Discounting in Time

Uli Fahrenberg Kim G. Larsen

Department of Computer Science
Aalborg University
Denmark

QAPL 2009

One-slide summary (tongue in cheek!)

In formalisms with both
time and
weights,
there is
only **one** way
to define total cost of infinite runs.

One-slide summary (tongue in cheek!)

In formalisms with both
time and
non-negative **weights**,
there is
essentially only **one** way
to define total cost of infinite runs.

One-slide summary (tongue in cheek!)

In formalisms with both

time and

non-negative **weights**,

there is

essentially only **one** way

to define total cost of infinite runs.

In particular, those people did it **the wrong way**:




One-slide summary (tongue in cheek!)

In formalisms with both
time and
non-negative **weights**,
there is
essentially only **one** way
to define total cost of infinite runs.

In particular, those people did it **the wrong way**:




And **economists** have been doing it **right** for 100s of years.

Contents

Motivation

The big picture: Optimal infinite scheduling

- Control for factory to achieve highest output

Motivation

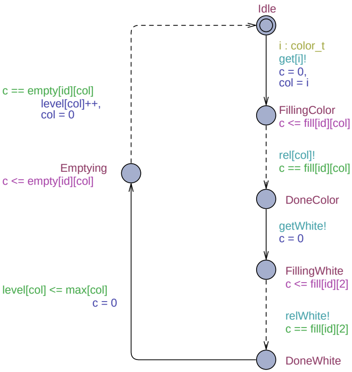
The big picture: Optimal infinite scheduling

- Control for factory to achieve highest output
- Control for car to achieve energy efficiency

Motivation

The big picture: Optimal infinite scheduling

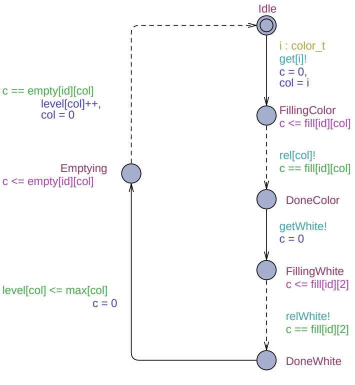
- Control for factory to achieve highest output
- Control for car to achieve energy efficiency
- *etc.*
- Weighted timed automata; weighted timed games; weighted time Petri nets
- UPPAAL; PHAVer; MATLAB® Simulink®; *etc.*



Motivation

The big picture: Optimal infinite scheduling

- Control for factory to achieve highest output
- Control for car to achieve energy efficiency
- *etc.*
- Weighted timed automata; weighted timed games; weighted time Petri nets
- UPPAAL; PHAVer; MATLAB® Simulink®; *etc.*

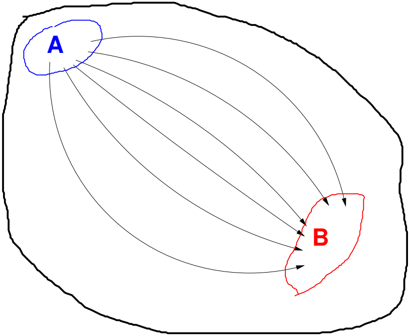


Underlying semantic model: Weighted timed transition systems

Motivation

Optimal finite scheduling:

- optimal reachability problem
- not difficult to solve for weighted timed automata
- tool support in UPPAAL



Find cheapest way from A to B

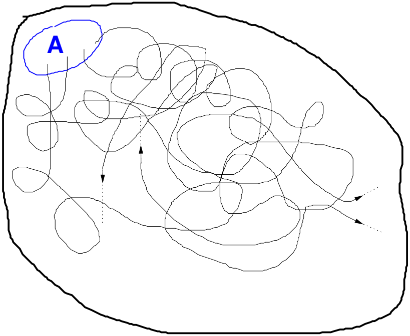
Motivation

Optimal finite scheduling:

- optimal reachability problem
- not difficult to solve for weighted timed automata
- tool support in UPPAAL

Optimal infinite scheduling:

- optimal safety problem (?)
- difficult even to *define*: What is the price of an infinite run ?



Find cheapest infinite execution starting in A

Motivation

What is the price of an infinite run

?

(and others ?)

Motivation

What is the price
of an infinite run
in a weighted timed transition system
with non-negative weights
?

(and others ?)

How to compute (finite) prices of infinite runs



- BBL, *HSCC* 2004: Mean-cost approach

$$P(\rho) = \liminf_{T\to\infty} \frac{P(\rho_{\upharpoonright T})}{T} \longleftarrow \text{run } \rho \text{ up to time } T$$

How to compute (finite) prices of infinite runs



- BBL, *HSCC* 2004: Mean-cost approach

$$P(\rho) = \liminf_{T\to\infty} \frac{P(\rho_{\upharpoonright T})}{T} \longleftarrow \text{run } \rho \text{ up to time } T$$


- JT, *FORMATS* 2008: Step-based discounting
  - fix *discounting factor* $0 < \lambda < 1$
  - after each discrete step, weights are multiplied by λ
  - (as if the model was discrete:

$$P\big(\xrightarrow[p_1]{} \xrightarrow[p_2]{} \xrightarrow[p_3]{} \cdots \big) = p_1 + \lambda p_2 + \lambda^2 p_3 + \cdots \qquad)$$

How to compute (finite) prices of infinite runs



- BBL, *HSCC* 2004: Mean-cost approach

$$P(\rho) = \liminf_{T\to\infty} \frac{P(\rho_{\upharpoonright T})}{T} \longleftarrow \text{run } \rho \text{ up to time } T$$


- JT, *FORMATS* 2008: Step-based discounting
  - fix *discounting factor* $0 < \lambda < 1$
  - after each discrete step, weights are multiplied by λ
  - (as if the model was discrete:

$$P\big(\xrightarrow[p_1]{} \xrightarrow[p_2]{} \xrightarrow[p_3]{} \cdots \big) = p_1 + \lambda p_2 + \lambda^2 p_3 + \cdots \quad)$$


- FL, *INFINITY* 2008: Time-based discounting
  - things which happen at time T, are discounted with λ^T

Discounting

- In economics: *future* loss or profit matters less than if it occurred *right now*
- Using *expected return rate* r, the *net present value* of a transaction x at time T is

$$x_{\text{NPV}} = \frac{1}{(1+r)^T} x = \lambda^T x$$

- with discount factor $\lambda = \dfrac{1}{1+r}$

Discounting

- In economics: *future* loss or profit matters less than if it occurred *right now*
- Using *expected return rate* r, the *net present value* of a transaction x at time T is

$$x_{\text{NPV}} = \frac{1}{(1+r)^T} x = \lambda^T x$$

- with discount factor $\lambda = \dfrac{1}{1+r}$
- Expected return rate depends on
  - interest rates
  - perceived risk
  - greed, *etc.*

Discounting

- In economics: *future* loss or profit matters less than if it occurred *right now*
- Using *expected return rate* r, the *net present value* of a transaction x at time T is

$$x_{\mathrm{NPV}} = \frac{1}{(1+r)^T} x = \lambda^T x$$

- with discount factor $\lambda = \dfrac{1}{1+r}$
- Of interest for us: Using this form of discounting, most infinite paths have finite total price
- (because the geometric series $1 + \lambda + \lambda^2 + \cdots$ converges)

Weighted timed transition systems

(S, T_s, T_d, w, r) :

- states S, switches $T_s \subseteq S \times S$, delays $T_d \subseteq S \times \mathbb{R}_{\geq 0} \times S$, weights $w : T_s \to \mathbb{R}$, rates $r : S \to \mathbb{R}$

Axioms for delays:

- trivial loops: $\forall s \in S : s \xrightarrow{0} s$
- determinacy: $s \xrightarrow{t} s_1 \wedge s \xrightarrow{t} s_2 \Rightarrow s_1 = s_2$
- additivity: $s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'} \Rightarrow s \xrightarrow{t+t'} s^{t+t'}$
- density (?): $s \xrightarrow{t} s^t \wedge t' \leq t \Rightarrow s \xrightarrow{t'} s^{t'} \xrightarrow{t-t'} s^t$

Weighted timed transition systems

(S, T_s, T_d, w, r) :

- states S, switches $T_s \subseteq S \times S$, delays $T_d \subseteq S \times \mathbb{R}_{\geq 0} \times S$, weights $w : T_s \to \mathbb{R}$, rates $r : S \to \mathbb{R}$

Axioms for delays:

- trivial loops: $\forall s \in S : s \xrightarrow{0} s$
- determinacy: $s \xrightarrow{t} s_1 \wedge s \xrightarrow{t} s_2 \Rightarrow s_1 = s_2$
- additivity: $s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'} \Rightarrow s \xrightarrow{t+t'} s^{t+t'}$
- density (?): $s \xrightarrow{t} s^t \wedge t' \leq t \Rightarrow s \xrightarrow{t'} s^{t'} \xrightarrow{t-t'} s^t$

Weighted timed transition systems

(S, T_s, T_d, w, r) :

- states S, switches $T_s \subseteq S \times S$, delays $T_d \subseteq S \times \mathbb{R}_{\geq 0} \times S$, weights $w : T_s \to \mathbb{R}$, rates $r : S \to \mathbb{R}$

Axioms for delays:

- trivial loops: $\forall s \in S : s \xrightarrow{0} s$
- determinacy: $s \xrightarrow{t} s_1 \wedge s \xrightarrow{t} s_2 \Rightarrow s_1 = s_2$
- additivity: $s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'} \Rightarrow s \xrightarrow{t+t'} s^{t+t'}$
- density (?): $s \xrightarrow{t} s^t \wedge t' \leq t \Rightarrow s \xrightarrow{t'} s^{t'} \xrightarrow{t-t'} s^t$

Weighted timed transition systems

(S, T_s, T_d, w, r) :

- states S, switches $T_s \subseteq S \times S$, delays $T_d \subseteq S \times \mathbb{R}_{\geq 0} \times S$, weights $w : T_s \to \mathbb{R}$, rates $r : S \to \mathbb{R}$

Axioms for delays:

- trivial loops: $\forall s \in S : s \xrightarrow{0} s$
- determinacy: $s \xrightarrow{t} s_1 \wedge s \xrightarrow{t} s_2 \Rightarrow s_1 = s_2$
- additivity: $s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'} \Rightarrow s \xrightarrow{t+t'} s^{t+t'}$
- density (?): $s \xrightarrow{t} s^t \wedge t' \leq t \Rightarrow s \xrightarrow{t'} s^{t'} \xrightarrow{t-t'} s^t$

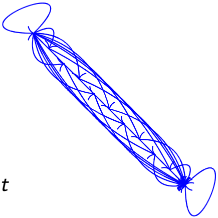
Weighted timed transition systems

(S, T_s, T_d, w, r) :

- states S, switches $T_s \subseteq S \times S$, delays $T_d \subseteq S \times \mathbb{R}_{\geq 0} \times S$, weights $w : T_s \to \mathbb{R}$, rates $r : S \to \mathbb{R}$

Axioms for delays:

- trivial loops: $\forall s \in S : s \xrightarrow{0} s$
- determinacy: $s \xrightarrow{t} s_1 \wedge s \xrightarrow{t} s_2 \Rightarrow s_1 = s_2$
- additivity: $s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'} \Rightarrow s \xrightarrow{t+t'} s^{t+t'}$
- density (?): $s \xrightarrow{t} s^t \wedge t' \leq t \Rightarrow s \xrightarrow{t'} s^{t'} \xrightarrow{t-t'} s^t$



Can view a delay as a continuum of intermediate states !

Discounted price of infinite paths

- Discounted price of *finite* alternating path $\pi = s_0 \xrightarrow{t_0} s_0^{t_0} \to s_1 \to \cdots \xrightarrow{t_{n-1}} s_{n-1}^{t_{n-1}} \to s_n$:

$$P(\pi) = \sum_{i=0}^{n-1} \Big(\int_{T_{i-1}}^{T_i} \lambda^t r(s_i^t) dt + \lambda^{T_i} p(s_i^{t_i} \to s_{i+1}) \Big)$$

with $T_i = \sum_{j=0}^{i} t_j$.

- Discounted price of *infinite* alternating path $\pi = s_0 \xrightarrow{t_0} s_0^{t_0} \to s_1 \to \cdots$: limit

$$P(\pi) = \lim_{n \to \infty} P\big(s_0 \xrightarrow{t_0} s_0^{t_0} \to \cdots \to s_{n-1}^{t_{n-1}} \to s_n\big)$$

provided that it exists. (!)

Discounted price of infinite paths

- Discounted price of *finite* alternating path $\pi = s_0 \xrightarrow{t_0} s_0^{t_0} \to s_1 \to \cdots \xrightarrow{t_{n-1}} s_{n-1}^{t_{n-1}} \to s_n$:

$$P(\pi) = \sum_{i=0}^{n-1} \Big(\int_{T_{i-1}}^{T_i} \lambda^t r(s_i^t) dt + \lambda^{T_i} p(s_i^{t_i} \to s_{i+1}) \Big)$$

with $T_i = \sum_{j=0}^{i} t_j$.

TL;DR

(Just a minute...)

- Discounted price of *infinite* alternating path $\pi = s_0 \xrightarrow{t_0} s_0^{t_0} \to s_1 \to \cdots$: limit

$$P(\pi) = \lim_{n\to\infty} P\big(s_0 \xrightarrow{t_0} s_0^{t_0} \to \cdots \to s_{n-1}^{t_{n-1}} \to s_n\big)$$

provided that it exists. (!)

INFINITY 2008 result

- Under certain assumptions, infinite paths with cheapest discounted price can be computed in weighted timed automata.
- (Similar results for mean-cost [BBL04] and for step-based discounting [JT08])
- But no efficient algorithms

Recursive property

Equivalent formulation of time-based discounting:

- For $s \underset{p}{\rightarrow} s'$ a switch and π a path out of the target of the switch:

$$P(s \underset{p}{\rightarrow} s' \circ \pi) = p + P(\pi)$$

For $s \xrightarrow{t} s^t$ a delay and π a path out of the end state of the delay:

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s^\tau) d\tau + \lambda^t P(\pi)$$

Recursive property

Equivalent formulation of time-based discounting:

- For $s \underset{p}{\rightarrow} s'$ a switch and π a path out of the target of the switch:

$$P(s \underset{p}{\rightarrow} s' \circ \pi) = p + P(\pi)$$

  For $s \xrightarrow{t} s^t$ a delay and π a path out of the end state of the delay:

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s^\tau) d\tau + \lambda^t P(\pi)$$

Recursive property

Equivalent formulation of time-based discounting:

- For $s \underset{p}{\rightarrow} s'$ a switch and π a path out of the target of the switch:

$$P(s \underset{p}{\rightarrow} s' \circ \pi) = p + P(\pi)$$

For $s \xrightarrow{t} s^t$ a delay and π a path out of the end state of the delay:

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s^\tau) d\tau + \lambda^t P(\pi)$$

- Nice recursive property
- Fixed-point computations ?
- Efficient, zone-based algorithms ?

Recursive property

Equivalent formulation of time-based discounting:

- For $s \xrightarrow[p]{} s'$ a switch and π a path out of the target of the switch:

$$P(s \xrightarrow[p]{} s' \circ \pi) = p + P(\pi)$$

For $s \xrightarrow{t} s^t$ a delay and π a path out of the end state of the delay:

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s^\tau) d\tau + \lambda^t P(\pi)$$

- Nice recursive property
- Fixed-point computations ?
- Efficient, zone-based algorithms ?
- Mean-cost and step-based-discounting approach do not have this property

Recursive property

$$P(s \underset{p}{\rightarrow} s' \circ \pi) = p + P(\pi)$$

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s^\tau) d\tau + \lambda^t P(\pi)$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

Recursive property

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s^\tau) d\tau + \lambda^t P(\pi)$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches

Recursive property

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s^{\cancel{\tau}}) d\tau + \lambda^t P(\pi)$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches
2. Declare rates a *discrete* property

Recursive property

$$P(s \xrightarrow{t} s^t \circ \pi) = \int_0^t \lambda^\tau r(s) d\tau + \lambda^t P(\pi)$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches
2. Declare rates a *discrete* property

Recursive property

$$P(s \xrightarrow{t} s^t \circ \pi) = P(s \xrightarrow{t} s^t) + \lambda^t P(\pi)$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches
2. Declare rates a *discrete* property
3. Simplify

Recursive property

$$P(s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'}) = P(s \xrightarrow{t} s^t) + \lambda^t P(s^t \xrightarrow{t'} s^{t+t'})$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches
2. Declare rates a *discrete* property
3. Simplify
4. Only look at paths of length 2

Recursive property

$$P(s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'}) = P(s \xrightarrow{t} s^t) + \lambda^t P(s \xrightarrow{t'} s^{t'})$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches
2. Declare rates a *discrete* property
3. Simplify
4. Only look at paths of length 2
5. Time shift

Recursive property

$$P(s \xrightarrow{t+t'} s^{t+t'}) = P(s \xrightarrow{t} s^t) + \lambda^t P(s \xrightarrow{t'} s^{t'})$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches
2. Declare rates a *discrete* property
3. Simplify
4. Only look at paths of length 2
5. Time shift
6. Simplify

Recursive property

$$P(s \xrightarrow{t+t'} s^{t+t'}) = P(s \xrightarrow{t} s^t) + \lambda^t P(s \xrightarrow{t'} s^{t'})$$

$$f(t+t') = f(t) + g(t) f(t')$$

Inverse question: We want this property.
What way can we define $P(\pi)$?

1. Forget about switches
2. Declare rates a *discrete* property
3. Simplify
4. Only look at paths of length 2
5. Time shift
6. Simplify
7. Generalize; translate to functional equation

Uniqueness

Theorem: If P is a continuous price function in a weighted timed transition system in which rates are a discrete property, and if

$$P(s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'}) = P(s \xrightarrow{t} s^t) + g(t)\, P(s^t \xrightarrow{t'} s^{t+t'})$$

then

$$g(t) = \lambda^t \quad \text{and} \quad P(s \xrightarrow{t} s') = \alpha(s) \int_0^t \lambda^t dt$$

for some $\lambda \in \mathbb{R}_{\geq 0}$ and $\alpha : S \to \mathbb{R}$.

Uniqueness

Theorem: If P is a continuous price function in a weighted timed transition system in which rates are a discrete property, and if

$$P(s \xrightarrow{t} s^t \xrightarrow{t'} s^{t+t'}) = P(s \xrightarrow{t} s^t) + g(t)\, P(s^t \xrightarrow{t'} s^{t+t'})$$

then

$$g(t) = \lambda^t \quad \text{and} \quad P(s \xrightarrow{t} s') = \alpha(s) \int_0^t \lambda^t dt$$

for some $\lambda \in \mathbb{R}_{\geq 0}$ and $\alpha : S \to \mathbb{R}$.

If you want the nice recursive property, you *have* to use time-based discounting.
And given additivity of delays, the property is quite natural.

Skip proof

Proof

Proof: (See paper for details) We need to solve the functional equation

$$f(x+y) = f(x) + g(x)f(y)$$

and we find inspiration in Cauchy's 1821 textbook *Cours d'analyse*:

Skip anyway

Proof

Proof: (See paper for details) We need to solve the functional equation

$$f(x+y) = f(x) + g(x)f(y)$$

and we find inspiration in Cauchy's 1821 textbook *Cours d'analyse*:

- By induction:

$$f(kx) = f((k-1)x) + g((k-1)x)f(x) = f(x)\Big(\sum_{i=0}^{k-1} g(ix)\Big)$$

$$f(kx) = f(x) + g(x)f((k-1)x) = f(x)\Big(\sum_{i=0}^{k-1} (g(x))^i\Big)$$

Proof

Proof: (See paper for details) We need to solve the functional equation

$$f(x+y) = f(x) + g(x)f(y)$$

and we find inspiration in Cauchy's 1821 textbook *Cours d'analyse*:

- By induction:

$$f(kx) = f((k-1)x) + g((k-1)x)f(x) = f(x)\Big(\sum_{i=0}^{k-1} g(ix)\Big)$$

$$f(kx) = f(x) + g(x)f((k-1)x) = f(x)\Big(\sum_{i=0}^{k-1} (g(x))^i\Big)$$

- Can show that $f(x) \neq 0$ for $x \neq 0$, hence $g(ix) = g(x)^i$

Proof

Proof: (See paper for details) We need to solve the functional equation

$$f(x+y) = f(x) + g(x)f(y)$$

and we find inspiration in Cauchy's 1821 textbook *Cours d'analyse*:

- By induction:

$$f(kx) = f((k-1)x) + g((k-1)x)f(x) = f(x)\Big(\sum_{i=0}^{k-1} g(ix)\Big)$$

$$f(kx) = f(x) + g(x)f((k-1)x) = f(x)\Big(\sum_{i=0}^{k-1} (g(x))^i\Big)$$

- Can show that $f(x) \neq 0$ for $x \neq 0$, hence $g(ix) = g(x)^i$
- Put $\lambda = g(1)$, then $g(n) = \lambda^n$. Also, $g(n) = g\big(k\frac{n}{k}\big) = g\big(\frac{n}{k}\big)^k$ hence $g\big(\frac{n}{k}\big) = \lambda^{\frac{n}{k}}$. By continuity, $g(x) = \lambda^x$

Proof

Proof: (See paper for details) We need to solve the functional equation

$$f(x+y) = f(x) + g(x)f(y)$$

and we find inspiration in Cauchy's 1821 textbook *Cours d'analyse*:

- By induction:

  $$f(kx) = f((k-1)x) + g((k-1)x)f(x) = f(x)\Big(\sum_{i=0}^{k-1} g(ix)\Big)$$

  $$f(kx) = f(x) + g(x)f((k-1)x) = f(x)\Big(\sum_{i=0}^{k-1} (g(x))^i\Big)$$

- Can show that $f(x) \neq 0$ for $x \neq 0$, hence $g(ix) = g(x)^i$
- Put $\lambda = g(1)$, then $g(n) = \lambda^n$. Also, $g(n) = g\left(k\frac{n}{k}\right) = g\left(\frac{n}{k}\right)^k$ hence $g\left(\frac{n}{k}\right) = \lambda^{\frac{n}{k}}$. By continuity, $g(x) = \lambda^x$
- Put $\beta = f(1)$, then $f(n) = \beta \cdot \sum_{i=1}^{n-1} \lambda^i = \beta \frac{1-\lambda^n}{1-\lambda}$ *etc.*

Conclusion

- If you want to discuss optimal scheduling problems in formalisms with both time and weight, use time-based discounting
- (unless you have your own good reasons not to).
- Then, and only then, you'll get a natural and useful recursive property ("additivity").

- Also, for weighted timed automata, optimal infinite paths are computable under timed-based discounting,
- and the recursive property lets us hope for an efficient, zone-based algorithm.