



HAL
open science

Optimally Controlling a Random Population

Hugo Gimbert, Corto Mascle, Patrick Totzke

► **To cite this version:**

Hugo Gimbert, Corto Mascle, Patrick Totzke. Optimally Controlling a Random Population. 2024.
hal-04797397

HAL Id: hal-04797397

<https://hal.science/hal-04797397v1>

Preprint submitted on 22 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open licence - etalab

Optimally Controlling a Random Population

Hugo Gimbert ✉

CNRS, LaBRI, Université de Bordeaux, France

Corto Mascle ✉

LaBRI, Université de Bordeaux, France

Patrick Totzke ✉

School of Electrical Engineering, Electronics and Computer Science, University of Liverpool, UK

Abstract

The random population control decision problem asks for the existence of a controller capable of gathering almost-surely a whole population of identical finite-state agents simultaneously in a final state. The controller must be able to satisfy this requirement however large the population, provided that it is finite. The problem was previously known to be decidable and EXPTIME-hard. This paper tackles the exact complexity: the problem is EXPTIME-complete.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases Dummy keyword

Digital Object Identifier 10.4230/LIPIcs...

Introduction

The population control problem was introduced in [1] in its non-deterministic version, and furthermore studied in [2] in its stochastic version, the RANDOM POPULATION CONTROL PROBLEM. The latter is a control problem for a finite population of agents, every agent is a copy of the same finite-state Markov decision process. This finite population is controlled by a central decision-maker, who can select one central action at every discrete time step, and this action applies simultaneously to every member of the population: every MDP is updated to a new state, according to the global action, the local state of the MPD, and the transition probabilities. The goal of the decision maker is to put all the MDPs simultaneously in a final state.

Some intuition can be conveyed by using a shepherd-like metaphor. One can depict the finite population as a herd of sheep, and the central decision maker is the shepherd. The shepherd can send his dog to bark here or there, and all the sheeps will perceive this noise, and react accordingly, in the typical randomized way the sheep react to this kind of stimulus. Part of the shepherd job consists in gathering the stray sheep back in the herd, which is doable in finite time as long as the population is finite, but would become too challenging if the herd was infinite, a case we do not consider in the current paper.

The RANDOM POPULATION CONTROL PROBLEM was shown to be decidable [2], and EXPTIME-hard [5]. The former paper provides a decision procedure whose termination relies on well-quasi-order arguments, and lead to non-elementary complexity upper-bounds. The latter paper shows that the problem is at least as hard to solve in its randomized version that it is in its deterministic version. The current paper shows that this lower-complexity bound is optimal: the problem is EXPTIME-complete (Theorem 39).

Organisation of the paper.

Section 1 introduces the model and the key ingredients for the decision procedure of [2]. Section 2 shows that the shepherd can win by counting up to $|S|$, the number of states in the MDP, independently of the size of the population. This relies on two playing modes:



© "CC-BY"; <http://creativecommons.org/licenses/by/3.0/>;
licensed under Creative Commons License CC-BY 4.0

Leibniz International Proceedings in Informatics

LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

XX:2 Optimally Controlling a Random Population

funnelling obedient sheeps (Section 2.3) and gathering stray sheeps (Section 2.4), between which the shepherd can alternate thanks to a key conceptual ingredient: the Isolation Lemma (Section 2.2). This leads to an EXPSPACE algorithm (Theorem 20). Finally, in Section 3, we improve the complexity down to EXPTIME, using data structures introduced in [1].

1 The Random population control problem

A controller tries to control a large, yet finite, set of tokens $T_N = 1 \dots N$. Each token is a copy of the same Markov Decision Process (MDP) $\mathcal{M} = (S, A, \delta : S \times A \rightarrow \Delta(S))$. One initial state $i \in S$. Final states $F \subseteq S$.

The N -population MDP is the synchronized product of N copies of the game. The initial state is $(i, i, \dots, i) \in S^N$ and the set of final states is F^N . Each copy represent a token.

RANDOM POPULATION CONTROL PROBLEM: does it hold that for every integer N , there is a strategy in the N -population MDP to enforce the population to almost-surely reach F^N when starting in the initial configuration i^N ?

For a fixed N the RANDOM POPULATION CONTROL PROBLEM is clearly decidable, since it boils down to a question about a single MDP. The parametrized version where N is arbitrarily large is co-recursively enumerable: it suffices to find one value of N where the answer is negative. Decidability was established in [2], in the rest of the section we provide the essential ingredients of the decidability proof.

1.1 Commits, arenas and strategies

Fix an automaton with a set of states S and a set of final states F .

► **Definition 1** (Configurations, commits and pathes). *Fix a set of tokens T . A configuration is a vector*

$$w \in S^T$$

which describes the position of every token. A commit is a pair

$$(v, a) \in S^T \times A .$$

If from a commit v there is positive probability to reach some position $w \in S^T$ when playing a then we say w is a successor of (v, a) and denote it

$$(v, a) \rightarrow w .$$

We extend the successor relation between commits: $(v, a) \rightarrow (w, b)$ iff $(v, a) \rightarrow w$. This naturally defines a notion of path between two commits, denoted $(v, a) \rightarrow^ (w, b)$.*

Sometimes, we do not care of the exact state of every token, we only need to pay attention to the number of tokens on every state.

► **Definition 2** (Anonymous configurations and commits). *An anonymous configuration is a vector in \mathbb{N}^S . An anonymous commit is a pair $(w, a) \in \mathbb{N}^S \times A$. The anonymous configuration associated with a configuration $w \in S^T$ is the vector:*

$$(|\{t \in T \mid w[t] = s\}|)_{s \in S} \in \mathbb{N}^S .$$

When clear from the context, we skip the "anonymous" qualificative and simply say a commit and a configuration.

► **Definition 3 (Arena).** *An arena is a set W of anonymous configurations and commits such that W is downward-closed and it satisfies two properties:*

- *No dead-end: every anonymous configuration of W has at least one successor commit in W .*
- *Stability under chance moves: every anonymous commit of W has all its successors configurations in W .*

We say that a (non-anonymous) configuration or commit $w \in S^T \cup S^T \times A$ belongs to W if the corresponding anonymous configuration or commit does.

Sometimes, an arena W is equipped with a target set $W_F \subseteq W$. In that case we relax the definition and we allow configurations in W_F to be dead-ends.

► **Definition 4.** *A strategy in W is a mapping which associates with every configuration $v \in W$ a probability distribution on the set of actions a such that $(v, a) \in W$.*

All strategies we consider are positional, which is enough for reachability games.

► **Definition 5.** *A strategy is state-based if the action chosen only depends on the number of tokens in every state, not the identity of those tokens.*

► **Definition 6 (Safe random walk).** *The strategy which consists in selecting randomly any action which guarantees for sure to stay in the set of winning configuration is called the "safe random walk".*

► **Lemma 7.** *The safe random walk is an almost-surely winning strategy. It is state-based.*

The goal is to almost-surely put all tokens in F .

► **Definition 8 (Almost-sure winning configurations and commits).** *A configuration is final if all tokens are in F . A configuration w_0 is almost-surely winning if there is a strategy in W which guarantees almost-surely that any play starting in w_0 eventually reaches a final configuration. A commit is almost-surely winning if all its successors are.*

1.2 A fix-point algorithm

The following characterisation is classical.

► **Lemma 9.** *The set of almost-surely winning configurations and commits is the largest arena W such that from every configuration in W there is a path within W to a final configuration.*

The *upper closure* of an (anonymous) commit (v, a) is the set of commits (w, a) such that $w \geq v$, pointwise on all coordinates.

XX:4 Optimally Controlling a Random Population

Fix-point algorithm computing the set of almost-surely winning commits

- Start with W equal to the whole set of commits.
- Repeat
 - Look for a commit (v, a) in W such that either of the following conditions holds:
 - * there is a chance to escape W from (v, a) , i.e. a successor w such that there is no commit (w, b) in W ; or
 - * there is no path in W from (v, a) to a final configuration.
 - If there is such a commit (v, a) , then remove from W the upper-closure of (v, a) and proceed to the next iteration. Otherwise terminate the computation.

The fix-point algorithm manipulates infinite sets of commits, and may, at first sight, not terminate. However, it can be implemented in finite space and time. We make use of the ordered monoid

$$\bar{\mathbb{N}} = \{0 < 1 < 2 < 3 < \dots < \omega\} .$$

► **Definition 10** (Symbolic configurations and commits). *A symbolic configuration is an element of $\bar{\mathbb{N}}^S$. Given $N \in \mathbb{N}$ and a symbolic configuration $w \in \bar{\mathbb{N}}^S$, denote $w[N]$ the configuration where all ω 's are replaced by N . A symbolic commit is an element of $\bar{\mathbb{N}}^S \times A$.*

The ideal generated by a symbolic configuration \bar{w} is denoted $\bar{w}\downarrow$, it is the set of anonymous configurations whose coordinates are below those of \bar{w} :

$$\bar{w}\downarrow = \{w \in \bar{\mathbb{N}}^S \mid \forall s \in S, w(s) \leq \bar{w}(s)\} .$$

The notation is extended to set of commits, by taking the union of the ideals.

$$\{(\bar{w}_i, a_i), i \in I\}\downarrow = \bigcup_{i \in I} \bar{w}_i\downarrow .$$

When $W \subseteq \bar{\mathbb{N}}^S$ is a set of anonymous configurations and $\bar{w} \in \bar{\mathbb{N}}^S$ a symbolic configuration, we say that \bar{w} belongs to W , denoted $\bar{w} \in W$, whenever

$$\bar{w}\downarrow \subseteq W .$$

It is well-known [reference].

► **Lemma 11** (Folklore, see [6]). *Any downward closed set of commits W is equal to a finite union of ideals, i.e. there exists a finite collection $\{(\bar{w}_i, a_i), i \in 0 \dots n\}$ of symbolic commits such that*

$$W = \{(\bar{w}_i, a_i), i \in 0 \dots n\}\downarrow .$$

The symbolic initial state is the symbolic configuration

$$\bar{i} \in \bar{\mathbb{N}}^S$$

where all components are 0 except for i . The symbolic final state is the symbolic configuration

$$\bar{F} \in \bar{\mathbb{N}}^S$$

where all components are 0 except for those in F . The RANDOM POPULATION CONTROL PROBLEM can be rephrased as follows

RANDOM POPULATION CONTROL PROBLEM: does there exist a strategy which almost-surely guarantees reaching \bar{F} when starting the play in \bar{i} ?

As a consequence

► **Lemma 12.** *The fix-point algorithm maintains as an invariant that W is downward-closed and can be effectively represented as the downward-closure of a finite set of symbolic commits. The algorithm terminates.*

The algorithm makes use of two tests in the loop. The first one is quite easy to check.

► **Lemma 13.** *Let W be an arena finitely presented as $W = \{(\bar{w}_i, a_i), i \in 0 \dots n\} \downarrow$. Whether there exists a commit $(v, a) \in W$ from which there is a chance to escape W is decidable.*

Proof. We enumerate all commits (\bar{w}_i, a_i) , one by one. Say a flow graph from \bar{w} to \bar{w}' is maximally spreading if for every state s with $\bar{w}(s) = \omega$, all edges from s are labelled by ω . We generate all possible maximally spreading flow graphs from \bar{w}_i to some symbolic configuration \bar{w}'_i . There are finitely many. We check that for every such \bar{w}'_i , there exists an action b such that $(\bar{w}'_i, b) \in W$. The answer is positive iff all checks are successful. ◀

1.3 The path problem

The second test can be expressed as the following decision problem.

► **Definition 14 (The path problem).** *Let W be an arena finitely presented as $W = \{(\bar{w}_i, a_i), i \in 0 \dots n\} \downarrow$. Let \bar{w} an symbolic configuration. The path decision problem asks whether, from every configuration in $\bar{w} \downarrow$, there is a path in the arena W to reach \bar{F} .*

Colcombet et al. could prove that the path problem is decidable.

► **Lemma 15.** [2] *There is an algorithmic reduction from the path problem to the emptiness problem for distance automata. In case the answer is negative, the decision procedure outputs an integer N such that there is no path from $\bar{w}[N]$ to a final configuration in W .*

This leads to a procedure with non elementary complexity, since the best upper-bound to our knowledge on the number of calls to the path problem is non-elementary.

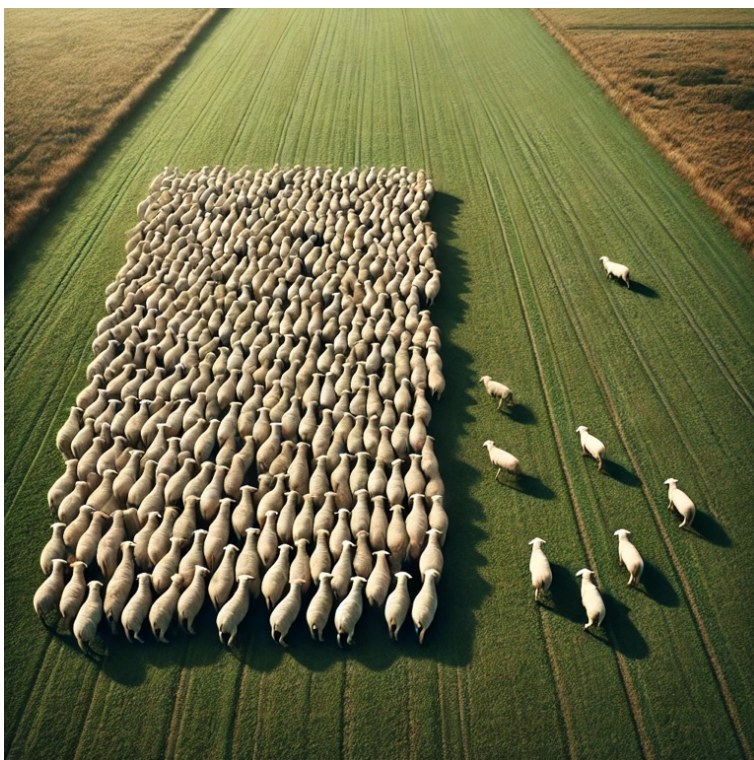
2 Counting up to $|S|$ sheep is enough for the shepherd

The non-elementary complexity in the algorithm of [2] arises from the fact that their algorithm requires to describe a set of configurations defined by potentially very large integers. It was conjectured [4] that computing an almost-surely winning strategy does not require such fine description of the winning zone, and that tracking a small number of tokens, plus a set of unbounded states, is enough to solve population MDPs.

► **Definition 16** ($\{0, \dots, m, \omega\}$ -configurations). *A $\{0, \dots, m, \omega\}$ -configuration is a symbolic configuration whose finite coordinates are $\leq m$.*

Sometimes we will illustrate concepts by associating a symbolic configuration into a shepherd-style metaphor: the tokens are sheep which splits between the *herd*, which occupies the ω -coordinates and the *stray sheep*, which occupies the finite coordinates, as illustrated on Figure 1.

XX:6 Optimally Controlling a Random Population



■ **Figure 1** The herd and the stray sheep.

Till the end of the section, denote W the arena of almost-surely winning configuration and commits, and for every $k \in \mathbb{N}$, denote

$$W_{0\dots k,\omega}$$

the set of $\{0\dots, k, \omega\}$ -configurations which belong to W , i.e. those symbolic configurations whose finite coordinates are $\leq k$ and which generate an ideal fully contained in W .

The configurations in $W_{0,\omega}$ are called purely symbolic winning configurations, which always contains \bar{F} . Moreover $W_{0,\omega}$ contains \bar{i} iff the answer to the RANDOM POPULATION CONTROL PROBLEM is positive.

Denote

$$m = |S| .$$

The following theorem shows that Corto's conjecture holds.

► **Theorem 17** (Almost-surely winning with a few stray sheep). *Let W be the arena of almost-surely winning configurations and commits. There exists a subarena Y of W such that:*

- Y contains $W_{0,\omega\downarrow}$; and
- Y is contained in $W_{0,\dots,m,\omega\downarrow}$; and
- the safe random walk in Y is almost-surely winning.

To show this theorem, we make use of almost-surely winning strategies which alternate between two playing modes.

2.1 A controller with two modes

The first mode is called “*Funneling obedient sheep*”. In this mode the tokens follow a predetermined path, called a *funnel path*, which leads to a final set. On the funnel path, only a few sheep at a time are allowed to leave the herd, and none of them ever meet: they are expected to stay in different states, and thus there are at most $|S|$ of them at any moment. The controller selects actions to stay on the funnel path, and expect the sheep to follow the predetermined path, which might occur with positive probability, but in general this is not for sure. In case the sheep leave the funnel path, the controller enters the second playing mode. This is illustrated on Figure 2a.



(a) Funneling obedient sheep



(b) Gathering the herd

■ **Figure 2** The two modes of the controller

The second mode is called “*Gathering the herd*”, it is entered when some non-obedient sheep have left the funnel path. At this moment the sheep that do not belong to the herd are considered as *stray sheep*. The controller objective is to gather all sheep back together into a herd, including the stray sheep. There is an extra constraint: at every moment, the controller is expected to control the stray sheep so that there are at most m of them (at most $|S|$ in one state, at most $|S| - 1$ in another state, and so on).

► **Lemma 18** (Funneling obedient sheep). *From every configuration in $W_{0,\omega}$ there exists a path which stays within $W_{0,1,\omega}$ and ends up in \bar{F} . Such a play is called a funnel path.*

The proof of lemma 18 is given in section 2.3, where a more general statement is proved (Lemma 28).

A funnel path might be very long, thus very unlucky to occur on first try. It might be that, by chance, this play occurs as planned, in which case the game is over and controller wins. Whenever the chance exits the funnel path, then the new configuration belongs to $W_{0,\dots,|S|}$ the strategy switches to a recovery strategy in order to reach back $W_{0,\omega}$.

► **Lemma 19** (Gathering stray sheep back in the herd). *There exists a subarena Z of W such that:*

- Z contains $W_{0,\omega\downarrow}$; and
- Z is contained in $W_{0,\dots,m,\omega\downarrow}$; and

XX:8 Optimally Controlling a Random Population

- *the safe random walk in Z almost-surely guarantees reaching $W_{0,\omega}$. Such a strategy is called a gathering strategy.*

The proof of lemma 19 is given in section ??.

Combining funnel pathes and gathering strategies is the key to keep a low profile.

Proof of Theorem 17. Let Y be the set of all configurations visited by all possible funnel paths, plus Z . From any $W_{0,\omega}$ configuration, when playing the safe random walk in W' , there is a non-zero chance to follow a funnel path and finitely reach \bar{F} . In case the lucky path is exited, there is probability 1 to reach $W_{0,\omega}$ again. Since there are finitely many possible configurations reachable from a given initial configuration, the non-zero chances are uniformly bounded from below. Thus almost-surely \bar{F} is eventually reached. ◀

► **Theorem 20.** *The random population control problem can be decided in EXPSPACE.*

Proof. According to Theorem 17, the answer to the RANDOM POPULATION CONTROL PROBLEM is positive iff there exists a downward-closed set Y of configurations and commits generated by a finite family $(w_i, a_i)_{i \in I}$ such that,

- every configuration $w_i, i \in I$ has finite coordinates $\leq m$; and
- Y is an arena: it is closed by chance moves and allows at least one action from any configuration; and
- Y contains both \bar{i} and \bar{F} ; and
- from any $\{0, \omega\}$ -configuration in Y there is a funnel path.

There are at most $(m+2)^{|S|}(1+|A|)$ different possible configurations and commits thus we can assume $|I| \leq (m+2)^{|S|}(1+|A|)$. The second condition is easy to check in time polynomial in $|I|$ and m , as proved in Lemma 13.

The existence of a winning path in Y from any $\{0, \omega\}$ -configuration in Y , can be performed using a distance automaton of size polynomial in $|I|$, which can be performed in space polynomial in $|I|$ (see [2, Theorem 5.1] and [3, Theorem 2.2] for details). ◀

2.2 The isolation lemma

The isolation lemma is one of the pillars of the proof of Theorem 17, informally it says that when a group of stray sheep leaves the herd, the stray sheep can be brought back in the herd without ever meeting any other stray sheep outside their group.

Assume that we play in some arena W , from some initial position w_0 . On some states we have a large number of tokens, that altogether form *the herd*, and on top of that there are also have a few other tokens called the *stray sheep*. In formal statement, the set of tokens T_ω typically denotes the herd while T_1 typically denotes the stray sheep.

In a configuration, we say that a set of states is an ω -base if an arbitrary amount of extra tokens could be placed on these states without exiting the arena W . This is formally defined as follows.

► **Definition 21** (Duplicating tokens). *Let W an arena, T a set of tokens and w a configuration in W . Let $T_\omega \subseteq T$ be a subset of tokens. We denote*

$$w[T_\omega * N]$$

*the configuration obtained from w by replacing every token $t \in T_\omega$ with N copies t_1, t_2, \dots, t_N . By extension, if S_ω is a set of states, and T_ω is exactly the set of tokens occupying those states in w , the configuration $w[T_\omega * N]$ is also denoted $w[S_\omega * N]$.*

Of course, we want to duplicate tokens without exiting our arena, which is captured by the notion of ω -base.

► **Definition 22** (ω -base and finite base). *Fix an arena W and a configuration $w \in W$. A set of tokens T_ω is an ω -base of w in W if*

$$\forall N \in \mathbb{N}, w[T_\omega * N] \in W .$$

By extension, a set of states T_ω is an ω -base of w in W if the set of tokens occupying those states in w is. Dually, a set of tokens T_1 is a finite base of w in W iff its complement is an ω -base of w in W .

Initially the set of states occupied by the herd is unbounded, but later on there might a few sheep escaping the herd and occupying states that are bounded.

In a configuration, we say that a group of stray sheep is *brought back in the herd* whenever the the set of states occupied by the herd and those sheep is unbounded. It might be the case that some states are simultaneously occupied by sheep from the herd and some of the stray sheep, but this is not required in the definition.

► **Definition 23** (Bringing back stray sheep in the herd). *Fix T_1 a set of tokens. In a configuration w , we say that some stray sheep of T_1 are brought back in the herd if there exists a strict subset of T_1 which is a finite base of w .*

As already noticed, this definition does *not* require that stray sheep actually meet some sheep of the herd, i.e. the previous definition does *not* impose $\exists t \in T_1, t' \in T_\omega, w[t] = w[t']$.

The main result of this section is the isolation lemma. Assume that from a configuration where the herd is unbounded, there exists a strategy which almost-surely guarantees to eventually bring back some stray sheep in the herd. Then the same holds with the extra requirement that until some stray sheep are brought back in the herd, all stray sheep stay *isolated* from the herd, in the sense where they never share a common state.

► **Definition 24** (Meetings and isolations). *We say that two tokens $t_1, t_2 \in T$ meet in a configuration $w \in S^T$ if they are placed on the same state i.e. if $w[t_1] = w[t_2]$. A set of tokens T_1 is isolated in a configuration w if it does not meet any token outside T_1 .*

► **Lemma 25** (Isolation Lemma). *Fix an arena W , a configuration w_0 of W and a finite base T_1 of w_0 in W . Denote F the set of configurations in which some stray sheep of T_1 are brought back in the herd. Assume F is almost-surely reachable from everywhere in W . Then there is a strategy σ in W which, when starting from w_0 :*

- *guarantees almost-surely to reach F ; and*
- *guarantees for sure that the stray sheep from T_1 stay isolated until F is reached.*

Proof. It is enough to show that from every configuration $w_0 \in W$, there is a path to F on which the stray sheep stay isolated until reaching F , and in every successor configuration of every commit of the path, the stray sheep are isolated as well. The strategy consists in trying following this path and in case of exit, try again a new path. That occurs in a finite set of reachable configurations, hence the strategy will eventually succeeds in following the path.

Denote C the set of tokens in w_0 which are not in T_1 , and S_ω the states occupied in w_0 by tokens in C . By hypothesis, S_ω is an ω -base of w_0 in W . Let M be a large constant, and $z_M = w_0[S_\omega * (M + 1)]$ the configuration obtained from w_0 by duplicating every token of S_ω into $M + 1$ copies, so that we get a larger set of tokens on S_ω , denoted C_M , called *the herd tokens* of z_M .

XX:10 Optimally Controlling a Random Population

We say a token t is *blowable* in a configuration if $\{t\}$ is an ω -base of this configuration. Denote F' the set of configurations where one of the stray sheep in T_1 is blowable. Then $F \subseteq F'$.

Let σ be the random walk in W , which guarantees to reach F from everywhere in W , a fortiori to reach F' .

Let t be one of the stray sheep of T_1 . When t meets one or more herd tokens of z_M , we call this event a *meeting*. Denote V the random variable counting the number of different herd tokens met by t before reaching F' .

We show that the expected value of V is finite and upper-bounded independently of M . Let B be the largest constant used to define W as a finite union of ideals. We argue that the probability that V is above $B \cdot |S| \cdot \ell$ decreases exponentially with ℓ . First, we show that, every time t meets some herd tokens of z_M , there is probability at most $1 - 1/B$ that one of the tokens met by t becomes blowable before t does. Until F' is reached, no more than $B - 1$ other tokens can share the same state than t . The random walk σ is defined with respect to anonymous configurations, like W is. Thus σ does not make a difference between t and the herd tokens meeting t , thus after a meeting, all of them have the same probability measure on their possible future trajectories. Since σ guarantees to almost-surely reach F someday, all of the herd tokens met by t will almost-surely become blowable someday, thus t as well. By symmetry, all of them have the same probability to be the first to become blowable, thus the probability that none of the $\leq B - 1$ herd tokens becomes blowable until t does is at least $1/B$. Look at those finite pathes where t meets at least $B|S|$ different herd tokens, and still t is not blowable. By definition of B , among those $B|S|$ herd tokens, at least 1 has became blowable. Thus, at every moment the probability that t meets $B|S|$ different herd tokens in the future is at most $1 - 1/B$, hence

$$\mathbb{P}_{\sigma, z_M}(V \geq B \cdot |S| \cdot \ell) \leq (1 - 1/B)^\ell . \quad (1)$$

So finally

$$\mathbb{E}_{\sigma, z_M}[V] \leq \sum_{\ell} B \cdot |S| \cdot (\ell + 1) \cdot (1 - 1/B)^\ell = K(B, |S|) . \quad (2)$$

Remark that the right handside of (2) is an upper-bound independent of M .

We deduce that for every herd token in Z_M , the probability that this herd token in particular meets t before t becomes blowable converges to 0 when M grows. Let s be one of the states on which the herd tokens are placed in w_0 . There are at least M herd tokens of C_M on s . Since σ_M is state based, all of them have the same probability $p_{t,s,M}$ to meet t before reaching F' . According to (2),

$$p_{t,s,M} \leq K(B, |S|)/M \rightarrow_M 0 , \quad (3)$$

which converges to 0 when M grows.

Recall that C denotes the set of tokens in w_0 which are not in T_1 and C_M their M copies in z_M . The strategy σ_M from z_M can be projected onto a strategy from w_0 by simulating the M copies of every herd token in $C_M \setminus C$. The resulting strategy is randomized. According to (3), the probability that a token in C meets a token in T_1 before F' occurs can be made arbitrarily small, by augmenting the number M of tokens being simulated, while guaranteeing at the same that F' occurs almost-surely. Since the set of configurations reachable from w_0 is finite, and the corresponding condition is a reachability condition under safety constraint, the probability can be turned to 0, i.e. F' can be reached almost-surely, while keeping the tokens in T_1 isolated.

We have not quite proved yet our goal: when starting the play in w_0 we can keep the tokens in T_1 isolated and reach almost-surely F' , but what about reaching F ? The inclusion $F \subseteq F'$ might be strict because when one of the stray sheep t in T_1 become blowable, it might be that other sheep from the herd prevent a strict subset of $T_1 \setminus \{t\}$ to be a finite base. However by definition of B , by deleting at most $B|S|$ sheep from the herd, we can allow $T_1 \setminus \{t\}$ to be a finite base. Call those sheep to be deleted the *blockers*. Then the expected number of blockers is $\leq B|S|$. This is true not only when the play starts from w_0 but also when it starts from z_M , because T_1 is also a finite base of z_M . But then the probability that a given sheep from the herd becomes a blocker is less than $B|S|/M$. Using the same argument than in the previous paragraph, it prove that from w_0 we can make the probability that a token in C is a blocker as close to 0 as we want, while ensuring reaching F' and keeping the tokens in T_1 isolated. Since the set of configurations reachable from w_0 in W is a finite MDP, we can make this probability 0, in which case reaching F' is equivalent to reaching F . ◀

2.3 Funelling obedient sheep

This is the first part of the proof of Theorem 17. Informally, in this section is exposed the way to guide a herd of *obedient* sheep to the objective.

► **Context.** *In the whole subsection 2.3:*

- we fix an arena W and a set of final configurations F such that the safe random walk in W almost-surely reaches F ; and
- we fix a set T_1 of tokens, called *stray sheep*; and
- we assume that tokens in T_1 are isolated in every configuration of W , except maybe in F ; and
- we assume that F contains all configurations in which some stray sheep of T_1 are brought back to the herd.

In order to keep track of the position of stray sheep, we extend abstract configurations with some mapping S^{T_1} .

► **Definition 26** (Extended abstract configurations). *The set of extended abstract configurations is*

$$\overline{\mathbb{N}}^S \times S^{T_1} .$$

The ideal generated by an extended abstract configuration (\overline{w}, w_1) is denoted

$$(\overline{w}, w_1) \downarrow ,$$

this is the set of (non-anonymous) configurations $w \in S^T$ for some set of tokens T satisfying two constraints¹. First, $T_1 \subseteq T$ and the tokens of T_1 occupy the positions specified by w_1 , i.e.

$$\forall t \in T_1, w(t) = w_1(t) .$$

Second, the capacity constraint \overline{w} is satisfied:

$$\forall s \in S, |w^{-1}(s)| \leq \overline{w}(s) ,$$

¹ Technically, in order to avoid annoying set-theory related subtleties, in the background we fix an infinite countable set of tokens T_∞ from which T is necessarily a subset.

XX:12 Optimally Controlling a Random Population

We extend the notation $W_{0\dots 1,\omega}$ used in the funneling Lemma 18 in order to cope with the set T_1 of stray sheep.

► **Definition 27.** For every $k \in \mathbb{N}$, denote

$$W_{0\dots k,\omega,T_1}$$

the set of extended abstract configurations (\bar{w}, w_1) such that:

- the whole ideal $(\bar{w}, w_1)\downarrow$ is included in W ; and
- the finite coordinates of \bar{w} are at most k .

We prove a slightly more general version of Lemma 18.

► **Lemma 28** (Funelling the herd, except for a few loners). From every configuration in W_{0,ω,T_1} there exists a path which stays within $W_{0,1,\omega,T_1}$ and ends up in F .

Proof. This is a direct corollary of:

► **Lemma 29** (Reducing capacities constraints down to 1). Let G be a directed acyclic graph with d sources. Assume there are capacity constraints on vertices with values in $0 \dots, K, \omega$, where ω means no bound. Assume that we can transmit through G a discrete flow of $d \cdot N$ tokens, with N tokens initially placed on each source. Consider the flow graph G' obtained from G by reducing all finite capacity constraints down to 1 (and vertices with unbounded capacities in ω stay unbounded). Then we can transmit through G' a discrete flow with $N/(Kd)$ tokens on every source.

Proof. This is a direct corollary of the characterisation of optimal solutions of the Maximum Concurrent Flow Problem [7, Lemma 2.2]. ◀

◀

2.4 Gathering stray sheep back in the herd

In this section we combine the isolation lemma (Lemma 25) and the funneling lemma (Lemma 28) in order to almost-surely gather the stray sheep back in the herd, while keeping the total number of stray sheep below a certain level.

► **Definition 30** (tracking of a strategy). A tracking ϕ of a strategy σ is a mapping

$$\phi : [\sigma] \rightarrow 2^T$$

which associates with every play π a set of tokens $T_\pi \subseteq T$ of the last configuration of π , called the trackers, with the following properties. Let π' a one-move extension of π consistent with the strategy. Then either of the following three cases holds:

- no change: $T_\pi = T_{\pi'}$; or
- tracker creation: $T_\pi \subsetneq T_{\pi'}$; or
- tracker deletion: $T_{\pi'} \subsetneq T_\pi$.

► **Definition 31** (properties of trackings). A tracking is state-based if for every play π ending in a configuration w , in a given state s , either all tokens or none of the tokens placed on the state s in w are trackers.

A tracking is a correct ω -abstraction in a play π if the set of trackers T_π is a finite base of the last configuration of π in W .

With a tracking which is both state-based and a correct ω -abstraction is naturally associated a mapping

$$\phi : [\sigma] \rightarrow \overline{\mathbb{N}}^S,$$

which associates with every finite play the symbolic configuration where trackers are counted and other tokens are abstracted away by ω . The correct ω -abstraction hypothesis ensures that the whole ideal of the symbolic configuration is contained in W .

The following lemma provides the induction step for proving the existence of recovery strategies.

► **Lemma 32 (Induction step).** *Let W be an arena and an initial configuration w_0 . Let T_1 be a finite base of w_0 , called the stray sheep, and denote S_ω the corresponding ω -base (i.e. the states occupied in w_0 by the other tokens). Let I be the set of configurations obtained from w_0 by adding arbitrarily many tokens on S_ω . Let F be the set of configurations where some stray sheep from T_1 are brought back to the herd (i.e. a strict subset of T_1 is a finite base). Assume:*

- *in every configuration of $W \setminus F$, the stray sheep are isolated; and*
- *in every configuration of $W \setminus F$, the stray sheep occupy at least d different states; and*
- *F is almost-surely reachable from everywhere in W .*

Then there exists a strategy from w_0 with a tracking ϕ such that F is almost-surely reached when playing the strategy σ from w_0 , and in every finite play occurring on the way to the first visit to F :

- *ϕ is a state-based correct ω -abstraction; and*
- *ϕ tracks at most $|T_1| + (1 + 2 + \dots + (|S| - d))$ tokens; and*
- *unless F is reached, ϕ tracks at least all tokens of T_1 , possibly more.*

Proof of Lemma 32. Without loss of generality, we assume that F is almost-surely reachable from any configuration or commit in W , if this is not the case, we restrict W to configuration and commits in W having this property.

Set $k = |S| - d$. The proof is done by induction on (k, ℓ) where $\ell = |T_1|$.

Assume $k = 0$. Then all tokens of w_0 are stray sheep and the tracking ϕ which constantly tracks T_1 , i.e. all tokens, satisfies the condition.

Let $\ell \in \mathbb{N}$. Assume the property is proved for all pairs of indices (k', ℓ') when $k' \leq k$ or $k' = k + 1 \wedge \ell' < \ell$. We prove the inductive step: the property holds for $(k + 1, \ell)$.

In case $\ell = 0$ then T_1 is empty hence F is empty, and we get a contradiction with the hypothesis F is almost-surely reachable from I .

Assume $\ell > 0$.

We build σ and ϕ as follows. We apply Lemma 28 to obtain a funnel path π from w_0 to F stays within $W_{0,1,\omega,T_1}$, i.e. apart from stray sheep of T_1 there is at most one fresh stray sheep by state not occupied by T_1 . Since at least d states are occupied by the stray sheep, which are isolated from other tokens, then the funnel path uses at most $|S| - d$ fresh stray sheep.

This funnel path π is the spine of the strategy, on which ϕ is tracking T_1 plus the at most $\leq |S| - d$ tokens used for pathfinding. It remains to define the rest of the strategy and the corresponding tracking: what does happen when sheep are no more obedient and exit the funnel path?

Consider a configuration w_1 immediately reachable when exiting the funnel path π .

XX:14 Optimally Controlling a Random Population

In case $w_1 \in F$ the strategy is over on this branch, and for the sake of completeness of the definition of ϕ , $\phi(w_1)$ tracks the corresponding strict subset of T_1 : the stray sheep of T_1 which have not joined the herd yet, if any, and the empty set, if none.

Otherwise, in case $w_1 \notin F$, there exists an ω -base S'_ω of w_1 such that the tokens outside S'_ω are the union of T_1 plus at most $|S| - d$ tokens T''_1 , that were fresh stray sheep of the funnel path before the exit to w_1 .

The set of trackers on the path π, w_1 is $T_1 \cup T''_1$. To define σ we rely on an induction on T''_1 .

- The base case is when $T''_1 = \emptyset$. In other words, apart from T_1 , all other tokens belong to the ω -base S_ω . We are somehow back to the initial situation with w_0 and T_1 except this time it is from w_1 not w_0 . In that case we start a new funnel path and start again following it.
- If $T''_1 \neq \emptyset$ then let $T'_1 = T_1 \cup T''_1$. All hypotheses of the Isolation Lemma (Lemma 25) are satisfied for the initial configuration w_1 and the finite base T'_1 . Let F' be the set of configurations where there exists a strict subset of T'_1 which forms a finite base. In particular, since any set of tokens containing a finite base is itself a finite base,

$$F \subseteq F' .$$

Let I' be the set of configurations obtained from w_1 by duplicating tokens outside of T_1 . Then F' is almost-surely reachable from I' , because $I' \subseteq W$ and $F \subseteq F'$. So all conditions are met to apply Lemma 25.

We get a strategy that isolates $T''_1 \cup T_1$ from the rest of the tokens, until reaching F' . Until F' is reached, neither if F reached, thus by hypothesis the tokens in T_1 stay isolated. Hence the tokens in $T'_1 = T''_1 \cup T_1$ do occupy at least $d + 1$ states until F' is reached. We can apply the induction hypothesis and obtain a strategy σ'' and a state-based tracking ϕ'' which correctly ω -abstract the plays. The strategy σ simply follows σ'' until reaching F' . Once in F' , we use the inductive definition for a smaller T''_1 . In the meantime, the tracking ϕ'' tracks at most $|T''_1| + (1 + 2 + \dots + (|S| - d - 1)) \leq |T_1| + (|S| - d) + (1 + 2 + \dots + (|S| - d - 1))$ tokens. ◀

The existence of recovery strategies follows.

Proof of existence of recovery strategies (Lemma 19). Fix w_0 an initial configuration in $W_{0, \dots, |S|, \omega \downarrow}$. Denote $F' = W_{0, \omega \downarrow}$. From w_0 we can almost-surely reach F' , because

$$F \subseteq F' .$$

Since $w_0 \in W_{0, \dots, |S|, \omega \downarrow}$ there exists some symbolic configuration \bar{w}_0 such that

$$w_0 \leq \bar{w}_0 \subseteq W .$$

Hence, we can isolate in any configuration of $\bar{w}_0 \downarrow$ at most $|S|$ stray sheep T_1 and apply the isolation lemma to w_0 and T_1 in W . We obtain a strategy σ' to gather the stray sheep back in the herd, and restrict the arena to the set of configurations and commits reachable with σ' from $\bar{w}_0 \downarrow$, denoted W' . To that arena W' we apply the induction step (Lemma 32) with $d = 1$ and S_ω . We get a strategy to reach F , while tracking at most $1 + 2 + \dots + (|S| - 1)$ tokens on top of T_1 with a state-based, correct ω -abstraction. The existence of such a tracking implies that the play stays in $W_{0, \dots, m, \omega \downarrow}$. ◀

3 An EXPTIME algorithm

According to [5], we cannot hope better than EXPTIME.

► **Definition 33** (The SYMBOLIC POPULATION REACHABILITY PROBLEM). *The input of the problem is a tuple $(\mathcal{M}, \bar{w}_0, \bar{W}, \bar{F})$ where:*

- \mathcal{M} is an MDP with state space S ; and
- \bar{W} is a finite set of symbolic configurations and commits in \mathcal{M} i.e. a finite subset of $\bar{\mathbb{N}}^S \times A$,
- $\bar{w}_0 \in \bar{W}$ and $\bar{F} \subseteq \bar{W}$.

The SYMBOLIC POPULATION REACHABILITY PROBLEM asks whether from every configuration in $\bar{w}_0 \downarrow$ there exists a path to $\bar{F} \downarrow$ under the constraint that all commits belong to $\bar{W} \downarrow$.

► **Definition 34.** *The largest constant of an instance of the problem is the largest finite coordinate which appears in one of the finite sets $(\bar{w}_0, \bar{W}, \bar{F})$ (by convention 0 in the rare cases where there is no such finite coordinate).*

► **Lemma 35.** *Assume that there exists an algorithm which solves the SYMBOLIC POPULATION REACHABILITY PROBLEM in time $\leq 2^{\mathcal{O}(|S|)}$ whenever the largest constant is ≤ 1 . Then the RANDOM POPULATION CONTROL PROBLEM is solvable in EXPTIME.*

Proof. We first prove that there exists an algorithm which solves the SYMBOLIC POPULATION REACHABILITY PROBLEM in time $\leq 2^{\mathcal{O}(|S|)}$ whenever the largest constant is $\leq |S|^2$. Let (\mathcal{M}, I, W, F) an instance. We create a new instance $(\mathcal{M}', \bar{w}'_0, \bar{W}', \bar{F}')$ whose finite constants are at most 1, whose size is polynomial in the size of $(\mathcal{M}, \bar{w}_0, \bar{W}, \bar{F})$, and for which the answer to the SYMBOLIC POPULATION REACHABILITY PROBLEM is the same than for $(\mathcal{M}, \bar{w}_0, \bar{W}, \bar{F})$.

The definition of $(\mathcal{M}', \bar{w}'_0, \bar{W}', \bar{F}')$ is as follows. Let K be the sum of the finite coordinates in \bar{w}_0 (at most $|S|$). Let T_0 be a set of K tokens. The MDP \mathcal{M}' consists in $K + 1$ disjoint copies of \mathcal{M} . The initial configuration w'_0 is obtained as follows. On the K first copies of \mathcal{M} , we place exactly 1 token of T_0 on one of the state, so that the projection matches the finite coordinates of w_0 . On the $K + 1$ copy, we place the ω like in w_0 . The final configurations F' are those whose sum over the different copies is in F . From every anonymous configuration w' of \mathcal{M}' one can obtain a configuration $w = \phi(w')$ in \mathcal{M} by summing up the coordinates in the different copies. Fix an abstract configuration \bar{w} in \mathcal{M} we denote $\psi(\bar{w})$ all the abstract configurations \bar{w}' of \mathcal{M}' such that:

- the coordinates of \bar{w}' on the K first copies are either 0 or 1; and
- the coordinates of \bar{w}' on the $K + 1$ -copy are in $\{0, 1, \omega\}$; and
- $\phi(\bar{w}' \downarrow) \subseteq \bar{w} \downarrow$.

Remark that there are at most $6^{|S|}$ such configurations. We define $W' = \psi(W)$ and $F' = \psi(F)$. According to Lemma 28, the answer to the SYMBOLIC POPULATION REACHABILITY PROBLEM is the same for $(\mathcal{M}', \{w'_0\}, W', F')$ and $(\mathcal{M}, \{w_0\}, W, F)$.

Now we provide the algorithm to solve the RANDOM POPULATION CONTROL PROBLEM in EXPTIME. Set \bar{W}_0 to be the set of all configurations and commits with coordinates in $\{0, 1, \dots, |S|^2, \omega\}$.

Repeat until no configuration can be removed. Remove from \bar{W}_i :

- any abstract configuration which is a dead-end; and
- any abstract commit with a successor which is not abstracted by \bar{W}_i ; and
- any abstract configuration \bar{w} such that the answer to the SYMBOLIC POPULATION REACHABILITY PROBLEM $(\mathcal{M}, \bar{w}, \bar{W}_i, \bar{F})$ is negative.

XX:16 Optimally Controlling a Random Population

That finally converges to some \overline{W}_∞ . A symbolic configuration survives the elimination iff it is winning. Finally check that \bar{i} belongs to \overline{W}_∞ .

In case \bar{i} belongs to \overline{W}_∞ then clearly from \bar{i} there is a winning strategy: try again and again to follow an obedient path to \overline{F} in \overline{W}_∞ , until it works.

In case \bar{i} does not belong to \overline{W}_∞ then we conclude with the same arguments than in the proof of 20. ◀

So finally the EXPTIME upper bound reduces to the proof of decidability of the SYMBOLIC POPULATION REACHABILITY PROBLEM with largest constant 1, which is done in the next section.

4 Solving the Symbolic population reachability problem with largest constant 1 in EXPTIME

In this section we prove:

► **Lemma 36.** *The SYMBOLIC POPULATION REACHABILITY PROBLEM with largest constant 1 is solvable in EXPTIME.*

Fix an instance $(\mathcal{M}, \overline{w}_0, \overline{W}, \overline{F})$ of the SYMBOLIC POPULATION REACHABILITY PROBLEM with largest constant 1. We define an ω -regular parity game that Eve wins iff the answer to the SYMBOLIC POPULATION REACHABILITY PROBLEM is positive for this instance. The game has exponentially many states and the winning condition is recognizable by a deterministic parity automaton with exponentially many states and polynomially many priorities.

- Vertices of Eve V_E : symbolic $\{0, 1, \omega\}$ -configurations in \overline{W}
- Vertices of Adam V_A : symbolic $\{0, 1, \omega\}$ -flow induced by a commit in \overline{W} .

Initial vertex ω on i . Legal moves:

- from v Eve can select any flow f such that $\text{dom}(f) = v$; and
- from f Adam can select any symbolic $\{0, 1, \omega\}$ -configuration v' such that $v' \leq \text{im}(f)$.

Eve wins a play of the game in two cases:

- when the play reaches a configuration in \overline{F} ; or
- when the finite capacity condition is violated: there exists a subset of vertices with no incoming edge and infinitely many outgoing edges.

An extended discussion and definition of the finite capacity condition is given in [1, Section 3], see in particular [1, Lemma 3.5].

4.1 Equivalence of the game and the problem

This section is dedicated to the proof of the following lemma.

► **Lemma 37.** *Eve wins the game iff the answer to the SYMBOLIC POPULATION REACHABILITY PROBLEM is positive.*

Proof. One direction is easy: if Eve wins the abstract game then for every N she uses her strategy in the N -game to generate an over-approximation of the N -play, and uses Adam moves to adjust the over-approximation such that it is tight, i.e. at every moment the play is over-approximated by a minimal abstract configuration. That guarantees the finite capacity constraint is satisfied, because a violation of the finite capacity constraint in the

tight over-approximation would imply a violation in the actual N -play, which is impossible, cf [1, Lemma 3.5].

For the converse implication, use the fact that if the answer to the SYMBOLIC POPULATION REACHABILITY PROBLEM is positive then the corresponding distance automata build in [2] on the alphabet of symbolic $\{0, 1, \omega\}$ -flow is unbounded. According to [8, Theorem 12], there exists a \sharp -expression e which witness unboundedness: the cost of the sequence of words $(e[n])_n$ is unbounded, where $e[n]$ denotes the expression obtained by replacing \sharp with n .

Such \sharp -expression e is called a \sharp -path from \bar{w}_0 to \bar{W} , where \bar{w}_0 is an $\{0, 1, \omega\}$ -configuration generating the initial configurations ideal, and \bar{W} is the set of final configurations. Using the dictionary provided by [2], it means the maximal number k_N of tokens that can be moved from $\bar{w}_0[N]$ to $\bar{W}\downarrow$ through the capacity graph formed by the sequence of symbolic matrices $e[N]$ is unbounded when N grows large.

The proof is by induction on the structure of the \sharp -expression, with induction hypothesis:

- (H) if there is a \sharp -path e from \bar{w}_0 to \bar{W} , then Eve wins the game starting in \bar{w}_0 and with the corresponding winning condition \bar{W} .

There are four cases, corresponding to the two operators (concatenation and iteration), plus the base case. If e is a sharp expression,

- If the \sharp -path from \bar{w}_0 to \bar{W} is a single letter a , then a winning strategy for Eve consists in playing a .
- If the \sharp -path from \bar{w}_0 to \bar{W} is a product $e \cdot f$, for every integer N denote k_N the maximal number of tokens that can be moved along the capacity graph $e[N]f[N]$ and π_N the corresponding trajectory of the k_N tokens, and w'_N the configuration on π_N reached at the end of $e[N]$ (hence at the beginning of $f[N]$). Denote \bar{w} a limit in the topological closure of $\{w'_N \mid N\}$ and set $\bar{W}' = \{\bar{w}\}$. Then e is a witness of $\bar{w}_0 \rightarrow \{\bar{w}\}$ and f is a witness of $\bar{w} \rightarrow \bar{W}$. Apply the induction hypotheses to e and f to get two strategies for Eve, σ_e and σ_f , which win in the corresponding games. Then a winning strategy for Eve consists in playing σ_e and then as soon a configuration $\leq \bar{w}$ is reached, switch to σ_f , until \bar{W} is reached.
- If the \sharp -path from \bar{w}_0 to \bar{W} is an iteration e^\sharp , then the corresponding capacity graph e is idempotent, thus its image \bar{w} is contained in its domain \bar{w}_0 . If $e = e^\sharp$ (i.e. e is stable), then also e is a sharp path from \bar{w}_0 to \bar{W} , and we conclude by induction hypothesis. If $e \neq e^\sharp$ (i.e. e is unstable) Eve can play again and again the strategy σ_e associated with e , to go from \bar{w}_0 to $\bar{w} \leq \bar{w}_0$. Each time \bar{w} is reached, Eve reboots the strategy σ_e from scratch \bar{w}_0 . We need to add a special dummy letter which is used to force a move from Adam to handle the possible strict inequality $\bar{w} \leq \bar{w}_0$. Since the idempotent is unstable, Adam must either violate the finite capacity constraint, or eventually empty any connected component which is not minimam in the topological order (i.e. a BSCC). In both cases, Eve eventually wins. ◀

4.2 Solving the game in EXPTIME

► **Lemma 38.** *The game can be solved in EXPTIME.*

Proof. The winning condition can be recognised by a parity automaton of exponential size and bounded number of colors [1, Section 4.3] ◀

By combining Lemma 35 and Lemma 36 we obtain our main result.

► **Theorem 39.** *The RANDOM POPULATION CONTROL PROBLEM is EXPTIME-complete.*

Conclusion

Next step is solving population parity games, which we conjecture are also in EXPTIME.

References

- 1 Nathalie Bertrand, Miheer Dewaskar, Blaise Genest, Hugo Gimbert, and Adwait Amit Godbole. Controlling a population. *Logical Methods in Computer Science*, Volume 15, Issue 3, July 2019. URL: <https://lmcs.episciences.org/5647>, doi:10.23638/LMCS-15(3:6)2019.
- 2 Thomas Colcombet, Nathanaël Fijalkow, and Pierre Ohlmann. Controlling a random population. In Jean Goubault-Larrecq and Barbara König, editors, *Foundations of Software Science and Computation Structures - 23rd International Conference, FOSSACS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings*, volume 12077 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2020. doi:10.1007/978-3-030-45231-5_7.
- 3 Daniel Kirsten. Distance desert automata and the star height problem. *RAIRO Theor. Informatics Appl.*, 39(3):455–509, 2005. URL: <https://doi.org/10.1051/ita:2005027>, doi:10.1051/ITA:2005027.
- 4 Corto Mascle. Personal communication, 2021.
- 5 Corto Mascle, Mahsa Shirmohammadi, and Patrick Totzke. Controlling a random population is exptime-hard, 2019. URL: <https://arxiv.org/abs/1909.06420>, arXiv:1909.06420.
- 6 Sylvain Schmitz. *Algorithmic Complexity of Well-Quasi-Orders*. Habilitation à diriger des recherches, École normale supérieure Paris-Saclay, November 2017. URL: <https://theses.hal.science/tel-01663266>.
- 7 Farhad Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *J. ACM*, 37(2):318–334, apr 1990. doi:10.1145/77600.77620.
- 8 Imre Simon. On semigroups of matrices over the tropical semiring. *Informatique Théorique et Applications*, 28(3-4):277–294, 1994.