Minimal Disclosure in Partially Observable Markov Decision Processes

Nathalie Bertrand[1], **Blaise Genest**[2]

[1]INRIA Rennes Bretagne Atlantique, France

[2]CNRS, UMI IPAL, joint with NUS and A*STAR/I2R, Singapore

ABSTRACT. For security and efficiency reasons, most systems do not give the users a full access to their information. One key specification formalism for these systems are the so called *Partially Observable Markov Decision Processes* (*POMDP* for short), which have been extensively studied in several research communities, among which AI and model-checking. In this paper we tackle the problem of the *minimal information* a user needs *at runtime* to achieve a simple goal, modeled as reaching an objective with probability one. More precisely, to achieve her goal, the user can at each step either choose to use the partial information, or pay a fixed cost and receive the full information. The natural question is then to minimize the cost the user needs to fulfill her objective. This optimization question gives rise to two different problems, whether we consider to minimize the *worst case cost*, or the *average cost*. On the one hand, concerning the worst case cost, we show that efficient techniques from the model checking community can be adapted to compute the optimal worst case cost and give optimal strategies for the users. On the other hand, we show that the optimal average price (a question typically considered in the AI community) cannot be computed in general, nor can it be approximated in polynomial time even up to a large approximation factor.

1 Introduction

Partially Observable Markov Decision Processes (POMDP for short) form a powerful model to describe systems where part of the information is not accessible at runtime by the user, and where the effect of the user actions is randomized. Partial observation happens virtually in every real life systems for various reasons, *e.g.* complexity, privacy or security. The usual question is given the observation at runtime and the (offline) POMDP description of the complete system, can a user achieve some goal or optimize some value?

In this paper, rather than considering that the partial information is rigidly fixed, we aim at evaluating several observation schemes. In applications where partial observation arises from complexity reasons, the system should provide *at runtime* the weakest observation which still allows to achieve a given goal, as it would also have the minimum cost of deploiement. On the contrary, in the context of security, the objective is to design a secure system preventing an attacker to achieve her goal, by giving her only partial access to the state of the system. Knowing that the partial observation scheme may be vulnerable, one may be extremely careful and analyze the additional information (obtained by punctually attacking the partial observation) an attacker needs *at runtime* to achieve her felony.

We analyze in depth the simplest such framework to which many problems can be reduced: The aim of the user is to reach with probability 1 a set Goal of states. We consider only two alternatives of information: by default, the user gets a fixed partial information;

If requested, she can also obtain full information on the current state of the system. In this framework, each execution is naturally assigned a "cost" – the number of times the full information is requested – and the user always aims at minimizing this amount. Now, *giving a value* to a strategy reaching Goal can be done in two different meaningful ways: either the *worst case cost* the user can have to pay or the *average cost* she pays, while following this strategy. For both options, the objective is to compute the optimal cost (among almost-sure winning strategies) and if possible synthesize a family of strategies approximating this optimal for smaller and smaller approximation factors.

A first contribution is to show that efficient model-checking techniques can be adapted to compute the worse case cost. Furthermore, we design optimal strategies in such cases and prove that strategies with finite memory, based on the set $\mathcal{B}$ of so-called (discrete) belief states, are optimal. A belief state represents the set of states the system can be in after a given sequence of actions and observations. First, we check in polynomial time that Goal can be reached with probability 1 using req unrestrictedly. If it is not the case, the whole procedure is pointless. Assuming Goal can be reached almost surely, we define a family of generic strategies $(\sigma^n_{can})_{n\in\mathbb{N}}$ with memory state in $\mathcal{B}$ and which is almost-surely winning. To improve the worst case cost of these strategies, we compute the set of states from which Goal can be reached with probability 1 without ever requesting the full information, then the set of states requesting the full information at most once, at most twice etc. The whole process terminates in time polynomial in $|\mathcal{B}|$, and gives associated strategies with finite memory $\mathcal{B}$. Of course, $|\mathcal{B}|$ is at worse exponential in the number of states of the POMDP. The memory-size of these strategies, as well as the complexity of our algorithm are thus optimal since deciding whether Goal can be reached almost-surely without any requests (a simple sub-problem) is EXPTIME-complete and requires strategies with exponential memory [5]. We illustrate the approach on a simple example for which we analyze the family of strategies, and show that it is optimal also for average cost on this particular instance.

However, this optimality result for average cost in this simple example is far from being generally true. Indeed, first of all, we prove that computing optimal average cost for a POMDP under a reachability objective is undecidable. Even worse, it is undecidable to even approximate it, whatever the approximation factor. At last, we give non approximability factors exponential in the size of the system, and prove that there is no algorithm running in polynomial time in the number of belief states to approximate the optimal average cost within that factor, even over finite horizon, for which the undecidability result does not hold.

Related work The POMDP model has been studied by at least two communities: first by the Artificial Intelligence and Operational Research community where mostly the problem consists in optimizing a reward function. Here, the results are twofold. First, they propose heuristics to obtain policies to get good rewards, using value iteration, drid-based algorithms [8], strategy improvement [7], etc.; see for instance [11] for a survey. Also, they analyze the complexity class in which such problem falls: It is undecidable in general to compute or approximate the optimal reward [10], and it is NP-complete to do it in finite horizon [9]. Compared to these works, we consider a particular cost function which cannot be expressed as a reward, and our lower bound results needs only a polynomial number of

belief states.

More recently, the Model Checking community considered POMDP, where the question was mainly qualitative (probability 1 or positive) over a navigational goal: reachability or safety (avoiding to reach a state), once or repetedly. The problem of reachability with probability 1 is the dual problem of safety with positive probability. These problems are EXPTIME-complete in general [5] and PSPACE-complete when the user has no observation at all [12, 3]. While visiting infinitely often a state almost-surely has the same complexity as almost sure reachability, reaching infinitely often a state with positive probability is undecidable for POMDP [1]. Surprisingly, with a slight constraint on these infinitely many visits, namely that the limit average number of times the goal is visited shall be positive, this last problem is decidable [12]. More complex systems than POMDP have been considered, where two partially informed players have opposite objectives: the results on POMDP mentionned above basically carry over [2]. More general winning conditions have also been studied [4]. Up to our knowledge, only very recently both a particular numerical function (energy) and navigational goal were considered [6].

2 Notations

Given S a finite set, let $\mathsf{Dist}(S)$ denote the set of *distributions* over S, that is functions $d : S \to [0,1]$ such that $\sum_{s\in S} d(s) = 1$. The *support* of distribution d is defined as $\mathsf{Supp}(d) = \{s \in S \mid d(s) > 0\}$. If d is the *Dirac distribution* associated with $s \in S$ (that is $d(s) = 1$ and $d(t) = 0$ for every $t \neq s$) we will abuse notation and simply write $d = s$.

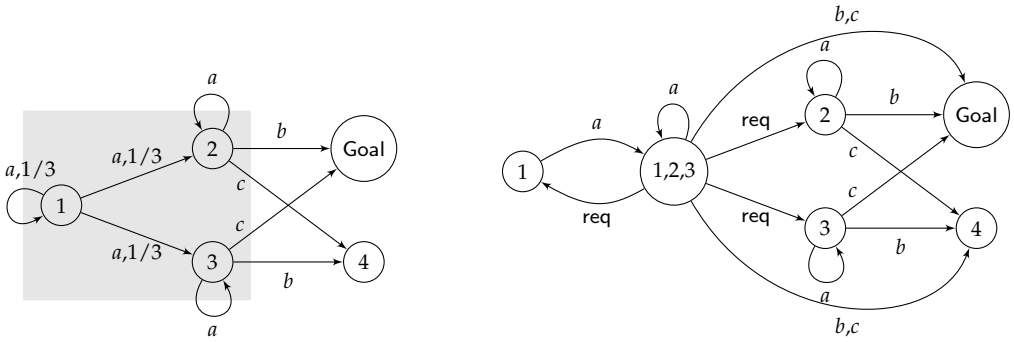
Definition 1. *[(PO)MDP] A* Markov decision process *(MDP) is tuple* $(Q, \mathsf{Act}, \Delta)$ *where* Q *is a finite set of states,* Act *is a finite set of actions, and* $\Delta : Q \times \mathsf{Act} \to \mathsf{Dist}(Q)$ *is the transition function.*

A partially observable MDP *(POMDP) is a tuple* $\mathcal{M} = (Q, \mathsf{Act}, \Delta, \mathsf{Part})$*, where* $(Q, \mathsf{Act}, \Delta)$ *is an MDP, and* Part *is a partition of* Q*. Elements of* Part *are called* observations.

Given a POMDP $(Q, \mathsf{Act}, \Delta, \mathsf{Part})$ the underlying MDP $(Q, \mathsf{Act}, \Delta)$ is alternatively called the FOMDP, for *Fully Observable* MDP. Intuitively, in a POMDP, from state s, if action a is chosen, the next state is t with probability $\Delta(s,a)(t)$ and the controller receives observation $O \in \mathsf{Part}$ such that $p \in O$. Of course, any MDP can be seen as a POMDP by setting Part as the set of all singletons sets. In the sequel, we always assume a fixed starting state s_0 and also distinguish a goal (set of) state(s) Goal. We assume for convenience that being in Goal is observable, that is for all $P \in \mathsf{Part}$, $P \subseteq 2^{\mathsf{Goal}} \cap P$ or $P \subseteq 2^{Q\setminus\mathsf{Goal}}$.

An example of POMDP is given in Figure 1(a), where $\mathsf{Part} = \big\{\{1,2,3\}, \{\mathsf{Goal}\}, \{4\}\big\}$ and the non-trivial set of the partition is represented by a grey area.

In this paper, given a POMDP $(Q, \mathsf{Act}, \Delta, \mathsf{Part})$, we assume that the controller can perform an extra action $\mathsf{req} \notin \mathsf{Act}$ whose effect is to disclose the precise state of the FOMDP. More precisely, the observation the controller receives after a request action req is $\{s\}$ if the current state is $s \in Q$. The set of possible observations, denoted $\mathcal{O}$, thus consists of the partition Part, together with $\mathsf{Single} = \{\{s\} \mid s \in Q\}$. The probabilistic transition function is extended to $\mathsf{Act}' = \mathsf{Act} \cup \{\mathsf{req}\}$ by defining $\Delta(s, \mathsf{req})$ as the Dirac distribution associated with observation $\{s\}$. Our high-level aim is to design a strategy of chosing actions to play



(a) A simple POMDP

(b) Corresponding graph of reachable beliefs

Figure 1: An example POMDP and its reachable belief state graph

based only on the sequence of actions played and observations received, such that the goal can be reached. Notice that from $s_0 = 1$, one needs to play a as no other choice is possible. A possible outcome is to reach state 1 again, hence for any choice of actions, there will always be a possibility to not reach Goal. Hence our aim is instead to ensure that Goal is reached, unless being terribly unluky. Formally, we want that the set of paths reaching Goal is of probability 1. We define now the probability space for G: the POMDP extended to Act'.

2.1 Belief States

Let $(a_1, O_1) \cdots (a_i, O_i) \in (\mathsf{Act} \times \mathsf{Part} \cup \{\mathsf{req}\} \times \mathsf{Single})^*$ be a sequence of actions played and observations received. We define $B(s_0, (a_1, O_1) \cdots (a_i, O_i))$ as the set of states $s \in Q$ such that for all $0 \leq j \leq i$ there exists t_j with $t_0 = s_0$, $t_i = s$, $t_j \in O_j$ and $\Delta(t_{j-1}, a_j)(t_j) > 0$. Intuitively if actions $a_1 \cdots a_i$ have been played and observations $O_1 \cdots O_i$ were received, the set of possible states are exactly those in $B(s_0, (a_1, O_1) \cdots (a_i, O_i))$. The set $B(s_0, (a_1, O_1) \cdots (a_i, O_i))$ can be computed inductively:

$$B(s_0, (a_1, O_1) \cdots (a_i, O_i)) = O_i \cap \{t \mid \exists s \in B(s_0, (a_1, O_1) \cdots (a_{i-1}, O_{i-1})),\ \Delta(s, a_i)(t) > 0\}.$$

The set Path of *possible finite paths* consists in all $\rho = (a_1, O_1) \cdots (a_n, O_n) \in (\mathsf{Act} \times \mathsf{Part} \cup \{\mathsf{req}\} \times \mathsf{Single})^*$ such that $B(s_0, (a_1, O_1) \cdots (a_n, O_n)) \neq \emptyset$. Moreover, the set of *reachable belief states* is defined by $\mathcal{B} = \{B(s_0, \rho) \mid \rho \in \mathsf{Path}\}$. For the example POMDP of Figure 1(a) the graph of reachable belief states is depicted in Figure 1(b). The number of belief states is at worst exponential in the number of states, but often, it is not that big.

Given a finite path ρ, $B(s_0, \rho)$ only gives the set of states the POMDP can be after executing ρ, but the precise probability to be in each state is unknown. The distributions $D(s_0, \rho)$ and $D(s_0, \rho, a)$ over states after a prefix of path (that is, ρ or ρa, with $\rho \in \mathsf{Path}$ and $a \in Act'$) can be defined by induction. Assuming $\rho = \rho'(a, O)$, and $D(s_0, \rho')$ is known, $D(s_0, \rho', a)$ is defined by $D(s_0, \rho', a)(s) = \sum_t D(s_0, \rho')(t) \times \Delta(t, a)(s)$. Then, taking the observation into account yields: $D(s_0, \rho)(s) = 0$ if $s \notin O$, else $D(s_0, \rho)(s) = D(s_0, \rho', a)(s) / \sum_{s \in O} D(s_0, \rho', a)(s)$.

Of course, the discrete belief after ρ can be recovered from the probabilistic belief: $B(s_0, \rho) = \mathsf{Supp}(D(s_0, \rho))$.

2.2 Strategy

A *strategy* $\sigma : \mathsf{Path} \to \mathsf{Dist}(\mathsf{Act}')$ for the controller is a function which associates with any possible path a distribution over the extended set of actions Act'. Given a strategy σ, a possible path $(a_1, O_1) \cdots (a_n, O_n) \in \mathsf{Path}$ is called a *σ-path* if $a_{i+1} \in \mathsf{Supp}\big(\sigma((a_1, O_1) \cdots (a_i, O_i))\big)$ for all $i < n$. When a fixed strategy σ is played, it is easy to define the probability that a possible path occurs, as follows. First of all, if $\rho = \rho'(a, O)$ is a σ-path, $\mathbb{P}_\sigma(\rho) = \mathbb{P}_\sigma(\rho') \cdot \sigma(\rho')(a) \cdot \sum_{s \in O} D(\rho', a)(s)$, and $\mathbb{P}(\rho) = 1$ if ρ is the empty path. Now, we can define the probability of a run $r = s_0 \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_n} s_n$ (every s_i is a state of the FOMDP) knowing that ρ occured. Let ρ be the unique (POMDP) path associated with r, obtained by replacing states with their associated observations. The probability of r assuming ρ is performed is given by $\mathbb{P}(r \mid \rho) = \prod_{i \leq n} D(s_0, \rho_i)(s_i)$, where ρ_i denotes the prefix of length i of ρ. Then we let $\mathbb{P}_\sigma(r) = \mathbb{P}_\sigma(\rho) \times \mathbb{P}(r \mid \rho)$. This probability measure on finite runs is extended in the usual way to the sigma-algebra they generate, and it is well-known that LTL properties on infinite runs are measurable for this measure [13]. The objective for the controller is to reach the Goal state. Thus, strategy σ is *almost-surely winning* if $\mathbb{P}_\sigma(\Diamond \mathsf{Goal}) = 1$. Clearly enough, if there is no almost-surely winning strategy in the FOMDP, then there will be none in the extended POMDP either.

Problem statement To every strategy σ, two quantities can be associated: first of all, the maximum number of requests actions σ takes, and second the expected number of requests following σ. In this paper we thus tackle the two distinct problems of finding almost-surely winning strategies which (1) minimize the worst-case cost, or (2) minimize the average cost.

3 Algorithms for the optimal worst case cost

Strategies are in general objects that do not have a finite presentation. In order to represent them effectively, it is common to restrict to finite-memory strategies, that are weaker than general strategies, but, as we will see, suffice when considering the optimal worst case cost problem. A *finite-memory strategy* on finite memory set M is given as $\sigma : M \to \mathsf{Dist}(\mathsf{Act}')$ together with an update function: $\mathsf{up} : M \times (\mathsf{Act} \times \mathsf{Part} \cup \{\mathsf{req}\} \times \mathsf{Single}) \to M$.

3.1 Reaching Goal with probability 1

We first propose a family $(\sigma^n_{can})_{n \in \mathbb{N}}$ of strategies with finite memory $\mathcal{B} \subseteq 2^Q$ the set of reachable belief states. The memory is initialized to $\{s_0\}$, where s_0 is the initial state. The update function is given by $\mathsf{up}(S, a, O) = O \cap \{t \mid \exists s \in S, \Delta(s, a_i)(t) > 0\}$. We extend it inductively to a path $\rho' = \rho \cdot (a, O)$ with $\mathsf{up}(S, \rho \cdot (a, O)) = \mathsf{up}(\mathsf{up}(S, \rho), a, O) = T$, and we say that the path ρ' reaches the memory state T. It is easy to see that the memory state M reached after some possible path ρ is exactly $B(s_0, \rho)$.

Let $n \in \mathbb{N}$. We now define how the strategy σ^n_{can} plays. First, we denote by $\mathsf{Lose}_F \subseteq Q$ the set of states s of the FOMDP associated with G such that there is no strategy in the FOMDP reaching Goal with probability 1 from s. We then denote by $\mathsf{Lose} \subseteq 2^Q$ the set of belief states S such that $S \cap \mathsf{Lose}_F \neq \emptyset$. In the example of Figure 1(a), 4 is the only state of the FOMDP from which there is no strategy reaching Goal almost-surely. In the reachable belief

graph of Figure 1(b), Lose is thus made of the single belief state $\{4\}$. Intuitively, any path reaching a memory state in Lose cannot reach Goal with probability 1. It is easy to see that if $\{s_0\} \in \mathsf{Lose}$, then there is no almost-surely winning strategy, since otherwise a strategy in the FOMDP reproducing σ (which is possible since it has at least as much information) would also reach Goal with probability 1.

Letting $\mathsf{Win} = \mathcal{B} \setminus \mathsf{Lose}$ the complementary set, we prove in the next theorem that under strategy σ^n_{can}, Goal is reached almost-surely from any belief state of Win. We partition Win in 3 sets: $W_0 = \{S \mid S \subseteq \mathsf{Goal}\}$ wins directly, W_N the *needing* belief states, and W_U the *non-needing* belief states. A state S is in W_N whenever for all $a \in \mathsf{Act}$, there exists $O \in \mathsf{Part}$, such that $B(S, a, O) \in \mathsf{Lose}$. Intuitively, from a memory state in W_N any almost-surely winning strategy needs to perform a request action req, as every other action leaves a chance to reach Lose. In the example of Figure 1(b), W_0 only contains the belief state $\{\mathsf{Goal}\}$, $W_N = \emptyset$ (in particular, actions a can be done safely from belief states $\{1\}$ and $\{1,2,3\}$), and W_U consists of the other belief states $\{1\}, \{1,2,3\}, \{2\}, \{3\}$. Strategy σ^n_{can} is then defined by:

- If $M \in W_0$, then $\sigma^n_{can}(M) = \emptyset$,
- if $M \in W_N$, $\sigma^n_{can}(M) = \mathsf{req}$,
- if $M \in W_U$, $\sigma^n_{can}(M)$ plays req with probability $1/n$, and plays uniformly all actions $a \in Act$ such that for every observation O, $\mathsf{up}(M, a, O) \notin \mathsf{Lose}$, and
- if $M \in \mathsf{Lose}$, then $\sigma^n_{can}(M)$ is the uniform distributions over all actions.

Furthermore, if $M \in \mathsf{Single}$ (that is, the actual state is known for sure), then we disallow σ^n_{can} to perform a req as it would be useless. Notice that the only infinite σ^n_{can}-paths are exactly those which never meet W_0.

THEOREM 2. *If* $\{s_0\} \in \mathsf{Win}$, *then* $\mathbb{P}_{\sigma^n_{can}}(\Diamond \mathsf{Goal}) = 1$.

Notice that we can compute in polynomial time the set of losing state in the FOMDP, and hence decide in polynomial time whether there exists an almost-surely strategy in G.

3.2 Optimizing for the worst case cost

Now that we know the set of belief states from which there is a strategy reaching Goal with probability 1, we can tune the canonical strategies σ^n_{can}. To do so, we compute inductively the set S_k of belief states from which one needs at most k actions req to win. The set S_0 is pretty easy to obtain, as the associated strategy cannot use any req.

Let $G_0 = (\mathcal{B}, \mathsf{Act}, \delta_0)$ be the *belief MDP* associated with the POMDP G, where each state is a belief state, and $\delta_0(B, a)$ is the uniform distribution over all belief states B' such that there exists a part $P \in \mathsf{Part}$ with $B' = P \cap \{t \mid \exists s \in B,\ \Delta(s, a)(t) > 0\}$. Notice that requests are not allowed in G_0. The MDP G_0 obtained from the POMDP in Figure 1(a) is very similar to the graph of Figure 1(b), except that the 3 edges req have been deleted. Let us denote by S_0 the set of belief states from which there exists a strategy σ_0 in G_0 reaching $W_0 = 2^{\mathsf{Goal}}$ almost-surely. The set S_0 can be computed in time linear in the number $|\mathcal{B}|$ of states of G_0, thus at worse in time exponential in the size of G. Taking our example of Figure 1(a), $S_0 = \big\{\{2\}, \{3\}, \{\mathsf{Goal}\}\big\}$. Indeed, $\{4\}$ is a losing state, and $\{1\}$ and $\{1,2,3\}$ as well since no req-action is allowed and playing b, c from $\{1,2,3\}$ has a positive probability to lead to $\{4\}$. Clearly, σ_0 can be chosen positional, and on the example it is given by $\sigma_0(\{2\}) = b$ and $\sigma_0(\{3\}) = c$.

Now, the canonical strategy σ^n_{can} is improved by letting $\sigma^n_{can}(B) = \sigma_0(B)$ if $B \in S_0 \setminus W_0$, and leaving it unchanged otherwise. Under this new definition, σ^n_{can} is still almost-surely reaching Goal, and from any $B \in S_0$, σ^n_{can} never proposes a req-action anymore:

PROPOSITION 3. *Assuming $s_0 \in$ Win, then: (i) σ^n_{can} is almost-surely winning, and (ii) for every σ^n_{can}-path $\rho = \rho_1\rho_2$ with $B(\rho_1) \in S_0$, ρ_2 contains no* req.

Given a strategy σ, we say that B is a *σ-belief state* if there exists a σ-path ρ with $B(\rho) = B$. We prove that S_0 is optimal, in the following meaning:

PROPOSITION 4. *Let σ be a strategy reaching* Goal *almost-surely from s_0, and $B \notin S_0$ a σ-belief state. Then there exists a σ-path $\rho = \rho_1\rho_2$ such that $B(\rho_1) = B$ and ρ_2 contains a* req.

PROOF. Let σ be an almost-surely winning strategy, and B a σ-belief state. Assume by contradiction that for every σ-path $\rho_1\rho_2$ with $B(\rho_1) = B$, ρ_2 contains no req-action, and let us prove that $B \in S_0$. We design a strategy σ' in the MDP G_0 from B as follows. For each run $r = Ba_1B_1 \cdots a_nB_n$ in G_0, let $\rho_r = Oa_1O_1 \cdots a_nO_n$ be the associated possible path in G with $B_i \subseteq O_i$ for all i. The choice of O_i is unique, and it is always a part of Part since $a_i \neq$ req. We let $\sigma'(r) = \sigma(\rho_1\rho_r)$. Now, it is easy to see that σ' reaches Goal almost surely from B, since ρ_1 is a finite σ-path and σ reaches Goal with probability 1. As a consequence, $B \in S_0$. ∎

We can now define the set S_1 of belief states for which at most one req-action is needed to reach Goal almost-surely. We let L_1 be the set of belief state B such that for all $s \in B$, $\{s\} \in S_0$. Playing a request from a state in L_1 obviously leads to some state in S_0, from which winning without request is possible. Clearly, $L_1 \cup S_0 \subseteq S_1$. In fact, S_1 is the set of all belief states from which there is a strategy to reach $L_1 \cup S_0$ in G_0 with probability one. S_1 can be computed as follows:

1. Initialize a set X of belief states at $\mathcal{B} \setminus$ Lose,
2. Compute the set $Y_X \subseteq \mathcal{B}$ of belief states which can reach $L_1 \cup S_0$ while staying in X, using a smallest fixed point:
   (a) initialize Y_X at $L_1 \cup S_0$,
   (b) add to Y_X all $B \in X$ such that there exists $a \in$ Act with $\mathsf{Supp}(\delta_0(B,a)) \subseteq X$ and $\mathsf{Supp}(\delta_0(B,a)) \cap Y_X \neq \emptyset$.
3. If $X \neq Y_X$ then set $X := Y_X$ and goto step 2 again, else set $S_1 = Y_X$ and quit.

The so-computed set S_1 of belief states is the largest one such that from all belief states of S_1 there is a strategy which allows to reach $L_1 \cup S_0$ and which ensures to stay in S_1. Now, we improve once again the canonical strategy σ^n_{can} with,

- if $B \in L_1 \setminus S_0$, $\sigma^n_{can}(B) =$ req,
- if $B \in S_1 \setminus (L_1 \cup S_0)$, $\mathsf{Supp}(\sigma^n_{can}(B))$ is the uniform distribution over the set of actions $a \in$ Act such that $\mathsf{Supp}(\delta_0(B,a)) \subseteq S_1$,
- otherwise, it is unchanged.

Notice that from $S_1 \setminus (L_1 \cup S_0)$, always at least one action $a \in$ Act satisfies $\delta_0(B,a) \subseteq S_1$.

After this modification, σ^n_{can} still reaches Goal with probability 1 and from any $B \in S_1 \setminus S_0$, σ^n_{can} proposes at most one req-action:

PROPOSITION 5. *Assuming $s_0 \in$ Win, then: (i) σ^n_{can} is almost-surely winning, and (ii) for every σ^n_{can}-path $\rho = \rho_1\rho_2$ with $B(\rho_1) \in S_1$, ρ_2 contains at most one* req.

PROOF. (ii) is fairly easy to establish. Let $\rho = \rho_1\rho_2$ be a σ^n_{can}-path with $B = B(\rho_1) \in S_1$. As $\rho_2 = B \xrightarrow{a_1} B_1\rho_2'$ is a σ^n_{can}-path, the first action a_1 ensures to stay in S_1: more precisely, if $B \in L_1$, then $B_1 \in S_0$, if $B \in S_0$, then $B_1 \in S_0$, and if $B \in S_1 \setminus (L_1 \cup S_0)$ then $B_1 \in S_1$. Iterating this argument, beliefs in ρ_2 always belong to S_1. In case L_1 is never reached, request are never performed. Otherwise, as soon as it reachs L_1, a request is played, and the next belief state is in S_0 from which no request are proposed anymore, according to the Proposition 3. Overall, (ii) is verified.

We now prove that from S_1, $L_1 \cup S_0$ is reached with probability 1, which proves that σ^n_{can} reaches Goal with probability 1 thanks to Proposition 3. The only paths from S_1 which does not reach $L_1 \cup S_0$ are those staying forever in $S_1 \setminus (L_1 \cup S_0)$. We now prove that from every state in S_1, there is a σ^n_{can}-path reaching $L_1 \cup S_0$, hence, a positive probability to reach $L_1 \cup S_0$. Together, these facts show that almost-surely $L_1 \cup S_0$ will be reached from S_1. Assume now by contradiction that there is a state $B \in S_1$ with $B(\rho_1) = B$ and such that for all σ^n_{can}-path $\rho = \rho_1\rho_2$, we have $B(\rho) \notin L_1 \cup S_0$. Hence these paths ρ_2 make no request. But this set of ρ_2 paths cover exactly the set of path from B which stay within S_1. Considering the last iteration of the construction of S_1. As it is the last iteration, $Y_X = X = S_1$ and $B \in X$. Now, in the construction of Y_X, B can never be added to Y_X. Hence $Y_X \neq X$, a contradiction. ∎

We prove that S_1 is optimal, in the following sense:

PROPOSITION 6. *Let σ be a strategy reaching* Goal *almost-surely from s_0 and $B \notin S_1$ a σ-belief state. Then there is a σ-path $\rho = \rho_1\rho_2$ with $B(\rho_1) = B$ and ρ_2 contains at least two* req*-actions.*

We can easily construct in this way by induction the sets $(S_k)_{k\in\mathbb{N}}$ of belief states requiering at most k requests, until stabilization: $S_{K+1} = S_K$, which happens at worse for $K = |\mathcal{B}|$. Computing each S_i takes $O(|\mathcal{B}|^2 \times |\mathsf{Act}|)$; overall, the procedure is in time $O(|\mathcal{B}|^3 \times |\mathsf{Act}|)$.

This easily improves the strategy σ^n_{can} by allowing requests only when they are needed. Then, denoting $S_\infty = 2^Q \setminus S_K$, we have the following optimality result:

PROPOSITION 7. *For every strategy σ reaching goal almost surely and every σ-belief states $B \in S_\infty$, for all $N \in \mathbb{N}$, there exists a σ-path $\rho = \rho_1\rho_2$ with $B(\rho_1) = B$ and ρ_2 contains at least N* req*-actions.*

In our example, notice that $S_1 = S_0$, and hence $S_\infty = \{\{1\}, \{1,2,3\}\}$. Hence, the improved canonical strategy in our running example is defined by: $\sigma^n_{can}(1) = a, \sigma^n_{can}(2) = b, \sigma^n_{can}(3) = c$ and $\sigma^n_{can}(\{1,2,3\})$ assigns probability $(n-1)/n$ to action a and probability $1/n$ to req. We show now that the family of canonical strategies is, on this particular example, also optimal for average cost! The proof is educational to understand that even with a fixed strategy, computing the average cost in a POMDP is not easy as the set of possible stochastic belief states (precise distributions over states in the discrete belief states) is potentially infinite.

Let $n \in \mathbb{N}$ and consider strategy σ^n_{can}. The game starts in state 1, and σ^n_{can} first decision is to perform an a. After this action, the discrete belief state is $\{1,2,3\}$ and the probability to be in state 1 is $1/3$. In the sequel, we denote by E_k the expected number of requests following σ^n_{can} from $\{1,2,3\}$ with probability $1/3^k$ to be in state 1. Thanks to the observation above, the expected number of requests for σ^n_{can} is exactly E_1.

Assume now that the current belief state is $\{1,2,3\}$ and the probability to be in state 1 is $1/3^k$. In this state, with probability $1/n$, a request is performed, which discloses state 1 with probability $1/3^k$; with probability $(n-1)/n$, a is played, and the resulting state is $\{1,2,3\}$ with probability $1/3^{k+1}$ to be in state 1. As a consequence, $E_k = \frac{1}{n}\left(1 + \frac{1}{3^k}E_1\right) + \frac{n-1}{n}E_{k+1}$. From there, we derive: $E_1 = 1 + \frac{1}{2n}$. Hence, the average number of req asked by σ^n_{can} is smaller than $1 + \frac{1}{2n}$, for all $n \in \mathbb{N}$. In fact, we can prove that this family of strategy is optimal in the sense that no almost surely winning strategy can achieve an average number of request of 1 or less on this particular example.

4 Hardness and undecidability results for the average cost

We turn now to a more general analysis of the strategies minimizing the average cost. Unfortunately, as we hinted before, this question is very hard to tackle. We show first that the problem of the existence of a strategy with cost smaller than a fixed threshold is in general undecidable, and that it is undecidable to approximate the optimal average cost. Moreover, we give concrete approximation factor (with respect to the size of the POMDP) up to which no optimal strategy can be computed with polynomial time algorithms. This obviously contrasts with the rather efficient algorithms we design in the previous section, which run in time polynomial in $|\mathcal{B}|$.

For a run r of G, we denote by $val(r)$ the number of req in r. Let σ be a strategy reaching $Goal$ from s_0 with probability 1. Then we denote by $val(\sigma)$ the expected value of $val(r)$, over all the σ-runs. Notice that $val(\sigma) < \infty$ since σ is almost-surely winning.

Definition 8. *The* value *of G is $val(G) = \inf\{val(\sigma) \mid \sigma$ almost-surely winning$\}$, where $val(\sigma)$ denotes the expected number of requests under strategy σ.*

We can now present our first negative result, namely that it is undecidable to compute the exact minimum average cost $val(G)$. This result should not be too surprising as optimizing a cost function in a POMDP is undecidable [10]. However, our result is stronger and harder to get, since our cost function is not arbitrary.

Theorem 9. *For all $K > 0$, it is undecidable to know whether $val(G) \leq K$.*

Proof. Let $\varepsilon \in (0, 1/2)$. Take a Probabilistic Finite Automaton $\mathcal{P}$ (a PFA for short, that is

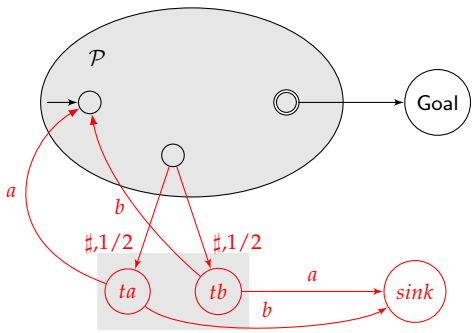


Figure 2: Reduction to a variant of the emptiness problem for PFA

a POMDP with $|\mathsf{Part}| = 1$) such that either there exists a word accepted with probability at least $1-\varepsilon$ or all words are accepted with probability less than ε. It is undecidable to know which case holds [10]. From $\mathcal{P}$, we build a POMDP G, as illustrated on Figure 2 by adding four new states. This reduction ensures the following: $\mathcal{P}$ accepts a word with probability greater than $1-\varepsilon$ if and only if $val(G) < \frac{\varepsilon}{1-\varepsilon}$. ∎

Actually, in the proof, we even show that if $val(G) \geq \varepsilon/(1-\varepsilon)$, then $val(G) \geq 1-\varepsilon$. That is, $(\varepsilon/(1-\varepsilon) + 1 - \varepsilon)/2$ is the best approximation one can make for this family of POMDP. The approximation factor is thus $\delta = (1/(1-\varepsilon) - \varepsilon)(1-\varepsilon)/2\varepsilon = (1-\varepsilon(1-\varepsilon))/2\varepsilon$, which converges to infinity as ε converges to 0. As a consequence:

COROLLARY 10. *For any δ, it is undecidable to approximate $val(G)$ with factor δ.*

Notice however that for bigger δ, the non-approximation result uses bigger and bigger PFA (and thus POMDP). The following result establishes the relationship between the number of states of the POMDP and the non-approximation factor.

THEOREM 11. *Assuming that $P \neq NP$, for any polynomial time algorithm $\mathcal{A}$, there exists a POMDP G with at most $3n^2 + n$ states and at most $9n^2 + 8n$ reachable belief states, such that $\mathcal{A}$ computes a strategy with value val on G with:*
approximation factor: $|val - val(G)|/val(G) \geq 2^{n-1}/n^2 - 1$, *and*
absolute approximation error: $|val - val(G)| \geq (2^{n-1}/n - n - 2)$.

PROOF. The proof is by reduction from 3-SAT. Let φ be a Boolean formula in conjunctive normal form with m clauses $C_1 \cdots C_m$ over k variables $x_1, \cdots x_k$. We let $n = k \times m$. We also let $\varepsilon = 1/2^n$. From φ, we derive a POMDP G such that φ is satisfiable if and only $val(G) \leq 3n\varepsilon$. We recall that φ is satisfiable if and only if for every clause C_i, one can choose one literal ℓ_i among the three literals of C_i such that for all $i, j \leq m$, the choices of ℓ_i and ℓ_j do not conflict.

The actions that can be played are any of the $2k$ literals $\{x_i, \overline{x_i} \mid i \leq k\}$, plus a dummy action. The POMDP starts in the initial state init, where only the dummy action can be fired, and with equal probability a variable $x_i, i \leq k$ is chosen leading to state labeled (C_1, i). Intuitively, the POMDP will remember actions played concerning this variable x_i and no other. All states $(C_1, 1), \cdots, (C_1, k)$ belong to the same part of the partition and thus the player does not know which variable is monitored.

From (C_1, i), three actions are enabled, corresponding to the three literals in clause C_1. If the action played is $x_j, \overline{x_j}$ with $j \neq i$, then the next state is (C_2, i) with probability $1-\varepsilon$. With probability $\varepsilon/2$ it is (C_2, x_i) and with probability $\varepsilon/2$ it is $(C_2, \overline{x_i})$. Intuitively, with small probability, the POMDP remembers wrongly that literal x_i or $\overline{x_i}$ has been chosen. If the action played is x_i, then the next state is (C_2, x_i) with probability $1-\varepsilon$, (C_2, i) with probability $\varepsilon/2$ and $(C_2, \overline{x_i})$ with probability $\varepsilon/2$. At last, if the action played is $\overline{x_i}$, then the next state is $(C_2, \overline{x_i})$ with probability $1-\varepsilon$, (C_2, i) with probability $\varepsilon/2$ and (C_2, x_i) with probability $\varepsilon/2$. Transitions from (C_ℓ, i) with $\ell < m$ and $i \leq k$ follow the same pattern.

Now, assume that the state is $(C_\ell, \overline{x_i})$, that is the POMDP recalls (possibly wrongly) that $\overline{x_i}$ has been played before. If the action played is x_i, then there is a conflict and the next state is test, which is the gadget, similar to the one in the proof of Theorem 9 that forces the player to perform a req-action in order not to lose, and then it reaches state ok. If the action played

is any other, then the next state is $(C_{\ell+1}, \overline{x_i})$ with probability $1-\varepsilon$, and $(C_{\ell+1}, x_i)$ or $(C_{\ell+1}, i)$ with probability $\varepsilon/2$ each. From state (C_ℓ, x_i), the transitions are symmetric. At last, for the last clause, from (C_m, x_i), all the actions lead to ok except action $\overline{x_i}$ which leads to the test gadget.

This reduction ensures: there exists a strategy σ reaching ok with probability 1 and such that $val(\sigma) \leq n\varepsilon$ if and only if the formula φ is satisfiable. Assume that there is a non conflicting choice of literals for every clause. Consider the strategy σ which chooses to play accordingly to this choice of literal, and when in the test gadget performs a req. With probability higher than $1 - n\varepsilon$, the POMDP remembers accurately the choice of literal x_i by σ at each step (there are less than n steps). As this choice is not conflicting, under this hypothesis, no req is played and ok is reached. Now, with probability less than $n\varepsilon$, the memory can be wrong at some point, and the worse case is to reach the test, in which case a unique requests is made before reaching ok. Thus, $val(\sigma) \leq n\varepsilon$. For the reverse implication, observe that under any strategy σ reaching ok with probability 1, there is at least one conflicting variable (successive choices of literal x_i and $\overline{x_i}$). As in the reduction of Theorem 9 we assume without loss of generality that σ does not propose any req but in test. With probability at least $1/n$, the POMDP remembers the conflicting variable, and with probability at least $(1 - n\varepsilon)$ the POMDP remembers accurately the first literal of x_i played, and then when $\overline{x_i}$ is played, the POMDP goes to test where a req is played. Overall, $val(\sigma) \geq (1/n - \varepsilon)$.

As SAT is NP-complete and assuming that $P \neq NP$, no polynomial time algorithm can decide whether $val(G) \geq (1/n - \varepsilon)$ or $val(G) \leq n\varepsilon$. To keep the error factor as low as possible, the safest is thus to play the average value, which proves the first item.

One can however notice that even if the approximation factor is large, the absolute gap between an approximation and the real value is rather small (less than 1), which would not be a huge concern in practice. We explain now how to keep the same factor while widening the absolute gap using a small trick. The previous reduction is enriched as follows: from state ok only the dummy action can be played, leading with probability ε to Goal, and with probability $1 - \varepsilon$ back to init. The probability to reach Goal after seeing init exactly i times is $\varepsilon(1-\varepsilon)^i$, that is the expected number of times init is seen is $\sum_i i \cdot \varepsilon \cdot (1-\varepsilon)^i \in [2^n - 2, 2^n]$ for n large enough. That is, if there is a non conflicting choice of literals, then $val(G) \leq n/2^n \cdot 2^n = n$. On the other hand, if all choice of 1 literal per clause is conflicting, then $val(G) \geq (1/n - \varepsilon) \cdot (2^n - 2) \geq 2^n/n - 2$. This concludes the proof. ∎

Notice that the first item of our result holds for POMDP without loops, that is in particular for finite horizon POMDP (horizon $\leq n$). Compared with other results on non approximability of optimal cost in (finite horizon) POMDP [9], our reduction does not rely on the (at worse exponentially many) discrete belief states to encode the problem, but uses the actual probability to be in a state. Indeed, the family of graph considered has a polynomial number of reachable belief states (for which the algorithm of the previous section are efficient). Actually, if we relax the constraint of a polynomial number of beliefs, the proof can easily be simplified to obtain an infinite non approximability factor, since $val(G) = 0$ if and only if the 3-SAT formula is satisfiable. It also proves that reasoning on beliefs is in some sense mandatory.

5 Conclusion

In this paper we investigated the problem of minimizing requests for full information in a POMDP in order to achieve a reachability objective with probability 1. On the one hand, the optimal worst-case cost is in $\mathbb{N} \cup \{\infty\}$ and can be computed in polynomial time in the number of discrete beliefs (that is, exponential time at worse), together with a finite memory strategy that guarantees this optimal cost. On the other hand, the optimal average cost is in $\mathbb{R}_{\geq 0}$ and can neither be computed nor approximated, and we provide large error factors for which no polynomial-time algorithm can approximate the average optimal cost up to that factor. In practice, despite the non-approximability result, quite accurate values can be obtained for some POMDPs using heuristics [11]. Our model can be enriched to allow several intermediate information levels, encoded by successive refinements of the partition, and for which the techniques developped here will certainly be useful.

References

[1] C. Baier, N. Bertrand, and M. Grösser. On decision problems for probabilistic Büchi automata. In *FOSSACS '08*, volume 4962 of *LNCS*, pages 287–301. Springer, 2008.

[2] N. Bertrand, B. Genest, and H. Gimbert. Qualitative determinacy and decidability of stochastic games with signals. In *LICS '09*, pages 319–328. IEEE, 2009.

[3] R. Chadha, A. P. Sistla, and M. Viswanathan. Power of randomization in automata on infinite strings. In *CONCUR '09*, volume 5710 of *LNCS*, pages 229–243. Springer, 2009.

[4] K. Chatterjee and L. Doyen. The Complexity of Partial-Observation Parity Games. In *LPAR '10 (Yogyakarta)*, LNCS 6397, pages 1–14, 2010.

[5] K. Chatterjee, L. Doyen, and T. A. Henzinger. Qualitative analysis of partially-observable Markov decision processes. In *MFCS '10*, volume 6281 of *LNCS*, pages 258–269. Springer, 2010.

[6] A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Toruczyk. Energy and Mean-Payoff Games with Imperfect Information. In *CSL*, LNCS 6247, pages 260–274, 2010.

[7] J. Fearnley. Exponential Lower Bounds for Policy Iteration. In *ICALP '10*, volume 6199 of *LNCS*, pages 551–562. Springer, 2010.

[8] W. Lovejoy. Computationally feasible bounds for partially observed markov decision processes. *OR*, 39:162–175, 1991.

[9] C. Lusena, J. Goldsmith, and M. Mundhenk. Nonapproximability results for partially observable markov decision processes. *JAIR*, 14, 2001.

[10] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1-2):5–34, 2003.

[11] K. Murphy. A Survey of POMDP Solution Techniques. Technical report, 2000.

[12] M. Tracol. Recurrence and transience for finite probabilistic tables. *TCS*, 412(12-14):1154–1168, 2011.

[13] M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS '85*, pages 327–338. IEEE, 1985.

Technical appendix

This appendix collects details ommitted in the core of the paper.

Computation of $val(\sigma^n_{can})$

$$E_k = \frac{1}{n}(1 + \frac{1}{3^k}E_1)) + \frac{n-1}{n}E_{k+1}.$$

This equality, multiplied by $\big((n-1)/n\big)^{k-1}$, and added up with all equalities, for $k \geq 1$ yields:

$$E_1 = \frac{1}{n}\sum_{k=1}^{\infty}\Big(\frac{n-1}{n}\Big)^{k-1}(1 + \frac{1}{3^k}E_1).$$

From which we compute:

$$\begin{aligned}
E_1 &= \frac{1}{n}\Big[\sum_{k=1}^{\infty}\Big(\frac{n-1}{n}\Big)^{k-1} + E_1\sum_{k=1}^{\infty}\Big(\frac{n-1}{n}\Big)^{k-1}\frac{1}{3^k}\Big] \\
&= \frac{1}{n}\Big[\frac{1}{1-(n-1)/n} + E_1\frac{n}{n-1}\sum_{k=1}^{\infty}\Big(\frac{n-1}{3n}\Big)^{k}\Big] \\
&= \frac{1}{n}\Big[\frac{1}{1/n} + E_1\frac{n}{n-1}\frac{(n-1)/(3n)}{1-(n-1)/(3n)}\Big] \\
&= \frac{1}{n}\Big[n + E_1\frac{n}{n-1}\frac{n-1}{2n+1}\Big] \\
&= 1 + E_1\frac{1}{2n+1}
\end{aligned}$$

Solving this last equation, we obtain: $E_1 = 1 + \frac{1}{2n}$, and thus $val(\sigma^n_{can}) = 1 + \frac{1}{2n}$. In fact, we can prove that this family of strategies is optimal in the sense that no almost surely winning strategy can achieve an average number of request of 1 or less on this particular example. By contradiction, assume that such a strategy σ exists. It is easy to see that σ cannot use b before using req, as there is a positive probability to be in state $1, 3$, and playing b would lead to the sink state 5, and σ cannot reach Goal with probability 1. The same applies for c. As b or c is needed for reaching Goal, all path which reach Goal uses at least 1 request. Now, take a smallest prefix of σ-path wich uses one req. This σ-path is necessarily of the form a^nreq, for some n. This σ path has positive probability, and being in state 1 after this σ-path has postiive probability. Hence playing req does not allow to win, and there is a σ-path which plays a second req later. As the prefix of this path after the second req as postive probability under σ to happen, $val(\sigma) > 1$.

Details for proof of Theorem 11. For the first item of the theorem, the reduction from 3-SAT described in the core of the paper is illustrated on Figure 3. The second item requires to repeat the behaviour, and the overview of the reduction is given in Figure 4.

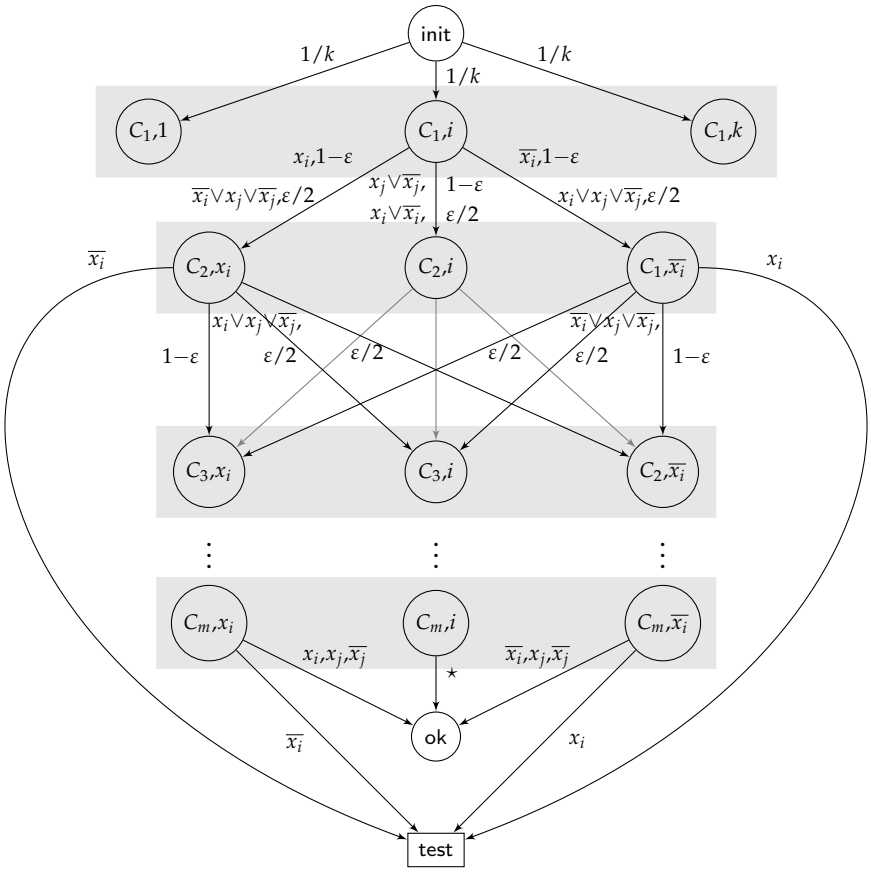


Figure 3: Reduction from 3-SAT

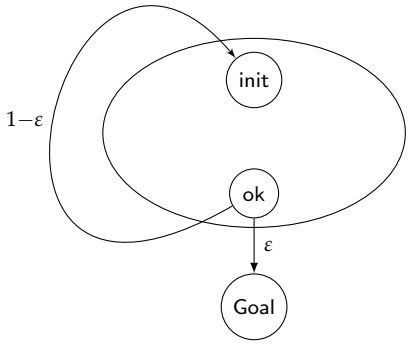


Figure 4: Bounding the absolute error in SAT reduction