Discrete Comput Geom 25:569–590 (2001)
DOI: 10.1007/s00454-001-0006-2

Discrete & Computational
Geometry

A Simple Sampling Lemma: Analysis and Applications in Geometric Optimization*

B. Gärtner and E. Welzl

Institut für Theoretische Informatik, ETH Zürich,
ETH Zentrum, CH-8092 Zürich, Switzerland
{gaertner,emo}@inf.ethz.ch

Abstract. Random sampling is an efficient method to deal with constrained optimization problems in computational geometry. In a first step, one finds the optimal solution subject to a random subset of the constraints; in many cases, the expected number of constraints still violated by that solution is then significantly smaller than the overall number of constraints that remain. This phenomenon can be exploited in several ways, and typically results in simple and asymptotically fast algorithms.

Very often the analysis of random sampling in this context boils down to a simple identity (the *sampling lemma*) which holds in a general framework, yet has not been stated explicitly in the literature.

In the more restricted but still general setting of *LP-type problems*, we prove tail estimates for the sampling lemma, giving Chernoff-type bounds for the number of constraints violated by the solution of a random subset. As an application, we provide the first theoretical analysis of *multiple pricing*, a heuristic used in the simplex method for linear programming in order to reduce a large problem to few small ones. This follows from our analysis of a reduction scheme for general LP-type problems, which can be considered as a simplification of an algorithm due to Clarkson. The simplified version needs less random resources and allows a Chernoff-type tail estimate.

1. Introduction

Random sampling and randomized incremental construction have become well-established, by now even classical, design paradigms in the field of computational geometry,

* The first author acknowledges support from the Swiss Science Foundation (SNF), Project No. 21-50647.97. A preliminary version of this paper appeared in the *Proceedings of the* 16*th Annual ACM Symposium on Computational Geometry* (*SCG*), 2000, pp. 91–99.

see [27]. Many algorithms following that paradigm have been simplified to a point where they can easily be taught in introductory CS courses, with almost no technical difficulties. This was not always the case; pioneering papers, notably the ones by Clarkson and Shor [6], [9], Mulmuley [26], and by Guibas et al. [18], still required more technical derivations.

This changed when Seidel popularized the *backwards analysis* paradigm for randomized algorithms [30]. Together with the abstract framework of *configuration spaces*, this technique allows us to treat many different algorithms in a simple and unified way [11].

The goal of this paper is to popularize and prove results around a simple identity (the *sampling lemma*) which underlies the analysis of randomized algorithms for many *geometric optimization* problems. By that we mean problems defined in a low-dimensional space, which usually implies that they have few constraints or few variables when written as mathematical programs.

As we show below, special cases of the identity, or inequalities implied by it, are used in many places, including the analysis of the general configuration space framework. To the knowledge of the authors, the identity itself, however, has not been noticed explicitly.

The Sampling Lemma

Let S be a set of size n and let φ be a function that maps any set $R \subseteq S$ to some value $\varphi(R)$.[1] Define

$$V(R) := \{s \in S \backslash R \mid \varphi(R \cup \{s\}) \neq \varphi(R)\},$$
$$X(R) := \{s \in R \mid \varphi(R \backslash \{s\}) \neq \varphi(R)\}.$$

$V(R)$ is the set of *violators* of R, while $X(R)$ is the set of *extreme* elements in R. Obviously,

$$s \text{ violates } R \quad \Leftrightarrow \quad s \text{ is extreme in } R \cup \{s\}. \tag{1}$$

For a random sample R of size r, i.e. a set R chosen uniformly at random from the set $\binom{S}{r}$ of all r-element subsets of S, we define random variables $V_r\colon\ R \mapsto |V(R)|$ and $X_r\colon\ R \mapsto |X(R)|$, and we consider the expected values

$$v_r := E(V_r),$$
$$x_r := E(X_r).$$

Lemma 1.1 (Sampling Lemma). *For* $0 \leq r < n$,

$$\frac{v_r}{n-r} = \frac{x_{r+1}}{r+1}.$$

[1] Here, the only purpose of φ is to partition 2^S into equivalence classes; later, the function-notation becomes clear.

Proof. Using the definitions of v_r and x_{r+1} as well as (1), we can argue as follows:

$$\begin{aligned}\binom{n}{r} v_r &= \sum_{R \in \binom{S}{r}} \sum_{s \in S \setminus R} [s \text{ violates } R] \\ &= \sum_{R \in \binom{S}{r}} \sum_{s \in S \setminus R} [s \text{ is extreme in } R \cup \{s\}] \\ &= \sum_{Q \in \binom{S}{r+1}} \sum_{s \in Q} [s \text{ is extreme in } Q] \\ &= \binom{n}{r+1} x_{r+1}.\end{aligned}$$

Here, $[\cdot]$ is the indicator variable for the event in brackets. Finally, $\binom{n}{r+1}/\binom{n}{r} = (n - r)/(r + 1)$. □

To appreciate the simplicity (if not triviality) of the lemma, one should consider it as a special case of the following observation: given a bipartite graph, the average vertex degree in one color class times the size of that class equals the average vertex degree in the other color class times its size.

In our case, the two color classes are the subsets of S of sizes r and $r+1$, respectively, and two sets R and $R \cup \{s\}$ share an edge if and only if s violates R (equivalently, if s is extreme in $R \cup \{s\}$). This means, the sampling lemma still holds if "violation" is individually defined for every pair (R, s).

A situation of quite similar flavor, where a simple bipartite graph underlies a probabilistic scenario, has been studied by Dubhashi and Ranjan [12].

We can also establish a version of the sampling lemma in the model of *Bernoulli* sampling, where R is chosen by picking each element of S independently with some fixed probability $p \in [0, 1]$ (we say R is a random *p-sample*). Let $V^{(p)}$ and $X^{(p)}$ denote the random variables for the number of violators and extreme elements, respectively, in a p-sample, and let $v^{(p)}$ and $x^{(p)}$ be the corresponding expectations.

Lemma 1.2 (p-Sampling Lemma). *For* $0 \le p \le 1$,

$$p v^{(p)} = (1 - p) x^{(p)}.$$

Proof. Each r-element set R occurs as a p-sample with probability

$$\binom{n}{r} p^r (1 - p)^{n-r}.$$

Using the Sampling Lemma 1.1 it follows that

$$\begin{aligned}p v^{(p)} &= p \sum_{r=0}^{n} \binom{n}{r} p^r (1 - p)^{n-r} v_r = p \sum_{r=0}^{n-1} \binom{n}{r} p^r (1 - p)^{n-r} v_r \\ &= p \sum_{r=0}^{n-1} \binom{n}{r+1} p^r (1 - p)^{n-r} x_{r+1} = p \sum_{r=1}^{n} \binom{n}{r} p^{r-1} (1 - p)^{n-r+1} x_r\end{aligned}$$

$$= (1-p)\sum_{r=1}^{n}\binom{n}{r}p^r(1-p)^{n-r}x_r = (1-p)\sum_{r=0}^{n}\binom{n}{r}p^r(1-p)^{n-r}x_r$$
$$= (1-p)x^{(p)}. \qquad \square$$

In the next section we discuss some well-known results obtained by random sampling and show that all of them easily follow from the sampling (respectively p-sampling) lemma. Concentrating on the Sampling Lemma 1.1, we elaborate on its connection to configuration spaces and backwards analysis. Section 3 deals with LP-type problems, which can be considered as functions φ with specific properties. Section 4 establishes Chernoff-type tail estimates for the random variable V_r, i.e. for the number of violators of a random sample. The sampling lemma and the tail estimates are finally used in Section 5 to analyze an algorithm for general LP-type problems, which can be considered as the "practical" version of Clarkson's reduction scheme [16]. Its specialization to linear programming is a variant of *multiple pricing* [5].

2. Incarnations of the Sampling Lemmata

Searching in a Sorted Compact List

A *sorted compact list* represents a set S of n ordered keys in an array, where the order among the keys is established by additional pointers linking each element to its predecessor in the order, see Fig. 1. It is well known that the smallest key in a sorted compact list can be found in $O(\sqrt{n})$ expected time [10, Problem 11-3].

For this, one draws a random sample R of $r = \Theta(\sqrt{n})$ keys, finds the smallest key s_0 in the sample, and finally follows the links from s_0 to the overall smallest key. The efficiency comes from the fact that an expected number of only $\Theta(\sqrt{n})$ keys is still smaller than s_0. In general, setting $\varphi(R) = \min(R)$ and observing that $X_{r+1} \equiv 1$, the sampling lemma yields

$$E(\#\{s \in S\backslash R \mid s < \min(R)\}) = \frac{n-r}{r+1}. \tag{2}$$

Note that $s < \min(R)$ is equivalent to $\min(R \cup \{s\}) \neq \min(R)$.

Property (2) was exploited by Seidel in the following observation: given a simple d-polytope P with n vertices, specified by its 1-*skeleton* (the graph of vertices and edges of P), one can find the vertex that minimizes some linear function f in expected time $O(d\sqrt{n})$. The corresponding randomized subroutine serves as a building block of a simple algorithm for computing the intersection of halfspaces, or, dually, the convex hull

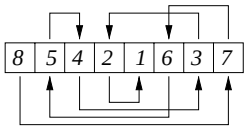

Fig. 1. A sorted list of eight keys, compactly stored in an array.

of points in d-dimensional space. For $d \geq 4$, this algorithm achieves optimal expected worst-case performance [31].

Smallest Enclosing Ball

Consider the problem of computing the smallest enclosing ball of a set S of n points in d-dimensional space, for some fixed d. Randomized incremental algorithms do this in expected $O(n)$ time [33], based on the following fact: if the points are added in random order, the probability that the nth point is outside the smallest enclosing ball of the first $n-1$ points is bounded by $(d+1)/n$. In general, it holds that if $R \subseteq S$ is a random sample of r points, and $\text{ball}(R)$ denotes the smallest enclosing ball of R, then

$$E(\#\{p \in S \backslash R \mid p \notin \text{ball}(R)\}) \leq (d+1)\frac{n-r}{r+1}. \tag{3}$$

Again, this follows from the sampling lemma, with $\varphi(R) = \text{ball}(R)$, together with the observation that any set R has at most $d+1$ extreme elements [33], and the fact that $s \notin \text{ball}(R) \Leftrightarrow \text{ball}(R \cup \{s\}) \neq \text{ball}(R)$.

Similar results hold for the smallest enclosing ellipsoid problem. The randomized incremental algorithm based on them was the first one to achieve an expected runtime of $O(n)$ for that problem, see [33]. The pioneering applications of randomized incremental construction along these lines were Clarkson's and Seidel's linear-time algorithms for linear programming with a fixed number d of variables [8], [29].

Planar Convex Hull

For a planar point set S, $|S| = n$, the randomized incremental construction adds the points in random order, always maintaining the convex hull of the points added so far. When a point p is added, it has to "locate" itself, i.e. it has to know whether it is outside the current convex hull, and in this case identify some hull edge e visible from p.

As it turns out, the amortized expected cost for doing this in the rth step (after which the points added so far form a random sample R of size r) is proportional to a_r/r, where

$$a_r := E(\#\{p \in S \backslash R \mid p \notin \text{conv}(R)\}).$$

The "trick" now is to express this in terms of another quantity:

$$b_r := E(\#\{p \in R \mid p \text{ vertex of } \text{conv}(R)\}).$$

The sampling lemma with $\varphi(R) = \text{conv}(R)$ then shows that

$$a_r = b_{r+1}\frac{n-r}{r+1}. \tag{4}$$

For this, we need the observation that $p \notin \text{conv}(R)$ is equivalent to $\text{conv}(R \cup \{s\}) \neq \text{conv}(R)$, which in turn means that p is a vertex of $\text{conv}(R \cup \{s\})$. The expected overall

location cost (which dominates the runtime) is then proportional to

$$\sum_{r=1}^{n} \frac{a_r}{r} \leq n \sum_{r=1}^{n} \frac{b_{r+1}}{r(r+1)}.$$

Because $b_{r+1} \leq r+1$, this gives an $O(n \log n)$ algorithm. However, the bound is much better in some cases. For example, if the input points are chosen randomly from the unit square (unit disk, respectively), we get $b_r = O(\log r)$ ($b_r = O(\sqrt[3]{r})$, respectively) [28], [20]. In both cases the algorithm actually runs in linear time. In higher dimensions, an analysis along these lines is available, but requires substantial refinements [9], [30].

Minimum Spanning Forests

Let $G = (V, E)$ be an edge-weighted graph, $|V| = n$. For $D \subseteq E$, let $\mathrm{msf}(D)$ denote the minimum spanning forest of the graph (V, D) (which we assume to be unique for all D). An edge $e \in E$ is called *D-light* if it either connects two components of $\mathrm{msf}(D)$ or it has smaller weight than some edge on the unique path in $\mathrm{msf}(D)$ between its two vertices. The expected linear-time algorithm for computing $\mathrm{msf}(E)$ due to Karger et al. [21], [25] relies (among other insights) on the following fact: if D is a random p-sample, the expected number of D-light edges is bounded by n/p. Using the p-Sampling Lemma 1.2, this fact is easily derived. Namely, it is a simple observation that e is D-light if and only if $\mathrm{msf}(D) \neq \mathrm{msf}(D \cup \{e\})$. With $\varphi(D) = \mathrm{msf}(D)$, this means that the set of D-light edges is exactly the set of violators of D. By the p-sampling lemma, if D is a random p-sample, their expected number is given by

$$v^{(p)} = \frac{1-p}{p} x^{(p)} \leq \frac{x^{(p)}}{p}.$$

It remains to observe that $x^{(p)} \leq n-1$, because $X(D)$ contains exactly the edges in $\mathrm{msf}(D)$, for all D.

Along these lines, Chan has proved a bound for the expected number of D-light edges in the case where D is a random sample of size r [4]. His argument uses backwards analysis and boils down to a proof of the Sampling Lemma 1.1 in this specific scenario.

Backwards Analysis and Configuration Spaces

The Sampling Lemma 1.1 in its full generality can be easily proved using backwards analysis, and as indicated in the previous subsection, this is usually the way its specializations are derived in the applications. For this, one considers the randomized incremental "construction" of $\varphi(S)$, via adding the elements of S in random order, and analyzes the situation in step $r+1$ [30].

There is also a connection to configuration spaces. In general, such a space consists of an abstract set of *configurations* over some set S, where each configuration Δ has a *defining* set $D(\Delta) \subseteq S$ and a *conflict* set $K(\Delta) \subseteq S$. Δ is *active* with respect to $R \subseteq S$ if and only if $D(\Delta) \subseteq R$ and $K(\Delta) \subseteq S \backslash R$. The goal is to compute the configurations

active with respect to S, by adding the elements in random order, always maintaining the active configurations of the current subset. The abstract framework provides bounds for the expected overall *structural change* (number of configurations ever becoming active) during that construction [9], [27], [11].

In our case, every subset R has exactly one active configuration $\Delta = \varphi(R)$ associated with it, where $D(\Delta) = X(R)$ and $K(\Delta) = V(R)$.[2] In this case the sampling lemma provides a bound for the expected structural change $v_r/(n-r)$ that occurs in step $r+1$. For example, it specializes to Theorem 9.14 of [11] if x_{r+1} is bounded by a constant d.

In the following we are interested not only in the expectation but also in the distribution of the random variable V_r, something the configuration space framework does not handle. For this, we concentrate on the case in which (S, φ) has the structure of an LP-type problem. This situation covers many important optimization problems, including linear programming and all motivating examples discussed above.

3. LP-Type Problems

If φ maps subsets to some ordered set $\mathcal{O}$, we can consider functions φ that are *monotone*, i.e. $\varphi(F) \leq \varphi(G)$ for $F \subseteq G$. In this situation, we can regard a pair (S, φ) as an optimization problem over $\mathcal{O}$, as follows: S is an abstract set of constraints, and for any $R \subseteq S$, $\varphi(R)$ represents the minimum value in $\mathcal{O}$ subject to the constraints in R. The examples above are all of this type, if we define appropriate orderings on the φ-values. For $\varphi(R) = \min(R)$ in the case of keys, we simply take the decreasing order on the keys. For S a point set and $\varphi(R) = \text{ball}(R)$, we can order the balls according to their radii, while for $\varphi(R) = \text{conv}(R)$, we may use the area of $\text{conv}(R)$.

Moreover, in all these examples, φ has another special property which we refer to as the *locality*. We say that φ is local if $R \subseteq Q$ and $\varphi(R) = \varphi(Q)$ implies $V(R) = V(Q)$, for all $R, Q \subset S$. An example for a nonlocal problem is the *diameter*: for a set S of points and $R \subseteq S$, we define $\varphi(R)$ to be the euclidean diameter of R. In Fig. 2 we have $\varphi(R) = \varphi(Q)$ for $R = \{q, s\}$ and $Q = \{p, q, s\}$, but $\emptyset = V(R) \neq V(Q) = \{r\}$.

Still, locality is present in many problems of practical relevance, the most prominent one being *linear programming* (LP). In a geometric formulation of linear programming, S is a set of halfspaces in d-dimensional space, and $\varphi(R)$ is the lexicographically smallest point among all the ones that minimize some fixed linear function over the intersection

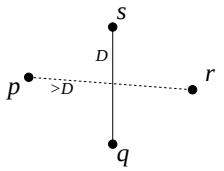

Fig. 2. The diameter problem: locality fails.

[2] Some care is in order here; in degenerate situations, R can define several configurations Δ with different sets $D(\Delta)$, in which case $X(R)$ is the intersection of all those sets.

of all halfspaces in R. If that intersection is empty, we set $\varphi(R) = \infty$, with the understanding that this value dominates all other values. If the function is unbounded over the intersection, we set $\varphi(R) = \perp$, standing for "undefined."

Linear programming is also the motivating example for the following definition [32].

Definition 3.1. Let S be a finite set, $\mathcal{O}$ some ordered set, and $\varphi\colon 2^S \to \mathcal{O} \cup \{\perp\}$ a function, where $\perp$ is assumed to be the minimum value in $\mathcal{O} \cup \{\perp\}$. The pair (S, φ) is called an *LP-type problem* if φ is monotone and local, i.e. if for all $R \subseteq Q \subseteq S$ with $\varphi(R) \neq \perp$,

(i) $\varphi(R) \leq \varphi(Q)$, and
(ii) $\varphi(R) = \varphi(Q)$ implies $V(R) = V(Q)$.

The concept of LP-type problems has proved useful in the understanding of geometric optimization, see for example [2]. For many problems (including linear programming and smallest enclosing ball), the currently best theoretical runtime bounds in the unit cost model can be obtained by an algorithm that works for general LP-type problems [16], [23].

We recall the following further notations only briefly and refer to the above literature for details.

Definition 3.2. Let $\mathcal{L} = (S, \varphi)$ be an LP-type problem.

(i) A *basis of* $R \subseteq S$ is an inclusion-minimal subset $B \subseteq R$ with $\varphi(B) = \varphi(R)$. A *basis in* $\mathcal{L}$ is a basis of some set $R \subseteq S$. A *basis in* R is a basis in $\mathcal{L}$ contained in R.
(ii) The *combinatorial dimension* of $\mathcal{L}$, denoted by $\delta = \delta(\mathcal{L})$, is the size of a largest basis in $\mathcal{L}$.
(iii) $\mathcal{L}$ is *regular* if all bases of sets R, $|R| \geq \delta$ (*regular* bases), have size exactly δ.
(iv) $\mathcal{L}$ is *nondegenerate* if every set R, $|R| \geq \delta$, has a unique basis $B(R)$.

The following implications can easily be derived.

Fact 3.3. *Let $\mathcal{L} = (S, \varphi)$ be an LP-type problem and $R \subseteq S$ with $\varphi(R) \neq \perp$. Then*

(i) $\varphi(R) = \varphi(S \backslash V(R))$, *and*
(ii) *the set $X(R)$ of extreme elements of R is the intersection of all bases of R.*

If $\mathcal{L}$ has combinatorial dimension δ, it follows that $|X(R)| \leq \delta$ for all R, so that the sampling lemma yields

$$v_r \leq \delta \frac{n-r}{r+1}.$$

In particular, a random sample of size $r \approx \sqrt{\delta n}$ has no more than r violators on average, and this is the "balancing" that will prove useful below.

In the next section we derive bounds for regular, nondegenerate LP-type problems that apply to the general case only in a weaker form. While regularity can be enforced in the nondegenerate case (we describe a well-behaved "regularizing" construction below),

nondegeneracy is a more subtle issue. It is not known how to make a general LP-type problem nondegenerate without substantially changing its structure [22]. For most geometric LP-type problems, however, a slight perturbation of the input will entail a nondegenerate problem, essentially equivalent to the original one. Most notably, this is the case for linear programming.

Enforcing Regularity

Given a nondegenerate LP-type problem (S, φ) of combinatorial dimension δ, the idea is to make it regular by "pumping up" bases which are too small. For this, we define an arbitrary linear order on S, and consider the function

$$\varphi'(R) := (\varphi(R), E(R)),$$

where $E(R)$ consists of the vector of the m largest elements in $R \setminus B(R)$, for $m = \min(\delta, |R|) - |B(R)|$. φ'-values are compared lexicographically, i.e. by the φ-component first. If the φ-values are equal, the lexicographic order of the E-components (well defined with respect to the chosen order on S) decides the comparison. φ' can be considered as a "refinement" of φ.

Lemma 3.4 [22]. *If $\mathcal{L} = (S, \varphi)$ is nondegenerate, then (S, φ') is a regular, nondegenerate LP-type problem of combinatorial dimension $\delta(\mathcal{L})$.*

Moreover, if $V(R)$ and $V'(R)$ denote the violating sets of $R \subseteq S$ with respect to φ and φ', we have the following simple but important fact:

$$V(R) \subseteq V'(R). \tag{5}$$

This holds because $\varphi(R \cup \{s\}) > \varphi(R)$ implies $\varphi'(R \cup \{s\}) > \varphi'(R)$. It follows that when we develop tail estimates for the expected size of $V'(R)$ (more generally, for *any* regular and nondegenerate LP-type problem), those estimates then also apply to nonregular problems.

4. Tail Estimates

In the following we consider regular and nondegenerate LP-type problems (S, φ) with $|S| = n$ and $\delta(S, \varphi) = d$, where we assume n and d to be fixed for the rest of this section.

For given parameters $r \geq d$ and k, we want to bound

$$\text{prob}(V_r \geq k).$$

The most important observation is that this quantity does not depend on the LP-type problem, but is merely a function of the parameters n, d, r, and k.

This follows from a result first proved by Clarkson [7] in the context of linear programming, and later generalized to LP-type problems by Matoušek [22]. We rederive the statement here.

Theorem 4.1. *Let (S, φ) be a regular, nondegenerate LP-type problem with $|S| = n$ and $\delta(S, \varphi) = d$. Then*

$$\text{prob}(V_r = k) = \frac{\binom{k+d-1}{d-1}\binom{n-d-k}{r-d}}{\binom{n}{r}}.$$

Proof. A basis B is the basis of a set R if and only if $B \subseteq R \subseteq S \backslash V(B)$. This means, for any regular basis B with k violators, there are $\binom{n-d-k}{r-d}$ sets R of size r which have B as their (unique) basis. It follows that

$$\text{prob}(V_r = k) = b_k \frac{\binom{n-d-k}{r-d}}{\binom{n}{r}}, \qquad r = d, \ldots, n,$$

where b_k is the number of regular bases with k violators in (S, φ). By summing over all k, we get

$$\binom{n}{r} = \sum_{k=0}^{n-d} b_k \binom{n-d-k}{r-d}, \qquad r = d, \ldots, n. \tag{6}$$

This system of linear equations can be written in the form

$$\left(\binom{n}{d}, \binom{n}{d+1}, \ldots, \binom{n}{n}\right) = (b_{n-d}, b_{n-d-1}, \ldots, b_0)\, T,$$

where T is an upper-triangular matrix with all diagonal entries equal to 1, therefore invertible. This means the b_k's are uniquely determined by the system (6), from which

$$b_k = \binom{k+d-1}{d-1}$$

follows via a standard binomial coefficient identity [17, equation (5.26)]. This proves the statement of the theorem. □

This result leads to an explicit formula for $\text{prob}(V_r \geq k)$, but useful tail estimates do not yet follow from that. By severe grinding it might be possible to extract good bounds directly from the formula (we did not succeed), but there is another approach: as we know that the quantity in question does not depend on the particular LP-type problem, we might as well use our favorite LP-type problem in the analysis. In fact, for any given parameters n and d, there is a "canonical" LP-type problem from which statements about the distribution of V_r can be extracted without pain.

The d-Smallest Number Problem

Let N be the set $\{1, \ldots, n\}$. For $R \subseteq N$, define $\min_d(R)$ as the d-smallest number in R (equivalently, the element of rank d in R). If $|R| < d$, this is undefined, and $\min_d(R) := \perp$. We have the following easy facts (proofs omitted).

Lemma 4.2.

(i) *(N, φ) with $\varphi(R) := \min_d(R)$ is a regular, nondegenerate LP-type problem of combinatorial dimension d, if φ-values are compared according to decreasing order in N.*
(ii) *The basis of any set R, $|R| \geq d$, consists of the d smallest numbers in R.*
(iii) *$s \in S \backslash R$ violates R if and only if s is smaller than the d-smallest number in R.*

For $d = 1$, we have $\min_d(R) = \min(R)$, thus we recover the LP-type problem underlying the efficient minimum search in a sorted compact list described in the Introduction.

As a warm-up exercise, we rederive the formula for the number of bases with exactly k violators in a regular and nondegenerate LP-type problem, by using the fact that this number does not depend on the actual LP-type problem, see Theorem 4.1.

Observation 4.3. *The d-smallest number problem has*

$$b_k = \binom{k+d-1}{d-1}$$

regular bases with exactly k violators.

Proof. Any set B with d elements is a regular basis. B has k violators if and only if the d-smallest number x in B is the $(k+d)$-smallest number in N. The elements in $B \backslash \{x\}$ can be any $d-1$ among the $k+d-1$ smaller numbers in N. □

The proof of this observation might be somewhat simpler than the one we had in the general case, but it does not lead to new insights. However, the next theorem about higher moments of V_r is an example of a statement which we think is not immediate to prove (let alone discover) without making use of the d-smallest number problem.

Theorem 4.4. *Let (S, φ) be a regular, nondegenerate LP-type problem, and let R be a random sample of size r. For $j \in \{0, \ldots, n-r\}$, we have*

$$E\left(\binom{V_r}{j}\right) = \frac{\binom{n}{r+j}\binom{j+d-1}{j}}{\binom{n}{r}}.$$

Proof. We evaluate the expectation for the d-smallest number problem and then use Theorem 4.1. For this, we need to count the expected number of sets J, $|J| = j$ with $J \subseteq V(R)$. Observe that this inclusion holds if and only if all elements of J are smaller than the d-smallest number in R, equivalently, if J is among the $j+d-1$ smallest numbers in $R \cup J$. For any set L of size $r+j$, there are $\binom{j+d-1}{j}$ pairs (R, J), $R \cup J = L$, with this property. Thus we get

$$\binom{n}{r} E\left(\binom{V_r}{j}\right) = \sum_{|R|=r} \sum_{\substack{J \subseteq S \backslash R \\ |J|=j}} [J \subseteq V(R)]$$

$$= \sum_{|L|=r+j} \binom{j+d-1}{j}$$
$$= \binom{n}{r+j}\binom{j+d-1}{j}. \qquad \square$$

When applied to $j = 2$, the theorem can be used to compute the variance of V_r, leading to a Chebyshev-type tail estimate. The higher moments give still better bounds. We are going for Chernoff-type bounds, by exploiting the special structure of the d-smallest number problem.

A Chernoff-Type Tail Estimate

To choose a random subset $R \subseteq N$ of size r, one can proceed in r rounds, where round i selects an element s_i uniformly at random among the ones not chosen so far. Equivalently, one may choose a "rank" ℓ_i uniformly at random in $\{1, \ldots, n+1-i\}$ and let s_i be the element of rank ℓ_i among the ones not chosen so far.

Fix some positive integer k and let U_k be the random variable for the number of indices i with $\ell_i \leq k$. We have the following relation to the random variable V_r.

Lemma 4.5. *Let $R = R(\ell)$ denote the set determined by $\ell = (\ell_1, \ldots, \ell_r)$. Then*

$$U_k(\ell) \geq d \quad \Rightarrow \quad V_r(R) \leq k-1.$$

Proof. We claim that $U_k \geq d$ implies $\min_d(R) \leq k+d-1$. Because the latter is equivalent to $V_r \leq k-1$, the lemma follows.

To prove the claim, we first note that

$$s_i = \ell_i + \#\{j < i \mid s_j < s_i\}. \tag{7}$$

Consider some set I of d indices i such that $\ell_i \leq k$ for $i \in I$. Such a set exists if $U_k \geq d$. If $s_i \leq k+d-1$ for all $i \in I$, we get $\min_d(R) \leq k+d-1$, as required. Otherwise, there is some $i \in I$ such that $s_i = k+e$, $e \geq d$. Then we get

$$\#\{j < i \mid s_j < k+e\} = k+e-\ell_i \geq e,$$

which implies $\#\{j < i \mid s_j < k+d\} \geq d$. As before, this means that $\min_d(R) \leq k+d-1$. $\square$

Corollary 4.6. $\text{prob}(V_r \geq k) \leq \text{prob}(U_k \leq d-1)$.

Chernoff-type bounds for U_k are easy to obtain now. U_k can be expressed as the sum of independent random variables $U_{k,i}$, $i = 1, \ldots, r$, where

$$U_{k,i} := \begin{cases} 1, & \text{if} \quad \ell_i \leq k, \\ 0, & \text{otherwise,} \end{cases}$$

and it holds that

$$\text{prob}(U_{k,i} = 1) = \frac{k}{n+1-i} =: p_i.$$

The following is one of the basic Chernoff bounds [19].

Lemma 4.7. *With $E(U_k) = (p_1 + \cdots + p_r)/r$ and $t \geq 0$,*

$$\text{prob}(U_k \leq E(U_k) - t) \leq \exp\left(-\frac{t^2}{2E(U_k)}\right).$$

Using $t = E(U_k) - d + 1$ (which is nonnegative for the values of k we will be interested in below), we obtain

$$\text{prob}(U_k \leq d-1) \leq \exp\left(-\frac{(E(U_k) - d + 1)^2}{2E(U_k)}\right).$$

Fix some value $\lambda \geq 0$ and choose k in such a way that $E(U_k) = (1+\lambda)d$. Then we get

$$\begin{aligned}\text{prob}(U_k \leq d-1) &\leq \exp\left(-\frac{(\lambda d + 1)^2}{2(1+\lambda)d}\right) \\ &\leq \exp\left(-\frac{\lambda^2}{2(1+\lambda)}d\right).\end{aligned}$$

The value of k that entails $E(U_k) = (1+\lambda)d$ satisfies

$$k = \frac{(1+\lambda)d}{\sum_{i=0}^{r-1} 1/(n-i)} \leq (1+\lambda)d\frac{n}{r},$$

and we obtain our result.

Theorem 4.8. *Let $\mathcal{L} = (S, \varphi)$ be a nondegenerate LP-type problem with $|S| = n$ and $\dim(S, \varphi) = d$. For $r \geq d$ and any $\lambda \geq 0$,*

$$\text{prob}\left(V_r \geq (1+\lambda)d\frac{n}{r}\right) \leq \exp\left(-\frac{\lambda^2}{2(1+\lambda)}d\right).$$

We have derived this bound only for regular problems, but as we have shown before, any problem can be regularized, and, by (5), the estimate then also holds for nonregular problems. Because $E(V_r) \leq d(n-r)/(r+1) \approx dn/r$, this bound establishes estimates for the tail "to the right" of the expectation. It might seem that the bound is rather weak, in particular because it does not depend on n and r. However, it is essentially best possible, as the following lower bound shows (the actual formulation has been chosen in order to minimize computational effort).

Theorem 4.9. *Let $\mathcal{L} = (S, \varphi)$ be a nondegenerate LP-type problem with $|S| = n$ and $\dim(S, \varphi) = d$. For $r \geq d$ and any $\lambda \geq 0$ such that $(1+\lambda)d \leq r/2$,*

$$\text{prob}\left(V_r > (1+\lambda)d\frac{n+1-r}{r} - d\right) \geq \exp\left(-(1+\lambda)d - \frac{(1+\lambda)^2 d^2}{r}\right).$$

Proof. With U_k as defined above and $R = R(\ell)$, relation (7) immediately entails

$$V_r(R) \leq k - d \quad \Leftrightarrow \quad \min_d(R) \leq k \quad \Rightarrow \quad U_k(\ell) \geq d,$$

so that we get $\text{prob}(V_r > k - d) \geq \text{prob}(U_k \leq d - 1)$. Furthermore,

$$\text{prob}(U_k \leq d-1) \geq \text{prob}(U_k = 0) = \prod_{i=1}^{r} \left(1 - \frac{k}{n+1-i}\right) \geq \left(1 - \frac{k}{n+1-r}\right)^r.$$

With $k = (1+\lambda)d(n+1-r)/r$, it follows that

$$\text{prob}(U_k \leq d-1) \geq \left(1 - \frac{(1+\lambda)d}{r}\right)^r \geq \exp\left(-(1+\lambda)d - \frac{(1+\lambda)^2 d^2}{r}\right),$$

using the inequality $1 - x \geq \exp(-x - x^2)$ for $x \leq \frac{1}{2}$. □

An open question is whether the statement of Theorem 4.8 also holds in the degenerate case. It is tempting to conjecture that $\text{prob}(V_r \geq k)$ is maximized for nondegenerate problems—this would yield Theorem 4.8 for the general case. Moreover, while the bound is tight in the regular case, one might be able to improve it for a given nonregular problem.

We conclude this section by proving a weaker tail estimate which applies to the general case. Using this, we can show that the number of violators exceeds the expected value by no more than a logarithmic factor, with high probability.

Theorem 4.10. *Let $\mathcal{L} = (S, \varphi)$ be an LP-type problem with $|S| = n$ and $\dim(S, \varphi) = d$. For $r \geq d$ and any $\lambda \geq 0$,*

$$\text{prob}\left(V_r \geq \left(\ln\frac{ne}{d} + \lambda\right) d\frac{n}{r}\right) \leq \exp\left(-\lambda d\right).$$

Proof. Let $\mathcal{B}_k$ denote the set of regular bases with exactly k violators (recall that a regular basis is a basis of some set R with $|R| \geq d$). Any fixed $B \in \mathcal{B}_k$ is a basis of all the sets R satisfying $B \subseteq R \subseteq S \backslash V(B)$. It follows that B is a basis of a random sample R of size r with probability

$$\frac{\binom{n-|B|-k}{r-|B|}}{\binom{n}{r}} \leq \frac{\binom{n-k}{r}}{\binom{n}{r}}.$$

We have $|V(R)| = k$ if and only if R has some basis (equivalently, all its bases) in $\mathcal{B}_k$, which gives

$$\text{prob}(V_r = k) \leq b_k \frac{\binom{n-k}{r}}{\binom{n}{r}}, \qquad b_k = |\mathcal{B}_k|.$$

Consequently,

$$\text{prob}(V_r \geq k) \leq \sum_{\ell=k}^{n-r} b_\ell \frac{\binom{n-\ell}{r}}{\binom{n}{r}},$$

where we know that

$$\sum_{\ell=k}^{n-r} b_\ell \le \binom{n}{\le d} := \sum_{i=0}^{d} \binom{n}{i},$$

because all bases have size at most d. Then we can further argue that

$$\text{prob}(V_r \ge k) \le \left(\sum_{\ell=k}^{n-r} b_\ell\right)\left(\max_{\ell=k\cdots n-r} \frac{\binom{n-\ell}{r}}{\binom{n}{r}}\right) \le \binom{n}{\le d}\frac{\binom{n-k}{r}}{\binom{n}{r}}.$$

Since (see [24])

$$\binom{n}{\le d} \le \left(\frac{ne}{d}\right)^d$$

and

$$\frac{\binom{n-k}{r}}{\binom{n}{r}} = \left(1-\frac{k}{n}\right)\left(1-\frac{k}{n-1}\right)\cdots\left(1-\frac{k}{n-r+1}\right) \le \left(1-\frac{k}{n}\right)^r,$$

we finally get, by substituting $k = (\ln(ne/d) + \lambda)\, d(n/r)$,

$$\begin{aligned}\text{prob}(V_r \ge k) &\le \left(\frac{ne}{d}\right)^d \left(1 - \frac{(\ln(ne/d)+\lambda)\, d}{r}\right)^r \\ &\le \left(\frac{ne}{d}\right)^d \exp\left(-\left(\ln\frac{ne}{d} + \lambda\right) d\right) = \exp\left(-\lambda d\right). \qquad \square\end{aligned}$$

5. Multiple Pricing and Clarkson's Reduction Scheme

The simplex method [5] is usually the most efficient algorithm to solve linear programming problems in practice. Even in the theoretical setting, all known algorithms to solve general LP-type problems boil down to variants of the (dual) simplex method, when they are applied to linear programming [13]. In this section we introduce and analyze an algorithm in the general framework, which—although being new in its precise formulation—follows a well-known design paradigm, whose simplex counterpart is known as *multiple pricing* [5]. The idea of multiple pricing is to reduce a large problem to a (hopefully) small number of small problems. This can be useful in case the whole problem does not fit into main memory, but it also helps in general to reduce the cost of a single simplex iteration. Taking a slightly different approach, *partial pricing* [5] is a related technique following the same paradigm. Applications have been found in the context of very large-scale linear programming [3], but also in geometric optimization [14], [15].

We do not elaborate on those simplex techniques here; the reader may verify that the algorithm we are going to present is actually a variant of multiple pricing, when translated into simplex terminology.

Consider an LP-type problem (S, φ) (not necessarily nondegenerate) of combinatorial dimension d, and assume we are given an algorithm `lp_type`(G, B) to compute for any subset G of S some basis B_G of G, given a candidate basis $B \subseteq G$. Of course, one can directly solve the problem of finding B_S by calling `lp_type` with the large set S and

some basis $B \subseteq S$. As we will see, an efficient alternative is provided by the following method, parameterized with a sample size r. We assume the initial basis B to be fixed for the rest of this section.

Algorithm 5.1.

```
lp_type_sampling_r(S, B):
  (* returns some basis B_S of S *)
  choose R with |R| = r, R ⊆ S\B at random
  G := R ∪ B
  REPEAT
    B := lp_type(G, B)
    G := G ∪ V(B)
  UNTIL V(B) = ∅
  RETURN B
```

`lp_type_sampling` reduces the problem to several calls of `lp_type`, and Fact 3.3(i) shows that if the procedure terminates, $V(B) = \emptyset$ implies that B is a basis of S. Moreover, it must eventually terminate, because every round adds at least one element to G. The algorithm captures the spirit of Clarkson's linear programming algorithm [8] (and its generalizations [1], [16]), but is simpler and more practical. To guarantee its theoretical complexity, Clarkson's algorithm draws a random sample in every round, and it restarts a round whenever $|V(B)|$ turns out to be too large. Thus, Algorithm 5.1 can be interpreted as the canonical simplification of Clarkson's algorithm for practical use, where one observes that resampling and restarting are not necessary (and even decrease the efficiency).

The general phenomenon behind this is that often the theoretically best algorithms are not competitive in practice, while the algorithms one actually chooses in an implementation cannot be analyzed. On the one hand this is due to the fact that the worst-case complexity is an inappropriate measure in many practical situations; on the other hand, sometimes algorithms used in practice are simply not understood, although they might allow a worst-case analysis.

In the case of Algorithm 5.1 we have the fortunate situation that it combines efficiency in practice with provable time bounds (developed below). With the procedure `lp_type` replaced by a call to a standard simplex implementation, the method has been successfully used in a linear programming code for geometric optimization [14], [15], without any further changes. In its original version, due to Clarkson, Algorithm 5.1 is a building-block of an ingenious linear-time algorithm for linear programming in constant dimension d [8], [16].

The theoretical analysis starts with a bound on the number of rounds.

Observation 5.2 [8]. *Fix some basis B_S of S. Then in every round except the last one, $V(B)$ contains an element of B_S. In particular, there are at most $d+1$ rounds.*

Proof. Assume that B_S is disjoint from $V(B)$. From Fact 3.3 and monotonicity we then get $\varphi(B) = \varphi(S\backslash V(B)) \geq \varphi(B_S) = \varphi(S)$, from which $\varphi(B) = \varphi(S)$ follows. Locality then implies $V(B) = V(S) = \emptyset$, which means that we are already in the last round. □

The critical parameter we are interested in is the size of G in the last round. If this is small, then all calls to `lp_type`(G, B) are cheap.

We fix some notation for that. We define $S' := S \backslash B$, B being the initial candidate basis plugged into `lp_type_sampling`. By

$$B_R^{(i)},\ V_R^{(i)},\ \text{and}\ G_R^{(i)}$$

we denote the sets B, $V(B)$, and G computed in round i. Furthermore, we set $G_R^{(0)} = R \cup B$, while $B_R^{(0)}$ and $V_R^{(0)}$ are undefined. This means we have

$$B_R^{(i)} \text{ is a basis of } G_R^{(i-1)}, \qquad V_R^{(i)} = V(G_R^{(i-1)}).$$

If the algorithm performs exactly ℓ rounds, sets with indices $i > \ell$ are defined to be the corresponding sets in round ℓ.

We will need a generalization of Observation 5.2.

Lemma 5.3. *For $j < i \le \ell$, $B_R^{(i)} \cap V_R^{(j)} \neq \emptyset$.*

Proof. Assume on the contrary that $B_R^{(i)} \cap V_R^{(j)} = \emptyset$. As in the proof of Observation 5.2, Fact 3.3 and monotonicity then imply

$$\varphi(G_R^{(j-1)}) = \varphi(S \backslash V_R^{(j)}) \ge \varphi(B_R^{(i)}) = \varphi(G_R^{(i-1)}),$$

a contradiction to the fact that $\varphi(G)$ strictly increases in every round but the last. □

The following lemma is the crucial result. It interprets Algorithm 5.1 as an LP-type problem itself! Under this interpretation, the set G in the last round is essentially the set of violators of the initial sample R. Then the techniques of the previous sections (the sampling lemma and the tail estimates) can be applied to bound the expected size of $|G|$, and even get Chernoff-type bounds for the distribution of $|G|$.

Lemma 5.4. *For $R \subseteq S' := S \backslash B$ define*

$$\varphi'(R) = \left(\varphi(G_R^{(0)}), \varphi(G_R^{(1)}), \ldots, \varphi(G_R^{(d-1)})\right).$$

Then the following holds:

(i) *(S', φ') is an LP-type problem of combinatorial dimension at most $\binom{d+1}{2}$, under the lexicographic order of the d-tuples $\varphi'(R)$.*

(ii) *The set $V'(R) := \{s \in S' \backslash R \mid \varphi'(R) \neq \varphi'(R \cup \{s\})\}$ of violators of R with respect to φ' is given by*

$$V'(R) = V_R^{(1)} \cup \cdots \cup V_R^{(d)} = G_R^{(d)} \backslash (R \cup B).$$

(iii) *If (S, φ) is nondegenerate, so is (S', φ').*

Before we go into the technical (although not difficult) proof, we derive the main result of this section, namely the analysis of Algorithm 5.1. This analysis is now merely a consequence of previous results.

Theorem 5.5. *For $R \subseteq S'$, a random sample of size r,*

$$E(|G_R^{(d)}|) \leq \binom{d+1}{2} \frac{n-d-r}{r+1} + (r+d).$$

Choosing $r = d\sqrt{n/2}$ yields

$$E(|G_R^{(d)}|) \leq 2(d+1)\sqrt{\frac{n}{2}}.$$

Proof. The first inequality directly follows from the sampling lemma, applied to the LP-type problem (S', φ'), together with part (ii) of the previous lemma. The second inequality is routine. □

The theorem shows that Algorithm `lp_type_sampling` reduces a problem of size n to at most d problems of expected size no more than $O(d\sqrt{n})$. This explains the practical efficiency of multiple pricing and similar reduction schemes if $d \ll n$.

If (S, φ) is nondegenerate, we get the following tail estimate, using part (iii) of Lemma 5.4 and Theorem 4.8. Again, routine computations yield

Theorem 5.6. *If (S, φ) is a nondegenerate LP-type problem, then for $R \subseteq S'$, a random sample of size $r = d\sqrt{n/2}$, and $\lambda \geq 0$,*

$$\text{prob}\left(|G_R^{(d)}| \geq (2+\lambda)(d+1)\sqrt{\frac{n}{2}}\right) \leq \exp\left(-\frac{\lambda^2}{2(1+\lambda)}\binom{d+1}{2}\right).$$

In the degenerate case, Theorem 4.10 can be used to derive the following weaker (but still useful) result.

Theorem 5.7. *If (S, φ) is a general LP-type problem, then for $R \subseteq S'$, a random sample of size $r = d\sqrt{(n \ln n)/2}$, and $\lambda \geq 0$,*

$$\text{prob}\left(|G_R^{(d)}| \geq (3+\lambda)\,(d+1)\sqrt{\frac{n \ln n}{2}}\right) \leq \exp\left(-\lambda\binom{d+1}{2}\right).$$

We conclude this section with the proof of Lemma 5.4.

Proof. We start by establishing an auxiliary claim:

Claim. *For any set Q with $Q = R \dot{\cup} T \subseteq S'$ and $i < d$,*

$$\varphi(G_R^{(j)}) = \varphi(G_Q^{(j)}), \qquad j \leq i,$$

implies

$$\begin{aligned} G_Q^{(j)} &= G_R^{(j)} \dot{\cup} T, \qquad j \leq i+1, \\ V_Q^{(j+1)} &= V_R^{(j+1)}, \qquad j \leq i. \end{aligned}$$

To prove the claim, we proceed by induction on i, noting that the statements hold for $i = 0$ by the locality of φ. Now assume the implications are true for $j \leq i-1$. Then we get

$$\begin{aligned} G_Q^{(i)} &= G_Q^{(i-1)} \dot{\cup} V_Q^{(i)} \\ &= G_R^{(i-1)} \dot{\cup} T \dot{\cup} V_R^{(i)} = G_R^{(i)} \dot{\cup} T. \end{aligned}$$

Because $\varphi(G_R^{(i)}) = \varphi(G_Q^{(i)})$, the locality of φ implies

$$V_Q^{(i+1)} = V_R^{(i+1)},$$

which in turn proves $G_Q^{(i+1)} = G_R^{(i+1)} \dot{\cup} T$. This establishes the claim.

To proceed, we first prove part (ii) of Lemma 5.4. Assume $s \in V'(R)$, set $Q := R \cup \{s\}$, and consider the largest index $i < d-1$ such that

$$\varphi(G_R^{(j)}) = \varphi(G_Q^{(j)}), \qquad j \leq i.$$

By the claim above, $G_Q^{(i+1)} = G_R^{(i+1)} \cup \{s\}$, and the monotonicity of φ implies

$$\varphi(G_R^{(i+1)}) < \varphi(G_Q^{(i+1)}). \tag{8}$$

This means, $s \in V_R^{(i+2)}$.

On the other hand, if $s \notin V'(R)$, then the precondition of the claim holds for $i = d-1$, implying

$$V_R^{(j+1)} = V_Q^{(j+1)} \not\ni s, \qquad j \leq d-1.$$

This means $s \notin V_R^{(1)}, \ldots, V_R^{(d)}$.

To prove (i), we need to verify the monotonicity and locality (see Definition 3.1). Inequality (8) shows that $\varphi'(R) \leq \varphi'(R \cup \{s\})$ in the lexicographic order, for all $s \in V'(R)$, and this implies monotonicity.

For locality, assume $R \subseteq Q$ with $\varphi'(R) = \varphi'(Q)$. From the claim and part (ii), we get

$$V'(R) = \bigcup_{i=1}^{d} V_R^{(i)} = \bigcup_{i=1}^{d} V_Q^{(i)} = V'(Q),$$

and this is the required property.

It remains to bound the combinatorial dimension of (S', φ'). To this end we prove that $\varphi'(B_R) = \varphi'(R)$, for

$$B_R := R \cap \bigcup_{i=1}^{d} B_R^{(i)}.$$

We equivalently show that $\varphi(G_R^{(j)}) = \varphi(G_{B_R}^{(j)})$, for $j \leq d-1$, using induction on j. For $j = 0$, we get

$$G_R^{(j)} = G_{B_R}^{(j)} \cup R \backslash B_R,$$

hence

$$\varphi(G_R^{(j)}) = \varphi(G_{B_R}^{(j)} \cup R \backslash B_R) = \varphi(G_{B_R}^{(j)}), \tag{9}$$

because $R \backslash B_R$ is disjoint from $B_R^{(1)}$, the basis of $G_R^{(0)}$. Hence, $R \backslash B_R$ can be removed from $G_R^{(0)}$ without changing the φ-value.

Now assume the statement holds for $j \leq d-2$ and consider the case $j = d-1$. By the claim, we get $G_R^{(j)} = G_{B_R}^{(j)} \cup R \backslash B_R$, so, as before, (9) follows, because $R \backslash B_R$ is disjoint from the basis $B_R^{(j+1)}$ of $G_R^{(j)}$.

To bound the size of B_R, we observe that

$$|R \cap B_R^{(i)}| \leq d + 1 - i,$$

for all $i \leq \ell$ (the number of rounds in which $V(B) \neq \emptyset$). This follows from Lemma 5.3: $B^{(i)}$ has at least one element in each of the $i-1$ sets $V_R^{(1)}, \ldots, V_R^{(i-1)}$, which are in turn disjoint from R. Hence we get

$$|B_R| \leq \sum_{i=1}^{\ell} |R \cap B_R^{(i)}| \leq \binom{d+1}{2}.$$

Proof of part (iii). Nondegeneracy of (S', φ') follows if we can show that every set $R \subseteq S'$ has the set B_R as its unique basis. To this end we prove that whenever we have $L \subseteq R$ with $\varphi'(L) = \varphi'(R)$, then $B_R \subseteq L$.

Fix $L \subseteq R$ with $\varphi'(L) = \varphi'(R)$, i.e.

$$\varphi(G_R^{(i)}) = \varphi(G_L^{(i)}), \qquad i \leq d-1.$$

By the claim, this implies

$$G_R^{(i)} = G_L^{(i)} \dot{\cup} (R \backslash L), \qquad i \leq d-1,$$

and the nondegeneracy of φ yields that $G_R^{(i)}$ and $G_L^{(i)}$ have the same unique basis $B_R^{(i+1)}$, for all i. It follows that $G_L^{(d-1)}$ contains

$$\bigcup_{i=1}^{d} B_R^{(i)},$$

so L contains

$$L \cap \bigcup_{i=1}^{d} B_R^{(i)} = R \cap \bigcup_{i=1}^{d} B_R^{(i)}.$$

The latter equality holds because $R \backslash L$ is disjoint from $G_L^{(d-1)}$, thus in particular from the union of the $B_R^{(i)}$. □

6. Conclusion

The curious fact that—in the regular and nondegenerate case—the distribution of V_r does not depend on the actual LP-type problem, deserves a word of warning: namely, this property does not mean that all nondegenerate LP-type problems with given parameters

n and d are equally difficult (or easy) to solve. On the contrary, because the random variable V_r does not depend on the actual problem, it does not carry any information about the difficulty of a particular problem. There are very easy problems (like d-smallest number), and very difficult ones (like linear programming). For example, Algorithm 5.1 never needs more than two rounds in the case of the d-smallest number, and for other easy LP-type problems characterized by the following property: for any sets $B \subseteq R$ such that $\varphi(B) = \varphi(R)$, and for any set T,

$$\varphi(B \cup T) = \varphi(R \cup T)$$

holds. This means elements in $R \setminus B$ can be "forgotten," as they will not contribute to the final solution. The absence of this property is what makes linear programming and other problems difficult.

In general, it seems that the combinatorial dimension of the LP-type problem (S', φ') derived from (S, φ) according to the definition in Lemma 5.4 is a more meaningful indicator of (S, φ)'s difficulty than $\delta(S, \varphi)$ itself. For example, in the case of the d-smallest number, we get $\delta(S', \varphi') = d$, much less than the $O(d^2)$ upper bound. This alternative notion of dimension needs to be further investigated.

An open problem that remains is to improve the tail estimates in the case of degenerate LP-type problems. Here, the distribution of V_r typically depends on the concrete instance, and so does b_k, the number of bases with k violators. Using only trivial bounds for the numbers b_k, we have obtained the weaker estimate given by Theorem 4.10, indicating that this estimate might not be the final answer.

Acknowledgment

We thank the referee for carefully pointing out simplifications and suggesting improvements in the presentation. In particular, we are grateful for the question concerning the sharpness of our main Chernoff-type bound.

References

[1] I. Adler and R. Shamir. A randomized scheme for speeding up algorithms for linear and convex programming with high constraints-to-variable ratio. *Math. Programming*, 61:39–52, 1993.

[2] N. Amenta. Helly-type theorems and generalized linear programming. *Discrete Comput. Geom.*, 12:241–261, 1994.

[3] R. E. Bixby, J. W. Gregory, I. J. Lustig, R. E. Marsten, and D. F. Shanno. Very large-scale linear programming: a case study in combining interior point and simplex methods. *Oper. Res.*, 40(5):885–897, 1992.

[4] T. Chan. Backwards analysis of the Karger–Klein–Tarjan algorithm for minimum spanning trees. *Inform. Process. Lett.*, 67:303–304, 1998.

[5] V. Chvátal. *Linear Programming*. Freeman, New York, 1983.

[6] K. L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.

[7] K. L. Clarkson. A bound on local minima of arrangements that implies the upper bound theorem. *Discrete Comput. Geom.*, 10:427–233, 1993.

[8] K. L. Clarkson. Las Vegas algorithms for linear and integer programming. *J. Assoc. Comput. Mach.*, 42:488–499, 1995.

[9] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom*., 4:387–421, 1989.
[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, MA., 1990.
[11] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*: *Algorithms and Applications*. Springer-Verlag, Berlin, 1997.
[12] D. Dubhashi and D. Ranjan. Great(er) expectations. *BRICS Newsletter*, 5: 11–13, 1996.
[13] B. Gärtner. Randomized Optimization by Simplex-Type Methods. Ph.D. thesis, Freie Universität, Berlin, 1995.
[14] B. Gärtner. Exact arithmetic at low cost—a case study in linear programming. *Comput. Geom. Theory Appl*., 13:121–139, 1999.
[15] B. Gärtner and S. Schönherr. An efficient, exact and generic quadratic programming solver for geometric optimization. In *Proc*. 16*th ACM Symp. Comput. Geom*., pages 110–118, 2000.
[16] B. Gärtner and E. Welzl. Linear programming—randomization and abstract frameworks. In *Proc*. 13*th Symp. Theoret. Aspects Comput. Sci*., volume 1046 of Lecture Notes in Computer Science, pages 669–687. Springer-Verlag, Berlin, 1996.
[17] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, Reading, MA, 1989.
[18] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7:381–413, 1992.
[19] T. Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Inform. Process. Lett*., 33:305–308, 1990.
[20] S. Har-Peled. On the Expected Complexity of Random Convex Hulls. Technical Report 330, School of Mathematical Sciences, Tel-Aviv University, 1998.
[21] D. Karger, P. N. Klein, and R. E. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *J. Assoc. Comput. Mach*., 42:321–328, 1995.
[22] J. Matoušek. On geometric optimization with few violated constraints. *Discrete Comput. Geom*., 14:365–384, 1995.
[23] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.
[24] J. Matoušek and J. Nešetřil. *Invitation to Discrete Mathematics*. Oxford University Press, Oxford, 1998.
[25] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, 1995.
[26] K. Mulmuley. A fast planar partition algorithm, I. *J. Symbolic Comput*., 10(3–4):253–280, 1990.
[27] K. Mulmuley. *Computational Geometry*: *An Introduction Through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, NJ, 1994.
[28] A. Rényi and R. Sulanke. Über die konvexe Hülle von *n* zufällig gewählten Punkten. *Z. Wahrsch*., 2:75–84, 1963.
[29] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom*., 6:423–434, 1991.
[30] R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of Algorithms and Combinatorics, pages 37–68. Springer-Verlag, New York, 1993.
[31] R. Seidel. Personal communication, 1996.
[32] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc*. 9*th Symp. Theoret. Aspects Comput. Sci*., volume 577 of Lecture Notes in Computer Science, pages 569–579. Springer-Verlag, Berlin, 1992.
[33] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of Lecture Notes in Computer Science, pages 359–370. Springer-Verlag, Berlin, 1991.

Received June 8, 2000, *and in revised form September* 10, 2000. *Online publication March* 26, 2001.