2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

# Expedited Learning in MDPs with Side Information

Melkior Ornik, Jie Fu, Niklas T. Lauffer, W. K. Perera, Mohammed Alshiekh,
Masahiro Ono, and Ufuk Topcu

*Abstract*— **Standard methods for synthesis of control policies in Markov decision processes with unknown transition probabilities largely rely on a combination of exploration and exploitation. While these methods often offer theoretical guarantees on system performance, the number of time steps and samples needed to initially explore the environment before synthesizing a well-performing control policy is impractically large. This paper partially alleviates such a burden by incorporating a priori existing knowledge into learning, when such knowledge is available. Based on prior information about bounds on the differences between the transition probabilities at different states, we propose a learning approach where the transition probabilities at a given state are not only learned from outcomes of repeatedly performing a certain action at that state, but also from outcomes of performing actions at states that are known to have similar transition probabilities. Since the directly obtained information is more reliable at determining transition probabilities than second-hand information, i.e., information obtained from similar but potentially slightly different states, samples obtained indirectly are weighted with respect to the known bounds on the differences of transition probabilities. While the proposed strategy can naturally lead to errors in learned transition probabilities, we show that, by proper choice of the weights, such errors can be reduced, and the number of steps needed to form a near-optimal control policy in the Bayesian sense can be significantly decreased.**

## I. Introduction

This paper concentrates on the problem of effectively controlling an agent moving in an unknown environment. This question was investigated in a number of previous papers (e.g., [4], [5], [9]) and is of significant practical interest. In particular, remote vehicles sent into an unexplored environment need to deal with uncertainties in the system dynamics [10], [12] while working to accomplish their task.

A natural example of a system operating in unknown environments, and one we use in the numerical experiments in Section V, is of a Mars rover performing tasks on the surface of the planet. In that case, some information on the motion dynamics generated by the soil has been previously obtained by orbiters, but such information is partial and coarse. The rover's mission is to traverse the planet's surface in order to investigate different areas of interest. As the chance of the vehicle breaking down over time increases, and maintenance is not feasible, it is imperative for the rover to perform its mission as efficiently as possible [3]. However, in order to perform the mission, the rover needs to learn, in some way and within some error bounds, the underlying dynamics that it will use to perform near-optimal planning with goal-directed and obstacle avoidance missions.

The setting in this paper is based on a finite Markov decision process (MDP). That is, the state space is finite (e.g., through a discretization of a continuous environment), and at every point in the state space, the agent can choose one action from a finite set of actions. The agent's state and the chosen action then yield a probability distribution on the allowed transitions. In many practical cases, these probabilities are not fully known, but are to be estimated from real data.

The usual approach for control in MDPs with unknown transition probabilities sets up a trade-off between exploration and exploitation, where the agent balances between a reward obtained by its expectation of successfully meeting the control objective and a reward for visiting previously less visited states, the latter of which helps the agent learn the unknown transition probabilities. A number of algorithms within such a framework have been produced, including algorithms in the *probably approximately correct in Markov decision processes* (PAC-MDP) class (e.g., [2], [8], [15]) and the *Bayesian exploration bonus* (BEB) [9].

A bottleneck of the above process is in learning of the unknown transition probabilities. The agent learns by trial; each visit to a state and a subsequent action generate a *sample* that indicates where the agent went after taking the said action. Given enough samples, the agent can infer the transition probabilities at that state-action pair with high confidence. However, the number of samples necessary to produce an accurate model of the MDP is often impractically high, especially in systems with a large number of states.

This paper proposes a novel strategy for collecting samples and speeding up the learning of the transition probabilities. More specifically, it is often the case that states that are physically closer to each other have similar, albeit not same, transition probabilities. A similar observation is notably referenced in Tobler's first law of geography: "Everything is related to everything else, but near things are more related than distant things" [16]. Thus, if we possess some side information in the form of bounds on differences between transition probabilities at different states, then samples could

This work was partly funded by awards 1646522 and 1728412 from the National Science Foundation, N000141712623 from the Office of Naval Research, W911NF-15-1-0592 from the Army Research Office, and W911NF-16-1-0001 from the Defense Advanced Research Projects Agency.

M. Ornik, N. T. Lauffer, M. Alshiekh, and U. Topcu are with the Institute for Computational Engineering and Sciences, University of Texas at Austin. mornik@ices.utexas.edu, nlauffer@utexas.edu, malshiekh@utexas.edu, utopcu@utexas.edu

J. Fu is with the Electrical & Computer Engineering Department, Worcester Polytechnic Institute. jfu2@wpi.edu

W. K. Perera and U. Topcu are with the Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin. kasun8191@utexas.edu

M. Ono is with the Jet Propulsion Laboratory, California Institute of Technology. ono@jpl.nasa.gov

be partly reused, with the importance of each sample corresponding to the similarity between the state from which it was obtained and the state at which it is being applied.

It is clear that the proposed approach of indirect sampling, where samples obtained from similar states are also used to estimate transition probabilities, will lead to errors in estimates of transition probabilities. Nevertheless, if the importance, i.e., *weight*, of each such indirectly obtained sample is properly chosen, we show that these errors will remain small. Additionally, for the price that we pay by possibly inaccurate estimates, we receive a speed-up in sample collection. The desired level of trade-off between accuracy and learning speed depends on the needs of a particular application, and lower accuracy may slow down agent's progress toward its goal. On the other hand, in the numerical experiments (in Section V) we show that a simple model of a Mars rover can use indirect sampling in order to reach its final objective significantly more quickly.

The outline of the paper is as follows. In Section II, we formally introduce the setting, and present a method for indirect sampling. In Section III, we provide bounds on the errors in transition probabilities estimated using an indirect sampling approach, depending on the weights assigned to indirectly collected samples. We further discuss a particularly attractive class of possible weights, and provide bounds on errors in estimated transition probabilities when those weights are used. Section IV focuses on the theoretical development of a control strategy for unknown MDPs using the method of indirect sampling. After providing a short description of the BEB method for generating a near-optimal control policy in the Bayesian sense for MDPs with unknown probabilities, we present an improved bound for the number of steps required to converge to a near-optimal control policy in the Bayesian sense using BEB with indirect sampling. Finally, Section V presents the numerical experiments performed on the setting of a Mars rover. Particularly, Section V-A examines the trade-off between accuracy and speed when the agent's sole objective is to learn the transition probabilities in an MDP, and Section V-B discusses the influence of indirect sampling on the reduction in a number of time steps necessary for an agent to complete a simple control objective.

*Notation:* For a finite set $\mathcal{A}$, $|\mathcal{A}|$ denotes the number of its elements. For $x \in \mathbb{R}^n$, $\|x\|_1$ denotes its 1-norm. For $x \in \mathbb{R}$, $\lceil x \rceil$ denotes the smallest integer $y$ such that $x \leq y$.

## II. Indirect Sampling

Consider a Markov decision process (MDP) $\mathcal{M} = (S, A, P)$, where $S$ and $A$ are the finite state set and set of actions, respectively, and $P : S \times A \times S \to [0, 1]$ is the transition probability function; it satisfies $\sum_{s' \in S} P(s, a, s') = 1$ for all $s \in S$, $a \in A$. We assume that the values of the transition probability function $P$ are unknown at the beginning of a system run.

We study two objectives:

(a) Learn the transition probabilities as quickly and as accurately as possible.

(b) Given a reward function $R : S \times A \to \mathbb{R}$, $\varepsilon > 0$, and a *horizon length* $H \geq 0$, find, as quickly as possible, a control policy which ensures a nearly maximal expected reward for the agent over $H$ time steps. That is, find a policy $\pi^* : S \times \{1, 2, \ldots, H\} \to A$ such that, for all $s_1 \in S$,

$$E\left[\sum_{\tau=1}^{H} R\left(s_\tau, \pi^*(s_\tau, \tau)\right)\right] \geq \max_{\pi} E\left[\sum_{\tau=1}^{H} R\left(s_\tau, \pi(s_\tau, \tau)\right)\right] - \varepsilon, \tag{1}$$

where $s_\tau$ is the state of the system at the $\tau$-th time step.

Learning the transition probabilities in unknown MDPs proceeds by collecting *samples* — outcomes of agent's actions taken at every time step. In standard exploitation-exploration strategies such as PAC-MDP [8] or BEB [9], an action taken at a single time step provides exactly one sample. More formally, this learning algorithm, which we will in the future refer to as *direct sampling*, requires defining a *success counter* $\alpha : S \times A \times S \to [0, +\infty)$ and a *sample counter* $\alpha_0 : S \times A \to [0, +\infty)$. Both counters are initialized to 0, and every time action $a$ is played at state $s$, $\alpha_0(s, a)$ is increased by 1. If such an action resulted in the system state moving to $s' \in S$, the success counter $\alpha(s, a, s')$ is also increased by 1. Then, the transition probability $P(s, a, s')$ is estimated by $\tilde{P}(s, a, s') = \alpha(s, a, s')/\alpha_0(s, a)$.

A state-action pair $(s, a)$ is considered *known* if $\alpha_0(s, a) \geq m$ for some threshold $m$. We note that, based on Hoeffding's inequality [7], when $\alpha_0(s, a) \geq \frac{1}{2\mu^2} \log \frac{2}{\delta}$, with probability $1-\delta$ the difference between the estimated and the true transition probability is at most $\mu$. Thus, after a sufficient number of samples has been collected, further sample collection is largely unnecessary.

In this paper, we wish to exploit the side information that we may have about the system in order to increase the speed of sample collection. In particular, assume that we know that some of the states have similar, but not necessarily same, transition probabilities. (The notion of similarity between transition probabilities will be formalized in Definition 1.) We will consider such states *similar*. The sampling approach in the direct sampling algorithm could then be modified in the following way: all similar states can be considered the same, and all samples collected at one state can then equally count at any state similar to it. This approach will certainly significantly increase the number of samples obtained at every time step. However, counting similar states as being exactly same is a vast simplification of the actual dynamics. It does not allow for nuances in similarity between different states, and may hence lead to large errors in learned transition probabilities.

The central contribution of this paper is to provide a framework for a more subtle approach, where each sample is assigned a certain weight when being counted towards learning the dynamics of states other than the one it was collected at. We remark that such an approach is similar to the methods used in kernel density estimation (see, e.g., [6]).

Let us first introduce a measure of distance between two transition probabilities. This definition is motivated by the notion of $\varepsilon$-bisimulation [17]. The difference is that the definition below applies to states within a single MDP, as opposed to a distance between states of two similar processes as in [17].

*Definition 1:* Let $s, \overline{s} \in \mathcal{S}$, and let $0 \leq \varepsilon \leq 1$. States $s$ and $\overline{s}$ are *$\varepsilon$-distant* if there exists a permutation $\Pi : \mathcal{S} \to \mathcal{S}$ such that for any $s' \in \mathcal{S}$ and any $a \in A$, $|P(s,a,s') - P(\overline{s},a,\Pi(s'))| \leq \varepsilon$.

In plain words, permutation $\Pi$ from Definition 1 encodes "similar movement". To take the example of a Mars rover moving over a discretized terrain, if $s'$ is the state immediately to the north of $s$, then $\Pi(s')$ is the state immediately north from state $\overline{s}$. We note that $\Pi$ depends on $s$ and $\overline{s}$. However, in order not to bloat the notation, we will only emphasize this dependence when necessary.

In the remainder, we use the following notation. Let map $d : S \times S \to [0,1]$ be defined so that $d(s,\overline{s})$ is the smallest value such that states $s, \overline{s}$ are $d(s,\overline{s})$-distant. The side information used in this paper consists of upper bounds on the values of $d$; these bounds can be obtained from prior observations and analysis of environmental features, as briefly described in the introduction and in Section V. Definition 2 introduces weighting functions, based on Definition 1 and the above distance function $d$.

*Definition 2:* Function $w : [0,1] \to [0,1]$ is a *weighting function* if it is monotonically decreasing and satisfies $w(0) = 1$, $w(1) = 0$.

The proposed approach to sampling, given a weighting function $w$, is given in Algorithm 3. As is usually done with direct sampling, in order to obtain the theoretical results of Section III and Section IV, we specify that lines 7-10 in Algorithm 3 are only performed if the state-action pair $(s,a)$ is not yet known, i.e., $\alpha_0(s,a) \leq m$.

*Algorithm 3 (Weighted sampling)*

1 Let $\alpha(s,a,s') = 0$ for all $s, s' \in S$, $a \in A$.
2 Let $\alpha_0(s,a) = 0$ for all $s \in S$, $a \in A$.
3 **repeat at each time step**
4 Let $s$ be the state of the system at the beginning of the time step.
5 Let $a$ be the performed action.
6 Let $s'$ be the resulting state of the system after performing action $a$.
7 **for all** $\overline{s} \in S$
8 $\alpha(\overline{s},a,\Pi(s')) := \alpha(\overline{s},a,\Pi(s')) + w(d(s,\overline{s}))$
9 $\alpha_0(\overline{s},a) := \alpha_0(\overline{s},a) + w(d(s,\overline{s}))$
10 **end for**
11 **for all** $\overline{s}, \overline{s}' \in S$, $\overline{a} \in A$
12 $\tilde{P}(\overline{s},\overline{a},\overline{s}') := \alpha(\overline{s},\overline{a},\overline{s}')/\alpha_0(\overline{s},\overline{a})$
13 **end for**
14 **end repeat**

*Remark 4:* In order to avoid dividing by 0, values of $\alpha_0$ in Algorithm 3 are, instead of being initialized to 0, sometimes initialized to a small positive value.

We note that the Algorithm 3 results in at least increasing the sample and success counters at every state-action pair that was played by 1, as $d(s,s) = 0$ by Definition 1, and $w(0) = 1$ by condition (i) of Definition 2. Hence, regardless of the weighting function used, the weighted sampling algorithm results in collecting samples at least as quickly as the direct sampling algorithm. In fact, Algorithm 3 is a generalization of the direct sampling algorithm: direct sampling occurs if we take $w(d) = 0$ for all $d > 0$, $w(0) = 1$.

## III. Bounding the Error in Probability Estimates

The error in estimated probabilities using the weighted sampling algorithm will depend on the choice of a weighting function. As mentioned, if we set $w(d) = 0$ for all $d \in (0,1]$, there will be no error in estimated transition probabilities caused by second-hand (i.e., indirectly collected) samples, but we would not be using any side information. In general, consider the process described in Section II for learning the transition probability $P(s,a,s')$. Before probabilities $P(s,a,\cdot)$ are deemed to be known, the algorithm needs to collect $m$ samples, i.e., reach $\alpha_0(s,a) = m$.

For simplicity, we define $\omega : S \times S \to [0,1]$ by

$$\omega(s,s') = w(d(s,s')).$$

Additionally, let $\#_0(s,a)$ denote the number of visits by the agent to $s$ where $a$ was performed (before $\alpha_0(s,a)$ exceeded $m$), and let $\#(s,a,s')$ denote the number of visits by the agent to $s$ where $a$ was performed, and the agent proceeded to move to $s'$. The estimate $\tilde{P}(s,a,s')$ of $P(s,a,s')$ based on the collected $m$ samples using the weighted sampling algorithm is

$$\begin{aligned}\tilde{P}(s,a,s') &= \frac{\alpha(s,a,s')}{m} = \sum_{i=1}^{k} \frac{\omega(s,s_i)\#(s_i,a,s_i')}{m} \\ &= \sum_{i=1}^{k} \omega(s,s_i) \cdot \frac{\#_0(s_i,a)}{m} \cdot \frac{\#(s_i,a,s_i')}{\#_0(s_i,a)} = \sum_{i=1}^{k} \omega_i p_i',\end{aligned} \tag{2}$$

where $s_1, \ldots, s_k \in S$ are states from which some samples were collected, and $s_i' = \Pi(s')$, with $\Pi$ being defined with respect to $s$ and $s_i$, as described in Definition 1. In (2), we defined $\omega_i = \omega(s,s_i)\#_0(s_i,a)/m \leq \omega(s,s_i)$, and $p_i' = \#(s_i,a,s_i')/\#_0(s_i,a)$. We note that $p_i'$ is an estimate of $P(s_i,a,s_i')$ using just directly collected samples. Additionally, $\sum \omega_i = 1$, because the total number of collected samples equals exactly $\sum \omega(s,s_i)\#_0(s_i,a) = m$, and $\#_0(s_i,a) \leq \alpha(s_i,a) = m$.

We want to find a bound on the error

$$e = |\tilde{P}(s,a,s') - P(s,a,s')| = \left| \sum_{i=1}^{k} \omega_i p_i' - P(s,a,s') \right|. \tag{3}$$

We make the following assumption.

*Assumption 5:* For all $s_i \in S$, $p_i' = P(s_i,a,s_i')$.

Assumption 5 ensures that the probability estimate $p_i'$, obtained from directly sampling $(s_i,a)$, is equal to the true

probability $P(s_i, a, s_i')$. Of course, this equality does not generally hold, but, if $(s_i, a)$ was directly sampled enough times, then by the law of large numbers $p_i'$ will indeed be a close approximation of $P(s_i, a, s_i')$.

Going back to (3), by triangle inequality, Assumption 5, and using that $\sum \omega_i = 1$, we obtain

$$e \leq \sum_{i=1}^{k} \omega_i |P(s_i, a, s_i') - P(s, a, s')|. \tag{4}$$

Assuming without loss of generality that $|P(s_i, a, s_i') - P(s, a, s')|$, $i = 1, \ldots, k$, is a decreasing sequence, and remembering that $\omega_i \leq \omega(s, s_i)$, the right-hand side in (4) can be bounded by $\sum_{i=1}^{l} \omega(s, s_i)|P(s_i, a, s_i') - P(s, a, s')| + \tilde{\omega}_{l+1}|P(s_{l+1}, a, s_{l+1}') - P(s, a, s')|$, where $l \leq k-1$ and $\tilde{\omega}_{l+1}$ are such that $\sum_{i=1}^{l} \omega(s, s_i) + \tilde{\omega}_{l+1} = 1$ and $\tilde{\omega}_{l+1} \leq \omega(s, s_{l+1})$.

We note that $|P(s_i, a, s_i') - P(s, a, s')| \leq d(s, s_i)$, by definition of $d$. Hence,

$$e \leq \sum_{i=1}^{l} \omega(s, s_i) d(s, s_i) + \tilde{\omega}_{l+1} d(s, s_{l+1}), \tag{5}$$

with $\omega(s, s_1) + \ldots + \omega(s, s_l) + \tilde{\omega}_{l+1} = 1$.

By a proper choice of the function $\omega$, i.e., $w$, error $e$ from (5) can be made as small as desired. However, such a choice may reduce the speed-up in sample collection. The optimal choice of $w$ is dependent on the application. In order to give a more concrete discussion of the trade-off between speed and accuracy, in the remainder of this section we examine the class of weighting functions given by

$$w_n(d) = (1-d)^n, \tag{6}$$

where $n \geq 0$, with $w_0(1) = 0$. The following statements hold:

(i) For all $n \geq m$ and all $d \in [0, 1]$, $w_n(d) \leq w_m(d)$,
(ii) for all $d \in (0, 1]$; $w_n(d) \to 0$ as $n \to \infty$,
(iii) for all $d \in [0, 1)$; and $w_n(d) \to 1$ as $n \to 0$.

By (ii), as $n \to \infty$, the weighting function $w_n$ places more and more importance on accuracy rather than the speed of sample collection, as the generated weights will be lower and lower. On the other hand, by (iii), as $n \to 0$, more importance is placed on the speed of sample collection than accuracy. These properties allow to explore the speed-accuracy trade-off in a simple fashion, by changing $n$.

Let us now interpret the error bound obtained in (5) for the class of weighting functions in (6). By defining $y_i = \omega_n(s, s_i) = (1 - d(s, s_i))^n$, from (5) we obtain

$$e \leq \sum_{i=1}^{l} y_i (1 - y_i^{1/n}) + \tilde{\omega}_{l+1}(1 - y_{l+1}^{1/n}), \tag{7}$$

with $y_1 + \ldots + y_l + \tilde{\omega}_{l+1} = 1$. Since $\tilde{\omega}_{l+1} \leq y_{l+1}$, the expression on the right-hand side of (7) can be bounded from above to obtain $e \leq \sum_{i=1}^{l} y_i(1 - y_i^{1/n}) + \tilde{\omega}_{l+1}(1 - \tilde{\omega}_{l+1}^{1/n}) = 1 - \sum_{i=1}^{l} y_i^{1+1/n} - \tilde{\omega}_{l+1}^{1+1/n}$. By Jensen's inequality (see, e.g., [13]) and by plugging in $\sum y_i + \tilde{\omega}_{l+1} = 1$, we get an upper error bound

$$e \leq 1 - \left(\frac{1}{l+1}\right)^{1/n} \leq 1 - \left(\frac{1}{|S|}\right)^{1/n}.$$

We note that for a fixed $|S|$, $e \to 0$ as $n \to \infty$. We also remark that the error bounds in the above section do not depend on the quality of the side information, i.e., the relationship between $d(s, s_i)$ and true differences $|P(s, a, s') - P(s_i, a, \Pi(s'))|$. Naturally, if the true differences between transition probabilities at $s$ and $s_i$, $s_i \in S$, are significantly smaller than $d(s, s_i)$, the obtained errors will be smaller than the above bounds guarantee.

## IV. Reduction in Sample Collection Bounds

Section III bounds the error in learned transition probabilities using the sampling approach described in Section II. We now discuss another aspect of the proposed approach, the increase in sample collection speed. Generally, this increase is difficult to quantify, for two reasons:

- The amount of samples that will be collected at every time step heavily depends on the "geometry of the MDP", i.e., on the state that the agent is in at any time and on the bounds on differences between transition probabilities at that state and all other states in the state space.
- Different approaches to control in unknown MDPs offer slight nuances of when to terminate learning, and how much importance to place on sample collection as opposed to environment exploitation.

In order to deal with the latter point, this section will concentrate on examining the increase in sample collection speed in the case of Bayesian exploration bonus (BEB) [9], with the understanding that similar discussions will hold for PAC-MDP or other popular approaches.

We now give a brief description of BEB. For more details, we refer the reader to [9]. BEB aims to find a solution to the problem of finding an optimal control strategy in the setting of unknown MDPs by choosing a control action that maximizes a combination of the expected reward (given the current estimates of the transition probabilities) and an exploration bonus that entices the agent to learn the transition probabilities more accurately. Formally, at every time step, if the agent is at state $s \in S$, it chooses the action $a^* \in A$ that maximizes $V_H(s, a)$ (recall that $H$ is the horizon length), where $V_k : S \times A \to [0, +\infty)$ is given recursively by

$$\begin{aligned} V_k(\overline{s}, \overline{a}) =& R(\overline{s}, \overline{a}) + \frac{\beta}{1 + \alpha_0(\overline{s}, \overline{a})} \\ &+ \sum_{s' \in S} \tilde{P}(\overline{s}, \overline{a}, s') V_{k-1}^*(s'), \\ V_0(\overline{s}, \overline{a}) =& R(\overline{s}, \overline{a}), \end{aligned} \tag{8}$$

where $\tilde{P}$ is the current estimate of the transition probability function $P : S \times A \times S \to [0, 1]$, and $V_{k-1}^*(s') = \max_a V_{k-1}(s', a)$. We note that, when calculating $V_{H-1}^*(s')$ in the recursion for $V_H(s, a)$, we are acting as if the system already went from $s$ to $s'$, and this additional time step

slightly changed $\alpha_0(s', a)$ and the estimated probabilities. Then, when calculating $V^*_{H-2}(s'')$ in the recursion for $V^*_{H-1}(s')$, we assume that the system went from $s$ through $s'$ to $s''$, and so on.

Like the alternative approaches based on PAC-MDP, after some initial finite number of time steps which are primarily dedicated to learning the transition probabilities, BEB develops a policy that is nearly optimal in the sense of (1). Unlike PAC-MDP, the policy developed in BEB is near-optimal only in the Bayesian sense, i.e., it makes the system behave nearly optimally *given the estimates of the transition probabilities*, which may not be the same as the true probabilities. The notion of estimate-based optimality used in BEB perfectly fits the framework of this paper, as weighted sampling can potentially lead to an error in learned transition probabilities.

The primary objective of this section is to give an estimate of the savings in the number of time steps needed before the BEB algorithm, now with weighted sampling, converges to a near-optimal Bayesian policy.

By [9], the number of time steps needed before BEB (with the direct sampling algorithm) converges to a near-optimal policy with probability $1-\delta$ is bounded by

$$O\left(\frac{|S||A|H^6}{\varepsilon^2}\log\frac{|S||A|}{\delta}\right), \tag{9}$$

with $H$ and $\varepsilon$ as defined in objective (b) in Section II. An outline of the derivation of (9) is provided in the Appendix as it relates to the upcoming discussion.

As the weighted sampling algorithm provides both directly and indirectly collected samples, the number of steps necessary to collect sufficiently many samples for convergence of BEB may be lower than for the direct sampling algorithm. Let us first give an improved bound for the case in which the distances between any two states in the state space are equal.

*Proposition 6:* Let $\tilde{d} \in (0,1)$. Assume that $d(s,s') = \tilde{d}$ for all $s, s' \in S$, $s \neq s'$. Let $p = w(\tilde{d})$ and $m = H^3/\varepsilon$. Then, BEB with weighted sampling converges to a near-optimal policy with probability $1-\delta$ after

$$O\left(\frac{|A|\left(\frac{H^3(1-(1-p)^{|S|})}{\varepsilon p}+|S|\right)H^3}{\varepsilon}\log\frac{|S||A|}{\delta}\right) \tag{10}$$

time steps.

Proposition 6 is proved in the Appendix. We note that, when $p \to 0$, bound (10) recovers the original bound (9).

It is of interest to generalize Proposition 6 to the case where the states are not equidistant. We present one such generalization, in which the states that are known to be similar are clustered together. Let $S$ be partitioned into $h$ subsets $S_1, \ldots, S_h$. Let $\mathrm{diam}_c(S) = \max_{1\le i\le h}\max_{s,s'\in S_i} d(s,s')$. We obtain the following result.

*Theorem 7:* Let $p = w(\mathrm{diam_c}(S))$, $m = H^3/\varepsilon$, and $M = \max_i |S_i|$. Then, the required number of time steps for convergence of BEB with weighted sampling with probability $1-\delta$ is

$$O\left(\frac{h|A|\left(\frac{H^3(1-(1-p)^M)}{\varepsilon p}+|M|\right)H^3}{\varepsilon}\log\frac{|S||A|}{\delta}\right). \tag{11}$$

The proof of Theorem 7 directly follows from Proposition 6, by separating any path $\Phi$ in $S\times A$ into paths in each of the clusters $S_1\times A, \ldots, S_h\times A$. Hence, we omit the details.

*Remark 8:* With some technical adjustments, and considering that Theorem 7 holds for any partition of $S$, (11) can be improved to

$$O\left(\min\frac{|A|H^3\left(|S|+\frac{H^3}{\varepsilon}\sum_{i=1}^{h}\frac{1-(1-p_i)^{|S_i|}}{p_i}\right)}{\varepsilon}\log\frac{|S||A|}{\delta}\right)$$

with $p_i = w(\max_{s,s'\in S_i} d(s,s'))$, and where the minimum goes over all $h$ and all partitions of $S$ into $S_1, \ldots, S_h$.

## V. Numerical Experiments

We now demonstrate the proposed algorithm on a setting of a Mars rover moving across a partially unknown terrain. In order to replicate the situation more accurately, the simulation setting is based on a high-resolution map of terrain types in the Mars Jezero crater obtained from the Mars Reconnaissance Orbiter. Jezero crater is of particular prominence as one of the potential landing sites of the Mars 2020 rover mission [18].

We converted the map into an MDP setting as follows. We discretized the terrain into tiles, with the set of all tiles forming the state space $S$ for the MDP. Each tile belongs to one of the three terrain types: *benign*, *rough*, and *rippled*. The types were roughly adapted from [11]. Fig. 1 illustrates the $50\times 50$ grid used in Section V-A.

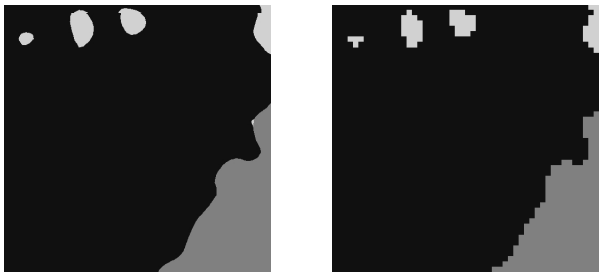

Fig. 1. The terrain map used as a state space in Section V-A. On the left is the original high-resolution Jezero crater map of terrain types. Each color represents a different terrain type: black terrain is benign, grey terrain is rough, and white terrain is rippled. The coarser $50\times 50$ grid used in simulations of Section V-A is given on the right.

Different types vary in the level of similarity between transition probabilities at neighboring tiles, as well as in the *slip rate*, i.e., the probability of the rover moving in a direction that was not intended. In our simulations, tiles of the benign type have an slip rate around $0.05$, and every two neighboring tiles of that type are known to be no more than $0.03$-distant. Rough type has a slip rate around $0.1$, and distances of up to $0.07$ between neighboring tiles. Rippled type has a slip rate around $0.15$, and neighboring tiles are up

to 0.03-distant. There is no known similarity between tiles of different types. We note that the particular values chosen for the bounds are intended to be merely illustrative.

In order to ensure that the agent does not move outside of the state space, we added additional border tiles on all sides of the state space. However, the transition probabilities at the borders are known and uninteresting, and their introduction is merely an artificial technical addition. Thus, we will not be mentioning the borders in the remainder of the section.

The set $A$ of actions consists of five *intended movements*: "up", "down", "left", "right", and "stay in the same position". However, because of the possibility of an error, there is always a positive probability that the agent will end up in a different neighboring tile than intended.

### A. Learning

In this subsection, we concentrate solely on learning of the transition probabilities. In particular, the agent moves as follows: it always chooses the intended movement that should result in it repositioning to the neighboring tile that has been visited the fewest number of times. Thus, after a large number of time steps, all tiles will be roughly equally visited.

We ran the system for 15 million steps, using the weighted sampling algorithm with the class of weight functions from (6). Fig. 2 presents the maximal error $e_{max} = \max_{s,a,s'} |\tilde{P}(s,a,s') - P(s,a,s')|$ for $n = 0$, $n = 20$, $n = 50$, as well as when using the direct sampling algorithm.

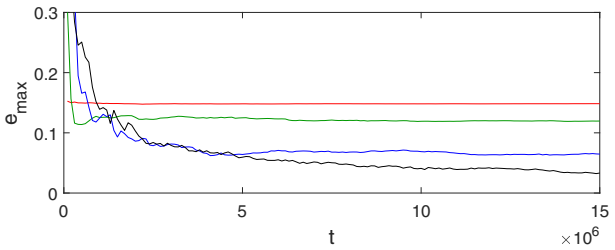


Fig. 2. The maximal errors obtained by the weighted sampling algorithm with $w_n(d) = (1-d)^n$. The red graph corresponds to errors for $n = 0$, green to $n = 20$, and blue to $n = 50$. The black graph corresponds to errors with the direct sampling algorithm. For memory reasons, the path taken was not saved between the runs for different values of $n$. Hence, the paths the agent took might slightly differ, without a significant influence on the results.

As expected, for lower values of $n$, the estimates of transition probabilities converge very quickly, but the error is higher. As $n \to \infty$, the estimates converges more slowly, but the error gets smaller. Additionally, the simulation results show that, at smaller numbers of time steps, learning using the weighted sampling algorithm is more accurate than with the direct sampling algorithm: for instance, after just $3 \times 10^5$ time steps, the weighted sampling algorithm with $n = 20$ only produced a maximal error around 0.12, while the error with the direct sampling algorithm was still around 0.3. The direct sampling algorithm took around four times as many time steps as the weighted sampling algorithm with $n = 20$ to reduce the $e_{max}$ to 0.12.

We note that, even with significant gains compared to using direct sampling, the number of steps required to learn the transition probabilities remains on the order of $10^6$. Such a high number is a natural consequence of the large number of system states: $|S| = 2500$. The required number of steps could be significantly reduced by heuristically assuming that some neighboring states, or states belonging to the same terrain type, have exactly the same transition probabilities. Nonetheless, the primary goal of this subsection is to shown that the weighted sampling algorithm, without any additional heuristics, is already significantly faster than direct sampling.

Finally, as seen in Fig. 2, while short-term behavior of the weighted sampling algorithm is significantly better than direct sampling, weighted sampling converges to a non-zero error. This issue can be resolved by imposing that the weights $\omega(s,s')$ are time-varying, i.e., that at every time step $t$, the number of collected samples is recomputed with new $\omega_t(s,s')$, where $\omega_t(s,s') \to 0$ for all $s \neq s'$ as $t \to \infty$. While we do not present a complete analysis of such an adjustment, it ensures that, for short-term objectives, large weights are still used, but that, on the other hand, the system asymptotically behaves in the same way as direct sampling.

### B. Control

The simulations presented in this subsection focus on a control objective of reaching a particular state $s^* \in S$. We want to measure how long it takes an agent to reach the goal state when it uses BEB with weighted sampling compared to the standard BEB, with direct sampling.

In the simulations that we conducted, the state space was a $10 \times 10$ grid based on the terrain from Fig. 1. Thus, $S = \{1, \ldots, 10\}^2$. In order to entice the agent to keep coming closer to the goal state, each state-action pair $(s,a)$ provides the reward $R(s,a)$ equal to $(\|s - s^*\|_1 + 0.02)^{-1}$. The addition of 0.02 is merely technical, in order to avoid $R(s^*,a) = \infty$. The agent then moves as described prior to (8). We set the horizon length to $H = 2$, with $\beta = 2H^2$. We chose the agent's initial state to be one of the corners of the grid, and the goal state to be the diagonally opposite corner.

As in Section V-A, we ran the simulation using the weighted sampling algorithm with weighting functions $w_n(d) = (1-d)^n$, for a number of different values of $n$. Since the number of time steps necessary to reach the goal state varies significantly from one system run to another, the simulation performed 20 runs each weighting function. Results are shown in Fig. 3.

Introducing weights produced a remarkable speed-up improvement of the agent's efficiency: with direct sampling, the agent needed around 136 time steps on average to reach the goal state. For the weighted sampling algorithm with $n = 35$, it required around 58. While there is a visible amount of variance in the average length of runs, which can be easily attributed to the inherent randomness involved in the simulation, Fig. 3 confirms that indirect sampling, with a good choice of a weighting function, can lead to substantial savings in time necessary to fulfill a control objective.

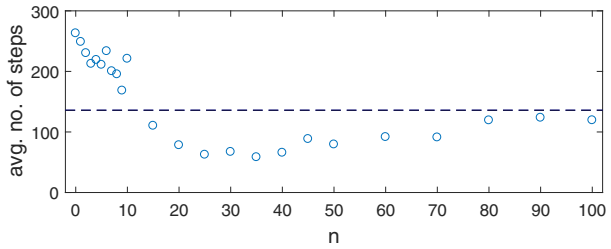


Fig. 3. The average number of time steps to reach the goal, with $w_n$ as in (6). The dashed line represents the number of time steps needed when using direct sampling.

## VI. Conclusions

This paper presents a novel strategy for expediting reinforcement learning in MDPs by using side information on similarities between transition probabilities at different states. The presented method rests on counting every collected sample both directly at the state-action pair at which it was collected, and — with a discounted weight — at state-action pairs with similar transition probabilities. The optimal choice of weights depends on the control objective, and determining the exact relationship between the two remains a crucial open question. Nonetheless, the theoretical results and numerical experiments presented in this paper show that the proposed method leads the system to learn the transition probabilities and satisfy the control objective significantly faster compared to algorithms that make no use of side information. Thus, in addition to open questions outlined within the paper, designing an analogous method to exploit available side information in the context of model-free learning constitutes a fruitful area of future research.

## References


[1] M. Araya-López, V. Thomas, and O. Buffet, "Near-optimal BRL using optimistic local transitions," in *29th International Conference on Machine Learning*, 2012, pp. 97–104.

[2] R. I. Brafman and M. Tennenholtz, "R-MAX — a general polynomial time algorithm for near-optimal reinforcement learning," *Journal of Machine Learning Research*, vol. 3, pp. 213–231, 2002.

[3] J. L. Bresina, A. K. Jónsson, P. H. Morris, and K. Rajan, "Activity planning for the Mars Exploration Rovers," in *15th International Conference on Automated Planning and Scheduling*, 2005, pp. 40–49.

[4] J. Fu and U. Topcu, "Probably approximately correct MDP learning and control with temporal logic constraints," in *Robotics: Science and Systems*, 2014.

[5] H. Gao, X. Song, L. Ding, K. Xia, N. Li, and Z. Deng, "Adaptive motion control of wheeled mobile robot with unknown slippage," *International Journal of Control*, vol. 87, no. 8, pp. 1513–1522, 2014.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[7] W. Hoeffding, "Probability inequalities for sums of bounded random variable," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[8] M. Kearns and S. Singh, "Near-optimal reinforcement learning in polynomial time," *Machine Learning*, vol. 49, pp. 209–232, 2002.

[9] J. Z. Kolter and A. Y. Ng, "Near-Bayesian exploration in polynomial time," in *26th International Conference on Machine Learning*, 2009, pp. 513–520.

[10] M. Mohammadi and A. M. Shahri, "Adaptive nonlinear stabilization control for a quadrotor UAV: Theory, simulation and experimentation," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 1, pp. 105–122, 2013.

[11] M. Ono, B. Rothrock, E. Almeida, A. Ansar, R. Otero, A. Huertas, and M. Heverly, "Data-driven surface traversability analysis for Mars 2020 landing site selection," in *2016 IEEE Aerospace Conference*, 2016, pp. 1–12.

[12] Z. Peng, D. Wang, Z. Chen, X. Hu, and W. Lan, "Adaptive dynamic surface control for formations of autonomous surface vehicles with uncertain dynamics," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 513–520, 2013.

[13] W. Rudin, *Real and Complex Analysis*. McGraw-Hill, 1987.

[14] A. L. Strehl, L. Li, and M. L. Littman, "Reinforcement learning in finite MDPs: PAC analysis," *Journal of Machine Learning Research*, vol. 10, pp. 2413–2444, 2009.

[15] A. L. Strehl and M. L. Littman, "An analysis of model-based interval estimation for Markov Decision Processes," *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1309 – 1331, 2008.

[16] W. R. Tobler, "A computer movie simulating urban growth in the Detroit region," *Economic Geography*, vol. 46, pp. 234–240, 1970.

[17] M. Tracol, J. Desharnais, and A. Zhioua, "Computing distances between probabilistic automata," in *9th Workshop on Quantitative Aspects of Programming Languages*, 2011, pp. 148–162.

[18] A. Witze, "Three sites where NASA might retrieve its first Mars rock," *Nature*, vol. 542, no. 7641, pp. 279–280, 2017.


## Appendix

*Outline of the proof of bound* (9)*:* Let $K \subseteq S \times A$ be the set of known state-action pairs. Every time a policy $\pi$ causes the agent to leave $K$, or remain in the set of unknown pairs, an estimate update (i.e., sample collection) occurs. In the direct sampling algorithm, each estimate update increases exactly one $\alpha_0(s,a)$ by 1. Thus, if we start from all $\alpha_0(s,a) = 0$, $\sigma = m|S||A|$ estimate updates will result in all state-action pairs becoming known.

Let us define $A_K$ as the event that $\pi$ results in an escape from $K$ within $H$ steps. Let $P(A_k)$ denote the probability of $A_k$ occurring. By a version of the Hoeffding inequality described in [1], [14], if $P(A_K) > \varepsilon/(2H^2)$, the event $A_K$ will occur at least $\sigma$ times after $O(\sigma H^3/\varepsilon \log(|S||A|/\delta))$ time steps, with probability $1-\delta$. Thus, after at most $O(\sigma H^3/\varepsilon \log(|S||A|/\delta))$, all the states will become known, with probability $1-\delta$. Now, if $P(A_K) \leq \varepsilon/(2H^2)$, which is certainly true once all states become known, it can be shown that BEB generates a near-optimal policy in the sense of (1).

*Proof of Proposition 6:* We will show that all elements of $S \times A$ become known after at most

$$|A| \left( \frac{(m+1)(1-(1-p)^{|S|})}{p} + 2|S| \right) \tag{12}$$

estimate updates. The claim of Proposition 6 then follows from the same discussion as the proof of bound (9).

Let $S = \{s_1, \ldots, s_{|S|}\}$. Bound (12) is a direct consequence of the following two claims.

*Claim 1:* Let $a \in A$. Let $\Phi$ be a *sequence of visits* to unknown elements in $S \times \{a\}$ defined as follows: $(s_1, a)$ is visited repeatedly until $\alpha_0(s_1, a)$ reaches $m$. Then $(s_2, a)$ is visited until its $\alpha_0(s_2, a)$ reaches $m$, and so on, until $\alpha_0(s_i, a) = m$ for all $i \in \{1, \ldots, |S|\}$. Let $v(\Phi)$ be the length of $\Phi$, i.e., the total number of visits until all states in $S \times \{a\}$ are known. Then, $v(\Phi) \leq (m+1)(1-(1-p)^{|S|})/p$.

*Claim 2:* Let $a \in A$. Let $\Phi'$ be any sequence of visits to unknown elements in $S \times \{a\}$. Then, if $\Phi$ is as defined in Claim 1, $v(\Phi') \leq v(\Phi) + 2|S|$.

We now give the proofs of the above two claims.

*Proof of Claim 1:* We note that $(s_1, a)$ will be visited $v_1 = m$ times. As a consequence of these visits, all elements $(s_i, a)$ will gain $v_1 p$ samples. Thus, $(s_2, a)$ will be visited $v_2 = \lceil m - v_1 p \rceil$ times. Analogously, before the visiting of $(s_3, a)$ starts, it will have gained $v_1 p + v_2 p$ samples, so it will be visited $v_3 = \lceil m - v_1 p - v_2 p \rceil$ times, unless $v_1 p + v_2 p > m$, in which case it will not be visited at all. We proceed analogously for all $(s_i, a)$. The total number of visits is $v(\Phi) = v_1 + \ldots + v_r$, where $v_i$ are generated by the recursion $v_i = \lceil m - v_1 p - \ldots - v_{i-1} p \rceil$, $v_1 = m$, and $r \leq |S|$ is such that $v_1 p + \ldots + v_r p > m$ or $r = |S|$. From this recursion, we obtain $v(\Phi) = v_1 + \ldots + v_r \leq v_1 + \ldots + v_{r-1} + m - v_1 p - \ldots - v_{r-1} p + 1 = m + 1 + (v_1 + \ldots v_{r-1})(1-p)$. Continuing inductively, we get $v \leq (m+1)(1-(1-p)^r)/p \leq (m+1)(1-(1-p)^{|S|})/p$.

*Proof of Claim 2:* The proof proceeds by induction on $|S|$. When $|S| = 1$, there is only one sequence of visits to unknown elements of $|S| \times \{a\}$, and it yields $v(\Phi') = v(\Phi)$. Now, assume that for *any* $m$ *and any* $|S| \leq k$, any sequence of visits $\tilde{\Phi}'$ will contain $v(\tilde{\Phi}') \leq v(\Phi) + 2|S|$ visits to unknown state-action pairs, with $\Phi$ defined as in Claim 2.

Now, suppose that there exists a sequence of visits $\Phi'$ on $S \times \{a\}$ with $|S| = k + 1$ such that

$$v(\Phi') > v(\Phi) + 2(k+1). \tag{13}$$

Assume that $(s_1, a)$ will become known first in $\Phi'$, then $(s_2, a)$, etc. This is taken without loss of generality: if it is not true, $\Phi$ as defined in Claim 1 can be modified so that its order of visits to state-action pairs matches the order in which the pairs in $\Phi'$ become known.

Let, for all $j \in \{1, \ldots, k+1\}$, $(s_j, a)$ become known under sequence $\Phi$ after a total of $t_j$ estimate updates. We note that $t_1 \leq t_2 \leq \ldots \leq t_{k+1} = v(\Phi)$. With analogous definitions, $t'_1 \leq t'_2 \leq \ldots \leq t'_{k+1} = v(\Phi')$.

We claim the following:

$$t'_j > t_j, \quad \text{for all } 1 \leq j \leq k+1. \tag{14}$$

Suppose (14) is incorrect, i.e., $t'_j \leq t_j$ for some $j$. Clearly, by the inductive assumption, $j \neq k+1$. By the construction of $\Phi$, after the $t_j$-th update in $\Phi$, pairs $(s_{j+1}, a), \ldots, (s_{k+1}, a)$ received exactly $t_j p$ samples each, as each pair was only obtaining samples indirectly. On the other hand, in sequence $\Phi'$, pairs $(s_{j+1}, a), \ldots, (s_{k+1}, a)$ have certainly received $t_j p$ sample data, but some pairs may also have received direct samples. Thus, the sample counts $\alpha_0(s_i, a)$, $i \geq j+1$, are at least as large when using $\Phi'$ as with sequence $\Phi$.

Consider now a new state space $\tilde{S} = \{s_{j+1}, \ldots, s_{k+1}\}$. Since under sequence $\Phi$ at time $t_j$ we have $\alpha_0(s_i, k) = t_j p$, we can instead reduce $m$ to $\tilde{m} = m - t_j p$ and have $\alpha_0(s_i, k) = 0$. We note that $\alpha_0(s_i, k)$ under sequence $\Phi'$, while nonnegative, might not all equal 0. However, that does not affect our proof. We also note that there might exist transition probabilities that force the agent to leave $\tilde{S}$. However, as all the other states in $S$ are known after time $t_j$, and we are only interested in counting $\alpha_0(s_i, k)$ for $i \in \{j+1, \ldots, k+1\}$, this possibility does not affect the remainder of our proof.

By the inductive assumption, any sequence of visits $\tilde{\Phi}'$ to unknown states in $\tilde{S} \times \{a\}$ satisfies $v(\tilde{\Phi}') \leq v(\tilde{\Phi}) + 2|\tilde{S}|$, where $\tilde{\Phi}$ is defined as in Claim 1 with respect to $\tilde{S}$. On the other hand, we assumed (13) and $t'_j \leq t_j$. Hence, the remainder of $\Phi'$ after the first $t_j$ visits is of length strictly greater than $v(\Phi) + 2(k+1) - t_j$. The remainder of $\Phi$ after the first $t_j$ visits is exactly $\tilde{\Phi}$. Hence, the remainder of $\Phi'$ is of length strictly greater than $v(\tilde{\Phi}) + t_j + 2(k+1) - t_j > v(\tilde{\Phi}) + 2|\tilde{S}|$. This is in contradiction with the inductive assumption. Thus, (14) is proved.

Let us now separate the sample data obtained by $\Phi'$ into four categories:

- $CD'$ *complete direct samples* — samples which were obtained directly (i.e., samples for $(s, a)$ obtained after playing $(s, a)$), which did not push $\alpha_0(s, a)$ above $m$,
- $ID'$ *incomplete direct samples* — samples which were obtained directly, and which pushed $\alpha_0(s, a)$ above $m$,
- $CI'$ *complete indirect samples* — indirectly collected samples which did not push $\alpha_0(s, a)$ above $m$,
- $II'$ *incomplete indirect samples* — indirectly collected samples which pushed $\alpha_0(s, a)$ above $m$.

Hence, the amount of samples need not be an integer. We claim the following holds:

$$CD' + ID' + CI' + II' = m(k+1), \tag{15a}$$
$$t_{k+1} \geq CD' + ID', \tag{15b}$$
$$II' \leq k+1, \tag{15c}$$
$$kp \geq (t'_1 + \ldots + t'_{k-1} + t'_k)p - CI' \geq 0, \tag{15d}$$
$$t'_{k+1} \leq CD' + k + 1. \tag{15e}$$

Statements (15a), (15b), (15c), and (15e) are obvious. For (15d), we note that in the first $t'_1$ visits made by $\Phi'$, each visit generated indirect samples for $k$ states. It is possible that $(s_1, a)$ became known by way of an indirect sample, which was then possibly incomplete, but all other samples are complete. Hence, before $t'_1$, $\Phi'$ produced at least $kpt'_1 - p$ complete indirect samples. By continuing analogously for $t'_2$ and onwards, we obtain between $kp(t'_1 - 1) + (k-1)p(t'_2 - t'_1 - 1) + \ldots + p(t'_k - t'_{k-1} - 1) = p(t'_1 + \ldots + t'_k) - pk$ and $p(t'_1 + \ldots + t'_k)$ complete indirect samples.

We analogously separate the samples obtained by $\Phi$ into $CD$, $ID$, $CI$, and $II$. Analogous claims to (15a)–(15e) hold for these values.

From (14) and (15d) we obtain that

$$\begin{aligned} CI' \geq & (t'_1 + \ldots + t'_k)p - kp \\ & \geq ((t_1 + 1) + \ldots + (t_k + 1))\, p - kp \geq CI. \end{aligned} \tag{16}$$

Now, we have

$$\begin{aligned} & v(\Phi) + 2|S| = t_{k+1} + 2|S| \geq m|S| - CI - II + 2|S| \\ & \geq m|S| - CI' + (|S| - II) + |S| \\ & \geq II' + ID' + CD' + |S| \geq t'_{k+1} = v(\Phi'), \end{aligned} \tag{17}$$

where the first line follows from (15a) and (15b), second line from (16), and third line from (15a), (15c), $II', ID' \geq 0$, and (15e). Claim (17) contradicts assumption (13). ■