Discrete Applied Mathematics 10 (1985) 57–77
North-Holland

THE SHORTEST-PATH PROBLEM FOR GRAPHS WITH RANDOM ARC-LENGTHS

A.M. FRIEZE

Dept. of Computer Science and Statistics, Queen Mary College, London University, Mile End Road, London E1 4NS, England

G.R. GRIMMETT

School of Mathematics, University of Bristol, Bristol BS8 1TW, England

Received 31 March 1982
Revised 21 January 1983

We consider the problem of finding the shortest distance between all pairs of vertices in a complete digraph on n vertices, whose arc-lengths are non-negative random variables. We describe an algorithm which solves this problem in $O(n(m+n\log n))$ expected time, where m is the expected number of arcs with finite length. If m is small enough, this represents a small improvement over the bound in Bloniarz [3]. We consider also the case when the arc-lengths are random variables which are independently distributed with distribution function F, where $F(0)=0$ and F is differentiable at 0; for this case, we describe an algorithm which runs in $O(n^2\log n)$ expected time.

In our treatment of the shortest-path problem we consider the following problem in combinatorial probability theory. A town contains n people, one of whom knows a rumour. At the first stage he tells someone chosen randomly from the town; at each stage, each person who knows the rumour tells someone else, chosen randomly from the town and independently of all other choices. Let S_n be the number of stages before the whole town knows the rumour. We show that $S_n/\log_2 n \to 1+\log_e 2$ in probability as $n\to\infty$, and estimate the probabilities of large deviations in S_n.

1. Introduction

We consider the problem of finding the shortest distances between each pair of vertices in a digraph in which all the arcs have non-negative lengths. An n-vertex problem can be solved in $O(n^3(\log\log n)^{1/3}/(\log n)^{1/3})$ time using the algorithm of Fredman [8] (in this paper all logarithms are natural unless explicitly stated otherwise). Fredman's algorithm represents a small improvement in worst-case running time over the $O(n^3)$ algorithms of Dijkstra [6] and Floyd [7].

Spira[10] examined the problem of finding an algorithm with a good expected running time, assuming the existence of a probability distribution on the set of non-negatively weighted digraphs. He proposed an algorithm which has an expected running time of $O((n\log n)^2)$ for quite general distributions. Spira did not deal with the case when arcs may have equal length, and this point was taken up in detail by Bloniarz, Meyer and Fischer [4]. More recently, Bloniarz [3] has improved Spira's method and found an algorithm which runs in $O(n^2 \log n \log^* n)$ expected time, where $\log^* n = \min\{i : \log^i n \leq 1\}$ and $\log^i$ denotes the ith iterate of the logarithm function.

The class of probability distributions for which these results hold is quite general. Informally, all that is required is that the joint distribution of the lengths of arcs in the digraph be independent of the vertices *to* which they point; it may however depend on the vertices *from* which they point. Bloniarz [3] gives the following definition. Let $V_n = \{1, 2, \ldots, n\}$ and let S_n be the set of all digraphs on the vertex set V_n which have non-negatively weighted arcs. We may identify S_n with the set of n by n matrices with entries in $[0, \infty]$; that is, $G \in S_n$ is identified with the n by n matrix $(c_G(i,j) : i, j \in V_n)$, where $c_G(i,j)$ is the length of the arc (i,j). If P is a probability measure on S_n, let

$$F_P(G) = P(\{G' \in S_n : c_{G'}(i,j) \leq c_G(i,j) \text{ for all } i, j \in V_n\}).$$

We say that P is *endpoint-independent* if, for all $i, j, k \in V_n$ and $G \in S_n$, we have that

$$F_P(G) = F_P(G'),$$

where G' is obtained from G by interchanging the lengths of arcs (i,j) and (i,k).

In this paper, we describe an algorithm which runs in $O(n(m + n \log n))$ expected time whenever the joint distribution of the arc-lengths is endpoint-independent; here, m is the expected number of edges of finite length in G. If $m = o(n \log n \log^* n)$, then this is a small improvement over the expected running time of Bloniarz's algorithm [3].

We consider another case in some detail. Suppose that the arc-lengths of G are independent, identically distributed random variables whose common distribution function F is such that one or other of the following conditions holds:

(i) $F(0) > 0$,

(ii) $F(0) = 0$ and $F'(0)$ exists with $F'(0) > 0$.

In this case our algorithm may be modified to run in $O(n^2 \log n)$ expected time.

In our treatment of the shortest-path problem we encounter a problem in combinatorial probability theory which is closely related to the study of the spreads of epidemics and rumours through finite populations. A town contains n people, exactly one of whom has heard a rumour from a neighbouring town, and this rumour spreads according to the following rules. At each epoch of time, each person who currently knows the rumour communicates it to somebody else in the town, chosen randomly from the entire population and independently of all previous choices. It is clear that the number S_n of stages before the whole town knows the rumour is at least $\log_2 n$; we show in Section 5 that, as $n \to \infty$,

$$\frac{S_n}{\log_2 n} \to 1 + \log 2 \quad \text{in probability,}$$

and we investigate the tail of S_n for large n. This process differs from the processes of Daley and Kendall [5] and Berg [2] in that individuals tire of gossiping only when everyone knows the gossip.

2. The algorithm SHORTPATH

The algorithm SHORTPATH described below is a modification of Spira's algorithm.

Let $\Gamma^+(v)$ (respectively $\Gamma^-(v)$) denote those vertices w for which the arc (v, w) (respectively (w, v)) has finite length. Before we do anything else, we construct for each $v \in V$ a list of the set $\Gamma^+(v)$, ordered by increasing value of arc-length $c(v, w)$ (we drop the suffix G from arc-lengths from now on). The procedure RESETNEXT sets pointers to the beginning of each list, and a call to NEXT(v) returns the current vertex, CURR(v), being pointed at, and moves the pointer to the next vertex in $\Gamma^+(v)$. NEXT returns 0 when the end of the list in question has been passed. We shall assume that, in constructing these and later orderings, arcs of equal length are ordered randomly.

We solve a sequence of shortest path problems, taking each vertex in turn as the source vertex s.

For a fixed vertex s, at each stage X denotes a set of vertices for which a shortest path length from s has been determined. Q is a heap (used as a priority queue [1]) of items of the form $\langle x{:}v{:}w\rangle$ where $v \in X$, $w \in V$, and $x = d(v) + c(v, w)$, where $d(v)$ is the length of a shortest path from s to v. The heap Q is ordered by the value x and is such that if $y = \min\{d(v) + c(v, w) \colon v \in X,\ w \notin X\}$, then Q contains an item $\langle y{:}v{:}w\rangle$ with $v \in X$, $w \notin X$.

The basic step is to execute MIN(Q), which removes the minimal object $\langle x{:}cv{:}cw\rangle$ from Q. If $cw \notin X$, then a shortest path from s to cw of length x has now been found; if this is so, then using NEXT(cv), NEXT(cw) we find the next nearest neighbours v, w to cv, cw respectively and add the two corresponding items to Q.

The proposed new feature of this algorithm is that, at those points of the algorithm at which $|X|$ reaches (approximately) $n/2$, $3n/4$, $7n/8, \ldots,$ all arcs of the form (v, w) where $w \in X$ are removed from further consideration, and we then reconstruct Q from the items $\langle d(v) + c(v, w){:}v{:}w\rangle$ where $v \in X$ and $w = \text{CURR}(v)$.

In order to delete arcs efficiently, we store the sorted set $\Gamma^+(v)$ as a doubly-linked list. For each $w \in \Gamma^+(v)$, we store $p_v(w)$ which is the position of w in the list $\Gamma^+(v)$. Thus, if $v \in \Gamma^-(w)$, then w can be removed from $\Gamma^+(v)$ in O(1) time.

Algorithm SHORTPATH

begin
  SORTARCS;
  for $s := 1$ **to** n **do** {use each vertex as a source in turn}

```
  begin
    d(s):=0; X:=∅; Q:=(⟨0:s:s⟩); RESETNEXT; k:=0; r:=⌈½n⌉;
    while r<n do
    begin
      while k<r do
      begin
A:      ⟨x:cv:cw⟩:=MIN(Q);
        v:=NEXT(cv); if v≠0 then INSERT(Q,⟨d(cv)+c(cv, v):cv:v⟩);
        if cw∉X then
        begin
          k:=k+1; X:=X∪{cw}; d(cw):=x;
          w:=NEXT(cw); if w≠0 then INSERT(Q,⟨d(cw)+c(cw, w):cw:w⟩);
        end
      end
      {remove some redundant arcs}
     ⎧for w∈X do for v∈Γ⁻(w) do remove w from Γ⁺(v);
C:   ⎨{naturally this need only be done once for each v∈V_n}
     ⎩re-construct Q using only ⟨d(v)+c(v, CURR(v)):v:CURR(v)⟩ for v∈X;
D:    r:=⌈½(n+r)⌉,
    end
  end
end
```

Note that although the above algorithm computes shortest distances rather than shortest paths, it may easily be adapted to find the latter also, at the cost of increasing the time complexity by a constant factor. The validity of SHORTPATH follows from the validity of Spira's algorithm.

3. Analysis of SHORTPATH

In this section we prove the following theorem.

Theorem 3.1. *If the arc-length distribution is endpoint-independent, then* SHORTPATH *runs in* $O(n(m+n\log n))$ *expected time, where* m *is the expected number of arcs of finite length in* G.

Proof. First fix a source vertex s. Let $p=\lfloor \log_2 n \rfloor+1$ and let $X_1, X_2, \ldots, X_p$ denote the sequence of values of X at successive executions of statement D of the algorithm. Let $X_0=\emptyset$ and $r_i=|X_i|$, $i=0,1,\ldots,p$.

Because of endpoint-independence and the 'clean up' operation C, we have that at line A if $X_i \subseteq X \subset X_{i+1}$, then

$$\text{Prob}(cw \notin X) \geq \frac{n-|X|}{n-r_i}.$$

Thus, the expected number e_k of calls to MIN needed to add the $(k+1)th$ vertex to X satisfies

$$e_k \leq \frac{n-r_i}{n-k},$$

where i is such that $r_i \leq k < r_{i+1}$. Thus the total expected number

$$e = \sum_{k=0}^{n-1} e_k$$

of calls to MIN is bounded above by

$$\begin{aligned} e &\leq \sum_{i=0}^{p-1} \sum_{k=r_i}^{r_{i+1}-1} \frac{n-r_i}{n-k} \\ &\leq \sum_{i=0}^{p-1} (n-r_i) \log\left(\frac{n-r_i}{n-r_{i+1}}\right) + \text{O}(\log n). \end{aligned}$$

Now, $r_{i+1} = \lceil \frac{1}{2}(n+r_i) \rceil$ implies

$$\frac{n-r_i}{n-r_{i+1}} \leq 2 + \frac{2}{n-r_i-1};$$

hence

$$e \leq 2n \log 2 + \text{O}(\log n).$$

For any given s we can divide the time spent finding shortest distances into

(a) calling MIN and INSERT (by the above, this takes $\text{O}(n \log n)$ expected time),

(b) deleting w from $\Gamma^+(v)$, for $w \in X$ and $v \in \Gamma^-(w)$ (this clearly takes $\text{O}(m+n)$ expected time),

(c) reconstructing the heap Q (this takes $\text{O}(pn)$ time as a heap can be constructed in $\text{O}(n)$ time [1]).

Thus, for each s, the above routine requires $\text{O}(m + n \log n)$ expected time. The initial sorting requires $\text{O}(n^2 \log n)$ time, and the theorem is proved. □

We note that Bloniarz, Meyer and Fischer [4] dealt with certain ambiguities in Spira's algorithm by treating equal-length arcs in $\Gamma^+(v)$ in blocks, and processing each such block as soon as the first of its arcs is chosen in A. The effect of this operation is to speed up the runtime of the algorithm, since fewer executions of MIN are performed.

4. Independent arc-lengths

In this section we shall assume that the arc-lengths are independent non-negative random variables with common distribution function F, and shall make use of the following algorithm.

Algorithm RANDOMSHORTPATH

begin
  $p := \min\{n-1, 20\log n\}$
  for $v := 1$ **to** n **do**
  begin
    find the p shortest arcs leaving v and construct a doubly-linked list comprising the vertices to which these arcs point, together with, for each vertex in the list, a pointer to its position in the list
  end;
  for $v := 1$ **to** n **do**
  begin
    construct a list of vertices w, whose list of p nearest neighbours contains v
  end;
    apply SHORTPATH to the digraph with vertices V_n and arcs joining each vertex to its p nearest neighbours as above;
    let $d(v, w)$ denote the distance computed by SHORTPATH from v to w for each pair $(v, w) \in A_n = V_n^2 - \{(v, v): v \in V_n\}$;
    $d\max := \max\{d(v, w): (v, w) \in A_n\}$;
    $e\min := \min\{c(v, w): w$ is the pth vertex in v's list of nearest neighbours$\}$;
    if $d\max \le e\min$ **then** terminate
        else apply Floyd's algorithm to the original weighted digraph.
end

We wonder whether, in line 2 of RANDOMSHORTPATH, 20 may be replaced by 2 without affecting the consequences. The next theorem is our main result.

Theorem 4.1. *Suppose that* $F(0) = 0$. *If* F *is differentiable at* 0 *and* $F'(0) > 0$, *then algorithm* RANDOMSHORTPATH *runs in* $O(n^2 \log n)$ *expected time.*

Proof. The initial sorting and list construction can be carried out in $O(n^2)$ time, as n heaps are constructed and the p minimal elements are drawn from each. By the results of Section 3, the application of SHORTPATH will run in $O(n(np + n\log n)) = O(n^2 \log n)$ time. If $d\max \le e\min$, then the path lengths computed by SHORTPATH are minimal for the complete digraph G, since the arcs omitted are too long to be used in any shortest path. Later in this section, we shall see that

$$\text{Prob}(d\max > e\min) = O(n^{-1}), \tag{4.1}$$

and the result follows immediately, since Floyd's algorithm runs in $O(n^3)$ time. □

Note that the conclusion of Theorem 4.1 holds whenever the arc-length distribution function F is such that $F(0)>0$, without any further assumption on F. It is not difficult to see this, since it may be shown that, with probability $1-O(n^{-2})$, all the arcs used in RANDOMSHORTPATH have length 0 and these arcs form a strongly connected subgraph of the complete digraph on V_n; thus all shortest paths have length 0 with probability $1-O(n^{-2})$.

The rest of the paper is devoted to the proof of equation (4.1). In the following analysis, we often use *real* quantities in positions where *integers* are required. It will be clear that trivial but cumbersome changes may be effected to correct such aberrations and their consequences. We shall prove equation (4.1) first for the case when F is the uniform distribution function on [0,1], and shall then relax this condition as indicated in the statement of Theorem 4.1. Here are some preliminary lemmas.

Lemma 4.2. *Let $X_{(k)}$ be a random variable distributed as the kth smallest of a sample of n independent random variables which are uniformly distributed on* [0,1].

(a) *If $a>0$, $\lambda<1$ and n is large, then*

$$\mathrm{Prob}\left(X_{(a\log n)}<\frac{\lambda a\log n}{n}\right)\le n^{a(1+\log\lambda-\lambda)}.$$

(b) *Suppose that $k_1+k_2+\ldots+k_m\le a\log n$, and $Y_1, Y_2,\ldots,Y_m$ are independent random variables with Y_i distributed as $X_{(k_i)}$ for $i=1,2,\ldots,m$. If $\mu>1$, then*

$$\mathrm{Prob}\left(Y_1+Y_2+\ldots+Y_m\ge\frac{\mu a\log n}{n+1}\right)\le n^{a(1+\log\mu-\mu)}.$$

Proof. (a) If $0<p<1$, k is a positive integer, and $B(n,p)$ is a random variable which is binomially distributed with parameters n and p, then

$$\mathrm{Prob}(X_{(k)}<p)=\mathrm{Prob}(B(n,p)\ge k)$$

since $X_{(k)}<p$ if and only if at least k of the uniform random variables defining $X_{(k)}$ are smaller than p.

We next use the standard inequality (see, for example, Grimmett and Stirzaker [9])

$$\mathrm{Prob}(Z\ge z)\le e^{-tz}\,\mathrm{Exp}(e^{tZ})\quad\text{for } t\ge 0, \tag{4.2}$$

for any random variable Z. Applying (4.2) to $B(n,p)$ we find that, for $k=a\log n$ and $p=\lambda a n^{-1}\log n$,

$$\mathrm{Prob}(X_{(k)}<p)\le e^{-at\log n}\,(1-p+pe^t)^n\quad\text{if } t\ge 0. \tag{4.3}$$

We choose t to minimize the right-hand side of (4.3), giving

$$e^t=\frac{(1-p)a\log n}{(n-a\log n)p}.$$

Substitution into (4.3) leads eventually to

$$\operatorname{Prob}(X_{(k)}<p)\le(\lambda e^{1-\lambda})^{a\log n} \tag{4.4}$$

for all n, and (a) is proved.

(b) The density function $f_k(x)$ of $X_{(k)}$ is

$$f_k(x)=\binom{n}{k}kx^{k-1}(1-x)^{n-k} \quad \text{for } 0\le x\le 1,$$

and hence the ith moment of $X_{(k)}$ is given by

$$\begin{aligned}\operatorname{Exp}(X_{(k)}^i)&=\int_0^1\binom{n}{k}kx^{i+k-1}(1-x)^{n-k}\,dx\\&=\binom{n}{k}k\frac{(i+k-1)!\,(n-k)!}{(n+i)!}\\&\le\frac{k(k+1)\dots(k+i-1)}{(n+1)^i}.\end{aligned}$$

Thus, if $0\le t<n+1$,

$$\operatorname{Exp}(e^{tX_{(k)}})\le\sum_{i=0}^{\infty}\left(\frac{-t}{n+1}\right)^i\binom{-k}{i}=\left(1-\frac{t}{n+1}\right)^{-k}.$$

If $Z=Y_1+Y_2+\dots+Y_m$, then

$$\begin{aligned}\operatorname{Exp}(e^{tZ})&=\prod_{i=1}^{m}\operatorname{Exp}(e^{tY_i})\\&\le\left(1-\frac{t}{n+1}\right)^{-a\log n} \quad \text{if } 0\le t<n+1.\end{aligned}$$

It follows from (4.2) that, for $0\le t<n+1$,

$$\operatorname{Prob}\left(Z\ge\frac{\mu a\log n}{n+1}\right)\le\left(1-\frac{t}{n+1}\right)^{-a\log n}\exp\left(-\frac{t\mu a\log n}{n+1}\right).$$

We choose $t=(n+1)(1-\mu^{-1})$ in order to minimize the right-hand side above, obtaining

$$\operatorname{Prob}\left(Z\ge\frac{\mu a\log n}{n+1}\right)\le(\mu e^{1-\mu})^{a\log n}. \qquad \square$$

Lemma 4.3. *Suppose that the arc-lengths $c(v,w)$ of G are independent random variables which are uniformly distributed on* [0,1], *and let $a(v,k)$ be the length of the kth shortest arc leaving vertex v. Then*

$$\operatorname{Prob}(\exists k\ge 19.70\log n,\ v\in V_n,\ \textit{such that } a(v,k)\le 12.02\,n^{-1}\log n)=O(n^{-1}).$$

Proof. The probability in question does not exceed

$$n\,\text{Prob}(a(v, 19.70\log n) \leq 12.02\,n^{-1}\log n),$$

and the conclusion follows by an application of Lemma 4.2(a). □

Lemma 4.4. *Suppose that the arc-lengths of G are independent random variables which are uniformly distributed on* [0,1], *and, for* $v = 1, 2, \ldots, n$, *let* $d(v)$ *be the length of the shortest path from vertex* 1 *to vertex* v *in* G. *Then*

$$\text{Prob}(\exists v \in V_n \textit{ such that } d(v) > 12\,n^{-1}\log n) = \text{O}(n^{-2}). \tag{4.5}$$

It is clear that equation (4.1) follows immediately (in the case of the uniform distribution) from Lemmas 4.3 and 4.4, since the latter implies that

$$\begin{aligned}\text{Prob}(d\text{max} > 12\,n^{-1}\log n) &= P(\exists\, v, w \text{ with } d(v, w) > 12\,n^{-1}\log n)\\ &= \text{O}(n^{-1}),\end{aligned} \tag{4.6}$$

while the former implies

$$\text{Prob}(e\text{min} \leq 12.02\,n^{-1}\log n) = \text{O}(n^{-1}). \tag{4.7}$$

Proof of Lemma 4.4. We describe an algorithm which, given such a digraph G, constructs a spanning tree T of G which is rooted at vertex 1 and has the property that the lengths of the paths in T from vertex 1 to all other vertices are not greater than 12.02 $n^{-1}\log n$, with the required probability.

We build the tree T recursively. It begins as T_0, the tree containing the single vertex 1. If $(1,v)$ is the shortest arc leaving vertex 1, then vertex v is added to T_0 together with the edge $(1,v)$, and we call this tree T_1. At the next stage we add the shortest arc leaving v and the second shortest arc leaving 1, and so on; we never include an edge which would complete a circuit in the ensuing graph. In the formal description below, NEXT(v) acts as in the algorithm SHORTPATH (except in that the underlying lists of arcs contain all arcs, regardless of whether their lengths are finite or infinite). The algorithm MAKETREE builds a sequence of rooted trees $T_0, T_1, \ldots$ where $T_k = (X_k, A_k)$, until the whole of V_n is spanned.

Algorithm MAKETREE

begin
  $k := 0$; $X_0 := \{1\}$, $A_0 := \emptyset$
while $X_k \neq V_n$ **do**
  begin
    $X_{k+1} := X_k$; $A_{k+1} := A_k$;
    for $v \in X_k$ **do**

```
begin
   w: = NEXT(v);
   if w ∉ X_k then begin
                       X_{k+1}: = X_{k+1} ∪ {w};
                       A_{k+1}: = A_{k+1} ∪ {(v, w)}
                    end
  end;
  k: = k + 1
 end
end
```

Let K be the value of k when this algorithm terminates. For $w \in V_n$, $w \neq 1$, let (v, w) be the unique arc such that $(v, w) \in A_K$, and let $r(w)$ be the position of (v, w) in the ordering of the arcs leaving v. Define $s(1)=0$ and $s(v)= r(u_1)+r(u_2)+\ldots+r(u_m)$ where $P_v=(1, u_1, u_2, \ldots, u_m=v)$ is the unique path from 1 to v in T_K. It is clear from the definition of MAKETREE that

$$\text{for } 0 \leq k \leq K, \quad \text{if } v \in X_k \text{ then } s(v) \leq k. \tag{4.8}$$

Furthermore, it is a consequence of Corollary 5.1, in the next section, that

$$\operatorname{Prob}(K > 4.45 \log n) = \mathrm{O}(n^{-2}). \tag{4.9}$$

The length of the path P_v, above, is the sum of independent random variables $Y_1, Y_2, \ldots, Y_m$ where Y_i is the $r(u_i)$th smallest of $n-1$ independent random variables which are uniformly distributed on [0,1]. We use (4.8), (4.9) and apply Lemma 4.2(b) to find that

$$\operatorname{Prob}(d(v) > 12 n^{-1} \log n) = \mathrm{O}(n^{-3}) \quad \text{for all } v, \tag{4.10}$$

and (4.5) follows.

We have used Corollary 5.1, from the next section, to prove equation (4.1) (and hence Theorem 4.1) for the case when F is the uniform distribution function. Next we show how to adapt the proof to deal with the more general case when

$$F(0)=0, \qquad F'(0)=D>0.$$

Let $\varepsilon > 0$ be such that

$$\frac{12}{D-\varepsilon} < \frac{12.02}{D+\varepsilon}, \tag{4.11}$$

and find $\delta = \delta(\varepsilon) > 0$ such that $0 < \delta < (2D)^{-1}$ and

$$(D-\varepsilon)x \leq F(x) \leq (D+\varepsilon)x \quad \text{for } 0 \leq x \leq \delta.$$

Let F_1 and F_2 be the two distribution functions given by

$$F_1(x) = \begin{cases} (D+\varepsilon)x & \text{if } 0 \leq x \leq \delta, \\ \max\{(D+\varepsilon)\delta, F(x)\} & \text{if } x > \delta, \end{cases}$$

$$F_2(x)=\begin{cases}(D-\varepsilon)x & \text{if } 0\le x<\delta,\\ F(x) & \text{if } x\ge\delta,\end{cases}$$

and note that

$$F_2(x)\le F(x)\le F_1(x) \quad \text{for all } x\ge 0. \tag{4.12}$$

Let $e\text{min}_1$ (respectively $d\text{max}_2$) be the value of $e\text{min}$ (respectively $d\text{max}$) in RANDOMSHORTPATH when the arc-lengths have distribution function F_1 (respectively F_2). We shall show that

$$\text{Prob}\left(e\text{min}_1\le\frac{12.02\log n}{(D+\varepsilon)n}\right)=\text{O}(n^{-1}), \tag{4.13}$$

$$\text{Prob}\left(d\text{max}_2>\frac{12\log n}{(D-\varepsilon)n}\right)=\text{O}(n^{-1}), \tag{4.14}$$

and equation (4.1) follows immediately by (4.11), since (4.12) implies that for all x,

$$\text{Prob}(e\text{min}\le x)\le\text{Prob}(e\text{min}_1\le x),$$

$$\text{Prob}(d\text{max}>x)\le\text{Prob}(d\text{max}_2>x).$$

First consider (4.13), and write

$$e\text{min}_1=\min\{Y_1, Y_2,\dots, Y_n\}$$

where the Y's are independent, identically distributed random variables, each distributed as the pth smallest of $n-1$ independent variables with distribution function F_1. Thus

$$\begin{aligned}\text{Prob}(Y_i\ge\delta \text{ for some } 1\le i\le n)&\le n\,\text{Prob}(Y_1\ge\delta)\\ &\le n\,\text{Prob}(B(n-1,\eta)\le p)\end{aligned}$$

where $\eta=(D+\varepsilon)\delta\in(0,1)$. By standard inequalities concerning the tail of the binomial distribution

$$\text{Prob}(Y_i\ge\delta \text{ for some } 1\le i\le n)=\text{o}(n^{-1}),$$

and thus, except for an event with probability $\text{o}(n^{-1})$, we have that $Y_i<\delta$ for all i. The distribution of Y_1, conditioned upon the event that $Y_1<\delta$, is the same as the distribution of the pth smallest, $X_{(p)}$, of $n-1$ independent random variables which are uniformly distributed on $[0,(D+\varepsilon)^{-1}]$ conditioned on the event that $X_{(p)}<\delta$. Hence

$$\begin{aligned}&\text{Prob}\left(e\text{min}_1\le\frac{12.02\log n}{(D+\varepsilon)n}\right)\\ &\quad\le(1+\text{o}(n^{-1}))\,\text{Prob}\left(e\le\frac{12.02\log n}{n}\right)+\text{o}(n^{-1})\end{aligned}$$

where e is the value of $e$min in RANDOMSHORTPATH when the arc-length distribution is uniform on [0,1]. Equation (4.7) yields (4.13).

An exactly analogous argument holds for $d\max_2$, showing that

$$\operatorname{Prob}\left(d\max_2 > \frac{12\log n}{(D-\varepsilon)n}\right)$$
$$\leq (1+o(n^{-1}))\operatorname{Prob}\left(d > \frac{12\log n}{n}\right) + o(n^{-1}),$$

where d is the value of $d$max in RANDOMSHORTPATH when the arc-length distribution is uniform on [0,1], and (4.6) implies (4.14). □

5. The telephone call problem

Consider the following problem. A town contains exactly n people $1, 2, \dots, n$, each of whom possesses a private working telephone. One person hears a rumour from another town and spreads it in the following way. He chooses someone randomly from the n people in the town (including himself), calls that person and tells him the rumour. The process is said to be in *state k* if exactly k people know the rumour. At the *stage* when the process is in state k, each of these k people who know the rumour selects someone else at random from the n people in the town, independently of all other choices, and calls that person to tell him the rumour. At the next stage the process is in state $k+l$ where l is the number of 'new' people called by the previous k. Thus the number of people who are 'in the know' grows stage by stage until, sooner or later, everyone knows the rumour. Let Y_i be the state of the process after i stages, so that $Y_0 = 1$, and define

$$S_n = \min\{i : Y_i = n\}$$

to be the number of stages until the whole town knows the rumour. We have two results about S_n, dealing with asymptotic behaviour and large deviation estimates for large n, respectively. As usual, all logarithms are natural unless otherwise stated. Also, non-integer-valued quantities are used in contexts where integers are called for; changes which are trivial in spirit but cumbersome in nature are necessary to correct the consequences of this aberration.

Theorem 5.1. *As* $n \to \infty$

$$\frac{S_n}{\log_2 n} \to \log 2 \quad \textit{in probability.}$$

Theorem 5.2. *If* $\gamma > 0$ *then, for all* $\varepsilon > 0$,

$$\operatorname{Prob}(S_n > (1+\varepsilon)\alpha(\gamma)\log_2 n) = o(n^{-\gamma}) \tag{5.1}$$

where $\alpha(\gamma)=1+(\gamma+1)\log 2$. *Furthermore, the constant* $\alpha(\gamma)$ *in* (5.1) *is best possible in the sense that if* $0\le\beta<\alpha(\gamma)$, *then*

$$\mathrm{Prob}(S_n>\beta\log_2 n)\neq o(n^{-\gamma}). \tag{5.2}$$

Before we prove these theorems, we note a corollary which was used in the proof of Theorem 4.1.

Corollary 5.3. *In the notation of the proof of Lemma* 4.4,

$$\mathrm{Prob}(K>4.45\log n)=O(n^{-2}).$$

Proof. In the above process, let Z_k be the set of people who know the rumour after k stages. The evolutions of the sequences $X_0, X_1, \ldots$ and $Z_0, Z_1, \ldots$ differ in various small respects, but it is clear that the X's grow at least as fast as the Z's in the sense that, for all $A\subseteq\{1,2,\ldots,n\}$ and $k\ge 0$,

$$\mathrm{Prob}(X_k\supseteq A)\ge\mathrm{Prob}(Z_k\supseteq A).$$

Writing $a=4.45$, it follows that

$$\begin{aligned}\mathrm{Prob}(K\le a\log n)&=\mathrm{Prob}(X_{a\log n}\supseteq\{1,2,\ldots,n\})\\&\ge\mathrm{Prob}(Z_{a\log n}\supseteq\{1,2,\ldots,n\})\\&=\mathrm{Prob}(S_n\le a\log n)=1-O(n^{-2})\end{aligned}$$

by Theorem 5.2. □

The rest of this section is devoted to the proofs of Theorems 5.1 and 5.2. We shall suppose that person 1 knows the rumour initially, and it is convenient to think of him as the person who makes *all* the telephone calls in sequence; thus, in state i, we allow 1 to make exactly i calls, sequentially, to people chosen independently at random. The following basic facts are useful. Let W_i be the total number calls required to move from state i to state $i+1$. Then

$$\mathrm{Prob}(W_i=r)=\left(\frac{i}{n}\right)^{r-1}\left(1-\frac{i}{n}\right)\quad\text{for } 1\le r<\infty, \tag{5.3}$$

$$\mathrm{Exp}(e^{tW_i})=\frac{n-i}{ne^{-t}-i},\quad\text{if } e^t<\frac{n}{i}, \tag{5.4}$$

and it follows that

$$\mathrm{Prob}(W_i\le x)\ge\mathrm{Prob}(W_j\le x)\quad\text{for all } x \text{ and } i<j, \tag{5.5}$$

$$\mathrm{Exp}(e^{tW_i})\le\mathrm{Exp}(e^{tW_j})\quad\text{for all } i<j \text{ and } t\ge 0. \tag{5.6}$$

The idea of the proof is as follows. We describe a policy which uses

$$(1+\varepsilon)(1+(\gamma+1)\log 2)\log_2 n \quad \text{stages}$$

and which informs the whole population with probability $1-o(n^{-\gamma})$. This policy prescribes 'targets' for each stage and stops when these targets are met; we shall show that the probability that all the targets are met is $1-o(n^{-\gamma})$. The actual process grows at least as fast as that controlled by the targets, and the upper bound for S_n will follow; the lower bound is much easier. The policy may be divided broadly into five main steps, defined in terms of target states to be attained by these steps.

Step I. From state 1 to state N, for some fixed large N.
Step II. From state N to state ξn, where ξ is small and positive.
Step III. From state ξn to state $(1-\eta)n$, where η is small and positive.
Step IV. From state $(1-\eta)n$ to state $n-R$, for some fixed large R.
Step V. From state $n-R$ to state n.

We shall estimate the number of stages required at each step. It turns out that these steps require the following numbers of stages, with the following probabilities (the constants $\alpha_1, \alpha_2, \alpha_3$ are small and positive):

Step I. $O(1)$, with probability $1-o(n^{-\gamma})$,
Step II. $(1+\alpha_1)\log_2 n$, with probability $1-o(n^{-\gamma})$,
Step III. $O(1)$, with probability $1-o(n^{-\gamma})$,
Step IV. $(1+\alpha_2)\log n$, with probability $1-o(n^{-\gamma})$,
Step V. $o(\log n)$, with probability $1-o(1)$, or
$(1+\alpha_3)\gamma\log n$, with probability $1-o(n^{-\gamma})$.

Here and later, o- and O- terms are non-random and refer to the limit as $n\to\infty$. Note that state $n-R$ is attainable in little more than $(1+\log 2)\log_2 n$ stages with probability $1-o(n^{-\gamma})$; it is only the final step which introduces the complication necessary to obtain the required error probability in Theorem 5.2. In the proofs, we shall make considerable use of (4.2).

Fix $\gamma, \varepsilon > 0$ and let $0<\eta<\frac{1}{3}$; later we shall take the limit as $\eta\downarrow 0$.

Step I. Let N be a positive integer such that

$$N > 4\gamma/\eta \tag{5.7}$$

and let T be the number of stages of the process until state N is attained. Then, by (4.2),

$$\operatorname{Prob}(T\ge 2N) \le \operatorname{Prob}(W_1+\ldots+W_N\ge 2N)$$
$$\le e^{-2Nt}\prod_{i=1}^{N}\frac{n-i}{ne^{-t}-i} \quad \text{for } 1\le e^t<\frac{n}{N}$$

$$\leq e^{-2Nt}\left(\frac{n-N}{ne^{-t}-N}\right)^N \qquad \text{by (5.6)}$$

$$\leq e^{-Nt}\left(\frac{n}{n-Ne^t}\right)^N$$

$$=\left(\frac{2N}{n}\right)^N 2^N \qquad \text{choosing } e^t=\frac{n}{2N}$$

$$=o(n^{-\gamma}) \qquad \text{since } N>\gamma.$$

Thus, after $2N$ stages the process is in state N at least, with probability $1-o(n^{-\gamma})$. Our policy requires that we *stop* making calls when state N has been attained, and move on to Step II.

Step II. We set the target of moving from state N to state ξn by multiplying the current state by $(2-\eta)$ at each stage. This is possible, with large probability, so long as $\xi=\xi(\eta)$ is sufficiently small. Suppose that $\xi=\xi(\eta)>0$ is small enough to ensure that

$$\left(\frac{2\xi(2-\eta)}{\eta}\right)^\eta<(1-\eta)^{1-\eta}. \tag{5.8}$$

If $t\geq 0$, then by (4.2), for $e^t<n/(i(2-\eta))$,

$$\text{Prob}(W_i+W_{i+1}+\ldots+W_{(2-\eta)i}\geq i)$$

$$\leq e^{-it}\prod_{j=i}^{(2-\eta)i}\frac{e^t(n-j)}{n-je^t}\leq e^{-\eta ti}\left(\frac{n-(2-\eta)i}{n-(2-\eta)ie^t}\right)^{(1-\eta)i}=(g(t))^i \tag{5.9}$$

where

$$g(t)=e^{-\eta t}\left(\frac{n-(2-\eta)i}{n-(2-\eta)ie^t}\right)^{1-\eta}.$$

Choose $t=\tau$ where

$$e^\tau=\frac{n\eta}{i(2-\eta)}, \tag{5.10}$$

noting that (5.8) implies that

$$1<e^\tau<\frac{n}{i(2-\eta)} \qquad \text{whenever } N\leq i\leq \xi n;$$

we have chosen τ so that $g(\tau)$ is a minimum. From (5.9) and (5.10), if $N\leq i\leq\xi n$, then

$$\text{Prob}(W_i+\ldots+W_{(2-\eta)i}\geq i)\leq\left(\left(\frac{i}{n}\right)^\eta\frac{(2-\eta)^\eta}{\eta^\eta(1-\eta)^{1-\eta}}\right)^i. \tag{5.11}$$

Suppose the process is in state i at some stage, and write $E(i)$ for the event that at the next stage the process is in some state strictly less than $(2-\eta)i$. By (5.11)

$$\mathrm{Prob}(E(i)) \le \left(\nu\left(\frac{i}{n}\right)^{\eta}\right)^{i} \quad \text{where } \nu=\nu(\eta)=\frac{(2-\eta)^{\eta}}{\eta^{\eta}(1-\eta)^{1-\eta}}.$$

Let K be the least integer such that $N(2-\eta)^K \ge \xi n$. Then

$$\mathrm{Prob}\left(\bigcup_{k=0}^{K-1} E(N(2-\eta)^k)\right) \le \sum_{k\le L}\left(\nu\left(\frac{N(2-\eta)^k}{n}\right)^{\eta}\right)^{N} + \sum_{k\ge L}(\nu\xi^{\eta})^{N(2-\eta)^k}$$

$$\le \nu^N K\left(\frac{m}{n}\right)^{\eta N} + \frac{(\nu\xi^{\eta})^m}{1-\nu\xi^{\eta}} \qquad \text{since } \nu\xi^{\eta}<1$$

by (5.8), where $m=N(2-\eta)^L$ and $0\le L<K$. Choose L such that $m=\sqrt{n}$, note that $K=\mathrm{O}(\log n)$, and use (5.7) to find that

$$\mathrm{Prob}\left(\bigcup_{k=0}^{K-1} E(N(2-\eta)^k)\right) = o(n^{-\gamma}).$$

Thus we fail to attain the targets of Step II with probability $\mathrm{o}(n^{-\gamma})$. If we meet these targets, then we attain state ξn at least, in no more than $\log_{2-\eta} n$ stages. We assume that no more calls are made in this step once state ξn has been attained.

Step III. We set the target of getting from ξn to $(1-\eta)n$ in O(1) stages. Choose $0<\nu<\frac{1}{8}\eta$ and define

$$g(x)=(1-\nu)\frac{2x}{1+x}. \tag{5.12}$$

Let $a>0$, $b=g(a)$, and note that

$$g(x)>x \quad \text{whenever } x<1-\tfrac{1}{4}\eta.$$

In the usual way, inequality (4.2) implies that

$$\mathrm{Prob}(W_{an}+\ldots+W_{bn}\ge an)\le (h(t))^n$$

where

$$h(t)=\mathrm{e}^{-(2a-b)t}\left(\frac{1-b}{1-b\mathrm{e}^t}\right)^{b-a} \quad \text{and} \quad 1\le \mathrm{e}^t<b^{-1}.$$

It is easy to check that, if $0<a<1-\frac{1}{4}\eta$, then there exists $\tau=\tau(a)$ such that $1\le \mathrm{e}^{\tau}<b^{-1}$ and $h(\tau)<1$ (just check that $h'(0)<0$). Let K be the least positive integer such that $g^K(\xi)\ge 1-\eta$, where g^K denotes the Kth iterate of g; note that

$$g^K(\xi)\le g(1-\eta)\le (1-\nu)(1-\tfrac{1}{4}\eta),$$

and the fixed point x of g, being the root of the equation $g(x)=x$, is given by

$$x=1-2\nu>(1-\nu)(1-\tfrac{1}{4}\eta).$$

For each $0 \le i < K$, there exists τ_i such that $h(\tau_i) < 1$ and

$$\text{Prob}(W_{g^i(\xi)n} + \ldots + W_{g^{i+1}(\xi)n} \ge g^i(\xi)n) \le (h(\tau_i))^n.$$

Define $h = \max\{h(\tau_i) : 0 \le i < K\}$, and write $E(k)$ for the event that, from state k, we fail to attain state $ng(k/n)$ by the next stage. Then

$$\text{Prob}(E(\xi n) \cup E(g(\xi)n) \cup \ldots \cup E(g^{K-1}(\xi)n)) \le Kh^n = \text{o}(n^{-\gamma})$$

as required. Thus, after a further K stages we attain at least state $(1-\eta)n$, with probability $1 - \text{o}(n^{-\gamma})$; we assume that we stop at exactly state $(1-\eta)n$.

Step IV. The total number of calls required to attain state $n - R$ from state $(1-\eta)n$ is at most

$$S = W_1 + \ldots + W_{n-R}.$$

Choose $R = R(\eta) > 2$ such that

$$R > 2\gamma/\eta. \tag{5.13}$$

In the usual way,

$$\Pi = \text{Prob}(W_1 + \ldots + W_{n-R} \ge (1+\eta)n \log n)$$

satisfies

$$\Pi \le \mathrm{e}^{-t(1+\eta)n \log n} \prod_{i=1}^{n-R} \frac{n-i}{n\mathrm{e}^{-t} - i} \quad \text{if } 1 \le \mathrm{e}^t < \left(1 - \frac{R}{n}\right)^{-1}.$$

Set $\mathrm{e}^{-t} = 1 - R(2n)^{-1}$ to obtain

$$\begin{aligned} \prod_{i=1}^{n-R} \frac{n-i}{n\mathrm{e}^{-t} - i} &= \prod_{j=R}^{n-1} \frac{j}{n - \frac{1}{2}R - (n-j)} \\ &= \prod_{j=R}^{n-1} \left(1 + \frac{R}{2j - R}\right) \\ &\le \exp\left(R \sum_{j=R}^{n-1} \frac{1}{2j - R}\right) \\ &\le \exp\left(R \int_1^n \frac{1}{2x}\,\mathrm{d}x\right) = n^{R/2}, \end{aligned}$$

and thus

$$\begin{aligned} \Pi &\le \left[\left(1 - \frac{R}{2n}\right)^{(1+\eta)n} \mathrm{e}^{R/2}\right]^{\log n} \\ &= \left[\left(1 - \frac{R}{2n}\right)^n \mathrm{e}^{R/2} \left(1 - \frac{R}{2n}\right)^{\eta n}\right]^{\log n} \\ &\le \exp(-\tfrac{1}{2}R\eta \log n) \\ &= \text{o}(n^{-\gamma}) \qquad \text{by (5.13).} \end{aligned}$$

Thus, with probability $1-o(n^{-\gamma})$, at most $(1+\eta)n\log n$ calls are required at this step. But, at each stage, there are at least $(1-\eta)n$ callers, and so the number of stages is at most

$$\frac{1+\eta}{1-\eta}\log n$$

with probability $1-o(n^{-\gamma})$. Assume now that we are in state $n-R$ exactly.

Step V. The total number of calls required to complete the spread of the rumour is

$$T=W_{n-R}+\ldots+W_{n-1}.$$

If we require an error probability which is only o(1), then not many stages are necessary, since

$$\begin{aligned}\text{Prob}(T\geq x) &\leq \text{Prob}\left(\bigcup_{i=1}^{R}\left\{W_{n-i}\geq\frac{x}{R}\right\}\right)\\ &\leq R\,\text{Prob}\left(W_{n-1}\geq\frac{x}{R}\right)\\ &= R\sum_{i=x/R}^{\infty}\left(1-\frac{1}{n}\right)^{i-1}\frac{1}{n}\\ &\sim R\left(1-\frac{1}{n}\right)^{x/R}\to 0 \quad \text{if } x=n\beta(n) \text{ where } \beta(n)\to\infty, \end{aligned} \tag{5.14}$$

giving that the required number of stages is at most

$$\frac{n\beta(n)}{n-R}\sim\beta(n)$$

with probability $1-o(1)$; set $\beta(n)=\log\log n$, say.

To obtain a smaller error probability we require a more sophisticated argument than that of (5.14). Set $a=\gamma+\eta>\gamma$. Then, if $1\leq e^{t}<(1-n^{-1})^{-1}$,

$$\text{Prob}(T>an\log n)\leq e^{-ant\log n}\prod_{j=1}^{R}\frac{j}{ne^{-t}-(n-j)}.$$

Set $t=\tau$ where

$$e^{-\tau}=1-\frac{\beta}{n} \quad \text{and} \quad 0<\beta<1.$$

Then

$$\begin{aligned}\text{Prob}(T>an\log n) &\leq \left(1-\frac{\beta}{n}\right)^{an\log n}\prod_{j=1}^{R}\frac{j}{j-\beta}\\ &\sim A(\beta,R)n^{-a\beta}\end{aligned}$$

where A is a constant. Set

$$\beta=\frac{2\gamma+\eta}{2(\gamma+\eta)}<1$$

to find that

$$\text{Prob}(T>an\log n)=\text{o}(n^{-\gamma}).$$

Hence, with probability $1-\text{o}(n^{-\gamma})$, the number of stages required for this step is at most

$$\frac{(\gamma+\eta)n\log n}{n-R}=(1+\text{o}(1))(\gamma+\eta)\log n.$$

To see that this is the best possible order of magnitude subject to an error probability of $\text{o}(n^{-\gamma})$, note that

$$\text{Prob}(W_{n-1}\geq bn\log n)=\sum_{i=bn\log n}^{\infty}\left(1-\frac{1}{n}\right)^{i-1}\frac{1}{n}$$
$$\sim\left(1-\frac{1}{n}\right)^{bn\log n}\sim n^{-b} \tag{5.15}$$

and thus, if n is large then with probability at least $\frac{1}{2}n^{-\gamma}$, we have that $W_{n-1}\geq\gamma n\log n$, implying that at least $\gamma\log n$ stages are needed to reach state n from state $n-1$.

This final step requires $\log\log n$ stages with probability $1-\text{o}(1)$, or $(1+\text{o}(1))(\gamma+\eta)\log n$ stages with probability $1-\text{o}(n^{-\gamma})$.

We are now ready to finish the proofs of Theorems 5.1 and 5.2. With probability $1-\text{o}(n^{-\gamma})$ all the above steps attain their targets and use in all at most

$$2N+\log_{2-\eta}n+\text{O}(1)+\frac{1+\eta}{1-\eta}\log n+(1+\text{o}(1))(\gamma+\eta)\log n$$

stages, where the o- and O-terms depend on η alone. Thus

$$\text{Prob}\left(\frac{S_n}{\log_2 n}\leq\log_{2-\eta}2+\left(\gamma+\eta+\frac{1+\eta}{1-\eta}\right)\log 2\right)=1-\text{o}(n^{-\gamma})$$

for all small, positive η. Let $\eta\downarrow 0$ to obtain that, for all $\varepsilon>0$,

$$\text{Prob}\left(\frac{S_n}{\log_2 n}\leq(1+\varepsilon)(1+(\gamma+1)\log 2)\right)=1-\text{o}(n^{-\gamma})$$

which proves (5.1).

From (5.1), for all $\varepsilon>0$,

$$\text{Prob}\left(\frac{S_n}{\log_2 n}>(1+\varepsilon)(1+\log 2)\right)\to 0\quad\text{as } n\to\infty \tag{5.16}$$

and Theorem 5.1 will follow as soon as we have shown that, for all $\varepsilon>0$,

$$\mathrm{Prob}\left(\frac{S_n}{\log_2 n}<(1-\varepsilon)(1+\log 2)\right)\to 0 \quad \text{as } n\to\infty; \tag{5.17}$$

this lower bound for S_n is easy to see. If $0<\delta<1$, then we require at least $\log_2(\delta n)$ stages to attain state δn from state 1. Furthermore, the total number

$$U=W_{\delta n}+\dots+W_{n-1}$$

of remaining required calls is such that

$$\mathrm{Prob}(U\le(1-\varepsilon)n\log n)\le\frac{\mathrm{Var}(U)}{((1-\varepsilon)n\log n-\mathrm{Exp}(U))^2}$$

by Chebyshev's inequality. But

$$\mathrm{Exp}(U)=\sum_{i=\delta n}^{n-1}\mathrm{Exp}(W_i)=\sum_{i=1}^{(1-\delta)n}\frac{n}{i}\sim n\log n,$$

$$\mathrm{Var}(U)=\sum_{i=\delta n}^{n-1}\frac{ni}{(n-i)^2}\sim n^2\sum_{i=1}^{\infty}\frac{1}{i^2}=Bn^2,$$

for some constant B, and hence

$$\mathrm{Prob}(U\le(1-\varepsilon)n\log n)\le\frac{Bn^2}{(\varepsilon n\log n)^2}\to 0 \quad \text{as } n\to\infty.$$

Therefore, with probability $1-o(1)$, at least $(1-\varepsilon)n\log n$ calls are required to attain state n from state δn, and this requires at least $(1-\varepsilon)\log n$ stages. Hence

$$\mathrm{Prob}(S_n<\log_2(\delta n)+(1-\varepsilon)\log n)\to 0 \quad \text{as } n\to\infty$$

which implies that (5.17) holds for all $\varepsilon>0$, and Theorem 5.1 is proved.

Finally we show that (5.2) holds for $\beta<\alpha(\gamma)$. Suppose $0<\mu<1$. To attain state $(1-\mu)n$ from state 1 requires at least $\log_2((1-\mu)n)$ stages. To attain state $n-1$ from state $(1-\mu)n$ requires

$$V=W_{(1-\mu)n}+\dots+W_{n-2}$$

calls, and a calculation similar to that of Step IV above shows that

$$\mathrm{Prob}(V\le(1-\mu)n\log n)=o(n^{-\gamma}).$$

By (5.15), if n is large then, with probability at least $\frac{1}{2}n^{-\gamma}$, at least $\gamma\log n$ stages are required to attain state n from state $n-1$; this implies that

$$\mathrm{Prob}(S_n\ge\log_2((1-\mu)n)+(1-\mu)\log n+\gamma\log n)\ge\tfrac{1}{2}n^{-\gamma}(1+o(1)).$$

Choose μ such that

$$\beta\le 1+\log_2(1-\mu)+(\gamma+1-\mu)\log 2<1+(\gamma+1)\log 2$$

to deduce (5.2). □

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms (Addison-Wesley, Reading, MA, 1974).

[2] S. Berg, Random contact processes, snowball sampling and factorial series distributions, J. Appl. Probability 20 (1983) 31–46.

[3] P.A. Bloniarz, A shortest-path algorithm with expected time $O(n^2 \log n \log^* n)$, Technical Report 80-3, Dept. of Computer Science, State Univ. of New York at Albany, August 1980.

[4] P.A. Bloniarz, R.M. Meyer and M.J. Fischer, Some observations on Spira's shortest path algorithm, Technical Report 79–6, Dept. of Computer Science, State Univ. of New York at Albany, December 1979.

[5] D. Daley and D.G. Kendall, Stochastic rumours, J. Inst. Math. Appl. 1 (1965) 42–55.

[6] E.W. Dijkstra, A note on two problems in connection with graphs, Numer. Math. 1 (1959) 269–271.

[7] R.W. Floyd, Algorithm 97: shortest path, Comm. ACM 5 (1962) 345.

[8] M. Fredman, New bounds on the complexity of the shortest path problem, SIAM J. Comput. 5 (1976) 83–89.

[9] G.R. Grimmett and D.R. Stirzaker, Probability and Random Processes (Clarendon Press, Oxford, 1982).

[10] P. Spira, A new algorithm for finding all shortest paths in a graph of positive edges in average time $O(n^2 \log^2 n)$, SIAM J. Comput. 2 (1973) 28–32.