# Graph Planning with Expected Finite Horizon

Krishnendu Chatterjee[†] Laurent Doyen[§]
[†] IST Austria [§] CNRS & LSV, ENS Paris-Saclay, France



**Abstract**

Graph planning gives rise to fundamental algorithmic questions such as shortest path, traveling salesman problem, etc. A classical problem in discrete planning is to consider a weighted graph and construct a path that maximizes the sum of weights for a given time horizon $T$. However, in many scenarios, the time horizon is not fixed, but the stopping time is chosen according to some distribution such that the expected stopping time is $T$. If the stopping time distribution is not known, then to ensure robustness, the distribution is chosen by an adversary, to represent the worst-case scenario.

A stationary plan for every vertex always chooses the same outgoing edge. For fixed horizon or fixed stopping-time distribution, stationary plans are not sufficient for optimality. Quite surprisingly we show that when an adversary chooses the stopping-time distribution with expected stopping time $T$, then stationary plans are sufficient. While computing optimal stationary plans for fixed horizon is NP-complete, we show that computing optimal stationary plans under adversarial stopping-time distribution can be achieved in polynomial time. Consequently, our polynomial-time algorithm for adversarial stopping time also computes an optimal plan among all possible plans.

## 1 Introduction

*Graph search algorithms.* Reasoning about graphs is a fundamental problem in computer science, which is studied widely in logic (such as to describe graph properties with logic [6, 2]) and artificial intelligence [13, 9]. Graph search/planning algorithms are at the heart of such analysis, and gives rise to some of the most important algorithmic problems in computer science, such as shortest path, travelling salesman problem (TSP), etc.

*Finite-horizon planning.* A classical problem in graph planning is the *finite-horizon* planning problem [9], where the input is a directed graph with weights assigned to every edge and a time horizon $T$. The weight of an edge represents the reward/cost of the edge. A *plan* is an infinite path, and for finite horizon $T$ the utility of the plan is the sum of the weights of the first $T$ edges. An *optimal* plan maximizes the utility. The computational problem for finite-horizon planning is to compute the optimal utility and an optimal plan. The finite-horizon planning problem has many applications: the qualitative version of the problem corresponds to finite-horizon reachability, which plays an important role in logic and verification (e.g., bounded until in RTCTL, and bounded model-checking [3, 1]); and the more general quantitative problem of optimizing the sum of rewards has applications in artificial intelligence and robotics [13, Chapter 10, Chapter 25], and in control theory and game theory [4, Chapter 2.2], [11, Chapter 6].

*Solutions for finite-horizon planning.* For finite-horizon planning the classical solution approach is dynamic programming (or Bellman equations), which corresponds to backward induction [8, 4]. This approach not only works for graphs, but also for other models (e.g., Markov decision processes [12]). A *stationary* plan is a path where for every vertex always the same choice of edge is made. For finite-horizon planning, stationary plans are not sufficient for optimality, and in general, optimal plans are quite involved, and represented as transducers optimal plans require storage proportional to at least $T$ (see Example 1). Since in general optimal plans are involved, a related computational question is to compute effective simple plans, i.e., plans that are optimal among stationary plans.

| | arbitrary | stationary | arbitrary | stationary |
|---|---|---|---|---|
| Fixed horizon | PTIME | **NP-complete** | $O(T)$ | $O(\|V\|)$ |
| Expected horizon | **PTIME** | | $\mathbf{O}(\|\mathbf{V}\|)$ | |

Table 1: Computational complexity (left) and plan complexity (right). New results in boldface.

*Expected finite-horizon planning.* A natural variant of the finite-horizon planning problem is to consider expected time horizon, instead of the fixed time horizon. In the finite-horizon problem the allowed stopping time of the planning problem is a Dirac distribution at time $T$. In expected finite-horizon problem the *expected* stopping time is $T$. A well-known example where the fixed finite-horizon and the expected finite-horizon problems are fundamentally different is playing Prisoner's Dilemma: if the time horizon is fixed, then defection is the only dominant strategy, whereas for expected finite-horizon problem cooperation is feasible [10, Chapter 5]. Another classical example that is very well-studied is the notion of *discounting*, where at each time step the stopping probability is $\lambda$, and this corresponds to the case that the expected stopping time is $1/\lambda$ [4].

*Specified vs. adversarial distribution.* For the expected finite-horizon problem there are two variants: (a) *specified distribution:* the stopping-time distribution is specified; and (b) *adversarial distribution:* the stopping-time distribution is unknown and decided by an adversary. The expected finite-horizon problem with adversarial distribution represents the robust version of the planning problem, where the distribution is unknown and the adversary represents the worst-case scenario. Thus this problem presents the robust extension of the classical finite-horizon planning that has a wide range of applications.

*Results.* In this work we consider the expected finite-horizon planning problems in graphs. To the best of our knowledge this problem has not been studied in the literature.

- Our first simple result is that for the specified distribution problem, the optimal value can be computed in polynomial time (Theorem 1). However, since the specified distribution generalizes the fixed finite-horizon problem, the optimal plan description as an explicit transducer is of size $T$. Hence the output complexity is not polynomial in general. Second, we consider the decision problem whether there is a stationary plan to ensure a given utility. We show that this problem is NP-complete (Theorem 2).

Our most interesting and surprising results are for the adversarial distribution problem, which we describe below:

- We show that stationary plans suffice for optimality (Theorem 3). This result is surprising and counter-intuitive. Both in the classical finite-horizon problem and the specified distribution problem the adversary does not have any choice, and in both cases stationary plans do not suffice for optimality. Surprisingly we show that in the presence of an adversary the simpler class of stationary plans suffices for optimality.
- For the expected finite-horizon problem with adversarial distribution, the backward induction approach does not work, as there is no a-priori bound on the stopping time. We develop new algorithmic ideas to show that the optimal value can still be solved in polynomial time (Theorem 4). Moreover, our algorithm also computes and outputs an optimal stationary plan in polynomial time. Note that our algorithm also computes stationary optimal plans (which are as well optimal among all plans) in polynomial time, whereas computing stationary optimal plans for fixed finite horizon is NP-complete.

Our results are summarized in Table 1 and are relevant for synthesis of robust plans for expected finite-horizon planning.

## 2 Preliminaries

*Weighted graphs.* A *weighted graph* $G = \langle V, E, w \rangle$ consists of a finite set $V$ of vertices, a set $E \subseteq V \times V$ of edges, and a function $w\colon E \to \mathbb{Z}$ that assigns a weight to each edge of the graph.

*Plans and utilities.* A *plan* is an infinite *path* in $G$ from a vertex $v_0$, that is a sequence $\rho = e_0 e_1 \dots$ of edges $e_i = (v_i, v'_i) \in E$ such that $v'_i = v_{i+1}$ for all $i \geq 0$. A path induces the sequence of utilities $u_0, u_1, \dots$ where $u_i = \sum_{0 \leq k \leq i} w(e_k)$ for all $i \geq 0$. We denote by $U_G$ the set of all sequences of utilities induced by the paths of $G$. For finite paths $\rho = e_0 e_1 \dots e_k$ (i.e., finite prefixes of paths), we denote by $\mathsf{start}(\rho) = v_0$ and $\mathsf{end}(\rho) = v'_k$ the initial and last vertex of $\rho$, and by $|\rho| = k + 1$ the length of $\rho$.

*Plans as transducers.* A plan is described by a *transducer* (Mealy machine or Moore machine [7]) that given a prefix of the path (i.e., a finite sequence of edges) chooses the next edge. A *stationary* plan is a path where for every vertex the same choice of edge is made always. A stationary plan as a Mealy machine has one state, and as a Moore machine has at most $|V|$ states. Given a graph $G$ we denote by $S_G$ the set of all sequences of utilities induced by stationary plans in $G$.

*Distributions and stopping times.* A *sub-distribution* is a function $\delta\colon \mathbb{N} \to [0, 1]$ such that $p_\delta = \sum_{t \in \mathbb{N}} \delta(t) \in (0, 1]$. The value $p_\delta$ is the probability mass of $\delta$. Note that $p_\delta \neq 0$. The support of $\delta$ is $\mathsf{Supp}(\delta) = \{t \in \mathbb{N} \mid \delta(t) \neq 0\}$, and we say that $\delta$ is the sum of two sub-distributions $\delta_1$ and $\delta_2$, written $\delta = \delta_1 + \delta_2$, if $\delta(t) = \delta_1(t) + \delta_2(t)$ for all $t \in \mathbb{N}$. A *stopping-time distribution* (or simply, a distribution) is a sub-distribution with probability mass equal to 1. We denote by $\Delta$ the set of all stopping-time distributions, and by $\Delta^{\uparrow\uparrow}$ the set of all distributions $\delta$ with $|\mathsf{Supp}(\delta)| \leq 2$, called the bi-Dirac distributions.

*Expected utility and expected time.* The *expected utility* of a sequence $u = u_0, u_1, \dots$ of utilities under a sub-distribution $\delta$ is $\mathbb{E}_\delta(u) = \frac{1}{p_\delta} \cdot \sum_{t \in \mathbb{N}} u_t \cdot \delta(t)$. In particular, the expected utility of the identity sequence $0, 1, 2, \dots$ is called the *expected time*, denoted by $\mathbb{E}_\delta$.

## 3 Expected Finite-horizon: Specified Distribution

Given a stopping-time distribution $\delta$ with finite support, we show that the optimal expected utility can be computed in polynomial time. This result is straightforward.

**Theorem 1.** *Let $G$ be a weighted graph. Given a stopping-time distribution $\delta = \{(t_1, p_1), \dots, (t_k, p_k)\} \subseteq \mathbb{N} \times \mathbb{Q}$, with all numbers encoded in binary, the optimal expected utility $\sup_{u \in U_G} \mathbb{E}_\delta(u)$ can be computed in polynomial time.*

A special case of the problem in Theorem 1 is the fixed-length optimal path problem, which is to find an optimal path (that maximizes the total utility) of fixed length $T$, corresponding to the distribution $\delta = \{(T, 1)\}$. A pseudo-polynomial time solution is known for this problem, based on a value-iteration algorithm [9, Section 2.3]. The algorithm runs in time $O(T \cdot |V|^2)$ (where $T$ is encoded in binary), and relies on the following recursive relation, where $A_t(v)$ is the optimal value among the paths of length $t$ that start in $v$:

$$A_t(v) = \max_{v' \in V} \; w(v, v') + A_{t-1}(v').$$

A polynomial algorithm running in $O(\log(T) \cdot |V|^3)$ to obtain $A_T(v)$ is to compute, in the max-plus algebra[1], the $T$-th power of the transition matrix $M$ of the weighted graph, where $M_{ij} = w(i, j)$ if $(i, j) \in E$, and $M_{ij} = -\infty$ otherwise. The power $M^T$ can be computed in time $O(\log(T) \cdot |V|^3)$ by successive squaring of $M$ and summing up according to the binary representation of $T$, which gives a polynomial algorithm to compute $A_T(v)$ since it is the largest element in the column of $M^T$ corresponding to $v$ (note that the entries of the matrix $M^T$ are bounded by $|V| \cdot W$, where $W$ is the largest absolute weight in the graph). We now present the proof of Theorem 1.

[1] In the max-plus algebra, the matrix product $C = A \cdot B$ is defined by $C_{ij} = \max_k A_{ik} + B_{kj}$.

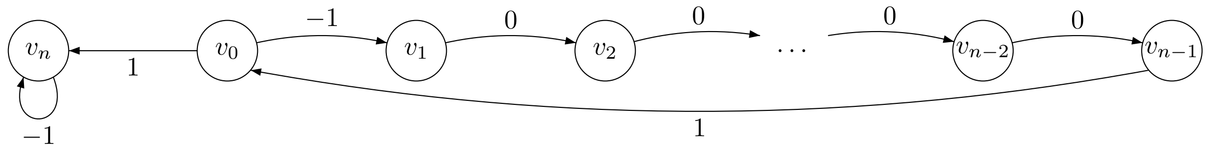



Figure 1: A weighted graph (with $n+1$ vertices) where the optimal path (of length $T = k \cdot n + 1$) is not simple: at $v_0$, the optimal plan chooses $k$ times the edge $(v_0, v_1)$, and then the edge $(v_0, v_n)$.

*Proof of Theorem 1.* Given the weighted graph $G = \langle V, E, w \rangle$ and the distribution $\delta = \{(t_1, p_1), \dots, (t_k, p_k)\}$, we reduce the problem to finding an optimal path of length $k$ in a layered graph $G'$ where the transitions between layer $i$ and layer $i+1$ mimic sequences of $t_{i+1} - t_i$ transitions in the original graph. For $t \geq 2$, define the $t$-th power of $E$ recursively by $E^t = \{(v_0, v_2) \mid \exists v_1 : (v_0, v_1) \in E \wedge (v_1, v_2) \in E^{t-1}\}$ where $E^1 = E$. Let $M$ be the transition matrix of the original weighted graph. We construct the graph $G' = \langle V', E', w' \rangle$ where

- $V' = V \times \{0, \dots, k\}$,
- $E' = \{(\langle v, i \rangle, \langle v', i+1 \rangle) \mid (v, v') \in E^{t_{i+1} - t_i} \wedge 0 \leq i < k\}$ where $t_0 = -1$, and
- $w'(\langle v, i \rangle, \langle v', i+1 \rangle) = (p_{i+1} + p_{i+2} + \dots + p_k) \cdot (M^{t_{i+1} - t_i})_{v,v'}$.

The optimal expected utility $\sup_{u \in U_G} \mathbb{E}_\delta(u)$ is the same as the optimal fixed-length path value for length $k$ in $G'$. The correctness of this reduction relies on the fact that the probability of not stopping before time $t_{i+1}$ is $p_{i+1} + p_{i+2} + \dots + p_k$ and the largest utility of a path of length $t_{i+1} - t_i$ from $v$ to $v'$ is $(M^{t_{i+1} - t_i})_{v,v'}$. Given a path $(v_0, v_1)(v_1, v_2) \dots (v_{k-1}, v_k)$ of length $k$ in $G'$ (that induces a sequence $w'_0 \dots w'_{k-1}$ of weights), we can construct a path of length $t_k + 1$ in $G$ (visiting $v_i$ at time $t_i$ and inducing a sequence $u$ of utilities), and we show that the value of the path of length $k$ in $G'$ is the same as the expected utility of the corresponding path in $G$ with stopping time distributed according to $\delta$, as follows (where $u_{t_0} = 0$):

$$
\begin{aligned}
\sum_{i=0}^{k-1} w'_i &= \sum_{i=0}^{k-1} \left( \sum_{j=i+1}^{k} p_j \right) \cdot (u_{t_{i+1}} - u_{t_i}) \\
&= \sum_{j=1}^{k} p_j \cdot \sum_{i=0}^{j-1} (u_{t_{i+1}} - u_{t_i}) \\
&= \sum_{j=1}^{k} p_j \cdot u_{t_j}
\end{aligned}
$$

Conversely, given an arbitrary path in $G$, let $v_i$ be the vertex visited at time $t_i$, and consider the path $(\langle v_0, 0 \rangle, \langle v_1, 1 \rangle)(\langle v_1, 1 \rangle, \langle v_2, 2 \rangle) \dots (\langle v_{k-1}, k-1 \rangle, \langle v_k, k \rangle)$ in $G'$, which has a total utility at least the same as the expected utility of the given path in $G$.

Therefore, the problem can be solved by finding the optimal fixed-length path value for length $k$ in $G'$, which can be done in polynomial time (see the remark after Theorem 1). ∎

In the fixed-horizon problem with $\delta = \{(T, 1)\}$, the optimal plan need not be stationary. The example below shows that in general the transducer for optimal plans require $O(T/|V|)$ states as Mealy machine, and $O(T)$ states as Moore machine.

**Example 1.** *Consider the graph of Figure 1 with $|V| = n+1$ vertices, and time bound $T = k \cdot n + 1$ (for some constant $k$). The optimal plan from $v_0$ is to repeat $k$ times the cycle $v_0, v_1, \dots, v_{n-1}$ and then switch to $v_n$. This path has value 1, and all other paths have lower value: if only the cycle $v_0, v_1, \dots, v_{n-1}$ is used,*

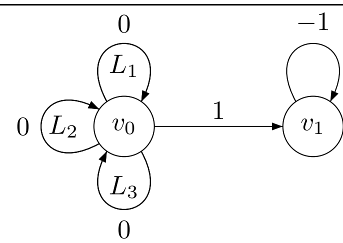



Figure 2: Three loops of respective length $L_1 = 6 = 2 \cdot 3$, $L_2 = 10 = 2 \cdot 5$, and $L_3 = 15 = 3 \cdot 5$. For $T = 32 = 6 + 10 + 15 + 1$, the optimal plan needs to visit each cycle once.

*then the value is at most* $0$*, and the same holds if the cycle on* $v_n$ *is ever used before time* $T$*. The optimal plan can be represented by a Mealy machine of size* $O(T/|V|)$ *that counts the number of cycle repetitions before switching to* $v_n$*. A Moore machine requires size* $T$ *as it needs a new memory state at every step of the plan.*

**Example 2.** *In the example of Figure 2 the optimal plan needs to visit several different cycles, not just repeating a single cycle and possible switching only at the end. The graph consists of three loops on* $v_0$ *with weights* $0$ *and respective length* $6$*,* $10$*, and* $15$*, and an edge to* $v_1$ *with weight* $1$*. For expected time* $T = 6 + 10 + 15 + 1$*, the optimal plan has value* $1$ *and needs to stop exactly when reaching* $v_1$ *(to avoid the negative self-loop on* $v_1$*). It is easy to show that the remaining length* $T - 1 = 31$ *can only be obtained by visiting each cycle once: as* $31$ *is not an even number, the path has to visit a cycle of odd length, thus the cycle of length* $15$*; analogously, as* $31$ *is not a multiple of* $3$*, the path has to visit the cycle of length* $10$*, etc. This example can be easily generalized to an arbitrary number of cycles by using more prime numbers.*

We now consider the complexity of computing optimal plans among stationary plans.

**Theorem 2.** *Let* $G$ *be a weighted graph and* $\lambda$ *be a rational utility threshold. Given a stopping-time distribution* $\delta$*, whether* $\sup_{u \in S_G} \mathbb{E}_\delta(u) \geq \lambda$ *(i.e., whether there is a stationary plan with utility at least* $\lambda$*) is NP-complete. The NP-hardness holds for the fixed-horizon problem* $\delta = \{(T, 1)\}$*, even when* $T$ *and all weights are in* $O(|V|)$*, and thus expressed in unary.*

*Proof.* The NP upper bound is easily obtained by guessing a stationary plan (i.e., one edge for each vertex of the graph) and checking that the value of the induced path is at least $\lambda$.

The NP hardness follows from a result of [5] where, given a directed graph $\mathcal{G}$ and four vertices $w, x, y, z$, the problem of deciding the existence of two (vertex) disjoint simple paths (one from $w$ to $x$ and the other from $y$ to $z$) is shown to be NP-complete. It easily follows that given a directed graph, and two vertices $v_1, v_2$, the problem of deciding the existence of a simple cycle that contains $v_1$ and $v_2$ is NP-complete. We present a reduction from the latter problem, illustrated in Figure 3. We construct a weighted graph from $\mathcal{G}$, by adding two vertices $\mathsf{start}$ and $\mathsf{sink}$, and all edges have weight 0 except those from $v_2$ with weight 1, and the edge $(v_1, \mathsf{sink})$ with weight $n + 1$ where $n$ is the number of vertices in $\mathcal{G}$. Let $T = n + 1$ and the utility threshold $\lambda = n + 2$.

If there exists a simple cycle containing $v_1$ and $v_2$ in $\mathcal{G}$, then there exists a stationary plan from $\mathsf{start}$ that visits $v_2$ then $v_1$ in at most $n$ steps. This plan can be prolonged to a plan of $n + 1$ steps by going to $\mathsf{sink}$ and using the self-loop. The total weight is $n + 2 = \lambda$.

If there is no simple cycle containing $v_1$ and $v_2$ in $\mathcal{G}$, then no stationary plan can visit first $v_2$ then $v_1$. We show that every stationary plan has value at most $n + 1 < \lambda$. First if a stationary plan uses the edge $(v_1, \mathsf{sink})$, then $v_2$ is not visited and all weights are 0 except the weight $n + 1$ from $v_1$ to $\mathsf{sink}$. Otherwise, if a stationary plan does not use the edge $(v_1, \mathsf{sink})$, then all weights are at most 1, and the total utility is at most $n + 1$. In both cases, the utility is smaller than $\lambda$, which establishes the correctness of the reduction. □

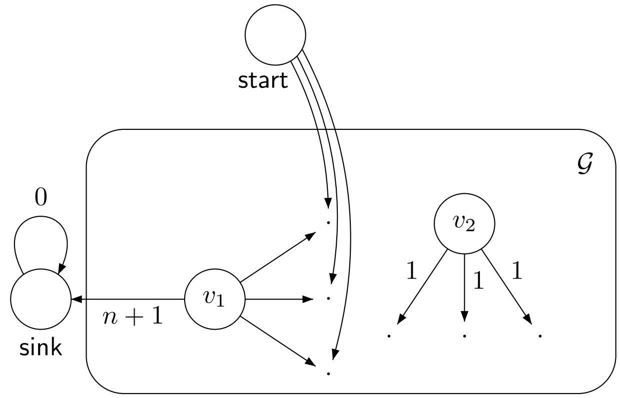



Figure 3: The NP-hardness reduction of Theorem 2.

# 4 Expected Finite-horizon: Adversarial Distribution

We now consider the computation of the following *optimal* values under adversarial distribution. Given a weighted graph $G$ and an expected stopping time $T \in \mathbb{Q}$, we define the following:

- *Optimal values of plans.* For a plan $\rho$ that induces the sequence $u$ of utilities, let

$$val(\rho, T) = val(u, T) = \inf_{\delta \in \Delta : \mathbb{E}_\delta = T} \mathbb{E}_\delta(u).$$

- *Optimal value.* The optimal value is the supremum value over all plans:

$$val(G, T) = \sup_{u \in U_G} val(u, T).$$

Our two main results are related to the plan complexity and a polynomial-time algorithm.

**Theorem 3.** *For all weighted graphs $G$ and for all $T$ we have*

$$val(G, T) = \sup_{u \in U_G} val(u, T) = \sup_{u \in S_G} val(u, T),$$

*i.e., optimal stationary plans exist for expected finite-horizon under adversarial distribution.*

**Remark 1.** *Note that in contrast to fixed finite-horizon problem, where stationary plans do not suffice, we show in the presence of an adversary, the simpler class of stationary plans are sufficient for optimality in expected finite-horizon. Moreover, while optimal plans require $O(T/|V|)$-size Mealy (resp., $O(T)$-size Moore) machines for fixed-length plans, our results show that under adversarial distribution optimal plans require $O(1)$-size Mealy (resp., $O(|V|)$-size Moore) machines.*

**Theorem 4.** *Given a weighted graph $G$ and expected finite-horizon $T$, whether $val(G, T) \geq 0$ can be decided in $O(|V|^{16} \cdot \log(T))$ time, and computing $val(G, T)$ can be done in $O(|V|^{16} \cdot \log(W \cdot |V|) \cdot \log(T))$ time.*

## 4.1 Theorem 3: Plan Complexity

In this section we prove Theorem 3. We start with the notion of sub-distributions. Two sub-distributions $\delta, \delta'$ are *equivalent* if they have the same probability mass, and the same expected time, that is $p_\delta = p_{\delta'}$ and $\mathbb{E}_\delta = \mathbb{E}_{\delta'}$. The following result is straightforward.

**Lemma 1.** *If $\delta_1, \delta_1'$ are equivalent sub-distributions, and $\delta_1 + \delta_2$ is a sub-distribution, then $\delta_1 + \delta_2$ and $\delta_1' + \delta_2$ are equivalent sub-distributions.*

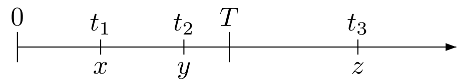



Figure 4: Timeline.

#### 4.1.1 Bi-Dirac distributions are sufficient

By Lemma 1, we can decompose distributions as the sum of two sub-distributions, and we can replace one of the two sub-distributions by a simpler (yet equivalent) one to obtain an equivalent distribution. We show that, given a sequence $u$ of utilities, for all sub-distributions with three points $t_1, t_2, t_3$ in their support (see Figure 4), there exists an equivalent sub-distribution with only two points in its support that gives a lower expected value for $u$. Intuitively, if one has to distribute a fixed probability mass (say 1) among three points with a fixed expected time $T$, assigning probability $p_i$ at point $t_i$, then we have $p_3 = 1 - p_1 - p_2$ and $p_1 \cdot t_1 + p_2 \cdot t_2 + p_3 \cdot t_3 = T$, i.e.,

$$\underbrace{p_1 \cdot (t_1 - t_3)}_{p'_1} + \underbrace{p_2 \cdot (t_2 - t_3)}_{p'_2} = T - t_3.$$

The expected utility is

$$p_1 \cdot u_{t_1} + p_2 \cdot u_{t_2} + p_3 \cdot u_{t_3} = p'_1 \cdot \frac{u_{t_1} - u_{t_3}}{t_1 - t_3} + p'_2 \cdot \frac{u_{t_2} - u_{t_3}}{t_2 - t_3} + u_{t_3}$$

which is a linear expression in variables $\{p'_1, p'_2\}$ where the sum $p'_1 + p'_2$ is constant. Hence the least expected utility is obtained for either $p'_1 = 0$, or $p'_2 = 0$. This is the main argument[2] to show that bi-Dirac distributions are sufficient to compute the optimal expected value.

**Lemma 2** (Bi-Dirac distributions are sufficient)**.** *For all sequences $u$ of utilities, for all time bounds $T$, the following holds:*

$$\inf\{\mathbb{E}_\delta(u) \mid \delta \in \Delta \wedge \mathbb{E}_\delta = T\} =$$
$$\inf\{\mathbb{E}_\delta(u) \mid \delta \in \Delta^{\upuparrows} \wedge \mathbb{E}_\delta = T\},$$

*i.e., the set $\Delta^{\upuparrows}$ of bi-Dirac distributions suffices for the adversary.*

*Proof.* First, we show that for all distributions $\delta \in \Delta$ with $\mathbb{E}_\delta = T$,

- $(i)$ there exists an equivalent distribution $\delta' \in \Delta$ such that $|\mathsf{Supp}(\delta') \cap [0, T-1]| \leq 1$ and $\mathbb{E}_{\delta'}(u) \leq \mathbb{E}_\delta(u)$, i.e., only one point before $T$ in the support is sufficient, and
- $(ii)$ there exists an equivalent distribution $\delta' \in \Delta$ such that $|\mathsf{Supp}(\delta') \cap [T, \infty)| \leq 1$ and $\mathbb{E}_{\delta'}(u) \leq \mathbb{E}_\delta(u)$, i.e., only one point after $T$ in the support is sufficient.

The result of the lemma follows from these two claims.

To prove claim $(i)$, first consider an arbitrary *sub-distribution* $\delta$ with $\mathsf{Supp}(\delta) = \{t_1, t_2, t_3\}$ where $t_1 < t_2 < t_3$. Then $t_1 < \mathbb{E}_\delta < t_3$ and either $\mathbb{E}_\delta \leq t_2$, or $t_2 \leq \mathbb{E}_\delta$.

We show that among the sub-distributions $\delta'$ equivalent to $\delta$ and with $\mathsf{Supp}(\delta') \subseteq \{t_1, t_2, t_3\}$, the smallest expected utility of $u$ is obtained for $\mathsf{Supp}(\delta') \subsetneq \{t_1, t_2, t_3\}$. We present below the argument in the case $t_2 \leq \mathbb{E}_\delta$, and show that either $\delta'(t_1) = 0$, or $\delta'(t_2) = 0$. A symmetric argument in the case $\mathbb{E}_\delta \leq t_2$ shows that either $\delta'(t_2) = 0$, or $\delta'(t_3) = 0$.

[2] This argument works here because $T > t_2$, which implies that $0 \leq p_2 \leq 1$ when $p_1 = 0$, and vice versa. A symmetric argument can be used in the case $T < t_2$, to show that then either $p_2 = 0$, or $p_3 = 0$.

Let $x = \delta'(t_1)$, $y = \delta'(t_2)$, and $z = \delta'(t_3)$. Since $\delta'$ and $\delta$ are equivalent, we have

$$
\begin{aligned}
x + y + z &= p_\delta \\
x \cdot t_1 + y \cdot t_2 + z \cdot t_3 &= p_\delta \cdot \mathbb{E}_\delta
\end{aligned}
$$

Hence

$$
\begin{aligned}
&z = p_\delta - x - y \\
&\underbrace{x \cdot (t_1 - t_3)}_{x'} + \underbrace{y \cdot (t_2 - t_3)}_{y'} = p_\delta \cdot (\mathbb{E}_\delta - t_3)
\end{aligned}
$$

The expected utility of $u$ under $\delta'$ is

$$
\begin{aligned}
\mathbb{E}_{\delta'}(u) &= x \cdot u_{t_1} + y \cdot u_{t_2} + z \cdot u_{t_3} \\
&= x \cdot (u_{t_1} - u_{t_3}) + y \cdot (u_{t_2} - u_{t_3}) + u_{t_3} \cdot p_\delta \\
&= x' \cdot \frac{u_{t_1} - u_{t_3}}{t_1 - t_3} + y' \cdot \frac{u_{t_2} - u_{t_3}}{t_2 - t_3} + u_{t_3} \cdot p_\delta \qquad (1)
\end{aligned}
$$

Since $x' + y'$ is constant and $x', y' \leq 0$, the least value of $\mathbb{E}_{\delta'}(u)$ is obtained either for $x' = 0$ (if $\frac{u_{t_1} - u_{t_3}}{t_1 - t_3} \leq \frac{u_{t_2} - u_{t_3}}{t_2 - t_3}$), or for $y' = 0$ (otherwise), thus either for $x = 0$, or for $y = 0$. Note that for $x = 0$, we have $y = \frac{p_\delta \cdot (\mathbb{E}_\delta - t_3)}{t_2 - t_3}$ and $z = \frac{p_\delta \cdot (t_2 - \mathbb{E}_\delta)}{t_2 - t_3}$, which is a feasible solution as $0 \leq y \leq 1$ and $0 \leq z \leq 1$ since $t_2 \leq \mathbb{E}_\delta \leq t_3$, and $0 < p_\delta \leq 1$. Symmetrically, for $y = 0$ we have a feasible solution.

As an intermediate remark, note that for $p_\delta = 1$ and $\mathbb{E}_\delta = T$, we get (for $y = y' = 0$, and symmetrically for $x = x' = 0$)

$$
\mathbb{E}_{\delta'}(u) = u_{t_3} + \frac{T - t_3}{t_1 - t_3} \cdot (u_{t_1} - u_{t_3}). \qquad (2)
$$

To complete the proof of Claim $(i)$, given an arbitrary distribution $\delta$ with $\mathbb{E}_\delta = T$, we use the above argument to construct a distribution equivalent[3] to $\delta$ with smaller expected utility and one less point in the support. We repeat this argument until we obtain a distribution $\delta'$ with support that contains at most two points in the interval $[0, k]$ where $k$ is such that $\sum_{i \leq k} \delta(i) \cdot i > T - 1$. Such a value of $k$ exists since $\mathbb{E}_\delta = \sum_{i \in \mathbb{N}} \delta(i) \cdot i = T$. By the construction of $\delta'$, we have $\sum_{i \leq k} \delta'(i) \cdot i > T - 1$ and therefore at most one point in the support of $\delta'$ lies in the interval $[0, T - 1]$, which completes the proof of Claim $(i)$.

To prove claim $(ii)$, consider a distribution $\delta$ with $\mathbb{E}_\delta = T$, and by claim $(i)$ we assume that $\delta(t_0) \neq 0$ for some $t_0 < T$, and $\delta(t) = 0$ for all $t < T$ with $t \neq t_0$. Let $\nu = \inf_{t \geq T} \frac{u_t - u_{t_0}}{t - t_0}$, and we consider two cases:

- if for all $t \geq T$ such that $t \in \mathsf{Supp}(\delta)$, we have $\frac{u_t - u_{t_0}}{t - t_0} = \nu$, then by an analogous of Equation (1), we get

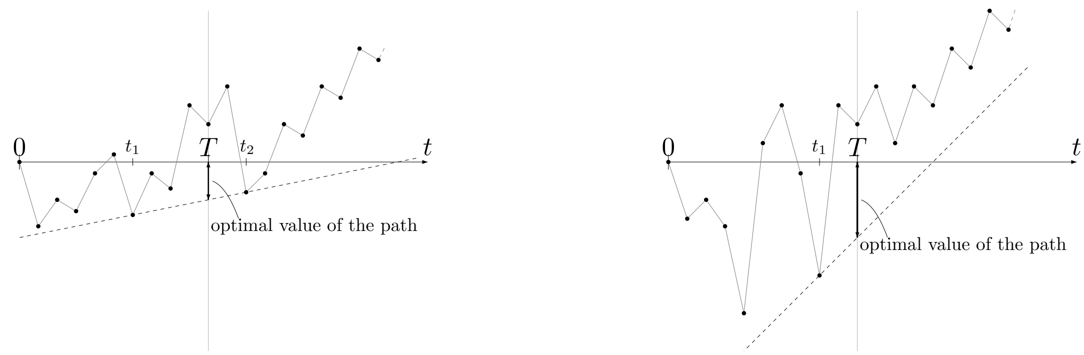
$$
\begin{aligned}
\mathbb{E}_\delta(u) &= u_{t_0} + \sum_{t \geq T} \delta(t) \cdot (t - t_0) \cdot \frac{u_t - u_{t_0}}{t - t_0} \\
&= u_{t_0} + \nu \cdot \sum_{t \geq 0} \delta(t) \cdot (t - t_0) = u_{t_0} + \nu \cdot (T - t_0)
\end{aligned}
$$

  which is the expected utility of $u$ under a bi-Dirac distribution with support $\{t_0, t\}$ where $t \geq T$ is any element of $\mathsf{Supp}(\delta)$ (see Equation (2));

- otherwise there exists $t \geq T$ such that $t \in \mathsf{Supp}(\delta)$ and $\frac{u_t - u_{t_0}}{t - t_0} > \nu$. By an analogous of Equation (1), we have

$$
\begin{aligned}
&\mathbb{E}_\delta(u) - u_{t_0} = \sum_{t \geq T} \delta(t) \cdot (t - t_0) \cdot \frac{u_t - u_{t_0}}{t - t_0} \\
&\text{where } \sum_{t \geq T} \delta(t) \cdot (t - t_0) = T - t_0,
\end{aligned}
$$

[3]Equivalence follows from Lemma 1.




(a) When an optimal distribution exists

(b) When no optimal distribution exists

Figure 5: Geometric interpretation of the value of a path.

that is $\frac{\mathbb{E}_\delta(u)-u_{t_0}}{T-t_0}$ is a convex combination of elements greater than or equal to $\nu$, among which one is greater than $\nu$. It follows that $\frac{\mathbb{E}_\delta(u)-u_{t_0}}{T-t_0} > \nu$, and thus there exists $\epsilon > 0$ such that $\frac{\mathbb{E}_\delta(u)-u_{t_0}}{T-t_0} > \nu + \epsilon$. Consider $t_1$ such that $\frac{u_{t_1}-u_{t_0}}{t_1-t_0} < \nu+\epsilon$ (which exists by definition of $\nu$), and let $\delta'$ be the bi-Dirac distribution $\delta'$ with support $\{t_0, t_1\}$ and expected time $T$. By an analogous of Equation (2), we have

$$\begin{aligned}\mathbb{E}_{\delta'}(u) - u_{t_0} &= \frac{T-t_0}{t_1-t_0} \cdot (u_{t_1} - u_{t_0}) \\ &< (T-t_0) \cdot (\nu+\epsilon) < \mathbb{E}_\delta(u) - u_{t_0}\end{aligned}$$

Therefore, $\mathbb{E}_{\delta'}(u) < \mathbb{E}_\delta(u)$ which concludes the proof since $\delta'$ is a bi-Dirac distribution with $\mathbb{E}_{\delta'} = T$.

□

### 4.1.2 Geometric interpretation

It follows from the proof of Lemma 2 (and Equation (2)) that the value of the expected utility of a sequence $u$ of utilities under a bi-Dirac distribution with support $\{t_1, t_2\}$ (where $t_1 < T < t_2$) and expected time $T$ is

$$u_{t_1} + \frac{T-t_1}{t_2-t_1} \cdot (u_{t_2} - u_{t_1}).$$

In Figure 5a, this value is obtained as the intersection of the vertical axis at $T$ and the line that connects the two points $(t_1, u_{t_1})$ and $(t_2, u_{t_2})$. Intuitively, the optimal value of a path is obtained by choosing the two points $t_1$ and $t_2$ such that the connecting line intersects the vertical axis at $T$ as down as possible.

**Lemma 3.** *For all sequences $u$ of utilities, if $u_t \geq a \cdot t + b$ for all $t \geq 0$, then the value of the sequence $u$ is at least $a \cdot T + b$.*

*Proof.* By Lemma 2, it is sufficient to consider bi-Dirac distributions, and for all bi-Dirac distributions $\delta$

with arbitrary support $\{t_1, t_2\}$ the value of $u$ under $\delta$ is

$$\begin{aligned}
& u_{t_1} + \frac{T - t_1}{t_2 - t_1} \cdot (u_{t_2} - u_{t_1}) \\
= & \frac{u_{t_1} \cdot (t_2 - T) + u_{t_2} \cdot (T - t_1)}{t_2 - t_1} \\
\geq & \frac{(a \cdot t_1 + b) \cdot (t_2 - T) + (a \cdot t_2 + b) \cdot (T - t_1)}{t_2 - t_1} \\
\geq & \, a \cdot T + b
\end{aligned}$$

□

It is always possible to fix an optimal value of $t_1$ (because $t_1 \leq T$ is to be chosen among a finite set of points), but the optimal value of $t_2$ may not exist, as in Figure 5b. The value of the path is then obtained as $t_2 \to \infty$. In general, there exists $t_1 \leq T$ such that it is sufficient to consider bi-Dirac distributions with support containing $t_1$ to compute the optimal value. We say that $t_1$ is a *left-minimizer* of the expected value in the path. Given such a value of $t_1$, let $\nu = \inf_{t_2 \geq T} \frac{u_{t_2} - u_{t_1}}{t_2 - t_1}$, and we show in Lemma 4 that $u_t \geq u_{t_1} + (t - t_1) \cdot \nu$, for all $t \geq 0$. This motivates the following definition.

*Line of equation $f_u(t)$.* Given a left-minimizer $t_1$, we define the line of equation $f_u(t)$ as follows:

$$f_u(t) = u_{t_1} + (t - t_1) \cdot \nu.$$

Note that the optimal expected utility is

$$\min_{0 \leq t_1 \leq T} \inf_{t_2 \geq T} u_{t_1} + \frac{T - t_1}{t_2 - t_1} \cdot (u_{t_2} - u_{t_1}) = \min_{0 \leq t_1 \leq T} u_{t_1} + (T - t_1) \cdot \nu = f_u(T).$$

In other words, $f_u(T)$ is the optimal value.

**Lemma 4** (Geometric interpretation)**.** *For all sequences $u$ of utilities, we have $u_t \geq f_u(t)$ for all $t \geq 0$, and the expected value of $u$ is $f_u(T)$.*

*Proof.* The result holds by definition of $\nu$ for all $t \geq T$. For $t < T$, assume towards contradiction that $u_t < u_{t_1} + (t - t_1) \cdot \nu$. Let $\varepsilon > 0$ such that $u_t = u_{t_1} + (t - t_1) \cdot \nu - \varepsilon$. We obtain a contradiction by showing that there exists a bi-Dirac distribution under which the expected value of $u$ is smaller than the optimal value of $u$. Consider a bi-Dirac distribution with support $\{t, t_2\}$ where the value $t_2$ is defined later.

We need to show that

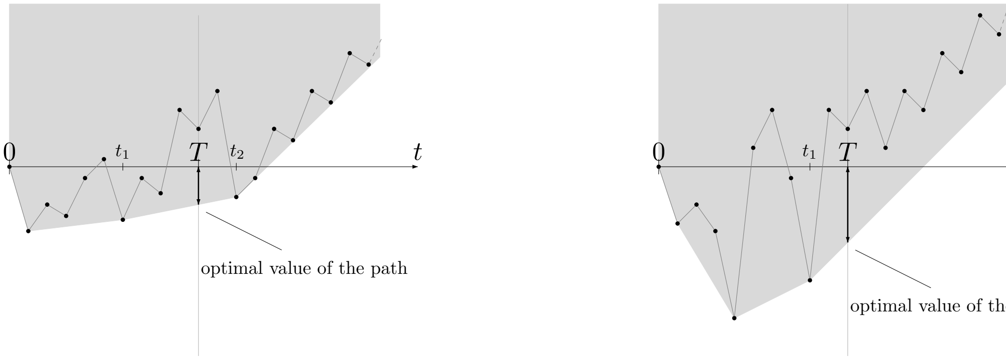
$$u_t + \frac{T - t}{t_2 - t} \cdot (u_{t_2} - u_t) < u_{t_1} + (T - t_1) \cdot \nu,$$

that is

$$\frac{u_t \cdot (t_2 - T) + u_{t_2} \cdot (T - t)}{t_2 - t} < u_{t_1} + (T - t_1) \cdot \nu$$

which, since $u_t = u_{t_1} + (t - t_1) \cdot \nu - \varepsilon$, holds if (successively)

$$\begin{aligned}
& u_{t_1} \cdot (t_2 - T) + (t - t_1) \cdot (t_2 - T) \cdot \nu + u_{t_2} \cdot (T - t) \leq \\
& \qquad \varepsilon \cdot (t_2 - T) + u_{t_1} \cdot (t_2 - t) + (t_2 - t) \cdot (T - t_1) \cdot \nu \\
& u_{t_1} \cdot (t - T) + u_{t_2} \cdot (T - t) \leq \\
& \qquad \varepsilon \cdot (t_2 - T) - \nu \cdot (t \cdot t_2 + t_1 \cdot T - t_2 \cdot T - t \cdot t_1) \\
& (u_{t_2} - u_{t_1}) \cdot (T - t) + \nu \cdot (t_2 - t_1) \cdot (t - T) \leq \\
& \qquad \varepsilon \cdot (t_2 - T) \\
& (T - t) \cdot \left( \tfrac{u_{t_2} - u_{t_1}}{t_2 - t_1} - \nu \right) \cdot (t_2 - t_1) \leq \varepsilon \cdot (t_2 - T)
\end{aligned}$$




(a) For the example of Figure 5a. (b) For the example of Figure 5b.

Figure 6: Convex hull interpretation of the value of a path.

We consider two cases: $(i)$ if the infimum $\nu$ is attained, then we have $\nu = \frac{u_{t_2} - u_{t_1}}{t_2 - t_1}$ for some $t_2 \geq T$, and the inequality holds; $(ii)$ otherwise, we can choose $t_2$ arbitrarily, and large enough to ensure that $(T - t) \cdot \left( \frac{u_{t_2} - u_{t_1}}{t_2 - t_1} - \nu \right)$ is smaller than $\frac{\varepsilon}{2}$, so that the inequality holds. □

A corollary of the geometric interpretation lemma is that the value of a path can be obtained as the intersection of the vertical line at point $T$ with the boundary of the convex hull of the region above the sequence of utilities, namely $convexHull(\{(t, y) \in \mathbb{N} \times \mathbb{R} \mid y \geq u_t\})$. This result is illustrated in Figure 6.

### 4.1.3 Simple lassos are sufficient

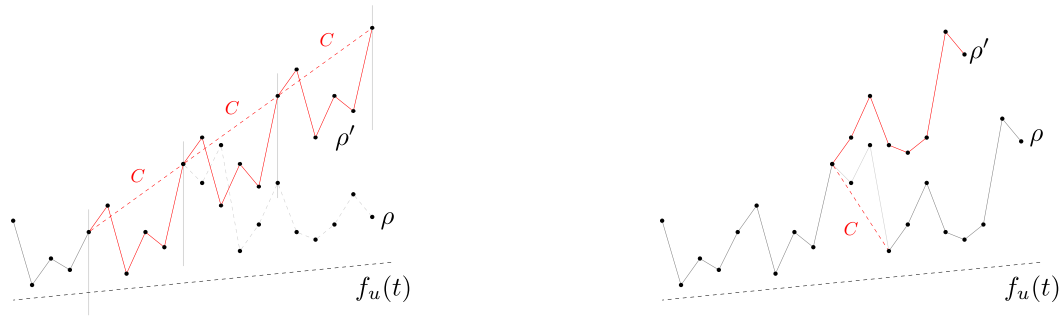
A *lasso* is a path of the form $AC^\omega$ where $A$ and $C$ are finite paths (with $C$ a nonempty cycle), where $AC^\omega$ is $A$ followed by infinite repetition of the cycle $C$. A lasso is *simple* if all strict prefixes of the finite path $AC$ are acyclic. In other words, simple lassos correspond to stationary plans.

We show that there is always a simple lasso with optimal value. Our proof has four steps. Given a path $\rho$ that gives the utility sequence $u$, let $\nu$ be the slope of $f_u(t)$. Given a cycle $C$ in the path $\rho$, let $S_C$ be the sum of the weights in $C$ and let $M_C = \frac{S_C}{|C|}$ be the average weight of the cycle edges. The cycle $C$ is *good* if $M_C \geq \nu$, i.e., the average weight of the cycle is at least $\nu$, and *bad* otherwise.

- First, we show (in Lemma 5) that every path contains a good cycle.
- Second, we show (in Lemma 6) that if the first cycle in a path is good, then repeating the cycle cannot decrease the value of the path.
- Third, we show (in Lemma 7) that removing a bad cycle from a path cannot decrease the value of the path.
- Finally, we show (in Lemma 8) that given any path, using the above two operations of removal of bad cycles and repetition of good cycles, we obtain a simple lasso that does not decrease the value of the original path.

Thus we establish that simple lassos (or stationary plans) are sufficient for optimality. To formalize the ideas we consider the notion of cycle decomposition.

*Cycle decomposition.* The *cycle decomposition* of a path $\rho = e_0 e_1 \ldots$ is an infinite sequence of simple cycles $C_1, C_2, \ldots$ obtained as follows: push successively $e_0, e_1, \ldots$ onto a stack, and whenever we push an edge that closes a (simple) cycle, we remove the cycle from the stack and append it to the cycle decomposition. Note that the stack content is always a prefix of a path of length at most $|V|$.




(a) Repeating a good cycle (Lemma 6).

(b) Removing a bad cycle (Lemma 7).

Figure 7: Constructing a lasso without decreasing the value (Lemma 6 and Lemma 7).

**Lemma 5.** *Let $T \in \mathbb{N}$. Given a path $\rho$ that induces a sequence $u$ of utilities, let $\nu = \min_{0 \le t_1 \le T} \inf_{t_2 \ge T} \frac{u_{t_2} - u_{t_1}}{t_2 - t_1}$. Then, in the cycle decomposition of $\rho$ there exists a simple cycle $C$ with $M_C \ge \nu$.*

*Proof.* Towards contradiction, assume that all the (finitely many) cycles $C$ in the cycle decomposition of $\rho$ are such that $M_C < \nu$. Let $t_1$ be a left-minimizer of $\rho$. Since all cycles in $\rho$ have average weight smaller than $\nu$, we have:

$$\liminf_{t_2 \to \infty} \frac{u_{t_2} - u_{t_1}}{t_2 - t_1} < \nu$$

Since the infimum is bounded by the liminf, it follows that

$$\min_{0 \le t_1 \le T} \inf_{t_2 \ge T} \frac{u_{t_2} - u_{t_1}}{t_2 - t_1} < \nu$$

which is in contradiction with the definition of $\nu$. $\square$

We show that repeating a good cycle, and removing a bad cycle from a path cannot decrease the value of the path.

**Lemma 6.** *Let $T \in \mathbb{N}$. If the first cycle $C$ in the cycle decomposition of a path $\rho$ is good, i.e., $M_C \ge \nu$ where $\nu = \min_{0 \le t_1 \le T} \inf_{t_2 \ge T} \frac{u_{t_2} - u_{t_1}}{t_2 - t_1}$, then there exists a lasso $\rho'$ such that $val(\rho', T) \ge val(\rho, T)$.*

*Proof.* Let $u$ be the sequence of utilities induced by $\rho$. Since $C$ is the first cycle in $\rho$, there is a prefix of $\rho$ of the form $AC$ where $A$ is a finite path. Consider the lasso $\rho' = AC^\omega$ and its induced sequence of utilities $u'$.

We show that the value of $\rho'$ is at least the value of $\rho$. By Lemma 4, the optimal value of $u$ is $f_u(T)$, and the sequence $u$ is above the line $f_u(t)$ (which has slope $\nu$), i.e., $u(t) \ge f_u(t)$ for all $t \ge 0$. By Lemma 3 it is sufficient to show that $u'$ is above the line $f_u(t)$ to establish that the optimal value of $u'$ is at least $f_u(T)$, that is $val(\rho', T) \ge val(\rho, T)$, and conclude the proof (the argument is illustrated in Figure 7a).

We show that $u'(t) \ge f_u(t)$ for all $t \ge 0$: either $t \le |A| + |C|$, and then $u'(t) = u(t) \ge f_u(t)$, or $t > |A| + |C|$, and then let $k \in \mathbb{N}$ such that $|A| \le t - k \cdot |C| \le |A| + |C|$, and we have

$$\begin{aligned}
u'(t) &= u(t - k \cdot |C|) + k \cdot S_C && (\rho' = AC^\omega) \\
&\ge f_u(t - k \cdot |C|) + k \cdot M_C \cdot |C| && (u \text{ is above } f_u(t) \text{ and } S_C = M_C \cdot |C|) \\
&\ge f_u(t) - \nu \cdot k \cdot |C| + k \cdot M_C \cdot |C| && (f_u(t) \text{ is linear with slope } \nu) \\
&\ge f_u(t) + k \cdot |C| \cdot (M_C - \nu) \\
&\ge f_u(t). && (M_C \ge \nu)
\end{aligned}$$

□

**Lemma 7.** *Let $T \in \mathbb{N}$. If a path $\rho$ contains a bad cycle $C$, that is such that $M_C < \nu$ where $\nu = \min_{0 \le t_1 \le T} \inf_{t_2 \ge T} \frac{u_{t_2} - u_{t_1}}{t_2 - t_1}$, then removing $C$ from $\rho$ gives a path $\rho'$ such that $val(\rho', T) \ge val(\rho, T)$.*

*Proof.* Let $u, u'$ be the sequences of utilities induced by respectively $\rho$ and $\rho'$, By the same argument as in the proof of Lemma 6 (using Lemma 3 and Lemma 4), it is sufficient to show that $u'$ is above the line $f_u(t)$. Since $C$ is a cycle in $\rho$, there is a prefix of $\rho$ of the form $AC$ where $A$ is a finite path, and for all $t \ge 0$ we have (the argument is illustrated in Figure 7b): either $t \le |A|$, then $u'(t) = u(t) \ge f_u(t)$, or $t > |A|$, and then

$$\begin{aligned}
u'(t) &= u(t + |C|) - S_C && (C \text{ is removed from } \rho \text{ to get } \rho') \\
&\ge f_u(t + |C|) - M_C \cdot |C| && (u \text{ is above } f_u(t) \text{ and } S_C = M_C \cdot |C|) \\
&\ge f_u(t) + \nu \cdot |C| - M_C \cdot |C| && (f_u(t) \text{ is linear with slope } \nu) \\
&\ge f_u(t) + |C| \cdot (\nu - M_C) \\
&\ge f_u(t). && (M_C < \nu)
\end{aligned}$$

□

Now we can show how to construct a simple lasso with value at least the value of a given arbitrary path, and it follows that simple lassos are sufficient for optimality.

**Lemma 8.** *Let $T \in \mathbb{N}$. There exists a simple lasso $AC^\omega$ such that $val(AC^\omega, T) = val(G, T)$.*

*Proof.* Given an arbitrary path $\rho$, we construct a simple lasso with at least the same value as $\rho$. It follows that the optimal value is obtained by stationary plans. The construction repeats the following steps:

1. Let $C$ be the first cycle in the cycle decomposition of $\rho$;
2. if $C$ is a bad cycle for the original path $\rho$, then we remove it to obtain a new path $\rho'$. We continue the procedure with $\rho'$ (go to step 1.);
3. otherwise $C$ is a good cycle for the original path $\rho$. Let $A$ be the prefix of $\rho$ until $C$ starts, and we construct the lasso $AC^\omega$.

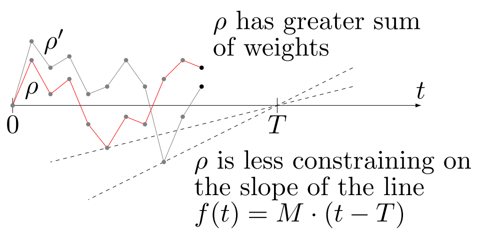
First, note that if the above procedure terminates, then the constructed lasso has a value at least the value of the original path $\rho$ (by Lemma 6 and Lemma 7), and it is a simple lasso by definition of the cycle decomposition.

Now we show that the procedure always terminates. By Lemma 5, there always exists a good cycle in the cycle decomposition of $\rho$, and thus eventually a good cycle becomes the first cycle in the path constructed by the above procedure, which then terminates. □

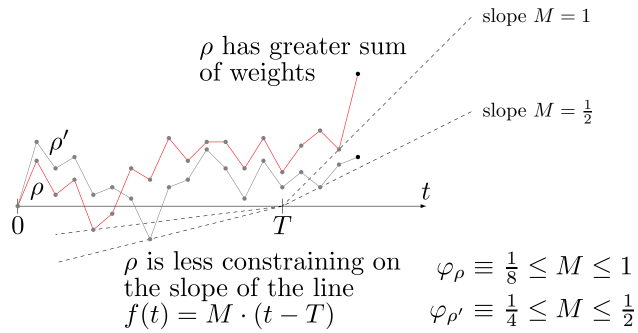
Theorem 3 follows from the above lemmas.

## 4.2 Theorem 4: Algorithm and Complexity Analysis

In this section we present our algorithm and then the complexity analysis.




(a) The path length is smaller than $T$.




(b) The path length is greater than $T$.

Figure 8: The path $\rho$ is preferred to $\rho'$.

### 4.2.1 Algorithm

The key challenges to obtain an algorithm are as follows. First, while for the fixed-horizon problem backward induction or powering of transition matrix leads to an algorithm, for expected time horizon with an adversary, there is no a-priori bound on the number of steps, and hence the backward induction approach is not applicable. Second, stationary optimal plans suffice, and as shown in Theorem 2 computing optimal stationary plans for the fixed horizon problem is NP-hard. We present an algorithm that iteratively constructs the *most promising* candidate paths according to a partial order of the paths, and the key is to define the partial order.

It follows from the geometric interpretation lemmas (Lemma 3 and Lemma 4) that the value of a path is at least 0 if its sequence of utilities is above some line that contains the point $(T, 0)$.

**Lemma 9.** *The value of a sequence $u$ of utilities is at least $0$ if and only if there exists a slope $M \in \mathbb{R}$ such that $u_t \geq M \cdot (t - T)$ for all $t \geq 0$.*

*Proof.* If the value of $u$ is at least 0, then $f_u(T) \geq 0$ and by Lemma 4 we have $u_t \geq f_u(t)$ for all $t \geq 0$. Then $u_t \geq f_u(t) - f_u(T)$ (which is a linear function of $t$) and we can take for $M$ the value of the coefficient of $t$ in the expression $f_u(t) - f_u(T)$.

To prove the other direction, consider the line of equation $f(t) = M \cdot (t - T)$, and by Lemma 3, the value of the sequence $u$ is at least $f(T) = 0$. □

The expression $u_t - M \cdot (t - T)$ that appears in the condition of Lemma 9 can be obtained by subtracting $M$ to each weight of the graph, and shifting the sum of the weights by the constant $T \cdot M$. Since $M$ is unknown, we can define the following symbolic constraint on $M$ (associated with a path $\rho$) that ensures, if it is satisfiable, that the sequence of utilities of $\rho = e_0 e_1 \dots e_k$ is above the line of equation $f(t) = M \cdot (t - T)$ :

$$\varphi_\rho \equiv \bigwedge_{0 \leq i \leq k} (u_i \geq M \cdot (i - T))$$

Note that $k = |\rho| - 1$, and the constraint $\varphi_\rho$ represents an interval (possibly empty, possibly unbounded) of values for $M$. Intuitively, a finite path is more promising (thus preferred) in order to be prolonged to an infinite path with value at least 0 if the total sum of weights is large and the constraint $\varphi_\rho$ is weak (see Figure 8a and Figure 8b). To each finite path $\rho$, we associate a pair $\langle u, \psi \rangle$ consisting of the sum $u$ of the weights in $\rho$, and the constraint $\psi = \varphi_\rho$.

Given two pairs $\langle u, \psi \rangle$, $\langle u', \psi' \rangle$ (associated with paths $\rho$ and $\rho'$ respectively), we write $\langle u, \psi \rangle \succeq \langle u', \psi' \rangle$ if $u \geq u'$ and $\psi'$ implies $\psi$, and we say that $\rho$ is *preferred* to $\rho'$ (this is a partial order). Given a set $S$ of such

**Algorithm 1** $\mathsf{BestPaths}(t_0, v_0, u_0, \psi_0)$

**Input** : $t_0 \in \mathbb{N}$ is an initial time point, $v_0$ is an initial vertex, $u_0$ is the initial sum of weights, and $\psi_0$ is the initial constraint on the slope parameter $M$.

**Output**: The table of $\succeq$-maximal values of paths from $v_0$ with initial values $t_0, u_0, \psi_0$.

**begin**
    /* initialization */
    $D[t_0, v_0] \leftarrow \{\langle u_0, \psi_0 \rangle\}$
    **for** $v \in V \setminus \{v_0\}$ **do**
        $D[t_0, v] \leftarrow \varnothing$
    /* iterations */
    **for** $i = 1, \dots, |V|$ **do**
        **for** $v \in V$ **do**
            $D[t_0 + i, v] \leftarrow \varnothing$
            **for** $v_1 \in V$ *and* $\langle u_1, \psi_1 \rangle \in D[t_0 + i - 1, v_1]$ **do**
                **if** $(v_1, v) \in E$ **then**
                    $u \leftarrow u_1 + w(v_1, v)$
                    $t \leftarrow t_0 + i - 1$
                    $\psi \leftarrow \psi_1 \wedge (r \geq M \cdot (t - T))$
                    $D[t_0 + i, v] \leftarrow D[t_0 + i, v] \cup \{\langle u, \psi \rangle\}$
            $D[t_0 + i, v] \leftarrow \lceil D[t_0 + i, v] \rceil$
    **return** $D$
**end**

pairs, denote by $\lceil S \rceil = \{z_1 \in S \mid \forall z_2 \in S : z_2 \succeq z_1 \rightarrow z_1 \succeq z_2\}$ the set of $\succeq$-maximal elements of $S$. Note that the elements of $\lceil S \rceil$ are pairwise $\succeq$-incomparable.

Intuitively, if $\rho$ and $\rho'$ end in the same vertex, and $\rho$ is *preferred* to $\rho'$, then it is easier to extend $\rho$ than $\rho'$ to obtain an (infinite) path with expected value at least 0. Formally, for all infinite paths $\pi$ with $\mathsf{start}(\pi) = \mathsf{end}(\rho) = \mathsf{end}(\rho')$ we have $\mathit{val}(\rho \cdot \pi, T) \geq \mathit{val}(\rho' \cdot \pi, T)$. We use this result in the following form.

**Lemma 10.** *Let $\rho_1$, $\rho_A$ be two paths of the same length with the same end state, i.e.,* $\mathsf{end}(\rho_1) = \mathsf{end}(\rho_A)$. *If $\rho_1$ is preferred to $\rho_A$, then for all paths $\rho_C$ with* $\mathsf{start}(\rho_C) = \mathsf{end}(\rho_A)$, *the path $\rho_1 \cdot \rho_C$ is preferred to the path $\rho_A \cdot \rho_C$.*

*Proof sketch.* Let $\rho_{1C} = \rho_1 \cdot \rho_C$ an $\rho_{AC} = \rho_A \cdot \rho_C$. Denote by $u_1$, $u_A$, $u_{1C}$, and $u_{AC}$ the sum of the weights of the paths $\rho_1$, $\rho_A$, $\rho_1 \cdot \rho_C$, and $\rho_A \cdot \rho_C$ respectively.

Since $u_1 \geq u_A$ and $\varphi_{\rho_A} \rightarrow \varphi_{\rho_1}$, it is easy to see that $u_{1C} \geq u_{AC}$, and that for every length $|\rho_1| \leq k \leq |\rho_1| + |\rho_C|$, the sum of the weights of the prefix of length $k$ of $\rho_1 \cdot \rho_C$ at least as large as the sum of the weights of the prefix of length $k$ of $\rho_A \cdot \rho_C$. It follows that $\varphi_{\rho_{AC}} \rightarrow \varphi_{\rho_{1C}}$ as well, hence $\rho_1 \cdot \rho_C$ is preferred to $\rho_A \cdot \rho_C$. ☐

Our algorithm uses the procedure $\mathsf{BestPaths}(t_0, v_0, u_0, \psi_0)$ (shown as Algorithm 1) that computes the $\succeq$-maximal pairs $\langle u, \psi \rangle$ corresponding to the paths $\rho_1$ of length $1, 2, \dots, |V|$ that start at time $t_0$ in vertex $v_0$ (see Figure 9), and that prolong a path $\rho_\sharp$ with sum of weight $u_0$ and constraint $\psi_0$ on $M$ (where $u$ is the sum of weights along $\rho_\sharp \cdot \rho_1$, and $\psi \equiv \varphi_{\rho_\sharp \cdot \rho_1}$). We give a precise statement of this result in Lemma 11.

**Lemma 11** (Correctness of $\mathsf{BestPaths}$)**.** *Let $\rho_\sharp$ be a finite path of length $t_0$, that ends in state* $\mathsf{end}(\rho_\sharp) = v_0$ *with sum of weight $u_0$ and associated constraint $\psi_0$ on $M$. Let* $D = \mathsf{BestPaths}(t_0, v_0, u_0, \psi_0)$. *Then,*

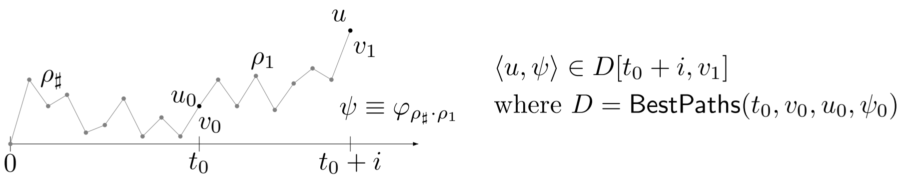



Figure 9: The result of the computation of $\mathsf{BestPaths}(t_0, v_0, u_0, \psi_0)$.

- *for all* $0 \leq i \leq |V|$*, for all* $v_1 \in V$*, for all pairs* $\langle u, \psi \rangle \in D[t_0 + i, v_1]$*, there exists a path* $\rho_1$ *of length* $i$ *with* $\mathsf{start}(\rho_1) = v_0$ *and* $\mathsf{end}(\rho_1) = v_1$*, such that*
  - $u$ *is the sum of weights of the path* $\rho_\sharp \cdot \rho_1$*, and*
  - $\psi \equiv \varphi_{\rho_\sharp \cdot \rho_1}$ *is the constraint on* $M$ *associated with the path* $\rho_\sharp \cdot \rho_1$*;*
- *for all paths* $\rho_1$ *of length* $i \leq |V|$ *such that* $\mathsf{start}(\rho_1) = v_0$ *and* $\mathsf{end}(\rho_1) = v_1$*, there exists a pair* $\langle u', \psi' \rangle \in D[t_0 + i, v_1]$ *such that* $\langle u', \psi' \rangle \succeq \langle u, \psi \rangle$ *where*
  - $u$ *is the sum of weights of the path* $\rho_\sharp \cdot \rho_1$*, and*
  - $\psi \equiv \varphi_{\rho_\sharp \cdot \rho_1}$ *is the constraint on* $M$ *associated with the path* $\rho_\sharp \cdot \rho_1$*.*

*Proof.* For the first item, the proof is by induction on $i$. The case $i = 0$ holds since $D[t_0, v_1]$ is nonempty only for $v_1 = v_0$ (lines 1-3 of Algorithm 1), and we can take for $\rho_1$ the empty path since then $D[t_0, v_0] = \{\langle u_0, \psi_0 \rangle\}$ contains the pair associated with $\rho_\sharp = \rho_\sharp \cdot \rho_1$.

For the inductive case, consider length $i \geq 1$ and assume that the result holds for length $i - 1$. Then for all pairs $\langle u_1, \psi_1 \rangle \in D[t_0 + i - 1, v_1]$ where $v_1 \in V$ (see also line 7 of Algorithm 1), there exists a path $\rho_1$ of length $i - 1$ such that $\langle u_1, \psi_1 \rangle$ is the pair associated with $\rho_\sharp \cdot \rho_1$. It is easy to see that the pair $\langle u, \psi \rangle$ added to $D[t_0 + i, v]$ at line 12 of Algorithm 1 is associated with the path $\rho_\sharp \cdot \rho_1 \cdot (v_1, v)$ where $u = u_1 + w(v_1, v)$ and $\psi \equiv \psi_1 \wedge (r \geq M \cdot (t - T))$ with $t = t_0 + i - 1 = |\rho_\sharp \cdot \rho_1 \cdot (v_1, v)| - 1$. Since the assignment at line 13 of Algorithm 1 can only remove pairs from $D[t_0 + i, v]$, the result follows.

For the second item, the result follows from similar arguments as above, a proof by induction on $i$ using Lemma 10, and the fact that the algorithm explores all successors $v$ of each vertex $v_1$ that ends a path associated with a pair $\langle u_1, \psi_1 \rangle \in D[t_0 + i - 1, v_1]$. □

As we know that simple lassos are sufficient for optimal value (Lemma 8), our algorithmic solution is to explore finite paths from the initial vertex, until a loop is formed. Thus it is sufficient to explore paths of length at most $|V|$. However, given a simple lasso $\rho_A \cdot \rho_C^\omega$, it is not sufficient that the finite path $\rho_A \cdot \rho_C$ lies above a line $M \cdot (t - T)$ (where $M$ satisfies the constraint $\psi_{AC}$ associated with $\rho_A \cdot \rho_C$) to ensure that the value of the lasso $\rho_A \cdot \rho_C^\omega$ is at least 0. The reason is that by repeating the cycle $\rho_C$ several times, the path may eventually cross the line $M \cdot (t - T)$. We show (in Lemma 12) that this cannot happen if the average weight $M_C$ of the cycle is greater than the slope of the line (i.e., $M_C \geq M$).

**Lemma 12.** *Given a lasso* $\rho_A \cdot \rho_C^\omega$*, let* $\psi_{AC}$ *be the symbolic constraint on* $M$ *associated with the finite path* $\rho_A \cdot \rho_C$*, and let* $M_C$ *be the average weight of the cycle* $\rho_C$*. The lasso* $\rho_A \cdot \rho_C^\omega$ *has value at least* 0 *if and only if the formula* $\psi_{AC} \wedge (M_C \geq M)$ *is satisfiable.*

*Proof.* First, if the lasso $\rho_A \cdot \rho_C^\omega$ has value at least 0, then by Lemma 9, there exists a slope $M \in \mathbb{R}$ such that $u_t \geq M \cdot (t - T)$ for all $t \geq 0$ (where $u_t$ is the sum of weights at time $t$ in $\rho_A \cdot \rho_C^\omega$). For such value of $M$, the formula $\psi_{AC}$ holds (by definition), and it is easy to see that $M_C \geq M$ (otherwise, there would exist $t \geq 0$ such that $u_t < M \cdot (t - T)$). Therefore $\psi_{AC} \wedge (M_C \geq M)$ is satisfiable.

Second, if the formula $\psi_{AC} \wedge (M_C \geq M)$ is satisfiable, then let $M$ be a satisfying value, and by Lemma 9 and a similar argument as above, the lasso $\rho_A \cdot \rho_C^\omega$ has value at least 0. □

**Algorithm 2** $\mathsf{ExistsPositivePath}(v_0)$

**Input** : $v_0$ is an initial vertex.
**Output**: $\mathsf{true}$ iff there exists a path from $v_0$ with expected utility at least 0.
**begin**

1 $\quad A \leftarrow \mathsf{BestPaths}(0, v_0, 0, \mathsf{true})$
2 $\quad$ **for** $i = 0, \ldots, |V|$ **do**
3 $\quad\quad$ **for** $\hat{v} \in V$ *and* $\langle u_1, \psi_1 \rangle \in A[i, \hat{v}]$ **do**
4 $\quad\quad\quad C \leftarrow \mathsf{BestPaths}(i, \hat{v}, u_1, \psi_1)$
5 $\quad\quad\quad$ **for** $j = 1, \ldots, |V| - i$ **do**
6 $\quad\quad\quad\quad$ **for** $\langle u_2, \psi_2 \rangle \in C[i + j, \hat{v}]$ **do**
7 $\quad\quad\quad\quad\quad$ **if** $\psi_2 \wedge \frac{u_2 - u_1}{j} \geq M$ *is satisfiable* **then return** $\mathsf{true}$
8 $\quad$ **return** $\mathsf{false}$

**end**

The algorithm $\mathsf{ExistsPositivePath}(v_0)$ explores the paths from $v_0$, and keeps the $\succeq$-preferred paths, that is those with the largest total weight and weakest constraint on $M$. There may be several $\succeq$-incomparable paths of a given length $i$ that reach a given vertex $\hat{v}$, therefore we need to compute a *set* $A[i, \hat{v}]$ of $\succeq$-incomparable pairs (line 1 of Algorithm 2).

Given a pair $\langle u_1, \psi_1 \rangle \in A[i, \hat{v}]$, the algorithm $\mathsf{ExistsPositivePath}$ further explores (for-loop at line 3 of Algorithm 2) the paths from $\hat{v}$, until a cycle $\rho_C$ of length $j$ is formed around $\hat{v}$, with average weight $M_C = \frac{u_2 - u_1}{j}$ and associated pair $\langle u_2, \psi_2 \rangle \in C[i + j, \hat{v}]$ (line 7 of Algorithm 2) such that $\psi_2 \wedge (M_C \geq M)$ is satisfiable. We claim that there exists such a cycle if and only if there exists a lasso with value at least 0. The claim is established in the following lemma.

**Lemma 13** (Correctness of $\mathsf{ExistsPositivePath}$)**.** *There exists an infinite path from $v_0$ with value at least 0 if and only if $\mathsf{ExistsPositivePath}(v_0)$ returns $\mathsf{true}$.*

*Proof.* **(First part)**

For the first direction of the proof, if there exists an infinite path with value at least 0, then by Lemma 8 there exists a lasso $\rho = \rho_A \cdot \rho_C^\omega$ with value at least 0.

Consider the call $A \leftarrow \mathsf{BestPaths}(t_0, v_0, u_0, \psi_0)$ in $\mathsf{ExistsPositivePath}$ (line 1 of Algorithm 2) where $t_0 = u_0 = 0$ and $\psi_0 \equiv \mathsf{true}$. Let $\hat{v} = \mathsf{end}(\rho_A)$ and let $i$ be the length of $\rho_A$ (note that $i < |V|$ because $\rho_A$ is acyclic). By the correctness result of $\mathsf{BestPaths}$ (Lemma 11 (item 2), where $\rho_\sharp$ is the empty path), there is a pair $\langle u_1, \psi_1 \rangle \in A[i, \hat{v}]$ such that $\langle u_1, \psi_1 \rangle \succeq \langle u_A, \psi_A \rangle$ where $\langle u_A, \psi_A \rangle$ is the pair associated with $\rho_A$, thus $u_1 \geq u_A$ and $\psi_A \to \psi_1$ holds. Then by Lemma 11 (item 1), there is a path $\rho_1$ of length $i$ from $v_0$ to $\hat{v}$, and $u_1$ is the sum of weights of $\rho_1$, and $\psi_1 \equiv \varphi_{\rho_1}$ is the constraint on $M$ associated with $\rho_1$ (i.e., $\rho_1$ is preferred to $\rho_A$).

Now consider the call $C \leftarrow \mathsf{BestPaths}(i, \hat{v}, u_1, \psi_1)$ in $\mathsf{ExistsPositivePath}$ (line 4 of Algorithm 2). Let $\rho_\sharp = \rho_1$ in Lemma 11 and note that the assumptions of that lemma are satisfied, namely $\langle u_1, \psi_1 \rangle$ is the pair associated with $\rho_1$, and $\hat{v} = \mathsf{end}(\rho_1)$.

Since $\rho_A \cdot \rho_C^\omega$ is a lasso, we have $\mathsf{start}(\rho_C) = \mathsf{end}(\rho_C) = \mathsf{end}(\rho_A) = \hat{v}$ and let $j$ be the length of $\rho_C$ (note that $i + j \leq |V|$). By Lemma 11 (item 2), there is a pair $\langle u_2, \psi_2 \rangle \in C[i + j, \hat{v}]$ such that $\langle u_2, \psi_2 \rangle \succeq \langle u_{1C}, \psi_{1C} \rangle$ where $\langle u_{1C}, \psi_{1C} \rangle$ is the pair associated with $\rho_1 \cdot \rho_C$, thus $u_2 \geq u_{1C}$ and $\psi_{1C} \to \psi_2$ holds, and by Lemma 11 (item 1), there is a path $\rho_2$ of length $j$ such that $\mathsf{start}(\rho_2) = \mathsf{end}(\rho_2) = \hat{v}$ and $u_2$ is the sum of weights of $\rho_1 \cdot \rho_2$, and $\psi_2 \equiv \varphi_{\rho_1 \cdot \rho_2}$ is the constraint on $M$ associated with $\rho_1 \cdot \rho_2$.

Now we show that $\psi_2 \wedge \frac{u_2-u_1}{j} \geq M$ is satisfiable, and thus $\mathsf{ExistsPositivePath}(v_0)$ returns $\mathsf{true}$ (Line 7 of Algorithm 2). First, by Lemma 12 the formula $\psi_{AC} \wedge (M_C \geq M)$ is satisfiable, and by Lemma 10 we have $\psi_{AC} \rightarrow \psi_{1C}$. We showed above that $\psi_{1C} \rightarrow \psi_2$, thus $\psi_2 \wedge (M_C \geq M)$ is satisfiable. Now, since the length of the cycle $\rho_C$ (and of $\rho_2$) is $j - i$ (i.e., the length of $\rho_A \cdot \rho_C$ minus the length of $\rho_A$), we have $M_C = \frac{S_C}{j}$. Moreover we showed above that $u_2 \geq u_{1C} = u_1 + S_C$, thus $M_C = \frac{S_C}{j} \leq \frac{u_2-u_1}{j}$, and since $\psi_2 \wedge (M_C \geq M)$ is satisfiable it follows that $\psi_2 \wedge \frac{u_2-u_1}{j} \geq M$ is satisfiable as well.

**(Second part)**

For the second direction of the proof, if $\mathsf{ExistsPositivePath}(v_0)$ returns $\mathsf{true}$, then there exists $i, j, \hat{v}, \langle u_1, \psi_1 \rangle, \langle u_2, \psi_2 \rangle$ (corresponding to the for-loops in lines 2, 3, 5, 6 of Algorithm 2) such that:

- $0 \leq i \leq |V|$ and $1 \leq j \leq |V| - i$,
- $\hat{v} \in V$,
- $\langle u_1, \psi_1 \rangle \in A[i, \hat{v}]$ and $\langle u_2, \psi_2 \rangle \in C[i + j, \hat{v}]$ where $A = \mathsf{BestPaths}(0, v_0, 0, \mathsf{true})$, and $C = \mathsf{BestPaths}(i, \hat{v}, u_1, \psi_1)$,
- $\psi_2 \wedge \frac{u_2-u_1}{j} \geq M$ is satisfiable.

Therefore, by Lemma 11 (item 1), there exist paths $\rho_A$ and $\rho_C$ such that:

- $\rho_A$ is a path of length $i$ from $v_0$ to $\hat{v}$, such that $u_1$ is the sum of weights of the path $\rho_A$, and $\psi_1 \equiv \varphi_{\rho_A}$;
- $\rho_C$ is a path of length $j$ with $\mathsf{start}(\rho_C) = \mathsf{end}(\rho_C) = \hat{v}$ (thus $\rho_C$ is a cycle), such that $u_2$ is the sum of weights of the path $\rho_A \cdot \rho_C$, and $\psi_2 \equiv \varphi_{\rho_A \cdot \rho_C}$ is the constraint on $M$ associated with the path $\rho_A \cdot \rho_C$.

Therefore, $u_2 - u_1$ is the sum of the weights along $\rho_C$, and thus $M_C = \frac{u_2-u_1}{j}$. Since the formula $\psi_2 \wedge \frac{u_2-u_1}{j} \geq M$ is satisfiable, it follows that $\varphi_{\rho_A \cdot \rho_C} \wedge (M_C \geq M)$ is satisfiable, and by Lemma 12, the lasso $\rho_A \cdot \rho_C^\omega$ has value at least 0. $\square$

*Optimal value.* We can compute the optimal value using the procedure $\mathsf{ExistsPositivePath}$ as follows. From Lemma 4, the optimal value is either of the form $\frac{u_{t_1} \cdot (t_2 - T) + u_{t_2} \cdot (T - t_1)}{t_2 - t_1}$, or of the form $u_{t_1} + (T - t_1) \cdot \nu$ where the following bounds hold ($\nu = \inf_{t_2 \geq T} \frac{u_{t_2} - u_{t_1}}{t_2 - t_1}$):

- $0 \leq t_1 \leq t_2 \leq |V|$
- $0 \leq t_2 - t_1 \leq |V|$
- $0 \leq T - t_1 \leq |V|$
- $0 \leq t_2 - T \leq |V|$
- $-W \cdot |V| \leq u_{t_1}, u_{t_2} \leq W \cdot |V|$
- $\nu$ is a rational number $\frac{p}{q}$ where $-W \cdot |V| \leq p \leq W \cdot |V|$ and $1 \leq q \leq |V|$

Therefore, in both cases we get the following result.

**Lemma 14.** *The optimal value belongs to the set*

$$\mathsf{ValueSpace} = \left\{ \frac{p}{q} \mid -2W \cdot |V|^2 \leq p \leq 2W \cdot |V|^2 \text{ and } 1 \leq q \leq |V| \right\}.$$

Given a value $\frac{p}{q}$, we can decide if there exists a path with expected value at least $\frac{p}{q}$ by subtracting $\frac{p}{q \cdot T}$ from all the weights the graphs, and asking if there exists a path with expected value at least 0 in the modified graph. Indeed, if we define $w'(e) = w(e) + \eta$ for all edges $e \in E$, then for all paths $\rho$, if $u$ is the sequence of utilities along $\rho$ according to $w$, and $u'$ is the sequence of utilities along $\rho$ according to $w'$, then

$$\sum_i p_i \cdot u'_i = \sum_i p_i \cdot (u_i + \eta \cdot i) = \eta \cdot \sum_i p_i \cdot i + \sum_i p_i \cdot u_i = T \cdot \eta + \sum_i p_i \cdot u_i,$$

thus the value of the path is shifted by $T \cdot \eta$. Then it follows from Lemma 14 that the optimal value can be computed by a binary search using $O(|\mathsf{ValueSpace}|) = O(\log(W \cdot |V|))$ calls to ExistsPositivePath.

*Optimal path.* An optimal path can be constructed by a slight modification of the algorithm. In BestPaths, we can maintain a path associated to each pair in $D$ as follows: the empty path is associated to the pair $\langle u_0, \psi_0 \rangle$ added at line 1 of Algorithm 1, and given the path $\rho_1$ associated with the pair $\langle u_1, \psi_1 \rangle$ (line 7 of Algorithm 1), we associate the path $\rho_1 \cdot (v_1, v)$ with the pair $\langle u, \psi \rangle$ added to $D$ at line 12 of Algorithm 1. It is easy to see that for every pair $\langle u, \psi \rangle$ in $D$, the associated path can be used as the path $\rho_1$ in Lemma 11 (item 1). Therefore, when ExistsPositivePath($v_0$) returns true (line 7 of Algorithm 2), we can output the path $\rho_1 \cdot \rho_2^\omega$ where $\rho_i$ is the path associated with the pair $\langle u_i, \psi_i \rangle$ ($i = 1, 2$).

### 4.2.2 Complexity analysis

We present the running-time analysis of ExistsPositivePath (Algorithm 2). The key challenge is to bound the number of $\succeq$-incomparable pairs. The number of such pairs corresponds to the number of simple paths in a graph, and hence can be exponential in general. Our main argument is to establish a polynomial bound on the number of $\succeq$-incomparable pairs.

To analyze the complexity of the algorithm, we need to bound the size of the array $D$ computed by BestPaths (Algorithm 1). We show that there cannot be too many different pairs in a given entry $D[t_0+i, v_1]$. By Lemma 11, to each pair $\langle u, \psi \rangle \in D[t_0 + i, v_1]$ we can associate a path $\rho$ of length $i$ with $\mathsf{start}(\rho) = v_0$ and $\mathsf{end}(\rho) = v_1$, such that (our analysis holds for all paths $\rho_\sharp$ in Lemma 11, and as $\rho_\sharp$ plays no role in the argument, we proceed with empty $\rho_\sharp$ for simplicity of the exposition[4]):

- $u$ is the sum of weights of the path $\rho$, and
- $\psi \equiv \varphi_\rho$ is the constraint on $M$ associated with the path $\rho$.

It is important to note that the constraint $\psi$ is determined by (at most) two points $t_L, t_R$ in $\rho$ (see also Figure 8a and Figure 8b), one before $T$ and one after $T$, namely

$$\psi \equiv \big(u_{t_L} \geq M \cdot (t_L - T)\big) \wedge \big(u_{t_R} \geq M \cdot (t_R - T)\big)$$

where $t_L = \mathrm{argmax}_{0 \leq i \leq T}(\frac{u_i}{i-T})$ and $t_R = \mathrm{argmin}_{T \leq i \leq |\rho|}(\frac{u_i}{i-T})$.

Note that the first constraint in the above expression is a lower bound on $M$ since $t_L \leq T$, and the second constraint (which may not exist, if $|\rho| < T$) is an upper bound on $M$. For simplicity of exposition, we assume that $|\rho| \geq T$. The case $|\rho| < T$ is handled analogously ($t_R$ is undefined in that case).

Define the *down-point* of $\rho = e_0 e_1 \dots e_{|\rho|-1}$ as $\mathsf{downpoint}(\rho) = \langle t_L, v_L, t_R, v_R \rangle$ where $t_L$ and $t_R$ are defined above, and $v_L = \mathsf{end}(e_0 e_1 \dots e_{t_L})$, and $v_R = \mathsf{end}(e_0 e_1 \dots e_{t_R})$ (for $|\rho| < T$, the down-point of $\rho$ is $\mathsf{downpoint}(\rho) = \langle t_L, v_L \rangle$).

Decompose $\rho$ into $\rho_L = e_0 e_1 \dots e_{t_L}$, $\rho_M = e_{t_L+1} e_{t_L+2} \dots e_{t_R}$, and $\rho_R = e_{t_R+1} e_{t_R+2} \dots e_{|\rho|-1}$. We claim that the paths corresponding to two different pairs in $D[t_0+i, v_1]$ have different down-points, which will give us a polynomial bound on the size of $D[t_0+i, v_1]$. Intuitively, and towards contradiction, if two down-points are the same in two different paths, then we can select the best pieces among $(\rho_L, \rho_M, \rho_R)$ from the two paths and construct a path that is preferred, and thus whose pair is in $D[t_0+i, v_1]$ and subsumes some pair in $D[t_0+i, v_1]$, which is a contradiction since the elements of $D[t_0+i, v_1]$ are $\succeq$-maximal.

[4]The proof can be carried out analogously by considering $\rho_\sharp \cdot \rho$ instead of $\rho$ with heavier notation.

**Lemma 15.** *Let $D = \mathsf{BestPaths}(t_0, v_0, u_0, \psi_0)$ and $1 \le i \le |V|$. For all pairs $\langle u, \psi\rangle, \langle u', \psi'\rangle \in D[t_0 + i, v_1]$, let $\rho, \rho'$ be their respective associated path; if $\langle u, \psi\rangle \neq \langle u', \psi'\rangle$, then the down-points of $\rho$ and $\rho'$ are different ($\mathsf{downpoint}(\rho) \neq \mathsf{downpoint}(\rho')$).*

*Proof.* We prove the contrapositive, for $|\rho| \ge T$ (the case $|\rho| < T$ is simpler, and proved analogously). Assume that $\langle t_L, v_L, t_R, v_R\rangle = \langle t'_L, v'_L, t'_R, v'_R\rangle$ (the down-points are equal), and we show that then $\langle u, \psi\rangle = \langle u', \psi'\rangle$.

First, since $t_L = t'_L$ and $v_L = v'_L$, we claim that the sum of weights at time $t_L$ is the same in $\rho$ and in $\rho'$, that is $u_{t_L} = u'_{t_L}$, and therefore, $\varphi_{\rho_L} \equiv \varphi_{\rho'_L}$ (remember that the constraint $\psi$ associated with $\rho$ and $\rho'$ is determined by $t_L = t'_L$). The proof of this claim is by contradiction. Assume that $u_{t_L} > u'_{t_L}$ (the argument for the case $u_{t_L} < u'_{t_L}$ is analogous). Consider the path $\overline{\rho} = \rho_L \cdot \rho'_M \cdot \rho'_R$, and note that $\overline{\rho}$ is indeed a path[5], as $\mathsf{end}(\rho_L) = v_L = v'_L = \mathsf{start}(\rho'_M)$. Comparing $\overline{\rho}$ and $\rho'$, since $u_{t_L} > u'_{t_L}$ it is easy to see that $\bar{u} > u'$ where $\bar{u}$ is the sum of weights of $\overline{\rho}$, and by the same argument we have $\psi' \to \psi_{\overline{\rho}}$. It follows that $\overline{\rho}$ is preferred to $\rho'$, and by Lemma 11 the set $D[t_0 + i, v_1]$ contains a pair $\langle u^*, \psi^*\rangle \succeq \langle \bar{u}, \varphi_{\overline{\rho}}\rangle \succeq \langle u', \psi'\rangle$. Since $D[t_0 + i, v_1]$ is a set of $\succeq$-maximal elements (line 13 of Algorithm 1), it follows that $\langle u', \psi'\rangle \notin D[t_0 + i, v_1]$, in contradiction with the assumption of the lemma.

Second, by an analogous argument, since $t_R = t'_R$ and $v_R = v'_R$, the sum of weights at time $t_R$ is the same in $\rho$ and in $\rho'$, that is $u_{t_R} = u'_{t_R}$, and therefore, $\varphi_{\rho_R} \equiv \varphi_{\rho'_R}$. Finally $u = u'$ and $\psi \equiv \psi'$, which concludes the proof. ◻

It follows from Lemma 15 that the size of all sets $D[t_0 + i, v_1]$ for $1 \le i \le |V|$ and $v_1 \in V$ is at most $|V|^4$, the maximum number of different down-points.

We now show that the worst-case complexity of BestPaths and ExistsPositivePath is polynomial, and thus the optimal expected value problem is solvable in polynomial time.

The worst-case complexity of BestPaths is $O(|V|^{10})$, as there are two nested for-loops over $V$ (line 4 and line 5 in Algorithm 1), in which the dominating operation is the computation of the $\succeq$-maximal elements of $D[t_0 + i, v]$ (line 13), which is quadratic in the size of $D[t_0 + i, v]$, thus in $O(|V|^8)$.

The worst-case complexity of ExistsPositivePath is $O(|V| \cdot |V| \cdot |V|^4 \cdot |V|^{10}) = O(|V|^{16})$, as a product of the size of the three outermost for-loops, and the dominating call to BestPaths (line 4) in $O(|V|^{10})$. Therefore we obtain Theorem 4.

## 5 Conclusion

In this work we consider the expected finite-horizon problem. Our most interesting results are for the case of adversarial distribution of stopping times, for which we establish stationary plans are sufficient, and present polynomial-time algorithms. In terms of algorithmic complexity, our main goal was to establish polynomial-time algorithms, and we expect that better algorithms and refined complexity analysis can be obtained.

## References


[1] A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, and Y. Zhu. Bounded model checking. *Advances in Computers*, 58:117–148, 2003.

[2] B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

[3] E. A. Emerson, A. K. Mok, A. P. Sistla, and J. Srinivasan. Quantitative temporal reasoning. *Real-Time Systems*, 4(4):331–352, 1992.

[4] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.


[5] Note that if $\rho$ and $\rho'$ have a common prefix (such as $\rho_\sharp$), then $\overline{\rho}$ also has the same prefix.

[5] S. Fortune, J. E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.

[6] E. Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications (Texts in Theoretical Computer Science. An EATCS Series).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[7] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[8] H. Howard. *Dynamic Programming and Markov Processes.* MIT Press, 1960.

[9] S. M. LaValle. *Planning algorithms.* Cambridge University Press, 2006.

[10] M. A. Nowak. *Evolutionary dynamics.* Harvard University Press, 2006.

[11] M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

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

[13] S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach (3rd ed.).* Pearson Education, 2010.