Combinatorics, Probability and Computing (1999) 8, 555–565. Printed in the United Kingdom
© 1999 Cambridge University Press

Searches on a Binary Tree with Random Edge-Weights

ALAN STACEY†

Department of Pure Mathematics and Mathematical Statistics,
University of Cambridge, 16 Mill Lane, Cambridge CB2 1SB, England
(e-mail: a.m.stacey@dpmms.cam.ac.uk)

Received 2 March 1998; revised 2 June 1998

Each edge of the standard rooted binary tree is equipped with a random *weight*; weights are independent and identically distibuted. The *value* of a vertex is the sum of the weights on the path from the root to the vertex. We wish to search the tree to find a vertex of large weight. A very natural conjecture of Aldous states that, in the sense of stochastic domination, an obvious greedy algorithm is best possible. We show that this conjecture is false. We prove, however, that in a weaker sense there is no significantly better algorithm.

1. Introduction

In [1], Aldous considers the usual rooted infinite binary tree, $T = (V, E)$, in which the root, r, has degree 2 and every other vertex has degree 3. To each edge, $e \in E$, is associated a random variable X_e, which is referred to as the *weight* of the edge. The weights are independent and all have the same distribution as some reference random variable, X. Each vertex, v, is considered to have a *value*, S_v, which is equal to the sum of the weights of the edges on the (unique) path from the root to that vertex. So, for example, in Figure 1, $S_r = 0$, $S_s = X_{e_1}$, $S_t = X_{e_2}$, $S_u = X_{e_1} + X_{e_3}$ and so forth.

One wishes to search the binary tree in order to find vertices with high values. At the start, one does not know the weights of any edges, so the only vertex whose value is known is the root: $S_r = 0$. At each step of the search procedure one can *evaluate* (*i.e.*, discover the weight of) one edge which must be incident with a vertex whose value is already known; thereby one discovers the value of a child of that vertex.

Let us denote by V_n the set of vertices whose value is known after n steps. We say that a vertex in V_n is *free* if it has a child which is not in V_n; we shall say that an edge is *free* if it connects a vertex in V_n (necessarily a free one) to a vertex outside V_n. So we have $V_0 = \{r\}$, $|V_n| = n + 1$; and at the $(n+1)$th step of the search, we may pick any free edge $e = \{v, w\}$, with $v \in V_n$ and $w \notin V_n$, and discover the weight X_e. (Which edge we choose

† Work supported in part by Engineering and Physical Science Research Council Grant GR/L15425.

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

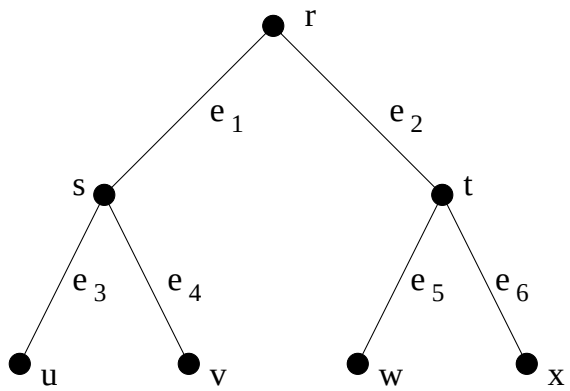


Figure 1 The top of the binary tree

to evaluate may – and usually will – depend on the weights evaluated in previous steps.) We then have $V_{n+1} = V_n \cup \{w\}$. We shall say that we have *expanded* v, and that we have *evaluated* w (and e).

There is a very natural greedy algorithm for finding high values: at each step one picks the highest-valued vertex from amongst the free vertices and evaluates one of its children. A tie-breaking convention may be required: for most purposes it does not matter what this convention is. (For some aspects of the analysis in [1], it is convenient to specify a convention; here, however, we shall not need to do so.)

We let M_n be the largest-valued vertex found after n steps of whatever search procedure is being used, so

$$M_n = \max\{S_v : v \in V_n\}.$$

Given a particular strategy for choosing which vertices to evaluate, M_n becomes a well-defined random variable. In the particular case where the greedy algorithm is being used, we shall use the notation M_n^0.

In [1], Aldous examines the behaviour of the process (M_n^0) and proves a substantial result concerning its asymptotic behaviour – which, of course, depends on the underlying distribution of the edge-weights. In [1] it is conjectured that this algorithm is best possible in a strong (and precise!) sense; as is said in [1], 'it seems intuitively obvious' and 'one would expect [this conjecture] to yield to some simple coupling argument, but the issue seems rather subtle'. Much of the appeal of this conjecture lies in the fact that many people, given a few moments to think about it, conclude that it is obviously true.

In Section 2 of this paper we shall show that the conjecture concerning the optimality of the greedy algorithm is – rather surprisingly – false. In Section 3 we shall show, however, that, in a somewhat weaker sense, there is no significantly better algorithm.

Although our main motivation for studying this problem is its intrinsic appeal, it is worth noting that algorithms on trees with some random structure have been studied in many different contexts. It is natural to think of $\{S_v : v \text{ is a vertex of the tree}\}$, as a tree-indexed random walk: a variety of questions about tree-indexed processes have been considered in recent years. Pemantle [3] is a wide-ranging survey. Further motivation can be found in [1].

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

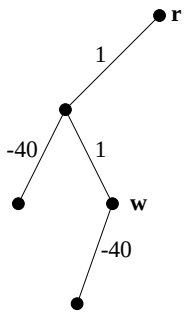


Figure 2 The position after the first four steps of the greedy algorithm

2. A counterexample to the optimality of the greedy algorithm

We begin this section by stating the conjecture precisely.

Conjecture 2.1 (Aldous [1, 5.1]). *Given a particular edge-weight distribution, let M_n^0 denote the maximum vertex value found after n steps of the greedy algorithm, and let M_n' denote the maximum value found using some other algorithm. Then M_n^0 stochastically dominates M_n' (written $M_n^0 \succeq M_n'$). That is to say, for any real x,*

$$\mathbb{P}(M_n^0 \geqslant x) \geqslant \mathbb{P}(M_n' \geqslant x). \tag{2.1}$$

At first sight it seems intuitively clear that this must be true. How could it possibly help us to expand a vertex with a lower value in preference to one with a higher value? To expand on this point, any two rooted subtrees (a rooted subtree being a vertex and all its descendants) have vertex values with an identical distribution apart from a constant factor: given the choice between examining the root of one or another, how could it possibly help us to examine a root where this constant factor is lower?

Nevertheless, in this section we present a counterexample to this conjecture. Although a little detailed work is required, it is small enough to understand fairly easily; however, it is somewhat less easy to see why our intuition – which leads us to regard Conjecture 2.1 as entirely natural – is wrong. After giving the detail of the counterexample we shall give a brief indication of how it was obtained.

The edge distribution we shall use is as follows:

$$\mathbb{P}(X_e = -40) = \epsilon, \quad \mathbb{P}(X_e = 1) = \epsilon, \quad \mathbb{P}(X_e = 8) = 1 - 2\epsilon,$$

where ϵ is small and positive (in fact we shall see later that taking $\epsilon = 1/8$ will do). The '−40' can be replaced by any lesser (*i.e.*, more negative) amount.

We shall suppose that we have used the greedy algorithm for the first four steps, and the weights evaluated have (in turn) been 1, −40, 1, −40, as shown in Figure 2.

We shall compare two algorithms for carrying out the next five steps. One will be the greedy algorithm; the alternative will be to expand the root (rather than w) on the first of these five steps, and then proceed greedily for the remaining four steps. Denoting by M_9^0 and M_9' the maxima found using the greedy and alternative algorithms, respectively,

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

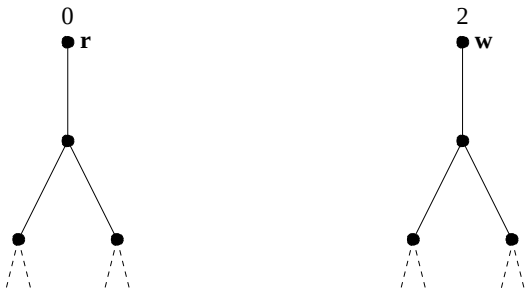


Figure 3 A position equivalent to Figure 2



Figure 4 The alternative algorithm after five steps

we shall show that

$$\mathbb{P}(M_9^0 \geqslant 11) < \mathbb{P}(M_9' \geqslant 11). \tag{2.2}$$

These probabilities, and all the others in this section, are conditional probabilities given that the outcome of the first four evaluations is as described above (*i.e.*, 1, −40, 1, −40). However, we can extend the definition of the two algorithms by saying that, if the first four steps are not successively 1, −40, 1 −40, then both algorithms do the next five steps greedily. Then (2.2) is true with the probabilities regarded as absolute (rather than conditional); this contradicts the optimality of the greedy algorithm (2.1).

To begin, let us note that expanding any free vertex with a value of −39 or −38 cannot possibly contribute to either probability in (2.2) (because $-38+8+8+8+8+8<11$). So we are only interested in r and w, their two unevaluated children (one each) and their descendants: we are effectively reduced to considering two separate trees in which the root of one has value 0 and the root of the other has value 2, as shown in Figure 3.

Let $\Omega = \{-40, 1, 8\}$, and let us denote an outcome of our five-step search by an element of Ω^5, with the weights listed in the order of evaluation. For example, $(1, 8, -40, -40, 8)$ means that the first edge evaluated (recall this is the really the fifth evaluated in the overall procedure, the first four having been assumed as described above) has weight 1, the second edge evaluated has weight 8 and so forth. The probability of this outcome, using our probabilities given earlier, is $\epsilon^3(1-2\epsilon)^2$. If we are using the alternative algorithm – in which we expand r first and then proceed greedily – then this outcome would give us the position in Figure 4 after five steps.

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

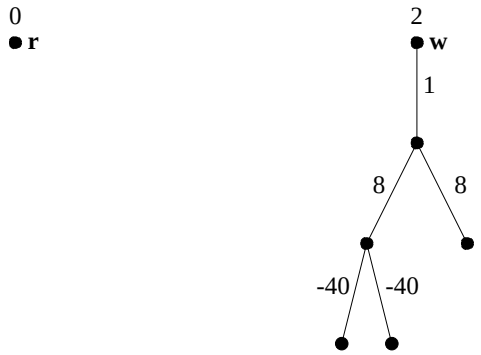


Figure 5 The greedy algorithm after five steps

So in this case we would have $M'_9 = 10$. In keeping with the fact that M'_9 is now regarded as a random variable defined on the probability space Ω^5, we shall write $M'_9(1, 8, -40, -40, 8) = 10$. It is equally easy to see that $M^0_9(1, 8, -40, -40, 8) = 11$; Figure 5 demonstrates this.

It is a straightforward matter to find M^0_9 and M'_9 for each of the 3^5 possible outcomes, and thereby count up all the contributions to the probabilities on both sides of inequality (2.2). In fact, it will be easier to count the contributions to both sides of

$$\mathbb{P}(M^0_9 < 11) > \mathbb{P}(M'_9 < 11), \tag{2.3}$$

which is trivially equivalent to (2.2).

By taking ϵ to be sufficiently small, we can neglect all the terms of high order (in ϵ). Let us count the contributions to both probabilities, starting with the zeroth order terms, until we observe a difference between the probabilities.

0th order terms. The only outcome which does not have probability $O(\epsilon)$ is, of course, $(8, 8, 8, 8, 8)$. With this outcome we clearly have $M^0_9 = 42$ and $M'_9 = 40$, so we obtain no contribution to either of the probabilities in (2.3).

1st order terms. These correspond to outcomes containing four 8s. Again it is easy to see that all such outcomes give both $M^0_9 \geqslant 11$ and $M'_9 \geqslant 11$, so we have no contributions.

2nd order terms. Any outcome which begins $(-40, -40, \ldots)$ has $M^0_9 = M'_9 = 2$ (of course there is only one such outcome which is second order). Counting *all* such outcomes (including those which have fewer than 3 8s) contributes ϵ^2 to both probabilities of interest. All the other second order outcomes are easily seen to give $M^0_9 \geqslant 11$ and $M'_9 \geqslant 11$.

3rd order terms. Here we have to do a little work. Firstly we note that any outcome involving two 8s and no more than one -40 will contribute to neither probability.

If an outcome ω contains two 8s and three -40s, then it is very easy to see that $M^0_9(\omega) \geqslant 11$ if and only if $M'_9(\omega) \geqslant 11$; this is true of any outcome that involves no 1s. If we bother to check them all we see that – ignoring outcomes which begin $(-40, -40, \ldots)$ all of which we counted above – we obtain a contribution of $3\epsilon^3(1-2\epsilon)^2$ to both probabilities.

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

So it remains to consider outcomes which are permutations of two 8s, two −40s and one 1: there are 30 such outcomes to consider.

The three which begin $(-40,-40,\ldots)$ have already been counted. Most of the others make no contribution to either probability: there are three exceptions.

The outcomes $(8,-40,-40,8,1)$ and $(8,-40,-40,1,8)$ both give $M_9^0 = 9$ and $M_9' = 11$; so we get a contribution of $2\epsilon^3(1-2\epsilon)^2$ to $\mathbb{P}(M_9^0 < 11)$ *only*.

The outcome $(1,8,-40,-40,8)$, which was illustrated above, contributes $\epsilon^3(1-2\epsilon)^2$ to $\mathbb{P}(M_9' < 11)$ *only*.

So, adding together all of the third and lower order terms, we see that

$$\mathbb{P}(M_9^0 < 11) = \epsilon^2 + 5\epsilon^3 + O(\epsilon^4),$$

whereas

$$\mathbb{P}(M_9' < 11) = \epsilon^2 + 4\epsilon^3 + O(\epsilon^4).$$

Therefore (2.3) (and hence (2.2)) indeed holds provided ϵ is sufficiently small.

Of course, it is not too hard to see exactly how small ϵ needs to be. Any outcome containing no 8s has both $M_9^0 < 11$ and $M_9' < 11$, so it only remains to consider those outcomes containing precisely one 8. There are 80 such outcomes: even if they all gave $M_9^0 \geqslant 11$ and $M_9' < 11$ then these contributions would be outweighed by the difference in third order contributions (above) provided

$$80\epsilon^4(1-2\epsilon) < \epsilon^3(1-2\epsilon)^2,$$

which reduces to $\epsilon < 1/82$. Of course a more careful analysis greatly improves this; in fact we obtain the requirement $\epsilon < 1/7$.

Taking $\epsilon = 1/8$, we have established the following result.

Theorem 2.2. *Let the binary tree have independent edge-weights (X_e) each with distribution*

$$\mathbb{P}(X_e = -40) = 1/8, \quad \mathbb{P}(X_e = 1) = 1/8, \quad \mathbb{P}(X_e = 8) = 3/4.$$

Then there exists an alternative search algorithm which is not stochastically dominated by the greedy algorithm: using M_9^0 and M_9' to denote the maximum vertex value found after nine steps under the greedy and alternative algorithms respectively, then

$$\mathbb{P}(M_9^0 \geqslant 11) < \mathbb{P}(M_9' \geqslant 11).$$

Hence Conjecture 2.1 *is false.* □

We close this section with a couple of remarks about the counterexample, briefly describing how it was found, and giving a little indication of why our intuition about the truth of Conjecture 2.1 was wrong.

Consider the search problem in which the binary tree is replaced by two disjoint half-lines, that is, trees in which every vertex has exactly one child. This is a rather simpler problem than the search on the binary tree: there are only two options at each stage. One can construct a coupling which demonstrates that, for this simpler search, the greedy algorithm is indeed best possible. To be precise about this coupling, let our underlying

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

probability space for the n-step search be Ω^n (where, as before, Ω is the set of possible edge-weights) endowed with the obvious product measure. Given $\omega = (\omega_1, \ldots, \omega_n) \in \Omega^n$, if our n-step search evaluates the ith edge of the first half-line at some stage then we assign ω_i as its weight; if the ith edge of the second half-line is evaluated then we assign ω_{n+1-i} as its weight. Each possible strategy gives rise to a different random variable M_n, and this procedure defines a simultaneous coupling of all these random variables. It is a reasonably straightforward matter to check that, for each ω, $M_n(\omega)$ is maximized by the greedy algorithm.

We then tried to adapt this coupling to the problem described above, in which we start with two trees, each of which has a root of degree 1 and all other vertices of degree 3, with two different values for the roots, as in Figure 3. The coupling extended naturally to four-step searches in this case, but clearly did not extend to five-step searches. By taking an example where the coupling failed to give the desired result, and fine-tuning it a little, we obtained the above counterexample.

Why does the 'intuitively obvious' conjecture turn out to be false? The glib answer, of course, is that our intuition was simply never right. More seriously, there does not seem to be a very quick way to see why this counterexample (or any other) works, although a few comments might be helpful.

Starting from the position of Figure 3, one has five steps to obtain a vertex with value at least 11. If one starts off very well, $(8, 8, \ldots)$, or very badly, $(-40, -40, \ldots)$, the choice of algorithm (out of the two under consideration) makes no difference. The significant difference between the algorithms occurs when one has mixed luck. If one starts off fairly badly, $(8, -40, -40, \ldots)$, then it becomes critical (in case an 8 and a 1 remain) that under the alternative algorithm there is a vertex of value 2, rather than value 0, remaining to expand. In the coupling of the corresponding algorithms when the structure is two half-lines, such a difference between the algorithms would be balanced out by the advantage the greedy algorithm has when a fairly good start, $(8, 1, \ldots)$, is followed by a fairly bad conclusion, for instance, $(\ldots, -40, -40, 8)$. There is no such symmetry for the binary tree, where a fairly good – $(8, 1, \ldots)$ – start provides so many good vertices to expand that a fairly bad conclusion will give success (*i.e.*, $M_n \geqslant 11$) under either algorithm.

3. An upper bound for the asymptotic behaviour of M_n

The main result of [1] (Theorem 1.1) is that, subject to certain mild conditions on the distribution of X, $n^{-1}M_n^0 \to c$ almost surely, where c is a strictly positive constant (possibly infinite) (and, as before, M_n^0 is the maximum vertex value found by the greedy algorithm in the first n steps). The constant c is defined in a probabilistic way which we explain in due course.

In this section we shall show that $\limsup_{n\to\infty} n^{-1}\mathbb{E}M_n \leqslant c$, whatever algorithm is used. Note that, since M_n is nonnegative, the fact that $n^{-1}M_n^0 \to c$ a.s. implies that $\liminf_{n\to\infty} n^{-1}\mathbb{E}M_n^0 \geqslant c$. Together these two theorems show that, in terms of the growth rate of the expectation of M_n, no algorithm is better than the greedy algorithm; it is perhaps also worth noting that it follows that no algorithm can give $n^{-1}M_n \to c'$ a.s. for some $c' > c$.

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

A key quantity in this section is the random variable Y (or more generally $Y(w)$ for each vertex w) originally defined in [1]. Given a vertex w, and a vertex v which is a descendant of w, we define $S_v^w = S_v - S_w$. So S_v^w is the sum of the edge-weights on the unique path leading from w to v. Then we define the random variable

$$Y(w) = \sup_{(w_i)} \inf_{0 \leqslant i < \infty} S_{w_i}^w$$

where the supremum is taken over all infinite paths $w = w_0 w_1 w_2 \ldots$ leading down from w. Note that $-\infty \leqslant Y \leqslant 0$, and that the different $Y(w)$s are identically distributed. Setting $Y = Y(r)$ it is not hard to see that

$$Y = \inf\{S_v : \text{ the greedy algorithm expands } v \text{ at some point}\}.$$

This is proved in [1] (Lemma 4.1).

We now define the constant referred to above. We set

$$c = \mathbb{E}(X + Y)^+ \tag{3.1}$$

where Y is as just defined and is independent of X.

In order to give a precise statement of the main theorem of this section, we must state two assumptions that we need to make about the distribution of X. They are the mild conditions referred to above, required for Theorem 1.1 of [1]. The first condition is

$$\exists a > 0 \text{ s.t. } \lim_{n \to \infty} n^{-1} \log \mathbb{P}\left(\sum_{i=1}^{n} X^{(i)} \geqslant an \right) > \log(1/2) \tag{3.2}$$

where the $X^{(i)}$ are independent copies of X; observe that a simple subaddivity argument (see, for example, [2, p. 70]) shows that the limit exists.

To motivate this condition, note that if the inequality were reversed for all $a > 0$ then one would have that the maximum of S_v over all v at level n is $o(n)$ almost surely, and so $n^{-1} M_n \to 0$ a.s.

Our second assumption is simpler:

$$\mathbb{E}X^- < \infty. \tag{3.3}$$

Having defined the constant c and our two assumptions about X we can now precisely state our theorem.

Theorem 3.1. *Suppose the edge-weight reference random variable X satisfies* (3.2) *and* (3.3)*, and let c be as defined by* (3.1)*. Then there is a sequence (a_n), with $\lim_{n\to\infty} a_n/n = c$, such that for any strategy we have $\mathbb{E}M_n \leqslant a_n$ for all n.*

So we have that for any strategy, $\limsup_{n\to\infty} M_n/n \leqslant c$, and furthermore this holds uniformly in all possible strategies.

We will need to make use of one result from [1] (see the end of Section 4.2 there).

Lemma 3.2 ([1]). *Suppose that* (3.2) *and* (3.3) *hold. Then $\mathbb{E}Y^- < \infty$.*

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

Now, for each vertex v, we define $S_v^* = S_v + Y(v)$. For an edge $e = \{v, w\}$, where v is a parent of w, we define $X_e^* = (X_e + Y(w))^+$. Note that, for any edge e, $\mathbb{E}X_e^* = c$.

Recall that we used V_n to denote the set of vertices v which have been evaluated (*i.e.*, for which S_v is known) after n steps. We define

$$M_n^* = \max\{S_v^* : v \in V_n\}.$$

Note that $Y(v)$ may depend on the value of edge-weights arbitrarily far below v; hence the fact that v has been evaluated does not imply that S_v^* is known. Letting $\mathscr{F}_n = \sigma(X_1, \ldots, X_n)$ be the knowledge at time n, M_n^* is not, in general, $\mathscr{F}_n$-measurable.

Henceforth we shall assume we are dealing with a particular strategy (and particular edge-weight distribution satisfying (3.2) and (3.3)) so M_n and M_n^* are well-defined random variables. Our first goal is to prove that, for any n, $\mathbb{E}M_n^* \leqslant cn$ (in this respect M_n^* is more well-behaved than M_n). We then show that M_n^* is, for the purposes of Theorem 3.1, a sufficiently good approximation to M_n. We begin with a simple non-probabilistic lemma which relates the starred edge-weights and vertex values.

Lemma 3.3. *Let v be a vertex in the kth generation of the tree and let $r = v_0 v_1 v_2 \ldots v_k = v$ be the path from r to v. Let $e_i = \{v_{i-1}, v_i\}$. Then*

$$S_v^* \leqslant \sum_{i=1}^{n} X_{e_i}^* + S_r^*.$$

Proof. We proceed by induction on k. The result is trivial in the case $k = 0$ (*i.e.*, $v = r$) so assume that $k > 0$ and the result holds for smaller values of k. Then we have

$$S_{v_{k-1}}^* \leqslant \sum_{i=1}^{k-1} X_{e_i}^* + S_r^*. \tag{3.4}$$

Let w denote the child of v_{k-1} other then v_k. Then we have that

$$\begin{aligned} Y(v_{k-1}) &= \min\big(\max\big(X_{e_k} + Y(v_k), X_{<v_{k-1},w>} + Y(w)\big), 0\big) \\ &\geqslant \min\big(X_{e_k} + Y(v_k), 0\big) \\ &= (X_{e_k} + Y(v_k))^-. \end{aligned} \tag{3.5}$$

Combining (3.4) and (3.5) we see that

$$\begin{aligned} \sum_{i=1}^{k} X_{e_i}^* + S_r^* &\geqslant S_{v_{k-1}}^* + X_{e_k}^* \\ &= S_{v_{k-1}} + Y(v_{k-1}) + X_{e_k}^* \\ &\geqslant S_{v_{k-1}} + (X_{e_k} + Y(v_k))^- + (X_{e_k} + Y(v_k))^+ \\ &= S_{v_{k-1}} + X_{e_k} + Y(v_k) \\ &= S_{v_k} + Y(v_k) \\ &= S_{v_k}^*, \end{aligned}$$

which is the required inequality. □

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

It is not hard to apply Lemma 3.3 to prove the following result.

Lemma 3.4. *For all* $n \geqslant 0$,

$$\mathbb{E}M_n^* \leqslant cn.$$

Proof. Let f_i be the ith edge evaluated by the algorithm; note that f_i is itself random. Let $X_i^* = X_{f_i}^*$. It is immediate from Lemma 3.3, and the fact that all the random variables X_i^* are positive, that $M_n^* \leqslant \sum_{i=1}^n X_i^*$. Hence

$$\mathbb{E}M_n^* \leqslant \sum_{i=1}^n \mathbb{E}X_i^*. \tag{3.6}$$

Note that X_i^* is determined by the weight of f_i and the edges below it in the tree. Consequently X_i^* is independent of the knowledge obtained prior to the evaluation of f_i, $\mathscr{F}_{i-1}$, so, regardless of the strategy used, $X_i^* \sim (X+Y)^+$. Hence $\mathbb{E}X_i^* = c$: combining this fact with (3.6) gives the result. □

Note that although conditioning on $\mathscr{F}_{i-1}$ does not alter the distribution of X_i^*, it is not in general true that the random variables X_i^* are independent of each other; note that X_i^* is not itself $\mathscr{F}_i$-measurable.

The preceding lemma states that $\mathbb{E}M_n^*/n \leqslant c$. We now wish to show that M_n^* is not too much smaller than M_n (in expectation).

Lemma 3.5. *Suppose that* (3.2) *and* (3.3) *hold. Then there exists a sequence* (b_n) *such that* $b_n \to 0$ *as* $n \to \infty$ *and such that whatever strategy is used* $(\mathbb{E}M_n - \mathbb{E}M_n^*)/n \leqslant b_n$.

Proof. Suppose (3.2) and (3.3) hold. Let $v_0 = r$ and, for $i > 0$, let v_i be the ith vertex evaluated. So

$$\begin{aligned} M_n^* &= \max\{S_{v_i} + Y(v_i) : i \leqslant n\} \\ &\geqslant \max_{i\leqslant n}(S_{v_i}) + \min_{i\leqslant n} Y(v_i) \\ &= M_n - \max_{i\leqslant n}[Y(v_i)]^-. \end{aligned}$$

So, letting $Z_n = \max_{i\leqslant n}[Y(v_i)]^-$, we have

$$M_n - M_n^* \leqslant Z_n,$$

with

$$\begin{aligned} \mathbb{E}(Z_n) &\leqslant \sum_{k=1}^{\infty} k\mathbb{P}(k-1 < Z_n \leqslant k) \\ &= \sum_{k=0}^{\infty} \mathbb{P}(Z_n > k) \\ &= \sum_{k<M} \mathbb{P}(Z_n > k) + \sum_{k\geqslant M} \mathbb{P}(Z_n > k) \end{aligned} \tag{3.7}$$

for any $M \in \mathbb{N}$.

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press

Now the random variables $[Y(v_i)]^-$, as i varies all have the same distribution, Y^- (although they are not generally independent). So we see that for each k we have that $\mathbb{P}(Z_n > k) \leqslant n\mathbb{P}(Y^- > k)$. Using this to bound the terms in the second sum of (3.7), and using the trivial bound of 1 for each term in the first sum, and dividing both sides through by n, we have

$$\frac{\mathbb{E}(Z_n)}{n} \leqslant \frac{M}{n} + \sum_{k \geqslant M} \mathbb{P}(Y^- > k).$$

Taking $M = \lfloor \sqrt{n} \rfloor$ (any function tending to infinity slower than n will do):

$$\frac{\mathbb{E}(Z_n)}{n} \leqslant \frac{1}{\sqrt{n}} + \sum_{k \geqslant \lfloor \sqrt{n} \rfloor} \mathbb{P}(Y^- > k). \tag{3.8}$$

Let us denote the r.h.s. of (3.8) by b_n. Note that b_n depends only on the distribution function of Y (which is determined by that of X) and does not depend on the particular search strategy adopted. Now, by Lemma 3.2, $\mathbb{E}Y^- < \infty$, and hence $\sum_{k=0}^{\infty} \mathbb{P}(Y^- > k) < \infty$. Therefore $b_n \to 0$ as $n \to \infty$. □

Theorem 3.1 now follows immmediately from Lemmas 3.4 and 3.5, with $a_n = (c + b_n)n$.

4. Conclusion

The second section of this paper showed that the greedy algorithm is not, in the strong sense conjectured in [1], optimal. The third section shows that it is optimal in a rather weaker sense. There are, however, interesting intermediate statements of optimality whose truth we have not determined.

One might try to replace stochastic domination in Conjecture 2.1 by some weaker claim, such as $\mathbb{E}M_n^0 \geqslant \mathbb{E}M_n'$. Although we have not constructed a counterexample to this, it does not seem likely to be true. Looking ahead a little further than the five steps considered by the counterexample of Section 2, quite a bit more can 'go wrong'; it seems reasonable to expect that with some work a slightly larger example could be constructed for which $\mathbb{E}M_n^0 < \mathbb{E}M_n'$.

On the other hand, one might try to strengthen Theorem 3.1 to prove that, in some stronger sense, the greedy algorithm is asymptotically best possible. One such strengthening would be to show that one also has, for any strategy, $\limsup_{n\to\infty} M_n/n \leqslant c$ almost surely. Even this result would be rather weak. It says that no algorithm beats the greedy algorithm by more than $o(n)$; it does not seem unreasonable to expect that a much tighter result holds.

References

[1] Aldous, D. (1992) Greedy search on the binary tree with random edge weights. *Combinatorics, Probability and Computing* **1** 281–293.

[2] Durrett, R. (1996) *Probability: Theory and Examples*, 2nd edn, Wadsworth, Pacific Grove, California.

[3] Pemantle, R. (1995) Tree-indexed processes. *Statistical Science* **10** 200–213.

https://doi.org/10.1017/S0963548399004083 Published online by Cambridge University Press