5 Upper bounds for $c(0+)$

This is written (November 2006) in the format of an Appendix to the paper **Percolating paths through random points** by David Aldous and Maxim Krikun, published as ALEA 1 (2006) 89-109. The purpose is to show some simulations and record (for the constant $c(0+)$ in dimension 2)

- a heuristic upper bound of 0.52
- an upper bound of 0.607 which is rigorous up to sampling error in a fixed-size Monte Carlo simulation.

These are based on a certain "crude algorithm" (described later) for finding paths with small average edge-length. This algorithm is not claimed to be close to optimal (developing near-optimal heuristic algorithms for the problems considered in this paper remains an open problem) but is easy to implement.

5.1 A heuristic upper bound

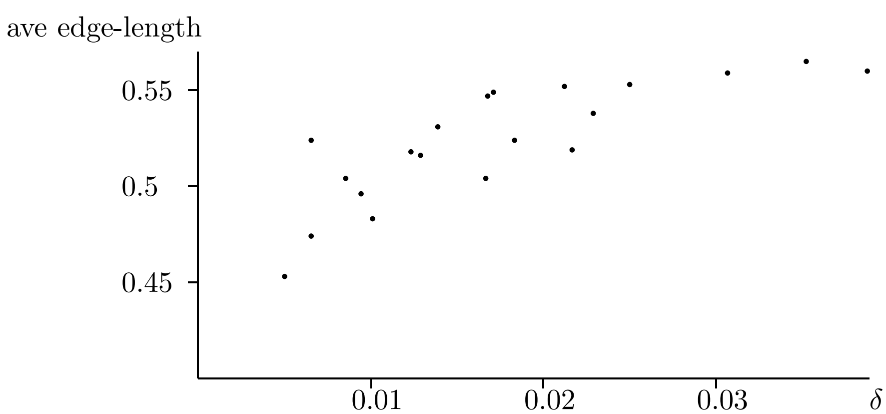


Figure 3. 20 simulations with $n = 9,000$. The crude algorithm picked one path in each simulation; the scatter diagram shows average edge-length in the path versus $\delta =$ (number of edges in path)/9,000.

Figure 3 seeks to study the limit function $c(\delta)$ via 20 simulations with $n = 9,000$ points. The crude algorithm was used to find one path in each simulation; the scatter diagram shows average edge-length in the path versus $\delta =$ (number of edges in path)/9,000. Using such data to estimate $c(\delta)$ is problematical for several reasons, the main reason being that we do not know at what value of δ (as a function of n) occurs the transition from supercritical to subcritical behavior (that is, from paths with order n edges and average edge-length $> c(0+)$ to paths with $o(n)$ edges and average edge-length $< c(0+)$). Nonetheless Figure 3 suggests that (for $n = 9,000$ and paths found by the crude algorithm) the transition value is around $\delta = 0.015$, and then a visual extrapolation of the data for $\delta > 0.015$ suggests a $\delta = 0+$ intercept around 0.52.

5.2 A rigorous Monte Carlo upper bound

Proposition 9 and the definition of β imply the following. If $s > 0$ and $c > 0$ are such that

$$E \min_{\pi \in \mathbf{\Pi}_{0,s}} (\tfrac{\ell(\pi)}{c} - m(\pi)) < -1$$

then $c(0+) \leq c$. So if a random path π in $\mathbf{\Pi}_{0,s}$ found by the crude algorithm satisfies

$$E(\tfrac{\ell(\pi)}{c} - m(\pi)) < -1 \tag{28}$$

then $c(0+) \leq c$. But the latter expectation can be estimated by simulation, for any values of s and c we care to choose. We took $s = 64$ (so $n \approx 4,000$ points) and $c = 0.607$. With 16 repetitions we estimate the left side of (28) as -3.1 with a standard error of 1.1.

5.3 The crude algorithm

To find a good path amongst n points, fix a parameter $\kappa > 0$. Build m-edge forests inductively on $1 \leq m \leq n-1$ via a rule below, ending ($m = n-1$) with a spanning tree. For each pair (i,j) of vertices, the path between them in the spanning tree has some length $\ell(i,j)$ and some number $m(i,j)$ of edges; we choose (as output of the algorithm) the path for the pair (i,j) which minimizes

$$f(i,j) := \tfrac{\ell(i,j)}{\kappa} - m(i,j).$$

Here is the rule for adding edges as m increases. The definition of $f(i,j)$ makes sense for vertex-pairs in the same tree-component. Define at each stage

$$g(i) := \min\{f(i,j) : \ j \text{ in same component as } i\}.$$

Add the edge (i',j') which minimizes

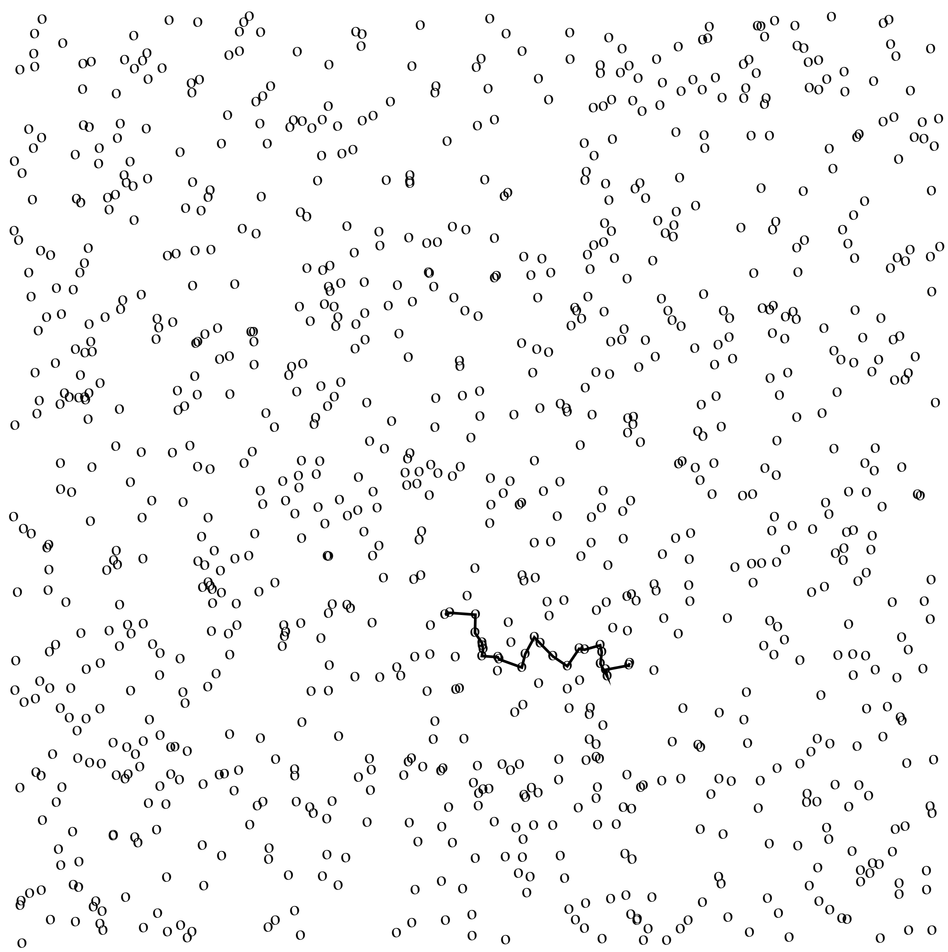
$$g(i') + g(j') + \tfrac{|\xi_{i'} - \xi_{j'}|}{\kappa}$$

subject to i' and j' being in different components.

The algorithm in this form was used to generate the data in Figure 3. For the data in section 5.2 we planted points at diagonally opposite corners and used the path between them in the final spanning tree.

5.4 Pictures of short paths

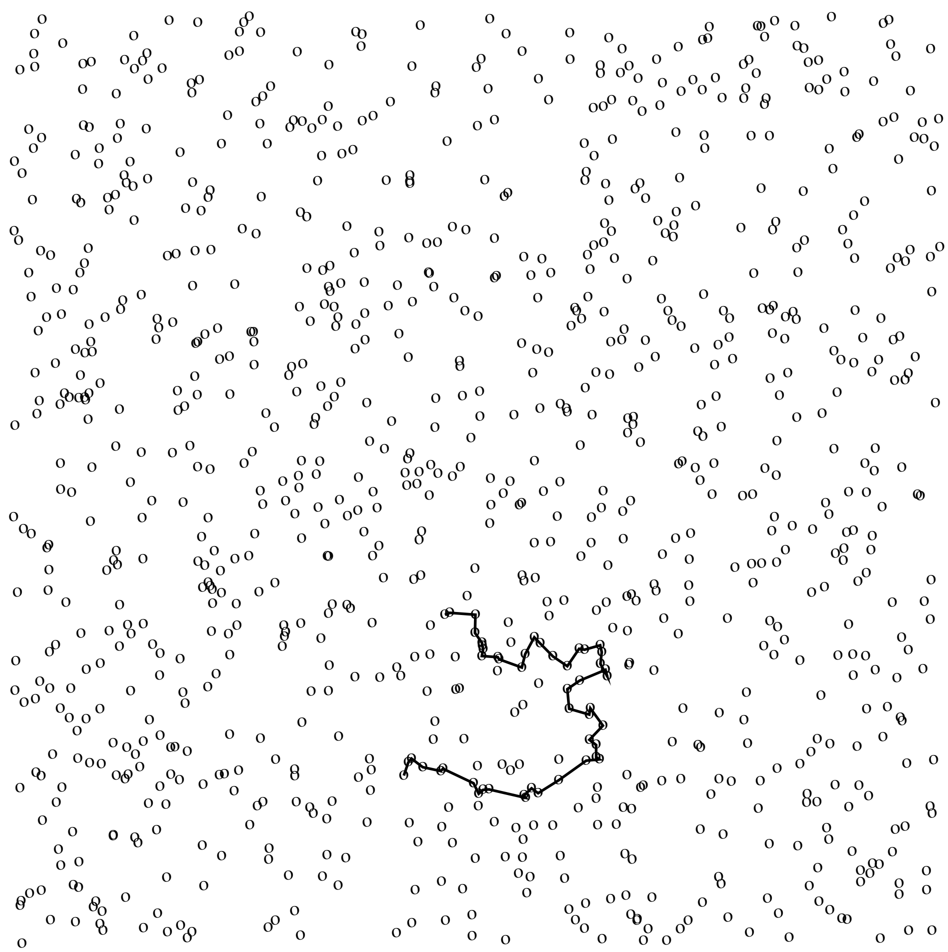
The following pictures show a single simulation with $n = 1000$ points and show paths found by the crude algorithm with different numbers of edges (from 24 to 309) and increasing average edge-lengths (from 0.429 to 0.603).



1000 points

24 edges in path

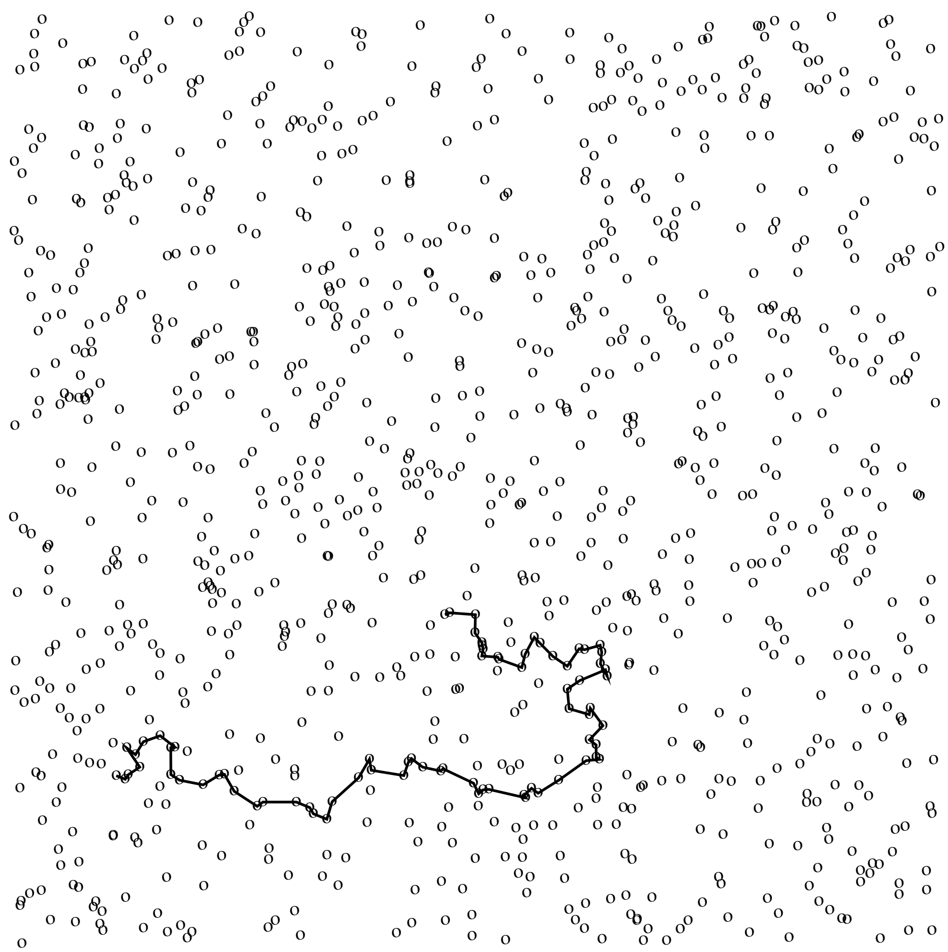
ave edge-length 0.429



1000 points

48 edges in path

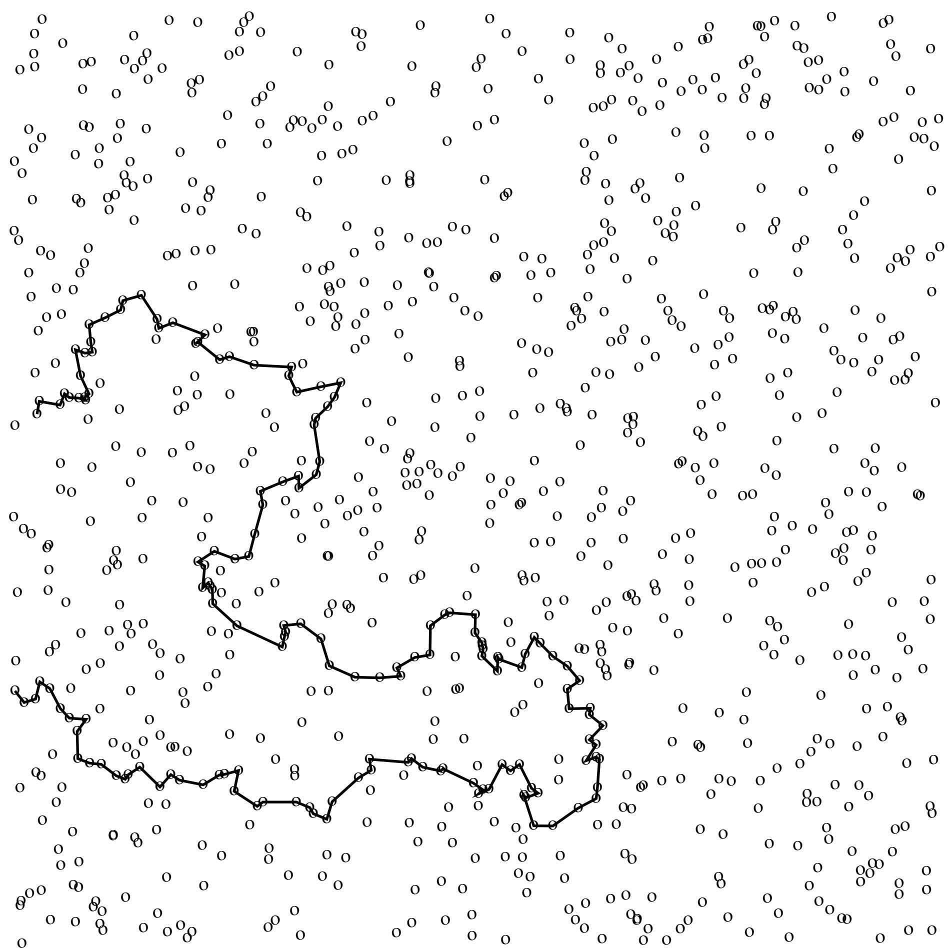
ave edge-length 0.479



1000 points

74 edges in path

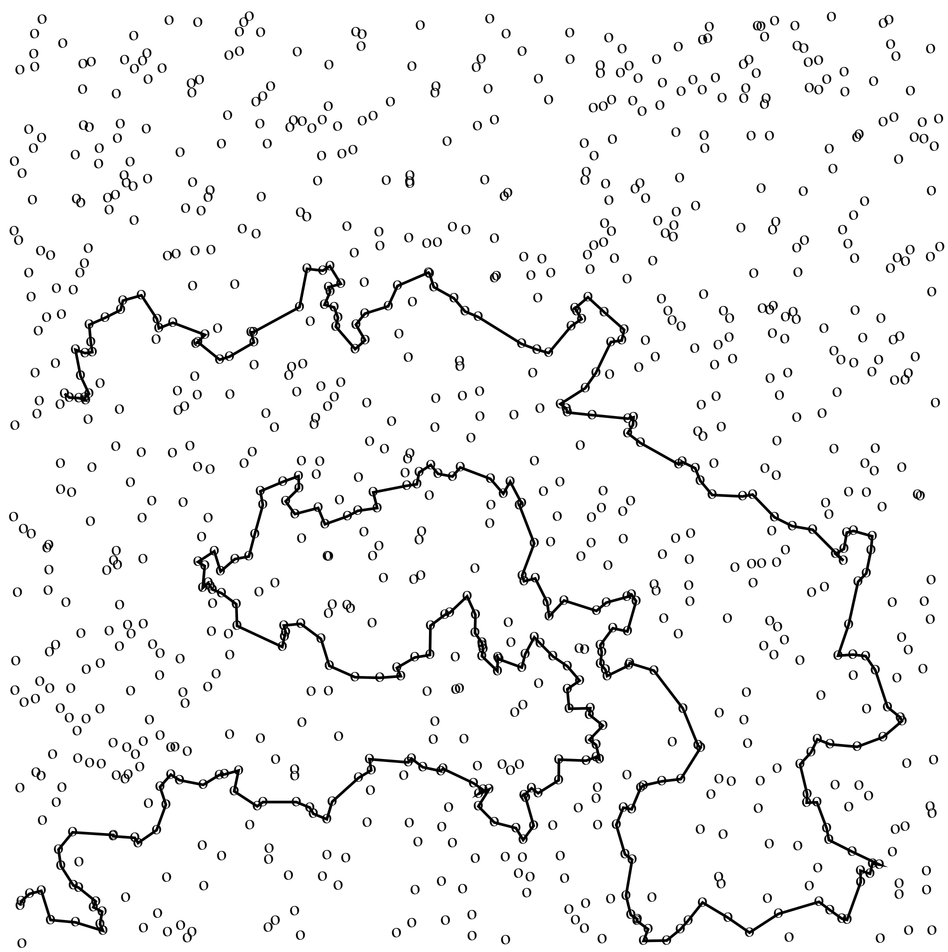
ave edge-length 0.517



1000 points

152 edges in path

ave edge-length 0.564



1000 points

309 edges in path

ave edge-length 0.603