Average Case Analysis of the Classical Algorithm for Markov Decision Processes with Büchi Objectives

Krishnendu Chatterjee¹, Manas Joglekar², and Nisarg Shah³

¹ IST Austria
² Stanford University
³ Carnegie Mellon University

Abstract. We consider Markov decision processes (MDPs) with ω-regular specifications given as parity objectives. We consider the problem of computing the set of almost-sure winning vertices from where the objective can be ensured with probability 1. The algorithms for the computation of the almost-sure winning set for parity objectives iteratively use the solutions for the almost-sure winning set for Büchi objectives (a special case of parity objectives). We study for the first time the average case complexity of the classical algorithm for computing almost-sure winning vertices for MDPs with Büchi objectives. Our contributions are as follows: First, we show that for MDPs with constant out-degree the expected number of iterations is at most logarithmic and the average case running time is linear (as compared to the worst case linear number of iterations and quadratic time complexity). Second, we show that for general MDPs the expected number of iterations is constant and the average case running time is linear (again as compared to the worst case linear number of iterations and quadratic time complexity). Finally we also show that given all graphs are equally likely, the probability that the classical algorithm requires more than constant number of iterations is exponentially small.

1 Introduction

Markov decision processes. Markov decision processes (MDPs) are standard models for probabilistic systems that exhibit both probabilistic and nondeterministic behavior [13], and widely used in verification of probabilistic systems [1]. MDPs have been used to model and solve control problems for stochastic systems [12]: there, nondeterminism represents the freedom of the controller to choose a control action, while the probabilistic component of the behavior describes the system response to control actions. MDPs have also been adopted as models for concurrent probabilistic systems [7], probabilistic systems operating in open environments [17], and under-specified probabilistic systems [2]. A specification describes the set of desired behaviors of the system, which in the verification and control of stochastic systems is typically an ω-regular set of paths. The class of ω-regular languages extends classical regular languages to infinite strings, and provides a robust specification language to express all commonly used specifications, such as safety, liveness, fairness, etc [19]. Parity objectives are a canonical way to define such ω-regular specifications. Thus MDPs with parity objectives provide the theoretical framework to study problems such as the verification and control of stochastic systems.

Qualitative and quantitative analysis. The analysis of MDPs with parity objectives can be classified into qualitative and quantitative analysis. Given an MDP with parity objective, the qualitative analysis asks for the computation of the set of vertices from where the parity objective can be ensured with probability 1 (almost-sure winning). The more general quantitative analysis asks for the computation of the maximal probability at each state with which the controller can satisfy the parity objective.

Importance of qualitative analysis. The qualitative analysis of MDPs is an important problem in verification that is of interest irrespective of the quantitative analysis problem. There are many applications where we need to know whether the correct behavior arises with probability 1. For instance, when analyzing a randomized embedded scheduler, we are interested in whether every thread progresses with probability 1 [9]. Even in settings where it suffices to satisfy certain specifications with probability $p < 1$, the correct choice of $p$ is a challenging problem, due to the simplifications introduced during modeling. For example, in the analysis of randomized distributed algorithms it is quite common to require correctness with probability 1 (see, e.g., [15, 14, 18]). Furthermore, in contrast to quantitative analysis, qualitative analysis is robust to numerical perturbations and modeling errors in the transition probabilities,
and consequently the algorithms for qualitative analysis are combinatorial. Finally, for MDPs with parity objectives, the best known algorithms and all algorithms used in practice first perform the qualitative analysis, and then perform a quantitative analysis on the result of the qualitative analysis [7, 8, 6]. Thus qualitative analysis for MDPs with parity objectives is one of the most fundamental and core problems in verification of probabilistic systems.

**Previous results.** The qualitative analysis for MDPs with parity objectives is achieved by iteratively applying solutions of the qualitative analysis of MDPs with Büchi objectives [7, 8, 6]. The qualitative analysis of an MDP with a parity objective with \(d\) priorities can be achieved by \(O(d)\) calls to an algorithm for qualitative analysis of MDPs with Büchi objectives, and hence we focus on the qualitative analysis of MDPs with Büchi objectives. The qualitative analysis problem for MDPs with Büchi objectives has been widely studied. The classical algorithm for qualitative analysis for MDPs with Büchi objectives was given in [7, 8], and the worst case running time of the classical algorithm is \(O(n \cdot m)\) time, where \(n\) is the number of vertices, and \(m\) is the number of edges of the MDP. Many improved algorithms has also been given in literature, such as [5, 3, 4], and the current best known worst case complexity of the problem is \(O(\min\{ n^2, m \cdot \sqrt{m} \})\). While the worst case complexity of the problem has been studied, to the best of our knowledge the average case complexity of none of the algorithms has been studied in literature.

**Our contribution.** In this work we study for the first time the average case complexity of the qualitative analysis of MDPs with Büchi objectives. Specifically we study the average case complexity of the classical algorithm for the following two reasons: (1) the classical algorithm is very simple and appealing as it iteratively uses solution of the standard graph reachability and alternating graph reachability algorithms; and (2) for the more involved improved algorithms it has been established that there are variants of the improved algorithms that never require more than linear time as compared to the classical algorithm. We study the average case complexity of the classical algorithm and establish that as compared to the quadratic worst case complexity, the average case complexity is linear. Our main contributions are summarized below:

1. **MDPs with constant out-degree.** We first consider MDPs with constant out-degree. In practice, MDPs often have constant out-degree: for example, see [10] for MDPs with large state space but constant number of actions, or [12, 16] for examples from inventory management where MDPs have constant number of actions (the number of actions correspond to the out-degree of MDPs). We show that for MDPs with constant out-degree, the expected number of iterations of the classical algorithm is at most logarithmic, and the average case running time is linear (as compared to the worst case linear number of iterations and quadratic time complexity).

2. **MDPs in the Erdős-Rényi model.** To consider the average case complexity of the general case, we consider MDPs where the underlying graph is a random graph according to the classical Erdős-Rényi random graph model [11]. We consider random graphs \(G_{n, p}\) over \(n\) vertices where each edge exists with probability \(p\) (independently of other edges). We show that if \(p \geq \frac{c \cdot \log(n)}{n}\), for any constant \(c > 2\), then the expected number of iterations of the classical algorithm is constant, and the average case running time is linear (again as compared to the worst case linear number of iterations and quadratic time complexity). Note that we obtain that the average case \((p = \frac{1}{2})\) running time for the classical algorithm is linear as a special case of our results for \(p \geq \frac{c \cdot \log(n)}{n}\), for any constant \(c > 2\). Moreover we show that when \(p = \frac{c}{n}\) (i.e., all graphs are equally likely), the probability that the classical algorithm will require more than constant iterations is exponentially small (less than \((\frac{1}{4})^{n}\)).

In summary our results show that the classical algorithm (the most simple and appealing algorithm) has excellent (linear-time) average case complexity as compared to the quadratic worst case complexity. As a consequence of our result it also follows that the improved algorithms also have linear time average case complexity, and thus we complete the average case analysis of the algorithms for the qualitative analysis of MDPs with Büchi objectives.

**Technical contributions.** The two key technical difficulties to establish our results are as follows: (1) Though there are many results for random undirected graphs, for the average case analysis of the classical algorithm we need to analyze random directed graphs; and (2) in contrast to other results related to random undirected graphs that proves results for almost all vertices, the classical algorithm stops when all vertices satisfy a certain reachability property; and hence we need to prove results for all vertices (as compared to almost all vertices). In this work we set up novel recurrence relations to estimate the expected number of iterations, and the average case running time of the classical algorithm. Our key technical results prove many interesting inequalities related to the recurrence relation for reachability properties of random directed graphs to establish the desired result. We believe the new interesting results related to reachabili-
ity properties we establish for random directed graphs will find future applications in average case analysis of other algorithms related to verification.

2 Definitions

Markov decision processes (MDPs). A Markov decision process (MDP) \( G = ((V, E), (V_1, V_P), \delta) \) consists of a directed graph \((V, E)\), a partition \((V_1, V_P)\) of the finite set \(V\) of vertices, and a probabilistic transition function \(\delta: V_P \to D(V)\), where \(D(V)\) denotes the set of probability distributions over the vertex set \(V\). The vertices in \(V_1\) are the player-1 vertices, where player 1 decides the successor vertex, and the vertices in \(V_P\) are the probabilistic (or random) vertices, where the successor vertex is chosen according to the probabilistic transition function \(\delta\). We assume that for \(u \in V_P\) and \(v \in V\), we have \((u, v) \in E\) if \(\delta(u)(v) > 0\), and we often write \(\delta(u,v)\) for \(\delta(u)(v)\). For a vertex \(v \in V\), we write \(E(v)\) to denote the set \(\{ u \in V \mid (v, u) \in E \}\) of possible out-neighbors. For technical convenience we assume that every vertex in the graph \((V, E)\) has at least one outgoing edge, i.e., \(E(v) \neq \emptyset\) for all \(v \in V\).

Plays, strategies and probability measure. An infinite path, or a play, of the game graph \(G\) is an infinite sequence \(\omega = \langle v_0, v_1, v_2, \ldots \rangle\) of vertices such that \(\langle v_k, v_{k+1} \rangle \in E\) for all \(k \in \mathbb{N}\). We write \(\Omega\) for the set of all plays, and for a vertex \(v \in V\), we write \(\Omega_v \subseteq \Omega\) for the set of plays that start from the vertex \(v\). A strategy for player 1 is a function \(\sigma: V^* \to D(V)\) that chooses the probability distribution over the successor vertices for all finite sequences \(w \in V^*: v_1\) of vertices ending in a player-1 vertex (the sequence represents a prefix of a play). A strategy must respect the edge relation: for all \(w \in V^*\) and \(u \in V_1\), if \(\sigma(w \cdot u)(v) > 0\), then \(v \in E(u)\). Once a starting vertex \(v \in V\) and a strategy \(\sigma \in \Sigma\) is fixed, the outcome of the MDP is a random walk \(\omega^v_\sigma\) for which the probabilities of events are uniquely defined, where an event \(A \subseteq \Omega\) is a measurable set of plays. For a vertex \(v \in V\) and an event \(A \subseteq \Omega\), we write \(P^v_\sigma(A)\) for the probability that a play belongs to \(A\) if the game starts from the vertex \(v\) and player 1 follows the strategy \(\sigma\).

Objectives. We specify objectives for the player 1 by providing a set of winning plays \(\Phi \subseteq \Omega\). We say that a play \(\omega\) satisfies the objective \(\Phi\) if \(\omega \in \Phi\). We consider \(\omega\)-regular objectives [19], specified as parity conditions. We also consider the special case of Büchi objectives.

- Büchi objectives. Let \(B\) be a set of Büchi vertices. For a play \(\omega = \langle v_0, v_1, \ldots \rangle \in \Omega\), we define \(\text{Inf}(\omega) = \{ \omega \in V \mid v_k = v \text{ for infinitely many } k \}\) to be the set of vertices that occur infinitely often in \(\omega\). The Büchi objectives require that some vertex of \(B\) be visited infinitely often, and defines the set of winning plays Büchi\((B) = \{ \omega \in \Omega \mid \text{Inf}(\omega) \cap B \neq \emptyset \}\).

- Parity objectives. For \(c, d \in \mathbb{N}\), we write \([c, d] = \{ c, c + 1, \ldots, d \}\). Let \(p: V \to [0, d]\) be a function that assigns a priority \(p(v)\) to every vertex \(v \in V\), where \(d \in \mathbb{N}\). The parity objective is defined as \(\text{Parity}(p) = \{ \omega \in \Omega \mid \min \{ p(\text{Inf}(\omega)) \} \text{ is even } \}\). In other words, the parity objective requires that the minimum priority visited infinitely often is even. In the sequel we will use \(\Phi\) to denote parity objectives.

Qualitative analysis: almost-sure winning. Given a player-1 objective \(\Phi\), a strategy \(\sigma \in \Sigma\) is almost-sure winning for player 1 from the vertex \(v\) if \(P^v_\sigma(\Phi) = 1\). The almost-sure winning set \(\langle 1 \rangle_{\text{almost}}(\Phi)\) for player 1 is the set of vertices from which player 1 has an almost-sure winning strategy. The qualitative analysis of MDPs correspond to the computation of the almost-sure winning set for a given objective \(\Phi\).

Algorithm for qualitative analysis. The almost-sure winning set for MDPs with parity objectives can be computed using \(O(d)\) calls to compute the almost-sure winning set of MDPs with Büchi objectives [6–8]. Hence we focus on the qualitative analysis of MDPs with Büchi objectives. The algorithms for qualitative analysis for MDPs do not depend on the transition function, but only on the graph \(G = ((V, E), (V_1, V_P))\). We now describe the classical algorithm for the qualitative analysis of MDPs with Büchi objectives and the algorithm requires the notion of random attractors.

Random attractor. Given an MDP \(G\), let \(U \subseteq V\) be a subset of vertices. The random attractor \(\text{Attr}_P(U)\) is defined inductively as follows: \(X_0 = U\), and for \(i \geq 0\), let \(X_{i+1} = X_i \cup \{ v \in V_P \mid E(v) \cap X_i \neq \emptyset \} \cup \{ v \in V_1 \mid E(v) \subseteq X_i \}\). In other words, \(X_{i+1}\) consists of (a) vertices in \(X_i\), (b) player-1 vertices whose all successors are in \(X_i\) and (c) probabilistic vertices that have at least one edge to \(X_i\). Then \(\text{Attr}_P(U) = \bigcup_{i \geq 0} X_i\). Observe that the random attractor is equivalent to the alternating reachability problem (reachability in AND-OR graphs).

Classical algorithm. The classical algorithm for MDPs with Büchi objectives is a simple iterative algorithm, and every iteration uses graph reachability and alternating graph reachability (random attractors). Let us denote the MDP
in iteration $i$ by $G^i$ with vertex set $V^i$. Then in iteration $i$ the algorithm executes the following steps: (i) computes the set $Z^i$ of vertices that can reach the current set of Büchi vertices; (ii) let $U^i = V^i \setminus Z^i$ be the set of remaining vertices; if $U^i$ is empty, then the algorithm stops and outputs $Z^i$ as the set of almost-sure winning vertices, and otherwise removes $\text{Attr}_P(U^i)$ from the graph, and continues to iteration $i + 1$. The classical algorithm requires at most $O(n)$ iterations, where $n = |V|$, and each iteration requires at most $O(m)$ time, where $m = |E|$. Moreover the above analysis is tight, i.e., there exists a family of MDPs where the classical algorithm requires $O(n)$ iterations, and total time $O(n \cdot m)$. Hence $\Theta(n \cdot m)$ is the tight worst case complexity of the classical algorithm for MDPs with Büchi objectives. In this work we consider the average case analysis of the classical algorithm.

3 Average Case Analysis for MDPs with Constant Out-degree

In this section we consider the average case analysis of the number of iterations and the running time of the classical algorithm for computing the almost-sure winning set for MDPs with Büchi objectives on family of graphs with constant out-degree.

**Family of graphs and results.** We consider graphs with vertex set of size $n$ (i.e., $|V| = n$) and the target set of Büchi vertices is of size $t$ (i.e., $|B| = t$). Every vertex $v$ has out-degree $d_v$, such that $d_{\min} \leq d_v \leq d_{\max}$, for given constants $d_{\min}$ and $d_{\max}$. Moreover every set of neighbors of size $d_v$ are equally likely and independent of neighbors of other vertices. We will show the following for this family of graphs: (a) if the target set $B$ has size more than $30 \cdot x \cdot \log(n)$ (i.e., $t \geq 30 \cdot x \cdot \log(n)$), then the expected number of iterations is $O(1)$ and the average running time is $O(n)$; and (b) if the target vertex set $B$ has size at most $30 \cdot x \cdot \log(n)$, then the expected number of iterations required is at most $O(\log(n))$ and average running time is $O(n)$, where $x$ is the number of distinct degrees.

**Notations.** We use $n$ and $t$ for the total number of vertices and the size of the target set, respectively. We will denote by $x$ the number of distinct out-degree $d_v$’s, and let $d_i$, for $1 \leq i \leq x$ be the distinct out-degrees. Since for all vertices $v$ we have $d_{\min} \leq d_v \leq d_{\max}$, it follows that we have $x \leq d_{\max} - d_{\min} + 1$. Let $a_i$ be the number of vertices with degree $d_i$ and $t_i$ be the number of target (Büchi) vertices with degree $d_i$.

The event $R(k_1, k_2, \ldots, k_x)$. The reverse reachable set of the target set $B$ is the set of vertices $u$ such that there is a path in the graph from $u$ to a vertex $v \in B$. Let $R(k_1, k_2, \ldots, k_x)$ denote the probability of the event that all vertices of any given set $S$ comprising of $k_i$ vertices of degree $d_i$, for $1 \leq i \leq x$ can reach $B$ via a path that lies entirely in $S$. Note that this probability only depends on $k_i$, $1 \leq i \leq x$ due to the symmetry between vertices. We will investigate the reverse reachable set of $B$, which contains $B$ itself. Recall that $t_i$ vertices in $B$ have degree $d_i$, and hence we are interested in the case when $k_i \geq t_i$ for all $1 \leq i \leq x$.

Consider a set $S$ of vertices that is the reverse reachable set, and let $S$ be composed of $k_i$ vertices of degree $d_i$ and of size $k$, i.e., $k = |S| = \sum_{i=1}^{x} k_i$. Since $S$ is the reverse reachable set, it follows that for all vertices $v$ in $V \setminus S$, there is no edge from $v$ to a vertex in $S$ (otherwise there would be a path from from $v$ to a target vertex and then $v$ would belong to $S$). Thus there are no incoming edges from $V \setminus S$ to $S$. Thus for each vertex $v$ of $V \setminus S$, all its neighbors must lie in $V \setminus S$ itself. This happens with probability

$$\prod_{i=1}^{x} \left( \frac{(n-k)}{d_i} \right)^{a_i-k_i},$$

since in $V \setminus S$ there are $a_i - k_i$ vertices with degree $d_i$ and the size of $V \setminus S$ is $n - k$. The probability that each vertex in $S$ can reach a target vertex is $R(k_1, k_2, \ldots, k_x)$. Hence the probability of $S$ being the reverse reachable set is given by:

$$\prod_{i=1}^{x} \left( \frac{(n-k)}{d_i} \right)^{a_i-k_i} \cdot R(k_1, k_2, \ldots, k_x)$$

There are $\prod_{i=1}^{x} \binom{a_i-t_i}{k_i-t_i}$ possible ways of choosing $k_i \geq t_i$ vertices (since the target set is contained) out of $a_i$. The value $k$ can range from $t$ to $n$ and exactly one these subsets of $V$ will be the reverse reachable set. So the sum of
probabilities of this happening is 1. Hence we have:

\[
1 = \sum_{k=1}^{n} \sum_{k_i=k,t_i \leq k_i \leq a_i} \left( \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right) \cdot \left( \frac{n-k}{d_i} \right)^{a_i-k_i} \right) \cdot R(k_1, k_2, \ldots, k_x) \tag{1}
\]

Let

\[
a_{k_1, k_2, \ldots, k_x} = \left( \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right) \cdot \left( \frac{n-k}{d_i} \right)^{a_i-k_i} \right) \cdot R(k_1, k_2, \ldots, k_x);
\]

\[
\alpha_k = \sum_{\sum k_i = k, t_i \leq k_i \leq a_i} a_{k_1, k_2, \ldots, k_x}.
\]

Our goal is to show that for \(30 \cdot x \cdot \log(n) \leq k \leq n-1\), the value of \(\alpha_k\) is very small; i.e., we want to get an upper bound on \(\alpha_k\). Note that two important terms in \(\alpha_k\) are \(\left( \frac{n-k}{d_i} \right)^{a_i-k_i}\) and \(R(k_1, k_2, \ldots, k_x)\). Below we get an upper bound for both of them. First note that when \(k\) is small, for any set \(S\) comprising of \(k\) vertices of degree \(d_i\) for \(1 \leq i \leq x\) and \(|S| = k\), the event \(R(k_1, k_2, \ldots, k_x)\) requires each non-target vertex of \(S\) to have an edge inside \(S\). Since \(k\) is small and all vertices have constant out-degree spread randomly over the entire graph, this is highly improbable. We formalize this intuitive argument in the following lemma.

**Lemma 1 (Upper bound on \(R(k_1, k_2, \ldots, k_x)\)).** We have

\[
R(k_1, k_2, \ldots, k_x) \leq \prod_{i=1}^{x} \left( 1 - \left( 1 - \frac{k}{n-d_i} \right)^{d_i} \right)^{k_i-t_i} \leq \prod_{i=1}^{x} \left( \frac{d_i \cdot k}{n-d_{\max}} \right)^{k_i-t_i}.
\]

**Proof.** Let \(S\) be the given set comprising of \(k\) vertices of degree \(d_i\), for \(1 \leq i \leq x\). Then for every non-target vertex of \(S\), for it to be reachable to a target vertex via a path in \(S\), it must have at least one edge inside \(S\). This gives the following upper bound on \(R(k_1, k_2, \ldots, k_x)\).

\[
R(k_1, k_2, \ldots, k_x) \leq \prod_{i=1}^{x} \left( 1 - \left( \frac{n-k}{d_i} \right)^{a_i-k_i} \right)^{k_i-t_i}.
\]

We have the following inequality for all \(d_i, 1 \leq i \leq x\):

\[
\frac{n-k}{d_i} = \prod_{j=0}^{d_i-1} \left( 1 - \frac{k}{n-j} \right) \geq \left( 1 - \frac{k}{n-d_i} \right)^{d_i} \geq 1 - \frac{d_i \cdot k}{n-d_i}.
\]

The first inequality follows by replacing \(j\) with \(d_i \geq j\), and the second inequality follows from standard binomial expansion. Using the above inequality in the bound for \(R(k_1, k_2, \ldots, k_x)\) we obtain

\[
R(k_1, k_2, \ldots, k_x) \leq \prod_{i=1}^{x} \left( 1 - \left( \frac{k}{n-d_i} \right)^{d_i} \right)^{k_i-t_i} \leq \prod_{i=1}^{x} \left( \frac{d_i \cdot k}{n-d_i} \right)^{k_i-t_i} \leq \prod_{i=1}^{x} \left( \frac{d_i \cdot k}{n-d_{\max}} \right)^{k_i-t_i}.
\]

The result follows.

Note that we have a loose and a strict bound on \(R(k_1, k_2, \ldots, k_x)\). We use the loose bound whenever it is sufficient and switch to the strict bound whenever required. Now for \(\left( \frac{n-k}{d_i} \right)^{a_i-k_i}\), we have the following trivial upper bound.

**Lemma 2.** We have \(\left( \frac{n-k}{d_i} \right)^{a_i-k_i} \leq (1 - \frac{k}{n})^{d_i(a_i-k_i)}\).
Proof. We have
\[
\left(\frac{n-k}{d_i} \right)^{a_i-k_i} = \left( \prod_{j=0}^{d_i-1} \left(1 - \frac{k}{n-j}\right) \right)^{a_i-k_i} \leq \left(1 - \frac{k}{n}\right)^{d_i(a_i-k_i)}
\]
The inequality follows since \(j \geq 0\) and we replace \(j\) by 0 in the denominator. The result follows.

Next we simplify the expression of \(\alpha_k\) by taking care of the summation.

**Lemma 3.** The probability that the reverse reachable set is of size exactly \(k\) is \(\alpha_k\), and

\[
\alpha_k \leq n^x \cdot \max_{\sum k_i = k, t_i \leq k_i \leq a_i} a_{k_1,k_2,...,k_x}.
\]

**Proof.** The probability that the reverse reachable set is of size exactly \(k\) is given by

\[
\alpha_k = \sum_{\sum k_i = k, t_i \leq k_i \leq a_i} \left( \prod_{i=1}^{x} \left( \frac{a_i-t_i}{k_i-t_i} \right) \cdot \frac{(n-k)}{(d_i/n)^{a_i-k_i}} \right) \cdot R(k_1,k_2,...,k_x),
\]
(refer to Equation 1). Since

\[
\alpha_k = \sum_{\sum k_i = k, t_i \leq k_i \leq a_i} a_{k_1,k_2,...,k_x},
\]
and there are \(x\) distinct degree's and \(n\) vertices, the number of different terms in the summation is at most \(n^x\). Hence

\[
\alpha_k \leq n^x \cdot \max_{\sum k_i = k, t_i \leq k_i \leq a_i} a_{k_1,k_2,...,k_x}.
\]
The desired result follows.

Now we proceed to achieve an upper bound on \(a_{k_1,k_2,...,k_x}\). As argued before if \(k\) is small, then \(R(k_1,k_2,...,k_x)\) is very small. On the other hand, consider the case when \(k\) is very large. In this case there are very few vertices that cannot reach the target set. Hence they must have all their edges within them, which is again highly improbale. Note that different factors bind \(\alpha_k\) depending on whether \(k\) is small or large. This suggests we should consider these cases separately. Our proof will consist of the following case analysis of the size \(k\) of the reverse reachable set: (1) when \(30 \cdot x \cdot \log(n) \leq k \leq c_1 \cdot n\) is small (for some constant \(c_1\)); (2) when \(c_1 \cdot n \leq k \leq c_2 \cdot n\) is large (for constants \(c_2 > c_1\)); and (3) when \(c_2 \cdot n \leq k \leq n - d_{\min} - 1\) is very large. The analysis of the constants will follow from the proofs. Note that since the target set \(B\) (with \(|B| = t\)) is a subset of its reverse reachable set, we have \(k < t\) is infeasible. Hence in all the three cases, we will only consider \(k \geq t\). We first consider the case when \(k\) is small.

### 3.1 Small \(k\): \(30 \cdot x \cdot \log(n) \leq k \leq c_1 n\)

In this section we will consider the case when \(30 \cdot x \cdot \log(n) \leq k \leq c_1 \cdot n\) for some constant \(c_1 > 0\). Note that this case only occurs when \(t \leq c_1 \cdot n\). We will assume this throughout this section. We will prove that there exists a constant \(c_1 > 0\) such that for all \(30 \cdot x \cdot \log(n) \leq k \leq c_1 \cdot n\) the probability \(\alpha_k\) that the size of the reverse reachable set is \(k\) is bounded by \(\frac{1}{n^x}\). Note that we already have a bound on \(\alpha_k\) in terms of \(a_{k_1,k_2,...,k_x}\). Next we convert \(a_{k_1,k_2,...,k_x}\) into a form that is easy to analyze. Let

\[
b_{k_1,k_2,...,k_x} = \prod_{i=1}^{x} \left( \frac{e \cdot (a_i-t_i)}{k_i-t_i} \right)^{k_i-t_i} \cdot e^{-\frac{t_i}{d_i} \cdot (a_i-k_i)} \cdot \left( \frac{d_i \cdot k}{n-d_{\max}} \right)^{k_i-t_i}.
\]

**Lemma 4.** We have \(a_{k_1,k_2,...,k_x} \leq b_{k_1,k_2,...,k_x} \).
Proof. We have

\[
  a_{k_1,k_2,...,k_x} = \left( \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right) \cdot \left( \frac{(n-k)}{d_i} \right)^{a_i-k_i} \right) \cdot R(k_1, k_2, ..., k_x)
\]

\[
  \leq \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right) \cdot \left( 1 - \frac{k}{n} \right) \cdot \left( \frac{d_i \cdot k}{n - d_{\text{max}}} \right)^{k_i-t_i}
\]

\[
  \leq \prod_{i=1}^{x} \left( e \cdot (a_i - t_i) \right)^{k_i-t_i} \cdot e^{-\frac{k}{n} d_i (a_i-k_i)} \cdot \left( \frac{d_i \cdot k}{n - d_{\text{max}}} \right)^{k_i-t_i}
\]

The first inequality follows from Lemma 1 and Lemma 2. The second inequality follows from the first inequality of Proposition 1 (in technical appendix) and the fact that \(1 - x \leq e^{-x}\). 

Maximum of \(b_{k_1,k_2,...,k_x}\). Next we show that \(b_{k_1,k_2,...,k_x}\) drops exponentially as a function of \(k\). Note that this is the reason for the logarithmic lower bound on \(k\) in this section. To achieve this we consider the maximum possible value achievable by \(b_{k_1,k_2,...,k_x}\). Let \(\partial_k b_{k_1,k_2,...,k_x}\) denote the change in \(b_{k_1,k_2,...,k_x}\) due to change in \(k_i\). Since \(\sum_{i=1}^{x} k_i = k\) is fixed, it follows that \(b_{k_1,k_2,...,k_x}\) is maximum when for all \(i\) and \(j\) we have \(\partial_k b_{k_1,k_2,...,k_x} = \partial_j b_{k_1,k_2,...,k_x}\). We have

\[
  \partial_k b_{k_1,k_2,...,k_x} = b_{k_1,k_2,...,k_x} \cdot \left( \frac{d_i \cdot k}{n} + \log \left( \frac{d_i \cdot k}{n - d_{\text{max}}} \right) + \log \left( \frac{a_i - t_i}{k_i - t_i} \right) \right)
\]

Thus, for maximizing \(b_{k_1,k_2,...,k_x}\), for all \(i\) and \(j\) we must have

\[
  \frac{d_i \cdot k}{n} + \log \left( \frac{d_i \cdot k}{n - d_{\text{max}}} \right) + \log \left( \frac{a_i - t_i}{k_i - t_i} \right) = \frac{d_j \cdot k}{n} + \log \left( \frac{d_j \cdot k}{n - d_{\text{max}}} \right) + \log \left( \frac{a_j - t_j}{k_j - t_j} \right)
\]

\[
  \Rightarrow \frac{k_i - t_i}{(a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k / n}} = \frac{k_j - t_j}{(a_j - t_j) \cdot d_j \cdot e^{d_j \cdot k / n}}
\]

\[
  \Rightarrow \frac{k_i - t_i}{(a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k / n}} = \frac{k_j - t_j}{(a_j - t_j) \cdot d_j \cdot e^{d_j \cdot k / n}}
\]

This implies that for all \(i\) we have

\[
  \frac{k_i - t_i}{(a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k / n}} = \frac{k - t}{\sum_{i=1}^{x} (a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k / n}}
\]

\[
  \Rightarrow k_i - t_i = \frac{(a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k / n}}{\sum_{i=1}^{x} (a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k / n}} \cdot (k - t)
\]

Lemma 5. Let \(L = \sum_{i=1}^{x} (a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k / n}\). We have

\[
  b_{k_1,k_2,...,k_x} \leq \left( \frac{L}{n - d_{\text{max}}} \right)^{-t} \cdot \left( \frac{L}{n - d_{\text{max}}} \cdot e^{1 - \frac{\sum_{i=1}^{x} d_i (a_i-t_i)}{n}} \right)^{k}
\]
The argument above shows that the maximum of $b_{k_1, k_2, \ldots, k_e}$ is achieved when for all $1 \leq i \leq x$ we have $k_i - t_i = \frac{(a_i - t_i) \cdot d_i \cdot k_i / n}{L} \cdot (k - t)$. Now, plugging the values in $b_{k_1, k_2, \ldots, k_e}$, we get

$$b_{k_1, k_2, \ldots, k_e} = \prod_{i=1}^{x} \left( e \cdot \frac{(a_i - t_i)}{k_i - t_i} \right)^{k_i - t_i} \cdot e^{-\frac{1}{k/n} \cdot d_i \cdot (a_i - t_i)} \cdot \left( \frac{d_i \cdot k}{n - d_{\text{max}}} \right)^{k_i - t_i}$$

$$\leq \prod_{i=1}^{x} \left( e \cdot \frac{L}{d_i \cdot e^{d_i \cdot k/n} \cdot (k - t)} \right)^{k_i - t_i} \cdot e^{-\frac{1}{k/n} \cdot d_i \cdot (a_i - t_i)} \cdot \left( \frac{d_i \cdot k}{n - d_{\text{max}}} \right)^{k_i - t_i}$$

$$= \prod_{i=1}^{x} \left( \frac{L}{n - d_{\text{max}}} \right)^{k_i - t_i} \cdot \left( e^{(k_i - t_i)} \cdot e^{-d_i \cdot (k/n) \cdot (k_i - t_i)} \cdot e^{-\frac{1}{k/n} \cdot d_i \cdot (a_i - t_i)} \right) \cdot \left( \frac{d_i \cdot k}{d_i \cdot (k - t)} \right)^{k_i - t_i}$$

(Rearranging denominators of first and third term, gathering powers of $e$ together)

$$= \left( \frac{L}{n - d_{\text{max}}} \right)^{\sum_{i=1}^{x} (k_i - t_i)} \cdot \left( e^{\sum_{i=1}^{x} (k_i - t_i)} \cdot e^{-\sum_{i=1}^{x} d_i \cdot (k/n) \cdot (a_i - t_i)} \right) \cdot \left( \frac{k}{(k - t)} \right)^{\sum_{i=1}^{x} (k_i - t_i)}$$

(Product is transformed to sum in exponent)

$$= \left( \frac{L}{n - d_{\text{max}}} \right)^{(k-t)} \cdot \left( e^{(k-t)} \cdot e^{-\sum_{i=1}^{x} d_i \cdot (a_i - t_i)} \right) \cdot \left( 1 + \frac{t}{(k - t)} \right)^{(k-t)}$$

(As $\sum_{i=1}^{x} k_i - t_i = k - t$)

$$\leq \left( \frac{L}{n - d_{\text{max}}} \right)^{k-t} \cdot e^{k-t} \cdot e^{-\sum_{i=1}^{x} d_i \cdot (a_i - t_i)} \cdot e^{t}$$

(Since $1 + x \leq e^x$ we have $\left( 1 + \frac{t}{k - t} \right) \leq e^{\frac{t}{k-t}}$)

$$= \left( \frac{L}{n - d_{\text{max}}} \right)^{-t} \cdot \left( \frac{L}{n - d_{\text{max}}} \right)^{\sum_{i=1}^{x} d_i \cdot (a_i - t_i)} \cdot e^{\frac{t}{k - t}}$$

(Arranging in powers by $t$ and $k$).

The desired result follows.

---

**Lemma 6.** Let $n$ be sufficiently large and let $c_1 \leq \frac{0.04}{d_{\text{max}}}$. Then for all $k \leq c_1 \cdot n$ we have

$$\left( \frac{L}{n - d_{\text{max}}} \cdot e^{1 - \sum_{i=1}^{x} d_i \cdot (a_i - t_i) / n} \right) \leq \frac{0.04}{10}.$$
Proof. We have the following inequality:

\[
\left( \frac{L}{n - d_{\max}} \cdot e^{d - 1} \cdot e^{\sum_{i=1}^{\infty} d_i \cdot (a_i - t_i)} \right) = \frac{\sum_{i=1}^{\infty} d_i \cdot (a_i - t_i) \cdot e^{d_i \cdot k/n}}{n - d_{\max}} \cdot e^{\sum_{i=1}^{\infty} d_i \cdot (a_i - t_i)} \]

\[
\leq \frac{e^{d_{\max} \cdot c_1}}{n - d_{\max}} \cdot \left( \sum_{i=1}^{\infty} d_i \cdot (a_i - t_i) \right) \cdot e^{\sum_{i=1}^{\infty} d_i \cdot (a_i - t_i)} \\
\text{(since } d_i \leq d_{\max} \text{ and } k \leq c_1 \cdot n) \\
\leq \frac{e^{d_{\max} \cdot c_1}}{n - d_{\max}} \cdot \frac{n}{n - d_{\max}} \cdot \sum_{i=1}^{\infty} d_i \cdot (a_i - t_i) \cdot e^{\sum_{i=1}^{\infty} d_i \cdot (a_i - t_i)} \\
\text{(multiplying numerator and denominator with } n) \\
= \frac{e^{d_{\max} \cdot c_1}}{n - d_{\max}} \cdot \frac{n}{n - d_{\max}} \cdot \frac{d}{e^{d - 1}} \\
\]

Here,

\[
d = \frac{1}{n} \sum_{i=1}^{\infty} d_i \cdot (a_i - t_i) \geq d_{\min} \cdot \frac{n - t}{n} \geq d_{\min} \cdot (1 - c_1) \geq 1
\]

The last inequality follows because \( c_1 \leq 0.5 \) and \( d_{\min} \geq 2 \). Since \( f(d) = d/e^{d-1} \) is a decreasing function for \( d \geq 1 \), we have \( f(d) \leq f(d_{\min} \cdot (1 - c_1)) \). Thus,

\[
\frac{e^{d_{\max} \cdot c_1}}{n - d_{\max}} \cdot \frac{d}{e^{d - 1}} \leq e^{d_{\max} \cdot c_1} \cdot \frac{n}{n - d_{\max}} \cdot \frac{d_{\min} \cdot (1 - c_1)}{e^{d_{\min} \cdot (1 - c_1) - 1}} \\
= e^{(d_{\min} + d_{\max}) \cdot c_1} \cdot \frac{n}{n - d_{\max}} \cdot \frac{d_{\min} \cdot (1 - c_1)}{e^{d_{\min} \cdot (1 - c_1) - 1}} \\
\leq e^{2d_{\max} \cdot c_1} \cdot \frac{n}{n - d_{\max}} \cdot \frac{2}{e} \cdot (1 - c_1) \leq 1 \text{ and } f(d_{\min}) \leq f(2) = 2/e \\
\leq 2 \cdot e^{-0.92} \cdot \frac{1}{0.9} \cdot \left( \frac{n}{n - d_{\max}} \leq \frac{1}{0.9} \text{ for sufficiently large } n \text{ and } c_1 \leq 0.04 \right) \\
\leq 0.9
\]

The desired result follows.

Lemma 7. For sufficiently large \( n \) and \( c_1 \leq 0.2 \) we have \( \frac{L}{n - d_{\max}} \geq 1 \).

Proof. We have the following inequality:

\[
L = \sum_{i=1}^{\infty} (a_i - t_i) \cdot d_i \cdot e^{d_i \cdot k/n} \\
\geq 2 \cdot \sum_{i=1}^{\infty} (a_i - t_i) \quad \text{(Since } e^{d_i \cdot k/n} \geq 1 \text{ and } d_i \geq d_{\min} \geq 2) \\
= 2 \cdot (n - t) \\
\geq 2 \cdot n \cdot (1 - c_1) \quad \text{(Since } t \leq c_1 \cdot n) \\
\geq 1.6 \cdot n \quad \text{(} c_1 \leq 0.2 \text{)}
\]

Also, \( n - d_{\max} < 1.6 \cdot n \) for large \( n \). The desired result follows.

Now we prove the bound on \( b_{k_1,k_2,\ldots,k_s} \).
Lemma 8 (Upper bound on $b_{k_1,k_2,...,k_x}$). There exists $c_1 > 0$ such that for sufficiently large $n$ and $t \leq k \leq c_1 \cdot n$, we have $b_{k_1,k_2,...,k_x} \leq \left( \frac{9}{10} \right)^k$.

Proof. Let $0 < c_1 \leq \frac{0.04}{d_{\max}} \leq 0.2$ as in Lemma 6. By Lemma 5 we have

$$b_{k_1,k_2,...,k_x} \leq \left( \frac{L}{n - d_{\max}} \right)^{-t} \cdot \left( \frac{L}{n - d_{\max}} \cdot e^{1 - \frac{\sum_{i=1}^{x} d_i (s_i - y_i)}{n}} \right)^k$$

By Lemma 7 we have $\left( \frac{L}{n - d_{\max}} \right) \geq 1$, and hence $\left( \frac{L}{n - d_{\max}} \right)^{-t} \leq 1$. By Lemma 6 we have

$$\frac{L}{n - d_{\max}} \cdot e^{1 - \frac{\sum_{i=1}^{x} d_i (s_i - y_i)}{n}} \leq \frac{9}{10}$$

The desired result follows trivially.

Taking appropriate bounds on the value of $k$, we get an upper bound on $a_{k_1,k_2,...,k_x}$.

Lemma 9 (Upper bound on $a_{k_1,k_2,...,k_x}$). There exists $c_1 > 0$ such that for $t \leq c_1 \cdot n$ and for all $30 \cdot x \cdot \log(n) \leq k \leq c_1 \cdot n$, we have $a_{k_1,k_2,...,k_x} < \frac{x}{n \cdot x}$, where $x = (d_{\max} - d_{\min} + 1)$.

Proof. By Lemma 4 we have $a_{k_1,k_2,...,k_x} \leq b_{k_1,k_2,...,k_x}$ and by Lemma 8 we have $b_{k_1,k_2,...,k_x} \leq \left( \frac{9}{10} \right)^k$. Thus for $k \geq 30 \cdot x \cdot \log(n)$,

$$a_{k_1,k_2,...,k_x} \leq \left( \frac{9}{10} \right)^{30 \cdot x \cdot \log(n)} = n^{30 \cdot x \cdot \log(9/10)} \leq \frac{1}{n^3}$$

The desired result follows.

Lemma 10 (Main lemma for small $k$). There exists $c_1 > 0$ such that for $t \leq c_1 \cdot n$ and for all $30 \cdot x \cdot \log(n) \leq k \leq c_1 \cdot n$, the probability that the size of the reverse reachable set $S$ is $k$ is at most $\frac{1}{n^2}$, where $x = (d_{\max} - d_{\min} + 1)$.

Proof. The probability that the reverse reachable set is of size $k$ is given by $\alpha_k$. By Lemma 3 and Lemma 9 it follows that the probability is at most $n^x \cdot n^{-3x} = n^{-2x} \leq \frac{1}{n^2}$. The desired result follows.

3.2 Large $k$: $c_1 \cdot n \leq k \leq c_2 \cdot n$

In this section we will show that for all constants $c_1$ and $c_2$, with $c_1 \leq c_2$, when $t \leq c_2 \cdot n$ the probability $\alpha_k$ is at most $\frac{1}{n^2}$ for all $c_1 \cdot n \leq k \leq c_2 \cdot n$. We start with a few notations that we will use in the proofs. Let $a_i = p_i \cdot n, t_i = y_i \cdot n, k_i = s_i \cdot n$ for $1 \leq i \leq x$ and $k = s \cdot n$ for $c_1 \leq s < c_2$. We first present a bound on $a_{k_1,k_2,...,k_x}$.

Lemma 11. For all $k_1,k_2,...,k_x$ such that $\sum_{i=1}^{x} k_i = k$, with $c_1 \cdot n \leq k \leq c_2 \cdot n$, we have

$$a_{k_1,k_2,...,k_x} \leq (n + 1)^x \cdot \text{Term}_1 \cdot \text{Term}_2,$$

where

$$\text{Term}_1 = \left( \prod_{i=1}^{x} \left( \frac{p_i - y_i}{s_i - y_i} \right)^{s_i - y_i} \left( \frac{p_i - y_i}{p_i - s_i} \right)^{p_i - s_i} \left( 1 - s \right)^{d_i (p_i - s_i)} \left( 1 - (1 - s)^{d_i} \right)^{s_i - y_i} \right)^n,$$

and

$$\text{Term}_2 = \prod_{i=1}^{x} \left( \frac{1 - \left( 1 - \frac{s_i}{s_i - y_i} \right)^{d_i}}{1 - (1 - s)^{d_i}} \right)^{n(s_i - y_i)}.$$
Proof. We have

\[ a_{k_1, k_2, \ldots, k_x} = \left( \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right) \cdot \left( \frac{(n-k)}{d_i} \right)^{a_i - k_i} \right) \cdot R(k_1, k_2, \ldots, k_x) \]

\[ \leq \left( \prod_{i=1}^{x} \left( \frac{a_i - t_i + 1}{k_i - t_i} \right) \cdot \left( \frac{a_i - t_i}{k_i - t_i} \right)^{-k_i - t_i} \cdot \left( \frac{a_i - t_i}{a_i - k_i} \right)^{a_i - k_i} \left( \frac{(n-k)}{d_i} \right)^{a_i - k_i} \right) \cdot R(k_1, k_2, \ldots, k_x) \]

(Applying second inequality of Proposition 1 with \( \ell = a_i - t_i \) and \( j = k_i - t_i \))

\[ \leq (n + 1)^x \cdot \left( \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right)^{k_i - t_i} \cdot \left( \frac{a_i - t_i}{a_i - k_i} \right)^{a_i - k_i} \left( \frac{(n-k)}{d_i} \right)^{a_i - k_i} \right) \cdot R(k_1, k_2, \ldots, k_x). \]

Proposition 1 is presented in the technical appendix. The last inequality above is obtained as follows: \((a_i - t_i + 1) \leq n + 1\) as \(a_i \leq n\). Our goal is now to show that

\[ Y = \left( \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right)^{k_i - t_i} \cdot \left( \frac{a_i - t_i}{a_i - k_i} \right)^{a_i - k_i} \left( \frac{(n-k)}{d_i} \right)^{a_i - k_i} \right) \cdot R(k_1, k_2, \ldots, k_x) \leq \text{Term}_1 \cdot \text{Term}_2. \]

We have (i) \(a_i - t_i = n(p_i - y_i)\); (ii) \(k_i - t_i = n(s_i - y_i)\); and (iii) \(a_i - k_i = n(p_i - s_i)\). Hence we have

\[ \prod_{i=1}^{x} \left( \frac{a_i - t_i}{k_i - t_i} \right)^{k_i - t_i} \cdot \left( \frac{a_i - t_i}{a_i - k_i} \right)^{a_i - k_i} \leq \prod_{i=1}^{x} \left( 1 - \frac{d_i \cdot n(p_i - s_i)}{n} \right)^{a_i - k_i} \]

By Lemma 2 we have

\[ \prod_{i=1}^{x} \left( \frac{n-k}{d_i} \right)^{a_i - k_i} \leq \prod_{i=1}^{x} \left( 1 - \frac{k}{n} \right)^{d_i \cdot n(p_i - s_i)} \]

By Lemma 1 we have

\[ R(k_1, k_2, \ldots, k_x) \leq \prod_{i=1}^{x} \left( 1 - \left( \frac{k}{n - d_i} \right) \right)^{n(s_i - y_i)} \]

Hence we have

\[ Y \leq \prod_{i=1}^{x} \left( \frac{p_i - y_i}{s_i - y_i} \right)^{n(s_i - y_i)} \left( \frac{p_i - y_i}{s_i - y_i} \right)^{n(p_i - s_i)} \left( 1 - \frac{k}{n} \right)^{d_i \cdot n(p_i - s_i)} \left( 1 - \left( \frac{k}{n - d_i} \right) \right)^{n(s_i - y_i)} \]

\[ = \prod_{i=1}^{x} \left( \frac{p_i - y_i}{s_i - y_i} \right)^{n(s_i - y_i)} \left( \frac{p_i - y_i}{p_i - s_i} \right)^{n(p_i - s_i)} \left( 1 - s \right)^{d_i \cdot n(p_i - s_i)} \left( 1 - \left( \frac{s}{1 - d_i/n} \right) \right)^{d_i \cdot n(s_i - y_i)} \]

\[ = \left( \prod_{i=1}^{x} \left( \frac{p_i - y_i}{s_i - y_i} \right)^{s_i - y_i} \left( \frac{p_i - y_i}{p_i - s_i} \right)^{p_i - s_i} \left( 1 - s \right)^{d_i(p_i - s_i)} \right)^n \cdot \left( \prod_{i=1}^{x} \left( 1 - \frac{s}{1 - d_i/n} \right)^{d_i \cdot n(s_i - y_i)} \right)^n \]

\[ = \left( \prod_{i=1}^{x} \left( \frac{p_i - y_i}{s_i - y_i} \right)^{s_i - y_i} \left( \frac{p_i - y_i}{p_i - s_i} \right)^{p_i - s_i} \left( 1 - s \right)^{d_i(p_i - s_i)} \left( 1 - \left( 1 - s \right)^{d_i} \right)^{s_i - y_i} \right)^n \]

\[ \cdot \left( \prod_{i=1}^{x} \left( \frac{1 - \left( 1 - s \right)^{d_i}}{1 - (1 - s)^{d_i}} \right) \right)^{n(s_i - y_i)} \]
The last equality is obtained by multiplying \((1 - (1 - s)d_i)^{n(s_i - y_i)}\) to \(X_1\) and dividing it from \(X_2\). Thus we obtain \(Y \leq \text{Term}_1 \cdot \text{Term}_2\), and the result follows.

**Lemma 12.** Term\(_2\) of Lemma 11, i.e., \(\prod_{i=1}^{\ell} \left(1 - \left(\frac{1 - \frac{s}{1-d_i/n}}{1 - (1-s)d_i}\right)^{1-d_i/(1-s)}\right)^{n(s_i - y_i)}\) is upper bounded by a constant.

**Proof.** We have

\[
\left(1 - \left(\frac{1 - \frac{s}{1-d_i/n}}{1 - (1-s)d_i}\right)^{1-d_i/(1-s)}\right)^{n(s_i - y_i)} \leq \left(1 - \left(1 - s(1 + \frac{2d_i}{n})\right)^{d_i}\right)^{n(s_i - y_i)}
\]

(for sufficiently large \(n\))

\[
\leq \left(1 - (1 - s)d_i + \left(\frac{d_i}{n}\right) \cdot \left(1 - (1-s)d_i\right)^{-1}\right)^{n(s_i - y_i)}
\]

(taking first two terms of binomial expansion)

\[
= \left(1 + \frac{(1-s)d_i, - \frac{2sd_i}{n}}{1-(1-s)d_i}\right)^{n(s_i - y_i)}
\]

\[\leq \left(1 + \frac{(1-s)d_i, - \frac{2sd_i}{n}}{1-(1-s)d_i}\right)^{n(s_i - y_i)}
\]

\[\leq \left(1 + \frac{(1-s)d_i, - \frac{2sd_i}{n}}{1-(1-s)d_i}\right)^{n(s_i - y_i)}
\]

\[\leq \left(1 + (1 + x) \leq \epsilon^2\right).
\]

Since \(c_1 \leq s \leq c_2\) we have \(s\) is constant, and similarly \(d_{\min} \leq d_i \leq d_{\max}\) and hence \(d_i\) is constant. Hence it follows that the above expression is constant and hence the product of those terms for \(1 \leq i \leq x\) is also bounded by a constant (since \(x\) is constant). The result follows.

**Lemma 13.** There exists \(0 < \eta < 1\) such that \(\text{Term}_1\) of Lemma 11 is smaller than \(\eta^n\) (exponentially small), i.e.,

\[
\prod_{i=1}^{\ell} \left(\frac{p_i - y_i}{s_i - y_i}\right)^{s_i - y_i} \left(\frac{p_i - y_i}{p_i - s_i}\right)^{p_i - s_i} (1 - s)^{d_i(p_i - s_i)} (1 - (1-s)^{d_i(s_i - y_i)}) \leq \eta^n
\]

**Proof.** Let

\[f(d_i) = \left(\frac{p_i - y_i}{s_i - y_i}\right)^{s_i - y_i} \left(\frac{p_i - y_i}{p_i - s_i}\right)^{p_i - s_i} (1 - s)^{d_i(p_i - s_i)} (1 - (1-s)^{d_i(s_i - y_i)})
\]

Note that \(f(d_i)\) is maximum when

\[\partial_{d_i} f(d_i) = 0 \Leftrightarrow d_i^* = \frac{\log \left(\frac{p_i - s_i}{p_i - y_i}\right)}{\log(1-s)}
\]

Moreover, \(f(d_i^*) = 1 \Rightarrow f(d_i) \leq 1\). If \(d_i = d_i^*\) for all \(i\), we have

\[d_i \geq 2 \Rightarrow (1 - s)^2 \geq \frac{p_i - s_i}{p_i - y_i} \Rightarrow (1 - s)^2 \geq \sum_{j \neq i} (p_j - s_j) \frac{p_i - s_i}{p_i - y_i} = \frac{1 - s}{1 - y} \Rightarrow (1 - s)(1 - y) \geq 1
\]

where the last inequality is impossible to satisfy since \(0 < s < 1\). Hence there exists \(i^*\) such that \(f(d_{i^*}) < 1\). Since \(d_{i^*} \in [d_{\min}, d_{\max}]\) has a compact domain and \(f\) is a continuous function, there exists a constant \(\eta < 1\) such that \(f(d_{i^*}) \leq \eta\). Since \(f(d_i) \leq 1\) for all \(i\), we have \(\prod_{i=1}^{\ell} f(d_i) \leq \eta^n\). The result thus follows.

**Lemma 14 (Main lemma for large \(k\)).** For all constants \(c_1\) and \(c_2\) with \(c_1 \leq c_2\), for all \(c_1 \cdot n \leq k \leq c_2 \cdot n\), and for all \(t \leq c_2 \cdot n\), the probability that the size of the reverse reachable set \(S\) is \(k\) is at most \(\frac{1}{n^t}\).

**Proof.** By Lemma 11 we have \(a_{k_1, k_2, \ldots, k_x} \leq (n + 1)^x \cdot \text{Term}_1 \cdot \text{Term}_2\), and by Lemma 12 and Lemma 13 we have \(\text{Term}_2\) is constant and \(\text{Term}_1\) is exponentially small in \(n\), where \(x = (d_{\max} - d_{\min} + 1)\). The exponentially small \(\text{Term}_1\) overrides the polynomial factor \((n + 1)^x\) and the constant \(\text{Term}_2\), and ensures that \(a_{k_1, k_2, \ldots, k_x} \leq n^{-3x}\). By Lemma 3 it follows that \(\alpha_k \leq n^{-2x} \leq \frac{1}{n^t}\).
3.3 Very large $k$: $\left(1 - 1/e^2\right)n$ to $n - d_{\text{min}} - 1$

In this subsection we consider the case when the size $k$ of the reverse reachable set is between $\left(1 - \frac{1}{e^2}\right) \cdot n$ and $n - d_{\text{min}} - 1$. Note that if the reverse reachable set has size at least $n - d_{\text{min}}$, then the reverse reachable set must be the set of all vertices, as otherwise the remaining vertices cannot have enough edges among themselves. Take $m = n - k$. Hence $d_{\text{min}} + 1 \leq m \leq n/e^2$. As stated earlier, in this case $a_{k_1, k_2, \ldots, k_x}$ becomes small since we require that the $m$ vertices outside the reverse reachable set must have all their edges within themselves; this corresponds to the factor of $\left(\frac{n - k}{d_i}\right)^{a_i - k_i}$. Since $m$ is very small, this has a very low probability. With this intuition, we proceed to show the following bound on $a_{k_1, k_2, \ldots, k_x}$.

Lemma 15. We have $a_{k_1, k_2, \ldots, k_x} \leq \left(x \cdot e \cdot \frac{m}{e}\right)^m$.

Proof. We have

$$a_{k_1, k_2, \ldots, k_x} = \prod_{i=1}^{x} \left(\frac{a_i - t_i}{k_i - t_i}\right) \left(\frac{n - k}{d_i}\right)^{a_i - k_i} \cdot R(k_1, k_2, \ldots, k_x)$$

$$\leq \prod_{i=1}^{x} \left(\frac{a_i - t_i}{k_i - t_i}\right) \left(\frac{n - k}{d_i}\right)^{a_i - k_i} \quad \text{(Ignoring probability value $R(k_1, k_2, \ldots, k_x) \leq 1$)}$$

$$\leq \prod_{i=1}^{x} \left(\frac{a_i - t_i}{a_i - k_i}\right) \left(\frac{n - k}{d_i}\right)^{a_i - k_i} \quad \text{(Since $\left(\frac{x}{y}\right) = \left(\frac{x}{x-y}\right)$)}$$

$$\leq \prod_{i=1}^{x} \left(\frac{e \cdot (a_i - t_i)}{a_i - k_i}\right)^{a_i - k_i} \left(\frac{n - k}{d_i}\right)^{a_i - k_i} \quad \text{(By Lemma 2)}$$

$$\leq \prod_{i=1}^{x} \left(\frac{e \cdot (a_i - t_i)}{a_i - k_i}\right)^{a_i - k_i} \left(\frac{n}{n - k}\right)^{d_i(a_i - k_i)} \quad \text{(Inequality 1 of Proposition 1)}$$

$$\leq e^m \cdot \left(\frac{m}{n}\right)^{2m} \prod_{i=1}^{x} \left(\frac{a_i - t_i}{a_i - k_i}\right)^{a_i - k_i} \quad \text{(Since $d_i \geq 2$ and } \sum_{i=1}^{x} (a_i - k_i) = m)$$

Proposition 1 is presented in the technical appendix. Since for all $i$ we have $(a_i - t_i) \leq n - t$, it follows that $\prod_{i=1}^{x} (a_i - t_i)^{a_i - k_i} \leq \prod_{i=1}^{x} (n - t)^{a_i - k_i} = (n - t)^m$. We also want a lower bound for $\prod_{i=1}^{x} (a_i - k_i)^{a_i - k_i}$. Note that $\sum_{i=1}^{x} (a_i - k_i) = m$ is fixed. Hence, this is a problem of minimizing $\prod_{i=1}^{x} y_i^{y_i}$ given that $\sum_{i=1}^{x} y_i = m$ is fixed. As before, this reduces to $\partial_{y_i} \prod_{i=1}^{x} y_i^{y_i} = \partial_{y_i} \prod_{i=1}^{x} y_i^{y_i}$ for all $a, b$. Hence, the minimum is attained at $y_i = m/x$, for all $i$. Hence, $\prod_{i=1}^{x} (a_i - k_i)^{a_i - k_i} \geq \left(\frac{m}{n}\right)^{2m}$. Combining these,

$$a_{k_1, k_2, \ldots, k_x} \leq e^m \cdot \left(\frac{m}{n}\right)^{2m} \cdot \prod_{i=1}^{x} \left(\frac{a_i - t_i}{a_i - k_i}\right)^{a_i - k_i}$$

$$\leq e^m \cdot \left(\frac{m}{n}\right)^{2m} \cdot \left(\frac{n - t}{\frac{m}{e}}\right)^m$$

$$\leq \left(x \cdot e \cdot \frac{m}{n}\right)^m$$

Hence we have the desired inequality.

We see that $(x \cdot e \cdot \frac{m}{e})^m$ is a convex function in $m$ and its maximum is attained at one of the endpoints. For $m = n/e^2$, the bound is exponentially decreasing with $n$ where as for constant $m$, the bound is polynomially decreasing in $n$. Hence, the maximum is attained at left endpoint of the interval (constant value of $m$). However, the bound we get is not sufficient to apply Lemma 3 directly. An important observation is that as $m$ becomes smaller and smaller, the number of combinations $\sum_{i=1}^{x} k_i = k$, $t_i \leq k_i \leq a_i$ in the expression of $a_i$ also decrease. Thus, we break this case into two sub-cases; $d_{\text{max}} + 1 < m \leq n/e^2$ and $d_{\text{min}} + 1 \leq m \leq d_{\text{max}} + 1$. 


Lemma 16. For \( d_{\text{max}} + 1 < m \leq n/e^2 \), we have \( a_{k_1, k_2, \ldots, k_x} < n^{-(2+x)} \) and \( \alpha_k \leq 1/n^2 \).

Proof. As we have seen, we only need to prove this for the value of \( m \) where \( a_{k_1, k_2, \ldots, k_x} \) attains its maximum i.e. \( m = d_{\text{max}} + 2 \). Note that \( d_{\text{max}} + 1 = x + d_{\text{min}} \geq x + 2 \). Hence,

\[
a_{k_1, k_2, \ldots, k_x} \leq \left( \frac{x \cdot e \cdot m}{n} \right)^m \quad \text{(By Lemma 15)}
\]

\[
\leq \left( \frac{x \cdot e \cdot d_{\text{max}} + 2}{n} \right)^{d_{\text{max}}+2}
\]

\[
= (x \cdot e \cdot (d_{\text{max}} + 2))^{d_{\text{max}}+2} \cdot n^{-(d_{\text{max}}+2)}
\]

\[
< n^{-(d_{\text{max}}+1)} \quad \text{(Since first term is a constant)}
\]

\[
\leq n^{-(2+x)}
\]

Hence we obtain the first inequality of the lemma. By Lemma 3 and the first inequality of the lemma we have \( \alpha_k \leq \frac{1}{n^2} \).

Lemma 17. There exists a constant \( h \) such that for \( d_{\text{min}} + 1 \leq m \leq d_{\text{max}} + 1 \), we have \( a_{k_1, k_2, \ldots, k_x} < h \cdot n^{-m} \) and \( \alpha_k \leq \frac{h}{n^2} \).

Proof. By Lemma 15 we have

\[
a_{k_1, k_2, \ldots, k_x} \leq \left( \frac{x \cdot e \cdot m}{n} \right)^m \leq (x \cdot e \cdot (d_{\text{max}} + 1))^{d_{\text{max}}+1} \cdot n^{-m}
\]

Let \( h = (x \cdot e \cdot (d_{\text{max}} + 1))^{d_{\text{max}}+1} \). Hence, first part is proved.

Now, for the second part, we note that since there are \( m \) vertices outside the reverse reachable set, and all their edges must be within these \( m \) vertices, they must have degree at most \( m - 1 \). Hence, there are now \( n \) vertices with at most \( m - d_{\text{min}} \) distinct degrees. Hence, in the summation \( \alpha_k = \sum_{k, t, k_t \leq k_1 \leq a_t} a_{k_1, k_2, \ldots, k_x} \), there are at most \( n^{m-d_{\text{min}}} \) terms. Thus we have

\[
\alpha_k \leq n^{m-d_{\text{min}}} \cdot h \cdot n^{-m} = h \cdot n^{-d_{\text{min}}} \leq \frac{h}{n^2}.
\]

The desired result follows.

Lemma 18 (Main lemma for very large \( k \)). For all \( t \), for all \( (1 - \frac{1}{\sqrt{e^2}}) \cdot n \leq k \leq n - 1 \), the probability that the size of the reverse reachable set \( S \) is \( k \) is at most \( O \left( \frac{1}{n^2} \right) \).

Proof. By Lemma 16 and Lemma 17 we obtain the result for all \( (1 - \frac{1}{\sqrt{e^2}}) \cdot n \leq k \leq n - d_{\text{min}} - 1 \). Since the reverse reachable set must contain all vertices if it has size at least \( n - d_{\text{min}} \), the result follows.

3.4 Expected Number of Iterations and Running Time

From Lemma 10, Lemma 14, and Lemma 18, we obtain that there exists a constant \( h \) such that

\[
\alpha_k \leq \frac{1}{n^2}, \quad a_k \leq \frac{1}{n^2}, \quad 30 \cdot x \cdot \log(n) \leq k < n - d_{\text{max}} - 1
\]

\[
a_k \leq \frac{h}{n^2}, \quad n - d_{\text{max}} - 1 \leq k \leq n - d_{\text{min}} - 1
\]

\[
a_k = 0 \quad n - d_{\text{min}} \leq k \leq n - 1
\]

Hence using the union bound we get the following result \( P(|S| < 30 \cdot x \cdot \log(n) \text{ or } |S| = n) \geq 1 - \frac{h}{n} \), where \( S \) is the reverse reachable set of target set (i.e., with probability at least \( 1 - \frac{h}{n} \) either at most \( 30 \cdot x \cdot \log(n) \) vertices reach the target set or all the vertices reach the target set). Let \( I(n) \) and \( T(n) \) denote the expected number of iterations and the
expected running time of the classical algorithm for MDPs on random graphs with \( n \) vertices and constant out-degree. Then from above we have
\[
I(n) \leq \left(1 - \frac{h}{n}\right) \cdot 30 \cdot x \cdot \log(n) + \frac{h}{n} \cdot n
\]
It follows that \( I(n) = O(\log(n)) \). For the expected running time we have
\[
T(n) \leq \left(1 - \frac{h}{n}\right) \cdot (30 \cdot x \cdot \log(n))^2 + \frac{h}{n} \cdot n^2
\]
It follows that \( T(n) = O(n) \). Hence we have the following theorem.

**Theorem 1.** The expected number of iterations and the expected running time of the classical algorithm for MDPs with Büchi objectives over graphs with constant out-degree are at most \( O(\log(n)) \) and \( O(n) \), respectively.

### 4 Average Case Analysis in Erdös-Rényi Model

In this section we consider the classical Erdös-Rényi model of random graphs \( G_{n,p} \), with \( n \) vertices, where each edge is chosen to be in the graph independently with probability \( p \) [11] (we consider directed graphs and then \( G_{n,p} \) is also referred as \( D_{n,p} \) in literature). First, in Section 4.1 we consider the case when \( p = \Omega \left(\frac{\log(n)}{n}\right) \), and then we consider the case when \( p = \frac{1}{4} \) (that generates the uniform distribution over all graphs). We will show two results: (1) if \( p \geq \frac{c \cdot \log(n)}{n} \), for any constant \( c > 2 \), then the expected number of iterations is constant and the expected running time is linear; and (2) if \( p = \frac{1}{4} \) (with \( p = \frac{1}{4} \) we consider all graphs to be equally likely), then the probability that the number of iterations is more than one falls exponentially in \( n \) (in other words, graphs where the running time is more than linear are exponentially rare).

#### 4.1 \( G_{n,p} \) with \( p = \Omega \left(\frac{\log(n)}{n}\right) \)

In this subsection we will show that given \( p \geq \frac{c \cdot \log(n)}{n} \), for any constant \( c > 2 \), the probability that not all vertices can reach the given target set is less than \( O(1/n) \). Hence the expected number of iterations of the classical algorithm for MDPs with Büchi objectives is constant and hence the algorithm works in average time linear in the size of the graph. Observe that to show the result the worst possible case is when the size of the target set is 1, as otherwise the chance that all vertices reach the target set is higher.

The probability \( R(n, p) \). For a random graph in \( G_{n,p} \) and a given target vertex, we denote by \( R(n, p) \) the probability that each vertex in the graph has a path along the directed edges to the target vertex. Our goal is to obtain a lower bound on \( R(n, p) \).

The key recurrence. Consider a random graph \( G \) with \( n \) vertices, with a given target vertex, and edge probability \( p \). For a set \( K \) of vertices with size \( k \) (i.e., \( |K| = k \)), which contains the target vertex, \( R(k, p) \) is the probability that each vertex in the set \( K \) has a path to the target vertex, that lies within the set \( K \) (i.e., the path only visits vertices in \( K \)). The probability \( R(k, p) \) depends only on \( k \) and \( p \), due to the symmetry among vertices.

Consider the subset \( S \) of all vertices in \( V \), which have a path to the target vertex. In that case, for all vertices \( v \) in \( V \setminus S \), there is no edge going from \( v \) to a vertex in \( S \) (otherwise there would have been a path from \( v \) to the target vertex). Thus there are no incoming edges from \( V \setminus S \) to \( S \). Let \( |S| = i \). Then the \( i \cdot (n - i) \) edges from \( V \setminus S \) to \( S \) should be absent, and each edge is absent with probability \( (1 - p) \). The probability that each vertex in \( S \) can reach the target is \( R(i, p) \). So the probability of \( S \) being the reverse reachable set is given by:
\[
(1 - p)^{i \cdot (n - i)} \cdot R(i, p).
\]
There are \( \binom{n-1}{i-1} \) possible subsets of \( i \) vertices that include the given target vertex, and \( i \) can range from 1 to \( n \). Exactly one subset \( S \) of \( V \) will be the reverse reachable set. So the sum of probabilities of the events that \( S \) is reverse reachable set is 1. Hence we have:
\[
1 = \sum_{i=1}^{n} \binom{n-1}{i-1} \cdot (1 - p)^{i \cdot (n - i)} \cdot R(i, p)
\]

15
Lemma 19. We have

\[ R(n, p) = 1 - \sum_{i=1}^{n-1} \binom{n-1}{i-1} \cdot (1-p)^{i(n-i)} \cdot R(i, p). \]  

(4)

Bound on \( p \) for lower bound on \( R(n, p) \). We will prove a lower bound on \( p \) in terms of \( n \) such that the probability that not all \( n \) vertices can reach the target vertex is less than \( O(1/n) \). In other words, we require

\[ R(n, p) \geq 1 - O\left(\frac{1}{n}\right) \]

(5)

Since \( R(i, p) \) is a probability value, it is at most 1. Hence from Equation 4 it follows that it suffices to show that

\[ \sum_{i=1}^{n-1} \binom{n-1}{i-1} \cdot (1-p)^{i(n-i)} \cdot R(i, p) \leq \sum_{i=1}^{n-1} \binom{n-1}{i-1} \cdot (1-p)^{i(n-i)} \leq O\left(\frac{1}{n}\right) \]  

(6)

to show that \( R(n, p) \geq 1 - O\left(\frac{1}{n}\right) \). We will prove a lower bound on \( p \) for achieving Equation 6. Let us denote by \( t_i = \binom{n-i}{i-1} \cdot (1-p)^{i(n-i)} \). The following lemma establishes a relation of \( t_i \) and \( t_{n-i} \).

**Lemma 19.** We have \( t_{n-i} = \frac{n-i}{i} \cdot t_i \).

**Proof.** We have

\[
t_{n-i} = \binom{n-1}{n-i-1} (1-p)^{i(n-i)} \\
= \binom{n-1}{i} \cdot (1-p)^{i(n-i)} \\
= \frac{n-i}{i} \binom{n-1}{i-1} (1-p)^{i(n-i)} \\
= \frac{n-i}{i} \cdot t_i
\]

The desired result follows.

Let \( g_i = t_i + t_{n-i}. \) From the previous lemma we have

\[
g_i = t_{n-i} + t_i = \frac{n}{i} \cdot t_i = \frac{n}{i} \cdot \binom{n-1}{i-1} \cdot (1-p)^{i(n-i)} = \binom{n}{i} \cdot (1-p)^{i(n-i)}. \]

We now establish a bound on \( g_i \) in terms of \( t_1 \). In the subsequent lemma we establish bound on \( t_1 \).

**Lemma 20.** If \( p \geq \frac{c \cdot \log(n)}{n} \) with \( c > 2 \), then for all \( 2 \leq i \leq \frac{n}{2} \) we have \( g_i \leq t_1 \).

**Proof.** Let \( p \geq \frac{c \cdot \log(n)}{n} \) with \( c > 2 \). Now

\[
t_1 = \frac{n}{i} \cdot (1-p)^{n-1} \geq \frac{1}{n^1 \cdot (1-p)^{(i-1)(n-i-1)}} \\
\geq \frac{1}{n^1 \cdot e^{c \cdot \frac{\log(n)}{n} \cdot (i-1)(n-i-1)}} \\
= n^{\frac{1}{2} (i-1)(n-i-1)-1} \text{ (Rearranging powers of } (1-p) \text{ and } \binom{n}{i} \leq n^i \text{)}
\]

(1 - x \leq e^{-x})
To show that \( t_1 \geq g_i \), it is sufficient to show that
\[
\frac{c}{n} (i - 1) \cdot (n - i - 1) - i \geq 0 \Leftrightarrow \frac{i \cdot n}{(i - 1) \cdot (n - i - 1)} \leq c
\]

Note that \( f(i) = \frac{\frac{i \cdot n}{(i - 1) \cdot (n - i - 1)}}{n \cdot (n - i)} \) is convex for \( 2 \leq i \leq n/2 \). Hence, its maximum value is attained at either of the endpoints. We can see that
\[
f(2) = \frac{2 \cdot n}{n - 3} \leq c \quad \text{(for sufficiently large } n \text{ and } c > 2)\]
and
\[
f(n/2) = 2 \cdot \left( \frac{n}{n - 2} \right)^2 \leq c \quad \text{(for sufficiently large } n \text{ and } c > 2)\]
The result follows.

**Lemma 21.** If \( p \geq \frac{n}{c \cdot \log(n)} \) with \( c > 2 \), then \( t_3 \leq \frac{1}{n^2} \) for \( n \) sufficiently large.

**Proof.** We have \( t_1 = (1 - p)^{n-1} \). For \( p \geq \frac{n}{c \cdot \log(n)} \) we have
\[
t_1 \leq \left( 1 - \frac{n}{c \cdot \log(n)} \right)^{n-1} \leq e^{-\frac{n}{c \cdot \log(n) \cdot (n-1)}} \quad \text{(Since } 1 - x \leq e^{-x})\]
\[
\leq e^{-2 \cdot \log(n)} = \frac{1}{n^2} \quad \text{(for sufficiently large } n \text{, } c > 2)\]
Hence, the desired result follows.

We are now ready to establish the main lemma that proves the upper bound on \( R(n, p) \) and then the main result of the section.

**Lemma 22.** For sufficiently large \( n \), for all \( p \geq \frac{n}{c \cdot \log(n)} \) with \( c > 2 \), we have \( R(n, p) \geq 1 - \frac{1.5}{n} \).

**Proof.** We first show that \( \sum_{i=1}^{n-1} t_i \leq \frac{1.5}{n} \). We have
\[
\sum_{i=1}^{n-1} t_i = t_1 + t_{n-1} + \sum_{i=2}^{n-2} t_i
\]
\[
= t_1 + t_{n-1} + \sum_{i=2}^{(n-2)/2} g_i
\]
\[
\leq n \cdot t_1 + \sum_{i=2}^{(n-2)/2} g_i \quad \text{(We apply } t_i + t_{n-i} = \frac{n}{i} \cdot t_i \text{ with } i = 1)\]
\[
\leq n \cdot t_1 + \sum_{i=2}^{(n-2)/2} t_1 \quad \text{(By Lemma 20 we have } g_i \leq t_1)\]
\[
= \frac{3 \cdot n}{2} \cdot t_1
\]
\[
\leq \frac{3 \cdot n}{2 \cdot n^2} \quad \text{(By Lemma 21 we have } t_1 \leq \frac{1}{n^2})\]

By Equation 6 we have that \( R(n, p) \geq 1 - \sum_{i=1}^{n-1} t_i \). It follows that \( R(n, p) \geq 1 - \frac{1.5}{n} \).

**Theorem 2.** The expected number of iterations of the classical algorithm for MDPs with Büchi objectives for random graphs \( G_{n,p} \), with \( p \geq \frac{n}{c \cdot \log(n)} \), where \( c > 2 \), is \( O(1) \), and the average case running time is linear.
Proof. By Lemma 22 it follows that $R(n, p) \geq 1 - \frac{1.5}{n}$, and if all vertices reach the target set, then the classical algorithm ends in one iteration. In the worst case the number of iterations of the classical algorithm is $n$. Hence the expected number of iterations is bounded by

$$1 \cdot \left(1 - \frac{1.5}{n}\right) + n \cdot \frac{1.5}{n} = O(1).$$

Since the expected number of iterations is $O(1)$ and every iteration takes linear time, it follows that the average case running time is linear.

4.2 Average-case analysis over all graphs

In this section, we consider uniform distribution over all graphs, i.e., all possible different graphs are equally likely. This is equivalent to considering the Erdős-Rényi model such that each edge has probability $\frac{1}{2}$. Hence we consider $G_{n, \frac{1}{2}}$ and show that the probability that not all vertices reach the target is exponentially small in $n$. It will follow that MDPs where the classical algorithm takes more than constant iterations is exponentially rare. We consider the same recurrence $R(n, p)$ as in the previous subsection and consider $t_k$ and $g_k$ as defined before. The following theorem shows the desired result.

Theorem 3. In $G_{n, \frac{1}{2}}$ with sufficiently large $n$ the probability that the classical algorithm takes more than one iteration is less than $(\frac{3}{4})^n$.

Proof. We first observe that Equation 4 and Equation 6 holds for all probabilities. Next we observe that Lemma 20 holds for $p \geq \frac{c \log(n)}{2n^2}$ and hence also for $p = \frac{1}{2}$ for sufficiently large $n$. Hence by applying the inequalities of the proof of Lemma 22 we obtain that

$$\sum_{i=1}^{n-1} t_i \leq \frac{3 \cdot n}{2} \cdot t_1.$$

For $p = \frac{1}{2}$ we have $t_1 = \binom{n-1}{0} \cdot (1 - \frac{1}{2})^{n-1} = \frac{1}{2^{n-1}}$. Hence we have

$$R(n, p) \geq 1 - \frac{3 \cdot n}{2 \cdot 2^{n-1}} > 1 - \frac{1.5^n}{2^n} = 1 - \left(\frac{3}{4}\right)^n.$$

The second inequality holds for sufficiently large $n$. It follows that the probability that the classical algorithm takes more than one iteration is less than $(\frac{3}{4})^n$. The desired result follows.

5 Conclusion

In this work we present the average case analysis of the classical algorithm for qualitative analysis of MDPs with Büchi objectives. Both for the general case and the important special case of MDPs with constant out-degree we establish that the average case running time is linear, as compared to the quadratic worst case complexity. Moreover, as for the improved algorithms it has been established that they require at most linear time more than the classical algorithm, it also follows that the average case running time of all the improved algorithms are also linear. As our results show many interesting analysis of random directed graphs and reachability properties, we believe our results will have other applications in average case analysis of related problems in verification.

References

1. C. Baier and J-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
2. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *FSTTCS 95*, volume 1026 of *LNCS*, pages 499–513. Springer-Verlag, 1995.
3. K. Chatterjee and M. Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In SODA. ACM-SIAM, 2011.
4. K. Chatterjee and M. Henzinger. An $O(n^2)$ algorithm for alternating Büchi games. In SODA. ACM-SIAM, 2012.
5. K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Simple stochastic parity games. In CSL’03, volume 2803 of LNCS, pages 100–113. Springer, 2003.
6. K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Quantitative stochastic parity games. In SODA’04, pages 121–130. SIAM, 2004.
7. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. Journal of the ACM, 42(4):857–907, 1995.
8. L. de Alfaro. Formal Verification of Probabilistic Systems. PhD thesis, Stanford University, 1997.
9. L. de Alfaro, M. Faella, R. Majumdar, and V. Raman. Code-aware resource management. In EMSOFT 05. ACM, 2005.
10. L. de Alfaro and P. Roy. Magnifying-lens abstraction for markov decision processes. In CAV, pages 325–338, 2007.
11. P. Erdős and A. Rényi. On the evolution of random graphs. Math. Inst. of the Hungarian Acad. of Sciences, pages 17–61, 1960.
12. J. Filar and K. Vrieze. Competitive Markov Decision Processes. Springer-Verlag, 1997.
13. H. Howard. Dynamic Programming and Markov Processes. MIT Press, 1960.
14. M. Kwiatkowska, G. Norman, and D. Parker. Verifying randomized distributed algorithms with prism. In Workshop on Advances in Verification (WAVE’00), 2000.
15. A. Pogosyants, R. Segala, and N. Lynch. Verification of the randomized consensus algorithm of Aspnes and Herlihy: a case study. Distributed Computing, 13(3):155–186, 2000.
16. M. L. Puterman. Markov Decision Processes. J. Wiley and Sons, 1994.
17. R. Segala. Modeling and Verification of Randomized Distributed Real-Time Systems. PhD thesis, MIT, 1995. Technical Report MIT/LCS/TR-676.
18. M.I.A. Stoelinga. Fun with FireWire: Experiments with verifying the IEEE1394 root contention protocol. In Formal Aspects of Computing, 2002.
19. W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, Handbook of Formal Languages, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.
6 Technical Appendix

Proposition 1 (Useful inequalities from Stirling inequalities). For natural numbers \( \ell \) and \( j \) with \( j \leq \ell \) we have the following inequalities:

1. \( \binom{\ell}{j} \leq \left( \frac{e^\ell}{j} \right)^j \).
2. \( \binom{\ell}{j} \leq (\ell + 1) \cdot \left( \frac{j}{e} \right)^j \cdot \left( \frac{\ell - j}{e} \right)^{\ell - j} \).

Proof. The proof of the results is based on the following inequality (Stirling inequality) for factorial:

\[ e \cdot \left( \frac{j}{e} \right)^j \leq j! \leq e \cdot \left( \frac{j + 1}{e} \right)^{j+1}. \]

We now use the inequality to show the desired inequalities:

1. We have

\[
\binom{\ell}{j} \leq \frac{\ell^j}{j!} \\
\leq \frac{\ell^j}{e \cdot j^j} \quad \text{(using Stirling inequality)} \\
\leq \frac{1}{e} \cdot \left( \frac{\ell}{j} \right)^j \\
\leq \left( \frac{\ell}{j} \right)^j
\]

2. We have

\[
\binom{\ell}{j} = \frac{\ell!}{j! \cdot (\ell - j)!} \\
\leq \left( e \cdot \left( \frac{\ell + 1}{e} \right)^{\ell+1} \right) \cdot \left( \frac{1}{e} \cdot \left( \frac{j}{e} \right)^j \cdot \left( \frac{\ell - j}{e} \right)^{\ell - j} \right) \\
= \frac{1}{e^2} \cdot (\ell + 1) \cdot \left( \frac{\ell + 1}{j} \right)^j \cdot \left( \frac{\ell + 1}{\ell - j} \right)^{\ell - j} \\
\leq \frac{1}{e^2} \cdot (\ell + 1) \cdot \left( \frac{\ell + 1}{\ell} \right)^{\ell} \cdot \left( \frac{j}{\ell} \right)^j \cdot \left( \frac{\ell}{\ell - j} \right)^{\ell - j} \\
\leq \frac{1}{e^2} \cdot (\ell + 1) \cdot e \cdot \left( \frac{j}{\ell} \right)^j \cdot \left( \frac{\ell}{\ell - j} \right)^{\ell - j} \quad \text{(Since } \left( \frac{1}{\ell} \right) \leq e \text{)} \\
\leq (\ell + 1) \cdot \left( \frac{j}{\ell} \right)^j \cdot \left( \frac{\ell}{\ell - j} \right)^{\ell - j}
\]

The first inequality is obtained by applying the Stirling inequality to the numerator (in the first term), and applying the Stirling inequality twice to the denominator (in the second term).