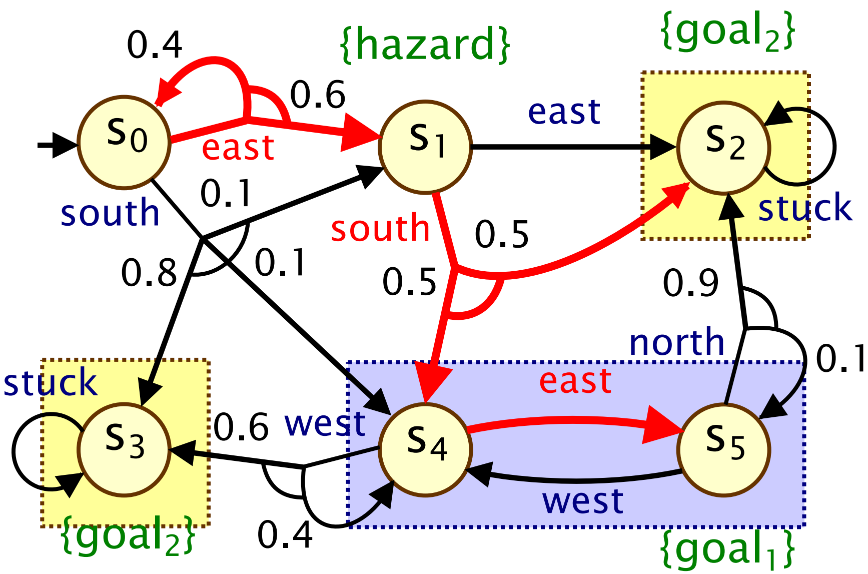
or

Find optimal strategy
$P_{max=?}$ [F goal$_1$]

Optimal strategies:
memoryless and deterministic

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

Example – Reachability



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

Synthesise strategy satisfying:
$P_{\geq 0.4}$ [F goal$_1$]

or

Find optimal strategy
$P_{max=?}$ [F goal$_1$] = 0.5

Optimal strategies:
memoryless and deterministic

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

MDPs – Other core properties

- Costs and rewards (expected, accumulated values)
  - e.g. $R_{min=?}$ [F $goal_2$] – "what is the minimum expected time needed to reach $goal_2$?"
  - optimal strategies: memoryless and deterministic
  - similar computation to probabilistic reachability

- Probabilistic LTL (multiple temporal operators)
  - e.g. $P_{max=?}$ [(G¬hazard) ∧ (GF $goal_1$)] – "maximum probability of avoiding hazard and visiting $goal_1$ infinitely often?"
  - optimal strategies: finite-memory and deterministic
  - build product MDP, graph analysis, probabilistic reachability

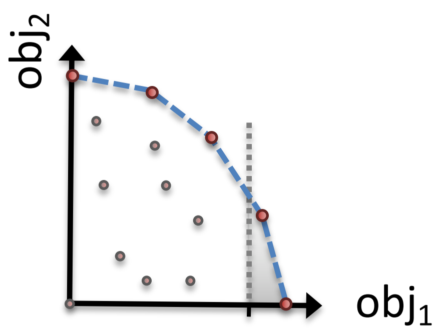
- Expected cost/reward to satisfy (co-safe) LTL formula
  - e.g. $R_{min=?}$ [$\neg zone_3$ U ($zone_1$ ∧ (F $zone_4$))] – "minimise exp. time to patrol zones 1 then 4, without passing through 3".

Overview

- Probabilistic model checking & PRISM
  - Markov decision processes (MDPs)
- Multi-objective probabilistic model checking
  - examples: robot navigation; task scheduling
- Partially observable models
  - POMDPs + real-time variants
  - examples: robot navigation; wireless scheduling
- Stochastic (multi-player) games
  - turn-based & concurrent games
  - examples: energy management, investor models

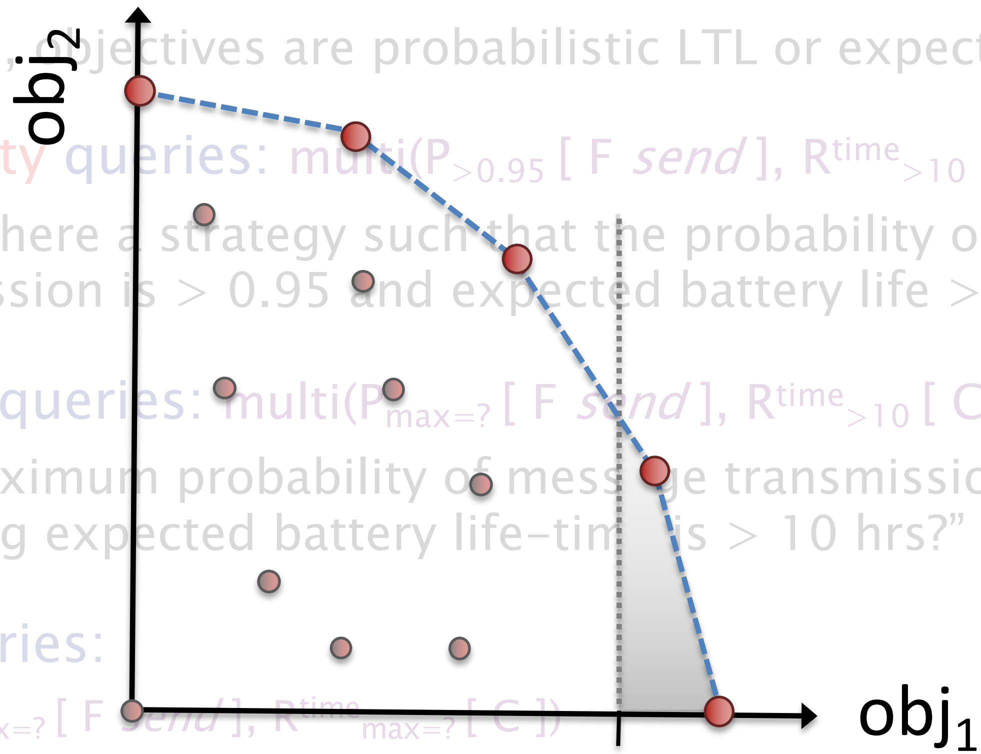
Multi-objective model checking

- Multi-objective probabilistic model checking
  - investigate trade-offs between conflicting objectives
  - in PRISM, objectives are probabilistic LTL or expected rewards
- Achievability queries: multi($P_{\geq 0.95}$ [F *send*], $R^{time}_{\geq 10}$ [C])
  - e.g. "is there a strategy such that the probability of message transmission is ≥ 0.95 <u>and</u> expected battery life ≥ 10 hrs?"
- Numerical queries: multi($P_{max=?}$ [F *send*], $R^{time}_{\geq 10}$ [C])
  - e.g. "maximum probability of message transmission, assuming expected battery life-time is ≥ 10 hrs?"
- Pareto queries:
  - multi($P_{max=?}$ [F *send*], $R^{time}_{max=?}$ [C])
  - e.g. "Pareto curve for maximising probability of transmission and expected battery life-time"

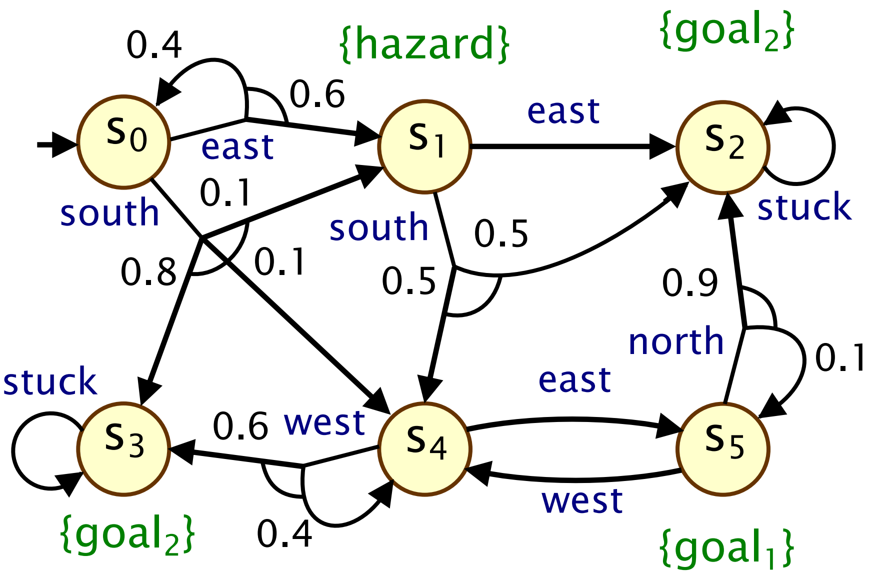


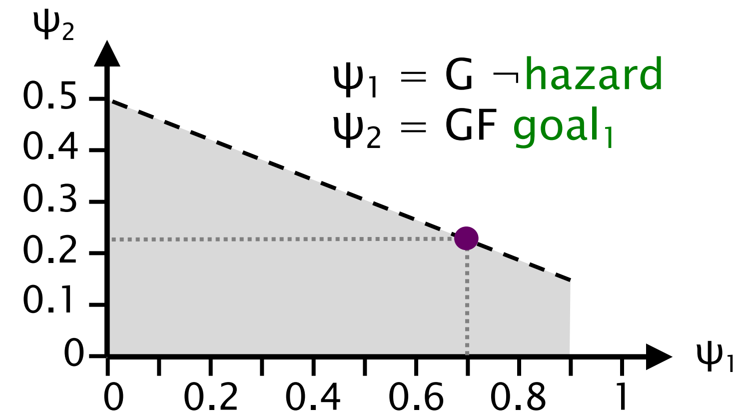
Multi-objective model checking

- Multi-objective probabilistic model checking
  - investigate trade-offs between conflicting objectives
  - in PRISM, objectives are probabilistic LTL or expected rewards
- Achievability queries: multi($P_{>0.95}$ [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($P_{max=?}$ [F *send*], $R^{time}_{>10}$ [C])
  - e.g. “maximum probability of message transmission, assuming expected battery life-time is > 10 hrs?”
- Pareto queries:
  - multi($P_{max=?}$ [F *send*], $R^{time}_{max=?}$ [C])
  - e.g. "Pareto curve for maximising probability of transmission and expected battery life-time”



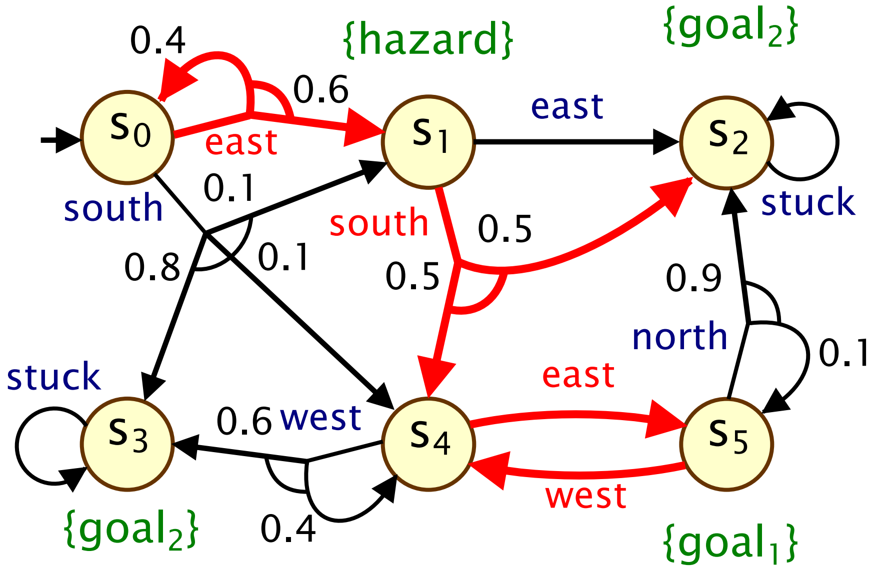
Example – Multi-objective





- Achievability query
  - $P_{\geq 0.7}$ [G ¬hazard] ∧ $P_{\geq 0.2}$ [GF goal$_1$] ? True (achievable)
- Numerical query
  - $P_{max=?}$ [GF goal$_1$] such that $P_{\geq 0.7}$ [G ¬hazard] ? ~0.2278
- Pareto query
  - for $P_{max=?}$ [G ¬hazard], $P_{max=?}$ [GF goal$_1$] ?

Example – Multi-objective



Strategy 1 (deterministic)

s_0 : east

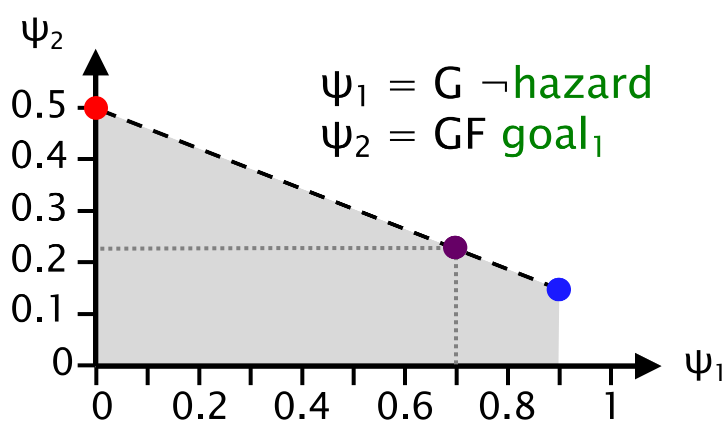
s_1 : south

s_2 : –

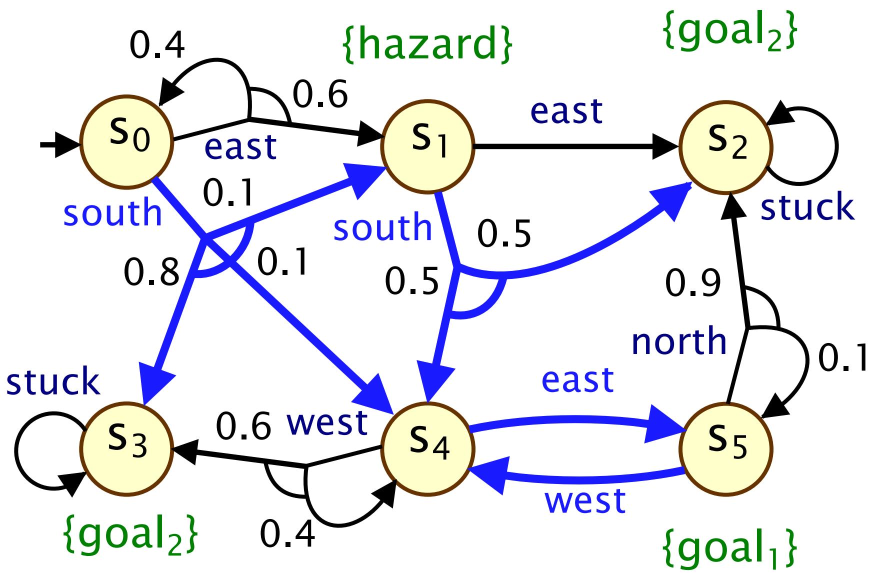
s_3 : –

s_4 : east

s_5 : west



Example – Multi-objective



Strategy 2 (deterministic)

s_0 : south

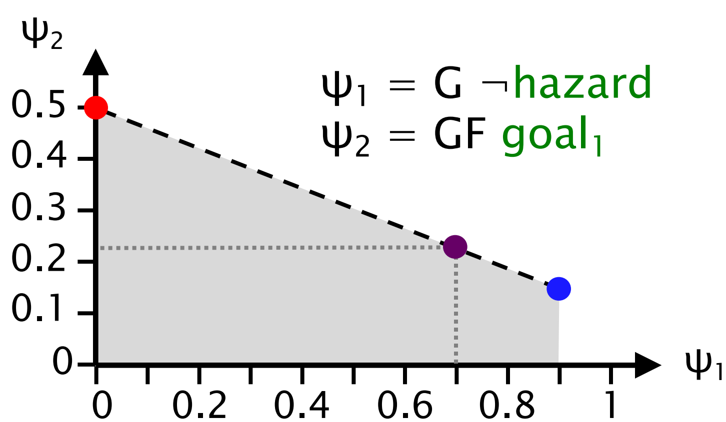
s_1 : south

s_2 : –

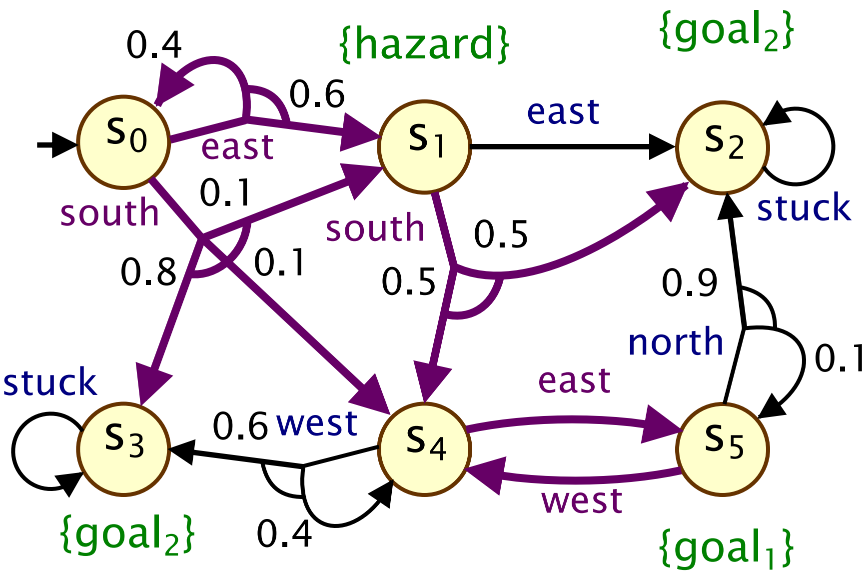
s_3 : –

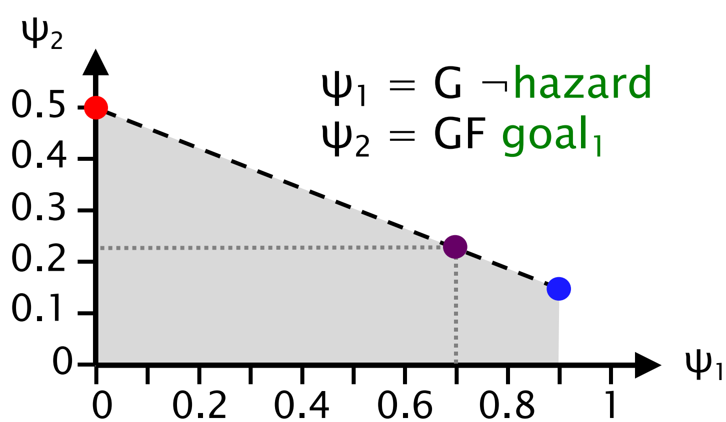
s_4 : east

s_5 : west



Example – Multi-objective



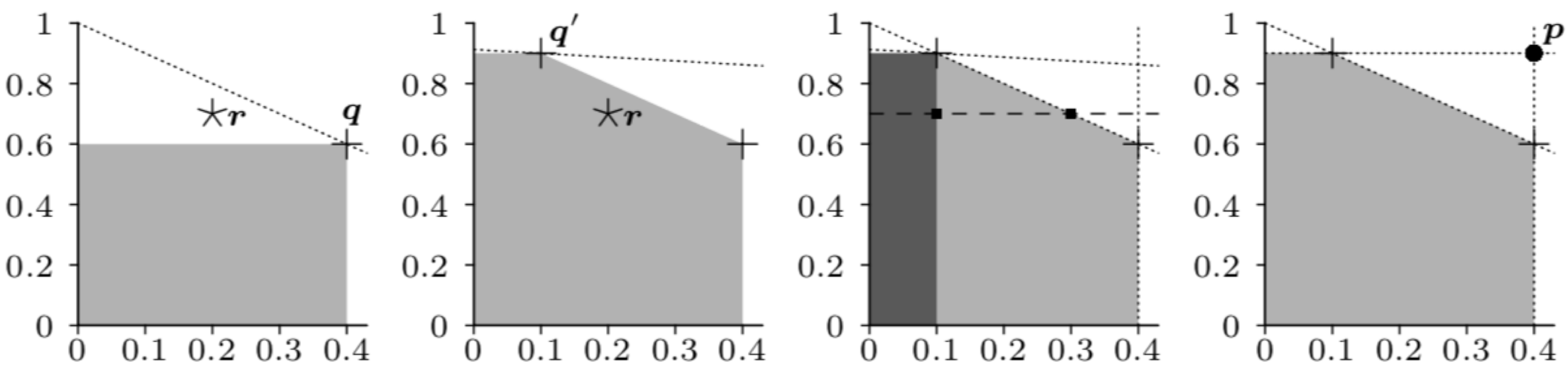


Optimal strategy:
(randomised)

s_0 : 0.3226 : east
0.6774 : south
s_1 : 1.0 : south
s_2 : –
s_3 : –
s_4 : 1.0 : east
s_5 : 1.0 : west

Multi-objective model checking

- PRISM implements two distinct approaches
- 1. Linear programming
  - solve dual problem to classical LP formulation
- 2. Value iteration based weighted sweep
  - approximate exploration/construction of Pareto curve
  - e.g. $P_{\geq r1}$ […] $\wedge$ $P_{\geq r2}$ […] for $\mathbf{r}=(r_1,r_2)=(0.2,0.7)$

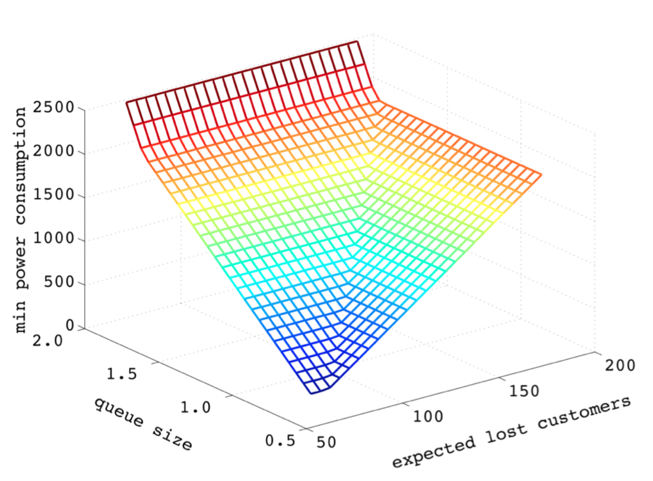


  - method 2 extends to step-bounded objectives

Applications – Multi-objective

- Examples of multi-objective controller synthesis with PRISM

Synthesis of dynamic power management controllers

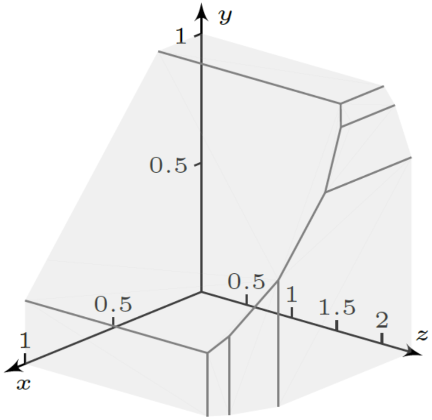


Minimise energy consumption, subject to constraints on:
(i) expected job queue size;
(ii) expected num. lost jobs

Motion planning for service robots using LTL

Partial task satisfaction; task progress metrics; efficient time bounded probabilistic guarantees

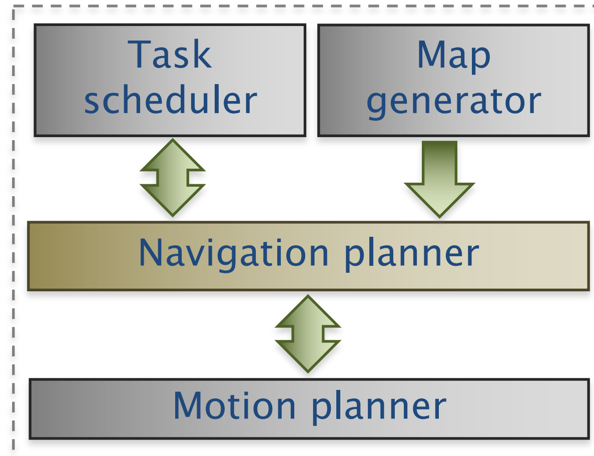
Synthesis of team formation strategies

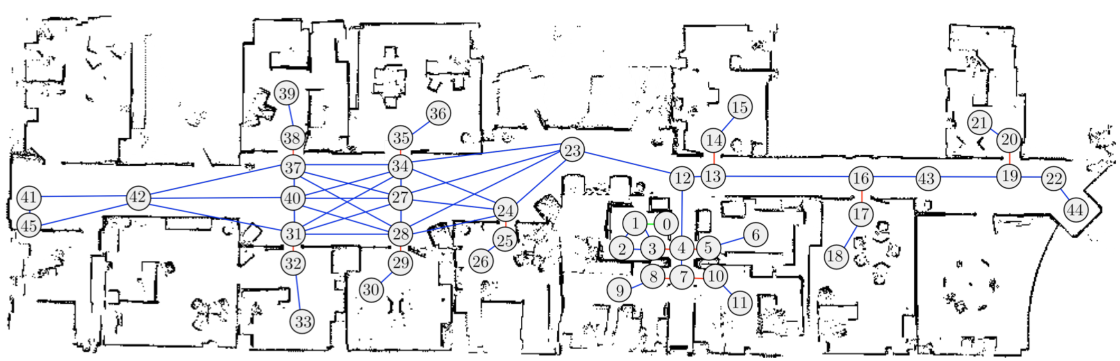


Pareto curve:
x="probability of completing task 1";
y="probability of completing task 2";
z="expected size of successful team"

Application: Robot navigation

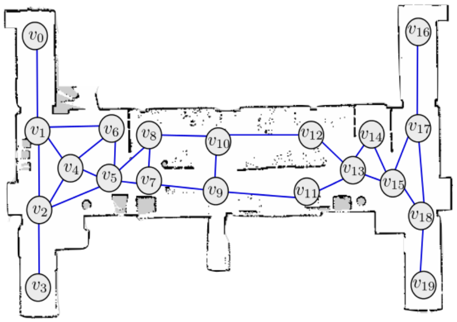
- Robot navigation planning: [IROS'14,IJCAI'15,ICAPS'17,IJRR'18]
  - learnt MDP models navigation through uncertain environment
  - co-safe LTL used to formally specify tasks to be executed by robot
  - synthesise finite-memory strategies to construct plans/controllers
  - ROS module based on PRISM
  - 100s of hrs of autonomous deployment





G4S Technology, Tewkesbury (STRANDS)

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,…
  - generates guarantees on performance
    - QoS guarantees fed into task planning

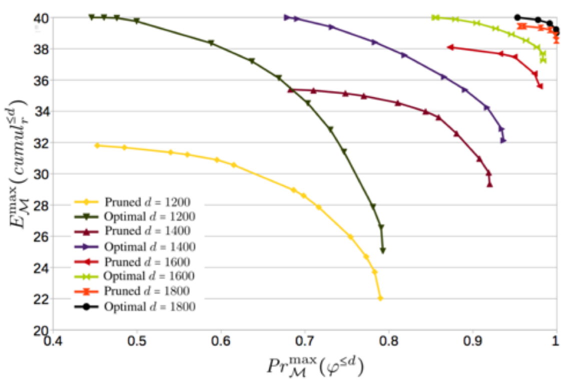
Multi-objective: Partial satisfiability

- Partially satisfiable task specifications
  - e.g. $P_{max=?}$ [¬$zone_3$ U ($room_1$ ∧ (F $room_4$ ∧ F $room_5$)] < 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 function constructed from DFA
  - (distance to accepting states, reward for decreasing distance)
- Encode prioritisation using multi-objective queries:
  - p = $P_{max=?}$ [task]
  - r = multi($R^{prog}_{max=?}$ [C], $P_{>=p}$ [task])
  - multi($R^{time}_{min=?}$ [task], $P_{>=p}$ [task] ∧ $R^{prog}_{>=r}$ [C])
- Or alternatively, using nested value iteration

Multi-obj: Time-bounded guarantees

- Often need probabilistic time-bounded guarantees
  - e.g. "probability of completing tasks within 5 mins is >0.99"
  - but verification techniques for these are less efficient/scalable
  - and often needed in conjunction with secondary objectives

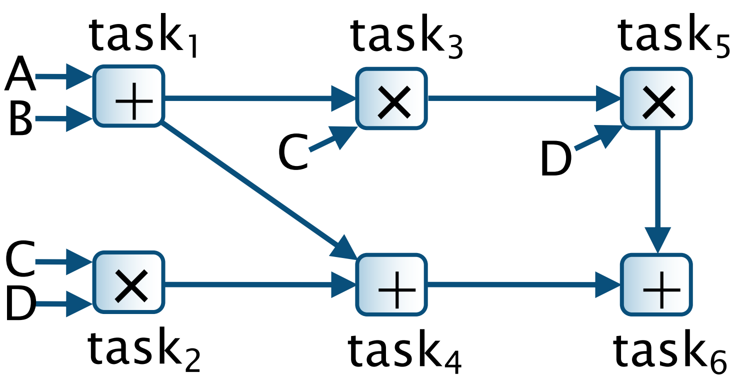
- Efficient generation of time-bounded guarantees [ICAPS'17]
  - implemented in the PRISM model checker



- Key ideas:
  - optimize secondary goal wrt. guarantee
  - two phase verification: initial exploration of Pareto front on coarser untimed model
  - then generate guarantee from pruned model
  - significant gains in scalability

Application: Task-graph scheduling

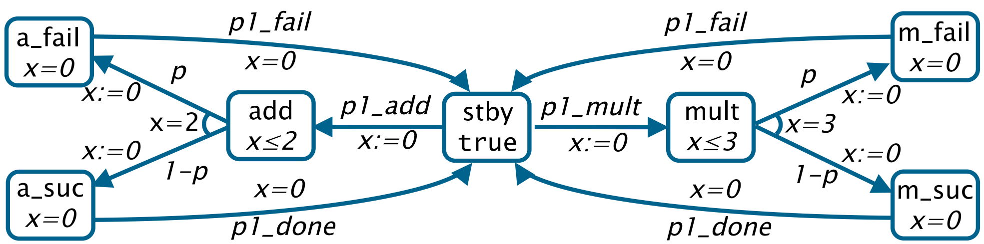
- Task-graph: tasks to complete + dependencies/ordering
  - e.g. for: real-time scheduling, embedded systems controllers
- Simple example: [adapted from BFLM11]
  - evaluate expression
    $D\times(C\times(A+B))+((A+B)+(C\times D))$
  - on multiple processors with differing time/energy usage
  - needs timing information
  - also probabilistic:
    uncertain delays + task failures



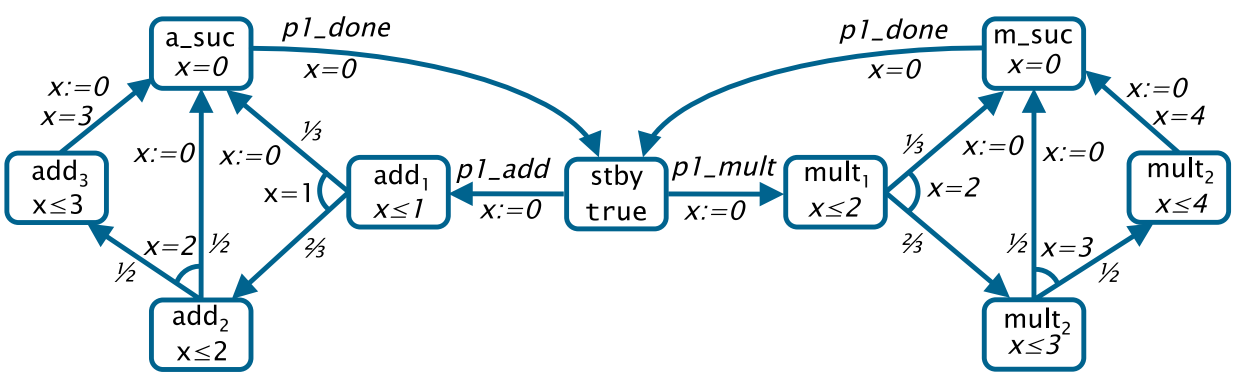
- Modelled using probabilistic timed automata (PTAs)
  - optimal strategy (wrt. time or energy) synthesised in PRISM and converted into optimal scheduling

PTA model components

- Faulty processors
  - third processor P_3: faster, but may fail to execute task

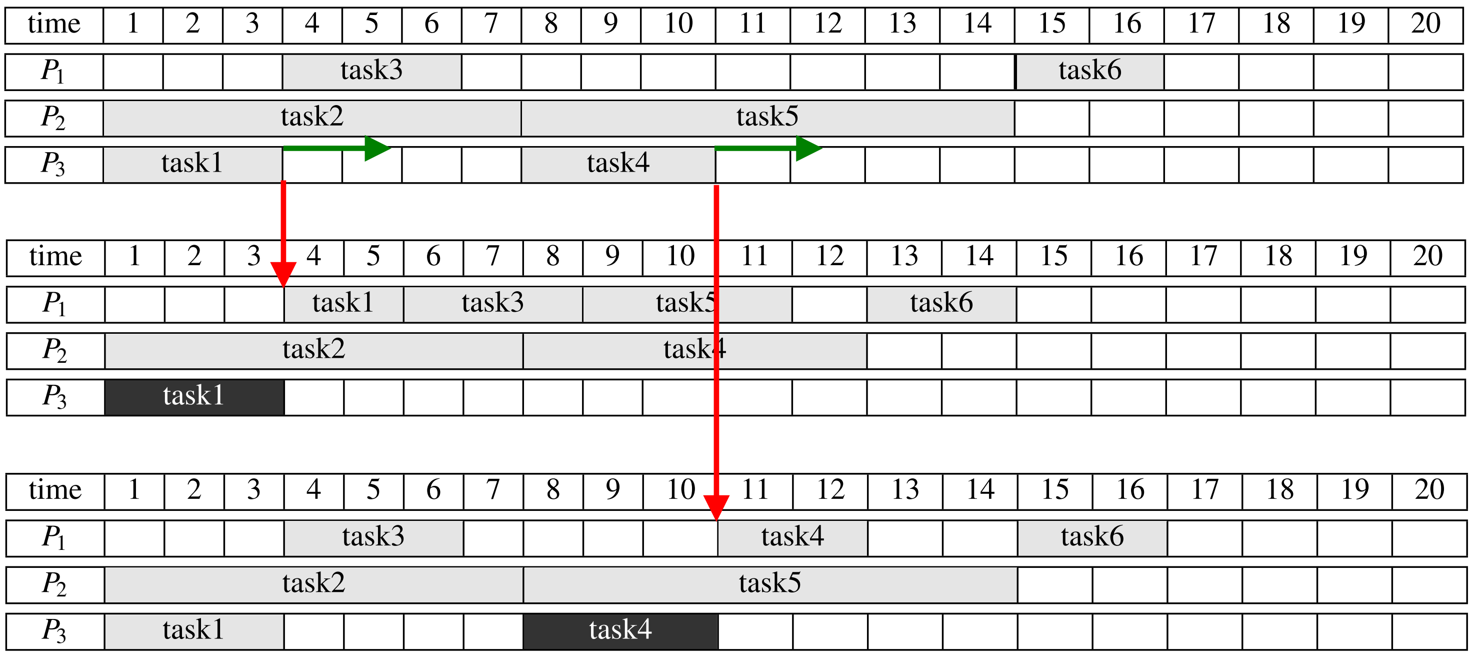


- Probabilistic task execution times
  - simple example: (deterministic) delay of 3 in processor P_1 replaced by distribution: ⅓:2, ⅓:3, ⅓:4



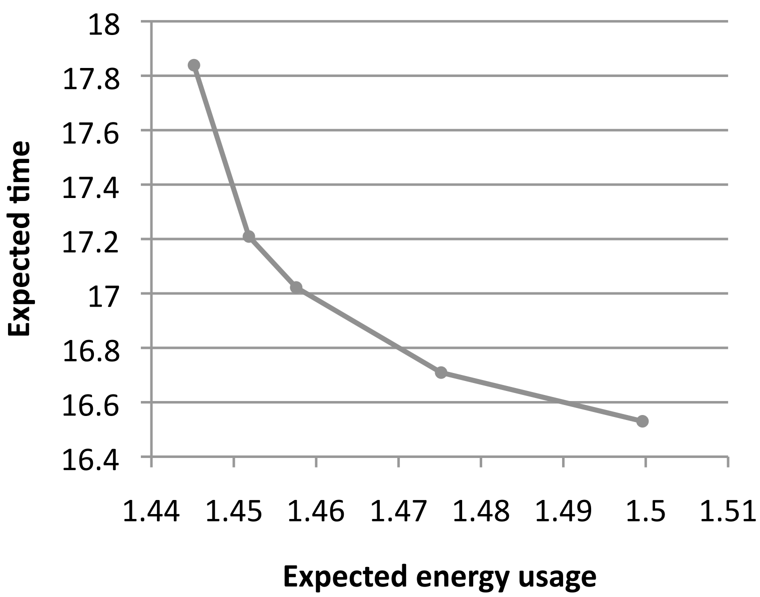
Schedulers (with faulty processor)

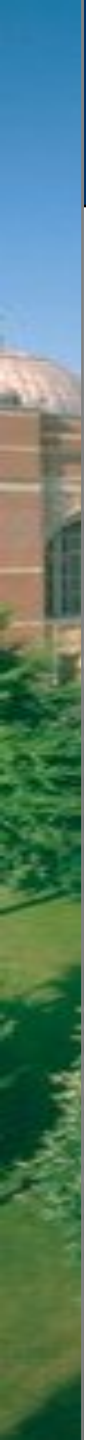
- Example (energy) optimal scheduling:
  – note responses to task failures (on processor P_3)



Multi-objective properties

- Multi-objective controller synthesis
  - explore trade-off between time/energy usage





Overview

- Probabilistic model checking & PRISM
  - Markov decision processes (MDPs)
- Multi-objective probabilistic model checking
  - examples: robot navigation; task scheduling
- Partially observable models
  - POMDPs + real-time variants
  - examples: robot navigation; wireless scheduling
- Stochastic (multi-player) games
  - turn-based & concurrent games
  - examples: energy management, investor models

Partial observability

- Partial observable Markov decision processes (POMDPs)
  - limit strategies ability to view precise states of the MDP
  - we assume an observation function from states to observations
- Optimal strategies
  - resolve actions based on observations only
  - maintain belief state about the true state of the MDP
- Motivation
  - e.g. because robot can only make decisions based on sensors
  - e.g. because scheduler cannot probe state of a component

Partial observability

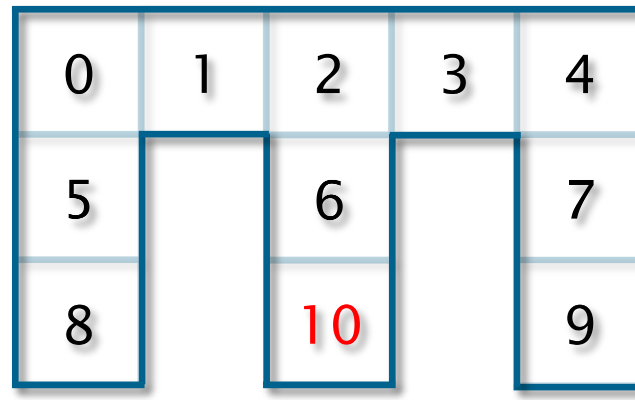
- Developed as an extension of PRISM
  - https://github.com/prismmodelchecker/prism-ext/tree/pomdps
  - PRISM model variables declared as observable/hidden
  - properties in standard PRISM logic

- Implementation on top of PRISM's explicit engine
  - (basic problem is undecidable)
  - computes lower/upper bounds for optimal values and a (possibly sub-optimal) strategy with grid-based approximations
  - applied to a range of case studies (POMDPs up to 60k states)

- Also extended to partially observable PTAs
  - PTA models with hidden (non-clock) variables

Example: Robot maze

- Robot placed uniformly at random in a maze
  - i.e. uncertainty about start state (and subsequent states)
  - 4 actions: north/south/east/west
  - aim to reach target state (10)

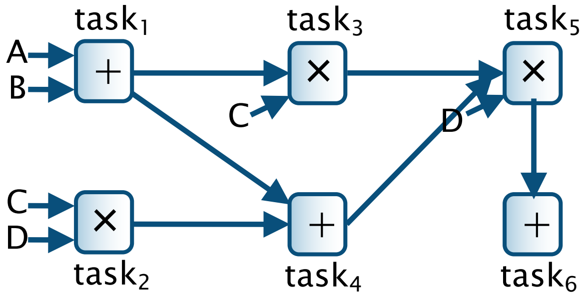


- Partial observability
  - the robot cannot see its current location, only surrounding walls
  - e.g. locations 5,6,7 yield the same observation and are equivalent

- Controller synthesis for $R^{steps}_{min=?}$ [C])
  - optimal (minimum) expected num. steps to reach target is 4.3
  - for the fully observable model (i.e., an MDP), it is 3.9

POMDP/POPTA Case studies

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



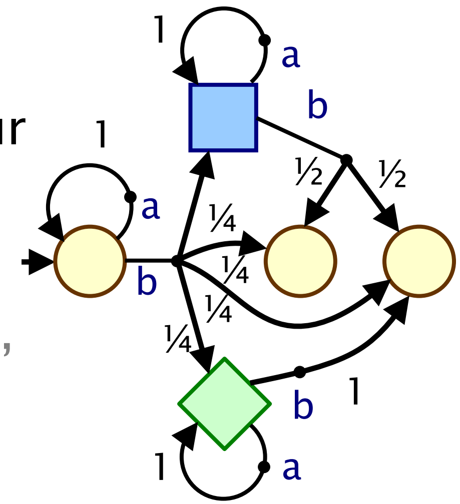
- Wireless network scheduling
  - schedule traffic to number of users/channels
  - packets have hard deadlines (packets not sent by their deadline are dropped) and priorities
  - status of channels is not available (unobservable)
  - generate optimal scheduling of packets, maximising priorities and minimising dropped packets
  - demonstrates that idling is sometimes the optimal choice

Overview

- Probabilistic model checking & PRISM
  - Markov decision processes (MDPs)
- Multi-objective probabilistic model checking
  - examples: robot navigation; task scheduling
- Partially observable models
  - POMDPs + real-time variants
  - examples: robot navigation; wireless scheduling
- Stochastic (multi-player) games
  - turn-based & concurrent games
  - examples: energy management, investor models

Stochastic multi-player games (SMGs)

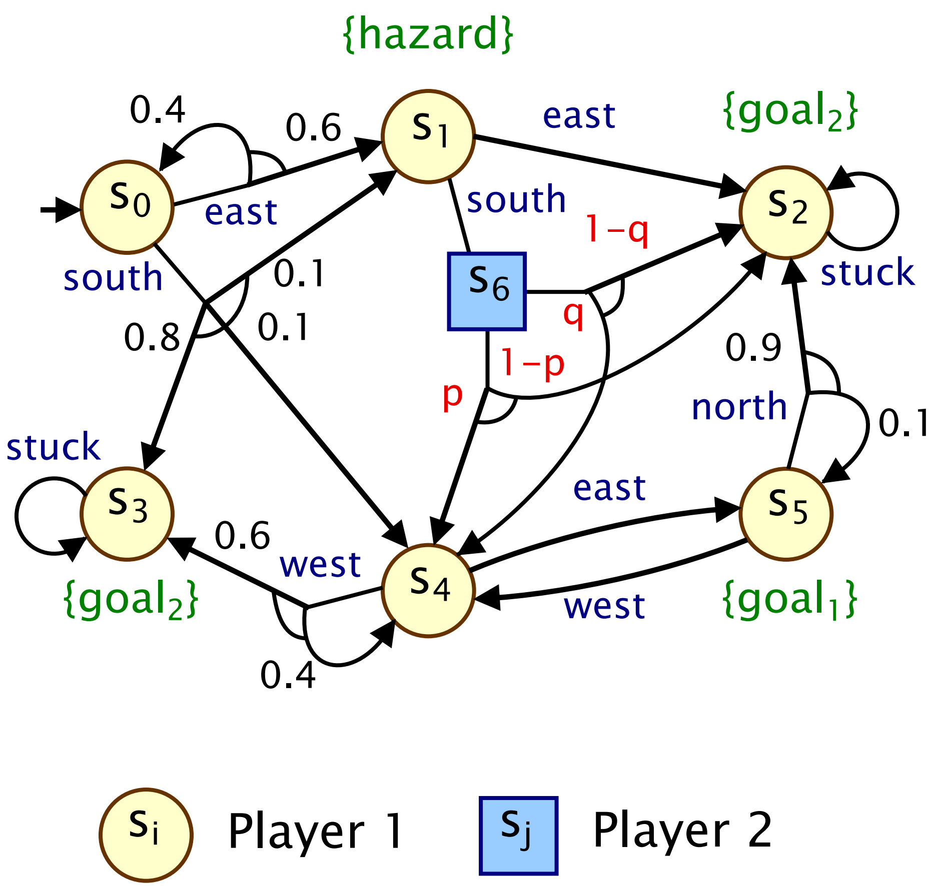
- Stochastic multi-player games
  - competitive/collaborative + stochastic behaviour
  - for now: turn-based (players control states)
  - applications: security (system vs. attacker), controller synthesis (controller vs. environment), distributed algorithms/protocols, …



- Property specifications: rPATL
  - $\langle\langle\{1,2\}\rangle\rangle\ P_{\geq 0.95}\ [\ F^{\leq 45}\ \mathit{done}\ ]$: "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: $\langle\langle C\rangle\rangle\psi$: do there exist strategies for players in C such that, for all strategies of other players, property ψ holds?
- Model checking
  - zero sum properties: analysis reduces to 2-player games
  - PRISM-games: www.prismmodelchecker.org/games

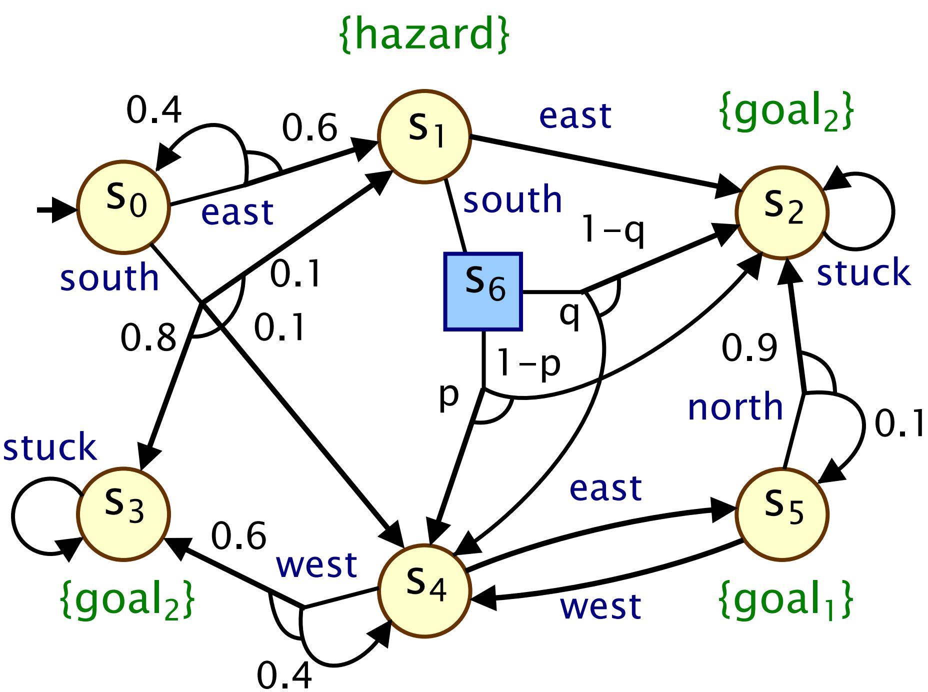
Example – Stochastic games

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



Example – Stochastic games

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



rPATL: ⟨⟨{1}⟩⟩ $P_{max=?}$ [F $goal_1$]

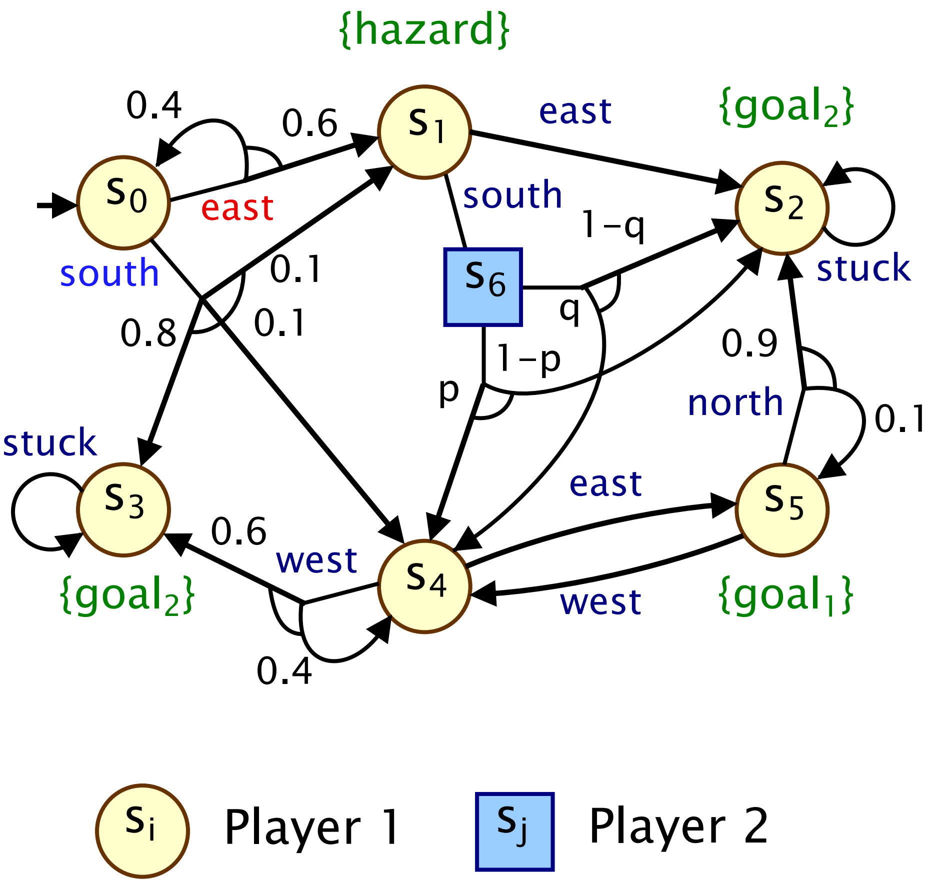
Optimal strategies:
memoryless and deterministic

Computation: graph analysis & numerical approximation

s_i Player 1 s_j Player 2

Example – Stochastic games

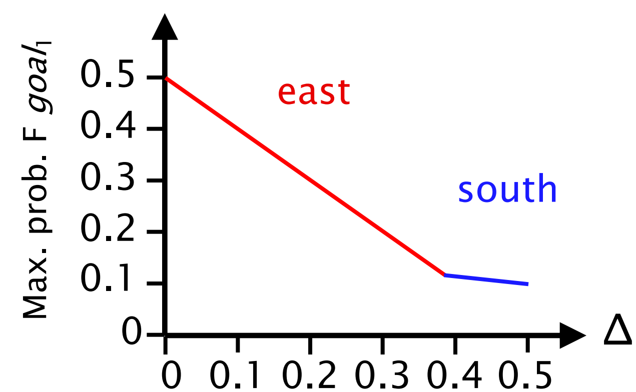
- Two players: 1 (robot controller), 2 (environment)
  - probability of s_1–south→s_4 is in [p,q] = [0.5−Δ, 0.5+Δ]



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

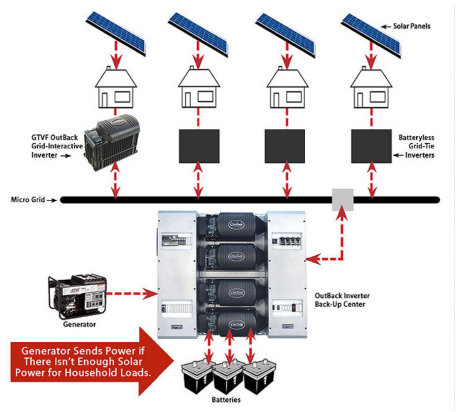
Optimal strategies:
memoryless and deterministic

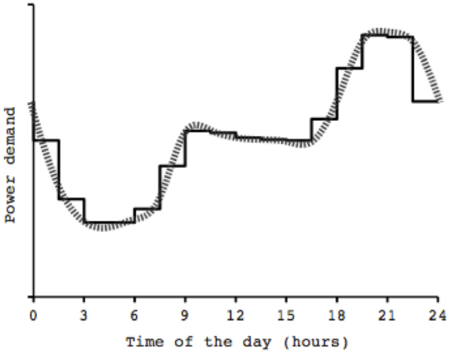
Computation: graph analysis
& numerical approximation



Application: Energy management

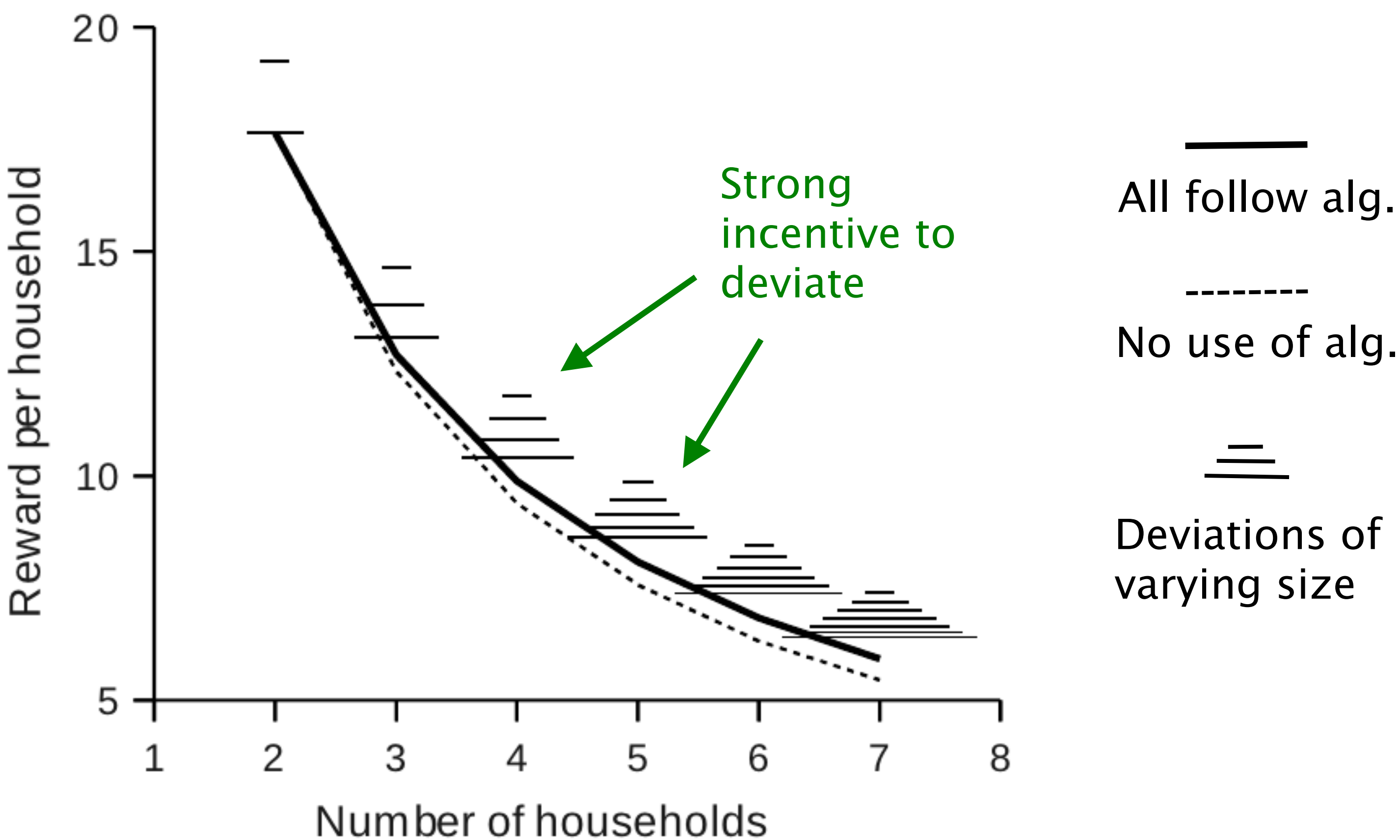
- Energy management protocol for Microgrid
  - randomised demand management protocol
  - random back-off when demand is high
- Original analysis [Hildmann/Saffre'11]
  - protocol increases "value" for clients
  - simulation-based, clients are honest
- Our analysis
  - stochastic multi-player game model
  - clients can cheat (and cooperate)
  - model checking: PRISM-games
  - exposes protocol weakness (incentive for clients to act selfishly
  - propose/verify simple fix using penalties





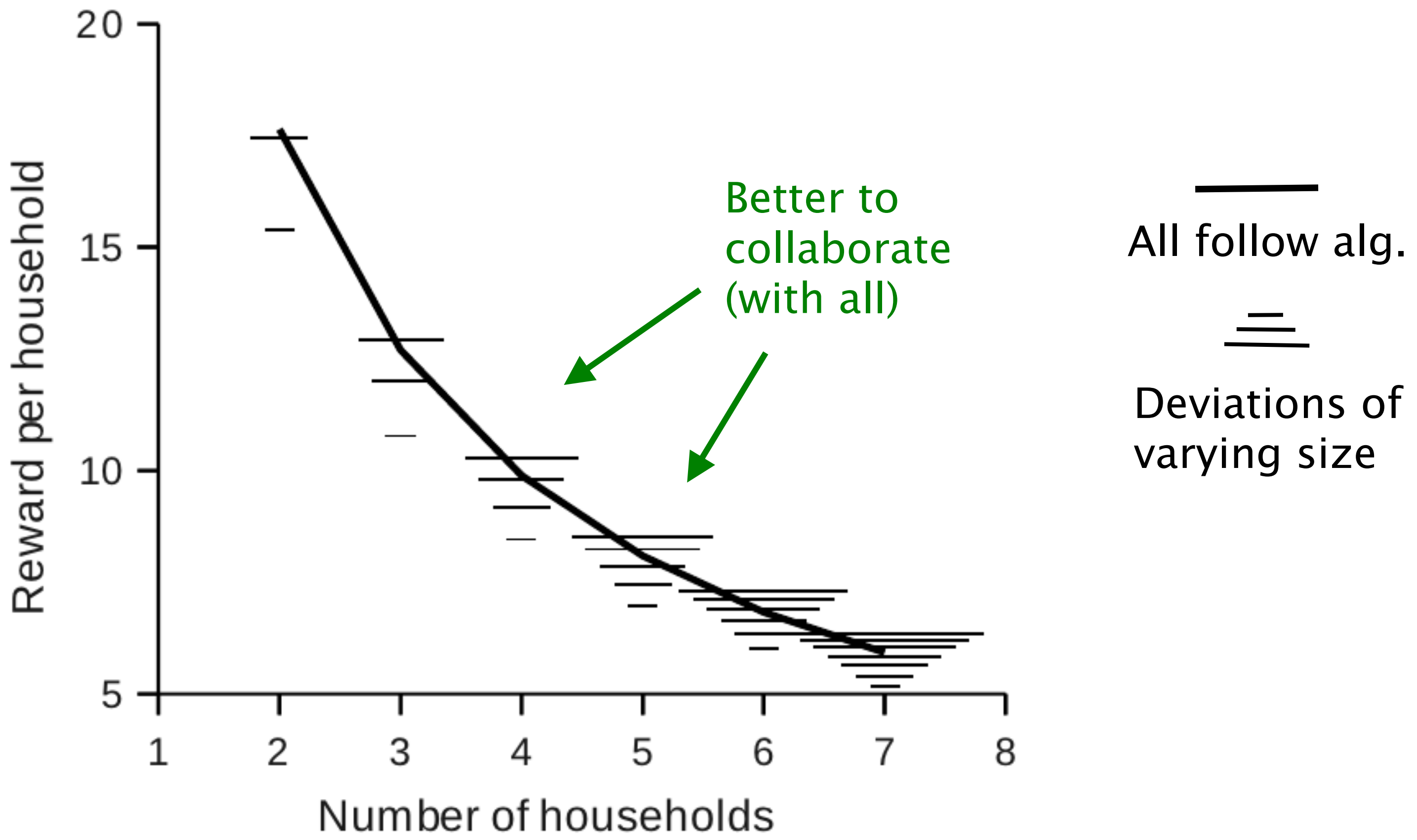
Results: Competitive behaviour

- Expected total value V per household
  - in rPATL: $\langle\langle C \rangle\rangle R^{r_C}_{max=?} [F^0 \; time=max\_time] \; / \; |C|$
  - where r_C is combined rewards for coalition C



Results: Competitive behaviour

- Algorithm fix: simple punishment mechanism
  - distribution manager can cancel some loads exceeding c_{lim}

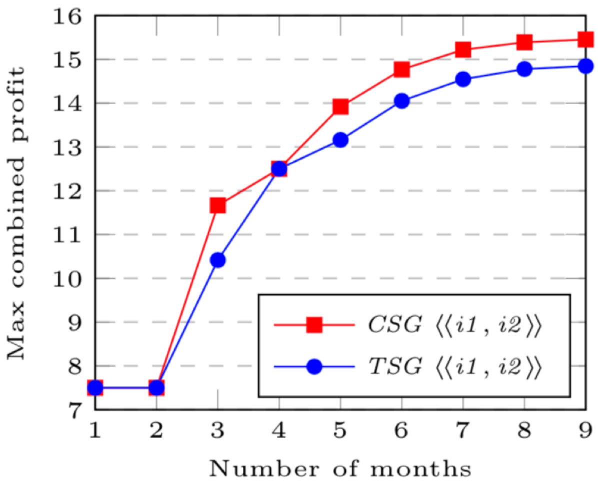


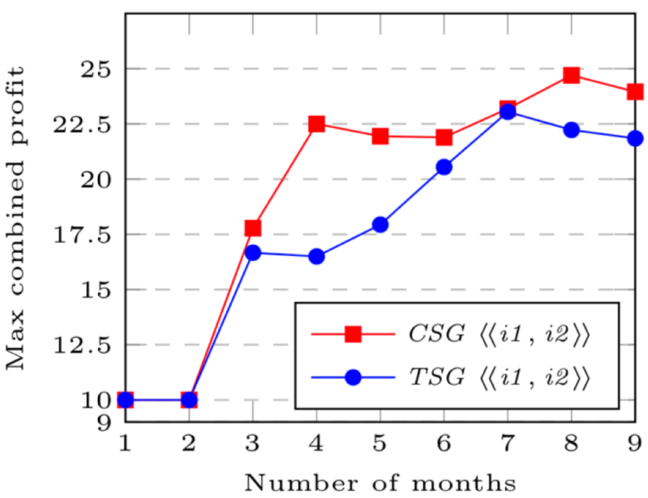
Concurrent stochastic games

- Concurrent stochastic games (CSGs) [QEST'18]
  - players choose actions concurrently
  - jointly determines (probabilistic) successor state
  - $\delta : S\times(A_1\times\ldots\times A_n) \rightarrow Dist(S)$, rather than $\delta_i : S_i\times A_i \rightarrow Dist(S)$
- Modelling & verification implemented in PRISM-games
  - modelling language assumes that each variable is under the control of exactly one module
- Model checking for (variant of) rPATL logic
  - reduces to finding optimal values of 2-player CSGs
  - basic problem is known to be PSPACE
  - we use value iteration + solution of matrix game for each state (LP problem of size |A|, where A = action set)
  - again, need randomised strategies for optimality

Application: CSGs

- Example: futures market investor
  - two investors i_1, i_2, operating in a (stochastic) market
  - market (third player) decides whether to bar investors
- Results (investors maximizing joint profit)
  - with (left) and without (right) fluctuations

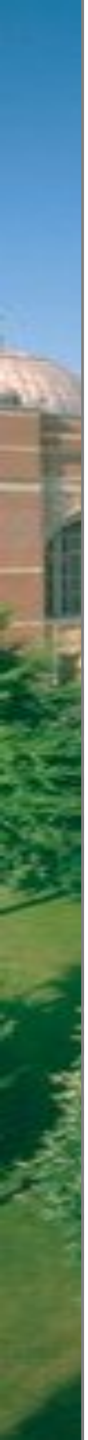




- Other applications: intrusion detection, network protocols

Conclusions

- Probabilistic model checking & PRISM
  - Markov decision processes & related models
- Recent extensions
  - multi-objective model checking
  - partially observable MDPs
  - stochastic games
- Challenges & directions
  - managing model uncertainty + integration with learning
  - partial information/observability: greater efficiency
  - scalability, e.g. symbolic methods, abstraction
  - stochastic games: multi-objective, equilibria, richer logics



Thanks for your attention

More info here:

www.prismmodelchecker.org