Two-sided convexity testing with certificates *

Adrian Dumitrescu†

May 9, 2023

Abstract

We revisit the problem of property testing for convex position for point sets in $\mathbb{R}^d$. Our results draw from previous ideas of Czumaj, Sohler, and Ziegler (ESA 2000). First, the algorithm is redesigned and its analysis is revised for correctness. Second, its functionality is expanded by (i) exhibiting both negative and positive certificates along with the convexity determination, and (ii) significantly extending the input range for moderate and higher dimensions.

The behavior of the randomized tester is as follows: (i) if P is in convex position, it accepts; (ii) if P is far from convex position, with probability at least 2/3, it rejects and outputs a $(d+2)$-point witness of non-convexity as a negative certificate; (iii) if P is close to convex position, with probability at least 2/3, it accepts and outputs an approximation of the largest subset in convex position. The algorithm examines a sublinear number of points and runs in subquadratic time for every fixed dimension d.

Keywords: property testing, convex position, approximation algorithm, randomized algorithm.

1 Introduction

A set of points in the d-dimensional space $\mathbb{R}^d$ is said to be: (i) in *general position* if any at most $d+1$ points are *affinely independent*; and (ii) in *convex position* if none of the points lies in the convex hull of the other points. It is known that every set of n points in general position in the plane contains $(1-o(1))\log n$ points in convex position, and this bound is tight up to lower-order terms [12, 26]. For $d \geq 3$, by the Erdős–Szekeres theorem, every set of n points in general position in $\mathbb{R}^d$ contains $\Omega(\log n)$ points in convex position: it suffices to find points whose projections onto a generic plane are in convex position. On the other hand, for every fixed $d \geq 2$, Károlyi and Valtr [17] and Valtr [27] constructed n-element sets in general position in $\mathbb{R}^d$ in which no more than $O(\log^{d-1} n)$ points are in convex position. A recent result of Pohoata and Zakharov [23] shows that a set of n points in $\mathbb{R}^d$, $d \geq 3$, already contains a subset of $\omega(\log n)$ points in convex position.

Given a point set in general position in $\mathbb{R}^d$, the problem of computing a maximum-size subset in convex position can be solved in polynomial time for $d = 2$ by the dynamic programming algorithm of Chvátal and Klincsek [6]; their algorithm runs in $O(n^3)$ time. In contrast, the general problem in $\mathbb{R}^d$ was shown to be NP-complete for every $d \geq 3$ by Giannopoulos, Knauer, and Werner [14], and moreover, no approximation algorithm is known.

*A preliminary version appears in Proceedings of the 12th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications, March 2023, Budapest, Hungary. The current expanded version corrects some inaccuracies present there.

†Algoresearch L.L.C., Milwaukee, WI, USA. Email ad.dumitrescu@algoresearch.org.

Throughout this paper we assume (in a standard fashion) that the input set is in *general position*. For Theorems 1 and 2 and Corollary 1, let P be a set of n points in $\mathbb{R}^d$, where d is considered constant.

The complexity of computing the convex hull of n points in $\mathbb{R}^d$ is summarized in the following result of Chazelle; see also [1, 25].

Theorem 1. (Chazelle [5]) *Given P, the convex hull of P can be computed in $O(n \log n + n^{\lfloor d/2 \rfloor})$ time using $O(n^{\lfloor d/2 \rfloor})$ space, which is asymptotically worst-case optimal.*

It is known that the number of faces, f, of the output polytope is $\Theta(n^{\lfloor d/2 \rfloor})$ in the worst case [21], i.e., exponential in d. On the other hand, a result of Chan shows that the set of extreme points of a set of n points in $\mathbb{R}^d$ can be computed in subquadratic time and essentially faster when their number h is small.

Theorem 2. (Chan [4]) *Given P, the h extreme points of P can be computed in time*

$$T(n,h) = O\left(n \log^{O(1)} h + (nh)^{\frac{\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \log^{O(1)} n\right). \tag{1}$$

Taking $n = h$ in the above expression yields a time that suffices for testing whether a set of n points is in convex position. From the other direction, it is conjectured that the problem of testing whether a set P is in convex position is asymptotically as hard as the problem of computing all extreme points of P [7].

Corollary 1. (Chan [4]) *Given P, determining whether P is in convex position can be done in time $T(n,n) = O\left(n^{\frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \log^{O(1)} n\right)$.*

For instance, the running time in Corollary 1 is $O\left(n \log^{O(1)} n\right)$ for $d = 2, 3$, $O\left(n^{4/3} \log^{O(1)} n\right)$ for $d = 4, 5$, $O\left(n^{3/2} \log^{O(1)} n\right)$ for $d = 6, 7$, and subquadratic in any fixed dimension d.

Similarly, the following holds (see Corollary 3.4 in [4]).

Corollary 2. (Chan [4]) *Given P and $S \subset P$, where $s = |S|$, determining whether all points in S are extreme in P can be done in time $T(n,s) = O\left(n \log^{O(1)} s + (ns)^{\frac{\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \log^{O(1)} n\right)$.*

In *property testing* one is concerned with the design of faster algorithms for approximate decision making [15]. In this scenario, instead of determining whether an input has a specific property, one determines if the input is *far* or perhaps *close* from satisfying that property. Such approximate decisions, usually involving random sampling or shortcuts in the computation, may be valuable in settings in which an exact decision is infeasible or just more expensive. For example, one may be interested in determining, given an input point set, how far it stands from being in convex position without needing to spend all resources that would be required for computing the convex hull of the respective set. Such a tool is obviously useful in the general area of testing properties of geometric objects and visual images for distinguishing a convex shape among other shapes.

The goal of *property testing* is to develop efficient *property testers*. Ideally, such a tester makes a sublinear number of queries of the input set, i.e., it does not look at the entire input. However, this does not mean — even for the ideal case — that the tester runs in time that is sublinear in the size of the input; in fact, it often doesn't. Moreover, if the tester is also required to return a possibly large subset of the input set (depending on the outcome) as a certificate, then its time requirements may be further increased.

Here we focus on the testing of *convex position*. As in the context of randomized algorithms, approximately deciding means returning the correct answer with some confidence, specifically with probability at least 2/3 as described below, see, e.g., [20]; however, the 2/3 threshold is not set in stone.

Testing algorithms may use samples of different sizes. Some intuition is as follows. Suppose that the input is far from convex position; the algorithm is likely to reject on large samples (the larger the sample, the easier it will be to find that out), and is likely to accept on small samples (the smaller the sample, the easier the algorithm will be fooled). On the other hand, if the input is close to convex position, the smaller the sample, the easier it will be for the algorithm to accept.

A key distinction with regard to the action (accept or reject) is that closeness must fit the goal, i.e., far and close need to be quantified appropriately. As it turns out, rejecting an input that is far from convex position is relatively insensitive to the distance from convex position. However, when accepting an input that is close to convex position, the input must be really close.

1.1 Preliminaries

Definitions and notation. Let $0 < \varepsilon < 1/2$. A set P of n points is ε-*far* from convex position if there is no set $X \subset P$ of size at most εn such that $P \setminus X$ is in convex position. Otherwise, i.e., if there is a set $X \subset P$ of size at most εn such that $P \setminus X$ is in convex position, P is ε-*close* to convex position. See Fig. 1. For a set point P, let $\texttt{Ext}(P)$ denote the set of extreme points of P.

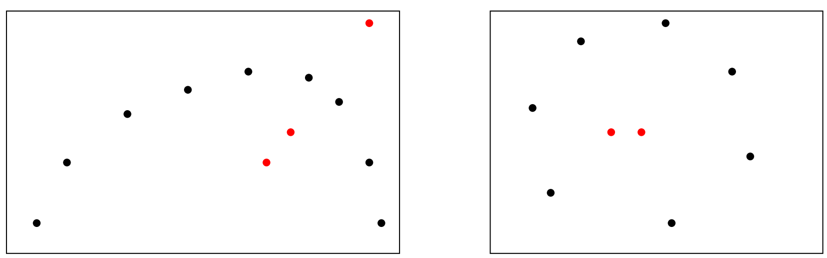

Figure 1: A 12-point set that is 1/4-close to convex position (left), and a 9-point set that is 2/9-close to convex position (right). Both sets are 1/5-far from convex position.

Here we use the convention that the approximation ratio of an algorithm is smaller than 1 for a maximization problem and larger than 1 for a minimization problem; see [28]. Unless specified otherwise, all logarithms are in base 2. For a set $W \subset \mathbb{R}^d$, its *interior* is denoted by $\mathring{W}$.

Nonconvexity certificates. By the well-known Carathéodory's Theorem, see, e.g., [19, p. 6], if X is finite point set in $\mathbb{R}^d$, every point of X can be expressed as a convex combination of at most $d+1$ points in X. This implies that every point set that is not in convex position contains a subset of $d+2$ points that are not in convex position, i.e., a short *certificate* of non-convexity. We will further assume that Chan's algorithm for testing of convex position outputs such a tuple when the input is not in convex position.

The convex position tester of Czumaj, Sohler, and Ziegler. The convex position tester of Czumaj et al. [7] draws a random sample of the input set and makes a decision based on the convexity of this sample. The algorithm is set up to work in $\mathbb{R}^d$, for any fixed dimension d. Given $\varepsilon > 0$, the tester accepts every point set in convex position, and rejects every point set that is ε-far from convex position with probability at least 2/3. If the input is not in convex position and is not ε-far from convex position, the outcome of the algorithm can go either way, i.e., there is no specified action for the situation in-between. Most of the technical justification is unpublished; for the present time, it can be found online [8]. The authors present two testers for convex position: `Convex-A` and `Convex-B`, see [7, p. 161]:

Algorithm `Convex-A`

Step 1: Choose a subset $S \subset P$ of size $s = 36 \cdot n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}$ uniformly at random.

Step 2: Compute all h extreme points of S.

Step 3: If $h < n$ then reject else accept.

Algorithm `Convex-B`

Step 1: Choose a subset $S \subset P$ of size $s = 4/\varepsilon$ uniformly at random.

Step 2: For each $p \in S$ [simultaneously] check whether p is extreme for $\mathrm{conv}(P)$. If p is not extreme for $\mathrm{conv}(P)$ then exit loop and reject.

Step 3: If all checks complete, accept.

The *query complexity*, i.e., the number of points requested from an *oracle* to perform the testing, is $O(n^{d/(d+1)}\varepsilon^{-1/(d+1)})$, which is claimed by the authors to be optimal (no proof is provided) [7]. The corresponding running time follows from Corollary 1 and is subquadratic in any fixed dimension d.

The correctness proof for `Convex-A` is only sketched in [7]. It is however similar in nature to the revised argument we give here based on Lemmata 2, 3 and 4. The correctness proof for `Convex-B`, also omitted in [7], is implied from the following.

Lemma 1. *Let $P \subset \mathbb{R}^d$ be ε-far from convex position. Then $|P \setminus \mathtt{Ext}(P)| > \varepsilon|P|$.*

Proof. Assume for contradiction that $|P \setminus \mathtt{Ext}(P)| \leq \varepsilon|P|$. Removing all points in $P \setminus \mathtt{Ext}(P)$ yields a convex set and thus P is ε-close to convex position, a contradiction. □

In fact the sample size in `Convex-B` can be reduced in half; i.e., one can set $s = 2/\varepsilon$, see below. If the input $P \subset \mathbb{R}^d$ is ε-far from convex position, then the set $Q = P \setminus \mathtt{Ext}(P)$ is large enough and the tester would reject P if at least one sample point is in Q. Since $|Q| \geq \varepsilon|P|$, we have

$$\mathrm{Prob}(S \cap Q = \emptyset) \leq (1-\varepsilon)^{2/\varepsilon} \leq \mathrm{e}^{-2} \leq \frac{1}{3},$$

by applying the standard inequality $1 - x \leq \mathrm{e}^{-x}$ for $0 \leq x \leq 1/2$. Thus P is rejected with probability at least 2/3, as required. Note that an input in convex position is accepted by either tester. In summary, by Corollary 1 and Corollary 2, negative testing (via `Convex-A` or `Convex-B`) can be accomplished in time

$$O\left(\min\left\{T\left(n^{\frac{d}{d+1}}\varepsilon^{-\frac{1}{d+1}}, n^{\frac{d}{d+1}}\varepsilon^{-\frac{1}{d+1}}\right), T\left(n, \varepsilon^{-1}\right)\right\}\right). \tag{2}$$

Unfortunately, the convex position tester of Czumaj et al. [7] suffers from structural and performance issues as explained below. One issue is an unreasonable dependence of the tester `Convex-A` of the input parameter ε; a second concerns a technical lemma that needs correction. Moreover, as mentioned earlier, most of the claims made in [7] are unverifiable since most proofs are omitted. Here we fix these problems and obtain a more performant negative tester. Further, its functionality is expanded by including positive certificates. Our paper is self-contained with all needed proofs included.

(i) The sample size used by tester `Convex-A` is

$$s = 36 \cdot n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}.$$

Since $s \leq n$ is a prerequisite for using the tester, this imposes the restriction $36^{d+1} \leq \varepsilon n$; equivalently, $\varepsilon \geq 36^{d+1}/n$. Since $\varepsilon < 1$, this implies $n > 36^{d+1}$. This requirement makes the tester impractical even for moderate values of d. For instance, if $d = 20$, tester `Convex-A` can only test sets with $n > 4.8 \cdot 10^{32}$ points. Similarly, if $d = 50$, tester `Convex-A` can only test sets with $n > 2.3 \cdot 10^{79}$ points, which is approximately the number of atoms in the observable universe. Arguably, such applications, if any, are rare. As such, the tester isn't functional in the range $d \geq 50$. In contrast, our Algorithm `Convex-` in Subsection 2.1 is only subject to the very modest restriction $\varepsilon \geq (d+1)/n$. Similarly, our Algorithm `Convex+` in Subsection 2.2 is subject to very modest restrictions.

(ii) Another issue is the correctness of Lemma 3.4 in [8], discussed in Section A. Our Lemma 4 is proposed as a replacement.

Our results. We revisit the problem of property testing for convex position for point sets in $\mathbb{R}^d$. Our results draw from previous design and ideas of Czumaj, Sohler, and Ziegler (ESA 2000). First, the algorithm is redesigned and its analysis is revised for correctness. Second, its functionality is expanded by (i) exhibiting both negative and positive certificates along with the convexity determination, and (ii) significantly extending the input range for moderate and higher dimensions. The tester is implemented by two procedures: `Convex-` and `Convex+`. Both run in $O\left(n^{\frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \log^{O(1)} n\right) = o(n^2)$ time, for every n and ε.

The behavior of Algorithm `Convex-` can be summarized as follows. Let $0 < \varepsilon < 1$ be an input parameter.

1. If P is in convex position, the algorithm accepts P.
2. If P is ε-far from convex position, with probability at least $2/3$ the algorithm rejects P and outputs a $(d+2)$-point witness of non-convexity (as a negative certificate).

The behavior of Algorithm `Convex+` can be summarized as follows. Let $0 < \varepsilon < 1$ be an input parameter, and $0 < \delta \leq 1/2$ be an adjustable parameter.

1. If P is in convex position, the algorithm accepts P.
2. If P is ε-close to convex position for some $\varepsilon > 0$ that satisfies $n^{-1} \leq \varepsilon \leq n^{\delta - 1}$, with probability at least $2/3$ the algorithm accepts P and outputs a $1/(6n^{\delta})$-approximation of the largest subset in convex position as a positive certificate.

Related work. Two early articles in the area of property testing are due to Blum et al. [3] and Ergün et al. [13]. Besides testing for convex position, testing for other geometric properties has been considered in [7]: pairwise disjointness of a set of generic bodies, disjointness of two polytopes, and Euclidean minimum spanning tree verification. A continuation of the work in [7] appears in [9]. A more recent article on property testing for point sets in the plane is due to Han et al. [16]. Two recent monographs dedicated to the general subject of property testing are [2] and [15]. The topic of property testing, including testing for convex position, is also addressed in a recent book by Eppstein [11]. A question from that book is discussed in Section 3.

2 An enhanced functionality tester for convex position

The tester is implemented by two procedures: Algorithm `Convex-` (in Subsection 2.1) and Algorithm `Convex+` (in Subsection 2.2). The two procedures may be run independently of each other. The goal of Algorithm `Convex-` is rejecting point sets that are far from convex position; whereas that of Algorithm `Convex+` is accepting point sets that are close to convex position. Each algorithm exhibits a suitable certificate along with its probabilistic determination. While the decision is randomized, the certificates produced are indisputable, i.e., a negative certificate is always a $(d+2)$-point set that is not in convex position, and a positive certificate output by Algorithm `Convex+` is always a $1/(6n^\delta)$-approximation of the largest subset in convex position.

Common tools. A randomized algorithm for generating a random s-set for a given s, $1 \leq s \leq n$, in $O(s \log s)$ time (and $O(s)$ expected time) from [22, Ch. 4], can be used to implement random sample selection. Alternatively, a linear-time algorithm for the same task from [24, Sec 5.2] can also be used.

2.1 Negative testing: Algorithm `Convex-`

Several constraints among the input parameters need to be respected usually for technical reasons. In particular, it is assumed that (note that these constraints are very mild):

- $n \geq 2^{10}$, this is needed in the proof of Lemma 4.
- $n \geq 32(d+1)$, this ensures that $\ell \leq n/32$ when using Lemma 4.
- $\varepsilon \geq \frac{10(d+1)}{n}$, this ensures that $k \geq 10$ in Step 1; compare this to the constraint $\varepsilon \geq 36^{d+1}/n$ in tester `Convex-A` that restricts its use to low dimensions.
- $\varepsilon \leq \frac{d-1}{2d}$, this ensures $\frac{(1-\varepsilon)}{d+1} \geq \frac{1}{2d}$ in the analysis.

Algorithm `Convex-`

Step 1: Let $k = \lfloor \frac{\varepsilon n}{d+1} \rfloor$, $\ell = d+1$, $s_0 = \ell + \frac{n-\ell}{(2k)^{1/\ell}}$, and $s = \lceil s_0 \rceil$. Repeat Step 2 and Step 3 in succession up to 22 times.

Step 2: Randomly select a subset $S \subset P$ of size s, with all s-subsets being equally likely.

Step 3: Test S for convex position using Chan's algorithm. If S is not in convex position, output a $(d+2)$-point witness of non-convexity and reject P. Otherwise go to Step 2 for the next repetition.

Step 4: If all 22 samples were determined to be in convex position, accept P.

Time analysis. It is easily verified that the setting for s in Step 1 yields

$$s = \Theta\left(n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}\right).$$

This is in accordance with the choice of the sample size for Algorithm `Convex-A` in [7]. As such, the runtime of Algorithm `Convex-` is

$$\begin{aligned} T(s,s) &= O\left(T\left(n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}, n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}\right)\right) \\ &= O\left(n^{\frac{d}{d+1} \cdot \frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \cdot \varepsilon^{-\frac{1}{d+1} \cdot \frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \cdot \log^{O(1)}(n/\varepsilon)\right). \end{aligned}$$

Since $\varepsilon = \Omega(1/n)$, the above expression becomes

$$T(s,s) = O\left(T(n,n)\right) = O\left(n^{\frac{2\lfloor d/2\rfloor}{\lfloor d/2\rfloor+1}} \log^{O(1)} n\right) = o(n^2), \text{ for every } n \text{ and } \varepsilon.$$

This can be also seen directly: since $s \le n$, $T(s,s) \le T(n,n) = o(n^2)$.

Rejecting the input with probability $\geq 2/3$. Assume that P is ε-far from convex position. We show that with probability at least 2/3, Algorithm `Convex-` rejects the input in step 3 and outputs a suitable $(d+2)$-point witness. We first recall the following lemmas (analogous to Lemma 3.1 and 3.2 from [8]), slightly rewritten here for convenience.

Lemma 2. (An earlier version in [8]). *Let $P \subset \mathbb{R}^d$ be a set of n points that is not in convex position and $p \in P$ be an interior point. Then there exist points $p_1, \ldots, p_d \in P$ and $U \subset P \setminus \{p_1, \ldots, p_d, p\}$ with $|U| \geq \frac{n-1}{d+1}$ such that $\{p_1, \ldots, p_d, p\} \cup \{q\}$ is not in convex position for every $q \in U$; more precisely, p is an interior point in the simplex $\Delta(p_1, \ldots, p_d, q)$ for every $q \in U$.*

Proof. Since $p \in P$ is an interior point, by Carathéodory's Theorem and by the general position assumption, there exists a set $W \subset P$ of size $d+1$ such that $p \in \mathring{W}$. See Fig. 2.

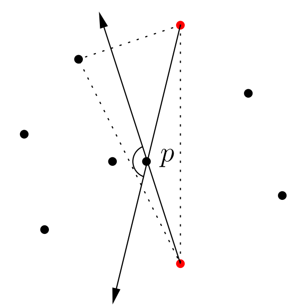

Figure 2: P is a set of 9 points in the plane. The cone determined by the two red points contains $4 \geq 8/3$ points in P.

Denote by W_i, $i = 1, \ldots, d+1$, the $d+1$ subsets of W of size d. We show that one of the subsets W_i of W satisfies the requirement in the lemma. We may assume without loss of generality that $p = (0, \ldots, 0)$. We partition $\mathbb{R}^d$ into $d+1$ cones as follows. Let W_i^-, $i = 1, \ldots, d+1$, denote the set of points $\{(-x_1, \ldots, -x_d) \colon (x_1, \ldots, x_d) \in W_i\}$. The conic combination of the point vectors in the set W_i^- defines a cone C_i, $i = 1, \ldots, d+1$. The union of these cones cover $\mathbb{R}^d$. Thus there is a cone C_j, $1 \le j \le d+1$, that contains at least $\frac{n-1}{d+1}$ points in P. Observe that for every $q \in P \cap C_j$ we have $p \in (W_j \mathring{\cup} \{q\})$. Consequently, one can set $\{p_1, \ldots, p_d\} = W_j$ to conclude the proof. $\square$

The following lemma applies to point sets that are far from convex position. The sets W_i and U_i constructed in the lemma are fixed before the samplings and are only used in the algorithm analysis.

Lemma 3. (An earlier version in [8]). *Let $P \subset \mathbb{R}^d$ be a set of n points that is ε-far from convex position and let $k = \lfloor \frac{\varepsilon n}{d+1} \rfloor$. Then there exist sets $W_i, U_i \subset P$ for $1 \le i \le k$, such that the following conditions are satisfied:*

(i) $|W_i| = d+1$ *for* $1 \le i \le k$,

(ii) $W_i \cap W_j = \emptyset$ *for all* $1 \le i < j \le k$,

(iii) $W_i \cap U_i = \emptyset$ *for* $1 \le i \le k$,

(iv) $W_i \cup \{q\}$ *is not in convex position for every* $q \in U_i$, *and*

(v) $|U_i| \ge \frac{n}{d+1} - k$ *for* $1 \le i \le k$. *In particular,* $|U_i| \ge \frac{(1-\varepsilon)n}{d+1}$.

Proof. We construct point sets $P_1, P_2, \dots, P_k$ iteratively. We initially set $P_1 := P$ and then iteratively find $W_i \subset P_i$ and set $P_{i+1} := P_i \setminus W_i$ for $i = 1, \dots, k$. By construction the sets W_i are pairwise disjoint, as required. Assuming that $|W_i| = d+1$ for $1 \le i \le k$, implies that

$$|P_i| = n - (d+1)(i-1) \ge n - (d+1)(k-1) > n - (d+1)\frac{\varepsilon n}{d+1} = (1-\varepsilon)n.$$

By the assumption in the lemma, P_i cannot be in convex position. By Lemma 2 there exist $p_1, \dots, p_d, p \in P_i$ and $U_i \subset P_i \setminus \{p_1, \dots, p_d, p\}$ with

$$\begin{aligned}|U_i| &\ge \frac{|P_i| - 1}{d+1} \ge \frac{n - (d+1)(i-1) - 1}{d+1} \ge \frac{n - (d+1)(k-1) - 1}{d+1} \\ &> \frac{n}{d+1} - k = \frac{n}{d+1} - \frac{\varepsilon n}{d+1} = \frac{(1-\varepsilon)n}{d+1},\end{aligned}$$

such that p is an interior point in the simplex $\Delta p_1, \dots, p_d, q$ for every $q \in U_i$. Let $W_i := \{p_1, \dots, p_d, p\}$ and observe that $W_i \cap U_i = \emptyset$. Note that all properties in the lemma have been verified. □

We also need another lemma suggested by Czumaj et al. [8]. Here we include a proof that follows the ideas of the original proof, however, it is revised for correctness and for a slightly restricted range of the parameters that suffices for our purposes. More details can be found in Section A.

Lemma 4. (An earlier version in [8]). *Let* Ω *be a set of size* n *and* $W_1, W_2, \dots, W_k \subset \Omega$ *be* k *pairwise disjoint subsets of* Ω *of size* ℓ, *where* $k \ge 10$ *and* $3 \le \ell \le n/32$. *Let* s *be a positive integer such that* $\ell + \frac{n-\ell}{(2k)^{1/\ell}} \le s \le n$ *and* $S \subset \Omega$ *be a subset of* Ω *of size* s *chosen uniformly at random. Then*

$$\mathrm{Prob}(\exists i \le k \colon (W_i \subset S)) \ge \frac{1}{4}.$$

Proof. Observe that $k\ell \le n$, hence $k \le n/\ell$. Let s_0 be the real number defined as follows:

$$s_0 = \ell + \frac{n-\ell}{(2k)^{1/\ell}}, \text{ or } k\left(\frac{s_0 - \ell}{n-\ell}\right)^{\ell} = \frac{1}{2}, \tag{3}$$

and note that $\ell < s_0 < n$. Indeed, the lower bound is clear and the upper bound $s_0 < n$ is equivalent to $(2k)^{1/\ell} > 1$ which is obvious. We first prove that

$$s_0 \ge 3\ell \log k. \tag{4}$$

It suffices to show that $n - \ell \ge 3\ell(2k)^{1/\ell} \log k$, or, since $\ell \le n/32$, that $3\ell(2k)^{1/\ell} \log k \le \frac{31n}{32}$. We have

$$3\ell(2k)^{1/\ell} \log k \le 3\ell \left(\frac{2n}{\ell}\right)^{1/\ell} \log\left(\frac{2n}{\ell}\right) \le \frac{31n}{32}.$$

Indeed, a standard verification shows that the function

$$f(x) = 3x\left(\frac{2n}{x}\right)^{1/x} \log\left(\frac{2n}{x}\right), x \in \left[3, \frac{n}{32}\right],$$

where $n \geq 2^{10}$, attains it maximum at $x = n/32$, thus

$$f(x) \leq f\left(\frac{n}{32}\right) = 3 \cdot \frac{n}{32} \cdot \left(\frac{2n}{n/32}\right)^{32/n} \log\left(\frac{2n}{n/32}\right)$$
$$= \frac{3n}{32} \cdot 64^{32/n} \cdot \log 64 \leq \frac{18n}{32} \cdot \frac{5}{4} \leq \frac{31n}{32}.$$

This concludes the proof of (4) and we next focus on the inequality in the lemma.

Since the probability in question increases as the sample size s grows, it suffices to prove the inequality for $s = \lceil s_0 \rceil$. Observe that $\ell + 1 \leq s \leq n$. By the Boole-Bonferoni inequality—see, e.g., [18, Ch. 2], we have

$$\text{Prob}(\exists i \leq k\colon (W_i \subset S)) \geq \sum_{i=1}^{k} \text{Prob}(W_i \subset S) - \sum_{1 \leq i < j \leq k} \text{Prob}((W_i \cup W_j) \subset S). \tag{5}$$

It is easily verified that

$$\text{Prob}(W_i \subset S) = \frac{\binom{n-\ell}{s-\ell}}{\binom{n}{s}} = \frac{(n-\ell)!}{(s-\ell)!(n-s)!} \cdot \frac{s!(n-s)!}{n!}$$
$$= \frac{(n-\ell)!s!}{n!(s-\ell)!} = \prod_{r=0}^{\ell-1} \frac{s-r}{n-r}, \text{ and}$$
$$\text{Prob}((W_i \cup W_j) \subset S) = \frac{\binom{n-2\ell}{s-2\ell}}{\binom{n}{s}} = \prod_{r=0}^{2\ell-1} \frac{s-r}{n-r}$$
$$= \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r}, \text{ for } 1 \leq i < j \leq k.$$

Substituting these into Inequality (5) and finally using (3) yields

$$\text{Prob}(\exists i \leq k\colon (W_i \subset S)) \geq k \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} - \binom{k}{2} \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r}$$
$$= k \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \left(1 - \frac{k-1}{2} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r}\right)$$
$$\geq k \cdot \prod_{r=0}^{\ell-1} \frac{s-\ell}{n-\ell} \cdot \left(1 - \frac{k}{2} \cdot \prod_{r=0}^{\ell-1} \frac{s-\ell}{n-\ell}\right)$$
$$= k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} \cdot \left(1 - \frac{k}{2} \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell}\right).$$

Let

$$F_1 = k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} \text{ and } F_2 = 1 - \frac{k}{2} \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell}.$$

It suffices to show that $F_1 \geq \frac{1}{2}$ and $F_2 \geq \frac{1}{2}$. For the first inequality, we have

$$F_1 = k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} \geq k \cdot \left(\frac{s_0-\ell}{n-\ell}\right)^{\ell} = \frac{1}{2}. \tag{6}$$

For the second, recall that $0 \le s - s_0 < 1$ and $s_0 \ge 6\ell \ge 3\ell$ by (4). Applying the standard inequality $1 + x \le \mathrm{e}^x$ for $0 \le x \le 1/2$ yields:

$$\left(\frac{s-\ell}{s_0-\ell}\right)^\ell = \left(1 + \frac{s-s_0}{s_0-\ell}\right)^\ell \le \left(1+\frac{1}{2\ell}\right)^\ell \le \exp(0.5) \le 2. \tag{7}$$

Using (7) and (3) once again yields

$$\begin{aligned} F_2 &= 1 - \frac{k}{2}\cdot\left(\frac{s-\ell}{n-\ell}\right)^\ell = 1 - \left(\frac{s-\ell}{s_0-\ell}\right)^\ell \cdot \frac{k}{2}\cdot\left(\frac{s_0-\ell}{n-\ell}\right)^\ell \\ &\ge 1 - 2\cdot\frac{k}{2}\cdot\left(\frac{s_0-\ell}{n-\ell}\right)^\ell = 1 - k\cdot\left(\frac{s_0-\ell}{n-\ell}\right)^\ell = \frac{1}{2}. \end{aligned} \tag{8}$$

Consequently, we have

$$\mathrm{Prob}(\exists i \le k\colon (W_i \subset S)) \ge F_1 \cdot F_2 \ge \frac{1}{2}\cdot\frac{1}{2} = \frac{1}{4},$$

as required. $\square$

Let $k = \lfloor \frac{\varepsilon n}{d+1} \rfloor$, $\ell = d+1$, and recall that Algorithm `Convex-` sets $s = \lceil s_0 \rceil$, where s_0 is given by Equation (3).

We next prove that the algorithm finds the sample S not convex with probability $\ge 1/20$ in each of the 22 repetitions in Step 2 and Step 3. Consider one execution of Step 2 and Step 3. For a fixed $i \le k$, let F_i be the event that $S \cap U_i = \emptyset$. By Lemma 3, we have $|U_i| \ge \frac{(1-\varepsilon)n}{d+1} \ge \frac{n}{2d}$. Observe that

$$\left(1 - \frac{1}{2d}\right)^{d+1} \le \frac{2}{3}, \text{ for } d \ge 2.$$

By (4) we have $s \ge s_0 \ge 3\ell \log k$, thus (recall also that $k \ge 10$, which us used in the last inequality of the chain below)

$$\begin{aligned} \mathrm{Prob}(F_i) &= \mathrm{Prob}(S \cap U_i = \emptyset) = \frac{\binom{n-|U_i|}{s}}{\binom{n}{s}} \\ &= \frac{(n-|U_i|)(n-|U_i|-1)\cdots(n-|U_i|-s+1)}{n(n-1)\cdots(n-s+1)} \le \left(1 - \frac{|U_i|}{n}\right)^s \\ &\le \left(1-\frac{1}{2d}\right)^s \le \left(1-\frac{1}{2d}\right)^{3\ell\log k} \\ &\le \left(\frac{2}{3}\right)^{3\log k} \le \frac{1}{5k}, \text{ for } i \in [k] \text{ and } d \ge 2. \end{aligned}$$

Let E_1 be the event that $S \cap U_i \neq \emptyset$ for every $i \le k$. By the union bound, we deduce that

$$\mathrm{Prob}(\overline{E_1}) \le k \cdot \mathrm{Prob}(F_1) \le \frac{1}{5}.$$

Let E_2 be the event that there exists $i \le k$ such that $W_i \subset S$. We next verify that the inequality $\ell + \frac{n-\ell}{(2k)^{1/\ell}} \le s \le n$ specified in Lemma 4 holds. Indeed,

$$s = \lceil s_0 \rceil \ge s_0 = \ell + \frac{n-\ell}{(2k)^{1/\ell}},$$

and $s_0 < n$ as shown in the proof of Lemma 4, whence $s = \lceil s_0 \rceil \leq n$. Hence by Lemma 4 we have

$$\text{Prob}(E_2) = \text{Prob}(\exists i \leq k \colon (W_i \subset S)) \geq \frac{1}{4}.$$

Putting these bounds together yields

$$\begin{aligned}
\text{Prob}(E_1 \cap E_2) &= 1 - \text{Prob}(\overline{E_1} \cup \overline{E_2}) \geq 1 - \text{Prob}(\overline{E_1}) - \text{Prob}(\overline{E_2}) \\
&\geq 1 - \frac{1}{5} - (1 - \text{Prob}(E_2)) = \text{Prob}(E_2) - \frac{1}{5} \\
&\geq \frac{1}{4} - \frac{1}{5} = \frac{1}{20}.
\end{aligned}$$

Let E be the event that Algorithm `Convex-` finds the sample not convex in at least one of the 22 executions of Step 2 and Step 3. The 22 repetitions are independent events, thus

$$\text{Prob}(E) \geq 1 - \left(1 - \frac{1}{20}\right)^{22} \geq \frac{2}{3}.$$

Thus with probability at least 2/3, Algorithm `Convex-` rejects the input, as required.

2.2 Positive testing: Algorithm `Convex+`

Assume for technical reasons that n is sufficiently large: $n \geq 1500$. Let $0 < \delta \leq 1/2$ be an adjustable parameter. Assume that P is ε-close to convex position for some $\varepsilon > 0$, where $n^{-1} \leq \varepsilon \leq n^{\delta - 1}$; note, this means that P can be made convex by removing at most $\varepsilon n \leq n^\delta$ points.

Algorithm `Convex+`

Step 1: Randomly select a subset $S \subset P$ of size $s = \lceil 1/(6\varepsilon) \rceil$, with all s-subsets being equally likely.

Step 2: Test S for convex position using Chan's algorithm. If S is not in convex position, output a $(d+2)$-point witness of non-convexity and reject P. Otherwise output S as a subset in convex position and accept P.

Time analysis. The setting $s = \lceil 1/(6\varepsilon) \rceil$ in Step 1 yields that the runtime of Algorithm `Convex+` is

$$T(s, s) = O\left(T(1/\varepsilon, 1/\varepsilon)\right) = O\left(\varepsilon^{-\frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \log^{O(1)} 1/\varepsilon\right).$$

Since $\varepsilon = \Omega(1/n)$,

$$T(s, s) = O\left(T(n, n)\right) = O\left(n^{\frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \log^{O(1)} n\right) = o(n^2), \text{ for every } n \text{ and } \varepsilon.$$

Accepting the input with probability $\geq 2/3$. We next show that with probability at least 2/3, Algorithm `Convex+` accepts P and outputs a subset of size $\lceil 1/(6\varepsilon) \rceil$ of P in convex position. By the assumption we can write $P = C \cup D$, where C is in convex position and $|D| \leq \varepsilon n =: t$. Recall that $s = \lceil 1/(6\varepsilon) \rceil$. Note that

$$st = \left\lceil \frac{1}{6\varepsilon} \right\rceil \cdot \varepsilon n \leq \frac{1}{6\varepsilon} \cdot \varepsilon n + \varepsilon n = \frac{n}{6} + \varepsilon n \leq \frac{100n}{595} \text{ for } n \geq 1500.$$

Indeed, $n \geq 1500 \implies n^{0.9} \geq 721 \implies \varepsilon \leq 1/n^{0.9} \leq 1/721$, for which the above inequality holds. In particular, we have $t \leq st \leq 100n/595$. We show that

$$\text{Prob}(S \cap D = \emptyset) = \text{Prob}(S \subseteq C) \geq \frac{2}{3}.$$

Applying the standard inequality $1 - x \geq \mathrm{e}^{-2x}$ for $0 \leq x \leq 1/2$ yields:

$$\begin{aligned}
\text{Prob}(S \subseteq C) &= \frac{\binom{|C|}{s}}{\binom{n}{s}} \geq \frac{\binom{n-t}{s}}{\binom{n}{s}} = \frac{(n-s)(n-s-1)\cdots(n-s-t+1)}{n(n-1)\cdots(n-t+1)} \\
&= \prod_{i=0}^{t-1}\left(1 - \frac{s}{n-i}\right) \geq \left(1 - \frac{s}{n-t+1}\right)^t \geq \exp\left(\frac{-2st}{n-t+1}\right) \\
&\geq \exp\left(\frac{-200}{495}\right) \geq \frac{2}{3},
\end{aligned}$$

as required. Hence with probability at least $2/3$, S is determined to be in convex position and output by the algorithm, as required. Let OPT denote the size of the largest convex subset of P. Since $\mathsf{OPT} \leq n$ and $\varepsilon n \leq n^\delta$, the approximation ratio of Algorithm `Convex+` is

$$\frac{s}{\mathsf{OPT}} \geq \frac{s}{n} = \left\lceil \frac{1}{6\varepsilon} \right\rceil \frac{1}{n} \geq \frac{1}{6\varepsilon n} \geq \frac{1}{6n^\delta}.$$

In particular, when $\delta = 0.1$, the ratio is at least $1/24$ for all $n \leq 10^6$.

3 Concluding remarks

Summary. We presented and analyzed a convexity-testing algorithm implemented by two procedures based on random sampling that has the following enhanced functionality:

1. For point sets that are ε-far from convex position, with probability $\geq 2/3$ the algorithm outputs a $(d+2)$-point witness of non-convexity as a negative certificate.
2. For point sets that are ε-close to convex position, with probability $\geq 2/3$ the algorithm outputs a $1/(6n^\delta)$-approximation of a maximum-size convex subset. [Comment: The current fastest algorithm for computing the largest subset in convex position takes $O(n^3)$ time for $d = 2$, see [6, 10]. In contrast, the problem of computing a largest subset of points in convex position is NP-complete for $d \geq 3$ [14], and moreover, no approximation algorithm is known.]
3. The input range for the tester is significantly extended — for moderate and higher dimensions — compared to the previous version in [7].

A clarifying remark (A question of Eppstein for the planar case). Four-point witnesses to non-convexity can be also viewed as forbidden configurations or obstacles in a convex set of points. Taking this view, sample-based property testing attains the following performance when the sample size is chosen based on the structure of the obstacle set.

Theorem 3. [11, Theorem 6.8] *Let $O_1, O_2, \ldots$ be a finite set of obstacles, whose maximum size is t, and let ε and p be numbers in the range $0 < \varepsilon < 1$ and $0 < p < 1$. Then there is a sample-based property testing algorithm for the property that avoids these obstacles whose sample size, on configurations of size n, is $O(n^{1-1/t})$ and whose false positive rate for configurations that are ε-far from this property is at most p.*

Recall that a *sawtooth* configuration of n points (where n is a multiple of 4) is obtained by adding $n/2$ points very close to the midpoints of the $n/2$ sides of a regular $n/2$-gon and interior to it [11, Definition 3.9]. It is known that a sawtooth configuration of n points is 1/4-close to convex i.e., it can be made convex by removing a quarter, but not fewer, of its points; see, e.g., [11, Observation 1.11]. By Theorem 3, letting $t = 4$ (by the witness structure), $\varepsilon = 1/4$, and $p = 1/3$, indicates that a sample-based convexity testing algorithm with sample size $O(n^{3/4})$ achieves a false positive rate at most 1/3 for configurations that are 1/4-far from convexity.

Likely unaware of the work of Czumaj et al. [7, 9], Eppstein asked the following natural question [11, Open Problem 11.10]: "Does the sample-based property testing algorithm for convexity, with sample size $O(n^{2/3})$, achieve constant false positive rate, or is sample size $\Omega(n^{3/4})$ needed?" Here achieving constant false positive rate means assuring that the false positive rate is bounded from above by a constant. The machinery developed by Czumaj et al. for convexity testing (this includes Lemmas 3.2 and 4.9 in [8]) and revisited here in Section 2 shows that a sample size $O(n^{2/3})$ suffices for that purpose and in general for any constant $0 < \varepsilon < 1$ and $0 < p < 1$. This answers Eppstein's question.

References

[1] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, *Computational Geometry*, 3rd edition, Springer, Heidelberg, 2008.

[2] Arnab Bhattacharyya and Yuichi Yoshida, *Property Testing*, Springer Nature Singapore, 2022.

[3] Manuel Blum, Michael Luby, and Ronitt Rubinfeld, Self-testing/correcting with applications to numerical problems, *Journal of Computer and System Sciences* **47(3)** (1993), 549–595.

[4] Timothy M. Chan, Output-sensitive results on convex hulls, extreme points, and related problems, *Discrete & Computational Geometry* **16(4)** (1996), 369–387.

[5] Bernard Chazelle, An optimal convex hull algorithm in any fixed dimension, *Discrete & Computational Geometry* **10** (1993), 377–409.

[6] Vašek Chvátal and Gheza T. Klincsek, Finding largest convex subsets, *Congressus Numerantium*, **29** (1980), 453–460.

[7] Artur Czumaj, Christian Sohler, and Martin Ziegler, Property testing in computational geometry (extended abstract), in *Proc. 8th Annual European Symposium on Algorithms* (ESA 2000), Springer, Heidelberg, vol. 1879 of LNCS, pp. 155–166. https://doi.org/10.1007/3-540-45253-2_15.

[8] Artur Czumaj, Christian Sohler, and Martin Ziegler, Testing convex position, https://www.researchgate.net/publication/228727099_Testing_Convex_Position. Online manuscript (16 pages), accessed in April 2022.

[9] Artur Czumaj and Christian Sohler, Property testing with geometric queries, in *Proc. 9th Annual European Symposium on Algorithms* (ESA 2001), Springer, Heidelberg, vol. 2161 of LNCS, pp. 266–277. https://doi.org/10.1007/3-540-44676-1_22.

[10] Herbert Edelsbrunner and Leonidas J. Guibas, Topologically sweeping an arrangement, *Journal of Computer and System Sciences* **38(1)** (1989), 165–194.

[11] David Eppstein, *Forbidden Configurations in Discrete Geometry*, Cambridge University Press, 2018.

[12] Paul Erdős and George Szekeres, A combinatorial problem in geometry, *Compositio Mathematica* **2** (1935), 463–470.

[13] Funda Ergün, Sampath Kannan, Ravi S. Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan, Spot-checkers, *Journal of Computer and System Sciences* **60(3)** (2000), 717–751.

[14] Panos Giannopoulos, Christian Knauer, and Daniel Werner, On the computational complexity of Erdős-Szekeres and related problems in $\mathbb{R}^3$, *Proc. 21st European Symposium on Algorithms*, vol. 8125 of LNCS (2013), pp. 541–552.

[15] Oded Goldreich, *Introduction to Property Testing*, Cambridge University Press, 2017.

[16] Jie Han, Yoshiharu Kohayakawa, Marcelo T. Sales, and Henrique Stagni, Property testing for point sets on the plane, *Proc. of Latin American Symposium on Theoretical Informatics* (LATIN 2018), Springer, vol. 10807 of LNCS, pp. 584–596.

[17] Gyula Károlyi and Pavel Valtr, Configurations in d-space without large subsets in convex position, *Discrete & Computational Geometry* **30(2)** (2003), 277–286.

[18] Lásló Lovász, *Combinatorial Problems and Exercises*, 2nd edition, Elsevier, Amsterdam, 1993.

[19] Jiří Matoušek, *Lectures on Discrete Geometry*, Springer, New York, 2002.

[20] Michael Mitzenmacher and Eli Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, 2nd edition, Cambridge University Press, 2017.

[21] Peter McMullen, The maximal number of faces of a convex polytope. *Mathematika* **17** (1970), 179–184.

[22] Albert Nijenhuis and Herbert S. Wilf, *Combinatorial Algorithms*, 2nd edition, Academic Press, New York, 1978.

[23] Cosmin Pohoata and Dmitrii Zakharov, Convex polytopes from fewer points, manuscript, August 2022. Preprint available at `arXiv.org/abs/2208.04878`.

[24] Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, New Jersey, 1977.

[25] Raimund Seidel, Convex hull computations, Chap. 26 in *Handbook of Discrete and Computational Geometry* (Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, eds.), 3rd edition, CRC Press, Boca Raton, 2017, pp.1057–1092.

[26] Andrew Suk, On the Erdős-Szekeres convex polygon problem, *Journal of the American Mathematical Society* **30** (2017), 1047–1053.

[27] Pavel Valtr, Convex independent sets and 7-holes in restricted planar point sets, *Discrete & Computational Geometry* **7(2)** (1992), 135–152.

[28] David P. Williamson and David B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, 2011.

A Remarks on Lemma 3.4 in [8]

The following lemma is suggested in [8]. Here we argue why the lemma cannot be used as is.

Lemma 5. [8]. *Let Ω be an arbitrary set set of n elements. Let k and ℓ be arbitrary integers (possibly dependent on n) and let s be an arbitrary integer such that $s \geq 2n/(2k)^{1/\ell}$. Let $W_1, W_2, \ldots, W_k$ be arbitrary disjoint subsets of Ω each of size ℓ. Let W be a subset of Ω of size s which is chosen independently and uniformly at random. Then*

$$\mathrm{Prob}(\exists j \in [k] \colon (W_j \subseteq W)) \geq \frac{1}{4}.$$

We make two points:

(i) The first point is minor: taking s as the smallest integer satisfying $s \geq 2n/(2k)^{1/\ell}$, namely $s = \lceil 2n/(2k)^{1/\ell} \rceil$ may result in an integer larger than n and thereby be infeasible. For example, the setting $n = 256$, $k = 8$, $\ell = 8$, yields $s = \lceil 2n/(2k)^{1/\ell} \rceil = 363 > 256$.

(ii) The second point requires attention. Reading through the first few lines of the proof suggests that one could take

$$s = \ell + \frac{n-\ell}{(2k)^{1/\ell}}, \text{ or } k\left(\frac{s-\ell}{n-\ell}\right)^{\ell} = \frac{1}{2}. \tag{9}$$

However, this value may be not an integer, and thereby be again infeasible. Suppose that one takes instead the ceiling in the expression of s:

$$s = \ell + \left\lceil \frac{n-\ell}{(2k)^{1/\ell}} \right\rceil. \tag{10}$$

For the above setting in (i), this yields $s = 8 + \left\lceil \frac{248}{(16)^{1/8}} \right\rceil = 8 + 176 = 184$. Then the two factors that appear in the calculation of the lower bound on the probability in question are

$$F_1 = k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} = 8 \cdot \left(\frac{176}{248}\right)^{8} = 0.5147\ldots,$$

$$F_2 = 1 - k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} = 1 - 8 \cdot \left(\frac{176}{248}\right)^{8} = 0.4852\ldots$$

It is now clear that $F_1 \cdot F_2 < \frac{1}{4}$. Taking the floor does not work either. The above example is not an exception, and this occurs whenever the value of s in (9) is not an integer, which happens most of the time.