Continuous Optimal Timing

Yuliya Butkova, Hassan Hatefi, Holger Hermanns, Jan Krčál

Saarland University – Computer Science, Saarbrücken, Germany

May 6, 2015

Outline

- Motivation
- Preliminaries
- Existing Algorithms
- Our Algorithm
- Empirical Evaluation
- Conclusion

Motivation

Probabilistic models

- **unreliable/unpredictable system behaviour:**

  message loss, component failure, ...

- **randomized algorithms:**

  the probability of reaching consensus in leader election algorithms is almost 1

Motivation

Models we work with:

- run in continuous time
- comprise non-deterministic and probabilistic behaviour

are good for:

- optimization over multiple available choices
- finding worst case results

properties:

Is the maximal probability of reaching a failure state within an hour < 0.01?

Motivation

- Model checking boils down to time-bounded reachability problem:

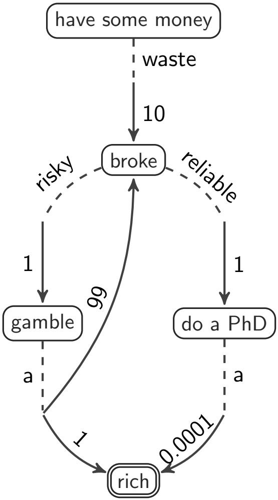
  What is the maximal/minimal probability to reach a given set of states within a given time bound?

- Several algorithms to tackle this problem are known
  - they are polynomial, but still slow on industrial size benchmarks
  - there is no proper comparison between all of them
  - no one has a clue which algorithm will be faster on a specific benchmark

Outline

- Motivation
- Preliminaries
- Existing Algorithms
- Our Algorithm
- Empirical Evaluation
- Conclusion

CTMDPs

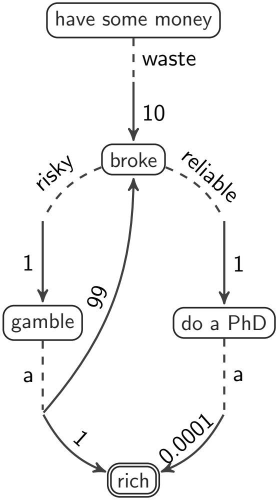

have some money
waste
10
broke
risky
reliable
1
1
99
gamble
do a PhD
a
a
1
0.0001
rich

CTMDPs

Continuous Time Markov Decision Process (CTMDP)

is a tuple $C = (S, Act, \boldsymbol{R})$, where

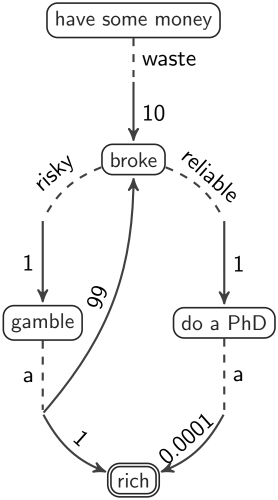
- S - set of states
- Act - set of actions
- $\boldsymbol{R} : S \times Act \times S \mapsto \mathbb{R}_{\geq 0}$ rate function



CTMDPs

Continuous Time Markov Decision Process (CTMDP)
is a tuple $C = (S, Act, \boldsymbol{R})$, where

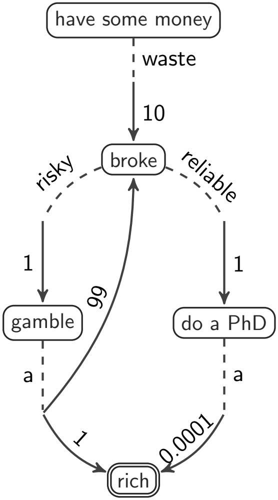
- S - set of states
- Act - set of actions
- $\boldsymbol{R} : S \times Act \times S \mapsto \mathbb{R}_{\geq 0}$ rate function



- **Exit Rate** $E(s, \alpha) = \sum_{s' \in S} \boldsymbol{R}(s, \alpha, s')$
- CTMDP is **Uniform** if exit rates over all states and all available actions are the same

Resolution of Non-Determinism. Schedulers.

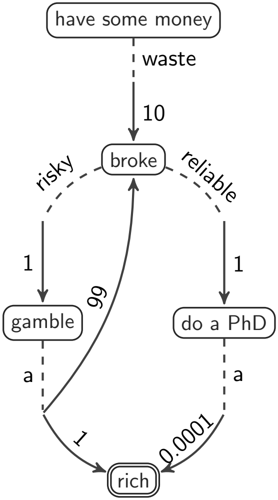
What is the probability of becoming reach before I die?



Resolution of Non-Determinism. Schedulers.

What is the probability of becoming reach before I die?

The answer depends on chosen actions



Resolution of Non-Determinism. Schedulers.

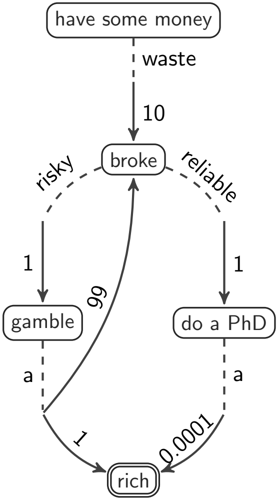
What is the probability of becoming reach before I die?

The answer depends on chosen actions

- A **Scheduler** σ (or **controller, policy**):

$$\sigma : \text{History} \rightarrow Act$$

- Classes of schedulers:
  - **Timed/Untimed** - knowledge of time passed (*Tim*/*Unt*)
  - **Early/Late** - decision is fixed on entering a state/maybe changed at any time later

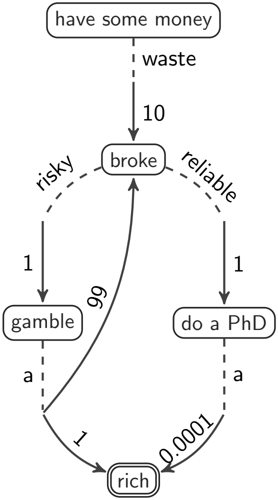


Reachability Problem

What is the maximal/minimal probability to reach a given set of states within given time?

$$\mathrm{val}^{\nabla}(s) := \sup_{\sigma \in Tim_{\nabla}} \Pr_{\sigma}^{s} \left[\Diamond^{\leq T} G \right]$$

$$\nabla \in \{\ell, e\}$$



Outline

- Motivation
- Preliminaries
- Existing Algorithms
- Our Algorithm
- Empirical Evaluation
- Conclusion

Existing Algorithms

Early

- Exponential Approximation
  EXPSTEP-1
  (by M. Neuhaeussar, L. Zhang)
- Improved Exponential Approximation
  EXPSTEP-k
  (by H. Hatefi, H. Hermanns)

Late

- Polynomial Approximation
  POLYSTEP-k
  (by J. Fearnley, M. Rabe, et al.)
- Adaptive Step Approximation
  ADAPTSTEP
  (by P. Buchholz, I. Schulz)

All existing approaches use **discretization**

Outline

- Motivation
- Preliminaries
- Existing Algorithms
- Our Algorithm
- Empirical Evaluation
- Conclusion

Our Approach

Features:

- Does NOT discretize the time horizon, instead
- approximate via different class of schedulers:
  - Less powerfull **Untimed** - for lower bound
  - More powerfull **“Prophetic”** - for upper bound

Our Algorithm (UNIF$^+$)

input : CTMDP $\mathcal{C} = (S, Act, \mathbf{R})$, goal states $G \subseteq S$, horizon $T \in \mathbb{R}_{>0}$, scheduler class $\nabla \in \{\ell, e\}$, and approximation error $\varepsilon > 0$

params: truncation error ratio $\kappa \in (0, 1)$

output : vector v such that $\|\mathrm{v} - \mathsf{val}^{\nabla}\|_{\infty} \leq \varepsilon$

$\lambda \leftarrow$ maximal exit rate E_{max} in $\mathcal{C}$

repeat

  $\mathcal{C}_{\lambda}^{\nabla} \leftarrow \nabla$-uniformisation of $\mathcal{C}$ to the rate λ

  $\underline{\mathrm{v}} \leftarrow$ approximation of the lower bound $\underline{\mathrm{val}}$ for $\mathcal{C}_{\lambda}^{\nabla}$ up to error $\varepsilon \cdot \kappa$

  $\overline{\mathrm{v}} \leftarrow$ approximation of the upper bound $\overline{\mathrm{val}}$ for $\mathcal{C}_{\lambda}^{\nabla}$ up to error $\varepsilon \cdot \kappa$

  $\lambda \leftarrow 2 \cdot \lambda$

until $\|\overline{\mathrm{v}} - \underline{\mathrm{v}}\|_{\infty} \leq \varepsilon \cdot (1 - \kappa)$

return $\underline{\mathrm{v}}$

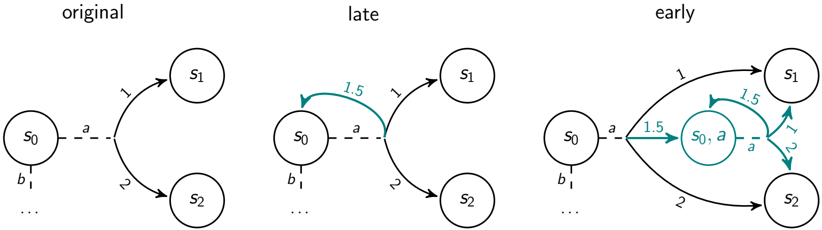
Our Algorithm ($\textsc{Unif}^{+}$)

input : CTMDP $\mathcal{C} = (S, Act, \mathbf{R})$, goal states $G \subseteq S$, horizon $T \in \mathbb{R}_{>0}$, scheduler class $\nabla \in \{\ell, e\}$, and approximation error $\varepsilon > 0$
params: truncation error ratio $\kappa \in (0, 1)$
output : vector v such that $\|\mathrm{v} - \mathrm{val}^{\nabla}\|_{\infty} \leq \varepsilon$

$\lambda \leftarrow$ maximal exit rate E_{max} in $\mathcal{C}$
repeat
  $\mathcal{C}_{\lambda}^{\nabla} \leftarrow$ **∇-uniformisation of $\mathcal{C}$ to the rate λ**
  $\underline{\mathrm{v}} \leftarrow$ approximation of the lower bound $\underline{\mathrm{val}}$ for $\mathcal{C}_{\lambda}^{\nabla}$ up to error $\varepsilon \cdot \kappa$
  $\overline{\mathrm{v}} \leftarrow$ approximation of the upper bound $\overline{\mathrm{val}}$ for $\mathcal{C}_{\lambda}^{\nabla}$ up to error $\varepsilon \cdot \kappa$
  $\lambda \leftarrow 2 \cdot \lambda$
until $\|\overline{\mathrm{v}} - \underline{\mathrm{v}}\|_{\infty} \leq \varepsilon \cdot (1 - \kappa)$
return $\underline{\mathrm{v}}$

UNIF$^+$. Uniformization

Uniformize to the rate 4.5:



Our Algorithm (UNIF$^+$)

input : CTMDP $\mathcal{C} = (S, Act, \mathbf{R})$, goal states $G \subseteq S$, horizon $T \in \mathbb{R}_{>0}$, scheduler class $\nabla \in \{\ell, e\}$, and approximation error $\varepsilon > 0$

params: truncation error ratio $\kappa \in (0, 1)$

output : vector v such that $\|\mathrm{v} - \mathrm{val}^{\nabla}\|_\infty \leq \varepsilon$

$\lambda \leftarrow$ maximal exit rate E_{max} in $\mathcal{C}$

repeat

  $\mathcal{C}_\lambda^\nabla \leftarrow \nabla$-uniformisation of $\mathcal{C}$ to the rate λ

  $\underline{\mathrm{v}} \leftarrow$ **approximation of the lower bound for** $\mathcal{C}_\lambda^\nabla$ **up to error** $\varepsilon \cdot \kappa$

  $\overline{\mathrm{v}} \leftarrow$ **approximation of the upper bound for** $\mathcal{C}_\lambda^\nabla$ **up to error** $\varepsilon \cdot \kappa$

  $\lambda \leftarrow 2 \cdot \lambda$

until $\|\overline{\mathrm{v}} - \underline{\mathrm{v}}\|_\infty \leq \varepsilon \cdot (1 - \kappa)$

return $\underline{\mathrm{v}}$

UNIF$^+$. Bounds

Lower Bound

$$\underline{\mathrm{val}}(s) := \sup_{\sigma \in Unt} \sum_{i=0}^{\infty} \mathrm{Pr}_{\sigma}^{\mathcal{C}_{\lambda}^{\nabla}, s}\left[\Diamond_{=i}^{\leq T} G\right]$$

Optimal reachability probability **over untimed schedulers**

Upper Bound

$$\overline{\mathrm{val}}(s) := \sum_{i=0}^{\infty} \sup_{\sigma \in Unt} \mathrm{Pr}_{\sigma}^{\mathcal{C}_{\lambda}^{\nabla}, s}\left[\Diamond_{=i}^{\leq T} G\right]$$

Optimal reachability probability **over “prophetic” schedulers**

Our Algorithm (Unif$^+$)

input : CTMDP $\mathcal{C} = (S, Act, \mathbf{R})$, goal states $G \subseteq S$, horizon $T \in \mathbb{R}_{>0}$, scheduler class $\nabla \in \{\ell, e\}$, and approximation error $\varepsilon > 0$
params: truncation error ratio $\kappa \in (0, 1)$
output : vector v such that $\|\mathrm{v} - \mathrm{val}^{\nabla}\|_{\infty} \leq \varepsilon$

```
λ ← maximal exit rate E_max in C
repeat
    C_λ^∇ ← ∇-uniformisation of C to the rate λ
    v_ ← approximation of the lower bound val_ for C_λ^∇ up to error ε · κ
    v¯ ← approximation of the upper bound val¯ for C_λ^∇ up to error ε · κ
    λ ← 2 · λ
until ‖v¯ − v_‖_∞ ≤ ε · (1 − κ)
return v_
```

$\lambda \leftarrow$ maximal exit rate E_{max} in $\mathcal{C}$
repeat
- $\mathcal{C}_\lambda^\nabla \leftarrow \nabla$-uniformisation of $\mathcal{C}$ to the rate λ
- $\underline{\mathrm{v}} \leftarrow$ approximation of the lower bound $\underline{\mathrm{val}}$ for $\mathcal{C}_\lambda^\nabla$ up to error $\varepsilon \cdot \kappa$
- $\overline{\mathrm{v}} \leftarrow$ approximation of the upper bound $\overline{\mathrm{val}}$ for $\mathcal{C}_\lambda^\nabla$ up to error $\varepsilon \cdot \kappa$
- $\lambda \leftarrow 2 \cdot \lambda$

until $\|\overline{\mathrm{v}} - \underline{\mathrm{v}}\|_{\infty} \leq \varepsilon \cdot (1 - \kappa)$
return $\underline{\mathrm{v}}$

Outline

Empirical Evaluation and Comparison

	max. $\|S\|$	max. $\curlywedge$	range of max. exit rates	best in early (# of cases)	best in late (# of cases)
PS:	743 969	7	5,6 – 129,6	U$^+$ **(32)**	U$^+$ **(47)**
QS:	16 924	36	6,5 – 44,9	U$^+$ **(32)**	PS-3**(18)**, U$^+$ (17), AS (15)
DPMS:	366 148	7	2,1 – 9,1	U$^+$ **(31)**, ES-2(3), N/A(1)	AS **(24)**, U$^+$ (14), PS-3(6)
GFS:	15 258	2	252 – 612	U$^+$ **(40)**	AS **(23)**, U$^+$ (11)
FTWC:	2 373 650	5	2 – 3,02	U$^+$ **(25)**	U$^+$ **(32)**
SJS:	18 451	72	3 – 32	U$^+$ **(57)**, ES-2(2)	U$^+$ **(70)**, AS (29)
ES:	30 004	2	10	U$^+$ **(23)**, ES-2(4), N/A(1)	U$^+$ **(28)**, PS-3(2)

Table: Overview of experiments summarizing which algorithm performed best how many times; N/A indicates that no algorithm completed within 15 minutes.

Outline

- Motivation
- Preliminaries
- Existing Algorithms
- Our Algorithm
- Empirical Evaluation
- Conclusion

Conclusion

- Unif^{+} performs very well for early scheduling problems
- Unif^{+} is competitive on late scheduling problems
- Results on late scheduling are inconclusive. Further insight into the problem is required
- The benefits of Unif^{+}:
  - it is easily switchable between early/late schedulers
  - a simplified version of Unif^{+} with only 1 iteration is very fast and may give good a posteriori error bounds

The End