Experimental evaluation of the height of a random set of points in a d-dimentional cube

E. Breimer
breime@cs.rpi.edu

M. Goldberg
goldberg@cs.rpi.edu

B. Kolstad
kolstb@cs.rpi.edu

M. Magdon-Ismail
magdon@cs.rpi.edu

Computer Science Department,
Rensselaer Polytechnic Institute,
110 8th Street, Troy, NY 12180.

Abstract

We investigate the asymptotic behavior of the length $H_d(n)$ of a maximal chain (longest totally ordered subset) of a set of n points drawn from a uniform distribution on unit d-dimensional Eulerian cube $\mathbf{V}_d = [0,1]^d$. For $d \geq 2$, it is known that $c_d(n) = H_d(n)/n^{1/d}$ converges in probability to a constant $c_d < e$, with $\lim_{d\to\infty} c_d = e$. For $d = 2$, the problem has been extensively studied, and, it is known that $c_2 = 2$. c_d is not currently known for any $d \geq 3$. We present an experimental approach to the estimation of c_d. Straight forward Monte Carlo simulations to obtain c_d have already been proposed, and shown to be beyond the scope of current computational resources. We propose two experiments that result from a more careful analysis of the location of maximal chains. The first experiment, *boosting*, improves upon the straight forward Monte Carlo by narrowing the search area for maximal chains and developing specialized algorithms for the search. The second experiment, *co–convergence*, further improves the efficiency and is based upon a conjecture regarding the asymptotic behavior of $c_d(n)$ that we formulate. We provide experimental support for our conjecture. Our experiments yield new estimates for c_d, for $d \in \{3, \ldots, 6\}$ which suggest that the sequence $\{c_d\}$ is increasing in d. *MRSC # 06A07,65C50*

1 Introduction

Erdös and Szekeres proved in [7] that every permutation of $\{1, 2, \ldots, p^2 + 1\}$ has either an increasing or decreasing subsequence of length $p+1$. This result inspired Ulam to investigate the expected value L_n of the length of a longest increasing subsequence in a random permutation of n numbers. The Monte Carlo simulations conducted by Ulam ([14]) led him to conjecture that the answer asymptotically approaches $c\sqrt{n}$ for some constant $c > 1$. Based on more extensive computations, Baer and Brock ([1]) conjectured that $c = 2$. Hammersley in [8], proved that $\pi/2 \leq c \leq e$, and presented heuristic arguments for the value $c = 2$. Logan and Shepp proved in [9] that $c \leq 2$, and Pilpel [12] presented an elementary proof of this inequality. Finally, Verśik and Kerov ([15]) settled the conjecture by proving $c \geq 2$. The structure of the distribution of L_n has been studied in [2], [5], [11].

The multidimensional generalization of the monotone subsequence problem was first considered by Steel in [13], who conjectured the existence of a constant that generalizes the Hammersley's constant c to the d-dimensional case. The correct generalization was found by Bollobás and Winkler in [4]. Let $\mathbf{V}_d = [0, 1]^d$ be the unit cube in d dimensions and let n random points $\mathbf{x}(1),\ \mathbf{x}(2), \ldots,\ \mathbf{x}(n)$ be chosen independently from the uniform distribution on $\mathbf{V}_d$. These points form the underlying set of a *random order* $\mathbf{P}_d(n)$ with a partial ordering given by $\mathbf{x}(i) \leq \mathbf{x}(j)$ if and only if $\mathbf{x}_k(i) \leq \mathbf{x}_k(j)$ for all $k = 1, \ldots, d$. Let the height $H_d(n)$ be the number of elements in a longest chain (totally ordered subset) of $\mathbf{P}_d(n)$. We are interested in the asymptotic behavior of the random variable

$$c_d(n) = \frac{H_d(n)}{n^{1/d}}. \tag{1}$$

Bollobás and Winkler ([4]) have demonstrated that $c_d(n)$ converges in probability to a constant c_d ($d \geq 0$). For convenience, we restate their theorem here.

Theorem 1.1 ([4]) *For $d = 1, 2, \ldots$, define $c_d = \limsup_{n\to\infty} E[c_d(n)]$, then*

1. *$c_d < e$.*
2. *For all $\epsilon > 0$ and $\delta > 0$, there exists $M(\epsilon, \delta)$ such that if $n > M(\epsilon, \delta)$ then*

$$P\left[\left|\frac{H_d(n)}{n^{1/d}} - c_d\right| \leq \epsilon\right] > 1 - \delta \tag{2}$$

3. *$\lim_{d\to\infty} c_d = e$.*

Except for $c_2 = 2$, no other c_d is currently known. Further, it is not even known whether or not the sequence $\{c_d\}$ is monotonically increasing in d. It is not hard to conceive a computational experiment that would, hopefully, lead to approximate values of c_d, for some $d > 2$.

Experiment 1:

1. For a large integer n, randomly generate a set S of n points from the uniform distribution on $\mathbf{V}_d = [0, 1]^d$.

2. Compute the height $H_d(S)$ of S, the length of a maximal chain and $c_d(S) = H_d(S)/n^{1/d}$.
3. Repeat these computations K times to obtain $c_d(S_1), c_d(S_2), \ldots, c_d(S_K)$.
4. Compute the mean, $\overline{c}_d(n)$, and the standard deviation, $\sigma_d(n)$, of the $c_d(S_i)$'s.

For K sufficiently large, one expects that $\overline{c}_d(n)$ will be close to $E[c_d(n)]$ (to within $\sigma_d(n)/\sqrt{K}$), and if n is sufficiently large, then $E[c_d(n)]$ should be close to c_d. The determination of whether n is sufficiently large is a difficult problem. One usually runs an experiment for the largest value of n that is practically feasible, given the current computational resources. Unfortunately, however, it has been discovered through experimental investigations ([10]) that the values of n that need to be considered in order to get a good approximation for c_d are not computationally feasible. In particular, according to [10], $n = 10^9$ points for $d = 3$ is not sufficiently large to suggest an estimate for c_3. Thus, we would need to consider n at least as large as 10^{10}. Since the standard dynamic programming algorithm for computing the length of a maximal chain is quadratic in n, solving even one example with $n = 10^{10}$ points would typically take a few weeks on a modern workstation. In order to develop reasonable statistics, one would need several years. An equally acute problem is with the memory requirements for such an experiment, namely, storing 10^{10} d-dimensional vectors in RAM is impossible on almost any advanced workstation, and storing them on disk would substantially slow down the algorithm.

Hence, the main challenge for an experimental approach to estimating c_d lies in the design of feasible computational experiments that estimate $c_d(n)$ for very large n, even if it is infeasible for Experiment 1. In this paper, we propose an approach that addresses both the computational efficiency and the memory issues. The core idea behind our approach is the observation that a maximal chain "close" to the diagonal of the unit cube $\mathbf{V}_d = [0,1]^d$ must exist as $n \to \infty$. We present a theoretical justification of this observation and use this fact to construct sequences that converge to c_d. Based upon geometrical considerations, we *conjecture* that each of these sequences converges at essentially the same rate. We present experimental data to support this conjecture, and then use this conjecture to derive, from the data, estimates for $\{c_d\}$, for $d = 3, 4, 5, 6$. The intervals for c_d for those values of d are shown in Table 1. Our estimates suggest that $\{c_d\}$ is a monotonically increasing sequence in d, a fact that has previously not been suggested by any experiments.

	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$
c_d	[1.998,2.002]	[2.363,2.366]	[2.514,2.521]	[2.583,2.589]	[2.607,2.617]

Table 1: Estimates of c_d. The intervals indicate the range in which we estimate c_d to lie.

The outline of the paper is as follows. The next section describes two approaches to the design of experiments that build upon Experiment 1. The presentation is informal, and we do not describe every detail of all attempts that were made. The only exception is for the approach that actually yielded estimates for c_d. In section 3, we present the data supporting our conjecture, in addition to the data used for deriving the estimates for c_d. All technical details, proofs of three theorems, and many plots are presented in the appendix.

Acknowledgment. We are grateful to Catherine McGeogh for introducing us to the problem and sharing the information about her experiments.

2 Location of Maximal Chains

Since a longest chain is only a very small subset of the set of n random points in $\mathbf{V}_d(r)$, we can expect to gain an advantage over the straightforward simulation in Experiment 1 by restricting our search for such a chain to a small region where we are likely to find the chain. Thus, our first task is to identify a region in the cube which with probability tending to 1 as $n \to \infty$ contains a maximal chain. It is intuitively apparent that a good candidate for such a region is the set of points that are "close" to the diagonal, say, at most a distance r from the diagonal. The justification of this observation is given in Theorem 2.1, which we prove using a modification of Bollobás and Winkler's proof of Theorem 1.1.

Let $\mathbf{V}_d(r)$ be the region of $\mathbf{V}_d = [0,1]^d$ obtained by translating a cube of side r along the diagonal as shown in Figure 1. One can show (Theorem A.4 in the appendix) that the volume $V_d(r)$ of $\mathbf{V}_d(r)$ is given by

$$V_d(r) = r^d + dr^{d-1}(1-r). \tag{3}$$

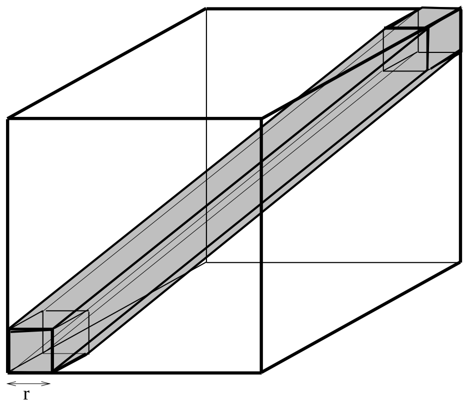


Figure 1: Diagonal volume element obtained by translation of a cube of side r

Let n points be sampled uniformly in $\mathbf{V}_d(r)$ to form a random partially ordered set $\mathbf{P}_d(n,r)$. Denote the height of this set by $H_d(n,r)$ and define the random variable $c_d(n,r)$ by

$$c_d(n,r) = \frac{H_d(n,r)}{(n/V_d(r))^{1/d}} \tag{4}$$

Intuitively, we expect that if one were to generate $n^* = \lceil n/V_d(r) \rceil$ points in the whole cube $\mathbf{V}_d$, then about n of them would fall in $\mathbf{V}_d(r)$. If n^* is large, a maximal chain should exist in $\mathbf{V}_d(r)$, thus $H_d(n, r)$ should approximate $H_d(n^*)$, the height of $\mathbf{P}_d(n^*)$. Therefore, for n sufficiently large,

$$c_d(n, r) \approx \frac{H_d(n^*)}{n^{*1/d}} = c_d(n^*) \tag{5}$$

Hence, for fixed r, $c_d(n, r)$ should converge to c_d. This is exactly the claim of the next theorem, which is essentially an analog of Theorem 1.1.

Theorem 2.1 *Fix $d \in \{1, 2, \ldots\}$ and $r > 0$. For all $\epsilon > 0$ and $\delta > 0$, there exists $M_r(\epsilon, \delta)$ such that if $n > M_r(\epsilon, \delta)$ then*

$$P\left[|c_d(n, r) - c_d| \leq \epsilon\right] > 1 - \delta \tag{6}$$

where the constants c_d are given in Theorem 1.1.

Proof: See the proof of theorem A.1 in the Appendix. ■

Essentially, Theorem 2.1 states that for every $r > 0$, there is a sufficiently large n such that a set of n points generated uniformly in a region close to the diagonal "represents" a much larger "virtual set" generated uniformily in the whole cube. Thus, computing the height of this smaller set would be equivalent (asymptotically) to computing the height of the larger virtual set. This result has several interesting implications for our experiments. The first is, of course, that we can restrict our search for a longest chain to points close to the diagonal. The second is that there is no need to generate points in the whole cube. Since we are interested exclusivly in the points that are close to the diagonal, we only need to generate these diagonal points in a way that would be consistent with having generated a larger number of points in the entire cube. A third, and crucial implication, is that if one fixes an $r > 0$, then eventually, for large enough n, a maximal chain should fall in this region close to the diagonal.

Thus, for a fixed $r > 0$, a Monte Carlo simulation to determine $\overline{c}_d(n, r)$ can be used to suggest a value for c_d, provided n is sufficiently large. One might hope that the experiment with a large and infeasible n^* that yields a good estimate for c_d would be equivalent to an experiment with a feasible $n \approx n^* V_d(r)$, for some $r > 0$. Thus, our general approach will be to estimate $E[c_d(n, r)]$ for various n and r by Monte Carlo simulation, and use these values to construct a final estimate for c_d.

3 Boosting

Not every combination of n and r is computationally equivalent to an $n^* = \lceil n/V_d(r) \rceil$. In fact, for fixed n, as $r \to 0$, $c_d(n, r) \to (rn)^{1-1/d} d^{1/d}$, which approaches zero. This simply indicates that r is too small for the given n, and it is unlikely that a maximal chain of a random set with n^* points in $\mathbf{V}_d$ is located inside $\mathbf{V}_d(r)$. On the other hand, if n is fixed and r is sufficiently large, but less than 1, then it is likely that about n out of n^* points fall in $\mathbf{V}_d(r)$ and a maximal chain exists within even a smaller distance of the diagonal than r. These observations suggest that given n, there exists an optimal $r = r_{opt}(n)$, which yields[1] a maximal value for $c_d(n, r_{opt})$. In the next section, we present an algorithm for computing $c_d(n, r)$ which requires $O(rnd)$ memory and runs in $O(rn^2d)$ time. Thus, it is of practical importance to select as small an r as possible for which $c_d(n, r)$ approximates c_d. We are thus led to the following experiment

Experiment 2 (Boosting):

1. Select a maximal feasible n.
2. Set $r = 1$, and obtain an estimate for $\bar{c}_d(n, r)$.
3. Decrease r ($r \leftarrow r - \epsilon$, for a suitably chosen ϵ) and re-estimate $c_d(n, r)$. Repeat until $\bar{c}_d(n, r)$ stops increasing.
4. Determine $r_{opt}(n)$, the r after which $\bar{c}_d(n, r)$ started to decrease.
5. If $r_{opt}(n)n^2d$ is much less than the available resources, increase n ($n \leftarrow n + n_0$, for a suitably chosen n_0). Compute $\bar{c}_d(n, r)$. Go back to step 3.
6. Once a maximal n has been reached, and its associated optimal r, $r_{opt}(n)$ has been determined, compute $\bar{c}_d(n, r_{opt})$ to the desired accuracy. Output this value as the estimate for c_d.

As the name of Experiment 2 suggests, given a maximal n for Experiment 1 (determined by computational resources), we can *boost* up to a higher n^* by going to a smaller r_{opt} and, perhaps, further increasing n as described in Experiment 2, while still using the same amount of computational resources. Unfortunatelly, boosting does not quite yield a computationally feasible experiment for estimating c_d. The root of the problem is that $r_{opt}(n)$ is not sufficiently small, thus the corresponding n^* is not sufficiently large. Additionally, due to statistical fluctuations, $c_d(n, r)$ needs to be computed many times to obtain a reliable estimate for $\bar{c}_d(n, r)$. Furthermore, in order to obtain $r_{opt}(n)$ to a reasonable accuracy, $\bar{c}_d(n, r)$ needs to be computed for many r, hence making the determination of $r_{opt}(n)$ computationally intensive, even for a feasible n. As an example, for $d = 3$, our simulations indicate that $r_{opt}(n)$ lies in the range $[0.08, 0.11]$; thus, to reach $n^* = 10^{10}$, we would need to process sets of size close to 10^8 or larger. The result is that boosting to an acceptable n^* is still "close" to infeasible.

[1] In fact, our experiments suggest that $c_d(n, r)$ has a single maximum as a function of r.

4 Co-convergence

Theorem 2.1 suggests yet another approach to estimating c_d. Select a set $\{r_i\}_{i=1}^T$ of values for r and consider T sequences $\{\bar{c}_d(n, r_i)\}$ for $i = 1 \dots T$. By Theorem 2.1, each of these sequences converges to the same limit c_d as $n \to \infty$. The problem now is one of estimating the common limit of these T sequences. Suppose $\bar{c}_d(n)$ $(= c_d(n, 1))$ converges at some rate $f(n)$ to c_d:

$$\bar{c}_d(n) = c_d - f(n). \tag{7}$$

Figure 6 suggests that chains in $\mathbf{V}_d(r)$ can be constructed by summing chains in the sub-cubes of side r along the diagonal. Hence the convergence of the chains in $\mathbf{V}_d(r)$ should be closely related to the convergence of the sum of chains in the diagonal sub-cubes, which in turn should be closely related to the convergence in each sub-cube. This convergence is governed by $f(n)$ and, thus, we are lead to *conjecture* that the order of convergence of $c_d(n, r)$ for all $r > 0$ is essentially the same.

Conjecture There exists a function $\mu(r) > 0$, such that

$$\forall r > 0,\ \bar{c}_d(n, r) = c_d - \mu(r) f(n) + o(f(n)) \tag{8}$$

The goal of Experiment 2 was to obtain the smallest r, namely $r_{opt}(n)$, for which $\mu(r_{opt})$ is a minimum. According to our conjecture, $r_{opt}(n)$ should be independent of n, at least asymptotically, which could be a first test of the conjecture. The conjecture suggests a strong interdependence between the sequences, and we might be able to exploit this interdependence in order to get a more accurate estimate of c_d. The traditional approach to obtaining the convergence point of the sequence $c_d(n)$ would be to assume that $f(n)$ has a certain form and then obtain a value for c_d consistent with this assumption and with the observed values $\bar{c}_d(n)$. The success of this kind of an approach depends largely on the validity of the assumption on $f(n)$. Our conjecture allows us to estimate c_d without estimating $f(n)$. Given r_i, r_j $(r_i \neq r_j)$, and n, (8) implies that the following two equalities hold simultaneously

$$\begin{cases} \bar{c}_d(n, r_i) &= c_d - \mu(r_i) f(n) + o(f(n)) \\ \bar{c}_d(n, r_j) &= c_d - \mu(r_j) f(n) + o(f(n)) \end{cases}$$

Resolving this system with respect to $f(n)$, we have

$$\bar{c}_d(n, r_i) = (1 - A(r_i, r_j))\, c_d + A(r_i, r_j) \bar{c}_d(n, r_j) + o(f(n)). \tag{9}$$

where $A(r_i, r_j) = \mu(r_i)/\mu(r_j)$. In other words, $\forall i, j \in [1, T]$ $\bar{c}_d(n, r_i)$ linearly depends on $\bar{c}_d(n, r_j)$, up to $o(f(n))$. Further, from the functions $\bar{c}_d(n, r_i)$ and $\bar{c}_d(n, r_j)$, one can estimate $A(r_i, r_j)$ from the slope of this dependence, and $(1 - A(r_i, r_j))\, c_d$ from the intercept. c_d can then be obtained by dividing this intercept by 1−slope. We have thus constructed an estimate for c_d *without* having to make any statements about $f(n)$. Further, from (9), we see that the convergence to linear behavior is at a rate $o(f(n))$, whereas the convergence of the sequences themselves is at the rate $f(n)$. Thus, the linear behavior will materialize at smaller n than the actual convergence. Hence, we expect to extract more accurate estimates for c_d in this way, given the computational resources. We are thus led to the following experiment.

Experiment 3 (Co-convergence):

1. Select a set $\mathcal{T} = \{r_1, \ldots, r_T\}$ of values for r.
2. Select a set $\mathcal{N} = \{n_1, \ldots, n_L\}$ of values for n. Let $\mathcal{N}_h = \{n_1, \ldots, n_h\}$ for $h = 1, \ldots, L$.
3. Compute $\bar{c}_d(n, r)$, $\forall r \in \mathcal{T}$ and $\forall n \in N$.
4. $\forall i, j \in [1, T]$ $(i \neq j)$, perform an analysis on the pair of sequences $\{\bar{c}_d(n, r_i)\}$ and $\{\bar{c}_d(n, r_j)\}$ for $n \in \mathcal{N}_h$ to obtain the slope, $A_d(h, i, j)$, and the intercept, $B_d(h, i, j)$, for $h = 2, \ldots, L$.
5. Evaluate $c_d(h, i, j) = B_d(h, i, j)/(1 - A_d(h, i, j))$ for $h = 1, \ldots, L$.

   /*$c_d(h, i, j)$ is a sequence of estimates for c_d; this sequence should converge to c_d at the rate at which the linear behavior arises, *i.e.*, $o(f(n))$. */
6. Compute $e_d(i, j)$, the value to which $c_d(h, i, j)$ converges with respect to h.
7. Repeat steps 4–6 to compute $e_d(i, j)$ for all $\binom{T}{2}$ distinct pairs $i, j \in \mathcal{T}$.
8. Let $L_d = \min_{i,j \in \mathcal{T}} e_d(i, j)$ and let $U_d = \max_{i,j \in \mathcal{T}} e_d(i, j)$.
9. Output the interval $[L_d, U_d]$ as an estimate for c_d.

Since the success of Experiment 3 largely depends on the validity of the conjecture, (8), we discuss the motivation for the conjecture and how one might experimentally verify it. Consider the quantity $c_d(n, r)$. Remember that n points in $\mathbf{V}_d(r)$ is "probabilistically" equivalent to n^* points in $\mathbf{V}_d$ and since $H_d(n, r) \leq H_d(n^*)$, we conclude that $\bar{c}_d(n, r) \leq c_d(n^*) = c_d - f(n/V_d(r))$. A lower bound can be obtained by considering the $1/r$ diagonal sub-cubes (Figure 6) inside $\mathbf{V}_d(r)$. There are about $n_c = nr^d/V_d(r)$ points in each subcube, with heights given by $H_c = (n_c)^{1/d}(c_d - f(n_c))$. Since $H_d(n, r) \geq H_c/r$, we have that $c_d(n, r) \geq c_d - f(nr^d/V_d(r))$. Thus, setting $\lambda_r = 1/V_d(r)$ and $\gamma_r = r^d\lambda_r$, we conclude (heuristically) that

$$c_d - f(\gamma_r n) \leq c_d(n, r) \leq c_d - f(\lambda_r n) \tag{10}$$

providing the motivation for the observation that $c_d(n, r)$ converges at essentially the same rate for all r. For large r, we expect the upper bound to be somewhat tight and for small r, we expect the lower bound to be somewhat tight. Hence, starting from large r, by decreasing r, we initially expect to get better values for c_d, and then continuing to decrease r leads to worse performance, indicating that an optimal r exists. Further, the bounds indicate that

$f(\lambda_r n) \leq f(\gamma_r n)$, suggesting that $f(\cdot)$ is monotonically decreasing (or that $c_d(n)$ is increasing in n). These were exactly the observations that led to boosting (Experiment 2).

We now discuss the experimental verification of the conjecture. We have seen that the conjecture implies a linear relationship (up to $o(f(n))$). Suppose that we observe such a relationship, can we then conclude that the conjecture holds? The following theorem tells us that we can.

Theorem 4.1 *Define a set of sequences $c(n,r)$ for every $r \in (0,1]$ and suppose that each sequence converges with respect to n to some value c independent of r. Suppose also that $c(n,r)$ is differentiable with respect to r. The following two statements are equivalent.*

1. *$\forall r \in (0,1],\ c(n,r) = c - \mu(r)f(n)$ for some functions $\mu(r) > 0$ and $f(n)$;*
2. *$\forall r_i, r_j \in (0,1],\ (r_i \neq r_j), c(n,r_i) = c(1 - A(r_i,r_j)) + A(r_i,r_j)c(n,r_j)$ for some differentiable function $A(r,r')$.*

PROOF: See the proof of Theorem A.2 in the appendix. ■

Thus, not only does the conjecture imply a very specific linear relationship between all pairs of sequences, but, if for some value of c, that same linear relationship holds for all pairs of sequences (*i. e.*, all $r_i \neq r_j$), then the conjecture holds. Experimentally, this means that $c(h, r_i, r_j)$ in Experiment 3 should converge (with respect to h) to the same value for all distinct pairs r_i, r_j, or, equivalently, the interval $[L_d, U_d]$ should be small. If this were to happen, then not only would we provide experimental justification for the conjecture, but we would also obtain an accurate estimate for c_d.

5 Algorithms

Both experimental approaches described in the previous section attempt to get more accurate estimates for c_d by effectively "accessing" higher n without actually computing with the higher n. Both techniques rely on efficient algorithms for computing $c_d(n,r)$.

The first task is to generate n points chosen independently from a uniform distribution in $\mathbf{V}_d(r)$. A trivial algorithm is the one that generates random points in $\mathbf{V}_d$ and keeps only those that fall in $\mathbf{V}_d(r)$, continuing until n points in $\mathbf{V}_d(r)$ have been generated. This can be highly inefficient, especially if r is small, as the acceptance rate will be extremely small. A more efficient approach is to generate random points in $\mathbf{V}_d(r)$ itself. This can be done quite efficiently, and further, it is possible to generate the points in a sequential manner so that the dynamic programming algorithm for computing $H_d(n,r)$ keeps in memory only a small portion of the total of n points, those that are necessary for executing the algorithm.

The second task is to compute $H_d(n,r)$. The standard dynamic programming algorithm has computational complexity $O(dn^2)$ ([6]). However, since the points are generated in a sequential manner, it is possible to design an algorithm that maintains a *working set* that takes advantage of the sequential point generation. Operations need only be performed on this working set, resulting in a factor of r reduction in both computational complexity and memory requirements.

5.1 Generation of Input Points

The generation of points is illustrated in Figure 2 (a), for the case of $d = 2$. First, an "origin" point, $\mathbf{t}(i) = (t(i), t(i), \ldots, t(i))$, is generated along the diagonal. Then one of the hypercubes $\mathbf{p}_k(i)$, $k = 1, \ldots, d$ of dimension $d-1$ is chosen at random and a point is generated from a uniform distribution on this hypercube. In this way one can generate n points in $\mathbf{V}_d(r)$. The probability density of the origin point coordinate $t(i)$ is uniform up to $t(i) = 1 - r$ and

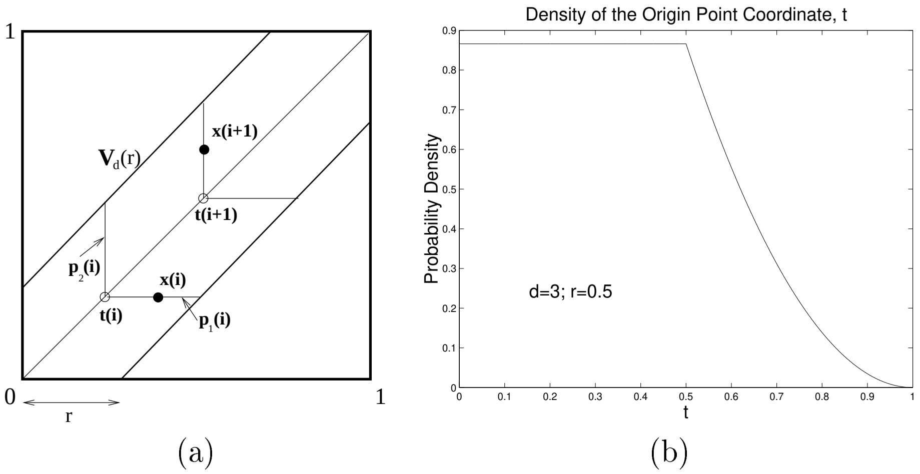


Figure 2: (a) The sequential generation of points in $\mathbf{V}_d(r)$ illustrated for the case $d = 2$. (b) Probability density of origin points for d=3, r=0.5

then decaying like $(1 - t(i))^{d-1}$ for $t(i) > 1 - r$. This density is shown in Figure 2 (b), for the case $d = 3$ and the case $r = 0.5$. Further, instead of generating the origin points in a random order, we generate the n order statistics for the origin points.

Algorithm 1 (Sequential generation of input points):

1. Set $i = 1$ and $t_{prev} = 0$.
2. Generate the i^{th} origin point $\mathbf{t}(i) = (t(i), t(i), \ldots, t(i))$ where $t(i) \in [t_{prev}, 1]$ is the i^{th} order statistic (given that the $(i-1)^{th}$ order statistic was at t_{prev}) of n points

generated from the distribution of origin points such as the one shown in Figure 2.

3. Generate a random vector $=\{v_1, v_2, \ldots, v_d\}$ where each v_i is in $[0, \min\{r, 1-t_i\}]$
4. Generate a random integer k from $\{1, 2, \ldots, d\}$ and set $v_k = 0$
5. Generate input point $\mathbf{x}(i) = \{t(i) + v_1, t(i) + v_2, \ldots, t(i) + v_d\}$.
6. Set $t_{prev} = t(i)$, $i = i + 1$ and go back to step 2 if $i \leq n$.

The detailed implementation of step 2 to generate the order statistics for the origin points is given in the appendix in section B.1. The correctness of the algorithm is given by theorem A.3 in the appendix. The next proposition states two (almost trivial) ordering properties that the points $\mathbf{x}(1), \ldots, \mathbf{x}(n)$ have. While these properties might seem trivial, they are of key importance to the algorithm for computing the height of the n points in $\mathbf{V}_d(r)$, as we shall see in the next section.

Proposition 5.1 (Ordering Properties) *Suppose points $\mathbf{x}(1), \ldots, \mathbf{x}(n)$ are generated according to the algorithm above. Then,*

1. *If $\mathbf{t}(i) \geq \mathbf{x}(j)$, then, for all $k \geq i$, $\mathbf{x}(k) \geq \mathbf{x}(j)$.*
2. *If $i < j$, it cannot be that $\mathbf{x}(i) \geq \mathbf{x}(j)$, more specifically, $P[\mathbf{x}(i) \geq \mathbf{x}(j)] = 0$.*

PROOF: The first claim follows because for $k \geq i$, $\mathbf{x}(k) \geq \mathbf{t}(i)$. Let the origin point of $\mathbf{x}(i)$ be $\mathbf{t}(i)$ and let $j > i$. One of the components of $\mathbf{x}(i)$ is $t(i)$, therefore that component of $\mathbf{x}(i)$ must be less than the corresponding component of $\mathbf{x}(j)$ (with probability 1), proving the second statement. ∎

5.2 Computing $H_d(n, r)$

Although the ordering of the origin points does not guarantee that the projected input points $\{\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(n)\}$ are ordered, it does guarantee the ordering properties stated in proposition 5.1. Property 2 ensures a newly generated point can never be below a previously generated point. Therefore, the height of newly generated points can be immediately computed without considering future points. At iteration i, property 1 ensures $\mathbf{x}(i)$ has a height of at least $h(\mathbf{x}(j)) + 1$ for all $\mathbf{x}(j) < \mathbf{t}(i)$. Among these input points, the $\mathbf{x}(j)$ with maximum height is identified as the *barrier point* and the rest are discarded. The remaining input points (all $\mathbf{x}(j) \geq \mathbf{t}(i)$ plus the *barrier point*) define the *working set* for iteration $i+1$. The height of $\mathbf{x}(i+1)$ can be determined by inspecting only the points in the *working set*.

$H_d(n, r)$ is computed using the following procedure where t and x are the origin point and projected input point as generated by sequential generator described above. $h(x)$ is the height of the maximal chain ending at point x, and $x_{barrier}$ is the *barrier point*.

Algorithm 2 (Computation of $H_d(n, r)$):

1. Set $h_{max} = 1$.
2. Using Algorithm 1, generate $t(1)$ and $\mathbf{x}(1)$ and add $\mathbf{x}(1)$ to the *working set*.
3. Set the height of $\mathbf{x}(1)$ to 1 ($h(\mathbf{x}(1)) = 1$).
4. For $i = 2$ to n
   (a) Using Algorithm 1, generate $t(i)$ and $\mathbf{x}(i)$.
   (b) Initialize the height of $\mathbf{x}(i)$ to 1 ($h(\mathbf{x}(i)) = 1$).
   (c) For every point $\mathbf{x}(j)$ in the *working set*
      i. If $\mathbf{x}(i)$ is above $\mathbf{x}(j)$ and $h(\mathbf{x}(j)) >= h(\mathbf{x}(i))$ then set $h(\mathbf{x}(i)) = h(\mathbf{x}(j))+1$.
      ii. If $\mathbf{x}(j)$ is below $t(i)$, if the *barrier point* is defined, and if $h(\mathbf{x}(j)) > h(x_{barrier})$, then remove the current *barrier point* and replace it with $\mathbf{x}(j)$.
      iii. If $h(\mathbf{x}(j)) <= h(x_{barrier})$ then remove $\mathbf{x}(j)$ from the *working set*
      iv. If $\mathbf{x}(j)$ is below $t(i)$ and the *barrier point* is not defined then set $\mathbf{x}(j)$ to be the *barrier point*.
   (d) Add $\mathbf{x}(i)$ to the *working set*.
   (e) If $h(\mathbf{x}(i)) > h_{max}$ then set $h_{max} = h(\mathbf{x}(i))$.
5. Return $h_{max}/n^{1/d}$.

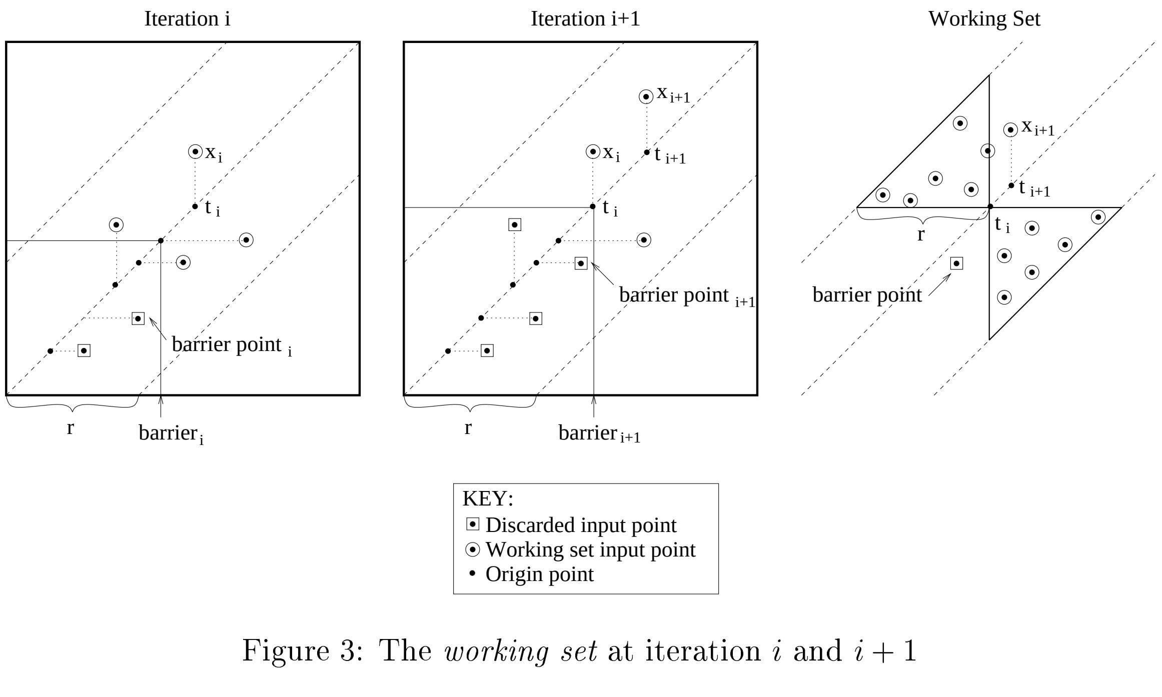


Figure 3: The *working set* at iteration i and $i + 1$

The algorithm computes the height of each $\mathbf{x}(i)$ as the input points are generated. $\mathbf{x}(i)$ is given a default height of 1. Every input point forms a chain of height 1 with itself as the beginning and end points. The new input point is compared with every $\mathbf{x}(j)$ in the *working set* to find the highest chain which can be continued by $\mathbf{x}(i)$. Each $\mathbf{x}(j)$ is also compared with $\mathbf{t}(i)$ to determine if it is below the *barrier*. If $\mathbf{x}(j)$ is below the *barrier*, it either is removed from the working set or (in the case where $h(\mathbf{x}(j)) > h(x_{barrier})$) replaces the current *barrier point*. After iteration i, the *working set* includes only the *barrier point* and the points that are not below the *Barrier* $t(i)$ (see Figure 3). This defines a subset of the $\mathbf{V}_d(r)$ which has a volume $V_d(r)r$. Since n points are uniformly distributed in $\mathbf{V}_d(r)$, it is expected that the *working set* contains rn points. Since the algorithm must iterate over the *working set* for every newly generated point, the expected number of point comparisons is $O(rn^2)$. Therefore, using a very small r not only increases the number of virtual points n^* but decreases the memory requirements and the number of computations by a factor of r.

Proposition 5.2 (Correctness of Algorithm 2) *Suppose $h(\mathbf{x}(i))$ for all $\mathbf{x}(1), \ldots, \mathbf{x}(n)$ is computed according to the algorithm above. Then,*

1. *$h(\mathbf{x}(i))$ reflects the height of the longest chain ending at $\mathbf{x}(i)$.*
2. *h_{max} is equal to the height of the longest chain in $\mathbf{V}_d(r)$.*

PROOF: At the first iteration of the algorithm, $\mathbf{x}(1)$ is assigned the correct height of 1. Each $\mathbf{x}(i+1)$ is compared with all $\mathbf{x}(j), j <= i$ with the exception of the $\mathbf{x}(j) < \mathbf{t}(i)$ which are guaranteed by proposition 5.1 to be bellow $\mathbf{x}(i+1)$. The maximum height of these points is represented by the *barrier point*. Therefore, all possible maximum length chains ending at $\mathbf{x}(i+1)$ are evaluated and $h(\mathbf{x}(i))$ is set to the maximum height. The second statement follows directly from the first by taking the maximum of all $h(\mathbf{x}(i))$. ■

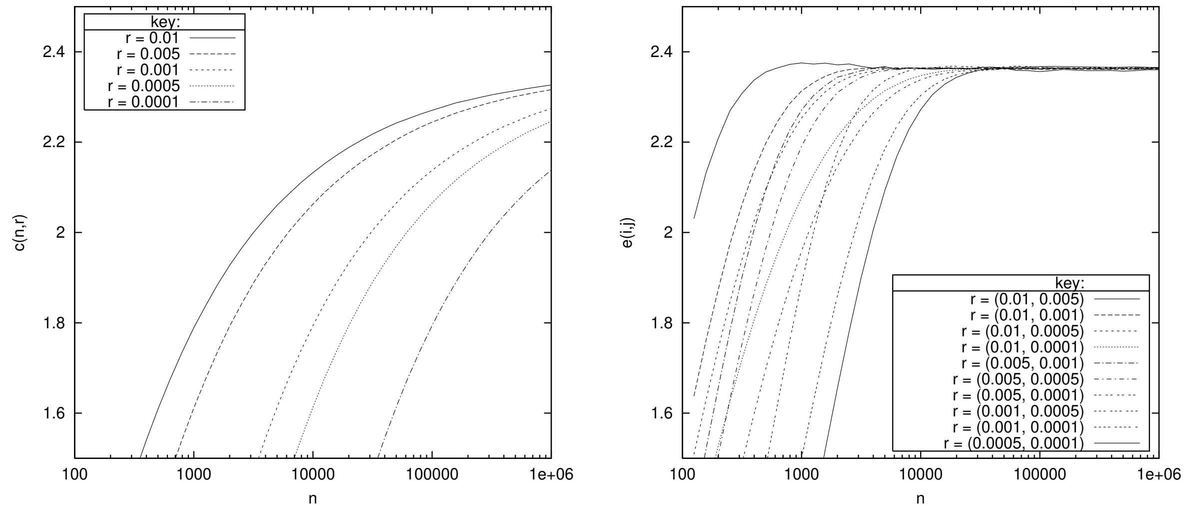
6 Experiments

In this section, we sumarize our results and the calculations used to obtain them. For illustration, we present the results for $d = 3$ here, and the other dimensions, $d = 2, 4, 5$, and 6, can be found in the appendix, section C. We have performed computer simulations to support the following claims:

1. The co-convergence of $\{c_d(n, r)\}$, leading to estimates of $\{c_d\}$.
2. The (asymptotic) independence (and uniqueness) of $r_{opt}(n)$ with respect to n, for $d = 2, 3, 4, 5, 6$.

The co-convergence simulations aim to confirm that for every pair of radii, $r_i \neq r_j$, and sufficiently large n, there exist constants A and B, depending on r_i and r_j, such that

$$\overline{c}(n, r_i) = B + A\overline{c}(n, r_j) + \epsilon. \tag{11}$$



(a) (b)

Figure 4: (a) Convergence of $\overline{c}_3(n,r)$. (b) Convergence of and $c_3(k;i,j)$

The ϵ signifies an error term that is composed of the $o(f(n))$ term, which governs the rate of co-convergence, and the statistical fluctuations in measuring $\bar{c}$. These statistical fluctuations can made arbitrarily small by taking a large enough sample for computing $\bar{c}$. Having obtained A and B, $B/(1-A)$ is an estimate for c_d. We selected n and r according to the following scheme.

$$n_k = \lfloor 10^{2+i_k \times 0.1} \rfloor \text{ for } k \in [0,40], \text{ and } r \in \{0.01, 0.005, 0.001, 0.0005, 0.0001\} \qquad (12)$$

The number of trials used for computing $\overline{c}_d(n,r)$ was heuristically selected to obtain (approximately) the same degree of statistical accuracy (within the given computational requirements) for each n and r. In principle, the larger the number of trials (for any given n and r), the more accurate the estimates will be.

Figure 4(a) demonstrates the convergence behavior of $c_3(n,r)$, for the five values of r. Even for $n = 10^6$, it is not clear to what value these sequences are converging. Given $c_d(n,r_i)$ and $c_d(n,r_j)$ for $n = n_1, \ldots, n_h$, we can estimate $A_d(h,i,j)$ and $B_d(h,i,j)$ for $h = 2, \ldots, L$ as in Experiment 3. The estimate given by $B(h,i,j)/(1-A(h,i,j))$ should converge with respect to h to c_d, independent of which particular pair i,j is used. The behavior of this ratio is demonstrated in figure 4 (b) for $d = 3$. We elaborate on how the estimates $A(h,i,j)$ and $B(h,i,j)$ were obtained in the appendix, section C.1, in addition to showing some sample behavior for the case $d = 3$.

It is clear that these curves are all converging to the same value, supporting our conjecture via theorem 4.1. Further, as expected, the convergence occurs earlier as compared with the convergence of $c_3(n,r)$, in accordance with the expected $o(f(n))$ behavior. Thus, we propose a range for c_3 by taking the range of values to which these curves are converging. Similar figures are shown in the appendix, section C.2, for the cases $d = 2,4,5,6$. We summarize

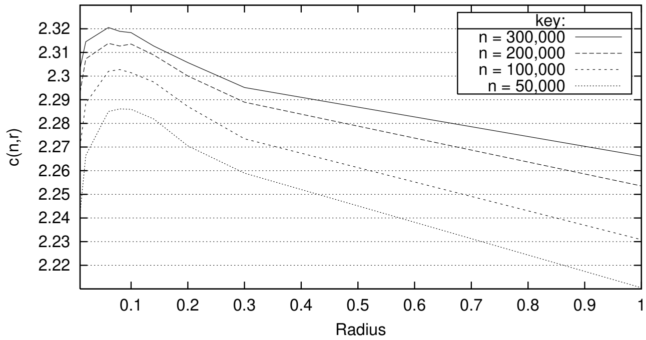


Figure 5: Dependence of $\overline{c}_d(n,r)$ on r for various n's.

the results, in table 2 where we present the estimates for c_d obtained from each of the ten pairs of sequences for each $d = 2, 3, 4, 5, 6$.

(r_j, r_k)	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$
$(0.01, 0.005)$	1.9985	2.3636	2.5144	2.5887	2.6105
$(0.01, 0.001)$	1.9978	2.3627	2.5139	2.5864	2.6145
$(0.01, 0.0005)$	1.9982	2.3629	2.5151	2.5868	2.6153
$(0.01, 0.0001)$	1.9985	2.3628	2.5165	2.5892	2.6156
$(0.005, 0.001)$	1.9999	2.3631	2.5207	2.5867	2.6158
$(0.005, 0.0005)$	2.0001	2.3638	2.5195	2.5866	2.6156
$(0.005, 0.0001)$	2.0006	2.3632	2.5192	2.5866	2.6167
$(0.001, 0.0005)$	2.0010	2.3658	2.5204	2.5827	2.6118
$(0.001, 0.0001)$	2.0015	2.3643	2.5186	2.5837	2.6096
$(0.0005, 0.0001)$	2.0018	2.3663	2.5168	2.5832	2.6074

Table 2: Estimates of c_d obtained from various pairs of sequences for $d = 2, 3, 4, 5, 6$.

As a further piece of evidence for our conjecture, we conducted experiments to investigate whether $r_{opt}(n)$ is independent of n. Our second series of simulations, used values of n and r given in the next table

n	50000	100000	200000	300000					
r	0.01	0.02	0.06	0.08	0.1	0.14	0.20	0.3	1.0

For every combination of n and r, over 350 samples were used to obtain $\bar{c}_d(n,r)$. The results are presented in Figure 5. The figure shows that $r_{opt}(n)$, the radius that maximizes $c_d(n,r)$, appears to be independent of n, and, further, since the curves appear very nearly parallel, the function $\mu(r)$ itself appears to be independent of n, as required by the conjecture.

7 Conclusions

We hope that the observations we have made, based upon our experiments, can be rigorously proved. Our data seems to provide ample support for the conjecture 8, but a rigorous justification is certainly desirable. The translated cube $\mathbf{V}_d$ was chosen on the grounds of simplicity. The analog of the Bollobás & Winkler's theorem can be proved for similar diagonal volume elements with slightly different shapes. It is not clear what the optimal such shape is, in terms of rate of convergence to c_d, with respect to n. The behavior of c_d with respect to d (other than being monotonic and converging to e) does not appear to display a simple enough behavior that would lead to an immediate conjecture for the constant c_d as a function of d. It is hoped that further experiments would provide some guidelines toward a theoretical understanding of the behavior of the maximal chains, and, of the constants c_d.

References

[1] R. M. Baer and P. Brook, *Natural sorting over permutation spaces,* Mat. Comp., **22**, (1968), pp. 385-510.

[2] J. Baik, P. Deift, and K. Johansonn, *On the distribution of the length of the longest increasing subsequence of random permutations,* J. of the Amer. Math. Soc., **12**, No 4, (1999), pp. 1119-1178

[3] P. Billingsley, *Perobability and Measure,* Wiley Series in Probability and Mathematical Statistics, Wiley, 1986.

[4] B. Bollobás and P. Winkler, *The longest chain among random points in eucleadian space,* Proceedings of the Amer. Math. Soc., **103**, No 2, (1988), pp. 347-353.

[5] A. Borodin, *Longest increasing subsequences of random colored permutations,* Electr. J. Comb., **6 (1)**, R13, (1999).

[6] T. Cormen, C. Leiserson, R. Rivest, *Introduction to Algorithms.* The MIT Press, Cambridge, Massachusets; London, England.

[7] P. Erdös and G. Szekeres, *A combinatorial problem in geometry*, Compositio Math., **2**, 1935, pp. 463-470.

[8] J. M. Hammersley, *A few seedlings of research,* Proc. Sixth Berkley Sympo. Math. Stat. Prob., Univ. of California Press, Berkeley, Calif. 1972, pp. 345-394.

[9] B. Logan and L. Shepp, *A variational problem for random Young tableaux,* Adv. in Math., **26**, (1977), pp. 206 - 222.

[10] Catherine McGeoch, (*privite communication*).

[11] A. M. Odlyzko and E.M. Rains, *On longest increasing subsequences in random permutations*, in *Analysis, Geometry, Number Theory: The Mathematics of Leon Ehrenpreis*, E. Grinberg, S. Berhanu, M. Knopp, and G. Mendoza, eds., Amer. Math. Soc., 2000 (to appear).

[12] S. Pilpel, *Descending Subsequences of Random Permutations*, J. of Combinatorial Theory, Series **A**, Vol. 53, No. 1, (1990), pp. 96 - 116.

[13] J. M. Steele, *Limit properies of random variables associated with a partial order in $\Re^d$*, Ann. Probab., **5**, (1977), pp. 395-403.

[14] S. M. Ulam, *Monte Carlo calculations in problems of mathematical physics*, in Modern Mathematics for Engineers, E.F. Beckenbach,

[15] A. M. Verśik and S. V. Kerov, *Asymptotics of the Plancherel measure of the symmetric group and the limiting form of Young tableaux,* Dokl. Acad. Nauk SSSR, **233**, (1977), pp. 1024-1028.

A Proofs of Results

Theorem A.1 *Fix $d \in \{1, 2, \ldots\}$ and $r > 0$, for all $\epsilon > 0$ and $\delta > 0$, there exists $M_r(\epsilon, \delta)$ such that if $n > M_r(\epsilon, \delta)$ then*

$$P\left[|c_d(n,r) - c_d| \leq \epsilon\right] > 1 - \delta \tag{13}$$

where the constants c_d are given in Theorem 1.1.

Proof: An equivalent way to generate n points uniformly in $\mathbf{V}_d(r)$ is to generate n^* points uniformly in $\mathbf{V}_d$ until n of them are in $\mathbf{V}_d(r)$. Thus n^* is a random variable with a distribution given by

$$P[n^* = n + k] = p^n (1-p)^k \binom{n-1+k}{k} \qquad k = 0, 1, \ldots$$

(where we have let $p = \mathbf{V}_d(r)$). The expected value and standard deviation of n^* are then given by $E[n^*] = n/p$ and $std[n^*] = \sqrt{n(1-p)/p^2}$. Given $\epsilon > 0$ and for any $\gamma > 0$, a straight forward application of Markov's inequality gives that

$$P\left[\left(1 - \frac{\epsilon p}{n^{1/2-\gamma}}\right)^{1/d} \leq \left(\frac{n^*}{E[n^*]}\right)^{1/d} \leq \left(1 + \frac{\epsilon p}{n^{1/2-\gamma}}\right)^{1/d}\right] > 1 - \frac{1-p}{(\epsilon p n^\gamma)^2}.$$

Thus, $(n^*/E[n^*])^{1/d}$ converges in probability to 1. First we show that the inequality $c_d(n,r) \leq c_d$ holds in probability as $n \to \infty$. Since $\mathbf{H}_d(n^*) \geq \mathbf{H}_d(n,r)$, we have that

$$c_d(n,r) = \frac{\mathbf{H}_d(n,r)}{(E[n^*])^{1/d}} \leq c_d(n^*) \left(\frac{n^*}{E[n^*]}\right)^{1/d}$$

and since the product of two sequences that converges in probability also converges in probability (to the product of the two limits), we see that the RHS of the inequality converges in probability to c_d.

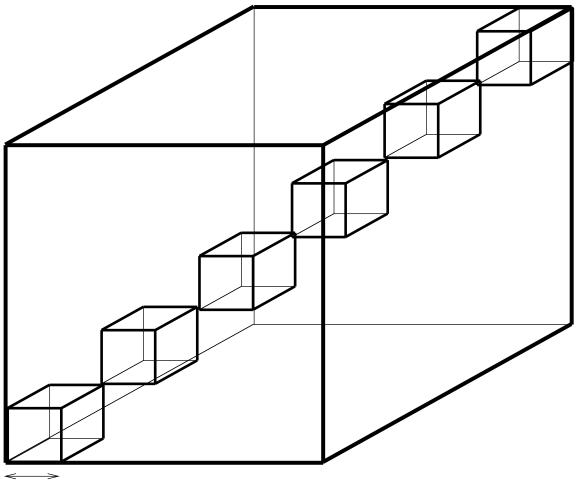


Figure 6: Diagonal sub-cubes of side $1/t < r$

In order to show the reverse inequality, we proceed as in [4] and introduce sub-cubes of side $1/t$ along the diagonal, given by $D_i = [(i-1)/t, i/t]^d$ (see Figure 6). Suppose that t is chosen large enough such that each sub cube is in the region $\mathbf{V}_d(r)$, i.e., $1/t < r$. Chains in these sub-cubes combine to form chains in $\mathbf{V}_d(r)$. Let a maximal chain in the sub-cube i be denoted $\mathbf{H}_d(n^*, i)$. It is then clear that $\mathbf{H}_d(n, r) \geq \sum_i \mathbf{H}_d(n^*, i)$, therefore

$$c_d(n, r) \geq \frac{\sum_{i=1}^{t} \mathbf{H}_d(n^*, i)}{E[n^*]^{1/d}} = \frac{\sum_{i=1}^{t} \mathbf{H}_d(n^*, i)}{(n^*)^{1/d}} \left(\frac{n^*}{E[n^*]} \right)^{1/d}$$

Bollobás and Winkler, [4], show that $\sum_i \mathbf{H}_d(n^*, i)/(n^*)^{1/d}$ converges in probability to c_d hence we conclude that the RHS of the last equality converges in probability to c_d proving that the inequality $c_d(n, r) \geq c_d$ holds in probability as $n \to \infty$ thus concluding the proof that $c_d(n, r)$ converges in probability to c_d. ■

Theorem A.2 *Define a set of sequences $c(n, r)$ for every $r \in (0, 1]$ and suppose that each sequence converges with respect to n to some value c independent of r. Suppose also that $c(n, r)$ is differentiable with respect to r. The following two statements are equivalent.*

1. *$c(n, r) = c - \delta(r) f(n)$ for some functions $\delta(r) > 0$ and $f(n)$ and for all $r \in (0, 1]$.*

2. *$c(n, r_1) = c(1 - A(r_1, r_2)) + A(r_1, r_2) c(n, r2)$ for some differentiable function $A(r, r')$ and for all distinct pairs $r_1, r_2 \in (0, 1]$.*

PROOF:

1. That the first statement implies the second has already been demonstrated in the main text for $A(r_1, r_2) = \delta(r_1)/\delta(r_2)$, and we do not repeat it here.

2. We show that the second implies the first. Suppose that the second statement holds. Then,

$$\begin{aligned} c_d(n, r_1) - c &= A(r_1, r_2)(c_d(n, r_2) - c) \\ &= A(r_1, r_2)Q(n, r_2) \end{aligned} \tag{14}$$

where $Q(n, r_2) = c_d(n, r_2) - c$. The LHS is independent of r_2, so taking the derivative of the log of both sides, we get that for all $r_1, r_2 \in (0, 1]$,

$$\frac{d}{dr_2} \log A(r_1, r_2) = -\frac{d}{dr_2} Q(n, r_2)$$

The RHS is independent of r_1, and thus, so must the LHS. Similarily, the LHS is independent of n, and thus so must the RHS. Hence, they must both be equal to some function $\zeta(r_2)$. Let $\alpha(r_2) = \int \zeta(r_2)$. After an integration, we conclude that for some functions $\kappa(r_1)$ and $g(n)$,

$$\begin{aligned} \log A(r_1, r_2) &= \alpha(r_2) + \log \kappa(r_1) \\ \log Q(n, r_2) &= -\alpha(r_2) + \log g(n) \end{aligned}$$

from which $A(r_1, r_2)Q(n, r_2) = \kappa(r_1)g(n)$. Substituting into (14) we find

$$c(n, r_1) = c + \kappa(r_1)g(n)$$

replacing κ with $-\kappa$ (as κ is arbitrary) concludes the proof. ∎

Theorem A.3 *If the algorithm above is used to generate points* $\mathbf{x}(1)$, $\mathbf{x}(2), \ldots,$ $\mathbf{x}(n)$ *sequentially in* $\mathbf{V}_d(r)$, *then the* $\mathbf{x}(i)$*'s are uniformly distributed in* $\mathbf{V}_d(r)$.

Proof: One can show that the first step in the algorithm generates the order statistics for the distribution of the origin coordinate, [3], where the origin coordinate distribution is uniform from 0 to $(1 - r)$ and then decaying like $(1 - x)^{d-1}$ from $(1 - r)$ to 1 at which point it is 0. The remainder of the algorithm then takes that as origin point and generates a point on one of the $d - 1$ dimensional faces of side r given by $\mathbf{x}_j = 0$. ∎

Theorem A.4 *The volume element* $V_d(r)$ *is given by*

$$V_d(r) = r^d + dr^{d-1}(1 - r).$$

PROOF: $\mathbf{V}_d(r)$ can be divided into disjoint parts with volumes defined as follows:

1. Polygons p_i for $1 \leq i \leq d$ which are defined by a $d-1$ dimensional cube surface of side r parallel to the plane $x_i = 0$ and translated along the diagonal from $(0, 0, \ldots, 0)$ to $(1-r, 1-r, \ldots, 1-r)$. Each polygon p_i has volume $V(p_i)$ given by the integral along the x_i axes
$$V(p_i) = \int_0^{1-r} r^{d-1} dx_i = r^{d-1}(1-r)$$
2. A d-dimesional cube of side r defined by $[1-r, 1]^d$ with volume given by r^d.

Therefore the total volume is given by

$$V_d(r) = \sum_{i=1}^{d} V(p_i) + r^d = r^d + dr^{d-1}(1-r)$$

B Implementation Algorithms

B.1 Generating the Order Statistics for the Origin Points

In this section we provide the details for the generation of the order statistics of the origin points. Suppose we wish to generate the i^{th} order statistic. Let t_{prev} be the coordinate for the $(i-1)^{th}$ order statistic. If $i = 1$ then $t_{prev} = 0$. Define functions $V_d(r)$, $F(l, r, d)$, $F^{-1}(u, r, d)$, and $G^{-1}(u, l, n, k, r, d)$ as follows.

$$V_d(r) = r^d + dr^{d-1}(1-r) \tag{15}$$

$$F(l, r, d) = \begin{cases} \dfrac{dr^{d-1} l}{V_d(r)} & 0 \leq l \leq (1-r) \\ \\ 1 - \dfrac{(1-l)^d}{V_d(r)} & (1-r) \leq l \leq 1 \end{cases} \tag{16}$$

$$F^{-1}(u, r, d) = \begin{cases} \dfrac{V_d(r) u}{dr^{d-1}} & 0 \leq u \leq 1 - \dfrac{r^d}{V_d(r)} \\ \\ 1 - \left[V_d(r)(1-u)\right]^{1/d} & 1 - \dfrac{r^d}{V_d(r)} \leq u \leq 1 \end{cases} \tag{17}$$

$$G^{-1}(u, l, n, k, r, d) = F^{-1}\left(1 - [1 - F(l, r, d)]\left[(1-u)^{1/(n-k+1)}\right], r, d\right) \tag{18}$$

Now, uniformily generate a random number $u \in [0,1]$. Then, the i^{th} origin point, $\mathbf{t}(i) = (t(i), \ldots, t(i))$ is given by

$$t(i) = G^{-1}(u, t_{prev}, n, i, r, d) \tag{19}$$

where n is the total number of points to be generated, r is the size of $V_d(r)$ and d is its dimension. A tedious but an elementary exercise (see [3]) shows that this procedure does indeed generate the correct order statistics, and we do not include that analysis here.

C Experimental Appendix

C.1 Method used to obtain $A_d(h, i, j)$ and $B_d(h, i, j)$

Here we breifly elaborate on step for of the co-convergence experiment, Experiment 4. For a pair of sequences given by r_i, r_j, let

$$\Delta_d(k, i) = \overline{c}_d(n_k, r_i) - \overline{c}_d(n_{k-1}, r_i) \quad \text{and} \quad a_d(k; i, j) = \frac{\Delta_d(k, i)}{\Delta_d(k, j)} \ (k = 1, \ldots, 40). \tag{20}$$

If constants $A_d(i, j)$ and $B_d(i, j)$ exist, then $a_d(k; i, j)$ would approximate $A_d(i, j)$. However, the $o(f(n))$ term and the statistical noise require that we take a more robust approach to get more reliable results. We, therefore, use the following procedure to approximate $A_d(i, j)$ by a sequence $A_d(h, i, j)$ for $h = 2, \ldots, L$.

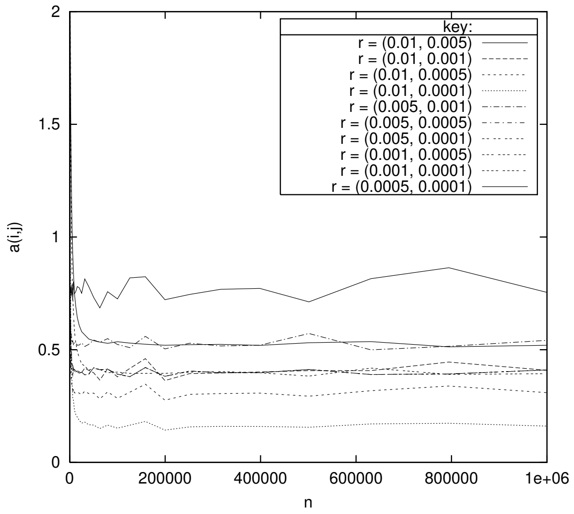
Obtaining $A_d(h, i, j)$: For every $g = 2, \ldots, h$, compute the average $A^{(g)}$ of $\{a_d(k; i, j)\}_{k=g}^{h}$. Compute the standard deviation $\delta^{(h)}$ of $\{a_d(k; i, j)\}_{k=g}^{h}$. Select the smallest $g = g_0$ for which $\delta^{(g)}$ is the minimum to within 0.01. Output $A^{(g_0)}$ as an approximation for $A_d(h, i, j)$.

The approach is designed to accomodate the statistical noise and the fact that the convergence of the $a(k, i, j)$'s might not have occurred at the begining of the sequence.

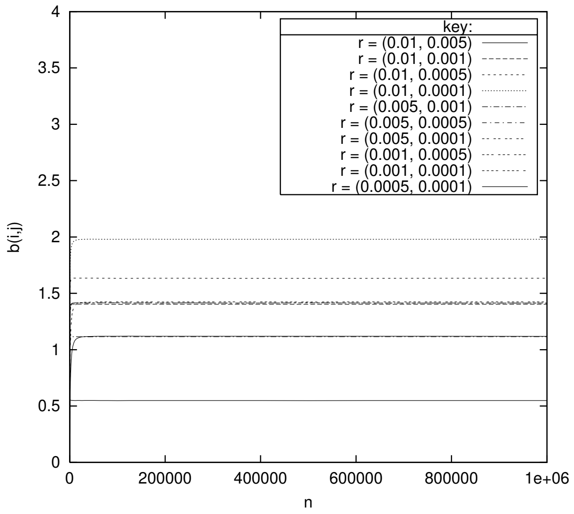
The procedure for approximating $B_d(i, j)$ is similar to the one for $A_d(i, j)$. and is as follows.

Obtaining $B_d(h, i, j)$: Compute $b_d(k; i, j) = c_d(n_k, r_i) - A_d(k, i, j)c_d(n_k, r_j)$. For every $g = 2, \ldots, h$, compute the average $B^{(g)}$ of $\{b_d(k; i, j)\}_{k=g}^{h}$. Compute the standard deviation $\delta^{(h)}$ of $\{b_d(k; i, j)\}_{k=g}^{h}$. Select the smallest $g = g_0$ for which $\delta^{(g)}$ is the minimum to within 0.01. Output $B^{(g_0)}$ as an approximation for $B_d(h, i, j)$.

We demonstrate the behavior of $A(h, i, j)$ and $B(h, i, j)$ for $d = 3$ in figure 7. As can be seen, the convergence occurs quite soon. The cases $d = 2, 4, 5, 6$ are similar and do not convey additional insight hence we do not show them here.



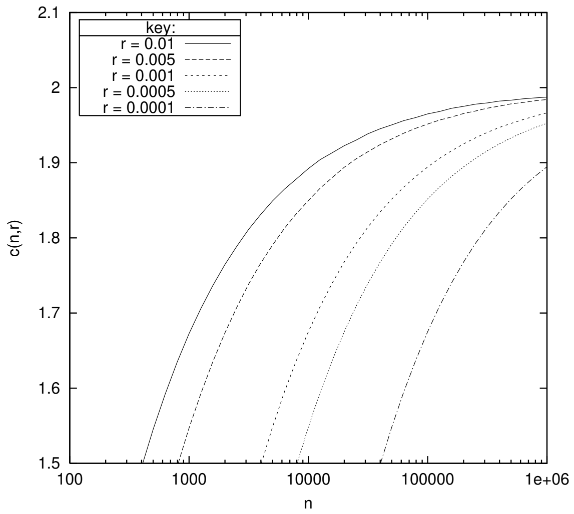
(a)



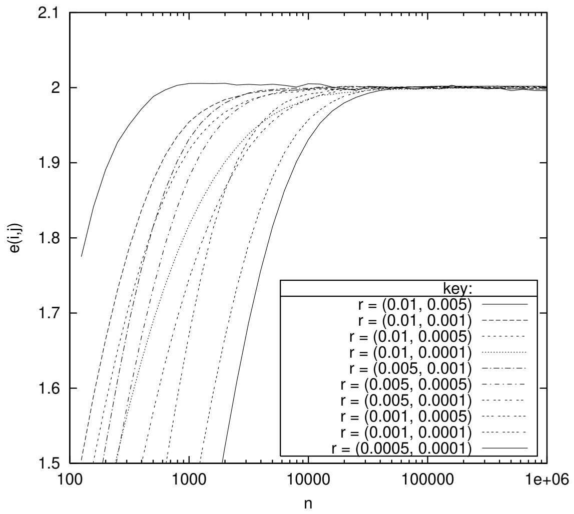
(b)

Figure 7: (a) Convergence of the estimates for $A_d(k; i, j)$ for $d = 3$. (b) Convergence of the estimates $B_d(k; i, j)$ for $d = 3$.

C.2 Additional Plots for $d = 2, 4, 5, 6$

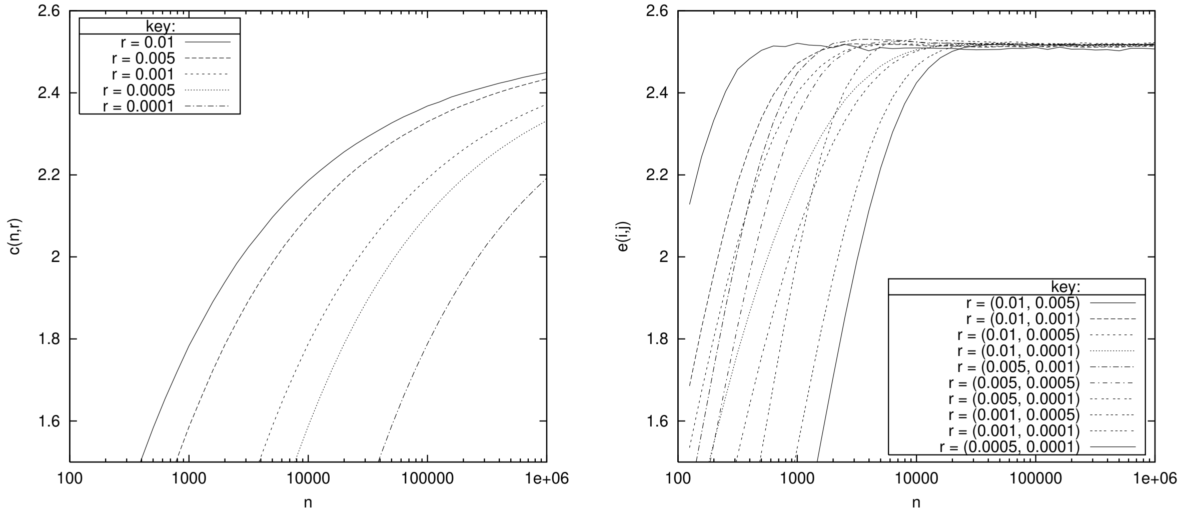


(a)



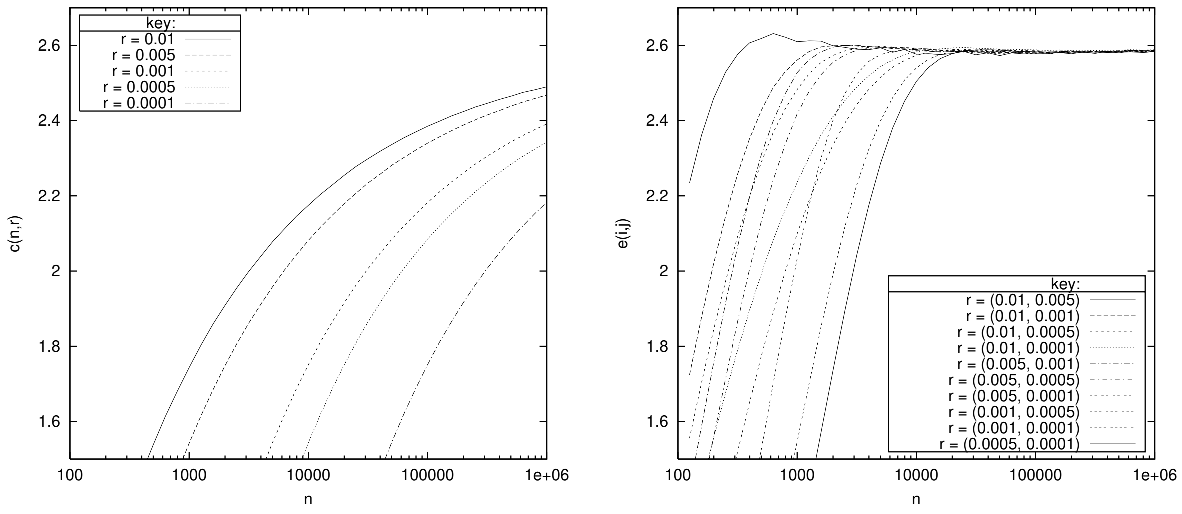
(b)

Figure 8: Convergence of $\overline{c}_d(n, r)$ and $c_d(k; i, j)$ for $d = 2$.



(a) (b)

Figure 9: Convergence of $\overline{c}_d(n, r)$ and $c_d(k; i, j)$ for $d = 4$.



(a) (b)

Figure 10: Convergence of $\overline{c}_d(n, r)$ and $c_d(k; i, j)$ for $d = 5$.

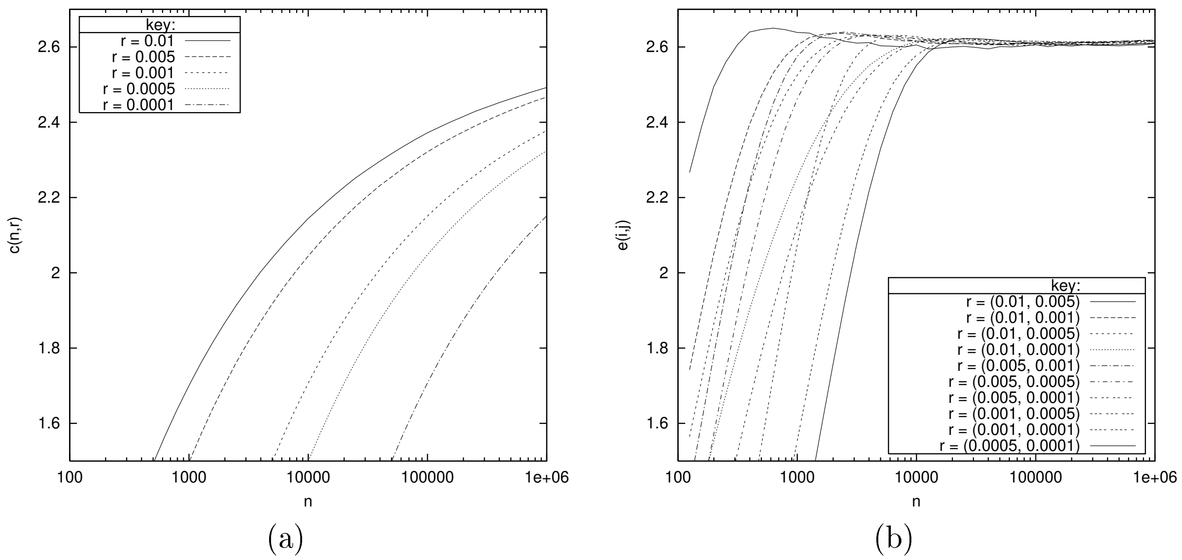


Figure 11: Convergence of $\overline{c}_d(n, r)$ and $c_d(k; i, j)$ for $d = 6$.