Controlling a population of identical NFA

Nathalie Bertrand

Inria Rennes

joint work with Miheer Dewaskar (ex CMI student), Blaise Genest (IRISA) and Hugo Gimbert (LaBRI)

SynCoP & PV workshops @ ETAPS 2018
Motivation

Control of gene expression for a population of cells

credits: G. Batt
Motivation

Control of gene expression for a population of cells

- cell population
- gene expression monitored through fluorescence level
- drug injections affect all cells
- response varies from cell to cell
- obtain a large proportion of cells with desired gene expression level
Motivation

Control of gene expression for a population of cells

- cell population
- gene expression monitored through fluorescence level
- drug injections affect all cells
- response varies from cell to cell
- obtain a large proportion of cells with desired gene expression level
- arbitrary nb of components
- full observation
- uniform control
- non-det. model for single cell
- global reachability objective
Problem formalisation

- population of $N$ identical NFA
- uniform control policy under full observation
- resolution of non-determinism by an adversary
Problem formalisation

- population of $N$ identical NFA
- uniform control policy under full observation
- resolution of non-determinism by an adversary

config:  # copies in each state
Problem formalisation

- population of $N$ identical NFA
- uniform control policy under full observation
- resolution of non-determinism by an adversary

config: # copies in each state

- controller chooses the action (e.g. $a$)
Problem formalisation

- population of $N$ identical NFA
- uniform control policy under full observation
- resolution of non-determinism by an adversary

Controller chooses the action (e.g. $a$)
Adversary chooses how to move each individual copy ($a$-transition)
Problem formalisation

- population of $N$ identical NFA
- uniform control policy under full observation
- resolution of non-determinism by an adversary

config: # copies in each state

- controller chooses the action (e.g. $a$)
- adversary chooses how to move each individual copy ($a$-transition)
Problem formalisation

- population of $N$ identical NFA
- uniform control policy under full observation
- resolution of non-determinism by an adversary

config: # copies in each state

- controller chooses the action (e.g. $a$)
- adversary chooses how to move each individual copy ($a$-transition)
Problem formalisation

- population of $N$ identical NFA
- uniform control policy under full observation
- resolution of non-determinism by an adversary

controller chooses the action (e.g. $a$)
adversary chooses how to move each individual copy ($a$-transition)

Question can one control the population to ensure that for all non-deterministic choices all NFAs simultaneously reach a target set?
Fixed $N$: build finite 2-player game, identify global target states, decide if controller has a winning strategy for a reachability objective.
Population control

**Fixed** $N$: build finite 2-player game, identify global target states, decide if controller has a winning strategy for a reachability objective

**Challenge:** Parameterized control

$$\forall N \exists \sigma \forall \tau (A^N, \sigma, \tau) \models \Diamond F^N?$$
Population control

**Fixed** $N$: build finite 2-player game, identify global target states, decide if controller has a winning strategy for a reachability objective

**Challenge:** Parameterized control

\[
\forall N \, \exists \sigma \, \forall \tau \, (A^N, \sigma, \tau) \models \Diamond F^N?
\]

This talk
- decidability and complexity
- memory requirements for controller $\sigma$
- admissible values for $N$
Monotonicity property

Monotonicity property: the larger $N$, the harder for controller

$$\exists \sigma \ \forall \tau(A^N, \sigma, \tau) \models \lozenge F^N \quad \implies \quad \forall M \leq N \ \exists \sigma \ \forall \tau(A^M, \sigma, \tau) \models \lozenge F^M$$
Monotonicity property and cutoff

**Monotonicity property:** the larger $N$, the harder for controller

$$
\exists \sigma \ \forall \tau (\mathcal{A}^N, \sigma, \tau) \models \Diamond F^N \quad \Rightarrow \quad \forall M \leq N \ \exists \sigma \ \forall \tau (\mathcal{A}^M, \sigma, \tau) \models \Diamond F^M
$$

**Cutoff:** smallest $N$ for which controller has no winning strategy
Monotonicity property and cutoff

**Monotonicity property:** the larger $N$, the harder for controller

$$\exists \sigma \ \forall \tau (A^N, \sigma, \tau) \models \lozenge F^N \implies \forall M \leq N \ \exists \sigma \ \forall \tau (A^M, \sigma, \tau) \models \lozenge F^M$$

**Cutoff:** smallest $N$ for which controller has no winning strategy

- winning $\sigma$ if $N < M$
- play $b$ then $a_i$ s.t. $q_i$ is empty
- winning $\tau$ for $N = M$
- always fill all $q_i$’s
- cutoff is $M$

Unspecified edges lead to a sink state.
Lower bound on the cutoff

\[ \forall N \leq 2^M, \ \exists \sigma, \ A^N \models \forall \sigma \diamond F^N \]
accumulate copies in bottom states, then \( u/d \) to converge

\[ \forall N > 2^M \text{ controller cannot avoid reaching the sink state} \]

\[ \text{Cutoff } \mathcal{O}(2^{|A|}) \]
Lower bound on the cutoff

\[ \forall N \leq 2^M, \ \exists \sigma, \ \mathcal{A}^N \models \forall \sigma \diamond F^N \]
accumulate copies in bottom states, then \( u/d \) to converge

\[ \forall N > 2^M \] controller cannot avoid reaching the sink state

Cutoff \( \mathcal{O}(2^{\left| A \right|}) \)

Controlling a population of NFA – Nathalie Bertrand
Lower bound on the cutoff

$$\forall N \leq 2^M, \exists \sigma, A^N \models \forall \sigma \Diamond F^N$$
accumulate copies in bottom states, then $u/d$ to converge

$$\forall N > 2^M$$ controller cannot avoid reaching the sink state

Cutoff $O(2^{|A|})$
Lower bound on the cutoff

\[ \forall N \leq 2^M, \exists \sigma, A^N \models \forall \sigma \Diamond F^N \]
accumulate copies in bottom states, then \( u/d \) to converge

\[ \forall N > 2^M \text{ controller cannot avoid reaching the sink state} \]

\[ \text{Cutoff } O(2^{|A|}) \]
Lower bound on the cutoff

![Diagram of a graph representing a population of NFA with nodes labeled by states and arrows indicating transitions. The diagram shows a path from a starting state labeled 'a', 'b', 'c', 'd' to a sink state labeled 'F' with additional states labeled 'u', 'd'.]

- \( \forall N \leq 2^M, \exists \sigma, \mathcal{A}^N \models \forall \sigma \Diamond F^N \)
- Accumulate copies in bottom states, then \( u/d \) to converge
- For \( N > 2^M \) controller cannot avoid reaching the sink state

Cutoff \( O(2^{|A|}) \)
Lower bound on the cutoff

\[ \forall N \leq 2^M, \exists \sigma, A^N \models \forall \sigma \diamond F^N \]
accumulate copies in bottom states, then \( u/d \) to converge

\[ \forall N > 2^M \text{ controller cannot avoid reaching the sink state} \]

\[ \text{Cutoff } O(2^{|A|}) \]
Lower bound on the cutoff

\[ \forall N \leq 2^M, \ \exists \sigma, \ A^N \models \forall \sigma \diamond F^N \]
accumulate copies in bottom states, then \( u/d \) to converge

\[ \forall N > 2^M \] controller cannot avoid reaching the sink state

Cutoff \( \mathcal{O}(2^{|A|}) \)

\[ 2M \text{ bottom states (here 6)} \]
Lower bound on the cutoff

∀N ≤ 2^M, ∃σ, A^N |= ∀σ ◻ F^N
accumulate copies in bottom states, then u/d to converge

for N > 2^M controller cannot avoid reaching the sink state

Cutoff O(2^|A|)
Lower bound on the cutoff

\[ \forall N \leq 2^M, \ \exists \sigma, \ \mathcal{A}^N \models \forall \sigma \diamond F^N \]
accumulate copies in bottom states, then \( u/d \) to converge

\[ \text{for } N > 2^M \text{ controller cannot avoid reaching the sink state} \]

Cutoff \( O(2^{|A|}) \)
Lower bound on the cutoff

\[ \forall N \leq 2^M, \ \exists \sigma, \ A^N \models \forall \sigma \Diamond F^N \]
accumulate copies in bottom states, then \( u/d \) to converge

\[ \forall N > 2^M \text{ controller cannot avoid reaching the sink state} \]

Cutoff \( O(2^{|A|}) \)

Combined with a counter, cutoff is even doubly exponential!
A natural attempt: the support game

Assumption: if state 2 or 4 is empty, controller wins
A natural attempt: the support game

Assumption: if state 2 or 4 is empty, controller wins

**Support game:** □ Eve chooses action
◇ Adam chooses transfer graph (footprint of copies' moves)
A natural attempt: the support game

Assumption: if state 2 or 4 is empty, controller wins

Support game: □ Eve chooses action
◊ Adam chooses transfer graph (footprint of copies’ moves)

objective for Eve: reach green states
A natural attempt: the support game

Assumption: if state 2 or 4 is empty, controller wins

Support game: □ Eve chooses action  
◇ Adam chooses transfer graph (footprint of copies' moves)

objective for Eve: reach green states

If Eve wins support game then controller has a winning strategy for all $N$
Support game is not equivalent to population game

- controller alternates $a$ and $b$;
- adversary must always fill 2 and 4 in the $b$-step.
Support game is not equivalent to population game

- controller alternates $a$ and $b$;
- adversary must always fill 2 and 4 in the $b$-step

Controlling a population of NFA – Nathalie Bertrand
Support game is not equivalent to population game

- controller alternates \( a \) and \( b \);
- adversary must always fill 2 and 4 in the \( b \)-step
Support game is not equivalent to population game

- controller alternates $a$ and $b$;
- adversary must always fill 2 and 4 in the $b$-step

ininitely many leaks from the white flow
Support game is not equivalent to population game

- controller alternates $a$ and $b$;
- adversary must always fill 2 and 4 in the $b$-step

Above play from support game is not realisable in population control

- Controller wins with $(ab)\omega$!
- Eve loses the support game
Capacity game: refining winning condition of support game

Finite capacity play: all accumulators have finitely many entries

Bounded capacity play: finite bound on \# entries for accumulators

Bounded capacity corresponds to realizable plays

Eve wins capacity game iff Controller has a winning strategy for all $N$. 

Controlling a population of NFA – Nathalie Bertrand
Finite capacity play: all accumulators have finitely many entries
Bounded capacity play: finite bound on \# entries for accumulators
Capacity game: refining winning condition of support game

Finite capacity play: all accumulators have finitely many entries
Bounded capacity play: finite bound on \# entries for accumulators

Bounded capacity
- corresponds to realizable plays
- does not seem to be regular
Capacity game: refining winning condition of support game

Finite capacity play: all accumulators have finitely many entries
Bounded capacity play: finite bound on \# entries for accumulators

Bounded capacity

- corresponds to realizable plays
- does not seem to be regular

Capacity game: Eve wins a play if either it reaches a subset of \( F \), or it does not have finite capacity.
Capacity game: refining winning condition of support game

Finite capacity play: all accumulators have finitely many entries
Bounded capacity play: finite bound on \# entries for accumulators

Bounded capacity
\> corresponds to realizable plays
\> does not seem to be regular

Capacity game: Eve wins a play if either it reaches a subset of \( F \), or it does not have finite capacity.

Eve wins capacity game iff Controller has a winning strategy for all \( N \)
Solving the capacity game in 2EXPTIME

The set of plays with infinite capacity is $\omega$-regular
Solving the capacity game in 2EXPTIME

The set of plays with infinite capacity is ω-regular
Solving the capacity game in 2EXPTIME

The set of plays with infinite capacity is $\omega$-regular

Non-deterministic Büchi automaton

1. guesses a step $i$, and state $q$
2. checks that the accumulator $\text{Future}(q, i)$ has infinitely many entries
Solving the capacity game in $2\text{EXPTIME}$

The set of plays with infinite capacity is $\omega$-regular

Non-deterministic Büchi automaton

1. guesses a step $i$, and state $q$
2. checks that the accumulator $\text{Future}(q, i)$ has infinitely many entries

- Non-det. Büchi determinized into det. parity automaton
- Resolution of doubly exp. parity game
Solving the capacity game in 2EXPTIME

The set of plays with infinite capacity is $\omega$-regular

Non-deterministic Büchi automaton

1. guesses a step $i$, and state $q$
2. checks that the accumulator $\text{Future}(q, i)$ has infinitely many entries

- Non-det. Büchi determinized into det. parity automaton
- Resolution of doubly exp. parity game

2EXPTIME decision procedure in the size of NFA $\mathcal{A}$
Solving the capacity game in EXPTIME

Ad-hoc deterministic parity automaton with

\# states = simply exponential in \(|A|\)  \# priorities = polynomial in \(|A|\)
Solving the capacity game in EXPTIME

Ad-hoc deterministic parity automaton with

\# states = simply exponential in |A| \quad \# priorities = polynomial in |A|

\[ G \xrightarrow{x} y \text{ enters accumulator } \text{Future}(q) \]

\[ H \]

- entries arise from separated pairs
- tracking transfer graphs separating new pairs is sufficient

Parity game: capacity game enriched with tracking lists in states
Priorities reflect how the tracking list evolves (removals, shifts, etc.)

Parity game is equivalent to capacity game.

Controlling a population of NFA – Nathalie Bertrand
Solving the capacity game in EXPTIME

Ad-hoc deterministic parity automaton with

\[ \# \text{ states } = \text{ simply exponential in } |A| \quad \# \text{ priorities } = \text{ polynomial in } |A| \]

\[
\begin{align*}
q & \rightarrow \quad x \rightarrow y \\
H & \quad G
\end{align*}
\]

\[ x \rightarrow y \text{ enters accumulator Future}(q) \]

\[ G \text{ separates pair } (t, x) \]

- entries arise from separated pairs
- tracking transfer graphs separating new pairs is sufficient

**Parity game:**
capacity game enriched with tracking lists in states
priorities reflect how the tracking list evolves (removals, shifts, etc.)
Solving the capacity game in EXPTIME

Ad-hoc deterministic parity automaton with

\# states = simply exponential in |A|  \# priorities = polynomial in |A|

\[
\begin{align*}
G & \quad x \rightarrow y \text{ enters accumulator } \text{Future}(q) \\
H & \quad G \text{ separates pair } (t, x)
\end{align*}
\]

- entries arise from separated pairs
- tracking transfer graphs separating new pairs is sufficient

**Parity game:**

capacity game enriched with tracking lists in states
priorities reflect how the tracking list evolves (removals, shifts, etc.)

Parity game is equivalent to capacity game.
Complexity of the population control problem

**Theorem:**
The population control problem is EXPTIME-complete.

**Upper bound:**
- population control problem $\equiv$ capacity game
- capacity game $\equiv$ ad hoc parity game
- solving parity game of size exp. and poly. priorities

**Lower bound:** encoding of poly space alternating Turing machine
Uniform control of a population of identical NFA

- parameterized control problem: gather all copies in $F$
- (surprisingly) quite involved!
- tight results for complexity, cutoff, and memory
  - complexity: EXPTIME-complete decision problem
  - bound on cutoff: doubly exponential
  - memory requirement: exponential memory (orthogonal to supports) is needed and sufficient for controller
Back to motivations

Control of gene expression for a population of cells

credits: G. Batt
Back to motivations

Control of gene expression for a population of cells

 credits: G. Batt

▶ need for truly probabilistic model
  → MDP instead of NFA
▶ need for truly quantitative questions
  → proportions and probabilities instead of convergence and (almost)-sure

\[
\forall N \max_\sigma P_\sigma (A^N \models \Diamond \text{ at least } 80\% \text{ of MDPs in } F) \geq 0.7
\]

Controlling a population of NFA – Nathalie Bertrand
Discrete approximation of probabilistic automata

\[ \text{Gap: optimal reachability probability not continuous when } N \to \infty \]

In the PA, the maximum probability to reach \( F \) is...
Probabilistic population

Discrete approximation of probabilistic automata

In the PA, the maximum probability to reach $F$ is $\frac{5}{2}$.

Good news? hope for alternative more tractable semantics for PA
Probabilistic population

Discrete approximation of probabilistic automata

Controlling a population of NFA – Nathalie Bertrand
Probabilistic population

Discrete approximation of probabilistic automata

In the PA, the maximum probability to reach $F$ is $\frac{5}{2}$.

Good news? hope for alternative more tractable semantics for PA.

Gap: optimal reachability probability not continuous when $N \to \infty$.
Probabilistic population

Discrete approximation of probabilistic automata

Gap: optimal reachability probability not continuous when $N \to \infty$
Probabilistic population

Discrete approximation of probabilistic automata

Gap: optimal reachability probability not continuous when $N \to \infty$

- $\forall N, \exists \sigma, P_\sigma(\Diamond F^N) = 1$.
- In the PA, the maximum probability to reach $F$ is $0.5$. 

Controlling a population of NFA – Nathalie Bertrand
Probabilistic population

Discrete approximation of probabilistic automata

Gap: optimal reachability probability not continuous when \( N \to \infty \)

Good news? hope for alternative more tractable semantics for PA
ευχάριστώ!