On the Complexity of Finite Markov Decision Processes

Danièle Beauquier[1]
Université Paris-12, France

Dima Burago[2] ♮
Laboratory for Theory of Algorithms, SPIIRAN†, St-Petersburg, Russia
and LRI, Université Paris-Sud, France

Michel de Rougemont[3]
L.R.I., Université Paris-Sud, France

Anatol Slissenko[4] ♭
Université Paris-12, France
and Laboratory for Theory of Algorithms, SPIIRAN†, St-Petersburg, Russia

Version of November 28, 2005

Abstract

[1] *Address: Université Paris-12, Equipe d'Informatique Fondamentale, 61, Ave. du Général de Gaulle, 94010 Créteil, France. E-mail: beauquier@univ-paris12.fr*

[2] *Address: Dept. of Mathematics, Pennsylvania State University, University Park, PA 16802, USA E-mail: burago@math.psu.edu*

♮ *The research of this author was supported by DRET and Armines contract 92-0171.00.1013.*

† *St-Petersburg Institute for Informatics and Automation of the Academy of Sciences of Russia*

[3] *Address: Université Paris-11, Centre Orsay, L.R.I., Bât. 490, F-91405 Orsay, France. E-mail: mdr@lri.lri.fr*

[4] *Address: Université Paris-12, Equipe d'Informatique Fondamentale, 61, Ave. du Général de Gaulle, 94010 Créteil, France. E-mail: slissenko@univ-paris12.fr*

♭ *The research of this author was partially supported by DRET contract 91/1061.*

1 Introduction

1.1

We consider a particular case of Markov decision processes, briefly MDP, (e. g. see [Put90]) from the point of view of computational complexity. This case concerns stationary processes with perfect (totally observed) and imperfect (partially observed) information with finite number of states and actions, and under concrete cost criteria. The motivation is, on the whole, standard, i. e. the analysis of situations where the processes entailed by our actions are predictable only with some probability. The common in these problems is that we consequently make decisions to undertake certain actions that change the state of the system, with a goal to reach some desirable state or to realize some behavior. As neither the exact result of the action nor the current state are known precisely, we are in the situation with two-fold uncertainty: we are subjected to probabilistic deviations from the planned results, and we get only partial information about the state where we arrive at.
The traditional formalization considers a finite set of states, a finite set of actions permissible at a state, with every action implying changing the system to another state with a known probability. We slightly deviate from the usual terminology to facilitate our applications, giving, however, references to classical terminology from, e. g. [Put90], [Ber76] .
The states of a decision process can be interpreted as vertices of a graph whose directed edges go from a vertice to all other reachable ones with non zero probability by some action. In other words, we act on a coloured digraph supplied with a function describing the probability to deviate from an edge chosen to go along. A policy , or a strategy, is a function from strings of colors (histories of realizations) to actions. While being realized, the policy traverses vertices, and the colour of a reached vertex is the only new information available at this vertex. The problem is to construct a policy to fulfil some task. One of the simplest tasks to carry out, is to reach target vertices from a source vertex with maximum probability.
Our specific motivations go back to robotics (e. g.[dRDF92], [DF93]) and to some applications related to analysis of programs. One computer science application is model-checking of timed probabilistic transition systems against respective logics, e. g. see [CY95], [BS98].
The first goal was to analyse the *complexity* of constructing policies optimal in different classes of policies, and as one of the further goals, to look at the complexity of optimal policies for situations with more diverse uncertainty. Different models of uncertainty, e. g. [Col87], [Val79], [Pap85], [PY91], [DKP91], [Zal92] remain separated.

1.2

In the next section 2 we give the basic notions from the theory of Markov decision processes related to the problems under consideration, and then specify the criteria of optimality of policies interesting from the point of view of our motivations, and make precise some computational aspects. Here we also introduce a type of graphs convenient for describing concrete processes. In this paper we use mainly as criterion the probability to reach target states from a starting state, maybe under some constraints on admissible behaviors.
Then in section 3 the complexity of the case of total observability (bijective coloring) is surveyed. We clarify the questions of the comlexity of constructing optimal policies. From the point of view of the used technical tools these results could be considered as "almost" known, though they were never explicitly stated in papers on traditional MDP, and moreover the related proofs from that papers are very long.
In section 4 we consider another criterion of quality of policies motivated by robot motion

planning and model-checking, and described in terms of constraints on the admissible behavior of the system. For example, we may demand that the system firstly goes to the state x then to y and then to z, and exactly in this order. Formally, constraints on the behavior are given by a language of admissible sequences (paths) of states (or colors). The basic criterion considered here is the probability to follow admissible paths starting from a source vertex. We restrict ourselves to behaviors described by regular (deterministic finite automaton) languages, for standard larger classes of languages the problem becomes computationally hard. For the same reason only the case of perfect information is analyzed. Here we show that optimal policies are not Markovian in general situation, but they can be found among finite memory ones. This class of policies seems to be of interest in more general settings. We prove that optimal finite memory policies can be found in polytime. Subsection 4.7 points out some limits on the power of finite memory policies. We prove that in the case of partial information, when an optimal policy exists (for infinite horizon), maybe no finite memory policy is optimal (opposite to the case of perfect information).

In section 5 we show that though randomized policies are not better than deterministic ones for finite horizon for the class of history remembering policies, they can be better in the class of finite memory ones.

In a short section 6 for the case of total uncertainty (unobservability) we strengthen corollary 2 from [PT87], and show that even very weak approximations to optimal policies are NP-hard.

Section 7 treats the case of unobservability bounded by a fixed parameter. In terms of colors this means that the number of vertices of the same color is bounded by the parameter. In other words, the set of states is partitioned into classes, the number of elements in every class is bounded by the parameter, and at any moment of execution of a policy we know only the class which the actual state belongs to. The parameter, say m, is called the *multiplicity* of coloring. We show that even for $m = 3$ constructing an optimal policy is NP-hard. But for any m polytime approximations are possible. Finally, relations with Max Word problem are discussed.

2 Main Notions

2.1 Uncertainty Model.

We consider only finite state Markov decision processes (MDP) that are defined by stationary conditions of functioning, see [Put90]. Complexity analysis of infinite or non stationary processes depend on the way of representing the infinite sets involved (say, as blackbox, algorithm of this or that type etc.). 'Being stationary' means here that all the sets and functions characterizing the process, such as the sets V, C, A and the functions clr, ρ, r introduced below, do not depend on time which is considered to be discrete. The latter means that the state of the system changes in moments of time enumerated by natural numbers.

A MDP is a tuple of the form

$$(V,\ A,\ C,\ clr,\ \rho, r), \quad \text{or} \quad (V,\ A,\ C,\ clr,\ \rho, r,\ s),$$

where

• V is a finite set of *states* of a system to control; the states are also interpreted as *vertices* of a graph representing the system.

• C is a finite set of *colors* that represent the observable information.

• $clr : V \to C$ is the *colouring* function. It defines a partition of the states into classes $clr^{-1}(c)$, $c \in C$, which characterizes uncertainty of determining the current state (that is th

type of observability).

• A is a finiteset of *actions*. We may consider that every state $v \in V$ has its own set A_v of actions. To evitate pathological cases we assume the the whole set of actions is polynomially bounded by the number $|V|$ of states.

• $\rho : V \times V \times A \to [0,1]$ is the *transition probability* function. It is usually supposed that for all $\alpha \in A$ and $u \in V$ • s is a probability distribution over the states called the *initial distribution*. It may be concentrated in one state, then we speak about initial state.

$$\sum_{v \in V} \rho(u, v, \alpha) = 1. \tag{1}$$

• $r : V \times A \to \boldsymbol{R}$ is the *reward* function. When positive its value $r(v, \alpha)$ can be thought of as income gained by the action α in the current state v, and when negative as cost to pay for the same action.

For briefness we will also write $\rho(uv, \alpha)$ in place of $\rho(u, v, \lambda)$.

The set A may be interpreted as a set of *actions* or *moves* or even *decisions*, and the function ρ describes the probabilities of deviations: $\rho(uv, \lambda)$ is the probability to arrive at v from u if the action α has been made. For example, one can think of A as local names of outcoming edges, and ρ gives the probability to follow an edge other than the chosen one.

Sometimes it is more convenient to attribute the reward to edges: $r(uv, \alpha)$ is the reward received if the state of the system is u, action α is selected, and the system is in the state v at the next moment of time. In terms of this reward function the one considered above can be represented as

$$r(x, \alpha) = \sum_{y \in V} r(xy, \alpha) \cdot \rho(xy, \alpha).$$

For specific applications, it may be convenient to consider formally more general settings for MDP. For example, one can attribute for each state v is own set A_v of actions. To reduce such a presentation to the initial one we take $A = \bigcup_{v \in V} A_v$ and extend ρ in the obvious way: $\rho(uv, \alpha) = 0$ for $\alpha \notin A_u$.

To avoid some trivialities, we assume $|A|$ is polynomially bounded with respect to $|V|$.

When treated as a part of input of algorithms, ρ is supposed to have rational values and to be represented as a usual table of its values.

Remark. Another generalization apt for some applications in model checking [**?**] presume to distinguish different ways of transition from one state to another. More precisely, we apppend one more object to the syntactic representation of MDP, namely, a set of edges E and 2 maps $^-$ and $^+$ from E to V giving the tail e^- and the head e^+ of $e \in E$. In this setting the transition probability ρ' is a mapping from $E \times A$ to $\boldsymbol{R}$. And, clear, we define in an approriate way the notion of a path representing the mode of changing the states. And again we can reduce such a model to the one given above within some additional complexity cost. We add to V vertices ξ_e for every edge e, and extend ρ: $\rho(\xi_e^- \xi_e, \alpha) = \rho'(e, \alpha)$, $\rho(\xi_e \xi_e^+, \alpha) = 1$, $\rho(xy, \alpha) = 0$ for all other pairs xy of states not related by edges from E, $\alpha \in A$.

To underline the view of MDP as a graph we will call it also *MDP-graph*. Interpreting this structure as a directed graph with the set of vertices V and edges uv defined by the condition $\exists \alpha \in A\, \rho(uv, \alpha) > 0$ is convenient, especially for describing examples, and will be used below.

2.2 Policies.

A *(deterministic) policy* on a MDP-graph G a is a function $\sigma : C^+ \to A$, where C^+ denote the set of all non empty words over alphabet C. So the policies are history remembering policies in the

terminology of the theory of Markov decision processes. Later we will define also probabilistic policies.

Notations for listes of states or, in other words, for *paths* in a MDP-graph G:

• $\mathcal{P}^k_{x,y}$ the set of all paths starting at the vertex x and ending at the vertex y and having k exactly vertices.

• $\mathcal{P}^k_x$ the set of all paths in the graph G having exactly k vertices and starting from x.

• $\mathcal{P}^k(T)$ the set of all paths in the graph G having exactly k vertices and containing a vertex from $T \subseteq V$.

• $\mathcal{P}^k =_{df} \bigcup_{x\in V} \mathcal{P}^k_x$.

• $first(P)$ denotes the first letter of the word P, $last(P)$ denotes the last letter of the word P, and $P[i,j]$ denotes the subword of P composed of letters on positions from i to j the extremities included.

Assume that a starting distribution $s \in V$ is given. I. e. the system can be in a state x with probability $s(x)$. The "semantics" of a policy σ is given by the probabilistic distributions $\mathbf{B}^\sigma$ on $\mathcal{P}^k$ defined as follows:

$$\mathbf{B}^\sigma(v_1 \dots v_{k-1}v_k) = s(v_1) \cdot \prod_{i=1}^{k-1} \rho(v_i v_{i+1}, \sigma(clr(v_1 \dots v_i)), \tag{2}$$

where $clr(v_1 \dots v_i) = clr(v_1) \dots clr(v_i)$. Informally speaking, $\mathbf{B}^\sigma(P)$ is the probability to follow a given path P of the length k when executing σ which acts on the basis on the colors observed during its execution.

Notice that we denote by $\mathbf{B}^\sigma$ many different probabilistic distributions on different discrete spaces. Thus, to avoid confusion we have to apply $\mathbf{B}^\sigma$ only to a subset of one $\mathcal{P}^k$. It will be clear from the context on what set $\mathbf{B}^\sigma$ is being considered.

In fact, one can look at the probabilistic measures generated by a policy σ from a more general point of view. Consider the tree $\mathcal{T}^\infty$ of all infinite paths of G having as a root some special verex not in G. Then with probability $s(v)$ one can arrive at v before executing σ, so all vertices of G are direct descendants of this special vertex as well as of any other one. Every finite path P of $\mathcal{T}^\infty$ starting from the root determines the set of infinite paths with the prefix P which measure is $\mathbf{B}^\sigma(\tilde{P})$, where $\tilde{P}$ denotes P without the first vertex. Then in a standard way one can define a probability distribution on the sigma-algebra of Borel sets of infinite paths of $\mathcal{T}^\infty$ generated by these set determined by finite paths.

The semantics of a policy can be treated from another point of view, namely, via considering a policy as a family of transformations of the set $\mathcal{D}(V)$ of probabilistic distributions on V. The probability of being at a vertex v after exactly k steps of executing σ is

$$\sigma^k(s)(v) = \sum_{u\in V} s(u) \cdot \sum_{P\in\mathcal{P}^k_u \,\&\, last(P)=v} \mathbf{B}^\sigma(P).$$

For a fixed string of colours $c_1 \dots c_k$ we define also the conditional probability

$$\sigma_{|c_1\dots c_k}(s)(v) = \sum_{u\in V} s(u) \cdot \sum_{P\in\mathcal{P}^k_u \,\&\, last(P)=v \,\&\, clr(P)=c_1\dots c_k} \mathbf{B}^\sigma(P).$$

Remark. If to follow our motivations one can notice that the history of actions, i. e. the sequence of chosen actions is an available information, and thus may be included into the argument of σ. One can define the semantics of this type of policies in a similar way as above. However, it is easy to show that for every policy of this 'generalized' type there exists a policy that depends only on the colours of visited vertices and determines the same probabilistic distribution on the set of paths.

2.3 Reliable Moves and Vertices. Simple Graphs.

An action $\alpha \in A$ is called *reliable* at v along vw, $w \in V$ if $\rho(vw, \alpha) = 1$. Such an edge vw will be denoted by $lbl(\alpha, v)$. Such edges will be also called *reliable*. A vertex is said to be *reliable* if every action is reliable at this vertex. A vertex is *random* if the function ρ does not depend on action on all the edges outcoming from this vertex.
A MDP-graph where every vertex is either random or reliable will be called *simple*. Such graphs are convenient to describe, and in particular, they will be used in our examples. On our drawings we use the notations shown on the Figure 1,

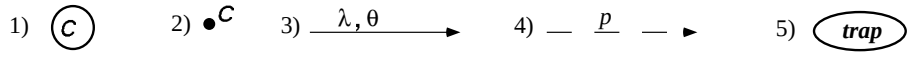

Figure 1: Representation of reliable and random edges and vertices.

where
1) is a reliable vertex coloured by colour c.
2) random vertex coloured by colour c.
3) reliable edge that corresponds to actions λ and θ.
4) edge outcoming from a random vertex; p is the value of ρ (that does not depend on actions).
5) ***trap***, i. e. a vertex where all actions lead back to itself.

One can show that the simple model (even under stronger constraints) is as powerful as the original one.

2.4 Criteria of Quality of Policies.

General definitions of criteria can be found in texts on Markov decision processes, e. g. [Put90], [Ber76]. Here, by a *criterion* we mean a function from the set of policies to real numbers that depends only on the probabilistic distribution defined by a policy. We define below the particular criteria considered in the paper, and just mention a criterion that probably was not considered and that may be of theoretical interest.

(1) Expected reward received in k steps. It is the most common traditional criterion (or a set of criteria). For an initial distribution s and the number of admissible steps (i. e. time) k we define

$$Er_k^s(\sigma) = \sum_{v_1 \dots v_k \in \mathcal{P}^k} \mathbf{B}^\sigma(v_1 \dots v_k) \cdot \Big(\sum_{1 \le i < k} r(v_i v_{i+1}, \sigma(clr(v_1 \dots v_i))) \Big).$$

(1a) Reliability: probability to reach target in not more than k steps. It is a particular case of the expected reward (1). Let $T \subseteq V$ be a target set to reach. This criteria, denoted here by $\boldsymbol{R}_k^{s,T}(\sigma)$, is defined as the probability to reach any vertex from T starting at s in not more than k steps of execution of σ. When s and T are clear from the context we drop them and use the notation $\boldsymbol{R}_k(\sigma)$. This criterion corresponds to the reward 1 on all the edges coming to T. We will often refer to this criterion as *policy reliability*.

(2) Probability of realizing a given behaviour. Let L be a set of paths interpreted as a set of allowed realizations. The criterion $\boldsymbol{R}_k^L(\sigma)$ is the probability to follow only realizations from L (cf. [BBS95], similar criterion was also considered in [**?**]) , see section 4.

(3) Entropy of location. This criterion could be of theoretical interest:

$$H_k(\sigma) = \sum_{v \in V} \sigma^k(s)(v) \cdot \ln \sigma^k(s)(v).$$

To maximize this criterion means to minimize the entropy (i. e. the uncertainty) of the location after k steps of executing σ.
For criterion $\boldsymbol{R}_k^{s,T}(\sigma)$ one can consider also its limit version $\boldsymbol{R}_\infty^{s,T}(\sigma) = \lim_{k\to\infty} \boldsymbol{R}_k^{s,T}(\sigma)$. Notice that the criterion $\boldsymbol{R}_k^{s,T}(\sigma)$ is non decreasing on k, and hence the limit does exist.
For any other criterion $\mathcal{R}$ we can consider $\mathcal{R}_\infty = \sup_k \mathcal{R}_k$.
Hearafter we consider mainly the criteria $\boldsymbol{R}_k^{s,T}$, $k \in \mathbb{N} \cup \{\infty\}$, i. e. the probability to reach T from s in not more than k steps (for natural k) or its limit version ($k = \infty$). If values of some of the parameters k, s or T are clear from the context they will be omitted.
Notation: $p_k^{opt}(v,T) =_{df} \sup\{\boldsymbol{R}_k^{v,T}(\sigma) : \sigma \text{ is a policy}\}$. Thus, $p_k^{opt}(v,T)$ is the 'optimal' probability to reach T from v in not more than k steps.
To make the notion of strategy more constructive we introduce a notion of *universal strategy* which may look a bit cumbersome. To get some intuition, imagine we got lost in a forest or a city, and are seeking to reach some goal. On what information would we base our decision where to go? We would use a map (MDP-graph in our context) that, however, does not allow us to recognize directions for sure. Evidently, our decisions depend on our purpose, that is on the criterion to value possible results of our actions (criterion $\mathcal{R}_r$ below), and that may be rather complex and contain, say, a description of regions that we would not cross, or the time at our disposal. We would also take into account the history of our wandering (a string W of colors).
We assume that possible criteria $\mathcal{R}_r$ are encoded as strings r of some language $\mathcal{X}$. Given a class of MDP-graphs $\mathcal{G}$ and a class of criteria $\mathcal{X}$, a *universal strategy* (for $\mathcal{G}$ and $\mathcal{X}$) is an algorithm σ whose input is of the form $G = ((V, A, C, clr, \rho, s), r, W)$, where $G \in \mathcal{G}$, $r \in \mathcal{X}$ and $W \in C^+$, and whose output is an action from A. For fixed G and r, a universal strategy σ determines a strategy $\sigma_{G,r} : C^+ \to A$.
The semantics of a universal strategy σ is the family of semantics of strategies $\sigma_{G,r}$.
As it was noticed at the beginning of this subsection there are different general notions of criteria of quality of policies. One simple observation is that many criteris are particular cases of the following general criterion $\boldsymbol{R}_k^F(\sigma)$, where F is a function, $F : \mathcal{P}_s \to \boldsymbol{R}$:

$$\boldsymbol{R}_k^F(\sigma) = \sum_{P \in \mathcal{P}_{s,k}} F(P) \cdot \mathbf{B}_k^\sigma(P).$$

2.5 Optimal Policies.

Speaking about optimal policies we distinguish not only criterion with respect to which we consider the optimality, but as well distinguish the amount of time given to optimize the value of the criterion. If the time is bounded by a natural number we speak about *finite horizon*, and we are interested in the limit behavior of our policies we speak about *infinite horizon*. In notation that will be distinguished as respectively $\mathcal{R}_k$ and $\mathcal{R}_\infty$.
From the point of view of observability we distinguish the cases of

– *total observability* (or *perfect information*) when the coloring clr is bijective;

– *bounded unobservability* when there is a constant M_0 such that $|clr^{-1}(c)| \leq M_0$ for all $c \in C$;

– *total unobservability* when $|C| = 1$;

- *partial observability* which represents the general case of coloring.

A policy σ is *maximum* with respect to a criterion $\mathcal{R}$, or $\mathcal{R}$-*maximum* if for every policy σ' we have $\mathcal{R}(\sigma') \leq \mathcal{R}(\sigma)$. In a dual way one can define *minimum policies*.
Obviously, an $\boldsymbol{R}_k$-maximum policy does exist for every finite k since the number of policies that differ for the first k steps is finite. However, there is no $\boldsymbol{R}_\infty$-maximum policy in the example described by Figure 2. Indeed, the actions after an odd number of steps are made at random

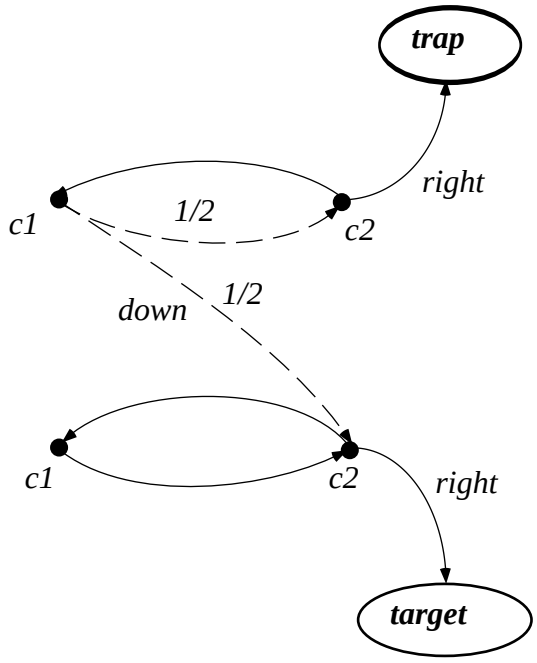

Figure 2: No maximum policy

vertices, and they do not influence the further behaviour. Before we make the action *right* after an even number of steps for the first time, we observe only the colour $c2$, and after this action we arrive either at trap or at target . Thus, any policy is characterized by one integer $2n$: the number of steps after which we decide to go *right*. One can see that $\boldsymbol{R}_\infty$-quality of this policy is $1 - 2^{-n}$.
In the known example given by Figure 3 the first action of an $\boldsymbol{R}_k$-maximum policy differs from

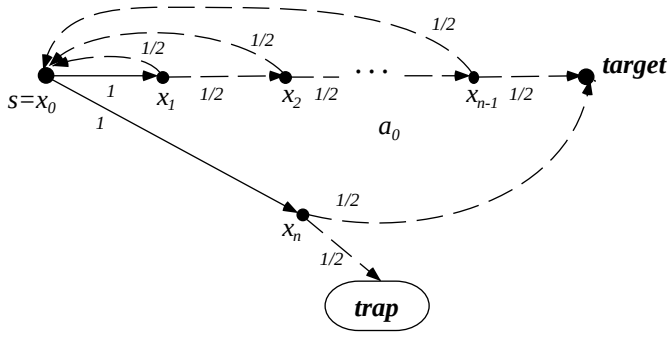

Figure 3: Slow convergence to the maximum probability

the first action of $\boldsymbol{R}_m$-maximum policies for all $m < k$ with k being exponentially greater than

the size of the graph.
Execution of k steps of a policy σ determines a probabilistic distribution of the current position $\sigma_{|c_1 \dots c_k}(s)$ (under the condition that colours of visited vertices constitute the sequence $c_1 \dots c_k$), see subsection 2.2. In many cases this distribution is the only information needed for policy. Namely, we say that a policy σ is PT-*policy* (PT from *probability* and *time* dependent) if there exists a function $f : \mathcal{D}(V) \times \mathbb{N} \to A$ such that

$$\sigma(c_1 \dots c_k) = f(\sigma_{|c_1 \dots c_k}(s), k).$$

(Remind that $\mathcal{D}(V)$ is a probabilistic distribution over V, see subsection 2.2.) E. g., one can prove the following proposition:

Proposition 1 *For every policy σ and for every k there exists a PT-policy σ' such that $\boldsymbol{R}_k(\sigma') \geq \boldsymbol{R}_k(\sigma)$.*

2.6 NP-hardness of Computing the Reliability of a policy.

For the case of bijective coloring, as it is known [Kal83], [Put90] and will be discussed later, that an optimal policy can be found among Markov policies and in polytime. Nevertheless calculating $\boldsymbol{R}_k(\sigma)$ for a particular policy σ maybe computationally hard even for the case of total observability. It can be shown by the following simple reduction of $3SAT$ problem to the mentioned one.
To make the construction easier to understand we represent a $3SAT$-formula over n variables $x_1, \dots, x_n$ as a table. Let

$$F = \bigwedge_{1 \leq i \leq m} \bigvee_{1 \leq j \leq 3} z_{i,j} \tag{3}$$

be such a formula where $z_{i,j}$ are literals, i. e. elements of the set $Z = \{x_1, \dots, x_n, \overline{x_1}, \dots, \overline{x_n}\}$. The table representation of F is shown on fig. 4. In this table of height 3 and length m the ith

$z_{1,1}$	$z_{2,1}$	.....	$z_{m,1}$
$z_{1,2}$	$z_{2,2}$	.....	$z_{m,2}$
$z_{1,3}$	$z_{2,3}$	.....	$z_{m,3}$

Figure 4: $3CNF$-formula F as a table

column corresponds to the ith clause (disjunction) of the formula (4).
A pair z_1, z_2 of literals is said to be *contrary* iff $z_1 \Leftrightarrow \overline{z_2}$.
We assume that no clause (column) in the formulas under consideration contains a contrary pair of literals.
A *path* in F is a horizontal path P in the table composed by picking up one literal of every clause, in other words, P is a list of literals of the form $(z_{1,j_1}, z_{2,j_2}, \dots z_{m,j_m})$, $1 \leq j_i \leq 3$, $1 \leq i \leq m$. We interpret such a path as an assignment of its literals by the value ***true***, which may be inconsistent with the repect to the variables. If such a path does not contain a contrary (and thus contradictory) pair of literals, it gives a boolean model of the $3CNF$-formula. We call a path in F without contrary pairs *open* or *satisfying* path, and a path with a contrary pair of literals *closed* or *contradictory* path.

[hbt]

For a given formula F we build the MDP-graph shown on the figure 5. The vertical triplets of vertices correspond to the columns of the table (figure 4). More precisely, one can consider a

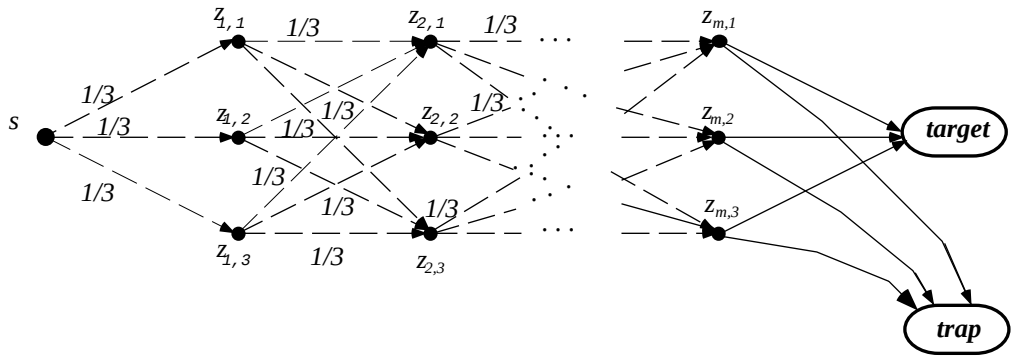

Figure 5: Reduction of $3SAT$ to computing the reliability of a policy.

vertex of the ith triplet labeled by $z_{i,j}$ in the graph of figure 5 either as a vertex with color $z_{i,j}$ or as a vertex with the name $(i, j, z_{i,j})$ (in the latter case the MDP-graph has bijective coloring). Assume that our MDP-graph has 2 actions $\{up,\ down\}$. The dashed edges are random, in other words, when the policy under definition, denote it by ξ, chooses any action, when being in a vertex with outgoing dashed edges, it goes to one of the vertices of the next 'column' with probability 1/3. On the contrary, the solid edges are reliable, i. e. the policy goes along the edge going to target if the action *up* has been chosen, and to trap if *down* has been chosen. The choice of actions is being done as follows. In all columns except the last one, i. e. corresponding to $(z_{m,1} \vee z_{m,2} \vee z_{m,1})$, the policy makes any action, and at a vertex of the last column it choses *up* is the path covered by it by this moment is open, and *down* otherwise. It is clear that F is satisfiable iff $\boldsymbol{R}_k(\xi) > 0$. So, the reduction gives more that we have claimed.

3 Optimal Policies under Total Observability (Bijective Coloring))

In section we consider Er- and R-criterion for the case of bijective coloring, i. e. total observability. We assume that $C = V$ and $clr = \mathrm{id}$.

3.1 M- and T-policies.

The notion of Markov policies, stationary (M-policies) and non stationary (T-policies) are formulated below. They prove to be sufficient to represent optimal policies in the case of total observability.

A policy σ is called M-*policy* if it depends on the last colour of the argument only, i. e. if there exists a function $\sigma' : C \to A$ such that

$$\forall\, W \ :\ \sigma(W) = \sigma'(last(W)).$$

A policy σ is called *T-policy* if it depends on the last colour and the length of its input (time) only, i. e. there exists a function $\sigma' : C \times \mathbb{N} \to A$ such that

$$\forall W \ : \ \sigma(W) = \sigma'(last(W), \mid W \mid),$$

where $\mid W \mid$ denotes the length of W.
When speaking about an M- or T-policy we mean its argument being of the form (v) or (v, m) respectively, where $v \in V$ and $m \in \mathbb{N}$. The underlying interpretation is $v = last(W)$ and $m = |W|$.
It is clear that M-policies correspond to stationary Markov chains, and T-policies to non-stationary ones. Sufficient information on Markov chains can be found in [KS60], [Fel68]. We start with computing R-criterion for T-policies and then describe how to construct optimal T-policies by backward dymanic programming.

3.2 Computation of Reliability of T-policies

Let σ be a T-policy and k be a natural number. As we are interested in computing $\boldsymbol{R}(\sigma, k)$, we can transform G in the following way. Glue all the vertices of T into one and redirect the edges coming out of T back to T. Now the probability to reach this vertex from $v \in V$ by one action $\alpha \in A$ is $\sum_{t \in T} \rho(vt, \alpha)$. Denote the new vertex by the same letter T. Now it is an absorbing vertex. Denote the new graph by the same letter G. So, all policies acting on G that reach T stay there forever. But the value of $R(\sigma, k)$ will be the same as for the original graph.
Now look at σ in this new G. The policy σ determines the Markov chain with transition probabilities

$$p_{vw}(k, k+1) = \rho(vw, \sigma(v, k)),$$

here and below $p_{vw}(k, m)$ denotes the probability to reach w at the moment of time m if the policy is in v at the moment of time k; we assume that $p_{vw}(k, k) =$ **If** $v \neq w$ **Then** 0 **Else** 1.
It is clear that for all $k \leq l \leq m$ the probabilities $p_{vw}(k, m)$ satisfy the (Kolmogorov-Chapman, see [Fel68]) equations:

$$p_{vw}(k, m) = \sum_{k \leq l \leq m} \ \sum_{0 \leq l \leq m} \ \sum_{u \in V} p_{vu}(k, l) \cdot p_{uw}(l, m). \tag{4}$$

We will use the following particular case of the equations (4):

$$p_{sv}(0, m) = \sum_{u \in V} p_{su}(0, m-1) \cdot p_{uv}(m-1, m). \tag{5}$$

Under the assumed conditions, we can express the reliabilty of our policy in terms of $p_{vw}(k, m)$:

$$R(\sigma, k) \ = \ \sum_{0 \leq i \leq k} p_{sT}(0, i). \tag{6}$$

Thus, the equalities (5) and (6) reduce the problem of computing $\boldsymbol{R}(\sigma, k)$ to computing the probabilities $p_{sv}(0, m)$ for $v \in V$, $0 \leq m \leq k$. But the equality (5) gives an evident polytime algorithm to accomplish the calculations.

3.3 Optimal T-policies.

A policy is *everywhere maximum (minimum)* if it is maximum (minimum) for every starting state.
The following result is known (e. g. see [Ber76] or [Put90]) and easily provable by usual dynamic programming which proceeds backward in time starting from the target T.

Proposition 2 *Policies everywhere* $\boldsymbol{R}$*-maximum/minimum for a finite horizon for the class of MDP-graphs with bijective coloring can be found among T-policies, and there is an algorithm that constructs an everywhere optimal T-policy for a given MDP-graph and a natural* k *in polytime. (The time of the algorithm can be estimated as:* $mboxO(k^2 \cdot |V|^2 \cdot (L + fanout(G)))$*, where* L *is the maximum length of the values of* ρ *and* $fanout(G)$=$\max_{v \in V} |\{u : \exists\, \alpha \in A\,(\rho(vu, \alpha) > 0)\}|$*.)*

Proof. The theorem is proved by the backward dymanic programming algorithm T-MAX of Fig. 6. This algorithm constructs a maximum T-policy. A minimum T-policy can be constructed by a similar algprithm, moreover a similar algorithm constructs maximum Er-policy, and in particular, maximum or minimum R-policies. To get this algorithm we are to change the sum to maximize in line 4. Firstly, we prove by induction on k that the algorithm is correct, and then estimate its complexity. □
The construction of optimal T-policies is useful for estimaing $\boldsymbol{R}$-criterion and for constructing M-policies.

T-MAX: **Algorithm for Constructing Maximum T-policy**
INPUT: a MDP-graph G of the form as above, and a natural k. Without loss of generality we assume that every vertex v of G has a "reliable loop", namely for a distinguished action ω: $\rho(vv, \omega) = 1$).
OUTPUT: a T-policy σ with $R_k^{v,T}(\sigma)$ maximum (for all $v \in V$) with respect to all the policies making not more than k steps, and the maximum probabilities $\{p(v,k)\}_{v \in V}$ to reach T from v by σ in not more than k steps.

CommentThe algorithm constructs maximum T-policies σ_i, $1 \leq i \leq k$, such that σ_i tries to reach T from any given vertex v in not more than i steps. It proceeds by induction on $i \leq k$, and together with the maximum policies σ_i the algorithm computes their probabilities $p(v,i) = R_i^{v,T}(\sigma_i)$ to reach T from v in not more than i steps.

Begin
CommentInitialization;
1: **ForAll**$v \in V$ **Do**
$p(v,0)$:= **If**$v \in T$ **Then**1 **Else**0;
CommentRecursion on i;
2: **For**$i := 1$ **To**k **Do**
3: **ForAll**$v \in V$ **Do**
Begin
4: find an action λ such that $\zeta_\lambda =_{df} \sum_{w \in V} \rho(vw, \lambda) \cdot p(w, i-1) \quad \overrightarrow{_{over\ A}} \quad$ max;
5: $\sigma_i(v,1) := \lambda$;
6: $p(v,i) := \zeta_\lambda$;
End;
7: **ForAll**$1 < j \leq i$ **ForAll**$v \in V$ **Do**
$\sigma_i(v,j) := \sigma_{i-1}(v, j-1)$;
end_for;
end_T-MAX;

Figure 6: Algorithm T-MAX for constructing an everywhere maximum T-policy.

3.4 Computation of R-criterion for Markov policies.

We continue to consider the case of total observability. For computing probability characteristics of M-policies the classical theory of finite Markov chains can be, obviously, applied.

3.4.1 Computation of Transient Vertices.

In definitions we can consider T as a set of vertices for generality. We say that a vertex $v \in V \setminus T$ is *transient* (with respect to T) for a T-policy τ and $k \in \mathbb{N}$ if τ reaches T from v with a positive probability in not more than k steps. Similarly, we say that a vertex $v \in V \setminus T$ is *transient* for a M-policy σ if σ reaches T from v with a positive probability in some number of steps. Denote the set of transient vertices for T-policy τ and $k \in \mathbb{N}$ by $VTR(\tau, k)$, and the set transient vertices of M-srategy σ by $VTR(\sigma)$.
Let σ be a M-policy acting on a MDP-graph of the form as above. Firstly, we notice that the set $VTR(\sigma)$ of transient states (vertices) can be found in polynomial time. The problem can be reduced to the problem of finding transient vertices of a T-policy for some k. That is implied by the observation that an M-policy can be treated as T-policy for all k. If a realization of σ, leading from a vertex v to a vertex in T, exists with a positive probability, then with a positive probability there exists an acyclic realization connecting the same vertices. But the length of an acyclic realization is not more than the number of vertices of the graph. Thus the set of transient vertices of a T-policy σ for $|V|$ steps constitute the set of transient vertices for M-policy σ. But the set of transient vertices of an arbitrary T-policy τ for a given k can be found by an algorithm similar to T-MAX. We represent a procedure to do it under the name T-TRANS, see Figure 7.

Lemma 1 *For every T-policy τ and natural k, as well as for every M-policy σ, their sets of transient vertices $VTR(\tau, k)$ and $VTR(\sigma)$ can be found in polytime.*

The classical theory of Markov chains gives rather efficient algorithms to compute $\boldsymbol{R}(\sigma, k)$ for a finite k for a T-policy σ and (for infinite k) for an M-policy σ. Let a MDP-graph G be given. The case $s \in T$ is trivial, and we assume $s \notin T$.

3.4.2 Computation of Reliability of M-policies

Let σ be an M-policy acting on a MDP-graph G.
By p_{uv} (for $u, v \in V$) we denote the *transition probabilities* of σ, i. e. $\rho(uv, \sigma(u))$. Denote by Q the transition matrix of the Markov chain σ on the set of transient vertices $VTR(\sigma)$, that is $Q = (p_{uv})_{u,v \in VTR(\sigma)}$. Without loss of generality we can consider that our chain has 2 absorbing states: one corresponding to T, and denoted by T, and the other corresponding to $V \setminus (T \cup VTR(\sigma))$, and denoted by P. Lemma 1 says that all the mentioned sets can be found in polytime. As was noticed above $\lim_{n\to\infty} Q^n = 0$ because the chain leaves the set $VTR(\sigma)$. This implies that the matrix $1 - Q$ is invertible. The matrix $N = (1 - Q)^{-1}$ permits to accomplish computing of reliability of σ and some other characterics, see proposition 3 below.
Following [KeSn60] denote by $\boldsymbol{M}_v(\varphi)$ and $\boldsymbol{D}_v(\varphi)$ the expectation and variance of a random variable φ for the chain σ starting from v.
Notations:

- ξ is vector-column of 1 of appropriate dimension.
- A_{square} is the matrix (in particular, vector) whose elements are squares of the respective elements of A.
- $\boldsymbol{t}$ is the time (number of steps) when σ rests in $VTR(\sigma)$.

Propositions 3.3.5 and 3.3.8 from [KeSn60] give

T-TRANS: **Algorithm for constructing the set of transient vertices of a T-policy**

INPUT: a MDP-graph G of the form as above, a T-policy τ and a natural k.
OUTPUT: the set $VTR(\tau, k)$ of transient vertices of τ for k and probabilities $p^\tau(v, k)$ for $v \in V$ to reach T from v by τ in not more than k steps.

comment The algorithm computes the probabilities $p^\tau_j(v, i)(forall\ v \in V,\ i + j \le k)$ to reach T from v by τ in not more than i steps if τ has arrived at v after j steps (and it is to accomplish the step $j + 1$).

Begin
CommentInitialization;
1: **ForAll** $j \in [0, k]$ **ForAll** $v \in V$ **Do**
2: $p^\tau_j(v, 0) :=$ **If** $v \in T$ **Then** 1 **Else** 0;
CommentRecursion on i, j, such that $i + j \le k$;
3: **For** $j := k$ **down to** 0 **Do**
4: **For** $i := 1$ **to** $k - j$ **do**
5: **ForAll**$v \in V$ **do**
6: $p^\tau_j(v, i) := \sum_{w \in V} \rho(vw, \tau(v, j + 1)) \cdot p^\tau_{j+1}(w, i - 1)$;
endfor
endfor;
7: **ForAll**$v \in V$ **do**
8: $p^\tau(v, k) := p^\tau_0(v, k)$;
9: $VTR(\tau, k) := \{v \in V \setminus T\ :\ p^\tau(v, k) > 0\}$;
end_T-TRANS

Figure 7: Algorithm T-TRANS

Proposition 3 *Whatever be M-policy* σ
– $\{\boldsymbol{M}_v(\boldsymbol{t})\}_{v \in VTR(\sigma)} = N \cdot \xi$;
– $\{\boldsymbol{D}_v(\boldsymbol{t})\}_{v \in VTR(\sigma)} = (2N - 1) \cdot N \cdot \xi - (N \cdot \xi)_{square}$;
– *the probabilities of absorption in* $a \in \{T, P\}$ *for starting states in* $VTR(\sigma)$ *are given by the vector* $N \cdot \rho_a$, *where* $\rho_a = (p_{ua})_{u \in V}$.

The cited formulas give an algorithm to compute rather efficiently not only $\boldsymbol{R}(\sigma)$ but also the expectation and variance of time to reach T from s. More detailed analysis of the behaviour of σ can be found in [KeSn60].

3.5 Optimal M-policies.

We say that a policy σ *is everywhere* $\mathcal{R}$*-maximum* if it is $\mathcal{R}$-maximum for every starting state. The following theorem is known (see [Put90], Theorem 7.7 or [Kal83]) even for the general case of positive/negative rewards. In our case it can be proven by a direct combinatorial argument.

Theorem 1 *For every MDP-graph with bijective coloring an everywhere $\boldsymbol{R}_\infty^{s,T}$-maximum, and as well as everywhere $\boldsymbol{R}_\infty^{s,T}$-minimum, policy does exist among M-policies.*

Proof. Firstly, we prove the existence of a maximum policy. Let $p_k(v) = p_k^{opt}(v,T)$, where $v \in V \setminus T$ and $k \in \mathbb{N}$, i. e. $p_k(v)$ is the probability to reach T from v in not more than k steps by an $\boldsymbol{R}_k^{v,T}$-maximum policy. Let $p(v) = p_\infty^{opt}(v,T) = \sup_k p_k(v)$. Clearly, we have the monotonicity: $p_k(v) \le p_{k+1}(v) \le p(v)$ and thus, $p(v) = \lim_{k\to\infty} p_k(v)$.
Let

$$z(\alpha, v) = p(v) - \sum_{vu \in E} \rho(vu, \alpha) \cdot p(u).$$

Clear, $z(\alpha, v) \ge 0$. A action α is *admissible* at $v \in V$ if $z(\alpha, v) = 0$.
Define a sequence of sets $V_i \subseteq V$, $V_i \subseteq V_{i+1}$ by induction. Let $V_0 = T$ and

$$V_{i+1} = V_i \cup \{u \in V : \exists \alpha \in A \, \exists \, v \in V_i \ (\rho(uv, \alpha) > 0 \text{ and } \alpha \text{ is admissible at } u)\}.$$

Denote $\hat{V} = \bigcup_i V_i$. Intuitively, $\hat{V}$ consists of the vertices from where T is reachable with non zero probability using only admissible moves. Our nearest purpose is to show that the sequence V_i exhausts all the vertices with non zero probability to reach T.

Lemma 2 *If $v \notin \hat{V}$ then $p(v) = 0$.*

Proof. Reasoning by contradiction, assume $p(v) > 0$. Define

$$\delta = \min\{z(\alpha, w) : \ w \in V \ \& \ \alpha \in A \ \& \ z(\alpha, w) \ne 0\}$$

and choose k such that $p_k(v) \ge \frac{p(v)}{2}$ and $p(v) - p_k(v) \le \frac{\delta \cdot p(v)}{3}$.
Consider an $\boldsymbol{R}_k^{v,T}$-maximum policy σ. Recall that $p_k(v) = \boldsymbol{R}_k^{v,T}(\sigma)$. Denote by $Q \subseteq \mathcal{P}_v$ the set of realizations P such that $|P| \le k$, the action $\sigma(P)$ is not admissible at $last(P)$ and for every proper prefix P' of P the action $\sigma(P')$ is admissible at $last(P')$.
Every realization of σ that leads from v to T has a prefix from Q since $v \notin \hat{V}$. Thus we have $p^\sigma(Q) \ge p_k(v)$ and

$$p_k(v) = \sum_{P \in Q} p^\sigma(P) \cdot p_{k-|P|}(last(P)). \tag{7}$$

Here we use that σ is $\boldsymbol{R}_k^{v,T}$-maximum, and thus, if σ leads to a vertex u in $i < k$ steps with non zero probability then $p_{k-i}(u) = \boldsymbol{R}_{k-i}^{u,T}(\sigma)$. On the other hand,

$$p(v) \ge \sum_{P \in Q} p^\sigma(P) \cdot p(last(P)). \tag{8}$$

Informally speaking, this means that executing several steps of a policy can only spoil the total probability to reach T, and in no case increases it. Subtracting the inequality (7) from (8) , we get

$$p(v) - p_k(v) \ge \sum_{P \in Q} p^\sigma(P) \cdot (p(last(P)) - p_{k-|P|}(last(P)).$$

Taking into account that

$$\sum_{P \in Q} p^\sigma(P) \ge p_k(v) \ge \frac{p(v)}{2}$$

we conclude that $p(last(P_0)) - p_{k-|P_0|}(last(P_0)) \leq \frac{2}{3} \cdot \delta$ for some $P_0 \in Q$. Let $u = last(P_0)$, $m = k - |P_0|$ and $\alpha = \sigma(P_0)$. Recall that α is not admissible, thus

$$p(u) \geq \delta + \sum_{w \in V} \rho(uw, \alpha) \cdot p(w). \tag{9}$$

On the other hand, σ is $\boldsymbol{R}_k$-maximum and α is a move of σ, hence

$$p_m(u) = \sum_{w \in V} \rho(uw, \alpha) \cdot p_{m-1}(w). \tag{10}$$

Subtracting (10) from (9), we get

$$p(u) - p_m(u) \geq \delta + \sum_{w \in V} \rho(uw, \alpha) \cdot (p(w) - p_{m-1}(w)),$$

that leads us to a contradiction with $p(u) - p_m(u) \leq \frac{2}{3} \cdot \delta$. (Recall that $p(w) \geq p_{m-1}(w)$ for all w).

The case of minimum policy can be treated in a similar way (see subsection 3.6.4). □

Let $d(v) = \min\{i : v \in V_i\}$ for $v \in \hat{V}$. We define an M-policy σ by setting $\sigma(v) = \alpha$, $v \in \hat{V}$, where α is some action admissible at v and such that $\rho(vu, \alpha) \neq 0$ for some $u \in V_{d(v)-1}$ (that exists by the definition of $V_{d(v)}$). Then we define σ somehow on $V \setminus \hat{V}$ (T is unreachable from $V \setminus \hat{V}$ and thus, it does not matter how to define σ on this set). We show that σ is R_∞-maximum. Notice that $p(s) \geq R_\infty(\sigma')$ for every policy σ' (by the definition of $p(s)$). Thus it suffices to show that $p(s) = \boldsymbol{R}_\infty(\sigma)$. We will prove that $p(v) = \boldsymbol{R}_\infty^{v,T}(\sigma)$ for all $v \in V' = \hat{V} \setminus T$ by standard arguments from the theory of finite Markov chains.
Denote $q(v) = R_\infty^{v,T}(\sigma)$, $a_{uv} = \rho(uv, \sigma(u))$, $u, v \in V'$ and $b_v = \sum_{t \in T} \rho(vt, \sigma(v))$. Consider the following system of linear equations for variables x_v, $v \in V'$:

$$x_v = b_v + \sum_{u \in V'} a_{vu} \cdot x_u, \quad v \in V'. \tag{11}$$

Both $\{p(v)\}$ and $\{q(v)\}$ satisfy the system (11). Hence $p(v) = q(v)$, $v \in V'$, since the system (11) has a unique solution. Indeed, the corresponding homogeneous linear system has the matrix $\mathbf{1} - \Pi$ where $\mathbf{1}$ is the unit matrix and $\Pi = \{a_{uv}\}_{u,v}$. Using the definition of V', one can easily show that $\Pi^k \to 0$, $k \to \infty$, and thus $\mathbf{1} - \Pi$ is invertable. □

3.6 Polytime Algorithms for Constructing maximum and minimum M-policies.

The following theorem is, in fact, known (see [Kal83], 3.5). In our case it can be proven rather simply using a known reduction to linear programming (e. g. see [Put90]).

Theorem 2 *For MDP-graphs with bijective coloring an everywhere maximum M-policy can be computed in polytime*

We start with some preliminary considerations from theory of finite Markov chains.

3.6.1 Linear Equations for Limit Probabilities

Let σ be a M-policy. It defines a (stationary) Markov chain with a finite number of states. The relation *'σ reaches w from v with a positive probability'* defines the division of the set V into strongly connected components which are partially ordered (see [KeSn60]). Each component is either an ergodic set (where the chain stays forever with probability 1), or a set transient with respect to the union of the ergodic sets (from where we come out with probability 1). A singleton ergodic set is an absorbing state. Without loss of generality, here we consider T as an absorbing state for all the policies under consideration (in formulas T will be often teated as a singleton set).

- $a_x^\beta = \rho(xT, \beta) = \sum_{t \in T} \rho(xt, \beta)$.
- $a_{xy}^\beta = \rho(xy, \beta)$,

where $x, y \in V$, $\beta \in A$.

For a given M-policy σ one can write the evident (see [Fel68], Ch.XV) equations for the limit probabilities to reach T moving only in a set of vertices $W \subseteq V \setminus T$ before having reached T:

$$p_x = a_x^{\sigma(x)} + \sum_{y \in W} a_{xy}^{\sigma(x)} \cdot p_y, \qquad x \in W, \tag{12}$$

where p_x are the probabilities to reach T from x by σ not leaving W before this. The system (12) will be denoted by $Eq^\sigma(W)$. One can easily notice that

Lemma 3 *For every M-policy σ its limit probabilities p_x to reach T from $x \in W$ not leaving W, satisfy the system $Eq^\sigma(W)$ whatever be $W \subseteq V \setminus T$.*

The system $Eq^\sigma(W)$ may have more than one solution for some $W \subseteq V \setminus T$. But on the transient vertices of a policy the probabilities are defined uniquely by the subsystem restricted to these vertices.

Lemma 4 *For every M-policy σ the system $Eq^\sigma(VTR(\sigma))$ has a unique solution.*

Proof. Classical fact. See [KeSn60], th.3.2.1. Let Q be the transition matrix of the Markov chain defined by σ on the set of its transient vertices (states). As the process leaves the set of transient vertices with probability 1, $Q^n \to 0$ when $n \to \infty$. Thus, the matrix of the system i. e. the matrix $\mathbf{1} - Q$, where $\mathbf{1}$ is the unit matrix, is invertible. □

A vertex $v \in V \setminus T$ is *transient (with respect to T)* if there exists a policy that reaches T from v with a positive probability. Denote by VTR the set of transient vertices.

A vertex is a *trap* if no policy leads to T from it with positive probability. Thus, all the vertices are of the 2 types: a *pithole*, i. e. a trap or a vertex of T, or a *transient* vertex. The pitholes are of 2 types : the target T and the traps which we denote by $TRAPS$.

Denote by p_x^σ the probability to reach T from x by an M-policy σ. Now we use a geometric representation for the limit probabilities. Denote by $\boldsymbol{R}^W$, where W is a finite set, a $|W|$-dimensional real space whose coordinates are named by the elements of W. For $W \subseteq V$ the space $\boldsymbol{R}^W$ is canonically embedded into $\boldsymbol{R}^V$. The set of limit probabilities $\{p_x^\sigma\}_{x \in VTR(\sigma)}$ of a policy σ determines a point in $\boldsymbol{R}^{VTR(\sigma)}$ which will be denoted by P^σ. It is clear that for a maximum policy γ we have $VTR(\gamma) = VTR$. Moreover, all the maximum policies γ determine the same point P^γ in $\boldsymbol{R}^{VTR}$, this point will be denoted by Γ.

Lemma 5 *The set VTR of transient vertices and the set $TRAPS$ can be found in polynomial time.*

Proof. Given by the algorithm T-TRANS, see lemma 1. □

3.6.2 Reduction to Linear Programming.

The next step of the construction is crucial. For an arbitrary M-policy σ consider the system of inequalities (cf. (12)) :

$$p_x \geq a_x^{\sigma(x)} + \sum_{y \in VTR(\sigma)} a_{xy}^{\sigma(x)} \cdot p_y, \qquad x \in VTR(\sigma), \tag{13}$$

System (13) defines a convex polyhedron in $\boldsymbol{R}^{VTR(\sigma)}$ that we denote by D^σ.
We say that a point $X \in Q$ is a *minimum* point of a polyhedron $Q \subseteq \boldsymbol{R}^H$, where H is a finite set, if every $Y \in Q$ satisfies the inequalities $Y - X \geq 0$, i. e. all the differences of the corresponding coordinates are non negative. (Clearly, not more than one minimum point may exist.)

Lemma 6 *For every policy σ the point P^σ has positive coordinates and is the minimum point of D^σ. In particular, Γ is the minimum point of all the polyhedrons D^γ for the maximum M-policies γ.*

Proof. The point $P^\sigma = \{p_x\}_{x \in VTR(\sigma)}$ has positive coordinates by its definition. Let $Q = \{q_x\}_{x \in VTR(\sigma)}$ be an arbitrary point of D^σ. We have the system of equations $Eq^\sigma(VTR(\sigma))$ for P^σ, and the system of inequalities

$$q_x \geq a_x^{\sigma(x)} + \sum_{y \in VTR(\sigma)} a_{xy}^{\sigma(x)} \cdot q_y, \qquad x \in VTR(\sigma), \tag{14}$$

for $Q = \{q_x\}_{x \in VTR(\sigma)}$. Subtract equations of $Eq^\sigma(VTR(\sigma))$ from the corresponding inequalities of (14). Using the notation $d_x = q_x - p_x$ we get

$$d_x \geq \sum_{y \in VTR(\sigma)} a_{xy}^{\sigma(x)} \cdot d_y, \qquad x \in VTR(\sigma). \tag{15}$$

It is clear that for some $x \in VTR(\sigma)$ we have positive $a_x^{\sigma(x)}$. And even more, from every point of $VTR(\sigma)$ the policy reaches some x with this property. Now consider the system of inequalities (15) for $\{d_x\}_{x \in VTR(\sigma)}$. Suppose that $d_x < 0$ for some $x \in VTR(\sigma)$. Let z be such that $d_z = \min\{d_x : x \in VTR(\sigma)\}$. We have

$$d_z \geq \sum_{y \in VTR(\sigma)} a_{zy}^{\sigma(z)} \cdot d_y \geq d_z \cdot \sum_{y \in VTR(\sigma)} a_{zy}^{\sigma(z)}. \tag{16}$$

If the sum $\sum_{y \in VTR(\sigma)} a_{zy}^{\sigma(z)}$ is less than 1 then we get a contradiction $d_z > d_z$. Hence the sum of $a_{zy}^{\sigma(y)}$ is 1, and we get the equalities $d_z = d_y$ for all $y \in VTR(\sigma)$ such that $a_{zy}^{\sigma(z)} > 0$. Applying the same procedure to these y we shall arrive at a situation with the sum less than 1 (since $VTR(\sigma)$ is transient), and thus to the contradiction $d_z > d_z$. Hence $d_x \geq 0$ for all $x \in VTR(\sigma)$. □

Now consider the *polyhedron* D defined by the system of inequalities

$$p_x \geq a_x^\lambda + \sum_{y \in VTR} a_{xy}^\lambda \cdot p_y, \qquad x \in VTR, \ \ \lambda \in A. \tag{17}$$

Clear, the size of the system (17) is polynomial with respect to the input. For every maximum policy γ the polyhedron D^γ contains D. This polyhedron reduces our problem to a linear programming problem described below.

Lemma 7 Γ *is a vertex of* D.

Proof. Let γ be a maximum policy. As it was noticed above $\Gamma = P^\gamma$. Since $D \subseteq D^\gamma$ and Γ is the minimum point of D^γ, it suffices to prove that $\Gamma \in D$.
Assume that $\Gamma \notin D$. Then for some $z \in VTR$ and some $\lambda \in D$ we have

$$p_z^\gamma < a_z^\lambda + \sum_{y \in VTR} a_{zy}^\lambda \cdot p_y^\gamma. \tag{18}$$

Consider the policy τ defined as γ on the vertices $x \neq z$, and as λ on z, i. e.

$$\tau(x) = \textbf{if } x \neq z \textbf{ then } \gamma(x) \textbf{ else } \lambda.$$

The limit probabilities of τ satisfy the system

$$p_z = a_z^\lambda + \sum_{y \in VTR} a_{zy}^\lambda \cdot p_y, \tag{19}$$

$$p_x = a_x^{\gamma(z)} + \sum_{y \in VTR} a_{xy}^{\gamma(x)} \cdot p_y, \qquad x \in VTR \setminus \{z\}, \tag{20}$$

i. e. if $p_x^\tau = p_x$, $x \in VTR$ then (19)–(20).

Claim 1. *The limit probabilities of* τ *constitute the only solution of the system (19)–(20).*
The straight line (20) and the hyperplane (19) have at least one common point, namely, the point $\{p_x^\tau\}_{x \in VTR}$ of the limit probabilities of τ (lemma 3), and the hyperplane cannot contain the line because the line contains the point Γ and the hyperplane does not contain it by the assumption.

Claim 2. *The policy* τ *is better than* γ.
Let Φ denote the mapping from $\boldsymbol{R}^{VTR}$ to itself defined by the right parts of the equations (19)–(20). This function Φ is non-decreasing, i. e.

$$(X \leq Y \Rightarrow \Phi(X) \leq \Phi(Y)),$$

and for points X with coordinates between $\mathbf{0}$ and $\mathbf{1}$, i. e. such that $\mathbf{0} \leq X \leq \mathbf{1}$,

$$\Phi(X) \leq \Phi(\mathbf{1}) \leq \mathbf{1},$$

here $\mathbf{0}$ and $\mathbf{1}$ are vectors whose components are zeros and ones respectively. Consider the sequence

$$\Gamma,\ \Phi(\Gamma), \ldots, \Phi^i(\Gamma), \ldots\ ,$$

where $\Phi^{i+1}(X) = \Phi(\Phi^i(X))$. Since $\Gamma < \Phi(\Gamma)$, and Φ is monotonous, the sequence $\Phi^i(\Gamma)$ converges to the solution of the system (19)–(20), i. e. to the limit probabilities of τ. That proves claim 2.

This contradicts to γ being a maximum policy. □

Proposition 4 Γ *is the unique point on* D *that minimizes (on* D*) the linear function* $\sum_{x \in VTR} p_x$.

Proof. It is implied by the fact that $D \subseteq D^\gamma$, lemma 7, and lemma 6. □

3.6.3 An algorithm

Proof of the theorem. A maximum policy can be found by the following polytime algorithm:
1. Find the set VTR, see lemma 5.
2. Find the minimum point Γ of the linear function $\sum_{x \in VTR} p_x$ over the convex polyhedron D by any polytime algorithm for linear programming (e. g. see [PaSt82]).
3. Construct the sets $\{\lambda_x\}_{x \in VTR}$ and U by the following procedure:
 $U \leftarrow T$; -- *Initialization*
 While $VTR \setminus U \neq \emptyset$ **Do**
  Choose x from $VTR \setminus U$ and an action λ_x such that
  $\gamma_x = a_x^{\lambda_x} + \sum_{y \in VTR} a_{xy}^{\lambda_x} \cdot \gamma_y$
   with $a_x^{\lambda_x} \neq 0$ or $a_{xy}^{\lambda_x} \neq 0$ for some $y \in U$ (*);
  Put x in U;
 end_while;
4. Define the output policy σ by the equations: $\sigma(x) = \lambda_x$ for $x \in VTR$.

Claim. *The policy defined by the algorithm is a maximum policy.*
Firstly, notice that **while**-loop exhausts the set VTR and then stops. Indeed, suppose $VTR \setminus U$ is not empty (i. e. VTR has not been exhausted). If one admits that there are no edges from $VTR \setminus U$ to U then one concludes that T is not reachable from some vertices of VTR, namely, from $VTR \setminus U$. A contradiction. Hence, some maximum policy σ leads from a vertex of this set to U. For this σ there are x and λ_x such that (*). But $P^\sigma = \Gamma$, hence the algorithm can find x and λ_x satisfying (*).
The policy σ defined by the equations $\sigma(x) = \lambda_x$ for $x \in VTR$ is maximum. Indeed, the system $Eq^\sigma(VTR(\sigma))$ has a unique solution (Lemma 4). But $VTR(\sigma) = VTR$ because of conditions (*), and Γ is a solution of this system. So we have $P^\sigma = \Gamma$. It proves that σ is a maximum policy. □

Remark 1. The point Γ may represent exponentially many policies, see figure 8 **Remark 2.**

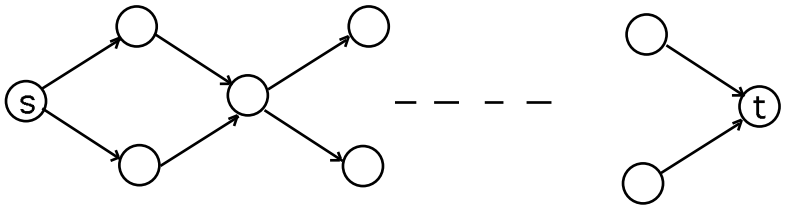

Figure 8: An example of a graph for which Γ represents exponentially many policies

The polyhedron D maybe degenerated, see figure 9. In this example, moves are certain, so any M-policy choosing an edge to t is an R-maximum one.

3.6.4 Constructing minimum M-policy.

Intuitively, it is clear that that the problem of minimum policy can be reduced to the problem of maximum policy. For example, minimum policy to reach T is maximum policy to reach the maximum recurrent set of the MDP not intersecting T. We will give this reduction in detail below.
A set W is *recurrent* if for any its vertex x there exists an action $\alpha \in A$ not leading out of W, i. e. such that for each edge $y \in V$ if $\rho(xy, \alpha) > 0$ then $y \in W$. (An equivalent definition: the

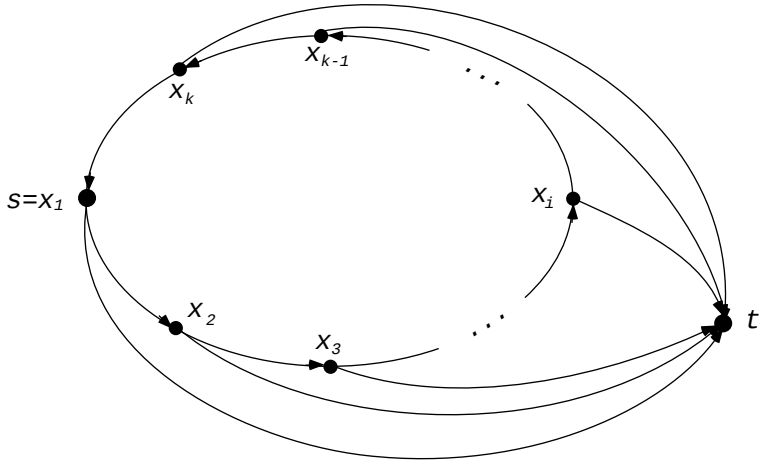

Figure 9: An example of degenerated D

set of vertices such that for every vertex there is a policy which probability to reach T is 0.) In particular, for any policy σ any its ergodic set is recurrent.
Now take all recurrent sets not intersecting T, and denote by U their union. Clearly, U is recurrent and does not intersect T. This set U may be empty. In this case T will be reached with probability 1 by any policy.

One can show that the minimum value of $\boldsymbol{R}_\infty^{s,F}$ is the complement to 1 of the maximum value

$R_0 := V \setminus T;\ i := 0;$
Repeat
$R_{i+1} := \{x \in R_i :\ \exists \alpha \in A\ ReachableVertices_\alpha(x) \subseteq R_i\};$
$i := i + 1$
Until$R_i = R_{i-1}$;
$U := R_i$.

Figure 10: Algorithm to compute U.

of $\boldsymbol{R}_\infty^{s,U}$ for U.
Indeed, let p^+ be the maximum probability to reach U starting from s, and let p^- be the minimum probability to reach T starting from s:

$$p^+ = max_\sigma\{\boldsymbol{R}_\infty^{s,U}(\sigma)\}, \quad p^- = min_\sigma\{\boldsymbol{R}_\infty^{s,F}(\sigma)\}.$$

Consider an arbitrary M-policy σ. Denote by $TRAPS(\sigma)$ the union of all ergodic sets of σ non intersecting (and thus, different from) T ($TRAPS(\sigma)$ may be empty). Clearly, $TRAPS(\sigma) \subseteq U$ because for every vertex of $TRAPS(\sigma)$ the policy σ assures not leaving this set. But with probability 1 policy σ will be trapped either by T or by $TRAPS(\sigma)$. Thus

$$\boldsymbol{R}_\infty^{s,T}(\sigma) + \boldsymbol{R}_\infty^{s,TRAPS(\sigma)}(\sigma) = 1. \tag{21}$$

Denote by $\widehat{\sigma}$ the following modification of σ : policy $\widehat{\sigma}$ coincides with σ for all vertices except vertices $x \in U$ such that $\boldsymbol{R}_\infty^{x,T}(\sigma) > 0$; for such an x define $\widehat{\sigma}(x) = \alpha$, where α is an action

providing $\rho(xy,\alpha)=0$ for all $y\in V$ such that and $y\notin U$. For the policy $\widehat{\sigma}$ the set U is contained in $TRAPS(\widehat{\sigma})$, and thus U is the union of all ergodic sets of $\widehat{\sigma}$ different from T. Hence, from the construction of $\widehat{\sigma}$ and (21) follows

$$\boldsymbol{R}_{\infty}^{s,T}(\widehat{\sigma})+\boldsymbol{R}_{\infty}^{s,U}(\widehat{\sigma})=1, \tag{22}$$

$$\boldsymbol{R}_{\infty}^{s,T}(\widehat{\sigma})\le\boldsymbol{R}_{\infty}^{s,T}(\sigma),\qquad \boldsymbol{R}_{\infty}^{s,U}(\sigma)\le\boldsymbol{R}_{\infty}^{s,U}(\widehat{\sigma}). \tag{23}$$

The latter equations are evident (look, e g. at the first one of (23)): any trajectory of $\widehat{\sigma}$ that reaches T does not touch U, and thus, is also a trajectory of σ.
Take a Markov policy ζ that realizes the probability $p^-=\boldsymbol{R}_{\infty}^{s,T}(\zeta)$. From the minimality of ζ and (22), (23) we get

$$p^-=\boldsymbol{R}_{\infty}^{s,T}(\widehat{\zeta})=1-\boldsymbol{R}_{\infty}^{s,U}(\widehat{\zeta})\ge 1-p^+ \tag{24}$$

because p^+ is the maximum probability to reach U.
Similarly, for an M-policy φ such that $p^+=\boldsymbol{R}_{\infty}^{x,U}(\varphi)$, we get

$$p^+=\boldsymbol{R}_{\infty}^{s,U}(\widehat{\varphi})=1-\boldsymbol{R}_{\infty}^{s,F}(\widehat{\varphi})\le 1-p^-. \tag{25}$$

The inequalities (24), (25) give $p^-=1-p^+$.

So, in order to compute p^- in polytime it suffices to prove that the set U can be computed in polytime. This can be done by the algorithm of Fig. 10, where $ReachableVertices_{\alpha}(x)$, $\alpha\in A$, represents the set of vertices y such that $\rho(xy,\alpha)>0$, i. e. the set of the vertices reachable from x in one step with a non zero probability by action α.
Clearly, the running time of the algorithm of Fig. 10 is polynomial in the size of the MDP-graph. Thus, a desirable policy can be found as a maximum policy to reach U.

3.7 Nonperiodicity of First Actions of Optimal T-policies

We fix now a MDP-graph G and T, and will be interested in the behaviour of $\boldsymbol{R}_k$-maximum T-policy when k grows. Remind that $p_k^{opt}(s,T)$ is the value of $\boldsymbol{R}_k^{s,T}$-criterion for an $\boldsymbol{R}_k^{s,T}$-maximum T-policy, and $p_{\infty}^{opt}(s,T)$ is the value of $\boldsymbol{R}_{\infty}^{s,T}$-criterion for an maximum M-policy. It is clear that $p_k^{opt}(s,T)$ converges to $p_{\infty}^{opt}(s,T)$ when k tends to infinity. The actions of $\boldsymbol{R}_k^{s,T}$-maximum T-policies also converge to the actions of an maximum M-policy in the following weak sense.
Denote by $D_k(v)$, $v\in V$, the set of actions of all $\boldsymbol{R}_k^{v,T}$-maximum T-policies on the input $(v,1)$. Thus $D_k(v)$ is the set of possible first moves of policies that lead to T from v in not more than k steps with maximum probability. Then there exists a natural N such that for every $k\ge N$ there exists an maximum M-policy σ such that $\sigma(v)\in D_k(v)$ for every vertex v. Such minimum N is not more than exponentially large on the size of G (that can be proved by using estimations on root separation for the characteristic polynomials). However, this convergency actually may be exponentially slow, see Figure 3. Moreover, the sequence of sets $D_k(v)$ not necessary stabilises when k grows, see Figure 3.7. It is not hard to see that the first action of an $\boldsymbol{R}_k$-maximum T-policy depends on $k \bmod 4$: $D_{4l+1}(s)=D_{4l+3}(s)=\{r,l\}$, $D_{4l}(s)=\{r\}$, $D_{4l+2}(s)=\{l\}$. In this example $D_k(v)$ depends on k (ultimately) periodically. However, this is not the general case:

Theorem 3 *([BBS94]) There exists a MDP-graph G such that the sequence $D_k(s)$ is not (ultimately) periodic on k.*

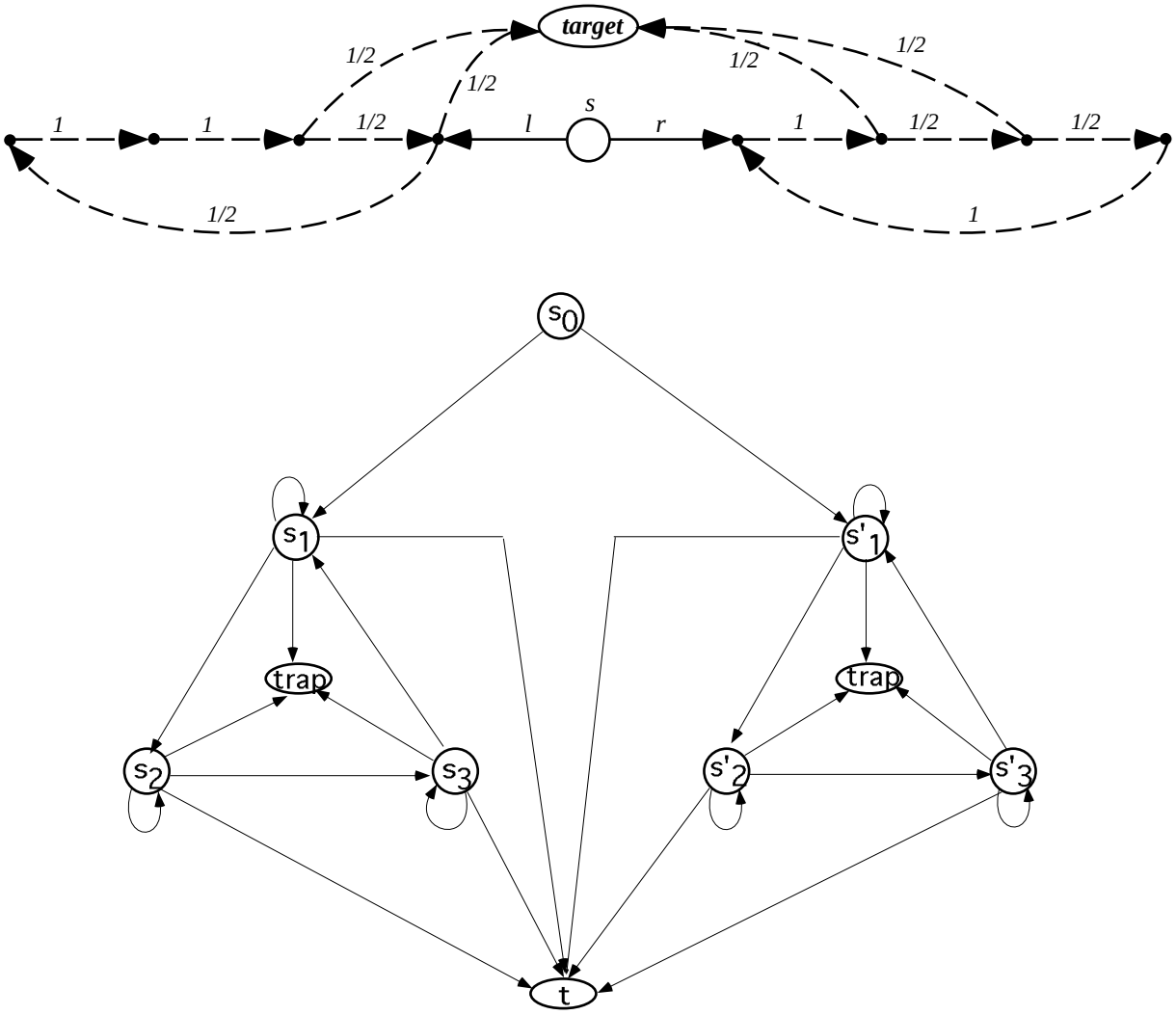

Figure 11: The graph G.

Proof. We divide the proof into 3 parts.

Construction of MDP-graph.

We start with the following MDP-graph G of Figure 11 defined as follows.

- The set of vertices is $V = \{s_0,\ s_1,\ s_2,\ s_3,\ s'_1,\ s'_2,\ s'_3,\ t,\ trap\}$.
- The set of actions $A = \{d,\ \bar{d}\}$.
- The vertex s_0 is reliable and $lbl(d, s_0) = s_0s_1,\ lbl(\bar{d}, s_0) = s_0s'1$.
- For the other vertices we have:
  - Action d:
  $\rho(\ s_is_{i+1}, d) = \frac{1-\alpha}{2}$ for $i = 1,\ 2,\ 3$,
  $\rho(\ s_is_i, d) = \alpha/2,\ \ \rho(s_it, d) = 1/2$,
  $\rho(s'_is'_{i+1}, d) = \frac{1-\alpha}{2}$ for $i = 1,\ 2,\ 3$
  $\rho(s'_is'_i, d) = \alpha/2,\ \ \rho(s'_it, d) = 1/2$
  (the sum of indices is mod 3).
  - Action $\bar{d}$:
  $\rho(s_it, \bar{d}) = 1/2,\ \ \rho(s_itrap, \bar{d}) = 1/2$ for $i = 2,\ 3$,

$\rho(s_1t,\bar{d}) = 1/2+\varepsilon, \quad \rho(s_1trap,\bar{d}) = 1/2-\varepsilon,$
$\rho(s'_it,\bar{d}) = 1/2, \quad \rho(s'_itrap,\bar{d}) = 1/2 \text{ for } i = 1,\ 3,$
$\rho(s'_2t,\bar{d}) = 1/2+\varepsilon, \quad \rho(s'_2trap,\bar{d}) = 1/2-\varepsilon,$
(ε is less than 1/4, and some restrictions on α, $0 < \alpha < 1$, will be imposed later). Figure 12 shows the probability distributions for edges according to action d or $\bar{d}$.

Structure of Actions of Optimal T-policies.

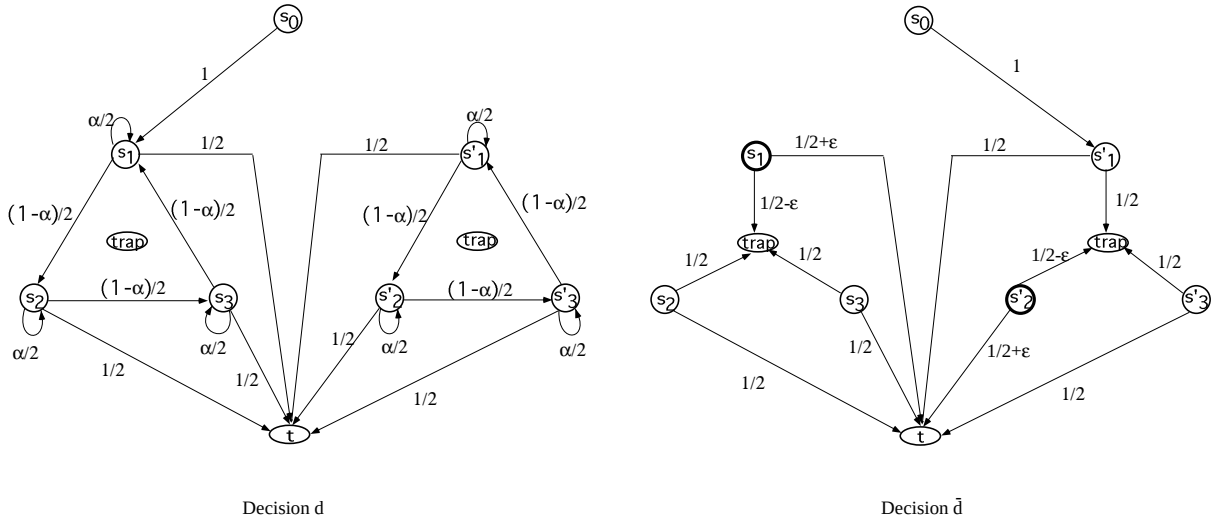

Figure 12: Probabilities of deviations for different actions.

Trivially, the first action of every maximum T-policy is either d or $\bar{d}$. After this first action every maximum policy makes action d as long as possible, i. e. up to the last move, because action d never leads to *trap*, and action $\bar{d}$ made in any vertex s_i or s'_i, $1 \le i \le 3$, definitely leads to t with probability not greater than $\frac{1}{2}+\varepsilon$ and to *trap* with probability not less than $\frac{1}{2}-\varepsilon$, and thus, is always worse than 2 consecutive actions d.

So, there may be 2 types of T-policies maximum for a given k, a policy σ of the first one makes action d at the first step, and a policy $\bar{\sigma}$ of the second one makes action $\bar{d}$ at the first step, and then the both make action d at steps $2, 3, \ldots, k-1$. Here we notice that that the probability distribution of σ being at s_i, $1 \le i \le 3$, after $k-1$ steps is the same as the probability distribution of $\bar{\sigma}$ being at s'_i after $k-1$ steps. And the probability of σ being at s_i, and respectively of $\bar{\sigma}$ being at s'_i, after $k-1$ steps is clearly greater than 0. For this reason $\bar{d}$ as the last action of any of the policies is better than d. So, we assume that the last action of the both policies is $\bar{d}$.

If we consider any 2 policies σ and $\bar{\sigma}$ of the just mentioned types they can be compared with respect to the probabilities of being respectively in s_1 and s'_2 after $k-1$ steps, because the probability to reach t from these vertices is greater than the same probability for other vertices reachable by the corresponding policy. More precisely, denote by $p_k(v)$ (respectively, $\bar{p}_k(v)$), $k \ge 0$, the probability to arrive at vertex v after having executed exactly k steps of σ (respectively, $\bar{\sigma}$). Then $p_k(v) = \bar{p}_k(v')$ for $k \ge 1$.

If $p_k(s_1) > \bar{p}_k(s'_2) = p_k(s_2)$ for $k \ge 1$ then σ is better than $\bar{\sigma}$, i. e. $R^{s_0,t}_{k+1}(\sigma) > R^{s_0,t}_{k+1}(\bar{\sigma})$. So, the first move of an maximum T-policy for $k \ge 2$ is determined by what of the inequalities

$$p_k(s_1) > p_k(s_2) \tag{26}$$
$$p_k(s_1) < p_k(s_2) \tag{27}$$

takes place for this k.

Computing probability distribution for transitive vertices.
We compute the probabilities $p_k(s_i)$ for the policy σ that makes d as the first k actions. We will denote these probabilities by $p_{i,k-1}$ for simplicity. The matrix of transition probabilities for action d is $\frac{1}{2}M$ where

$$M = \begin{pmatrix} \alpha & 0 & 1-\alpha \\ 1-\alpha & \alpha & 0 \\ 0 & 1-\alpha & \alpha \end{pmatrix}$$

(Notice that M is a bistochastic matrix.)
And we have (for $k \geq 0$)

$$\begin{pmatrix} p_{1,k} \\ p_{2,k} \\ p_{3,k} \end{pmatrix} = (\frac{1}{2}M)^k \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{28}$$

To compute M^k we transform M to its eigenvectors basis and then come back to the initial basis. The characteristic polynomial of M, i. e. the polynomial $\det(t \cdot E - M)$, where E is the unity matrix, is $(t-\alpha)^3 - (1-\alpha)^3$. The eigenvalues of M are 1, $\lambda = \alpha + (1-\alpha)\zeta$, and $\bar{\lambda}$, where $\zeta = \mathrm{e}^{\frac{2\pi i}{3}}$, $i = \sqrt{-1}$, $\bar{z}$ is the complex number conjugate to z. The eigenvectors of M are

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \bar{\zeta} \\ \zeta \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \zeta \\ \bar{\zeta} \\ 1 \end{pmatrix}.$$

Transforming M to its eigenbasis we get

$$C^{-1}MC = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ \zeta & \bar{\zeta} & 1 \\ \bar{\zeta} & \zeta & 1 \end{pmatrix}\begin{pmatrix} \alpha & 0 & 1-\alpha \\ 1-\alpha & \alpha & 0 \\ 0 & 1-\alpha & \alpha \end{pmatrix}\begin{pmatrix} 1 & \bar{\zeta} & \zeta \\ 1 & \zeta & \bar{\zeta} \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \bar{\lambda} \end{pmatrix}.$$

Hereupon we get

$$C^{-1}M^kC = (C^{-1}MC)^k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \lambda^k & 0 \\ 0 & 0 & \bar{\lambda}^k \end{pmatrix},$$

and hence

$$M^k = \frac{1}{3}\begin{pmatrix} 1 & \bar{\zeta} & \zeta \\ 1 & \zeta & \bar{\zeta} \\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \lambda^k & 0 \\ 0 & 0 & \bar{\lambda}^k \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 \\ \zeta & \bar{\zeta} & 1 \\ \bar{\zeta} & \zeta & 1 \end{pmatrix} = \begin{pmatrix} 1+\lambda^k+\bar{\lambda}^k & \dots & \dots \\ 1+\bar{\zeta}\lambda^k+\zeta\bar{\lambda}^k & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix}.$$

The latter equalities and (28) give $p_{1,k} = \frac{1}{2^k}(1+\lambda^k+\bar{\lambda}^k)$ and $p_{1,k} = \frac{1}{2^k}(1+\bar{\zeta}\lambda^k+\zeta\bar{\lambda}^k)$. So, the inequality, say, (26) becomes

$$\lambda^k + \bar{\lambda}^k > \bar{\zeta}\lambda^k + \zeta\bar{\lambda}^k.$$

The latter is equivalent to $\lambda^k(1-\bar{\zeta}) + \bar{\lambda}^k(1-\zeta) > 0$, and hence to $2\mathrm{Re}(\lambda^k(1-\bar{\zeta})) > 0$ where $\mathrm{Re}(z)$ denotes the real part of a complex number z. Denoting by θ the argument of λ we can rewrite the latter inequality as

$$\cos(k\theta + \frac{\pi}{6}) > 0. \tag{29}$$

Now we look at the definition of λ:

$$\begin{aligned}\lambda &= \alpha + (1-\alpha)\zeta = \alpha + (1-\alpha)(\cos\frac{2\pi}{3} + i\sin\frac{2\pi i}{3}) \\ &= \alpha + (1-\alpha)(-\frac{1}{2} + i\frac{\sqrt{3}}{2}) = \frac{1-3\alpha}{2} + i\frac{(1-\alpha)\sqrt{3}}{2}\end{aligned}$$

that shows

$$\cos\theta = \frac{1-3\alpha}{2\sqrt{1-3\alpha+3\alpha^2}}\,. \tag{30}$$

One can choose a rational α sufficiently small, as required by our reasoning, and such that θ is not a rational multiple of π.
Indeed, consider an equation

$$\cos\frac{p}{q}\pi = f(\alpha) \tag{31}$$

where $(p,q) = 1$, $0 < p < q$, $\alpha \in [\alpha_0, \alpha_1]$, $\alpha_0 < \alpha_1$ and f is an algebraic function over $\mathbb{Q}$, for simplicity supposed to be without critical points and monotone in $[\alpha_0, \alpha_1]$. Let f be defined by the equation

$$a_0(x) + a_1(x)y + \cdots + a_d(x)y^m = 0$$

where $a_j(x) \in \mathbb{Q}[x]$, $0 \le j \le d$. All values $f(\alpha)$ for rational $\alpha \in [\alpha_0, \alpha_1]$ are algebraic numbers of the degree not greater than m. On the other hand, $\cos\frac{p}{q}\pi = \frac{1}{2}(\zeta_q^p + \zeta_q^{-p})$ where $\zeta_q = \mathrm{e}^{\frac{2\pi i}{q}}$, and thus is also an algebraic number, but of the degree not less than $\phi(2q)$ where ϕ is Euler function. It is known that $\phi(n) \ge c\,\frac{n}{\ln\ln n}$ for some constant $c > 0$. Hence, the set of algebraic numbers of degree not greater than m having the form $\cos\frac{p}{q}\pi$ (under the mentioned conditions on p and q) is finite. So, the equation (31) may be satisfied at most by a finite number of triples (p, q, α) with $\alpha \in \mathbb{Q}$.
Now let α be a rational number such that (30) and θ is not a rational multiple of π. Denote by D_k the Boolean value of (29). Suppose that the sequence $\{D_k\}_{k\in\mathbb{N}}$ is ultimately periodic with period T, and the periodicity begins at k_0. Suppose, without loss of generality, that $\cos(k_0\theta + \frac{\pi}{6}) \in [0,1]$. Then, denoting $\eta_0 = k_0\theta + \frac{\pi}{6}$ and $\eta = T\theta$ we have $\cos(\eta_0 + i\eta) \in [0,1]$ for all $i \in \mathbb{N}$. But η is not commensurable with π, and hence $\{i\eta\}_{i\in\mathbb{N}} \pmod{2\pi}$ is dense in $[0, 2\pi]$. Thus, the sequence $\{\cos(\eta_0 + i\eta)\}_i$ is dense in $[-1, 1]$. □

4 Finite Memory policies

4.1 Policies under Finite Automata Constraints.

In this section we consider the strategies whose realizations are confined to a regular (finite automaton) language [BBS95]. We give a generalization of the results concerning Markov strategies. As above, the case of bijective colouring is treated, so we identify the vertices and their colours.
In the case of robot motion planning, one can think about some constraints which the robot is submitted to, for example, during its motion the robot has to fulfil some tasks, which implies its trajectory cannot be anyone. Consider the following 3 examples of reasonable behaviours and their formal descriptions.
Exemple 1. Every time the robot reaches some place u it has to visit place v before visiting u later.

Exemple 2. During its motion from s to T, the robot has to visit places u, v and w only once and in this order.
Exemple 3. The robot cannot visit the place u more often than the place v.
These constraints can be formalized by defining the allowed paths of the robot from s to T as a language $L \subset V^*$. In the above examples, the languages are respectively

$$L_1 = (V \setminus \{u\})^*\{u\}(V \setminus \{u,v\})^*\{v\}(V \setminus \{u\})^*)^*(V \setminus \{u\})^*,$$

$$L_2 = V'^* u V'^* v V'^* w V'^*,$$

$$L_3 = \{w \in V^* \mid \mid w \mid_u \leq \mid w \mid_v\},$$

where $V' = V \setminus \{u, v, w\}$, and $|w|_u$ denotes the number of occurrences of u in w.
The languages L_1 and L_2 are regular (finite automata) languages, opposite to L_3 which is not regular.

4.2 R^L-criterion.

The R^L-criterion has been defined together with the other ones in subsection 2.4. Let a MDP-graph G is fixed. Denote by $\mathcal{P}^{s,T}_{\leq k}(G, L)$ or simply by $\mathcal{P}^{s,T}_{\leq k}(L)$if G is clear, the set of all paths of the length not greater than k between s and a vertex of T in G, which belong to L and without any proper prefix with the same property. Sometimes we call paths with the latter property *simple realisations*. For a policy σ acting on G define

$$\boldsymbol{R}^L(\sigma, k) = \sum_{W \in \mathcal{P}^{s,T}_{\leq k}(L)} \mathbf{Prob}(W \text{ is a } L\text{-realization of } \sigma) \tag{32}$$

The unbounded version of this criterion serves to formulate the problem of maximizing the probability to reach T from s via L-realizations without limitations on time. Its formal definition is

$$\boldsymbol{R}^L_\infty(\sigma) = \sup\{\boldsymbol{R}^L_k(\sigma) : \ k \in \mathbb{N}\}.$$

As $\boldsymbol{R}^L_k(\sigma)$ is non decreasing on k we can replace sup by lim is this definition.
For general languages L optimization or even computation of $\boldsymbol{R}^L$-criterion is hard. So, we deal only with the $\boldsymbol{R}^L$-criterion for regular (finite automata) languages L of sequences of states.
Recall the definition of regular language.

4.3 Regular languages.

The following notions are standard, see e. g. [Eil74].
A *finite (complete and deterministic) automaton* $\mathcal{A}$ over the alphabet Σ is a quintuple $(\Sigma, Q, \delta, q_0, F)$, where

- Σ is the input alphabet,
- Q is a finite set of *states*,
- δ is a function: $Q \times \Sigma \to \Sigma$ called the *transition function*,
- $q_0 \in Q$ is the *initial state*,
- $F \subset Q$ is a set of *final states*.

The function δ can be extended to $Q \times \Sigma^*$ as follows:

$$\delta(q, \varepsilon) = q, \ \delta(q, Wa) = \delta(\delta(q, W), a) \text{ for } q \in Q,\ W \in \Sigma^*,\ a \in \Sigma,$$

where ε denote the empty word.
For short, $\delta(q, W)$ will be denoted $q.W$. A word $W \in \Sigma^*$ is *accepted by* $\mathcal{A}$ if $\delta(q_0, W)$ belongs to F. The set of words accepted by $\mathcal{A}$ is denoted by $L(\mathcal{A})$. A language $L \subset \Sigma^*$ is a *regular language* if there exists a finite (deterministic and complete) automaton $\mathcal{A}$ such that $L = L(\mathcal{A})$. If the set of final states is not precised, the automaton is called a *finite transition system.*

4.4 R^L-optimal policies are not Markovian.

According to the previous section, as far as we consider the $\boldsymbol{R}$-criterion, the class of M-policies or T-policies is sufficient to find a maximum one, for infinite and finite horizon respectively. It is no longer the case for the R^L criterion as it is proved by the following example.

Proposition 5 *There exists a MDP-graph and a regular language L for which no T-policy is $\boldsymbol{R}^L_k$-optimal ($k \geq 4$), and no M-policy is $\boldsymbol{R}^L_\infty$-optimal.*

Proof. Consider the graph of Fig. 13, and let L be the language specifying that the system

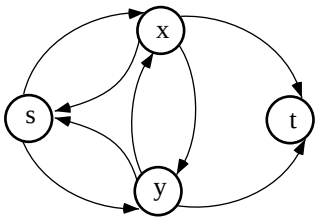

Figure 13: A counter-example.

starts in state s, each time it is in state x, it has to reach next time state y, it has to be in state x at least once, and it has to stop in state t. In other words, $L = s\{y, s, t\}^*(xy\{y, s, t\}^*)^+t$. The set of actions here is $\{s, x, y, t\}$. We define ρ in the following way:

• when the system is in state s, if action x or y is taken the system goes in state x or state y with the same probability $1/2$.

• when the system is in state x, if action x is choosen, the system remains in state x. If another action d is taken, the system goes in state d with probability $1 - 2\epsilon$ and in both states of $\{y, s, t\} \setminus \{d\}$ with probability ϵ.

• the same rule holds for y permuting x and y. For state t, whatever is the action, the system remains in state t. We suppose $\varepsilon < 1/6$ (for technical reasons).

Let σ be a T-policy, and $\boldsymbol{R}^L(\sigma, k)$ be the criterion to evaluate ($k \geq 4$). Suppose that σ is $\boldsymbol{R}$-optimal *among T-policies*. It is easy to prove that whatever is the value of $\sigma(y, 3)$, there exists a policy σ' better than σ, i. e. satisfying $R(\sigma', k) > R(\sigma, k)$. The same kind of reasoning holds for M-policies. More detailed demonstration for T-policies is as follows.
Let σ be a T-policy, and $\boldsymbol{R}^L{}_k(\sigma)$ be the criterion to evaluate ($k \geq 4$). Suppose that σ is $\boldsymbol{R}$-optimal *among T-policies*. We will prove that whatever be the value of $\sigma(y, 3)$, there exists a policy σ' better than σ, i. e. satisfying $\boldsymbol{R}(\sigma', k) > \boldsymbol{R}(\sigma, k)$.
Consider 3 possible cases corresponding to 3 possible actions of σ from y after $k - 3$ steps. Actually there are 2 cases.

Case 1. $\sigma(y, 3) = t$ or $\sigma(y, 3) = s$. Consider a path W from s satisfying the three following conditions:

(i) W has length $k - 2$,

(ii) W has no occurrence of t and no occurrence of x,
(iii) y is a suffix of W.
Such a string is a realization in $k-3$ steps of any policy with a positive probability. Clearly, σ is not a "good" policy for W. There exists a better policy σ' (but σ' is not a T-policy). The policy σ' is the same as σ, except for W, Wx and Wxy. We define $\sigma'(W) = x$, $\sigma'(Wx) = y$ and $\sigma'(Wxy) = t$. Suppose that after $k-3$ steps, a realization of σ or σ' is W. In that situation, the probability to reach t in an admissible way with respect to L in at most k steps is less than ε for the policy σ and equal to $(1-2\varepsilon)^3$ for policy σ'. On the other hand, if the policy σ or σ' has followed in $k-3$ steps a path $W', W' \neq W$, the probability to reach t in an admissible way with respect to L in at most k steps is the same for σ and σ'. So, we have $\boldsymbol{R}(\sigma', k) > \boldsymbol{R}(\sigma, k)$.

Case 2. $\sigma(y, 3) = x$. Consider a string W satisfying the three following conditions:
(i) W has length $k-2$,
(ii) W has no occurrence of t and has one occurrence of x,
(iii) xy is a suffix of W.
There exists a better policy σ' (but σ' is not a T-policy), which is the same as σ, except for W. We define $\sigma'(W) = t$. Suppose that after $k-3$ steps, a realization of σ or σ' is W. The probability to reach t in an admissible way with respect to L in at most k steps is less than $(1-2\varepsilon)^3 + \varepsilon$ for the policy σ and more than $1-2\varepsilon$ for the policy σ'. So, $\boldsymbol{R}(\sigma', k) > \boldsymbol{R}(\sigma, k)$. □

4.5 Finite memory policies.

A natural generalization of M-policy is *finite memory* policy. We prove below that the class of finite memory policies contains optimal policies for the $\boldsymbol{R}^L_\infty$-criterion, when L is a regular language. Moreover, for the finite horizon, a generalization of T-policy in the same way, leads to optimal solutions.
A (deterministic) policy σ is called a *finite memory policy* or *F-policy* if there exists a finite transition system $\mathcal{T} = (\Sigma, Q, \delta, q_0)$, and a function $\sigma' : Q \to A$ such that $\forall\, W \in \Sigma^* \, (\sigma(W) = \sigma'(q_0.W))$. The *size of the memory* is the size of the transition system $\mathcal{T}$.
A (deterministic) policy σ is called *dependent only on time and finite memory* or *FT-policy* if there exists a finite transition system $\mathcal{T} = (\Sigma, Q, \delta, q_0)$, and a function $\sigma' : Q \times \mathbb{N} \to A$ such that $\forall\, W \in \Sigma^* \, (\sigma(W) = \sigma'(q_0.W, |W|)$.

Notice that T- and FT-policies are not finite memory ones.

4.6 R^L-optimal policies.

In this section we suppose the information is perfect (i. e. continue to consider the case of bijective coloring). And we restrict ourselves to behavior constraints from a regular language represented as a deterministic automaton.
The theorems proven here say that $\boldsymbol{R}^L$-optimal policies can be found in the class of F-policies for infinite horizon, and in the class of FT-policies for finite horizon, and one can construct such a policy in polytime.

Encoding a $\boldsymbol{R}^L$ criterion into a $\boldsymbol{R}^{s,T}$ one.

Let $G = (V, A, \rho, s, T)$ be a MDP-graph with bijective coloring and fixed target T (if not then there is no T), and L a regular language over V recognizable by a finite automaton $\mathcal{A} = (V, Q, \delta, q_0, F)$. Without loss of generality we can suppose that A contains a *nil* action ω, such that for every vertex $v \in V$, $\rho(vv, \omega) = 1$. We need the following definitions to describe the mentioned reductions. A path in G is called *normal* if it belongs to sV^*. A policy on G is

a *normal policy* if its value is ω for not normal paths. Clearly if σ is an $\boldsymbol{R}^L$-optimal policy, then the unique normal policy σ_1 defined as σ on normal paths is also an $\boldsymbol{R}^L$-optimal policy. Moreover if σ is a T-policy (resp. a M-policy) then σ_1 is a T-policy (resp. a M-policy).
To reduce the problem of constructing optimal policies to the previous case of section 3 we go to MDP $\langle G, \mathcal{A}\rangle$ defined as (V', A', ρ', s', T') where

- $V' = V \times Q$,
- $A' = A$,
- $\rho'((u,q),(v,q.v),\alpha) = \rho(u,v,\alpha)$,
- $s' = (s, q_0.s)$,
- $T' = T \times F$ (If not to fix the target set, we do not define T').

Thus an edge of MDP G from u to v that the system follows under an action α with probability $\rho(uv,\alpha)$ and a transition from q to $q.v$ give in $\langle G, \mathcal{A}\rangle$ a transition from (u,q) to $(v,q.v)$ with probability $\rho'((u,q)(v,q.v),\alpha) = \rho(uv,\alpha)$.
Now we define a mapping Φ from paths and policies of G to respectively paths and policies of $\langle G, \mathcal{A}\rangle$.
For every path $W \in V^*$, $W = w_1 \dots w_n$, $w_i \in V$, $1 \le i \le n$, denote by $\Phi(W)$ the path in V'^* such that $\Phi(W) = (w_1, p_1)(w_2, p_2)\dots(w_n, p_n)$ where $p_i = q_0.w_1 \dots w_i$, $1 \le i \le n$.
For a policy σ on the MDP-graph G we define a normal policy $\sigma' = \Phi(\sigma)$ on $\langle G, \mathcal{A}\rangle$ as follows:

$$\sigma'(W') = \begin{cases} \sigma(W) & \text{if } W' \in s'V'^* \text{ and } W' = \Phi(W), \\ \omega & \text{otherwise.} \end{cases}$$

It is clear that

$$(W' = \Phi(w) \;\&\; last.W' = (u,q)) \;\Rightarrow\; q_0.W = q.$$

That means that the policy σ' simulates the policy σ on the path W in the graph $\langle G, \mathcal{A}\rangle$ which contains in its states the information concerning the behavior of $\mathcal{A}$ on the path W in the graph G.
It is easy to verify that the mapping $\sigma \to \Phi(\sigma)$ is a one-to-one correspondence between the normal policies in G and the normal policies in $\langle G, \mathcal{A}\rangle$. Denote by $\mathcal{T}(\mathcal{A})$ the transition system $(V \times Q, V, \dot{,}, (s, q_0.s))$ where the transition function is defined by $(u,q).v = (v, q.v)$. Note that if $W \in V^*$ then $(s, q_0.s).W = (last.sW, q_0.sW)$.

Lemma 8 *Let $G = (V, D, \rho, s)$ be a MDP-graph, and $\mathcal{A}$ a finite automaton recognizing a regular language L. The following property holds for the MDP-graph $\langle G, \mathcal{A}\rangle$:*

if σ' is a normal $\boldsymbol{R}_k^{s',T'}$-optimal (resp. $\boldsymbol{R}_\infty$-optimal) T-policy (resp. M-policy) on $\langle G, \mathcal{A}\rangle$ then $\Phi^{-1}(\sigma')$ is a normal ${\boldsymbol{R}^L}_k$-optimal (resp. $\boldsymbol{R}^L_\infty$-optimal) FT-policy (resp. F-policy) on G for the finite transition system $\mathcal{T}(\mathcal{A})$.

Proof. Let σ' be a normal M-strategy on $\langle G, \mathcal{A}\rangle$ and $\sigma = \Phi^{-1}(\sigma')$. The strategy σ is also normal and

$$\forall\, W \in sV^*\, (\sigma(W) = \sigma'(\Phi(W))).$$

Denote by σ'' the mapping from V'^* to A' such that $\sigma''(W') = \sigma'(last.W')$ for all $W' \in V'^*$. This σ'' maps the set of states of the transition system $\mathcal{T}(\mathcal{A})$ into A'. Since $(s, q_0.s)$ is the initial state of $\mathcal{T}(\mathcal{A})$ and

$$\forall\, W \in sV^*\, (\sigma(W) = \sigma''((s, q_0.s).W))$$

we conclude that σ is a normal F-strategy with transition system $\mathcal{T}(\mathcal{A})$.

Along the same lines one can show that $\Phi^{-1}(\sigma')$ is a normal FT-strategy if σ' is a normal T-strategy. $\square$

Lemma 9 *Let $G = (V, D, \rho, s)$ be a MDP-graph, and $\mathcal{A}$ a finite automaton recognizing a regular language L. The following properties holds for the MDP-graph $\langle G, \mathcal{A}\rangle$:*

(i) $\boldsymbol{R}^L(\sigma, k) = \boldsymbol{R}(\Phi(\sigma), k)$ and $\boldsymbol{R}^L_\infty(\sigma) = \boldsymbol{R}_\infty(\Phi(\sigma))$ for every normal strategy σ on G;

(i) (ii) if σ' is a normal $\boldsymbol{R}$-optimal (resp. $\boldsymbol{R}_\infty$-optimal) T-strategy (resp. M-strategy) on $\langle G, \mathcal{A}\rangle$ then $\Phi^{-1}(\sigma')$ is a normal $\boldsymbol{R}^L$-optimal (resp. $\boldsymbol{R}^L_\infty$-optimal) FT-strategy (resp. F-strategy) on G with a finite transition system $\mathcal{T}(\mathcal{A})$.

Proof. Let σ be a strategy on G.
We have:

$$\begin{aligned}\boldsymbol{R}^L(\sigma, k) &= \sum_{W \in \mathcal{P}^{s,T}_{\leq k}(G,L)} \mathbf{Prob}(W \text{ is a L-realization of } \sigma)\\ &= \sum_{W' \in \mathcal{P}^{s',T'}_{\leq k}(G',L)} \mathbf{Prob}(W' \text{ is a realization of } \Phi(\sigma))\\ &= \boldsymbol{R}(\Phi(\sigma), k) \qquad (33)\end{aligned}$$

Equality (33) implies that:

$$\boldsymbol{R}^L{}_\infty(\sigma) = \boldsymbol{R}^L(\sigma) = \sup_k \boldsymbol{R}^L(\sigma, k) = \sup_k R(\Phi(\sigma), k) = \boldsymbol{R}(\Phi(\sigma)). \qquad (34)$$

Since Φ is a one-to-one correspondence between the normal strategies on G and the normal strategies on $\Phi(G)$, we deduce (i) using (33) and (34).
Let σ'_k be a normal $\boldsymbol{R}$-optimal T-strategy on G' for a given k. Then Lemma 8 implies that $\Phi^{-1}(\sigma'_k)$ is a normal $\boldsymbol{FT}$-strategy on G for the same k with transition system $\mathcal{T}(\mathcal{A})$. Furthermore, by (i), the strategy $\Phi^{-1}(\sigma'_k)$ is a $\boldsymbol{R}^L$-optimal $\boldsymbol{FT}$-strategy. In the same way, if σ' is a $\boldsymbol{R}$-optimal T-strategy on G', then $\Phi^{-1}(\sigma')$ is a $\boldsymbol{R}^L$-optimal T-strategy on G with finite transition system $\mathcal{T}(\mathcal{A})$. □

The following 2 theorems are immediate consequences of Lemmas 8-9.

Theorem 4 *For every MDP-graph with perfect information and every regular language L a policy $\boldsymbol{R}^L$-optimal for a given $k \in \mathbb{N}$ does exist among FT-policies, and such a FT-policy can be constructed in time polynomial in k, in the size of the MDP-graph and in the size of a deterministic automaton defining L.*

Theorem 5 *For every MDP-graph with perfect information and every regular language L $\boldsymbol{R}^L_\infty$-optimal policy does exist among F-policies, and such an F-policy can be constructed in time polynomial in the size of the MDP-graph and in the size of a deterministic automaton defining L.*

4.7 Limits of finite memory policies

We come back to the partial information case. We prove that, opposite to the perfect information situation, in the case when there exists a $\boldsymbol{R}^r_\infty$-optimal policy, maybe no finite memory policy is optimal.
We start with the MDP-graph G shown on the Figure 14). Its formal description is a sfollows.

- The set of vertices is $V = \{s_0,\ s_1,\ s_2,\ s_3,\ s'_1,\ s'_2,\ s'_3,\ t,\ trap\}$.
- The set of actions is $D = \{d,\ \bar{d}\}$.
- The transition probability function ρ is defined as:
  $\rho(s_0, s_1, d) = \rho(s_0, s'_1, d) = \rho(s_0, s_1, \bar{d}) = \rho(s_0, s'_1, \bar{d}) = 1/2$.
  For $x_i \in \{s_2, s_3, s'_1, s'_3\}$:

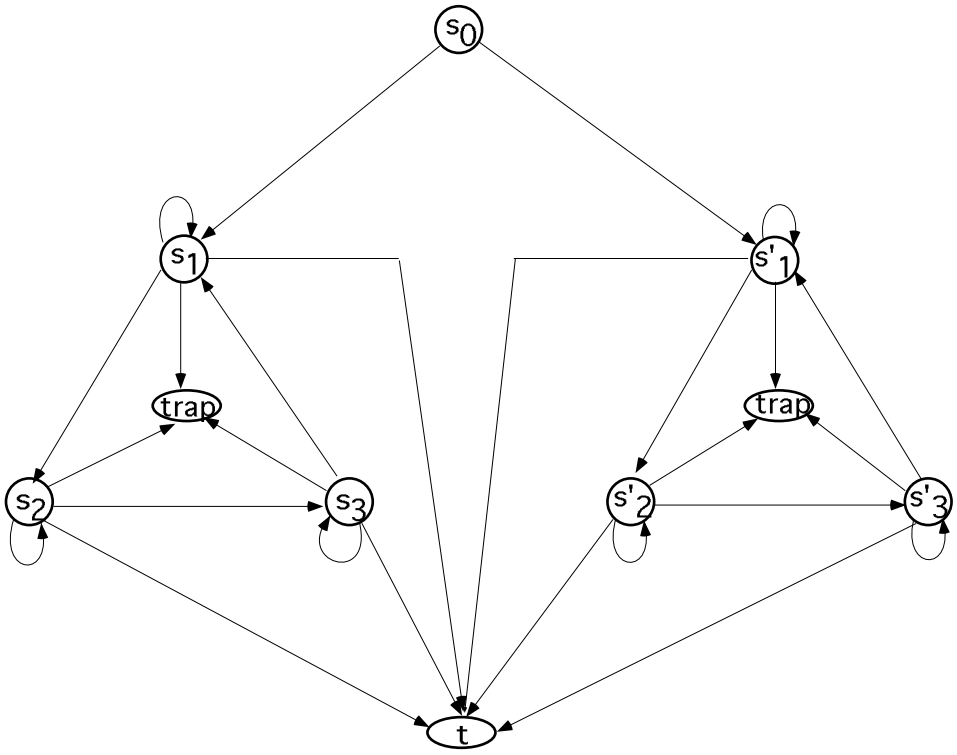

Figure 14: The graph G.

$\rho(x_i,\ x_{i+1}, \lambda) = \frac{1-\alpha}{2},\ \rho(x_i,\ x_i, \lambda) = \alpha/2,\ \rho(x_i,\ t, \lambda) = 1/2, \quad \lambda \in \{d, \bar{d}\}$
(the sum of indices is mod 3).
$\rho(s_1, s_1, d) = \alpha/2, \quad \rho(s_1, trap, d) = 1/2 - \beta,$
$\rho(s_1, s_2, d) = \frac{1-\alpha}{2}\ \rho(s_1, t, d) = \beta,$
$\rho(s_1, s_1, \bar{d}) = \alpha/2,\ \ \rho(s_1, trap, d) = \beta,$
$\rho(s_1, s_2, \bar{d}) = \frac{1-\alpha}{2}\rho(s_1, t, d) = 1/2 - \beta,$
$\rho(s'_2, s'_2, d) = \alpha/2, \quad \rho(s'_2, trap, d) = \beta,$
$\rho(s'_2, s'_3, d) = \frac{1-\alpha}{2}\ \rho(s'_2, t, d) = 1/2 - \beta,$
$\rho(s'_2, s'_2, \bar{d}) = \alpha/2,\ \ \rho(s_1, trap, d) = 1/2 - \beta,$
$\rho(s'_2, s'_3, \bar{d}) = \frac{1-\alpha}{2}\ \rho(s'_2, t, d) = \beta.$

- Vertices $s_1, s_2, s_3, s'_1, s'_2, s'_3$ have the same color, we will call this color c.

After one step, the state of the system has a color c, and while state t or state $trap$ is not reached, the system remains in a state with color c. So, at step $n+1$, a policy σ acts on the word $clr(s_0)c^n$, and it can be considered as a T-policy.
Denote by $p_n(s_i)$ (resp. $p_n(s'_i)$) the probability to be in state s_i (resp. s'_i) after n steps applying some policy σ, and knowing that the system is in a state with color c. Whatever is σ we have: $p_n(s_i) = p_n(s'_i) = p_{i,n}$ for $i = 1,\ 2,\ 3$ with

$$\begin{pmatrix} p_{1,n} \\ p_{2,n} \\ p_{3,n} \end{pmatrix} = \frac{1}{2} M^n \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \text{where} \quad M = \begin{pmatrix} \alpha & 0 & 1-\alpha \\ 1-\alpha & \alpha & 0 \\ 0 & 1-\alpha & \alpha \end{pmatrix}. \tag{35}$$

So at step $n,\ n \geq 1$, if the system is in a state with color c, the probability to reach t at the next step depends only on the action made at this moment, because the probability distribution at

this moment is independant of σ, and this probability to reach t at the next step is:
$p_n(s_1)\beta + p_n(s'_2)(1/2-\beta) + 1/2(p_n(s_2)+p_n(s_3)+p_n(s'_1)+p_n(s'_3))$ if action d is made,
$p_n(s_1)(1/2-\beta) + p_n(s'_2)\beta + 1/2(p_n(s_2)+p_n(s_3)+p_n(s'_1)+p_n(s'_3))$ if action $\bar{d}$ is made.

Thus at step n the best action is d if

$$p_n(s_1)\beta + p_n(s'_2)(1/2-\beta) > p_n(s_1)(1/2-\beta) + p_n(s'_2)\beta$$

which is equivalent to

$$p_n(s_1) > p_n(s'_2) \tag{36}$$

if $\beta < 1/4$. Using M, a computation of the probabilities $p_k(s_i)$ shows that (36) is equivalent to :

$$\cos(n\theta + \frac{\pi}{6}) > 0. \tag{37}$$

where θ is the argument of $\lambda = \alpha + (1-\alpha)\zeta$, with $\zeta = \mathrm{e}^{\frac{2\pi i}{3}}$, where λ is one of the eigenvalues of M, the others are 1 and $\bar{\lambda}$.
The definition of λ leads to the equality:

$$\cos\theta = \frac{1-3\alpha}{2\sqrt{1-3\alpha+3\alpha^2}}\ . \tag{38}$$

One can choose a rational α sufficiently small, as required by our reasoning, and such that θ is not a rational multiple of π.

Now let α be a rational number such that (38) and θ is not a rational multiple of π. Denote by D_k the boolean value of (37).

Lemma 10 *The sequence $\{D_k\}_{k\in\mathbb{N}}$ is not ultimately periodic.*

Proof. Suppose that the sequence $\{D_k\}_{k\in\mathbb{N}}$ is ultimately periodic with period T, and the periodicity begins at k_0. Suppose, without loss of generality, that $\cos(k_0\theta + \frac{\pi}{6}) \in [0,1]$. Then, denoting $\eta_0 = k_0\theta + \frac{\pi}{6}$ and $\eta = T\theta$ we have $\cos(\eta_0 + i\eta) \in [0,1]$ for all $i \in \mathbb{N}$. But η is not commensurable with π, and hence $\{i\eta\}_{i\in\mathbb{N}} \pmod{2\pi}$ is dense in $[0, 2\pi]$. Thus, the sequence $\{\cos(\eta_0 + i\eta)\}_i$ is dense in $[-1,1]$, a contradiction. □

Notice that there is a unique (except for the first step where, d or $\bar{d}$ can be chosen arbitrarily) $\boldsymbol{R}^{s,t}_{\infty}$-optimal policy τ defined bythe equality

$$\tau(clr(s_0)c^n) = \textbf{If } D_n \textbf{ Then } d \textbf{ Else } \bar{d}.$$

Moreover, we know that the boolean value D_n of (37) is not ultimately periodic (Lemma 10). But a T-policy which has a finite memory is exactly an ultimately periodic function. Therefore, for every finite memory policy σ there exists infinitely many n such that $\sigma(clr(s_0)c^n) \neq \tau(clr(s_0)c^n)$. So every policy σ which is ultimately periodic satisfies the following inequality:

$$\boldsymbol{R}^{s,t}_{\infty}(\sigma) < \boldsymbol{R}^{s,t}_{\infty}(\tau).$$

We can claim

Theorem 6 *For the MDP-graph G defined above, there exists a $\boldsymbol{R}^{s,t}_{\infty}$-optimal policy τ, and for every finite memory policy σ*

$$\boldsymbol{R}^{s,t}_{\infty}(\sigma) < \boldsymbol{R}^{s,t}_{\infty}(\tau).$$

Since $\boldsymbol{R}^{s,t}$ criterion is a particular case of both $\boldsymbol{R}^{r}$ and $\boldsymbol{R}^{L}$ ones we have:

Corollary 1 *For $\boldsymbol{R}^{r}$ criterion as well as $\boldsymbol{R}^{L}$ one, in the case of partial information and infinite horizon, in general, no finite memory policy is optimal.*

5 Randomized policies

We consider a more rich class of policies, namely, randomized policies. As many algorithmic problems related to constructing optimal deterministic policies are computationally hard, one can hope that similar problems for randomized policies will be simpler.

5.1 Definitions.

A *randomized policy* is a random function of the same type as deterministic policy. Such a function τ can be represented by the function Λ^τ that gives a distribution of probabilities to choose this or that action: $\Lambda^\tau : C^+ \times A \to [0,1]$, such that for all $W \in C^+$ $\sum_{\alpha \in A} \Lambda^\tau(W,\alpha) = 1$ for all $W \in C^+$.

The distributions $\mathbf{B}^\tau(v_1 \dots v_{k-1}v_k, \lambda_1 \dots \lambda_{k-1})$ generated by a randomized policy τ are defined similarly to the deterministic case:

$$\mathbf{B}^\tau(v_1 \dots v_{k-1}v_k) = s(v_1) \cdot \prod_{i=1}^{k-1} (\sum_{\alpha \in A} \Lambda^\tau(clr(V_i),\alpha) \cdot \rho(v_i v_{i+1},\alpha)),$$

where $s(v)$ is an initial distribution.
But this distibution is not as well productive because it does not determine uniquely the actions of the strategy corresponding to a given path, and we cannot directly find the reward obtained on the path. The distribution on sequences of states and sequences of actions is defined by

$$\mathbf{B}^\tau(v_1 \dots v_{k-1}v_k, \alpha_1 \dots \alpha_{k-1}) = s(v_1) \cdot \prod_{i=1}^{k-1} \Lambda^\tau(clr(V_{1,i}),\alpha_i) \cdot \rho(v_i v_{i+1},\alpha_i)).$$

Informally speaking, $\mathbf{B}^\tau(P)$ is the probability to follow a given path P of the length k when executing a strategy τ.

5.2 Probabilistic versus Deterministic: general Policies.

In the case of perfect information it is known that randomized policies are not better than deterministic ones for the R-criterion [Kal83]. This result can be extended to the partial information case.

Theorem 7 *For every randomized policy τ, an initial distribution s and natural k there exists a deterministic policy σ such that $R^s_k(\tau) \le R^s_k(\sigma)$.*

The proof is rather straightforward but technical, and is based on considerations of convexity of R^s with respect to some kind of addition of policies.

In a similar way as for deterministic policies, we define the notion of *finite memory randomized policy.*
A randomized policy τ defined by the function $\Lambda^\tau : C^+ \times D \to [0,1]$, is a *finite memory policy* if there exists a finite transition system $\mathcal{T} = (States,\ C,\ \delta,\ p_0)$ and a function $\Lambda^{\tau'} : States \times A \to [0,1]$, such that for all $W \in C^+$

$$\Lambda^\tau(W,\alpha) = \Lambda^{\tau'}(p_0.W,\alpha) \tag{39}$$

and, for all $q \in States$,

$$\sum_{\alpha \in A} \Lambda^{\tau'}(q, \alpha) = 1. \tag{40}$$

The *size of the memory* is the size of the transition system.

5.3 Randomized versus Deterministic Finite Memory Policies.

We give here an example when finite memory randomized policies are better than the deterministic ones. Constructing examples of this type it is reasonable to take into account the complexity of MDP-graphs and of policies, in particular, the complexity of random number generators with respect to the simple uniform 0-1 Bernoulli source. Our example deals with grids, for less 'geometric' graphs one can find simpler examples.

Proposition 6 *Given an integer M, there exists a MDP-graph G_M for which the following property holds: there exists a randomized policy τ with finite memory bounded by M such that for every deterministic policy σ with finite memory bounded by M, we have $R_\infty^{s,T}(\tau) > R_\infty^{s,T}(\sigma)$.*

Proof. Consider the square grid $n \times n$, n being odd and specified later. The set of actions is $D = \{N,\ E,\ S,\ W\}$, the actions corresponding to cardinal points. One side of the square is fixed as the target. We use two colors, *black* and *white*. All the vertices are *white* except the target vertices which are *black*. We take four copies of the same grid but with four different target sides (with respect to cardinal points) with centres $o_1,\ o_2,\ o_3,\ o_4$. We add a *white* source vertex s. Define the function ρ as follows. For s, we put $\rho(s, o_i, d) = 1/4$ whatever be the action d, $i = 1, \ldots, 4$. So starting from s, after one step, we reach one of the four centres with the same probability. For every vertex u in the grids and every direction $d \in D$, if v is a d-neighbour of u then $\rho(uv, d) = 1$ otherwise $\rho(uu, d) = 1$. It means that taking action d in state u, we go to its d-neighbour with probability 1 if this d-neighbour exists otherwise we remain in state u with the same probability 1.
Denote the described graph by G.

Let σ be a finite memory deterministic policy with memory size bounded by M. Consider its behavior when it starts from the centre of one grid. While the policy sees white states its behavior is defined by a deterministic automaton without input. But the state transition diagram of such an automaton is a (directed) cycle, say Z, with a directed simple path, say Y, of vertices coming in it. Thus, once having done Y, the policy starts some periodic behavior.
Suppose that $n > M$. The sequence Y determines some displacement inside the interior of the square. After having done this displacement the policy follows the periodic pattern defined by Z. If this pattern is strictly inside the square the displacement is determined by the vector going from the initial to the end vertex of the pattern. We will call it the *displacement vector*. But when the policy reaches a white boundary, further displacements are along the projection of the vector on the boundary. On the whole the policy can reach vertices situated on at most two adjacent boundaries but without end extremities different from their joint vertex.
Now launch in G a deterministic policy σ with the memory bounded by M. Suppose, without loss of generality, that its displacement vector belongs to North-West quadrant. With probability $1/2$ it arrives in a grid where the target is either the North or West boundary. Thus σ reaches the target with probability not more than $1/2$.
On the other hand, the randomized policy τ which defines the usual uniform random walk in

the square reaches the target with probability 1. And τ has a very small constant memory even if it uses for random generation of directions the uniform zero-one Bernoulli source. ■

6 Unobservable Processes

In this section we consider the class of MDP-graphs with one color, i. e. with the set of colors consisting of one element. We will call such graphs *non-colored*. The argument of a strategy is a string of one and the same character, and hence it contains only the information on the number of executed steps in unary notation. Thus, the action of a strategy depends only on time, and we consider a strategy σ as a function $\sigma : \mathbb{N} \to A$ that may be represented also by the string of its values $d_1 d_2 \ldots$. So, this is a particular case of T-strategy. The following result was proven in [PT87] as a corollary of a more general theorem on the complexity of partially observed Markov decision processes.

Theorem 8 ([PT87], Corollary 2) *The following problem is NP-complete:*
Given *a non-colored MDP-graph with k vertices, a starting vertex s, and a set of target vertices T,*
to recognize *whether there exists a strategy σ with $R_k^{s,T}(\sigma) = 1$.*

The following theorem [BdRS96] shows that the problem of computing an optimal strategy in the case of total unobservability does not admit even very weak approximations.

Theorem 9 *The following problem is NP-hard:*
Given *a noncolored MDP-graph with k vertices, a starting vertex s and a set of target vertices T (such that $p_k^{opt}(s,T)$ equals 1 or is less than $\exp(-\sqrt{k})$),*
recognize *whether there exists a strategy σ which leads from s to T in k steps with probability not less than $\exp(-\sqrt{k})$.*

Proof. We use the notations related to 3SAT-problem from sunsection 2.6.
Given a formula F, we construct the following simple MDP-graph H_F.

- The set of actions is $D = \{1, 2, 3\}$.
- Reliable vertices are $\{t, trap,\ 1, 2, \ldots m-1\} \cup (Z \times \{1, 2, \ldots m-1\} \times \{1, 2\})$, $s = 1$.
- Random vertices are $D \times \{1, 2, \ldots m-1\}$.
- Reliable edges for action λ:
  $lbl(\lambda, t) = (t, t)$, $lbl(\lambda, trap) = (trap, trap)$, $lbl(\lambda, i) = (i, (\lambda, i))$,
  $lbl(\lambda, (z, i, a)) =$
  case 1.1: $a = 1$ & $z = \overline{z_{i+1,\lambda}} \Longrightarrow$ edge to $trap$;
  case 1.2: $a = 1$ & $z \neq \overline{z_{i+1,\lambda}} \Longrightarrow$ edge to $(z, i, 2)$;
  case 2.1: $a = 2$ & $i < m-1 \Longrightarrow$ edge to $(z, i+1, 1)$;
  case 2.2: $a = 2$ & $i = m-1 \Longrightarrow$ edge to t.
- Two random edges from a vertex (λ, i), $i < m-1$:
  an edge "right" to $i+1$ with probability $\frac{m-i-1}{m-i}$ and
  an edge "down" to $(z_{i,\lambda}, i, 1)$ with probability $\frac{1}{m-i}$.
- One edge "down" to $(z_{m-1,\lambda}, m-1, 1)$ from vertex $(\lambda, m-1)$.

For an example of graph H_F see Figure 5. **Claim.** *If the path $P = z_{1,d_1} z_{2,d_3} \ldots z_{m,d_{2m-1}}$ determined by a strategy $\sigma = d_1 d_2 \ldots d_{2m}$ is contradictory then σ traverses H_F from s to t with probability not more than $1 - \frac{1}{m-1}$.*

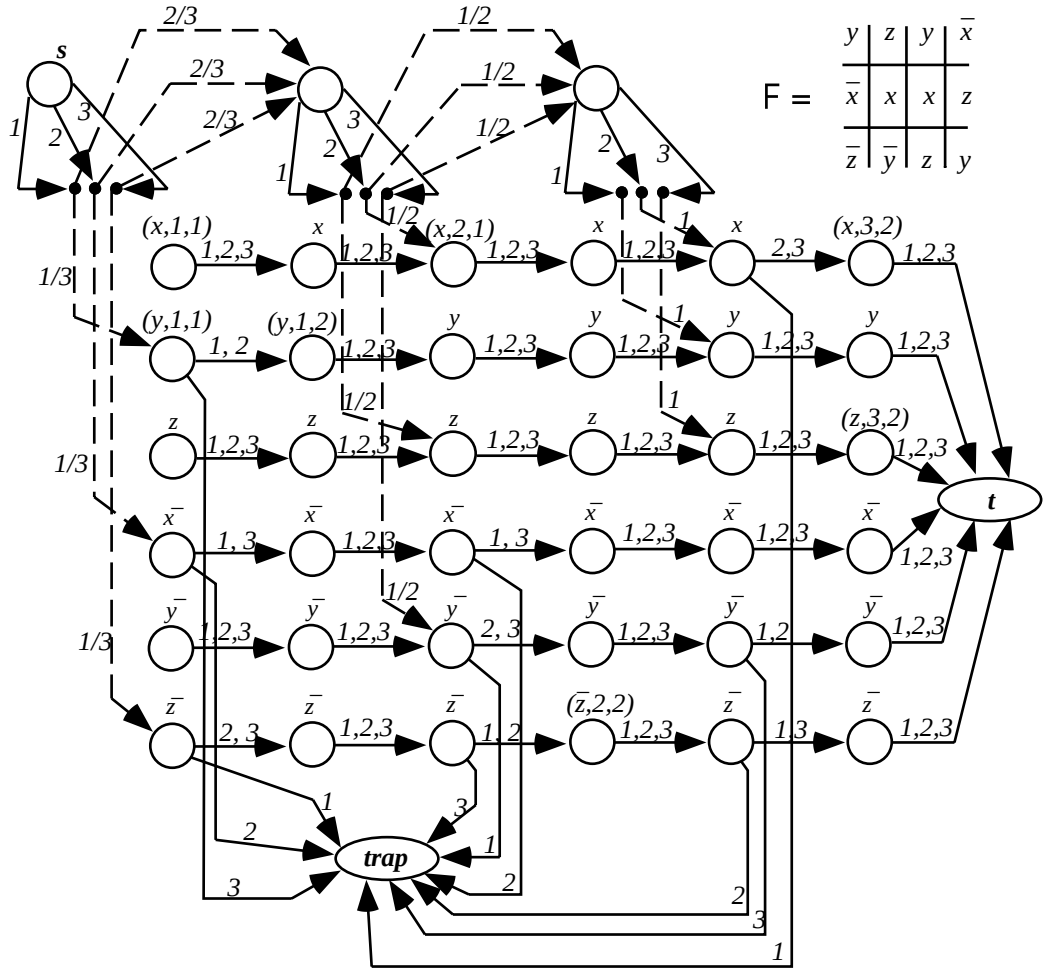

Figure 5. Some of reliable vertices of the set $Z \times \{1, 2, \ldots m\text{-}1\} \times \{1, 2\}$ are labeled by all their coordinates and others are marked by their first coordinate.

If P is open then $R^{s,\{t\}}_{2m+1}(\sigma) = 1$.

Indeed, if P is a contradictory path we have $z_{i,d_{2i-1}} = \bar{z}_{j,d_{2j-1}}$ for some $1 \le i < j \le m$. When executing σ, we follow from the vertex 1 to $D \times \{1\}$ then "right" to 2 and to $D \times \{2\}$ etc. until the first "down" move. The probability to go "down" from the $D \times \{i\}$ is exactly $\frac{1}{m-1}$. Then at the $2j-1$th step we arrive at the condition of the case 1.1 and go to *trap*. Thus, such a strategy traverses H_F successfully from s to t with probability not more than $1 - \frac{1}{m-1}$.
The proof of the second assertion of the claim is similar. □

Graph H_F contains less than $20m^2$ vertices (for m large enough). We construct now the desired graph $\hat{H}_F$ as follows. Take $20m^4$ copies of H_F denoted by $H^1_F, \ldots, H^{20m^4}_F$. We consider the vertex $1 = s$ of H^1_F as a *starting* vertex for $\hat{H}_F$ and the vertex t of $H^{20m^4}_F$ as a *target* vertex for $\hat{H}_F$ and redefine the reliable edges from vertices t of H^i_F's. We put a unique reliable edge from t of H^i_F to s of H^{i+1}_F for all $i < 20m^4$. Thus we get sequential composition of the initial graphs. Obviously, $\hat{H}_F$ has not more than $k = 20m^2 \cdot 20m^4 = (20m^3)^2$ vertices.
Consider a strategy $\sigma = d_1 \ldots d_{2m \cdot 20m^4}$ for traversing $\hat{H}_F$ from *starting* to *target*. If the paths $z_{1,d_{(2m+1)i+1}} z_{2,d_{(2m+1)i+3}} \ldots z_{m,d_{(2m+1)i+(2m-1)}}$ are contradictory for all $0 \le i < 20m^4$ then

the claim implies that σ traverses each of H_F^{i+1} with probability not greater than $1 - \frac{1}{m-1}$, and hence the total probability for σ to traverse $\hat{H}_F$ from *starting* to *target* is not more than $(1-\frac{1}{m-1})^{20m^4} < \exp(-20m^3) = \exp(-\sqrt{k})$. Thus for any strategy σ whose probability of success is not less than $\exp(-\sqrt{k})$, one of the paths $z_{1,d_{(2m+1)i+1}} z_{2,d_{(2m+1)i+3}} \dots z_{m,d_{(2m+1)i+(2m-1)}}$ is open. Since k is not more than polynomially greater than the size of F. □

7 Bounded Unobservability

7.1 Bounded Unobservability

It was shown in [PT87] than the problem of computating an optimal strategy for partially observed processes is PSPACE-complete. We consider here the partially observed processes when this uncertainty concerning observability of states is bounded by a fixed parameter citeBdRS96.

7.1.1 Graphs with fixed multiplicity of colors.

We say that a MDP-graph has a *coloring of multiplicity* m if the pre-image of each color contains not more than m vertices. That is, when the color is known, the location is determined up to not more than m vertices. Obviously, bijective coloring corresponds to multiplicity 1. As intermediate case between bijective coloring and total unobservability we consider MDP-graphs with fixed multiplicity of coloring $m > 1$. The notion of PT-strategy gives a reasonable generalization of T-strategies for this case, and proposition 1 shows that in some sense it suffices to consider PT-strategies only.

Consider the first non trivial case $m = 2$, and assume for simplicity that the set of moves D is $\{right, left\}$. For a color $v \in C$ we denote by v^+ and v^- the two vertices of this color. When traversing the graph we actually have just one 'hidden parameter' (+ or −) that influences, however, the probabilities of further transitions. Having arrived at a color v after k steps a PT-strategy σ makes its next action basing it on k and on the probabilities p^+ and p^-, $p^+ + p^- = 1$, of being at v^+ and v^- respectively. Thus, σ induces a partition of $[0, 1]$ into two sets L and R such that σ goes $right$ if $p^+ \in R$ and goes $left$ otherwise. One might expect that if it is more profitable to go $right$ from v^+ and to go $left$ from v^- then there should exist some boundary probability p_0 such that if $p^+ \geq p_0$ then it is better to go $right$, and if $p^+ \leq p_0$ then it is better to go $left$. But this is not the case. In fact, the sets R and L may contain exponentially many (on k) intervals that alternate.

7.1.2 Complexity of optimization.

The following theorem shows that computing an optimal strategy for graphs with small multiplicity of colors is NP-hard.

Theorem 10 *Every optimal universal strategy for the class of MDP-graphs with coloring of multiplicity* 3 *and the class of* $R_k^{s,T}$*-criteria,* $k \in \mathbb{N}$*, is universal for NP (with respect to polytime Turing reducibility). In simpler words, constructing an optimal strategy for MDP-graphs with multiplicity of coloring* 3 *is NP-hard.*

It is an interesting open question related to Max Word Problem (see subsection 7.1.3) whether the theorem holds for multiplicity 2 and/or for a class of MDP-graphs containing only one graph.

We can reformulate theorem 10 as NP-hardness of recognizing whether there exists a strategy with probability of success not less than a given parameter.

However, contrary to the case of total uncertainty, the problem of computing an optimal strategy for graphs with small multiplicity of colors does admit a reasonable polytime approximation.

A universal strategy σ is said to be ϵ-*optimal* if it is optimal up to an additive error ε, i. e.

$$R_k^{s,T}(\sigma_{G,(k,s,T)}) \geq R_k^{s,T}(\zeta) - \varepsilon$$

for all $G,\ k,\ s,\ T$ and for all strategies ζ.

We can consider the property to be ϵ-optimal as a criterion with the value 1 on ϵ-optimal strategies and 0 otherwise.

Theorem 11 *There exists an optimal universal strategy σ with respect to the criterion of ϵ-optimality such that for the class of MDP-graphs with a fixed multiplicity of coloring m it is computable in time polynomial on the size of input graphs and $1/\epsilon$. In particular, this means that for a fixed multiplicity of colors optimal strategies admit polytime approximations with an additive error.*

Theorem 11 is interesting in the context of Theorem 10, taken as itself it may seem very natural.

The proofs of Theorems 11 and 10 are given in subsections 7.1.4 and 7.1.7.

7.1.3 Relations with Max Word Problem for Stochastic Matrices.

Recall that Max Word Problem for stochastic matrices is the following one. Given a set $S = \{M_i\}_{1\leq i\leq n}$ of stochastic $m\times m$-matrices with rational entries, $M_i = (M^i_{\alpha\beta})$, $\sum_\alpha M^i_{\alpha\beta} = 1$, two (row) vectors V, W with positive coordinates and an integer k in unary notation, the problem is to find a sequence $M_{i_1}, \ldots, M_{i_k}$ which maximizes the product $\langle V, (\prod_{j=1}^k W M_{i_j})\rangle$.

It was shown in [Con91] that the Max Word Problem for stochastic matrices is NP-hard as well as its approximation version up to any multiplicative factor.

Max Word Problem for stochastic $m \times m$-matrices can be reduced to the problem of constructing an optimal strategy for MDP-graphs with coloring of multiplicity m (see (i) below in this subsection). Together with theorem 11 this implies that for every fixed m Max Word Problem for stochastic $m \times m$-matrices admits polytime approximations with every additive precision.

The problem of constructing an optimal strategy for MDP-graphs with one color can be straightforwardly reduced to Max Word Problem for stochastic matrices (see (ii) below in this subsection). With theorem 9 this implies that Max Word Problem for stochastic matrices does not admit polytime approximations within additive precision $\exp(-\sqrt{k})$.

The reductions mentioned above are described as follows.

(i) For an input $M_i = (M^i_{\alpha\beta})$, $1 \leq i \leq n$, $V = (v_\alpha)$, $W = (w_\beta)$, $1 \leq \alpha, \beta \leq m$ and k of Max Word Problem for stochastic $m \times m$-matrices we build a MDP-graph with vertices $\boldsymbol{s}$, $\{v_{i,\alpha}\}_{1\leq i\leq k+1, 1\leq\alpha\leq m}$, $\boldsymbol{t}$ and $\boldsymbol{trap}$, and with the set of actions $\{1, \ldots, n\}$. Every action leads from $\boldsymbol{s}$ to $v_{1,\alpha}$ with the probability $\frac{w_\alpha}{\sum_{1\leq\beta\leq m} w_\beta}$ and from $v_{k+1,\beta}$ to $\boldsymbol{t}$ with the probability $\frac{v_\beta}{\sum_{1\leq\beta\leq m} v_\beta}$. An action i leads from $v_{j,\alpha}$ to $v_{j+1,\beta}$ with the probability $M^i_{\alpha\beta}$.

A simple consideration shows that the probability of success of a strategy σ which makes the actions $(i_0 i_1 \ldots i_k i_{k+1})$ is

$$\frac{1}{\sum_{1\leq\beta\leq m} w_\beta} \cdot \langle V, W(\prod_{j=1}^{k} M_{i_j})\rangle.$$

(ii) For a MDP-graph with one color and the set of vertices $\{v_1 = \boldsymbol{s}, v_2, \dots, v_m = \boldsymbol{t}\}$ and the set of actions $\{d_1, \dots, d_n\}$ the problem of computing an optimal strategy to reach $\boldsymbol{t}$ from $\boldsymbol{s}$ in k steps is equivalent to the Max Word Problem for stochastic $m \times m$-matrices with the input $M_i = (\mu(d_i, v_\alpha v_\beta))_{\alpha,\beta}$, $W = (1, 0, \dots, 0)$, $V = (0, \dots, 0, 1)$, k.

Remark. Approximabilities with additive and multiplicative errors are equivalent unless the the value of an optimization problem under consideration is more than polynomially large or small. So, this difference occurs when either the value under approximation or its inverse are too small.

7.1.4 Proof of theorem 11.

The proof shows that the partially observed problem is smooth enough, and it may look tedious as compared with the underlying ideas which are usual in the theory of Markov decision processes.

Enumerate the vertices of the graph G by 2 indices i and α such that the first one is a color, so $V = \{v_{i,\alpha} : \ i = 1, \dots, n, \quad \alpha = 1, \dots, m\}$.

We supply $\boldsymbol{R}^m$ with l^1 metric $\|(x_i)_i - (y_i)_i\| = \sum_i |x_i - y_i|$, and will consider Lipschits property with respect to this metric.

A point of the simplex S (in $\boldsymbol{R}^m$) defined by the inequalities:

$$\sum_{i=1}^{m} x_i = 1, \quad x_i \geq 0$$

can be treated as a distribution of probabilities over the set $\{v_{i,1}, \dots, v_{i,m}\}$ of vertices of color i.

Let $P^i = (p^i_1, \dots, p^i_m)$ be this probability distribution, i. e. $P^i(v_{jk}) = p_k \delta_{ij}$, where δ_{ij} is Kronecker's delta.

Let $F_{N,i}(P^i)$ be the probability to reach T starting with the distribution P^i in not more than N steps by an optimal strategy.

Lemma 11 *All $F_{N,i}$ are Lipschits-1 functions, i. e. $|F_{N,i}(P) - F_{N,i}(Q)| \leq \|P - Q\|$ for $P, Q \in S$.*

Proof. Extend the functions $F_{N,k}$ onto the points

$$P \in \tilde{S} = \{(p_1, \dots, p_m) : \ \textstyle\sum_{i=1}^m p_i \leq 1 \ \& \ p_i \geq 0\}$$

in the following way. We append a new trap to our graph, and treat $P^k \in \tilde{S}$ as the probability distribution of being at v_{jl} with the probability $p_l \delta_{kj}$ and at the new trap with the rest probability $1 - \sum_{i=1}^m p_i$. Now the function $F_{N,k}$ is defined in all the points of the simplex $\tilde{S}$, again as the optimal probability to reach the target starting with the distribution P^k.

To verify the Lipschitz property of $F_{N,k}$ consider 2 points $P, Q \in S$. Let $d = \|P - Q\|$. Then for some vectors $A_i = (0, \dots, 0, a_i, 0, \dots, 0)$ with the only non zero ith coordinate we have $\sum |a_i| = d$, $Q = P + \sum_{1 \leq i \leq m} A_i$. It suffices to check $|F_{N,k}(P + A) - F_{N,k}(P)| \leq a$ where $A = (0, \dots, 0, a, 0, \dots, 0)$, $a > 0$ occupies the ith coordinate of A and $P, P + A \in \tilde{S}$.

It is clear that $F_{N,k}(P + A) - F_{N,k}(P) \geq 0$. Reasoning by contradiction assume that

$$|F_{N,k}(P + A) - F_{N,k}(P)| > a.$$

Then for some strategy σ we have $R_N^{(P+A)^k}(\sigma) > F_{N,k}(P) + a$. On the other hand (recall that $\mathcal{P}^k(T)$ is the set of all k-vertex paths containing a vertex from T),

$$R_N^{(P+A)^k}(\sigma) = \sum_{w_1 \dots w_N \in \mathcal{P}^N(T)} (P + A)^k(w_1) \cdot p^\sigma(w_1 w_2 \dots w_N)$$

$$= \sum_{w_1 \dots w_N \in \mathcal{P}^N(T)} P^k(w_1) \cdot p^\sigma(w_1 w_2 \dots w_N) + \sum_{v_{ki} w_2 \dots w_N \in \mathcal{P}^N(T)} a \cdot p^\sigma(v_{ki} w_2 \dots w_N)$$

$$= R_N^{P^k}(\sigma) + \sum_{v_{ki} w_2 \dots w_N \in \mathcal{P}^N(T)} a \cdot p^\sigma(v_{ki} w_2 \dots w_N) \leq F_{N,k}(P) + a$$

that is a contradiction. ■

The family of functions $F_{N,i}$ satisfies the following recurrent system of equations

$$F_{N,i}(p_1, \dots, p_m) = \max_{d \in D} \sum_{j=1}^{n} q_j^{i,d}(P) \cdot F_{N-1,j}(T_j^{i,d}(P)), \tag{41}$$

where $q_j^{i,d}(P)$ is the probability to arrive at the color j starting with distribution P^i by the action d and $T_j^{i,d}(P)$ is the conditional distribution on the vertices of the color j if this color has been observed after the move d from the distribution P^i. More formally,

$$q_j^{i,d}(P) = \sum_{\alpha=1}^{m} \sum_{\beta=1}^{m} p_\alpha \cdot \mu(d, v_{i,\alpha}, v_{j,\beta}) \tag{42}$$

and

$$(T_j^{i,d}(P))_h = \frac{1}{q_j^{i,d}(P)} \cdot \sum_{\alpha=1}^{m} p_\alpha \cdot \mu(d, v_{i,\alpha}, v_{j,h}). \tag{43}$$

Notice that $q_j^{i,d}(P) \geq 0$ and

$$\sum_{j=1}^{m} q_j^{i,d}(P) = 1. \tag{44}$$

7.1.5

Let $\delta = \frac{\epsilon}{2K}$, where K is the number of steps and ϵ is a chosen precision. Let M be the smallest integer greater than $\frac{1}{\delta}$.

We subdivide S into M^{m-1} equal simplices by hyperplanes parallel to the faces of S.

Consider the class $\mathcal{F}$ of continuous functions on S whose restriction onto every tiny simplex of our partition is linear.

For a function f we denote by f^* the unique function from $\mathcal{F}$ that coincides with f on all vertices of the simplices of the partition. Clear that $\sup_S |f - f^*| \leq \delta$ for every Lipschitz-1 function f.

7.1.6 Algorithm

For constructing our strategy we, firstly, define recursively the functions $\tilde{F}_{N,i} : S \longrightarrow \boldsymbol{R}^m$, $N \geq 0$, and $d_{N,i} : S \longrightarrow D$, $n \geq 1$.

$N = 0$. $\tilde{F}_{N,i}(P) = P^i(T \cap \{v_{i,1}, \dots, v_{i,m}\})$ where $P(\emptyset) = 0$.

$N > 0$. Put

$$\hat{F}_{N,i}(p_1, \dots, p_m) = \max_{d \in D} \sum_{j=1}^{n} q_j^{i,d}(P) \cdot \tilde{F}_{N-1,j}(T_j^{i,d}(P)), \quad \tilde{F}_{N,i} = \hat{F}_{N,i}^* \tag{45}$$

and put $d_{N,i}(P)$ to be an element of D maximizing the righthand side of (45).

Before the description of the desired strategy σ we prove Claims 1-3.

Claim 1 *$F_{0,k} = \tilde{F}_{0,k}$ for all k.*

Proof. By the definition. ■

Claim 2 *$|F_{1,k} - \tilde{F}_{1,k}| \le \delta$ for all k.*

Proof. We have $\hat{F}_{1,k} = F_{1,k}$ because the both functions are defined by the same equations as given by Claim 1. Hence $\tilde{F}_{1,k} = F^*_{1,k}$, and thus $|F_{1,k} - \tilde{F}_{1,k}| \le \delta$ since $F_{1,k}$ is Lipschitz-1 (Lemma 11). ■

Claim 3 *$|F_{N,k} - \tilde{F}_{N,k}| \le N\delta$ for all k, N.*

Proof. Induction on N. As the base of the induction we use Claim 1. Suppose the inequalities are valid for $N-1$:

$$|F_{N-1,k} - \tilde{F}_{N-1,k}| \le (N-1)\delta. \tag{46}$$

Consider a point $P = (p_1, \ldots, p_m)$. The inequality (46) implies that for some ζ

$$\tilde{F}_{N-1,k}(P) = F_{N-1,k}(P) + \zeta, \quad |\zeta| \le (N-1)\delta \tag{47}$$

By definition, we have

$$F_{N,i}(P) = \max_{d\in D} \sum_{j=1}^{n} q_j^{i,d}(P) \cdot F_{N-1,j}(T_j^{i,d}(P)), \tag{48}$$

$$\hat{F}_{N,i}(P) = \max_{d\in D} \sum_{j=1}^{n} q_j^{i,d}(P) \cdot \tilde{F}_{N-1,j}(T_j^{i,d}(P)). \tag{49}$$

From these equations and (47) we get

$$|\hat{F}_{N,i}(P) - F_{N,i}(P)|$$

$$= |\max_{d\in D} \sum_{j=1}^{n} q_j^{i,d}(P) \cdot (F_{N-1,j}(T_j^{i,d}(P)) + \zeta) - F_{N,i}(P)|$$

$$= |\max_{d\in D}\{\sum_{j=1}^{n} q_j^{i,d}(P)\cdot F_{N-1,j}(T_j^{i,d}(P)) + \sum_{j=1}^{n} q_j^{i,d}(P)\cdot\zeta\} - \max_{d\in D}\sum_{j=1}^{n} q_j^{i,d}(P)\cdot F_{N-1,j}(T_j^{i,d}(P))| \tag{50}$$

$$\le |\zeta \cdot \sum_{j=1}^{n} q_j^{i,d}(P)| \le (N-1)\delta \tag{51}$$

since the coefficients $q_j^{i,d}(P)$ are non negative with the sum equal to 1, see (44). Hence,

$$|F_{N,k} - \hat{F}_{N,k}| \le (N-1)\delta. \tag{52}$$

The point $P = (p_1, \ldots, p_m)$ lies in a tiny simplex of our partition, let it be a simplex with vertices $X_1, \ldots, X_m$. Then $P = \sum_{1 \le i \le m} \beta_i \cdot X_i$ for some non negative β_i, $\sum \beta_i = 1$.

Since $F_{N,k}$ is Lipschitz-1 and the diameter of our tiny simplex is not greater than δ we have $|F_{N,k}(P) - F_{N,k}(X_i)| \le \delta$ for all i. Adding these inequalities with the coefficients β_i we get

$$|F_{N,k}(P) - \sum \beta_i \cdot F_{N,k}(X_i)| \le \delta. \tag{53}$$

On the other hand,

$$\tilde{F}_{N,k}(P) = \hat{F}^*_{N,k}(P) = \sum \beta_i \cdot \hat{F}_{N,k}(X_i). \tag{54}$$

Together with (53) and (52) this gives the required inequality $|F_{N,k}(P) - \tilde{F}_{N,k}(P)| \le N\delta$, since the coefficients β_i are non negative with the sum equal to one. ■

Now we describe our *strategy* σ. Firstly, it computes and stores all the functions $\tilde{F}_{N,i}$, $0 \le N \le K$, $1 \le i \le m$ as tables of their values at the vertices of our partition. This can be done in polytime. After that for every $P \in S$ the value of the function $d_{N,i}(P)$ is computed in polytime due to (45) by trying all the $d \in D$. For a string of colors $W = c_1 \ldots c_N$ the strategy computes the probability distribution of being at vertices of the color c_N. This distribution is represented as a point P of S. Then the action to make is defined by $\sigma(W) = d_{N,c_N}(P)$.

Claim 4 *$|R_N^{P^i}(\sigma) - \tilde{F}_{N,i}(P)| \le N\delta$ for all i.*

Proof. Similar to the proof of Claim 3 using the fact that $R_N^{P^i}$ is Lipschitz-1 on the argument P that can be shown as in lemma 11. ■

The latter Claim 4 together with Claim 3 immediately imply

$$\boldsymbol{R}_K^{P^i}(\sigma) \ge F_{K,i}(P^i) - \epsilon$$

that completes the proof of Theorem 11.

7.1.7 Proof of theorem 10.

Our proof is based on a reduction of the *Partition Problem* [GJ79], A3.2: *Given* a set $\{z_a\}_{a \in A}$ of natural numbers indexed by natural numbers from A, *to find* whether there exists a subset $A' \subset A$ such that $\sum_{a \in A'} z_a = \sum_{a \in A \setminus A'} z_a$. If such a subset A' exists we say that the instance of the problem *admits* a partition.

As in the proof of Theorem 11 we treat the distributions of probabilities as points of the approriate simplex and vice versa.

For a given instance of the Partition Problem represented by a set $\{z_a\}_{a \in A}$ we construct a MDP-graph G in the following way.

Let $k = |A|$, $p = \sum_{i \in A} z_i$ and $\alpha_i = \frac{\pi z_i}{p}$. Without loss of generality we can assume that $\alpha_i < \frac{\pi}{2}$. Denote by $\hat{R}^i$ the matrix of rotation in $\boldsymbol{R}^3$ with the axis $x = y = z$ and the angle α_i, and by $\hat{H}^i$ the 3×3-matrix with the eigenvectors $(1,1,1)$, $(1,0,-1)$ and $(1,-2,1)$ and the eigenvalues 1, $e^{-c\alpha_i}$ and $e^{-c\alpha_i}$, where c is a constant that guarantees the elements of the matrices $\hat{M}^i$ defined below being positive. (Recall that the positiveness of elements of a matrix M is equivalent to that M maps the positive quadrant into itself.)

Let $\hat{M}^i = \hat{R}^i \cdot \hat{H}^i = \hat{H}^i \cdot \hat{R}^i$.

The graph G is constituted by (the notations are of the same type as in the proof of Theorem 11):

• the vertices: $V = \{v_{i,\alpha} : \ i = 1, \ldots, k+1, \quad \alpha = 1, 2, 3\} \cup \{\boldsymbol{t}\} \cup \{\boldsymbol{trap}\}$, $\boldsymbol{s} = v_{1,1}$;

• the edges go from every vertex $v_{i,\alpha}$ to all vertices $v_{i+1,\alpha}$ with exception of the last layer with $i = k+1$ from where there are edges to both $\boldsymbol{t}$ and $\boldsymbol{trap}$;

• the set of actions: $D = \{skip, take\}$;

• the function of deviations:

$$\mu(skip, v_{i,\alpha}v_{i+1,\alpha}) = 1,\ \mu(take, v_{i,\alpha}v_{i+1,\beta}) = \hat{M}^i_{\alpha\beta},\ i \neq k+1,$$
$$\mu(skip, v_{k+1,\alpha}\boldsymbol{t}) = \mu(take, v_{k+1,\alpha}\boldsymbol{t}) = l_\alpha,$$
$$\mu(skip, v_{k+1,\alpha}\boldsymbol{trap}) = \mu(take, v_{k+1,\alpha}\boldsymbol{trap}) = 1 - l_\alpha,$$

where l_α will be chosen later.

7.1.8

For every realization of a strategy σ up to the $(k+1)$th step the observed sequence of colors is $1, 2, \ldots, k+1$, so a strategy is determined by a sequence of its actions $d_1 \ldots d_k$ since the last action does not matter.

After k steps of executing σ the probability distribution of being in vertices $v_{k+1,\alpha}$ is P^{k+1} where $P = \prod_{d_i = take}(1,0,0)\hat{M}^i$. (Recall that we continue to use the notations for P^{k+1} of the previous proof.)

To deal with the distribution P^{k+1} we use the following geometric interpretation (see Figure 6).

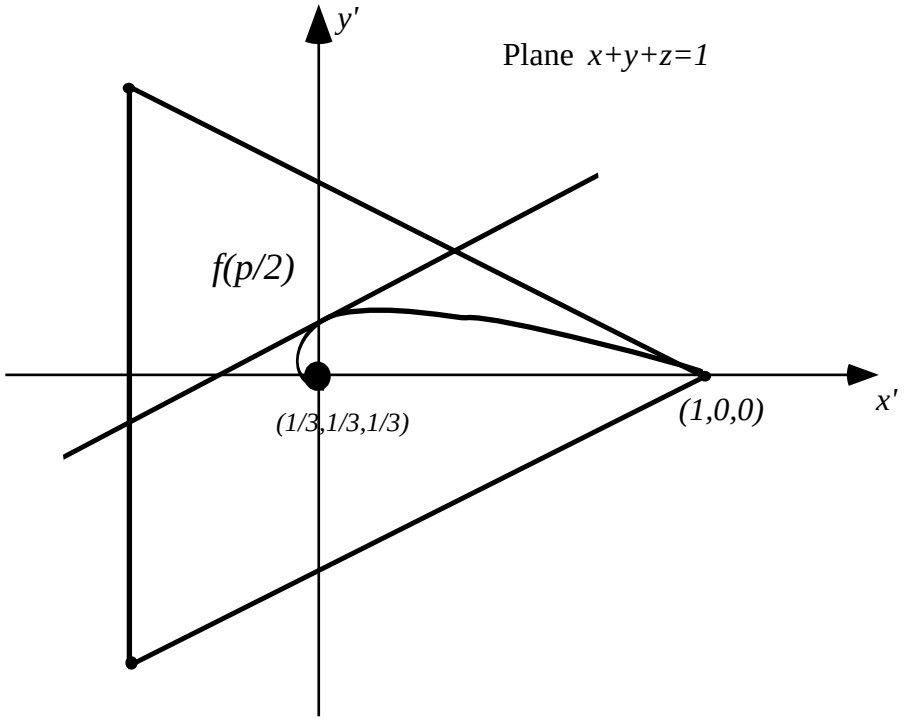

Figure 6: Geometric interpretation

Clear, all our matrices $\hat{R}^i$, $\hat{H}^i$ and $\hat{M}^i$ preserve the plane $x + y + z = 1$. Consider the

restrictions R^i, H^i and M^i of these matrices onto this plane. The matrices R^i are rotations with angles α_i, and H^i are homotheties with coefficients $e^{-c\alpha_i}$.

Supply the plane $x+y+z=1$ with Cartesian coordinates (x',y') centered at $(\frac{1}{3},\frac{1}{3},\frac{1}{3})$, with the x'-axis containing $(1,0,0)$. Consider the logarithmic spiral $\phi(t)=e^{-ct}(\cos t,\sin t)$, the parameter t can be taken as the coordinate of a point on the spiral. One can see that our matrices $\hat{M}^i$ preserve the spiral, and being restricted on the spiral they act by adding α_i to the coordinate t. Thus, the point $P=\prod_{d_i=take}(1,0,0)\hat{M}^i$ lies on the spiral and has the coordinate $t=\sum_{\{i:d_i=take\}}\alpha_i$.

Now choose a linear function $L:\boldsymbol{R}^3\to\boldsymbol{R}$, $L(x)=\langle l,x\rangle$, $l=(l_i)\in\boldsymbol{R}^3$, $0<l_i\le 1$ such that the point $\phi(\frac{\pi}{2})$ maximizes L on the spiral. This can be done along the following lines.

Let T be a tangent vector to our spiral $\phi(t)$ at the point $\phi(\frac{\pi}{2})$. Take a vector l such that $\langle l,T\rangle=0$, and $L(\phi(\frac{\pi}{2}))>L(\frac{1}{3},\frac{1}{3},\frac{1}{3})$.

This vector l can be chosen by a small rotation of vector $(\frac{1}{3},\frac{1}{3},\frac{1}{3})$ around T.

We use the coordinates l_α as transition probabilities to arrive at $\boldsymbol{t}$ from the vertices $v_{k+1,\alpha}$. Thus the probability of success of σ is $L(P)$. Notice that $\sum\alpha_i=\pi$. So, if a subset of A with the desired property does exist then every optimal strategy has the sum $\sum_{\{i:d_i=take\}}\alpha_i=\frac{\pi}{2}$, and thus, $\sum_{\{i:d_i=take\}}z_i=\frac{p}{2}$.

7.1.9

The construction above does not take into consideration the rationality of the probabilities of deviations. For this reason we take appropriate rational approximations to the values defined above.

Suppose that the set A admits a partition. Denote it by $\bar{A}$.

Now we replace the values of the function μ by some rational approximations with polynomial number of digits. We show that every optimal strategy for this MDP-graph also provides us with a partition of the set A.

Assume that $\sum z_i\le e^n$ and $k\le n$.

To make necessary estimates we need the following inequality

Lemma 12 *For a constant $\theta>0$*

$$L(\phi(\frac{\pi}{2}))-L(\phi(\sum_{i\in A'}\alpha_i))\ge\theta e^{-n}$$

whenever $\sum_{i\in A'}z_i\ne\sum_{i\in A\setminus A'}z_i$.

Proof. Consider the function $\psi:[0,2\pi]\to\boldsymbol{R}$, defined by $\psi(t)=L(\phi(\frac{\pi}{2})-\phi(t))$. The inequality to prove can be rewritten as

$$\psi(\frac{\pi}{2}-r)\ge\theta e^{-n},$$

where $r=\frac{\pi}{2}-\sum_{i\in A'}\alpha_i$. The bound on $\sum z_i$ implies that $r\ge e^{-n}$.

Clear,

(i) $\psi(\frac{\pi}{2})=0$ and $\psi(t)\ne 0$ for $t\ne\frac{\pi}{2}$;

(ii) $\psi'(\frac{\pi}{2})=L(-\phi'(\frac{\pi}{2}))=q>0$. (Direct computation.)

Thus, using Taylor expansion, we can state that for some absolute constants $\epsilon>0$ and $\eta>0$ the inequality

$$|\psi(\frac{\pi}{2}-r)|\ge|\psi(\frac{\pi}{2})-\frac{1}{2}\psi'(\frac{\pi}{2})r|-\eta|r|^2$$

holds for all $0 \le |r| \le \epsilon$ and thus we have

$$|\psi(\frac{\pi}{2} - r)| \ge \frac{q}{4}|r|$$

for all $0 \le |r| \le \epsilon$.

If $|r| \ge \epsilon$ then for some absolute constant $\delta > 0$ we have

$$\psi(\frac{\pi}{2} - r) \le \delta.$$

Put $\theta = \min\{\delta, \frac{1}{4}q\}$. Now the statement of the lemma follows from the above inequalities.

Indeed, if $|r| < \epsilon$ then

$$\psi(\frac{\pi}{2} - r) \ge \frac{q}{4}|r| \ge \theta e^{-n}.$$

Otherwise, $\psi(\frac{\pi}{2} - r) \ge \delta \ge \delta e^{-n} \ge \theta e^{-n}$. ■

To complete the proof we compute in polytime matrices $\tilde{M}^i$ and a linear function $\tilde{L}$ such that

$$||\tilde{M}^i - M|| \le \frac{1}{10}\theta e^{-n^3},$$

$$||\tilde{L} - L|| \le \frac{1}{10}\theta e^{-n^2}.$$

Then for every $B \subset A$ we have

$$||\prod_{i \in B} \tilde{M}^i - \prod_{i \in B} M^i|| \le \frac{1}{10}\theta e^{-n^2} \tag{55}$$

since $||M^i|| \le 1$.

Consider a strategy $\bar{\sigma}$ with actions $d_i = take$ whenever $i \in \bar{A}$.

The probability of success of $\bar{\sigma}$ is

$$\boldsymbol{R}(\bar{\sigma}) = \tilde{L}(\prod_{i \in \bar{A}} \tilde{M}^i) \cdot (1,0,0)$$

$$\ge \tilde{L}(\prod_{i \in \bar{A}} M^i \cdot (1,0,0)) - \frac{3}{10}\theta e^{-n^2}$$

$$\ge L(\prod_{i \in \bar{A}} M^i \cdot (1,0,0)) - \frac{3}{10}\theta e^{-n^2} - \frac{1}{10}\theta e^{-n^2}$$

$$= L(\phi(\frac{\pi}{2})) - \frac{4}{10}\theta e^{-n^2}.$$

(We used (55), $||\prod_{i \in \bar{A}} M^i|| \le 1$ and $l_i \le 1$.)

Let an optimal strategy σ have actions $d_i = take$ for $i \in A'$. Then

$$\boldsymbol{R}(\sigma) = \tilde{L}(\prod_{i \in A'} \tilde{M}^i) \cdot (1,0,0)$$

$$\le \tilde{L}(\prod_{i \in A'} M^i \cdot (1,0,0)) + \frac{3}{10}\theta e^{-n^2}$$

$$\leq L(\prod_{i\in A'} M^i \cdot (1,0,0)) + \frac{3}{10}\theta e^{-n^2} + \frac{1}{10}\theta e^{-n^2}$$

$$= L(\phi(\sum_{i\in A'} \alpha_i)) + \frac{4}{10}\theta e^{-n^2}.$$

Applying lemma 12 and comparing $\boldsymbol{R}(\sigma)$ and $\boldsymbol{R}(\bar{\sigma})$ we see that the optimality of σ implies that A' is also a partition.

References

[BBS94] D. Beauquier, D. Burago, and A. Slissenko. First decisions of an optimal T-strategy can be non periodic. *Manuscript*, 1994.

[BBS95] D. Beauquier, D. Burago, and A. Slissenko. On the complexity of finite memory strategies for Markov decision processes. In J. Wiedermann and P. Hájek, editors, *Mathematical Foundation of Computer Science. Proceeding of MFCS'95*, pages 191–200. Springer Verlag, August/September 1995. Lect. Notes in Comput. Sci, vol. 969.

[BdRS96] D. Burago, M. de Rougemont, and A. Slissenko. On the complexity of partially observed Markov decision processes. *Theor. Comput. Sci.*, 157(1):161–183, 1996.

[Ber76] D. P. Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, New York, 1976.

[BS98] D. Beauquier and A. Slissenko. Polytime model checking for timed probabilistic computation tree logic. *Acta Informatica*, 35:645–664, 1998.

[Col87] C. J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, New York, Oxford, 1987.

[Con91] A. Condon. The complexity of the Max Word problem. In *Proc. 8th Symp. on Theoretical Aspects of Computer Sci.*, pages 456–465, 1991.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the Association for Computing Machinery*, 42(4):857–907, 1995.

[DF93] J. F. Diaz-Frias. *Vérification probabiliste de problèmes de graphes: applications à la robotique mobile*. PhD thesis, Université Paris-Sud, 1993.

[DKP91] X. Deng, T. Kameda, and C. H. Papadimitriou. How to learn an unknown environment. In *Proc. 32nd Annu. Symp. on Foundations of Computer Science*, pages 298–303, 1991.

[dRDF92] M. de Rougemont and J. F. Diaz-Frias. A theory of robust planning. In *Proc. IEEE Intern. Conf. on Robotics and Automation*, pages 2453–2459, 1992.

[Eil74] C. J. Eilenberg. *Automata, Languages and Machines*. Academic Press, New York, 1974. Vol. A.

[Fel68] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley & Sons, Ins., 1968.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[Kal83] L.C.M. Kallenberg. Linear programming and finite Markovian control problems. Technical Report 148, Mathematics Centrum Tract, Amsterdam, 1983.

[KS60] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. D Van Nostad Co., Inc., Princeton, N.J., 1960.

[Pap85] C. H. Papadimitriou. Games against nature. *J. Computer and System Sciences*, 31:288–301, 1985.

[PT87] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision procedures. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[Put90] M.L. Puterman. Markov decision processes. In D.P. Heyman and M.J. Sobel, editors, *Handbooks in Operations Research and Management Science. Stochastic Models.*, pages 331–434. North Holland, 1990. Vol. 2.

[PY91] C. H. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theor. Comput. Sci.*, 84:127–150, 1991.

[Val79] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. on Comput.*, 8:410–421, 1979.

[Zal92] V. A. Zalgaller. A discussion of one question of Bellman. Technical report, St.Petersburg State University, 1992. Registered in VINITI, No 849-B92, 34 p. In Russian.