Discounting versus Averaging in Dynamic Programming

Ehud Lehrer
Northwestern University, Evanston, Illinois 60201

AND

Dov Monderer
Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Technion City, Haifa 32000, Israel

Received July 18, 1989

We explore relationships between the long-run average values of a dynamic programming problem and the limit of the discounted values when the discount factor goes to one. Journal of Economic Literature Classification Numbers: C72, C73. © 1994 Academic Press, Inc.

1. INTRODUCTION

Let $S$ be a state space. For each $s \in S$ let $0 \neq \Gamma(s) \subseteq S$, and let $f$ be a bounded function defined on $S$. Consider the dynamic programming problem where on day $t$ the decision maker has to choose a new state $s_{t+1} \in \Gamma(s_t)$, and receives a payoff $f(s_t)$. Let $V_\lambda(s)$ be the value of the discounted problem

$$V_\lambda(s) = \sup_{(s_t)_{t=0}^\infty} (1 - \lambda) \sum_{t=0}^\infty \lambda^t f(s_t),$$

where the supremum ranges over all plays $(s_t)_{t=0}^\infty$ (i.e., $s_{t+1} \in \Gamma(s_t)$) with $s_0 = s$.

Assume $V_\lambda(s)$ converges for all $s \in S$ when $\lambda \to 1$, say to $V(s)$. $V(s)$...
may then be viewed as a long-run value of the problem. However, there are various other values that may be considered as well. Consider the following two values: The lower long-run average value

$$\underline{V}(s) = \sup_{(s_i)_{i=0}^\infty} \lim_{T \to \infty} \frac{1}{T+1} \sum_{i=0}^T f(s_i)$$

and the upper long-run average value

$$\bar{V}(s) = \sup_{(s_i)_{i=0}^\infty} \lim_{T \to \infty} \frac{1}{T+1} \sum_{i=0}^T f(s_i).$$

This paper is devoted to exploring the various relationships between the three long-run values mentioned above. It can be easily verified that for all states $s$ $V(s) \geq \underline{V}(s)$ and $\bar{V}(s) \geq V(s)$. Mertens and Neyman (1981) proved (in the context of stochastic zero-sum games) that $V = \underline{V}$ whenever the convergence of $V_{\lambda}$ to $V$ holds in a very strong sense. Mertens (1987) posed the following problem (it was also suggested by him at the Open Problems session of the International Conference of Game Theory in June 1987):

Suppose $V_{\lambda}(s)$ when $\lambda \to 1$ uniformly in $s \in S$. Does it imply that $V = \underline{V}$?

In Section 3 we construct a counterexample to Mertens’ conjecture. It turns out, however, that the uniform convergence condition does have a positive (surprising) implication. In Section 2 we show that it implies the equality $V = \bar{V}$.

Another long-run value has been extensively studied in the literature of dynamic programming:

For $T \geq 1$ and $s \in S$ set

$$u_T(s) = \sup_{(s_i)_{i=0}^T} \frac{1}{T+1} \sum_{i=0}^T f(s_i),$$

where the supremum ranges over all $T$-length plays $(s_i)_{i=0}^T$ with $s_0 = s$. Suppose $\lim_{T \to \infty} V_T(s)$ exists for all $s \in S$ and denote it by $W(s)$. Then $W$ may also be considered as a long-run value. Lehrer and Sorin (1992) proved that if either one of the limits $\lim_{\lambda \to 1} V_{\lambda}(s)$, or $\lim_{T \to \infty} V_T(s)$ exists uniformly in $s \in S$, then the other limit also exists uniformly, and the limit functions coincide. We briefly discuss implications of Lehrer and Sorin’s result in Section 4.

For a general discussion about dynamic programming the reader is
and Bertsekas (1976).

2. The Upper Limit Value

In this section we prove that if \( V_\lambda \to V \) uniformly then \( \bar{V} = V \). In
Theorem A we prove that \( \bar{V} \geq V \), and in Theorem B we prove that \( \bar{V} \geq V \).
We now give an outline of the proof of Theorem A.

Let \( s_0 \in S \) and let \( \varepsilon > 0 \). Our task is to find a sequence \( s_0 < s_1 < s_2 < \ldots \) (where \( s_{i+1} > s_{i+2} \) means \( s_{i+1} \in \Gamma(s_i) \)) such that the upper Cesaro limit of the payoffs' sequence \( (f(s_i))_{n=0}^{\infty} \) is at least \( V(s_0) - \varepsilon \). We define sequences \( \lambda_n \to 1 \) and \( \delta_n \to 0 \) with properties that are described later and proceed as follows:

\( \lambda_1 \) is chosen sufficiently close to 1 so that \( V_{\lambda_1}(s_0) \geq V(s_0) - \delta_1 \). We then find a sequence \( s_0 = a_0 < a_1 < a_2 < \ldots \) of states which is \( \delta_1 \)-optimal with respect to the discount factor \( \lambda_1 \). That is, the \( \lambda_1 \)-Abel series of \( (f(a_i))_{i=0}^{\infty} \) is at least \( V_{\lambda_1}(s_0) - \delta_1 \). However, for our sequence \( (s_i)_{i=0}^{\infty} \), we take only the first \( t_1 + 1 \) states, where \( t_1 \geq 0 \) is chosen in such a way that both the partial average of the head up to \( t_1 \) and \( V_{\lambda_1}(a_{t_1+1}) \) are at least \( V_{\lambda_1}(s_0) - 2\delta_1 \). Such \( t_1 \) exists because each Abel series can be written as a convex combination of the partial averages (see (2.3)).

\( \lambda_2 \) is chosen so that \( V_{\lambda_2}(a_{t_1+1}) \geq V_{\lambda_1}(a_{t_1+1}) - (\delta_1 + \delta_2) \) (here we are using the uniform convergence of \( V_{\lambda_1} \)). We then find another sequence \( a_{t_1+1} = b_0 < b_1 < b_2 < \ldots \), which is \( \delta_2 \)-optimal with respect to \( \lambda_2 \). We take the first \( t_2 + 1 \) states of this sequence to be the next states in our sequence \( (s_i)_{i=0}^{\infty} \), where \( t_2 \) is chosen so that both the partial average of the head up to \( t_2 \) and \( V_{\lambda_2}(b_{t_2+1}) \) are at least \( V_{\lambda_2}(b_0) - 2\delta_2 \). The next \( t_2 + 1 \) states in \( (s_i)_{i=0}^{\infty} \) come from a sequence corresponding to \( \lambda_2 \), and so on. At the end of the induction process we have an infinite sequence

\[
\begin{align*}
s_0 &= a_0 < a_1 < \cdots < a_{t_1+1} = b_0 < b_1 < \cdots < b_{t_2+1} \\
&= c_0 < c_1 < \cdots < c_{t_3+1} = d_0 < \ldots
\end{align*}
\]

This sequence has the property that the partial averages from the \( (T_1 + 1) \)th state to the \( T_{n+1} \)th state are at least \( V(s_0) - \varepsilon \), where \( T_n = (\sum_{i=1}^{n} t_i) + n - 1 \). Hence, \( \bar{V}(s_0) \geq V(s_0) - \varepsilon \).

**Theorem A.** Let \((S, \Gamma, f)\) be a dynamic programming problem as defined in the introduction. Assume \( V_\lambda(s) \) converges to \( V(s) \) uniformly in \( s \in S \). Then, for every \( s_0 \in S \) and for every \( \varepsilon > 0 \) there exists a path \( s_{t+1} \in \Gamma(s_t) \) such that

\[
\text{...}
\]
\[ \limsup_{T \to \infty} \frac{1}{T + 1} \sum_{t=0}^{T} f(s_t) \geq V(s_0) - \varepsilon. \]

**Proof.** Without loss of generality we may assume that \( 0 \leq f(s) \leq 1 \) for all \( s \in S \). Let \( (\delta_k)_{k=1}^{\infty} \) be a sequence of positive numbers such that \( \sum_{k=1}^{\infty} \delta_k < (\varepsilon/8) \). Since \( V_\lambda \to V \) uniformly, we can find an increasing sequence \( (\lambda_k)_{k=1}^{\infty} \) converging to 1 and satisfying the two properties

\[ 1 - \lambda_k < \delta_k \quad \text{for all } k \geq 1 \quad (2.1) \]

and

\[ ||V_{\lambda_k} - V|| < \delta_k \quad \text{for all } k \geq 1, \quad (2.2) \]

where for \( x \in R^S \) \( ||x|| = \sup_{s \in S} |x(s)| \).

Before proceeding to the construction of the sequence \( s_0 < s_1 < s_2 < \ldots \) (that is, \( s_{t+1} \in \Gamma(s_t) \)) we need the following lemmas.

**Lemma 2.1.** For every sequence \( b = (b_t)_{t=0}^{\infty} \) of real numbers, for every \( 0 \leq \lambda < 1 \), and for every \( T \geq 0 \)

\[ (1 - \lambda) \sum_{t=0}^{T} \lambda^t b_t = (1 - \lambda)^2 \sum_{t=0}^{T-1} \lambda^t (t+1) A_t(b) + (1 - \lambda) \lambda^T (T + 1) A_T(b), \]

where \( \sum_{t=0}^{T-1} = 0 \), and

\[ A_t(b) = \frac{1}{t + 1} \sum_{i=0}^{t} b_i. \]

**Proof of Lemma 2.1.** The proof is based on a simple direct computation, and therefore is omitted. \( \blacksquare \)

Lemma 2.1 implies that for every bounded sequence \( b = (b_t)_{t=0}^{\infty} \)

\[ (1 - \lambda) \sum_{t=0}^{\infty} \lambda^t b_t = (1 - \lambda)^2 \sum_{t=0}^{\infty} \lambda^t (t+1) A_t(b). \quad (2.3) \]

**Lemma 2.2.** Let \( 0 \leq \lambda < 1 \) and let \( b = (b_t)_{t=0}^{\infty} \) be a bounded sequence. Then, there exists \( T \geq 0 \) such that
Discounting Versus Averaging

\[ A_{\gamma}(b) \geq (1 - \lambda) \sum_{i=0}^{\infty} \lambda^i b_i. \]

Proof of Lemma 2.2. To get the result combine \((1 - \lambda)^2 \sum_{i=0}^{\infty} \lambda^i (t + 1) = 1\) with (2.3).

We now construct the path \(s_0 < s_1 < s_2 < \cdots\).

Step 1. Denote \(u_0 = s_0\). Take a path \(u_0 = a_0 < a_1 < a_2 < \cdots\) such that

\[ V_{\lambda_1}(a_0) - \delta_1 \leq (1 - \lambda_1) \sum_{i=0}^{\infty} \lambda_1^i f(a_i). \]

By Lemma 2.2 there exists \(T \geq 0\) with

\[ A_{\gamma}(f(a)) \geq (1 - \lambda_1) \sum_{i=0}^{\infty} \lambda_1^i f(a_i) \geq V_{\lambda_1}(u_0) - \delta_1, \]

where \(f(a) = (f(a_i))_{i=0}^{\infty}\).

Let \(t_1\) be the smallest \(t\) for which \(A_{\gamma}(f(a)) \geq V_{\lambda_1}(u_0) - \delta_1\).

Define \(s_1 = a_1, s_2 = a_2, \ldots, s_{t_1 + 1} = a_{t_1 + 1}\). And for use in the next step define \(u_1 = a_{t_1 + 1}\).

Step 2. Take a path \(u_1 = a_0 < a_1 < \cdots\) with

\[ V_{\lambda_2}(u_1) - \delta_2 \leq (1 - \lambda_2) \sum_{i=0}^{\infty} \lambda_2^i f(a_i). \]

Let \(t_2 \geq 0\) be the smallest integer satisfying

\[ A_{\gamma}(f(a)) \geq V_{\lambda_2}(u_1) - \delta_2. \]

Define, \(s_{t_1 + i} = a_{t_1 + i}\) for \(2 \leq i \leq t_2 + 2\), and \(u_2 = a_{t_2 + 1}\).

Step \(n\). Denote \(q_k = \sum_{i=k}^{\infty} l_i\).\n
\(u_{n-1}\) has already been defined in the \((n-1)\)th step (as \(s_{q_{n-2} + n-1}\)).

Take a sequence \(u_{n-1} = a_0 < a_1 < a_2 < \cdots\) with

\[ V_{\lambda_n}(u_{n-1}) - \delta_n \leq (1 - \lambda_n) \sum_{i=0}^{\infty} \lambda_n^i f(a_i). \tag{2.4} \]

Let \(t_n \geq 0\) be the smallest integer satisfying
Define \( s_{q_n+1,i} = a_{i-n+1} \) for \( n \leq i \leq t_n + n \), and \( u_n = a_{t_n+1} \).

At the end of the induction process we have a path

\[ s_0 > s_1 > s_2 > \ldots, \]

a sequence \((t_n)_{n \geq 1}\) of nonnegative integers, and a sequence \((u_n)_{n \geq 0}\) of states such that \( u_n = s_{q_n+n} \), where \( q_0 = 0 \) and for \( n \geq 1 \) \( q_n = \sum_{i=n}^{t_n} t_i \).

We now need the following two lemmas. The first one asserts that by moving from \( u_{n-1} \) to \( u_n \) and from \( \lambda_n \) to \( \lambda_{n+1} \) we may lose only a small payoff. In the second lemma we show that \( V_{\lambda_n}(u_{n-1}) \) is close to \( V(s_0) \).

**Lemma 2.3.** For every \( n \geq 1 \),

\[ V_{\lambda_n}(u_{n-1}) - V_{\lambda_{n+1}}(u_n) \leq 3 \delta_n + \delta_{n+1}. \]

**Proof of Lemma 2.3.** For \( t \geq 0 \), set \( A_t = A_t(f(a)) = (f(a))_{t=0}^t \). By the construction, and by Lemma 2.1, we have at the \( n \)th step

\[
V_{\lambda_n}(u_{n-1}) - \delta_n \leq (1 - \lambda_n) \sum_{t=0}^{t_n} \lambda_n^t f(a_t) \\
= (1 - \lambda_n) \sum_{t=0}^{t_{n-2}} \lambda_n^t f(a_t) + \lambda_n^{t_n}(1 - \lambda_n) \sum_{t=0}^{t_{n} - t_n} \lambda_n^t f(a_t) \\
= (1 - \lambda_n)^2 \sum_{t=0}^{t_{n-2}} \lambda_n^t (t + 1) A_t + (1 - \lambda_n) \lambda_n^{t_n - t_n} t_n A_{t_n - 1} \\
+ \lambda_n^{t_n}(1 - \lambda_n) \sum_{t=t_n}^{\infty} \lambda_n^t f(a_t).
\]

The right-hand side of (2.6) is a convex combination of all \( A_t, 0 \leq t \leq t_n - 1 \), and \( \sum_{t=t_n}^{\infty} (1 - \lambda_n) \lambda_n^t f(a_t) \). Since \( A_t < V_{\lambda_n}(u_{n-1}) - \delta_n \), for all \( 0 \leq t \leq t_n - 1 \), we have

\[
(1 - \lambda_n) \sum_{t=t_n}^{\infty} \lambda_n^t f(a_t) \geq V_{\lambda_n}(u_{n-1}) - \lambda_n.
\]

Hence,
\[ V_{\lambda}(u_n) = V_{\lambda}(a_{n+1}) \geq (1 - \lambda_n) \sum_{i=t_n}^{n} \lambda_n^{i-(u_{i+1})} f(a_i) \]
\[ \geq (1 - \lambda_n) \sum_{i=t_n}^{n} \lambda_n^{i-1} f(a_i) - (1 - \lambda_n) f(a_n) \]
\[ \geq V_{\lambda}(u_{n-1}) - 2\delta_n. \]

Therefore,

\[ V_{\lambda}(u_{n-1}) - V_{\lambda_{n-1}}(u_n) \leq V_{\lambda}(u_{n-1}) - V_{\lambda}(u_n) \]
\[ + V_{\lambda}(u_n) - V(u_n) + V(u_n) - V_{\lambda_{n-1}}(u_n) \]
\[ \leq 3\lambda_n + \delta_{n-1}. \]

This completes the proof of Lemma 2.3. ■

**Lemma 2.4.** For all \( n \geq 1 \)

\[ V(s_0) - V_{\lambda}(u_{n-1}) \leq 4 \sum_{k=n}^{\infty} \delta_k. \]

**Proof of Lemma 2.4.**

\[ V(s_0) - V_{\lambda}(u_{n-1}) \leq V(s_0) - V_{\lambda}(s_0) + \sum_{k=1}^{n-1} (V_{\lambda}(u_{k-1}) - V_{\lambda_{k-1}}(u_k)). \]

Therefore by Lemma 2.3

\[ V(s_0) - V_{\lambda}(u_{n-1}) \leq \delta_1 + \sum_{k=1}^{n-1} (3\delta_k + \delta_{k+1}) \leq 4 \sum_{k=n}^{\infty} \delta_k. \]

**End of Proof of Theorem A.** Denote \( T_n = q_n + n - 1 \), where \( q_n = \sum_{k=1}^{n} t_k \) (\( q_0 = 0 \)). Then

\[ \frac{1}{T_n + 1} \sum_{j=0}^{T_n} f(s_j) = \sum_{j=0}^{n} \left[ \frac{1}{T_n + 1} \sum_{j=0}^{T_n} f(s_j) \right] \]
\[ \geq \sum_{j=0}^{n} \left( \frac{t_j + 1}{T_n + 1} \right) (V_{\lambda}(u_{j-1}) - 2\delta_j) \geq V(s_0) - \varepsilon \]

by Lemma 2.4. Therefore
\[
\limsup_{T \to \infty} \frac{1}{T + 1} \sum_{t=0}^{T} f(s_t) \geq V(s_0) - \varepsilon. \quad \blacksquare
\]

**Theorem B.** Let \((S, \Gamma, f)\) be a dynamic programming problem as described in the Introduction. Assume \(V(\lambda(s)) \to V(s)\) uniformly in \(s \in S\). Then for every sequence \((s_n)_{n=0}^{\infty}\) with \(s_{n+1} \in \Gamma(s_n)\)

\[
V(s_0) \geq \limsup_{T \to \infty} \frac{1}{T + 1} \sum_{t=0}^{T} f(s_t).
\]

**Proof.** We need the following two lemmas. Lemma 2.5 is proved in Lehrer and Sorin (1991). For the sake of completeness we include a proof here.

**Lemma 2.5.** Let \(0 \leq \alpha_t \leq 1\) for all \(0 \leq t \leq N\). Let \(\varepsilon \geq 0\), and let \(K < \varepsilon N\), where \(t, N, \) and \(K\) are integers. Then there exists an integer \(0 \leq L \leq N - K\) such that

\[
\frac{1}{T + 1} \sum_{t=0}^{T} \alpha_{L+t} \geq A_N(\alpha) - 2\varepsilon \quad \text{for all} \quad 0 \leq T \leq K,
\]

where \(\alpha = (\alpha_t)_{t=0}^{N}\).

**Proof of Lemma 2.5.** Otherwise for every \(0 \leq L \leq N - K\) there exists \(0 \leq T \leq K\) with

\[
\frac{1}{T + 1} \sum_{t=0}^{T} \alpha_{L-t} < A_N(\alpha) - 2\varepsilon.
\]

Therefore we can find a partition \(B_1, B_2, \ldots, B_{p+1}\) of \(\{0, 1, \ldots, N\}\) such that \(\beta_{p+1} \leq K\), and such that

\[
\frac{1}{\beta_i} \sum_{t \in B_i} \alpha_t < A_N(\alpha) - 2\varepsilon \quad \text{for all} \quad i \leq p,
\]

where \(\beta_i\) denotes the number of elements in \(B_i\). Therefore
\[ A_N(\alpha) = \sum_{i=1}^{n+1} \frac{\beta_i}{N+1} \frac{1}{\beta_i} \sum_{i \in B_i} \alpha_i \]
\[ \leq \frac{N + 1 - \beta_{n+1}}{N+1} (A_N(\alpha) - 2\varepsilon) + \frac{\beta_{n+1}}{N+1} \leq A_N(\alpha) - \varepsilon, \]

a contraction. ■

The proof of the next lemma is obvious and is omitted.

**Lemma 2.6.** Assume \( V_s(s) \to V(s) \) for all \( s \in S \). Then \( V(u) \leq V(v) \) for all \( u, v \in S \) with \( u \in T(v) \). ■

We now turn to the proof of Theorem B.

Denote

\[ M = \limsup_{r \to \infty} \frac{1}{T+1} \sum_{i=0}^{T} f(s_i). \]

We show that for every \( \varepsilon > 0 \) \( V(s_0) \geq M - \varepsilon \). Let \( \lambda_0 \) satisfy

\[ |V_{\lambda_0}(u) - V(u)| < \frac{\varepsilon}{4} \text{ for all } u \in S. \]

Let \( K \) be an integer satisfying

\[ (1 - \lambda_0)^2 \sum_{t=0}^{K} \lambda_0^t (t + 1) \geq 1 - \frac{\varepsilon}{4}. \]

Choose \( N \) large enough such that \( (\varepsilon/4)N > K \), and such that \( A_N(f(s)) > M - (\varepsilon/4) \). By (2.3) and Lemma 2.5 there exists an integer \( L \) such that

\[ V(s_L) + \frac{\varepsilon}{4} \geq V_{\lambda_0}(s_L) \geq \left( 1 - \frac{\varepsilon}{4} \right) \left( A_N - \frac{\varepsilon}{2} \right), \]

where \( A_N = A_N(f(s)) \).

Therefore, by Lemma 2.6

\[ V(s_0) \geq V(s_L) \geq A_N - \varepsilon. \]

**Corollary C.** Let \( (S, \Gamma, f) \) be a dynamic programming problem as defined in the Introduction. Assume \( V_\lambda(s) \) converges to \( V(s) \) for all \( s \in S \), and that the function \( \lambda \to V_\lambda(s) \) is nondecreasing. Then \( V = \overline{V} \).
Proof. The proof goes exactly along the lines of the two previous theorems' proofs and therefore is omitted. ■

To conclude this section we remark that Theorem A, Theorem B, and Corollary C, as well as their proofs, remain valid if the assumption \( V_\lambda \rightarrow V \) is replaced by the weaker assumption:

There exists a sequence \( \lambda_n \not\rightarrow 1 \) such that \( V_\lambda \rightarrow V \) (uniformly in Theorem A and in Theorem B, and monotonically in Corollary C).

The "subsequence version" of Theorems A and B yields a necessary and sufficient condition for the existence of the uniform limit of \( (V_\lambda)_\lambda \):

Let \( B(S) \) be the Banach space of all bounded functions \( g \) on \( S \) with \( \| g \| = \sup_{s \in S} |g(s)| \). A subset of \( B(S) \) is relatively compact if its closure is compact. Since any norm convergent subsequence \( V_{\lambda_n} \) must converge to \( V \), we have:

**Corollary D.** Let \( (S, \Gamma, f) \) be a dynamic programming problem as defined in the introduction. Then, \( \lim_{\lambda \rightarrow 1} V_\lambda \) exists uniformly iff \( \{ V_\lambda; \lambda \geq 0 \} \) is a relatively compact subset of \( B(S) \).

3. **The Lower Limit Value**

In this section we construct a dynamic programming problem in which \( V_\lambda \rightarrow V \) uniformly, but for some state \( s_0 \), \( V(s_0) > V(s_0) \), where

\[
V(s_0) = \sup_{s_{-1} \in f(s_0)} \liminf_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} f(s_t).
\]

**An Outline of the Construction:** Every rooted directed tree without terminal nodes naturally defines a dynamic programming problem when we attach payoffs to the nodes.

Imagine a tree in which every node except for the root has an outdegree one, and the root \( s_0 \) has denumerable number of branches. At the \( n \)th branch, the first \( n \) nodes (including the root) are associated with the payoff zero. The proceeding \((n \log n) - n + 1\) nodes have the payoff one, and from the next node the payoff is forever zero.

It is verified that \( \lim_{\lambda \rightarrow 1} V_\lambda(s) = V(s) \) for all nodes \( s \), where \( V(s_0) = 1 \) and \( V(s) = 0 \) for all other nodes \( s \). Although the lower limit value of the root equals zero, this example cannot serve as our counterexample because the convergence of the discounted values is not uniform. In order to achieve uniform convergence we modify the last example, and then use the modified version as a building stone to a recursive construction of our example.
Let \((\delta_n)_{n=1}^\infty\) be a decreasing vanishing sequence. The tree \(T(c, x)\), with the parameters \(0 \leq x \leq c \leq 1\) is the tree obtained from the previous tree by changing the following payoffs: The new payoff of the root is \(x\), and the new payoff of each node in the \(n\)th branch, which was equal one, is now changed to \(\max(c - \delta_n, 0)\). In this modified tree, \(V(s_0) = c\), and \(V(s) = 0\) otherwise.

Set \(T^1 = T(1, 1)\). The tree \(T^2\) is obtained by attaching copies of the trees \(T(c, x)\) to each of the nodes of \(T^1\) except for the root: if \(s\) is a node on the \(n\)th branch, then we attach to it the tree \(T(\max(1 - \delta_n, 0), f(s))\). Let \(s\) be a node in \(T^1\); then there exist two positive integers \(n_1, n_2\) such that the unique path from the root to \(s\) goes through the \(n_1\)th branch of the root up to some node \(u\), and then through the \(n_2\)th branch of the tree that was attached to \(u\) previously. The tree \(T^3\) is obtained by attaching the tree \(T(\max(c - \delta_{n_1} - \delta_{n_2}, 0), f(s))\) to all such nodes \(s\).

Continuing recursively we end up with a nested sequence of trees \((T^n)_{n=1}^\infty\). Our example is the tree \(T^*\) which is the union of these trees.

It is verified that in this tree the convergence of \((V_s)\) is uniform, and that \(V(s_0) = 1\). However, if the decision maker wants to maximize the lower limit value, then he must avoid paths with payoffs of zeros forever. So, if the integers \(n_1, n_2, \ldots\), describe the branches of the path, then they must satisfy \(\sum_{k=1}^\infty \delta_{n_k} \leq 1\). We choose the sequence \((\delta_n)_{n=1}^\infty\) such that the convergence of \(\sum_{k=1}^\infty \delta_{n_k}\) implies that the sequence \((n_k)\) goes rapidly to infinity, which in turn implies that the path must go through a long period of zeros. So the lower limit value is zero in any case.

The following terminology and notations is used in the construction of the example.

A numerical tree is a rooted directed tree with a real valued function (called the payoff function) defined on the set of vertices. Every numerical tree without terminal nodes can be naturally identified with a dynamic programming problem:

The set of states is identified with the set of vertices, and \(\Gamma(s)\) is the set of all vertices \(t\) for which \((s, t)\) is an edge.

We denote by \(G_T, E_T, s_T,\) and \(f_T\) the set of vertices, set of edges, root, and payoff function, respectively, of the numerical tree \(T\). We say that \(T_1 \subseteq T_2\), if \(G_{T_1} \subseteq G_{T_2}\), \(E_{T_1} \subseteq E_{T_2}\), \(s_{T_1} = s_{T_2}\), and \(f_{T_1}\) is the restriction of \(f_{T_2}\) to \(G_{T_1}\). The union numerical tree \(T = \bigvee_{i=1}^k T_i\) of the sequence \(T_1 \subseteq T_2 \subseteq \cdots\) is naturally defined.

Let \(T_1, T_2\) be numerical trees with disjoint set of vertices, and let \(u \in G_{T_i}\) satisfy \(f_{T_i}(u) = f_{T_1}(s_{T_i});\) the numerical tree \(T\), obtained from attaching \(T_2\) to \(T_1\) at \(u\), is defined as follows: Identify \(u\) with \(s_{T_2}\), and then define \(G_T = G_{T_1} \cup G_{T_2}\), \(E_T = E_{T_1} \cup E_{T_2}\), and \(f_T(u) = f_{T_1}(u)\) if \(u \in G_{T_1}\).

Construction of the Tree. The tree \(T\) in our example is the union of
an inductively defined increasing sequence \( T_1 \leq T_2 \leq T_3 \leq \cdots \) of numerical trees.

A few more definitions help to facilitate the description of the induction step.

Let \((A^n)_{n=3}^\infty\) be a sequence of mutually disjoint copies of the set of nonnegative integers. The \(k\)th element in \(A^n\) is denoted by \(a_k^n\). We identify all \(a_k^n\) with \(a_k^0 = a_0\). We now define a numerical tree \(T(\gamma, \beta)\) for every \(\gamma, \beta \geq 0\). The vertices’ set is \(\bigcup_{n=0}^\infty A^n\). The edges are all pairs \((a_k^n, a_{k+1}^n)\), and the payoff function \(f\) is defined as follows: \(f(a_0^n) = \beta\), and \(f(a_k^n) = \max(\gamma - (1/\sqrt{\log n}), 0)\), if \(k \in I_n\), and \(f(a_k^n) = 0\) otherwise, where \(I_n\) is the interval \((n, [n \log n])\) of integers.

**Stage 1.** \(T_1\) is \(T(1, 1)\).

**Stage 2.** Let \(D_2\) be the set of all vertices \(u\) of \(T_1\) whose distance from \(a_0\) is \(1\) (i.e., there exists a directed path with one edge from \(a_0\) to \(u\)). For every \(u \in D_2\) let

\[
\gamma(u) = \max \{f(v) : v \text{ is a vertex of } T_1 \text{ and } v \not\geq u\},
\]

where \(v \not\geq u\) means that there is a directed path from \(u\) to \(v\). The tree \(T_2\) is obtained from \(T_1\) by attaching the tree \(T(\gamma(u), f(u))\) to \(T_1\) at each vertex \(u \in D_2\).

**Stage \(n\).** \(T_n\) is obtained from \(T_{n-1}\) exactly as \(T_2\) is obtained from \(T_1\). That is, we define \(D_n\) to be the set of all vertices in \(T_{n-1}\) whose distance from \(a_0\) is \(n\). For each \(u \in D_n\) we define

\[
\gamma(u) = \max\{f(v) : v \in T_{n-1} \text{ and } v \not\geq u\}.
\]

To get \(T_n\) we attach the tree \(T(\gamma(u), f(u))\) to \(T_{n-1}\) at each \(u \in D_n\).

Finally, define \(T = \bigvee_{n=1}^\infty T_n\). Note that \(\gamma(u)\) is defined now for every \(u \in G_T\).

**Preliminary Remarks**

(a) If \(v \not\geq u\) then \(f(v) \leq \gamma(u)\) (where \(\gamma(a_0) = 1\)). Therefore \(V_\lambda(u) \leq \gamma(u)\) for all \(0 < \lambda < 1\).

(b) Denote \(\lambda_m = 1 - (1/m \sqrt{\log m})\). Then,

\[
\lambda_m(I_m) = (1 - \lambda_m) \sum_{i \in I_m} \lambda_m^i = \lambda_m^{\lceil m \log m \rceil - 1} = \theta_m \to 1,
\]

when \(m \to \infty\).

(c) Because of (b)
\[ V_{\lambda}(u) \geq \theta_m \left( \frac{\gamma(u) - \frac{1}{\sqrt{\log m}}}{\lambda} \right) \to \gamma(u). \]

(d) For all \( \lambda_m < \lambda < \lambda_{m+1} \),

\[ V_{\lambda}(u) \geq \lambda(I_m) \left( \frac{\gamma(u) - \frac{1}{\sqrt{\log m}}}{\lambda} \right) \geq \lambda_m(I_m) \left( \frac{1 - \lambda}{1 - \lambda_m} \right) \left( \frac{\gamma(u) - \frac{1}{\sqrt{\log m}}}{\lambda} \right) \]

\[ \geq \lambda_m(I_m) \left( \frac{1 - \lambda_{m+1}}{1 - \lambda_m} \right) \left( \gamma(u) - \frac{1}{\sqrt{\log m}} \right). \]

(e) Since \((1 - \lambda_{m+1})/(1 - \lambda_m) \to 1\) we get from (a), (b), and the last inequality that \( V_{\lambda}(u) \to \gamma(u) \) as \( \lambda \to 1 \).

(f) Observe that if \( \lambda_m \leq \lambda \leq \lambda_{m+1} \), then by (d)

\[ \gamma(u) - V_{\lambda}(u) \leq \gamma(u) - \lambda_m(I_m) \left( \frac{1 - \lambda_{m+1}}{1 - \lambda_m} \right) \left( \gamma(u) - \frac{1}{\sqrt{\log m}} \right). \]

(g) Set

\[ \epsilon_m = 1 - \lambda_m(I_m) \frac{1 - \lambda_{m+1}}{1 - \lambda_m} + \frac{\lambda_m(I_m)(1 - \lambda_{m+1})/1 - \lambda_m}{\sqrt{\log m}}. \]

Combining \( \gamma(u) \leq 1 \) for all \( u \) and (f) yields that

\[ \gamma(u) - V_{\lambda}(u) \leq \epsilon_m \quad \text{for all } \lambda_m \leq \lambda \leq \lambda_{m+1}. \]

(h) Since \( \epsilon_m \to 0 \), \( V_{\lambda}(u) \to \gamma(u) \) uniformly in all vertices \( u \) of \( T \).

The Main Proposition

Every sequence \(((R_k, y_k))\) of pairs of integers defines a path in the tree \( T \) (starting from \( u^0 \)) as follows:

\( y_1 \) paces on the \( r_1 \)th branch of \( T_1 \), up to a vertex \( a_1 \). Then \( y_2 \) paces on the \( r_2 \)th branch of \( T_{a_1} \) (the subtree that was attached to \( a_1 \) at stage \( y_1 \)), up to the vertex \( a_2 \). Then \( y_3 \) paces on the \( r_3 \)th branch of \( T_{a_2} \) (the subtree that was attached to \( a_2 \) at stage \( y_1 + y_2 \)). And so on.

Note that if a path \( a_0, a_1, a_2, \ldots \) cannot be described as above, then it actually means that for some \( k \) \( y_k = \infty \), and therefore \( \lim_{T \to \infty} (1/T + 1) \sum_{t=0}^{T} f(a_t) = 0 \).

Therefore, if \( \lim \inf (1/(T + 1)) \sum_{t=0}^{T} f(a_t) > 0 \) then the path can be described as above.
Suppose now that \( \lim \inf \left( 1/(T + 1) \right) \sum_{j=0}^{T} f(a_j) > 0 \). Since

\[
1 - \sum_{k=1}^{x} \frac{1}{\sqrt{\log r_k}} \geq \lim \inf \frac{1}{T + 1} \sum_{r=0}^{T} f(a_r)
\]

for all \( K \), we get that

\[
1 - \sum_{k=1}^{x} \frac{1}{\sqrt{\log r_k}} \geq 0.
\]

In particular

\[
\sum_{k=1}^{x} \frac{1}{\sqrt{\log r_k}} < \infty. \quad (3.1)
\]

W.l.o.g. we can assume that for all \( k \), \( y_k \geq r_k \). Otherwise \( y_{k_0} < r_{k_0} \) for some \( k_0 \) and we can enlarge the lower Cesaro limit of the path by replacing it with the path \( ((r_k, y_k))_{k \neq k_0} \).

In the next proposition we show that if \( \lim \inf \left( 1/(T + 1) \right) \sum_{r=0}^{T} f(a_r) > 0 \) then a subsequence of the sequence \( (r_k)_{k=1}^{\infty} \) is rapidly increasing.

**Proposition.** If \((3.1)\) holds then \( \lim \inf_{k \to \infty} d_k = 0 \), where

\[
d_k = \frac{\sum_{k<K} r_k \log r_k}{r_K}.
\]

**Proof of the Proposition.** If the assertion does not hold, then there exists \( d > 0 \) such that

\[
\sum_{k<K} r_k \log r_k > d \quad \text{for all} \quad K.
\]

Define \( f(x) = C x \log x \).

**Claim 1.** For \( C \) big enough \( \sum_{k<K} r_k \log r_k \leq f^{(K-1)}(r_1) \) for all \( K \geq 2 \). (\( f^{(j)} \) is the \( j \)th iterate of \( f \)).

**Proof of Claim 1.** We prove the claim by induction on \( K \).

For \( K = 2 \)
\[ r_i \log r_i \leq C r_i \log r_i. \]

Suppose \( \sum_{k<K} r_k \log r_k \leq f^{(K-1)}(r_i); \) then

\[
\sum_{k<K+1} r_k \log r_k \leq f^{(K-1)}(r_i) + r_{K+1} \log r_{K+1} \\
\leq f^{(K-1)}(r_i) + \frac{1}{d} \left( \sum_{k<K} r_k \log r_k \right) \log \left( \frac{1}{d} \sum_{k<K} r_k \log r_k \right) \\
\leq f^{(K-1)}(r_i) + \frac{1}{d} f^{(K-1)}(r_i) \log \left( \frac{1}{d} f^{(K-1)}(r_i) \right) \leq f^{(K)}(r_i)
\]

if \( C \) is big enough and \( f^{(K-1)}(r_i) \rightarrow \infty \). The latter is true because by the induction hypothesis

\[ f^{(K-1)}(r_i) \geq \sum_{k<K} r_k \log r_k > d r_K \]

and by (3.1) \( r_K \) tends to \( \infty \).

**Claim 2.** If \( f(r_i) \leq A^{r_i^2} \) for some \( A > 0 \), then for all \( K \geq 0 \)

\[ f^{(K+1)}(r_i) \leq (A + K)^{(A+K)^2}. \]

**Proof of Claim 2.** The proof can be obtained by a simple induction and therefore is omitted.

Using Claims 1 and 2 we deduce

\[
\sum_{k=1}^{\infty} \frac{1}{\sqrt{\log r_K}} \geq \sum_{k=1}^{\infty} \frac{1}{\sqrt{\log ((1/d) \sum_{k<K} r_k \log r_k)}} \geq \sum_{k=1}^{\infty} \frac{1}{\sqrt{\log ((1/d) f^{(K-1)}(r_i))}} \\
\geq \sum_{k=1}^{\infty} \frac{1}{\sqrt{\log ((1/d)(A + K - 2)^{(A+K-2)^2})}} \\
\geq \sum_{k=1}^{\infty} \frac{1}{\sqrt{(A + K)^2 \log (A + K) - \log d}} = \infty.
\]

This contradicts (3.1) and the proposition is established.
The Proof

Observe that the average of the payoffs after taking \( r_K - 1 \) zeros in the \( K \)th step (before walking into the nonzero zone of the \( r_K \)th branch) is bounded from above by

\[
\frac{0(r_K - 1) + 1 \sum_{k<K} r_k \log r_k}{r_K - 1 + \sum_{k<K} r_k \log r_k} = \frac{d_K}{1 - (1/r_K) + d_K}.
\]

This is because at each former step step \( k \) the path goes through at most \( r_k \log r_k \) positive numbers which are smaller than 1.

Now, if \( \lim \inf (1/(T + 1)) \sum_{i=0}^T f(a_i) = 0 \), then (3.1) holds, and therefore, by the proposition, \( \lim \inf d_K = 0 \). This in its turn implies that \( \lim \inf (1/(T + 1)) \sum_{i=0}^T f(a_i) = 0 \), a contradiction. This shows that the lower Cesaro limit of any path equals 0 while \( V(a_0) = 1 \).

4. Remarks

As we showed, strong conditions are necessary to ensure \( V \leq \bar{V} \). Several such sufficient conditions were given by Mertens and Neyman (1981), e.g., \( V \leq \bar{V} \) if the vector valued function \( \lambda \rightarrow V_\lambda \) has a bounded variation with respect to the supremum norm on \( B(S) \).

For every \( T \geq 0 \) denote

\[
V_T(s_0) = \sup_{s_0 \in \mathbb{R}^n} \frac{1}{T + 1} \sum_{t=0}^T f(s_t).
\]

It is proved in Lehrer and Sorin (1992) that \( V_\lambda \rightarrow V \) uniformly iff \( V_T \rightarrow V \) uniformly. Therefore, Theorem A, Theorem B, and the counterexample of Section 3 remain valid (with the necessary notational changes) if \( V_\lambda \rightarrow V \) is replaced everywhere by \( V_T \rightarrow V \).

References


