Polynomial time algorithms for Branching Markov (Decision) Processes

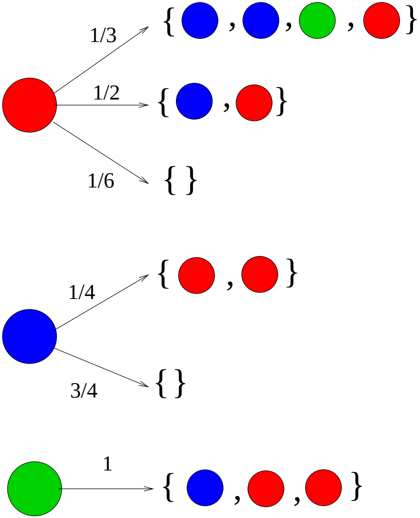
Alistair Stewart

U. of Edinburgh

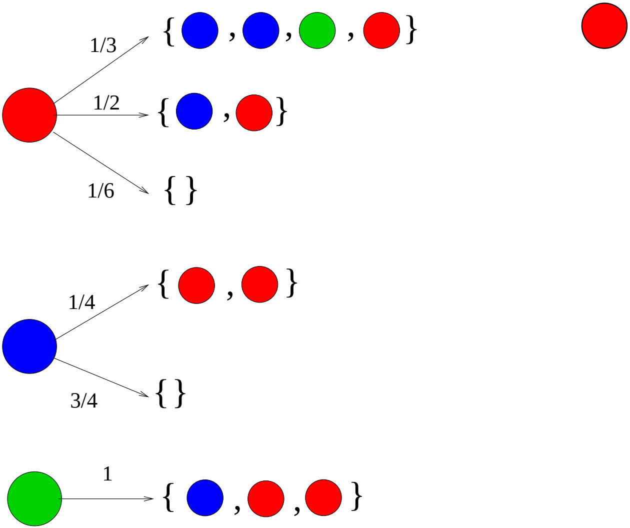
Joint work with:
Kousha Etessami, U. of Edinburgh
and
Mihalis Yannakakis, Columbia U.

CTW
August nth, 2012

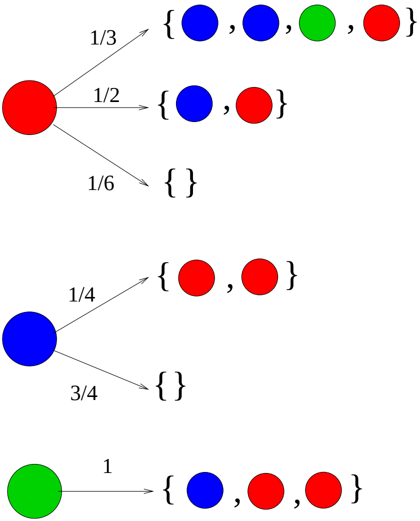
Multi-type Branching Processes (Kolmogorov,1940s)

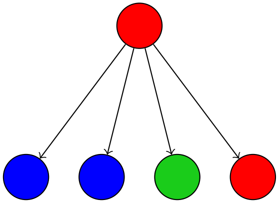


Multi-type Branching Processes (Kolmogorov,1940s)

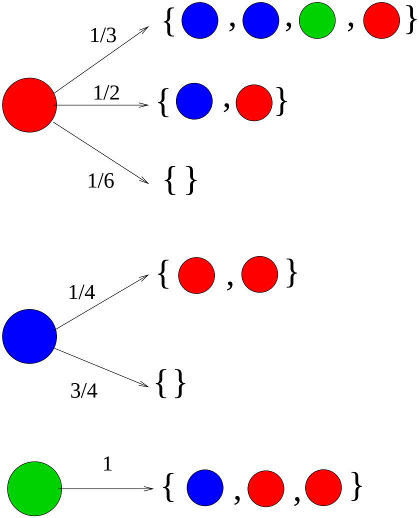


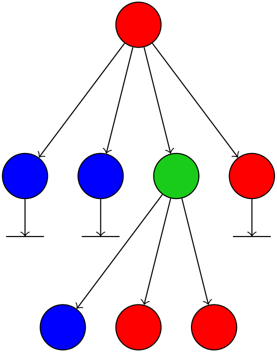
Multi-type Branching Processes (Kolmogorov,1940s)



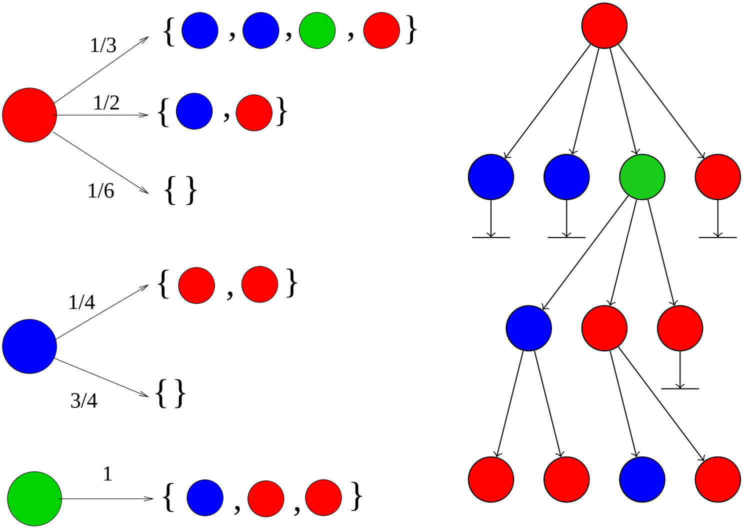


Multi-type Branching Processes (Kolmogorov,1940s)

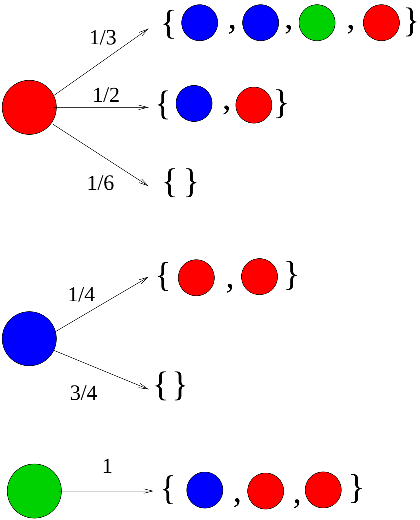


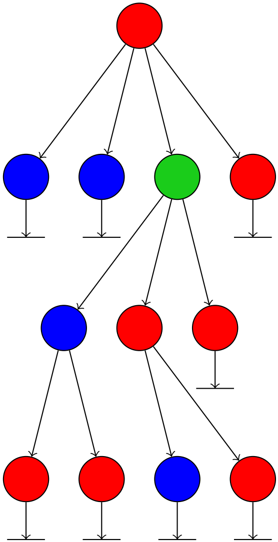


Multi-type Branching Processes (Kolmogorov,1940s)

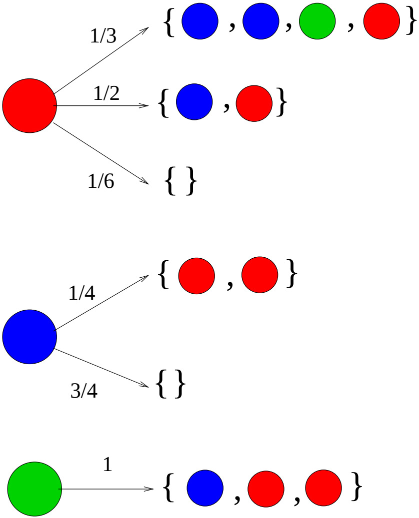


Multi-type Branching Processes (Kolmogorov,1940s)





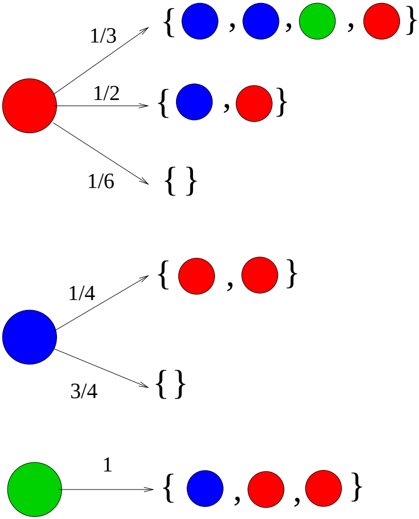
Multi-type Branching Processes (Kolmogorov 1940s)



Question

What is the probability of eventual extinction, starting with one  ?

Multi-type Branching Processes (Kolmogorov 1940s)

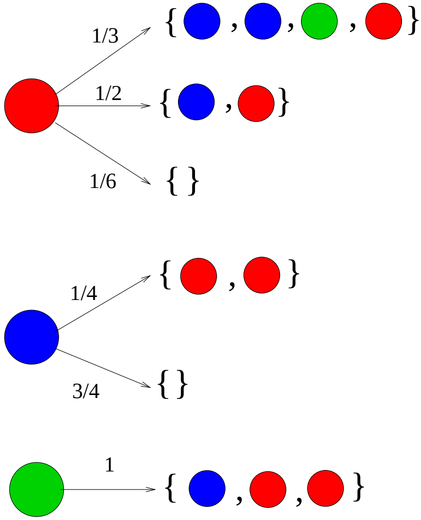


Question

What is the probability of eventual extinction, starting with one red ?

$x_R =$

Multi-type Branching Processes (Kolmogorov 1940s)

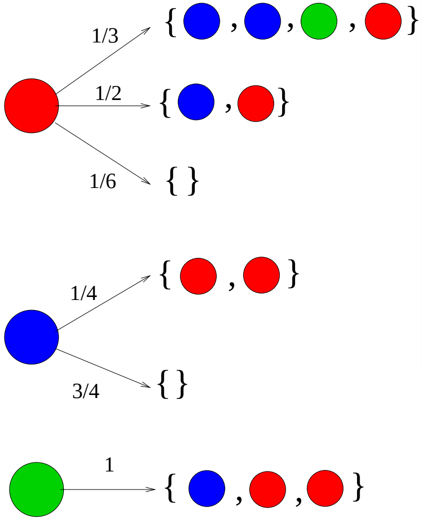


Question

What is the probability of eventual extinction, starting with one  ?

$$x_R = \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}$$

Multi-type Branching Processes (Kolmogorov 1940s)



Question

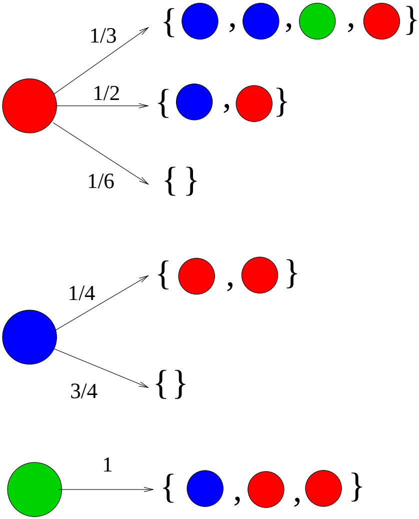
What is the probability of eventual extinction, starting with one ?

$$x_R = \frac{1}{3} x_B^2 x_G x_R + \frac{1}{2} x_B x_R + \frac{1}{6}$$

$$x_B = \frac{1}{4} x_R^2 + \frac{3}{4}$$

$$x_G = x_B x_R^2$$

Multi-type Branching Processes (Kolmogorov 1940s)



Question

What is the probability of eventual extinction, starting with one [red] ?

$$x_R = \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}$$

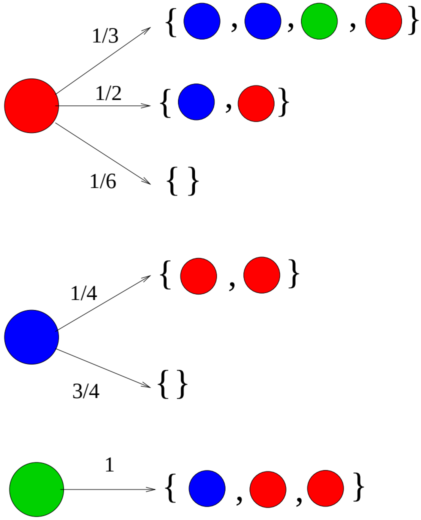
$$x_B = \frac{1}{4}x_R^2 + \frac{3}{4}$$

$$x_G = x_B x_R^2$$

We get fixed point equations,

$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$

Multi-type Branching Processes (Kolmogorov 1940s)



Question

What is the probability of eventual extinction, starting with one ?

$$x_R = \frac{1}{3} x_B^2 x_G x_R + \frac{1}{2} x_B x_R + \frac{1}{6}$$

$$x_B = \frac{1}{4} x_R^2 + \frac{3}{4}$$

$$x_G = x_B x_R^2$$

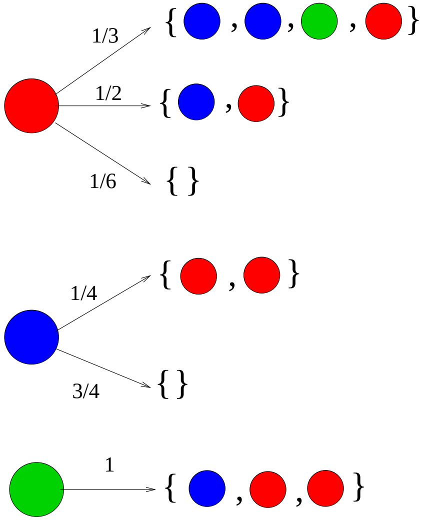
We get fixed point equations,

$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$

Fact

The extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Multi-type Branching Processes (Kolmogorov 1940s)



Question

What is the probability of eventual extinction, starting with one  ?

$$x_R = \frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}$$
$$x_B = \frac{1}{4}x_R^2 + \frac{3}{4}$$
$$x_G = x_B x_R^2$$

We get fixed point equations,

$$\bar{\mathbf{x}} = P(\bar{\mathbf{x}}).$$

Fact

The extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.
$q^*_R = 0.276$; $q^*_B = 0.769$; $q^*_G = 0.059$.

Probabilistic Polynomial Systems of Equations

$$\frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}$$

is a Probabilistic Polynomial: the coefficients are positive and sum to 1.

A Probabilistic Polynomial System (PPS) is a system

$$\mathbf{x}_i = P_i(\mathbf{x}) \qquad i = 1, \ldots, n$$

of n equations in n variables, where each $P_i(x)$ is a probabilistic polynomial. We denote the entire system by:

$$\mathbf{x} = P(\mathbf{x})$$

Every multi-type Branching Process (BP) with n types corresponds to such a PPS and vice-versa.

Basic properties of PPSs, $\mathbf{x} = P(\mathbf{x})$

$P(x)$ defines a continuous map, $P : [0,1]^n \rightarrow [0,1]^n$.
$P : [0,1]^n \rightarrow [0,1]^n$ defines a monotone operator on $[0,1]^n$.

Proposition

- *Every PPS, $x = P(x)$ has a least fixed point (LFP), $q^* \in [0,1]^n$. (q^* can be irrational.)*
- *q^* is the vector of extinction/termination probabilities for the corresponding BP (SCFG).*

Question

Can we compute the probabilities q^* efficiently (in P-time)?

Main Result: P-time approximation

Answer

Yes

Theorem (Main Result of [Etessami-S.-Yannakakis, STOC '12])

Given a PPS, $\mathbf{x} = P(\mathbf{x})$, with LFP $\mathbf{q}^* \in [0,1]^n$, we can compute a rational vector $\mathbf{v} \in [0,1]^n$ such that $||\mathbf{v} - \mathbf{q}^*||_\infty \leq 2^{-j}$, in time polynomial in the encoding size $|P|$ of the equations, and in j.

Newton's method

Newton's method

Seeking a solution to $F(\mathbf{x}) = 0$, we start at a guess $\mathbf{x}^{(0)}$, and iterate:

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} - (F'(\mathbf{x}^{(k)}))^{-1} F(\mathbf{x}^{(k)})$$

Here $F'(\mathbf{x})$, is the *Jacobian matrix*:

$$F'(\mathbf{x}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}$$

For PPSs, Newton iteration looks like this:

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} + (I - P'(\mathbf{x}^{(k)}))^{-1} (P(\mathbf{x}^{(k)}) - \mathbf{x}^{(k)})$$

where $P'(\mathbf{x})$ is the Jacobian of $P(\mathbf{x})$.

Newton on PPSs

We can decompose $\mathbf{x} = P(\mathbf{x})$ into its strongly connected components (SCCs), based on variable dependencies, and eliminate "0" variables.

[Etessami-Yannakakis'05] showed that this decomposed Newton's method converges monotonically to the LFP $\mathbf{q}^*$.

But...

- They gave no upper bounds for the number of iterations needed for PPSs.
- They proved hardness results for approximating the LFP of Monotone Polynomial Systems (MPSs), for which the same Newton's method works.

What is Newton's worst case behavior for PPSs?

[Esparza,Kiefer,Luttenberger,2010] studied Newton's method on MPSs

- Gave bad examples of PPSs, $\mathbf{x} = P(\mathbf{x})$, requiring exponentially many iterations, as a function of the encoding size $|P|$ of the equations, to converge to within additive error $< 1/2$.
- For strongly-connected equation systems they gave an exponential upper bound.
- They gave no upper bounds for arbitrary PPSs.

Qualitative problems for PPSs are in P-time

Theorem ([Kolmogorov-Sevastyanov'47,Harris'63])

For certain classes of strongly-connected PPSs, $q_i^* = 1$ for all i if the spectral radius of the "moment matrix", $P'(1)$, is > 1. Otherwise $q_i^* < 1$ for all i.

Theorem ([Etessami-Yannakakis'05])

For a PPS, $\mathbf{x} = P(\mathbf{x})$, deciding whether $q_i^* = 1$ is in P-time.

(Deciding whether $q_i^* = 0$ is also in P-time.)

Algorithm

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$.

Algorithm

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$.

Theorem

Given a PPS $\mathbf{x} = P(\mathbf{x})$ with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, if we apply Newton starting at $\mathbf{x}^{(0)} = \mathbf{0}$, then after $k \geq 4|P| + j$ iterations $||\mathbf{q}^* - \mathbf{x}^{(k)}|| \leq 2^{-j}$.

Algorithm with rounding

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$.
3. After each iteration, round down to a multiple of 2^{-h}

Theorem

If, after each Newton iteration, we round down to a multiple of 2^{-h} where $h := 4|P| + j + 2$, after h iterations $\|\mathbf{q}^* - \mathbf{x}^{(h)}\|_\infty \leq 2^{-j}$.

Thus we have a P-time algorithm (in the standard Turing model).

Proof outline: key lemmas

$(\mathbf{1} - \mathbf{q}^*)$ is the vector of non-termination, or survival, probabilities.

Proof outline: key lemmas

$(\mathbf{1} - \mathbf{q}^*)$ is the vector of non-termination, or survival, probabilities.

Lemma

If $\mathbf{q}^* - \mathbf{x}^{(k)} \leq \lambda(\mathbf{1} - \mathbf{q}^*)$ *for some* $\lambda > 0$, *then* $\mathbf{q}^* - \mathbf{x}^{(k+1)} \leq \frac{\lambda}{2}(\mathbf{1} - \mathbf{q}^*)$.

Proof outline: key lemmas

$(\mathbf{1} - \mathbf{q}^*)$ is the vector of non-termination, or survival, probabilities.

Lemma

If $\mathbf{q}^* - \mathbf{x}^{(k)} \leq \lambda(\mathbf{1} - \mathbf{q}^*)$ *for some* $\lambda > 0$, *then* $\mathbf{q}^* - \mathbf{x}^{(k+1)} \leq \frac{\lambda}{2}(\mathbf{1} - \mathbf{q}^*)$.

Lemma

For any PPS with LFP $\mathbf{q}^*$, *such that* $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, *for any* i,

$$q_i^* \leq 1 - 2^{-4|P|}.$$

Proof outline: key lemmas

$(\mathbf{1} - \mathbf{q}^*)$ is the vector of non-termination, or survival, probabilities.

Lemma

If $\mathbf{q}^* - \mathbf{x}^{(k)} \leq \lambda(\mathbf{1} - \mathbf{q}^*)$ *for some* $\lambda > 0$, *then* $\mathbf{q}^* - \mathbf{x}^{(k+1)} \leq \frac{\lambda}{2}(\mathbf{1} - \mathbf{q}^*)$.

Lemma

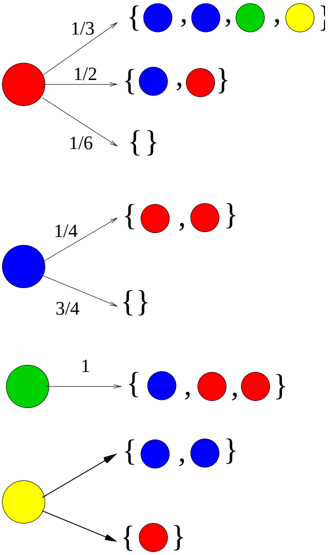
For any PPS with LFP $\mathbf{q}^*$, *such that* $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, *for any* i,

$$q_i^* \leq 1 - 2^{-4|P|}.$$

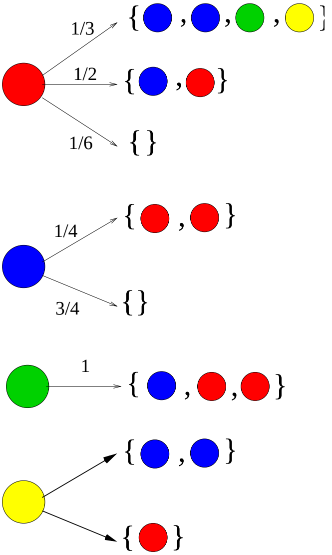
Theorem

Given a PPS $\mathbf{x} = P(\mathbf{x})$ *with LFP* $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, *if we apply Newton starting at* $\mathbf{x}^{(0)} = \mathbf{0}$, *then after* $k \geq 4|P| + j$ *iterations* $||\mathbf{q}^* - \mathbf{x}^{(k)}|| \leq 2^{-j}$.

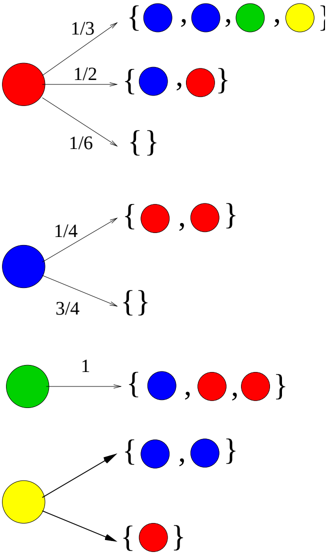
Branching Markov Decision Processes

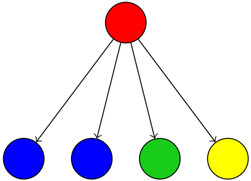


Branching Markov Decision Processes

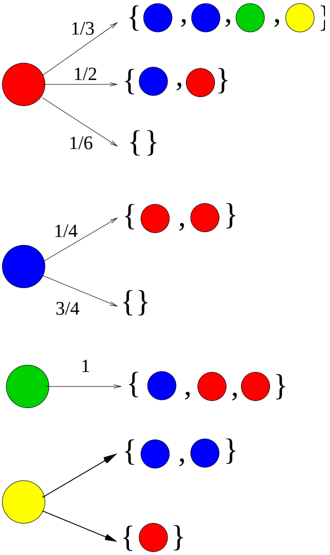


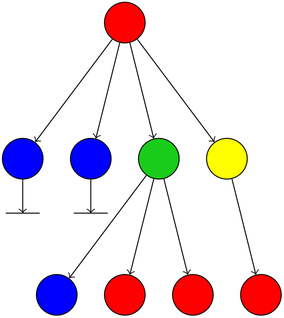
Branching Markov Decision Processes



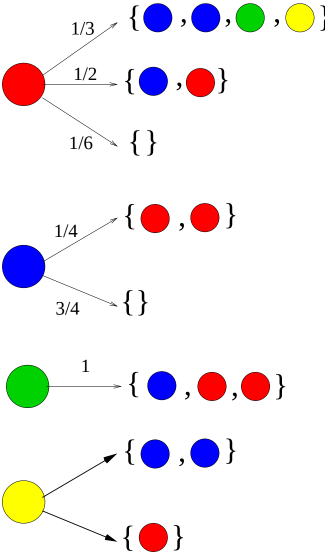


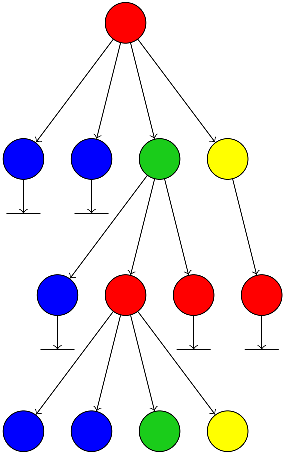
Branching Markov Decision Processes



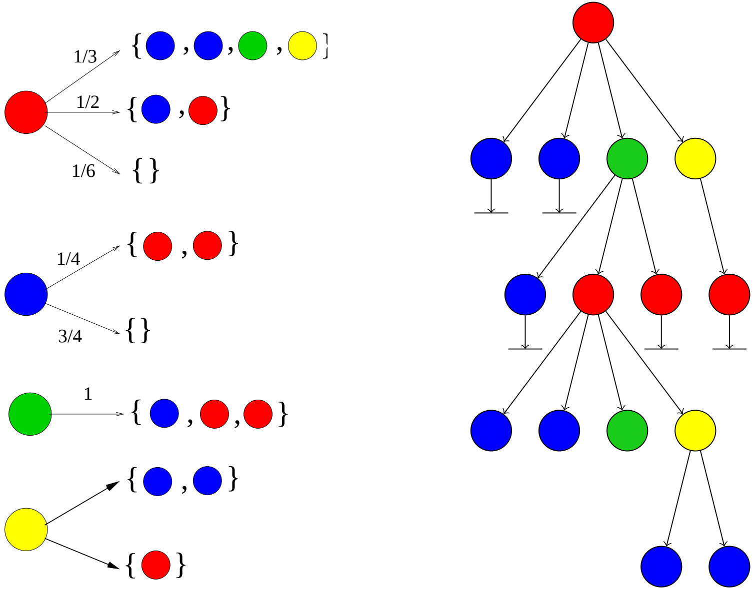


Branching Markov Decision Processes

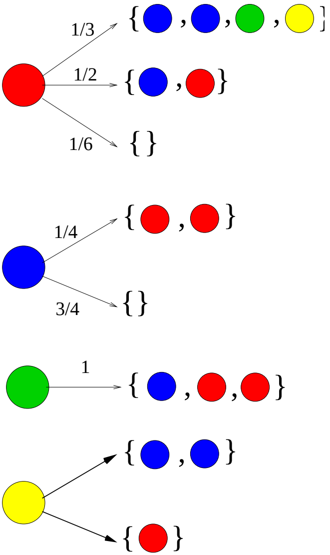




Branching Markov Decision Processes



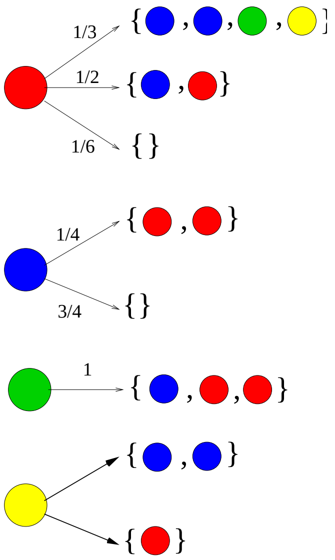
Branching Markov Decision Processes



Question

What is the maximum probability of extinction, starting with one ?

Branching Markov Decision Processes



Question

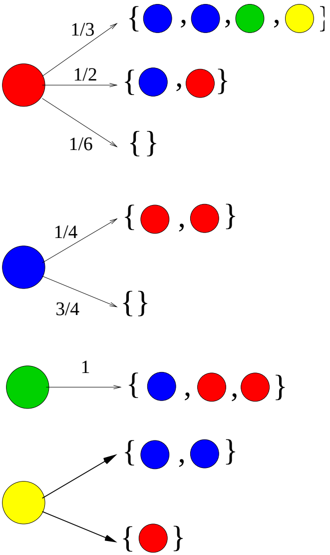
What is the maximum probability of extinction, starting with one ?

$$x_R = \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6}$$

$$x_B = \frac{1}{4}x_R^2 + \frac{3}{4}$$

$$x_G = x_B x_R^2$$

Branching Markov Decision Processes

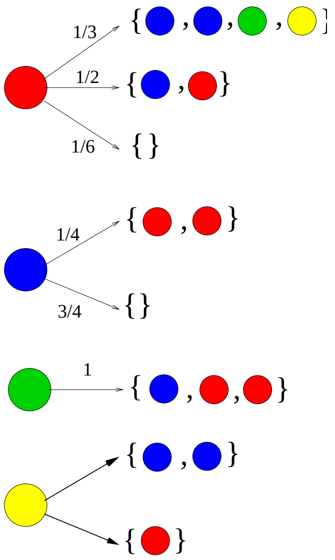


Question

What is the maximum probability of extinction, starting with one [red] ?

$$x_R = \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6}$$
$$x_B = \frac{1}{4}x_R^2 + \frac{3}{4}$$
$$x_G = x_B x_R^2$$
$$x_Y = \max\{x_B^2, x_R\}$$

Branching Markov Decision Processes



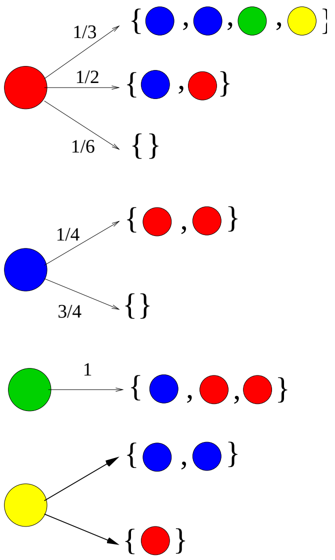
Question

What is the maximum probability of extinction, starting with one  ?

$$
\begin{aligned}
x_R &= \frac{1}{3} x_B^2 x_G x_Y + \frac{1}{2} x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4} x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2 \\
x_Y &= \max\{x_B^2, x_R\}
\end{aligned}
$$

We get fixed point equations, $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Branching Markov Decision Processes



Question

What is the maximum probability of extinction, starting with one red ?

$$x_R = \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6}$$

$$x_B = \frac{1}{4}x_R^2 + \frac{3}{4}$$

$$x_G = x_B x_R^2$$

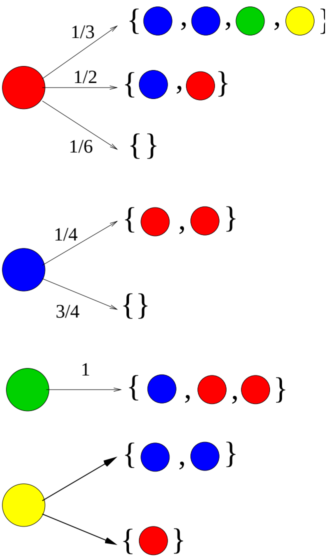
$$x_Y = \max\{x_B^2, x_R\}$$

We get fixed point equations, $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Fact

The maximum extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Branching Markov Decision Processes



Question

What is the minimum probability of extinction, starting with one  ?

$$
\begin{aligned}
x_R &= \frac{1}{3}x_B^2 x_G x_Y + \frac{1}{2}x_B x_R + \frac{1}{6} \\
x_B &= \frac{1}{4}x_R^2 + \frac{3}{4} \\
x_G &= x_B x_R^2 \\
x_Y &= \min\{x_B^2, x_R\}
\end{aligned}
$$

We get fixed point equations, $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Fact

The minimum extinction probabilities are the least fixed point, $\mathbf{q}^* \in [0,1]^3$, of $\bar{\mathbf{x}} = P(\bar{\mathbf{x}})$.

Maximum Probabilistic Polynomial Systems of Equations

$$\frac{1}{3}x_B^2 x_G x_R + \frac{1}{2}x_B x_R + \frac{1}{6}$$

is a Probabilistic Polynomial: the coefficients are positive and sum to 1.

A Maximum Probabilistic Polynomial System (maxPPS) is a system

$$\mathbf{x}_i = \max\{p_{i,j}(\mathbf{x}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

of n equations in n variables, where each $p_{i,j}(x)$ is a probabilistic polynomial. We denote the entire system by:

$$\mathbf{x} = P(\mathbf{x})$$

These are the Bellman equations for a maximizing BMDP with n types

A max/minPPS is either a maxPPS or an minPPS

Basic properties of max/minPPSs, $\mathbf{x} = P(\mathbf{x})$

$P(x)$ defines a continuous map, $P : [0,1]^n \to [0,1]^n$.
$P : [0,1]^n \to [0,1]^n$ defines a monotone operator on $[0,1]^n$.

Proposition

- *Every max/minPPS, $x = P(x)$ has a least fixed point (LFP), $q^* \in [0,1]^n$.*
  (q^* can be irrational.)
- *q^* is the vector of optimal extinction probabilities for the corresponding BMDP.*

Question

Can we compute the probabilities q^* efficiently (in P-time)?

Main Result: P-time approximation

Answer

Yes

Theorem (Main Result of [Etessami-S.-Yannakakis, ICALP '12])

Given a max/minPPS, $\mathbf{x} = P(\mathbf{x})$, with LFP $\mathbf{q}^* \in [0,1]^n$, we can compute a rational vector $\mathbf{v} \in [0,1]^n$ such that $||\mathbf{v} - \mathbf{q}^*||_\infty \leq 2^{-j}$, in time polynomial in the encoding size $|P|$ of the equations, and in j.

if we had no nonlinear terms...

- If the polynomials $p_{i,j}(x)$ were all *linear*, then $x = P(x)$ are the Bellman optimality equations for maximizing/minimizing the *hitting (reachability) probability* on a finite-state MDP.
- The LFP solution, q^*, yields the optimal probabilities. In the maximization case, q^* can be computed easily in P-time using LP:

$$\begin{array}{ll} \textbf{minimize}: \sum_i x_i; & \\ \textbf{subject to}: & \\ p_{i,j}(x) \leq x_i, & \text{for all } i,j \\ x_i \geq 0 & \text{for all } i \end{array}$$

- In the minimization case, a little graph-theoretic analysis, combined with a different LP, can be used to solve the problem in P-time.

A Newton iteration as a first-order approximation

An iteration of Newton's method on a PPS, applied on current vector $y \in \mathbb{R}^n$, solves the equation

$$P^{\mathbf{y}}(\mathbf{x}) = \mathbf{x}$$

where $P^{\mathbf{y}}(\mathbf{x}) \equiv P(\mathbf{y}) + P'(\mathbf{y})(\mathbf{x} - \mathbf{y})$ is a linear approximation of $P(x)$

Generalised Newton's method

Linearisation

Given a maxPPS

$$(P(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{x}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

We define the linearisation, $P^y(x)$, by:

$$(P^{\mathbf{y}}(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{y}) + \nabla p_{i,j}(\mathbf{y}).(\mathbf{x} - \mathbf{y}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

Generalised Newton's method

Linearisation

Given a maxPPS

$$(P(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{x}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

We define the linearisation, $P^y(x)$, by:

$$(P^{\mathbf{y}}(\mathbf{x}))_i = \max\{p_{i,j}(\mathbf{y}) + \nabla p_{i,j}(\mathbf{y}).(\mathbf{x} - \mathbf{y}) : j = 1, \ldots, m_i\} \qquad i = 1, \ldots, n$$

Generalised Newton's method, applied at vector y

For a maxPPS,

$$\text{minimize } \textstyle\sum_i x_i \text{ subject to } P^{\mathbf{y}}(\mathbf{x}) \leq \mathbf{x};$$

For a minPPS,

$$\text{maximize } \textstyle\sum_i x_i \text{ subject to } P^{\mathbf{y}}(\mathbf{x}) \geq \mathbf{x};$$

These give a solution to $P^{\mathbf{y}}(\mathbf{x}) = \mathbf{x}$, and yield the GNM iteration we need.

Algorithm for PPSs(Recap)

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$. After each iteration, round down to a multiple of 2^{-h}

Algorithm for PPSs(Recap)

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$.
2. On the resulting system of equations, run Newton's method starting from $\mathbf{0}$. After each iteration, round down to a multiple of 2^{-h}

Theorem

Given a PPS $\mathbf{x} = P(\mathbf{x})$ with LFP $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$, if we apply Newton starting at $\mathbf{x}^{(0)} = \mathbf{0}$, using $h := 4|P| + j + 1$ bits of precision, then after $k \geq 4|P| + j + 1$ iterations $||\mathbf{q}^* - \mathbf{x}^{(k)}|| \leq 2^{-j}$. We can do all this in time polynomial in $|P|$ and j.

Algorithm for max/minPPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$. (Computable in P-time [Etessami-Yannakakis'06].)
2. On the resulting system of equations, run Generalized Newton's Method, starting from $\mathbf{0}$. After each iteration, round down to a multiple of 2^{-h}.
   Each iteration of GNM can be computed in P-time by solving an LP.

Algorithm for max/minPPSs

1. Find and remove all variables x_i such that $q_i^* = 0$ or $q_i^* = 1$. (Computable in P-time [Etessami-Yannakakis'06].)
2. On the resulting system of equations, run Generalized Newton's Method, starting from $\mathbf{0}$. After each iteration, round down to a multiple of 2^{-h}.
   Each iteration of GNM can be computed in P-time by solving an LP.

Theorem

Given a max/minPPS $\mathbf{x} = P(\mathbf{x})$ *with LFP* $\mathbf{0} < \mathbf{q}^* < \mathbf{1}$*, if we apply GNM starting at* $\mathbf{x}^{(0)} = \mathbf{0}$*, using* $h := 4|P| + j + 1$ *bits of precision, then after* $k \geq 4|P| + j + 1$ *iterations* $||\mathbf{q}^* - \mathbf{x}^{(k)}|| \leq 2^{-j}$*. We can do all this in time polynomial in* $|P|$ *and* j*.*

Optimal and ϵ-optimal policies/strategies for max/minPPSs

Theorem ([Etessami-Yannakakis'05])

Any (maximizing or minimizing) BMDP has a static optimal policy.

A static policy (or strategy) is one that, for every controlled type, always deterministically chooses the same single rule.

Computing an optimal policy is hard (as hard as the SQRT-SUM and PosSLP problem).

Theorem

Given a BMDP (or max/minPPS $x = P(x)$), and given $\epsilon > 0$, we can compute an ϵ-optimal static policy in time polynomial in $|P|$ and $\log(1/\epsilon)$.

- First compute an approximation z to q^* of the corresponding max/minPPS with $\|q^* - z\|_\infty \leq 2^{-14|P|-2}\epsilon$.
- For a minimizing BMDP, we choose, for the type corresponding to x_i, the rule which gives the lowest approximate minimal extinction probability $p_{i,j}(y)$.
- The maximizing BMDP case is more complicated but we start in a similar way.

Conclusions

- P-time algorithms for computing extinction probabilities for MT-BPs.
- We can approximate the optimal extinction probabilities of a maximizing or minimizing Branching Markov Decision Process in polynomial time.
- We can compute ϵ-optimal policies of a BMDP in polynomial time.