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
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

Yuliya Butkova, Hassan Hatefi, Holger Hermanns, Jan Krčál

Continuous Optimal Timing
Continuous Time Markov Decision Process (CTMDP) is a tuple $C = (S, Act, R)$, where

- $S$ - set of states
- $Act$ - set of actions
- $R : S \times Act \times S \rightarrow \mathbb{R}_{\geq 0}$ rate function
Continuous Time Markov Decision Process (CTMDP)
is a tuple $C = (S, Act, R)$, where

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

- **Exit Rate** $E(s, \alpha) = \sum_{s' \in S} R(s, \alpha, s')$

- CTMDP is **Uniform** if exit rates over all states and all available actions are the same
What is the probability of becoming rich before I die?
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** $\sigma$ (or **controller**, **policy**):
  
  $$\sigma : \text{History} \rightarrow \text{Act}$$

- Classes of schedulers:
  - **Timed/Untimed** - knowledge of time passed ($\text{Tim}/\text{Unt}$)
  - **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?

\[ \text{val}^\nabla(s) := \sup_{\sigma \in T_{im}} \Pr^s_{\sigma} \left[ \diamondsuit \leq T \right] \]

\[ \nabla \in \{\ell, e\} \]
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 powerful Untimed - for lower bound
  - More powerful “Prophetic” - for upper bound
Our Algorithm (UNIF$^+$)

input : CTMDP $C = (S, Act, R)$, 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 $\|v - \text{val}_\nabla\|_\infty \leq \varepsilon$

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

2. repeat
3. $C_\lambda^{\nabla} \leftarrow \nabla$-uniformisation of $C$ to the rate $\lambda$
4. $v \leftarrow$ approximation of the lower bound $\text{val}$ for $C_\lambda^{\nabla}$ up to error $\varepsilon \cdot \kappa$
5. $\overline{v} \leftarrow$ approximation of the upper bound $\overline{\text{val}}$ for $C_\lambda^{\nabla}$ up to error $\varepsilon \cdot \kappa$
6. $\lambda \leftarrow 2 \cdot \lambda$

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

8. return $\overline{v}$
Our Algorithm (UNIF$^+$)

**input**: CTMDP $C = (S, Act, R)$, 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 $\|v - \text{val}^\nabla\|_{\infty} \leq \varepsilon$

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

2. repeat
3. $\quad C^\nabla_\lambda \leftarrow \nabla$-uniformisation of $C$ to the rate $\lambda$
4. $\quad v \leftarrow$ approximation of the lower bound $\text{val}$ for $C^\nabla_\lambda$ up to error $\varepsilon \cdot \kappa$
5. $\quad \bar{v} \leftarrow$ approximation of the upper bound $\overline{\text{val}}$ for $C^\nabla_\lambda$ up to error $\varepsilon \cdot \kappa$
6. $\quad \lambda \leftarrow 2 \cdot \lambda$
7. until $\|\bar{v} - v\|_{\infty} \leq \varepsilon \cdot (1 - \kappa)$
8. return $v$
Uniformization

Uniformize to the rate 4.5:

Original:

```
\begin{align*}
\text{s}_0 & \xrightarrow{a} \text{s}_1 \\
\text{s}_1 & \xrightarrow{b} \text{s}_2 \\
\text{s}_2 & \xrightarrow{c} \text{s}_0
\end{align*}
```

Late:

```
\begin{align*}
\text{s}_0 & \xrightarrow{a} \text{s}_1 \\
\text{s}_1 & \xrightarrow{b} \text{s}_2 \\
\text{s}_2 & \xrightarrow{c} \text{s}_0
\end{align*}
```

Early:

```
\begin{align*}
\text{s}_0 & \xrightarrow{a} \text{s}_1 \\
\text{s}_1 & \xrightarrow{b} \text{s}_2 \\
\text{s}_2 & \xrightarrow{c} \text{s}_0
\end{align*}
```
Our Algorithm (UNIF$^+$)

---

**input**: CTMDP $C = (S, Act, R)$, 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 $\|v - \text{val}^\nabla\|_\infty \leq \varepsilon$

1. $\lambda \leftarrow$ maximal exit rate $E_{\text{max}}$ in $C$

2. repeat
3. \quad $C^\nabla_{\lambda} \leftarrow \nabla$-uniformisation of $C$ to the rate $\lambda$
4. \quad $\underline{v} \leftarrow$ approximation of the lower bound for $C^\nabla_{\lambda}$ up to error $\varepsilon \cdot \kappa$
5. \quad $\overline{v} \leftarrow$ approximation of the upper bound for $C^\nabla_{\lambda}$ up to error $\varepsilon \cdot \kappa$
6. \quad $\lambda \leftarrow 2 \cdot \lambda$

7. until $\|\overline{v} - \underline{v}\|_\infty \leq \varepsilon \cdot (1 - \kappa)$
8. return $\underline{v}$
Unif$^+$. Bounds

**Lower Bound**

$$\overline{\text{val}}(s) := \sup_{\sigma \in \text{Unt}} \sum_{i=0}^{\infty} \text{Pr}_{\sigma}^{C_L,s} [\Diamond \leq T G]$$

Optimal reachability probability over untimed schedulers

**Upper Bound**

$$\text{val}(s) := \sum_{i=0}^{\infty} \sup_{\sigma \in \text{Unt}} \text{Pr}_{\sigma}^{C_L,s} [\Diamond \leq T G]$$

Optimal reachability probability over “prophetic” schedulers
Our Algorithm (**UNIF\(^+\)**)

Input: CTMDP \(\mathcal{C} = (S, Act, R)\), 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 \(\|v - \text{val}^{\nabla}\|_{\infty} \leq \varepsilon\)

1. \(\lambda \leftarrow \text{maximal exit rate} \ E_{\text{max}} \ \text{in} \ \mathcal{C}\)

2. repeat
3. \(\mathcal{C}^{\nabla}_{\lambda} \leftarrow \nabla\text{-uniformisation of} \ \mathcal{C} \ \text{to the rate} \ \lambda\)
4. \(v \leftarrow \text{approximation of the lower bound} \ \text{val} \ \text{for} \ \mathcal{C}^{\nabla}_{\lambda} \ \text{up to error} \ \varepsilon \cdot \kappa\)
5. \(\overline{v} \leftarrow \text{approximation of the upper bound} \ \overline{\text{val}} \ \text{for} \ \mathcal{C}^{\nabla}_{\lambda} \ \text{up to error} \ \varepsilon \cdot \kappa\)
6. \(\lambda \leftarrow 2 \cdot \lambda\)
7. until \(\|\overline{v} - v\|_{\infty} \leq \varepsilon \cdot (1 - \kappa)\)
8. return \(v\)
Outline

- Motivation
- Preliminaries
- Existing Algorithms
- Our Algorithm
- Empirical Evaluation
- Conclusion
## Empirical Evaluation and Comparison

|            | max. $|S|$ | max. $\lambda$ | range of max. exit rates | best in early (# of cases) | best in late (# of cases) |
|------------|-------|----------------|---------------------------|-----------------------------|-----------------------------|
| PS         | 743969| 7              | 5.6 – 129.6               | $U^+$ (32)                  | $U^+$ (47)                  |
| QS         | 16924 | 36             | 6.5 – 44.9                | $U^+$ (32)                  | PS-3 (18), $U^+$ (17), AS (15) |
| DPMS       | 366148| 7              | 2.1 – 9.1                 | $U^+$ (31), ES-2(3), N/A (1) | AS (24), $U^+$ (14), PS-3 (6) |
| GFS        | 15258 | 2              | 252 – 612                 | $U^+$ (40)                  | AS (23), $U^+$ (11)         |
| FTWC       | 2373650| 5              | 2 – 3.02                  | $U^+$ (25)                  | $U^+$ (32)                  |
| SJS        | 18451 | 72             | 3 – 32                    | $U^+$ (57), ES-2(2)         | $U^+$ (70), AS (29)         |
| ES         | 30004 | 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

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