ICLA 2021 (virtual) 5th March, 2021

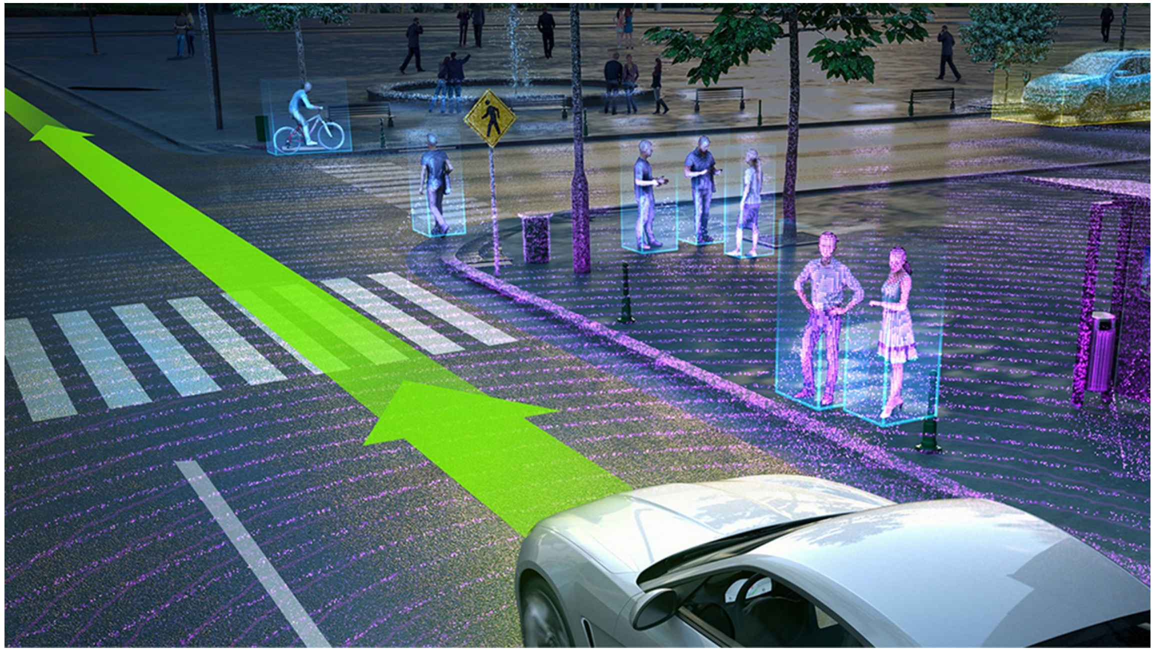
Probabilistic model checking for strategic equilibria-based decision making

Prof. Marta Kwiatkowska

Department of Computer Science
University of Oxford

Software everywhere

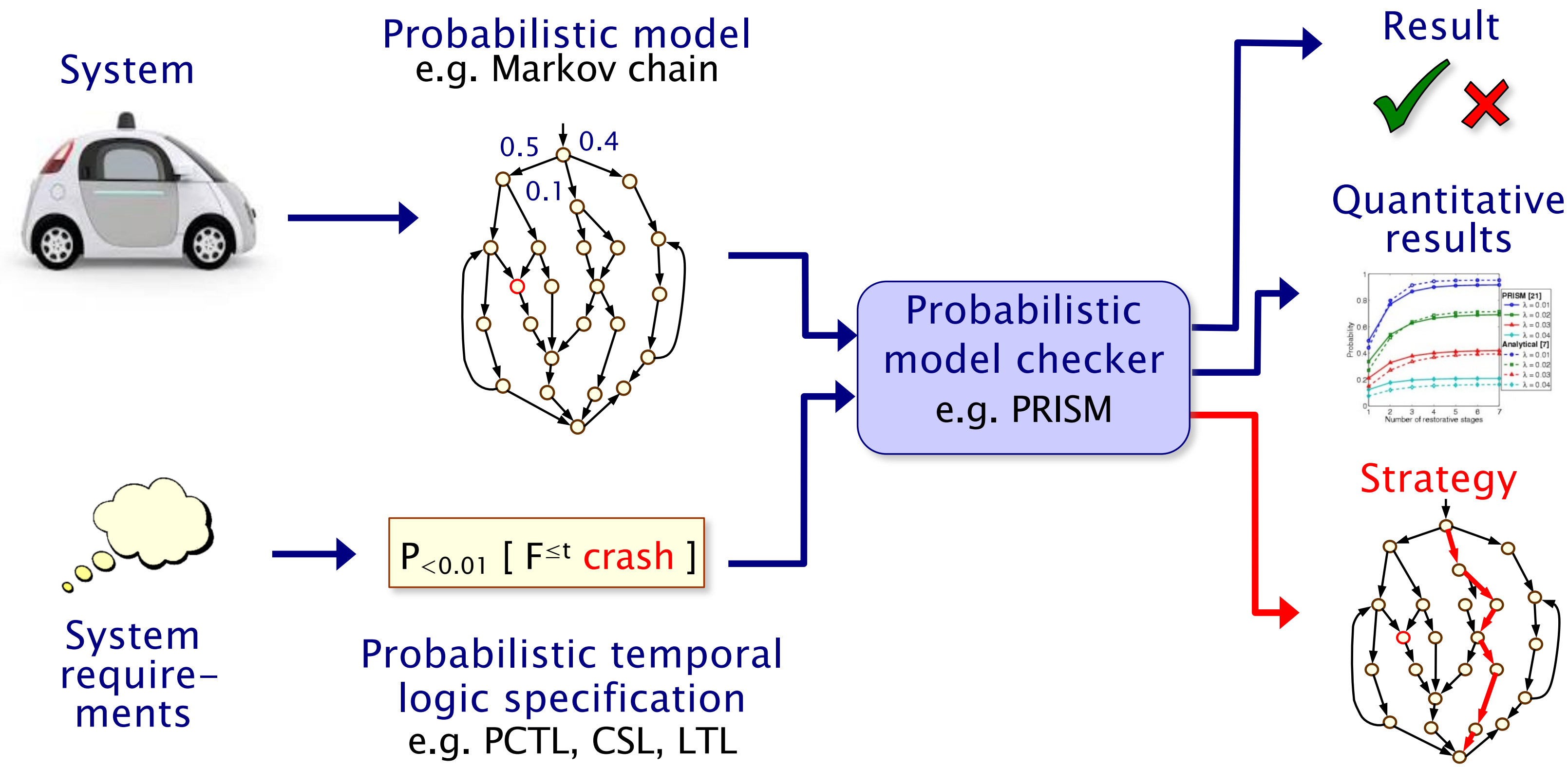
- Users expect: predictability & high integrity in presence of
  - component failure, environmental uncertainty, communication delays, …



- Safety, security, reliability, robustness, … can be expressed probabilistically
  - “the probability of an airbag failing to deploy within 0.02s is less than 0.001”

Probabilistic model checking

Automatic verification and strategy synthesis from temporal logic properties for probabilistic models



System

Probabilistic model
e.g. Markov chain

System require-ments

$P_{<0.01}$ [$F^{\leq t}$ crash]

Probabilistic temporal logic specification
e.g. PCTL, CSL, LTL

Probabilistic model checker
e.g. PRISM

Result

✓ ✗

Quantitative results

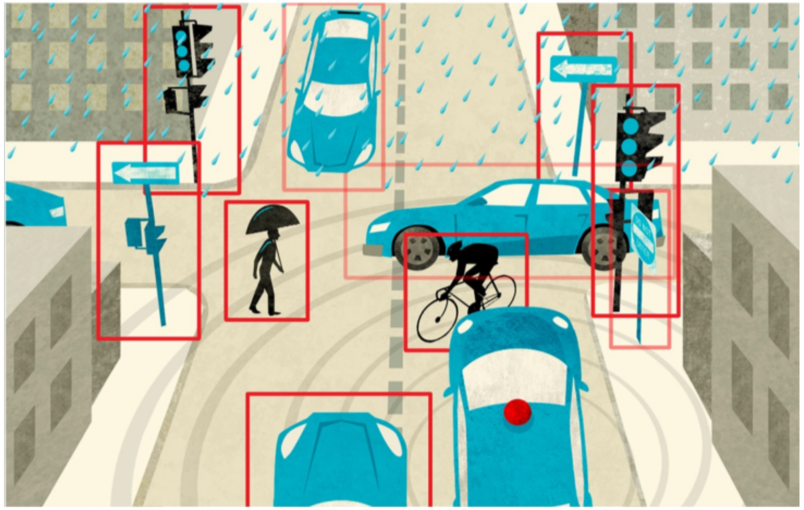
Strategy

Tool support: PRISM

- First algorithms proposed in 1980s
  - algorithms [Vardi, Courcoubetis, Yannakakis, …]
  - [Hansson, Jonsson, de Alfaro] & first implementations
- 2001: general purpose tools released
  - PRISM: efficient extensions of symbolic model checking [Kwiatkowska, Norman, Parker, and many more, …]
- Now mature area, of industrial relevance, new model checkers, tool competition
  - PRISM successfully used by non-experts in many domains
    - distributed algorithms, communication protocols, security protocols, biological systems, quantum cryptography, planning, robotics, …
  - genuine flaws found and corrected in real-world systems
  - www.prismmodelchecker.org

But which modelling abstraction?

- Complex decisions!
  - human and artificial agents
  - distinct goals
  - autonomy
  - competitive/collaborative behaviour
  - context



- Natural to adopt a game-theoretic view
  - need to account for the uncontrollable behaviour of components, possibly with differing/opposing goals
  - in addition to controllable events
- Many occurrences in practice
  - e.g. decision making in economics, security, energy management, …

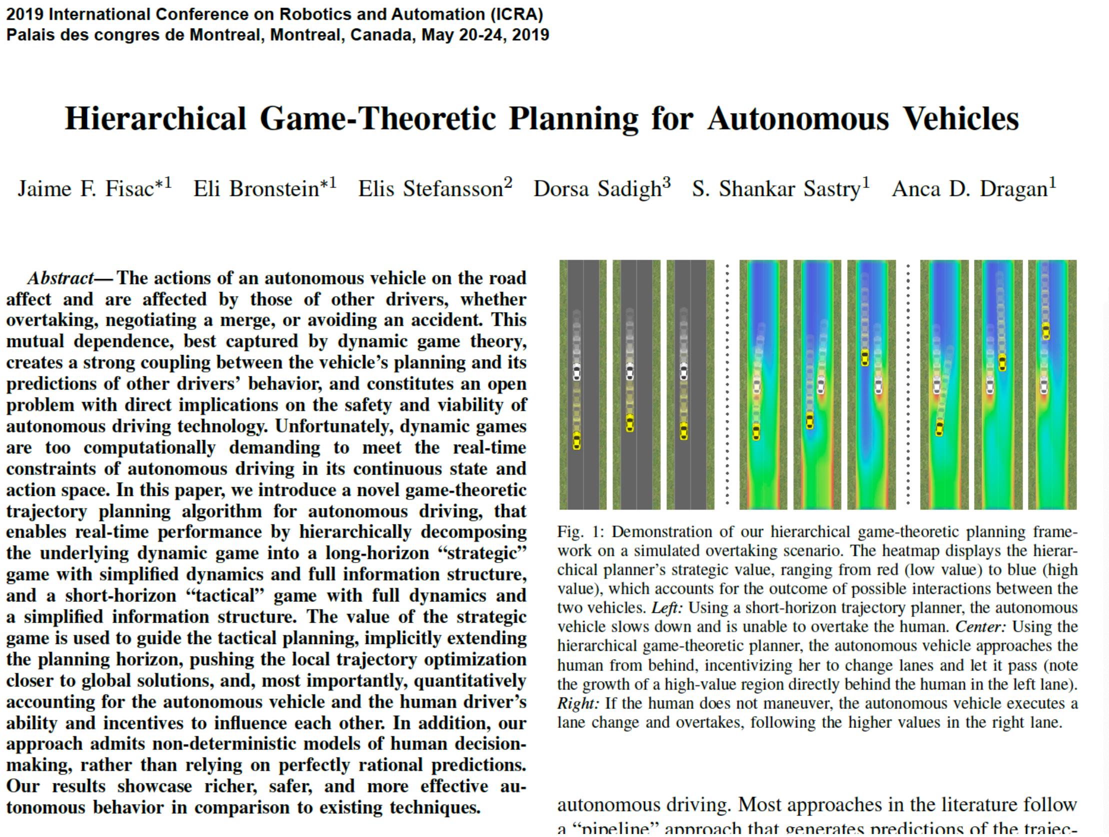
Why games?

- Games serve as abstractions for negotiation, strategic game playing and incentives to achieve more effective behaviour
  - virtual vs human agents, turn-based vs concurrent, zerosum vs non-zerosum, rational vs non-rational behaviours, individual vs social gain, …

- Relevant for a multitude of autonomous and AI scenarios
  - e.g. automated decisions, resource sharing, distributed coordination protocols, virtual assistants, etc
  - even autonomous driving…

Driving as a game-theoretic problem

- Merging into traffic difficult for autonomous cars
- Human drivers are not behaving rationally!
- Here dynamic games, and interplay between long-horizon strategic game playing with short-horizon tactical moves, plus incentives



2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Hierarchical Game-Theoretic Planning for Autonomous Vehicles

Jaime F. Fisac*[1] Eli Bronstein*[1] Elis Stefansson[2] Dorsa Sadigh[3] S. Shankar Sastry[1] Anca D. Dragan[1]

***Abstract*— The actions of an autonomous vehicle on the road affect and are affected by those of other drivers, whether overtaking, negotiating a merge, or avoiding an accident. This mutual dependence, best captured by dynamic game theory, creates a strong coupling between the vehicle's planning and its predictions of other drivers' behavior, and constitutes an open problem with direct implications on the safety and viability of autonomous driving technology. Unfortunately, dynamic games are too computationally demanding to meet the real-time constraints of autonomous driving in its continuous state and action space. In this paper, we introduce a novel game-theoretic trajectory planning algorithm for autonomous driving, that enables real-time performance by hierarchically decomposing the underlying dynamic game into a long-horizon "strategic" game with simplified dynamics and full information structure, and a short-horizon "tactical" game with full dynamics and a simplified information structure. The value of the strategic game is used to guide the tactical planning, implicitly extending the planning horizon, pushing the local trajectory optimization closer to global solutions, and, most importantly, quantitatively accounting for the autonomous vehicle and the human driver's ability and incentives to influence each other. In addition, our approach admits non-deterministic models of human decision-making, rather than relying on perfectly rational predictions. Our results showcase richer, safer, and more effective autonomous behavior in comparison to existing techniques.**

Fig. 1: Demonstration of our hierarchical game-theoretic planning framework on a simulated overtaking scenario. The heatmap displays the hierarchical planner's strategic value, ranging from red (low value) to blue (high value), which accounts for the outcome of possible interactions between the two vehicles. *Left:* Using a short-horizon trajectory planner, the autonomous vehicle slows down and is unable to overtake the human. *Center:* Using the hierarchical game-theoretic planner, the autonomous vehicle approaches the human from behind, incentivizing her to change lanes and let it pass (note the growth of a high-value region directly behind the human in the left lane). *Right:* If the human does not maneuver, the autonomous vehicle executes a lane change and overtakes, following the higher values in the right lane.

autonomous driving. Most approaches in the literature follow a "pipeline" approach that generates predictions of the trajec-

This lecture…

- Introduce stochastic multi-player games
  - modelling abstraction for competitive/cooperative behaviour, in adversarial environments
  - stochasticity to model e.g. failure, sensor uncertainty
  - turn-based and concurrent games
- Property specification: rPATL (based on PCTL and ATL)
  - zerosum and equilibria
  - model checking and strategy synthesis
  - case studies
- Tool support: PRISM-games 3.0
- Challenges and future directions

PRISM-games 3.0: Stochastic Game Verification with Concurrency, Equilibria and Time, Kwiatkowska et al., In *Proc* CAV 2020

Stochastic multi-player games (SMGs)

- A stochastic game involves
  - multiple players (competitive or collaborative behaviour)
  - nondeterminism (decisions, control, environment)
  - probability (failures, noisy sensors, randomisation)

- Game variants
  - turn-based vs concurrent, zero sum vs distinct goals
  - here, complete information games

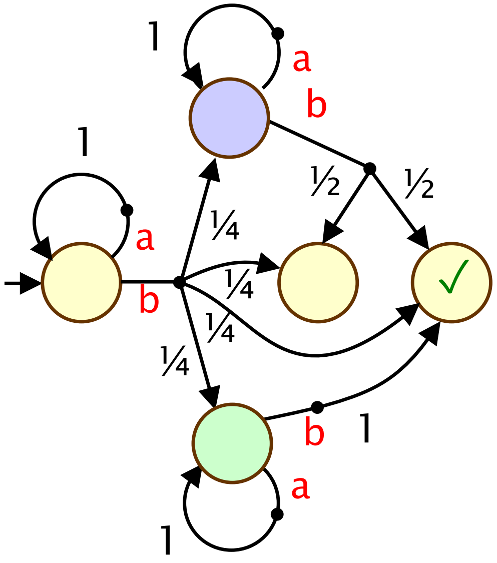
- Widely studied, esp. algorithmic complexity, many applications
  - autonomous traffic (risk averse vs risk taking)
  - distributed coordination (selfish agents vs unselfish)
  - controller synthesis (system vs. environment)
  - security (defender vs. attacker)

Stochastic multi-player games

- Stochastic multi-player game (SMGs)
  - multiple players + nondeterminism + probability
  - generalisation of MDPs: each state controlled by unique player

- A (turn-based) SMG is a tuple $(\Pi, S, \langle S_i \rangle_{i \in \Pi}, A, \Delta, L)$:
  - Π is a set of n players
  - S is a (finite) set of states
  - $\langle S_i \rangle_{i \in \Pi}$ is a partition of S
  - A is a set of action labels
  - $\Delta : S \times A \to Dist(S)$ is a (partial) transition probability function
  - $L : S \to 2^{AP}$ is a labelling with atomic propositions from AP

- NB concurrent games coming later



Strategies, probabilities & rewards

- Strategy for player i: resolves choices in S_i states
  - based on execution history, i.e. $\sigma_i : (SA)^*S_i \rightarrow Dist(A)$
  - can be: deterministic (pure), randomised, memoryless, finite-memory, …
  - Σ_i denotes the set of all strategies for player i

- Strategy profile: strategies for all players: $\sigma=(\sigma_1,\ldots,\sigma_n)$
  - probability measure Pr_s^σ over (infinite) paths from state s
  - expectation $E_s^\sigma(X)$ of random variable X over Pr_s^σ

- Rewards (or costs)
  - non-negative integers on states/transitions
  - e.g. elapsed time, energy consumption, number of packets lost, net profit, …

Property specification: rPATL

- rPATL (reward probabilistic alternating temporal logic)
  - branching-time temporal logic for SMGs
- CTL, extended with:
  - coalition operator $\langle\langle C\rangle\rangle$ of ATL
  - probabilistic operator P of PCTL
  - generalised (expected) reward operator R from PRISM
- In short:
  - zero-sum, probabilistic reachability + expected total reward
- Example:
  - $\langle\langle\{1,3\}\rangle\rangle\ P_{<0.01}\ [\ F^{\leq 10}\ \text{error}\ ]$
  - “players 1 and 3 have a strategy to ensure that the probability of an error occurring within 10 steps is less than 0.01, regardless of the strategies of other players”

rPATL syntax/semantics

- Syntax:

$$\phi ::= \text{true} \mid a \mid \neg\phi \mid \phi \wedge \phi \mid \langle\langle C \rangle\rangle \mathrm{P}_{\bowtie q}[\psi] \mid \langle\langle C \rangle\rangle \mathrm{R}^{r}_{\bowtie x}\ [\rho]$$

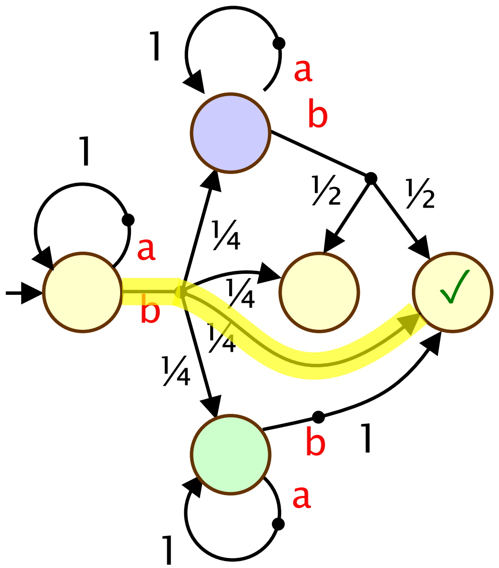
$$\psi ::= \mathrm{X}\ \phi \mid \phi\ \mathrm{U}^{\leq k}\ \phi \mid \phi\ \mathrm{U}\ \phi$$

$$\rho ::= \mathrm{I}^{=k} \mid \mathrm{C}^{\leq k} \mid \mathrm{F}\ \phi$$

- where:
  - $a \in AP$ is an atomic proposition, $C \subseteq N$ is a coalition of players, $\bowtie\ \in \{\leq,<,>,\geq\}$, $q \in [0,1] \cap \mathbb{Q}$, $x \in \mathbb{Q}_{\geq 0}$, $k \in \mathbb{N}$, r is a reward structure

- Semantics:
- e.g. P operator: $s \models \langle\langle C \rangle\rangle \mathrm{P}_{\bowtie q}[\psi]$ iff:
  - "there exist strategies for players in coalition C such that, for all strategies of the other players, the probability of path formula ψ being true from state s satisfies ⋈ q"

Examples



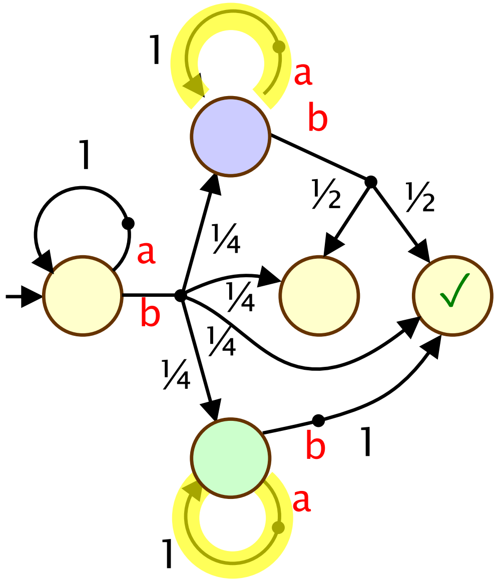
⟨⟨🟡⟩⟩$\mathrm{P}_{\geq 1/4}$[F ✓]

true in initial state

⟨⟨🟡⟩⟩$\mathrm{P}_{\geq 1/3}$ [F ✓]

⟨⟨🟡,🔵⟩⟩$\mathrm{P}_{\geq 1/3}$ [F ✓]

Examples



$\langle\langle$○$\rangle\rangle P_{\geq 1/4}$[F ✓]

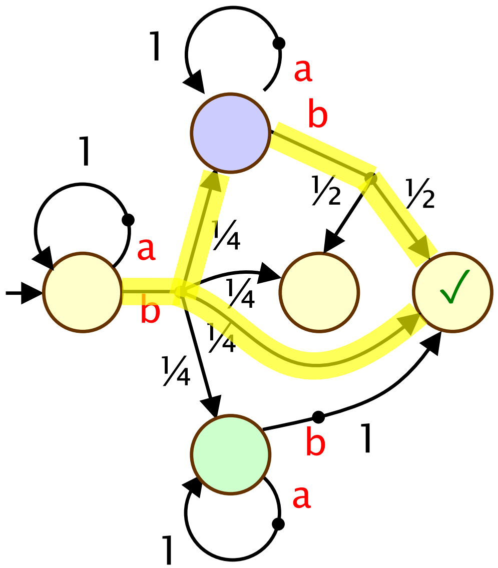
true in initial state

$\langle\langle$○$\rangle\rangle P_{\geq 1/3}$ [F ✓]

false in initial state

$\langle\langle$○,●$\rangle\rangle P_{\geq 1/3}$ [F ✓]

Examples



$\langle\langle$🟡$\rangle\rangle P_{\geq 1/4}[\ F$ ✓ $]$

true in initial state

$\langle\langle$🟡$\rangle\rangle P_{\geq 1/3}\ [\ F$ ✓ $]$

false in initial state

$\langle\langle$🟡,🔵$\rangle\rangle P_{\geq 1/3}\ [\ F$ ✓ $]$

true in initial state

Verification and strategy synthesis

- The verification problem is:
  - Given a game G and rPATL property ϕ, does G satisfy ϕ?
- e.g. $\langle\langle C\rangle\rangle P_{\bowtie q}[\psi]$ is true in state s of G iff:
  - in coalition game G_C:
  - $\exists \sigma_1 \in \Sigma_1$ such that $\forall \sigma_2 \in \Sigma_2$. $Pr_s^{\sigma_1,\sigma_2}(\psi) \bowtie q$

- The synthesis problem is:
  - Given a game G and a coalition property ϕ, find, if it exists, a coalition strategy σ that is a witness to G satisfying ϕ

- Reduce to computing optimal values and winning strategies in 2-player games
  - e.g. $\langle\langle C\rangle\rangle P_{\geq q}[\psi] \Leftrightarrow \sup_{\sigma_1 \in \Sigma_1} \inf_{\sigma_2 \in \Sigma_2} Pr_s^{\sigma_1,\sigma_2}(\psi) \geq q$
  - complexity NP ∩ coNP (this fragment), cf P for MDPs

Automatic Verification of Competitive Stochastic Systems, Chen et al., In *Proc* TACAS 2012

Example: Probabilistic reachability

- E.g. $\langle\langle C \rangle\rangle P_{\geq q}[\ F\ \phi\ ]$: max/min reachability probabilities
  - compute $\sup_{\sigma_1 \in \Sigma_1} \inf_{\sigma_2 \in \Sigma_2} Pr_s^{\sigma_1,\sigma_2}(F\ \phi)$ for all states s
  - deterministic memoryless strategies suffice

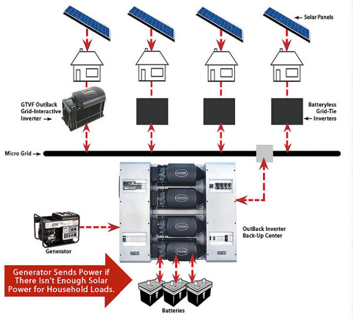
- Value $p(s)$ for state s is least fixed point of:

$$p(s) = \begin{cases} 1 & \text{if } s \in Sat(\phi) \\ \max_{a \in A(s)} \Sigma_{s' \in S}\ \delta(s,a)(s') \cdot p(s') & \text{if } s \in S_1 \backslash Sat(\phi) \\ \min_{a \in A(s)} \Sigma_{s' \in S}\ \delta(s,a)(s') \cdot p(s') & \text{if } s \in S_2 \backslash Sat(\phi) \end{cases}$$

- Computation (value iteration):
  - start from zero, propagate probabilities backwards
  - guaranteed convergence; apply “usual” termination criteria

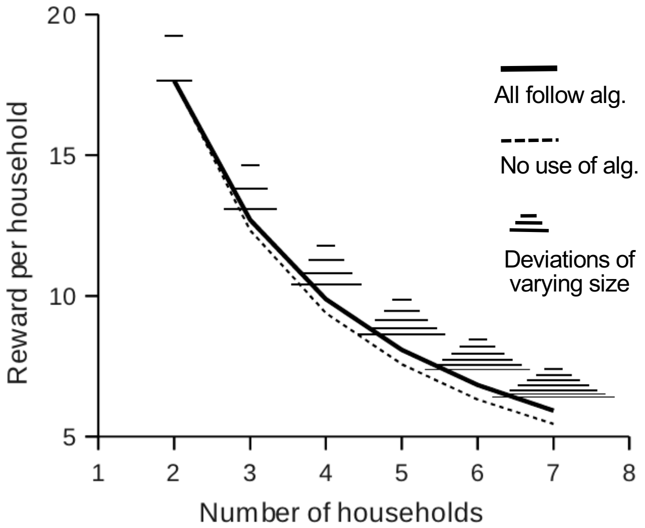
Case study: Energy management

- Energy management protocol for Microgrid
  - Microgrid: local energy management
  - randomised demand management protocol [Hildmann/Saffre'11]
  - probability: randomisation, demand model, …


Solar Panels
GTVF OutBack Grid-Interactive Inverter
Batteryless Grid-Tie Inverters
Micro Grid
OutBack Inverter Back-Up Center
Generator
Generator Sends Power if There Isn't Enough Solar Power for Household Loads.
Batteries

- Existing analysis
  - simulation-based
  - assumes all clients are unselfish

- Our analysis
  - stochastic multi-player game
  - clients can cheat (and cooperate)
  - exposes protocol weakness
  - propose/verify simple fix



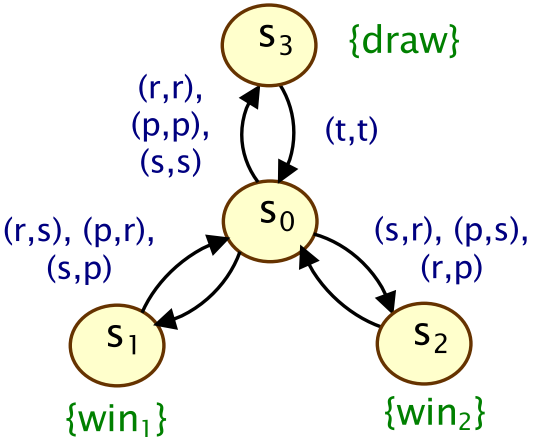
Automatic Verification of Competitive Stochastic Systems, Chen et al., In *Proc* TACAS 2012

Concurrent stochastic games

- Concurrent stochastic games (CSGs)
  - players choose actions concurrently
  - jointly determines (probabilistic) successor state
  - generalises turn-based stochastic games
- Key motivation:
  - more realistic model of components operating concurrently, making action choices without knowledge of others
- Formally
  - set of n players N, state space S, actions A_i for player i
  - transition probability function $\delta : S \times A \rightarrow Dist(S)$
  - where $A = (A_1 \cup \{\perp\}) \times \ldots \times (A_n \cup \{\perp\})$
  - strategies $\sigma_i : (SA)^*S_i \rightarrow Dist(A_i)$, strategy profiles $\sigma=(\sigma_1,\ldots,\sigma_n)$
  - probability measure Pr_s^{σ}, expectations $E_s^{\sigma}(X)$

Example CSG: rock-paper-scissors

- Rock-paper-scissors game
  - 2 players repeatedly draw rock (r), paper (p), scissors (s), then restart the game (t)
  - rock > scissors, paper > rock, scissors > paper,
  - otherwise draw
- Example CSG
  - 2 players: N={1,2}
  - $A_1 = A_2 = \{r,p,s,t\}$
  - NB: no probabilities here



Matrix games

- Matrix games solved in each state
  - finite, one-shot, 2-player, zero-sum games
  - utility function $u_i : A_1 \times A_2 \rightarrow \mathbb{R}$ for each player i
  - represented by matrix Z where $z_{ij} = u_1(a_i,b_j) = -u_2(a_i,b_j)$
- Example:
  - one round of rock-paper-scissors

$$Z = \begin{array}{c|ccc} & r & p & s \\ \hline r & 0 & -1 & 1 \\ p & 1 & 0 & -1 \\ s & -1 & 1 & 0 \end{array}$$

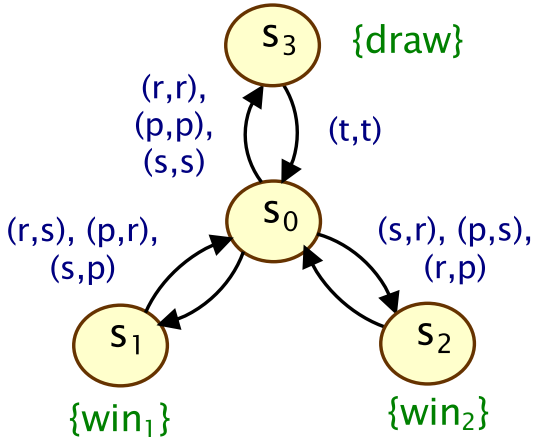
- Optimal (player 1) strategy via LP solution (minimax):
  - compute value val(Z): maximise value v subject to:
  - $v \leq x_p - x_s$
    $v \leq x_s - x_{r,}$
    $v \leq x_s - x_p$
    $x_r + x_p + x_s = 1$
    $x_r \geq 0_, \ x_p \geq 0_, \ x_s \geq 0$

Optimal strategy (randomised):
$(x_r,x_p,x_s) = (⅓,⅓,⅓)$

rPATL for CSGs

- We use the same logic rPATL as for SMGs

- Examples for rock-paper-scissors game:
  - $\langle\langle 1 \rangle\rangle \, P_{\geq 1} [\, F \, win_1 \,]$ – player 1 can ensure it eventually wins a round of the game with probability 1
  - $\langle\langle 2 \rangle\rangle \, P_{max=?} [\, \neg win_1 \; U \; win_2 \,]$ – the maximum probability with which player 2 can ensure it wins before player 1
  - $\langle\langle 1 \rangle\rangle \, R^{utility_1}_{max=?} [\, C^{\leq 2K} \,]$ – the maximum expected utility player 1 can ensure over K rounds (utility = 1/0/−1 for win/draw/lose)



rPATL model checking for CSGs

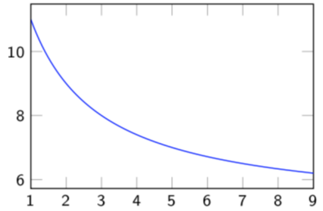
- Extends model checking algorithm for SMGs [QEST'18]
  - key ingredients are solution of (zero-sum) 2-player CSGs

- E.g. $\langle\langle C\rangle\rangle P_{\geq q}[\ F\ \phi\ ]$: max/min reachability probabilities
  - compute $\sup_{\sigma_1 \in \Sigma_1} \inf_{\sigma_2 \in \Sigma_2} Pr_s^{\sigma_1,\sigma_2}(F\ \phi)$ for all states s
  - note that optimal strategies are now randomised
  - solution of the 2-player CSG is in PSPACE
  - we use a value iteration approach

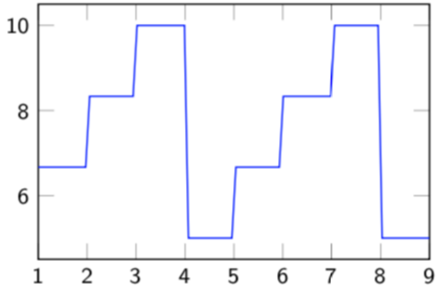
- Value $p(s)$ for state s is least fixed point of:
  - $p(s) = 1$ if $s \in Sat(\phi)$ and otherwise $p(s) = val(Z)$ where:
  - Z is the matrix game with $z_{ij} = \Sigma_{s' \in S}\ \delta(s,(a_i,b_j))(s') \cdot p(s')$
  - so each iteration requires solution of a matrix game for each state (LP problem of size $|A|$, where A = action set)

Automatic Verification of Concurrent Stochastic Games, Kwiatkowska et al., In *Proc* QEST 2018

Case study: Future markets investor



- Model of interactions between:
  - stock market, evolves stochastically
  - two investors i_1, i_2 decide when to invest
  - market decides whether to bar investors



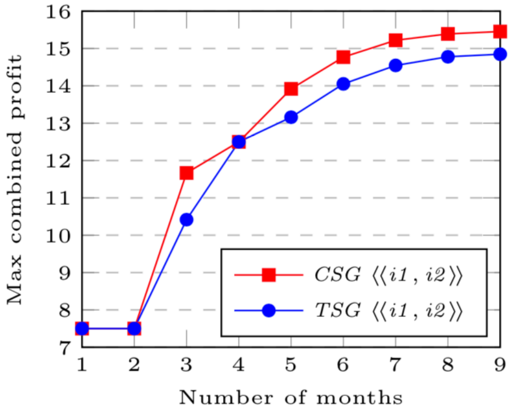
- Modelled as a 3-player CSG
  - extends simpler model originally from [McIver/Morgan'07]
  - investing/barring decisions are simultaneous
  - profit reduced for simultaneous investments
  - market cannot observe investors' decisions

- Analysed with rPATL model checking & strategy synthesis
  - distinct profit models considered: 'normal market', 'later cash-ins' and 'later cash-ins with fluctuation'
  - comparison between turn-based and concurrent game models

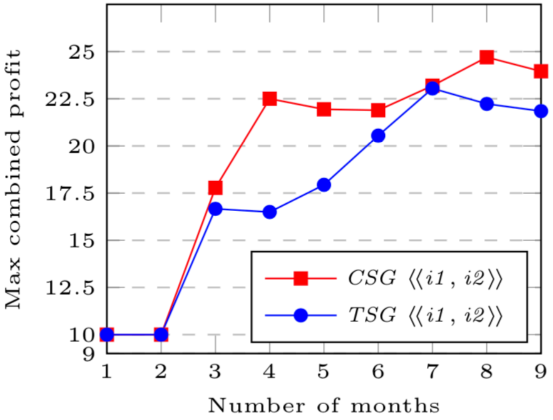
Case study: Future markets investor

- Example rPATL queries:
  - $\langle\langle \text{investor}_1 \rangle\rangle\ \mathrm{R}^{\text{profit}_1}_{\text{max}=?}\ [\ \mathrm{F}\ \text{finished}_1\ ]$
  - $\langle\langle \text{investor}_1, \text{investor}_2 \rangle\rangle\ \mathrm{R}^{\text{profit}_{1,2}}_{\text{max}=?}\ [\ \mathrm{F}\ \text{finished}_{1,2}\ ]$
  - i.e. maximising individual/joint profit

- Results (joint profit) – cooperation pays off in a concurrent game model
  - optimal (randomised) investment strategies synthesised



with fluctuations



without

Multiple objectives: Nash equilibria

- Now consider distinct objectives X_i for each player i
  - i.e., no longer restricted to zero-sum goals

- We use Nash equilibria (NE)
  - no incentive for any player to unilaterally change strategy
  - more precisely subgame-perfect ϵ-Nash equilibrium
  - a strategy profile $\sigma=(\sigma_{1,\dots,}\sigma_n)$ for a CSG is a subgame-perfect ϵ-Nash equilibrium for objectives $X_1,\dots,X_n$ iff:

    $E_s^{\sigma}(X_i) \geq \sup \{ E_s^{\sigma'}(X_i) \mid \sigma'=\sigma_{-i}[\sigma_i'] \text{ and } \sigma_i' \in \Sigma_i \} - \epsilon$ for all i, s
  - ϵ-NE (but not 0-NE) guaranteed to exist for CSGs

- In particular: social welfare/cost Nash equilibria (SWNE/SCNE)
  - NE which maximise/minimise sum $E_s^{\sigma}(X_1) + \dots E_s^{\sigma}(X_n)$

Equilibria-based Model Checking for Concurrent Stochastic Games, Kwiatkowska et al., In *Proc* FM 2019

rPATL + Nash operator

- Extension of rPATL for Nash equilibria

  $\phi ::= \text{true} \mid a \mid \neg\phi \mid \phi \wedge \phi \mid \langle\langle C \rangle\rangle P_{\bowtie q}[\psi] \mid \langle\langle C \rangle\rangle R^{r}_{\bowtie x}\ [\rho] \mid \langle\langle C,C' \rangle\rangle_{\max \bowtie x}\ [\theta]$

  $\theta ::= P[\psi]+P[\psi] \mid R^{r}[\rho]+R^{r}[\rho]$

  $\psi ::= X\ \phi \mid \phi\ U^{\leq k}\ \phi \mid \phi\ U\ \phi$

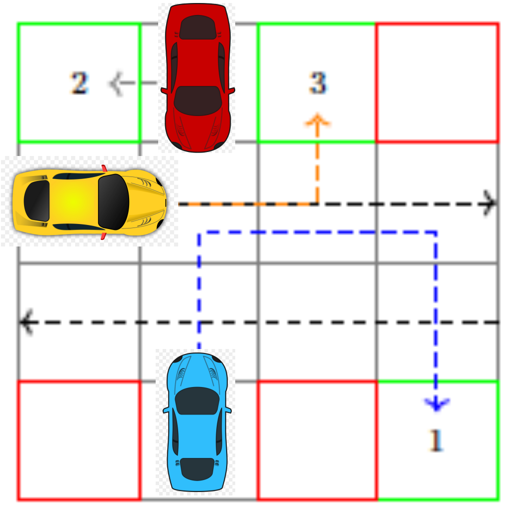
  $\rho ::= I^{=k} \mid C^{\leq k} \mid F\ \phi$

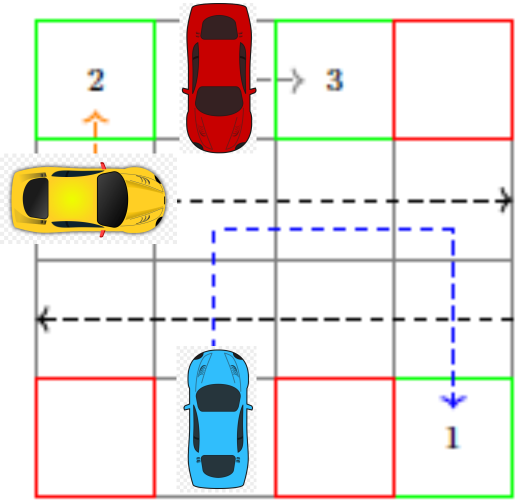
- where:
  - $a \in AP$ is an atomic proposition, $C \subseteq N$ is a coalition of players and $C' = N \setminus C$
  - $\bowtie \in \{\leq,<,>,\geq\}$, $q \in [0,1] \cap \mathbb{Q}$, $x \in \mathbb{Q}_{\geq 0}$, $k \in \mathbb{N}$
  - r is a reward structure
- Semantics: $\langle\langle C,C' \rangle\rangle_{\max \bowtie x}\ [\theta]$
  - is satisfied if there exist strategies for all players that form a SWNE between coalitions C and C'(=N\C), and under which the sum of the two objectives in θ is $\bowtie x$
  - now also extended to m-coalitional properties

Multi-player Equilibria Verification for Concurrent Stochastic Games, Kwiatkowska et al., In *Proc* QEST 2020

Example – Automated parking

- Safe parking
  $\langle\langle v_1,v_2,v_3\rangle\rangle_{\max\geq 3}$ $(P[\neg coll_1 U\ park_1]+ P[\neg coll_2 U\ park_2] + P[\neg coll_3 U\ park_3])$
- Minimise collective travelled distance
  $\langle\langle v_1,v_2,v_3\rangle\rangle_{\min=?}$ $(R[F\ park_1]+R[F\ park_2]+R[F\ park_3])$



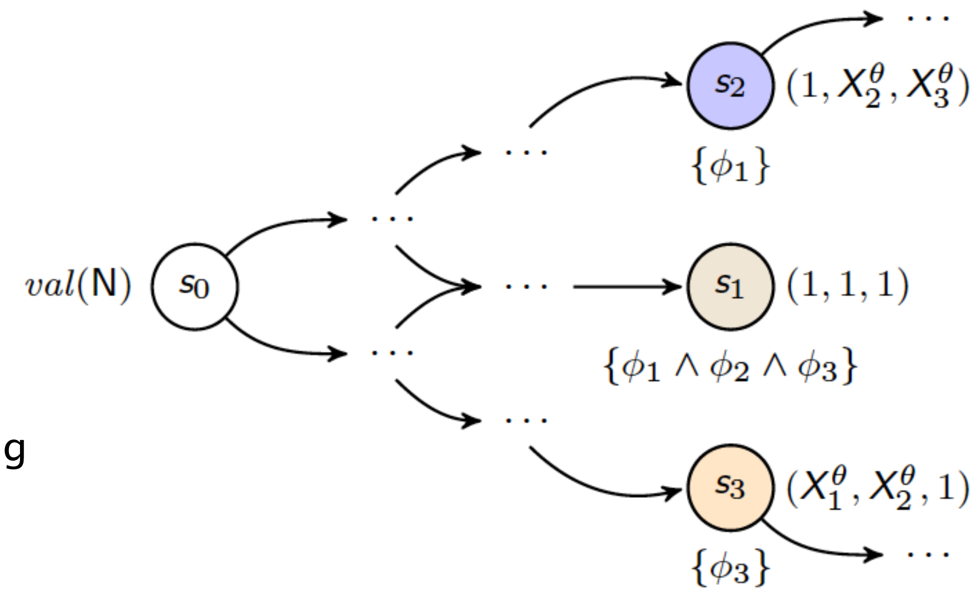


Two equilibria: distance 10

distance 8 (SCNE)

Model checking for equilibria properties

- Need to compute NE in each state, under certain restrictions on games (PSPACE)
  - bimatrix games (labelled polytopes), support enumeration
- For finite-horizon
  - property values computed via backward induction
- For infinite-horizon
  - use value iteration to approximate values as the limit
- For two-coalitional equilibria
  - employ MDP model model checking
- Implementation makes use of SMT solvers and non-linear optimization
- Also strategy synthesis


$val(\mathsf{N})$
s_0
s_1
$(1, 1, 1)$
$\{\phi_1 \wedge \phi_2 \wedge \phi_3\}$
s_2
$(1, X_2^\theta, X_3^\theta)$
$\{\phi_1\}$
s_3
$(X_1^\theta, X_2^\theta, 1)$
$\{\phi_3\}$
$\cdots$

Equilibria-based Model Checking for Concurrent Stochastic Games, Kwiatkowska et al., In *Proc* FM 2019

Case study: Secret sharing

- Protocol proposed by Halpern and Teague, theoretical analysis only
- Each agent has an unfair coin with the same bias (α)

- Step 1: agents flip their coins. If the coin lands on heads, the agent is supposed to send its share of the secret to others
- Step 2: everyone reveals the value of their coin to the other agents
- The protocol ends if:

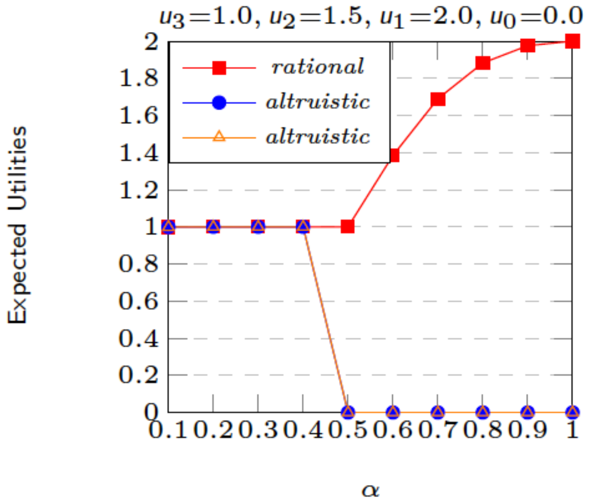
    All agents obtain all shares and therefore can all reconstruct the secret

    An agent cheats, i.e. fails to send their share to another agent when supposed to

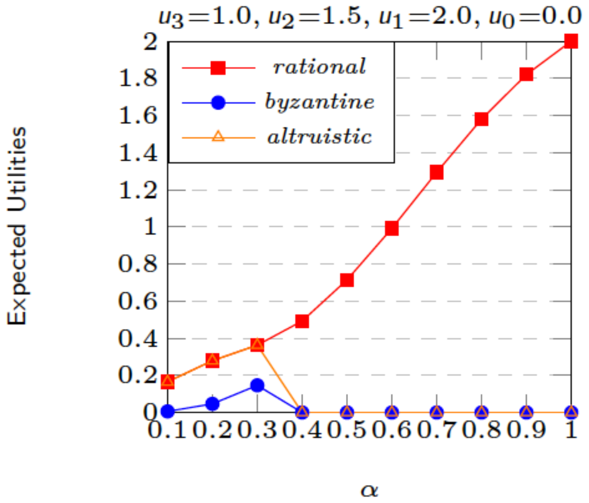
Rational Secret Sharing and Multiparty Computation, Halpern and Teague, In *Proc ACM STOC* 2004

Case study: Secret sharing

- Cannot be modelled using two-coalitional variant of rPATL
- Property $\langle\langle usr_1,usr_2,usr_3 \rangle\rangle_{max=?}$ (R[F done]+R[F done]+R[F done]) for $p_{fail} = 0.2$



((a)) 2 altruistic and 1 rational

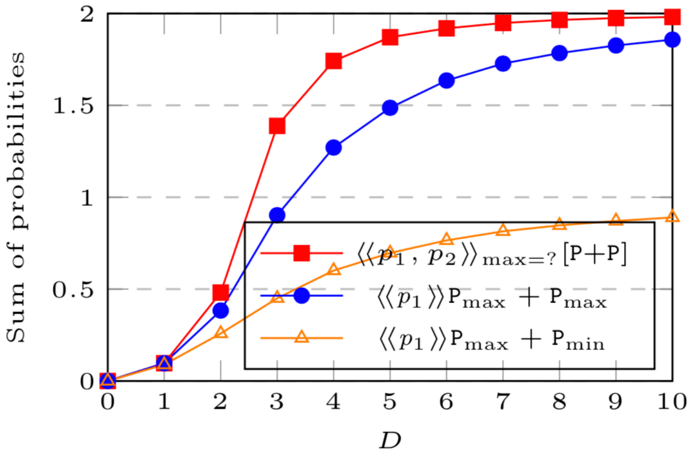


((b)) 1 altruistic, 1 byzantine and 1 rational

- Confirms theoretical analysis: a rational player would cheat given a high enough probability other players would share

PRISM-games 3.0

- Implementation in PRISM-games
  - needed further extensions to modelling language
  - implements CSG rPATL model checking
  - SMT plus backward induction and value iteration
  - scales up to CSGs with ~2 million states
- Extensions (not discussed)
  - multiobjective properties, Pareto sets, probabilistic real-time games
- Applications
  - robot navigation in a grid, medium access control, Aloha communication protocol, power control
  - SWNE strategies outperform those found with rPATL
  - ϵ-Nash equilibria found typically have ϵ=0



PRISM-games 3.0: Stochastic Game Verification with Concurrency, Equilibria and Time, Kwiatkowska et al., In *Proc* CAV 2020

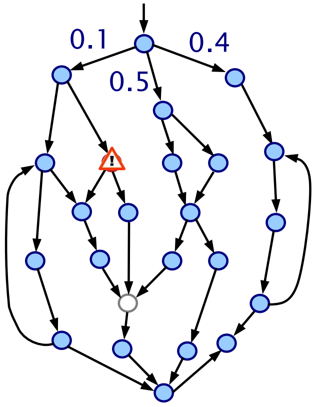
Case studies (selected)

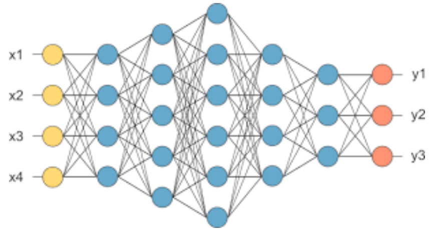
- Turn-based games
  - futures market investor model [McIver & Morgan]
  - energy management in microgrids [TACAS'12]
  - DNS bandwidth amplification attack [Deshpande et al]
  - self-adaptive software architectures [Camara, Garlan et al]
  - attack-defence scenarios in RFID goods managament [Aslanyan et al]
- Multi-objective turn-based games
  - UAV path planning with operator (multi-objective) [ICCPS'15]
  - aircraft electric power control (compositional) [TACAS'15]
- Concurrent games
  - public good game [CAV 2020]
  - intrusion detection policies [QEST 2018]
  - Aloha protocol [QEST 2020]

https://www.prismmodelchecker.org/games/casestudies.php

Where next for probabilistic model checking?

- Data rich, data-enabled models
  - achieved through learning
  - parameter estimation
  - continuous adaptation





- Bayesian neural networks have prior on weights
  - account for noise, uncertainty in measurements, etc
  - return an uncertainty measure along with the output

- Define safety with prob 1−ε: $Prob(\exists y \in \eta \text{ s.t. } f(x) \neq f(y) \mid D) \leq \varepsilon$
- i.e. conditioned on training data D
- Extending probabilistic verification to such settings

Probabilistic Safety for Bayesian Neural Networks, Wicker et al., In *Proc* UAI 2020

Conclusions

- Probabilistic model checking: PRISM & PRISM-games
  - theory and tool implementation
  - rPATL model checking for
    - stochastic multi-player games (SMGs)
    - concurrent stochastic games (CSGs)
  - CSGs + (social welfare) Nash equilibria
  - wide variety of case studies studied

- Challenges & directions
  - scalability, e.g. symbolic methods, abstraction
  - partial information/observability & greater efficiency
  - further applications and case studies
  - neural networks as games
  - probabilistic verification for Bayesian neural networks

Acknowledgements

- My group and collaborators in this work
- Project funding
  - ERC Advanced Grant VERIWARE
  - EPSRC Mobile Autonomy Programme Grant

- See also
  - PRISM www.prismmodelchecker.org

- New ERC Advanced Grant fun2model

  "From FUNction-based TO MOdel-based automated probabilistic reasoning for DEep Learning"

  - Postdoc positions