

## Average-Case Analysis of the Double Description Method and the Beneath-Beyond Algorithm

Karl Heinz Borgwardt

Institut für Mathematik, Universität Augsburg,  
86159 Augsburg, Germany  
Borgwardt@math.uni-augsburg.de

**Abstract.** This paper deals with the average computational effort for calculating all vertices of a polyhedron described by  $m$  inequalities in an  $n$ -dimensional space, when we apply the so-called “Double Description Method” (from a dual point of view, i.e. for finding all facets of the convex hull of  $m$  given points, this is equivalent to application of the “Beneath-Beyond Algorithm”). Both are incremental algorithms, i.e. they develop the information about the polyhedron stepwise by taking the inequalities/points successively into regard.

The average-case analysis is done with respect to the Rotation-Symmetry Model, which is well known from the corresponding analysis of the Simplex Method for linear programming. In this model degenerate problems occur with probability 0. So the (finite) effort to solve those problems has no impact on the expected effort in our model.

All the derived results and complexities apply equivalently to both algorithms and to the corresponding primal and dual problems.

### 1. Introduction

#### 1.1. *The Problem*

One of the most fundamental questions arising in computational geometry concerns the arithmetical effort to get a complete (second) description of a polyhedron, for which another (first) description is already available. Two such configurations are well known:

- (Primal question)  $m$  restriction hyperplanes in  $\mathbb{R}^n$ —bounding feasible halfspaces—are given. We are interested in the intersection of these  $m$  halfspaces (this is the polyhedron under consideration). Find all vertices of the resulting (primal) polyhedron.
- (Dual question)  $m$  generating points in  $\mathbb{R}^n$  are given. We are interested in the convex hull of these points (this is the polytope under consideration). Find all the facets

(and if required even all faces) of the convex hull of the points (dual polytope or polyhedron).

Under mild nondegeneracy assumptions (and standardizations) it can be seen that both mathematical problems are dual to each other and under that aspect are somehow equivalent. Besides theoretical interest it is important to select a best possible algorithm to do the job mentioned above (i.e. providing the second information). All practical and theoretical experience shows that there is nothing like an overall superior algorithm. For some polyhedra one method works well, for others a completely different method does the job much better.

So it makes sense to check different approaches to algorithms and—if possible—to study these arithmetical questions even from a probabilistic point of view (average-case analysis). This is what we are doing in this paper, where we restrict our concentration on the first formulation, namely the “vertex-finding problem”.

Before we specialize to one of these algorithms, we present two typical categories of how to approach the challenge:

- **Sequential** algorithms start from one vertex, which had previously been found. Then they determine a next and another next vertex (one after another) until all the vertices are discovered.
- **Incremental** algorithms start with somewhat easier problems. They first solve the vertex-finding problem for only  $n$  of the restrictions (i.e. for an auxiliary polyhedron). Then the information for that stage is saved and one more restriction is introduced. This makes an update of the vertex structure necessary, which is done on the basis of the previously available information. This action is repeated iteratively until all  $m$  restrictions are taken into account. Then the problem has been solved for the true (original) polyhedron.

Prominent representatives of the first (sequential) category are the pivoting algorithm proposed by Charnes et al. [15, pp. 62–70], the further developed version by Dyer [18] and the algorithm of Avis and Fukuda [2]. All of these algorithms entail a walk on the surface of the polyhedron under consideration in a manner similar to the Simplex Method and they combine this procedure with a kind of depth-first search on the graph of vertices and edges of the primal polyhedron. Each time a vertex is reached one proceeds to a new adjacent vertex—if such an undiscovered vertex is available. Else, i.e. if all adjacent vertices are already detected, one has to walk back on the path used before, until there is a new adjacent vertex. Finally, all vertices will have been detected. It is important for the complexity of these methods to store the path constructed so far, this is necessary to find the way back.

This algorithm has been translated and transformed for the dual question under the name the Gift-Wrapping Algorithm by Chand and Kapur [14] (see also [33]). Here a sequence of facets of  $\text{Conv}\{a_1, \dots, a_m\}$  is constructed (these are boundary simplices, resp. convex hulls, of special sets of  $n$  points). Every forward step can be seen as crossing a ridge (convex hull of  $n - 1$  out of those  $n$  points) of the actual facet in order to get to an adjacent facet.

Avis and Fukuda [2] gave another version for the primal question (and implicitly for the dual) relying on reverse Pivot search and avoiding the effort to store the history of the path. This strategy makes use of the fact that in a linear programming problem with the

primal polyhedron as the feasible region the following holds: if the objective function  $c^T x$  has a unique optimal vertex, then every vertex is connected with the optimal vertex by a unique Simplex-Path generated by the Simplex variant under use. So it is possible for each edge to decide whether it belongs to such a path or not. The algorithm of Avis and Fukuda uses only edges belonging to such a Simplex-Path (in reverse direction) for forward steps, because then the optimization process of the Simplex Method will find the right way back without any stored information.

Another important example is the Shelling Algorithm of Seidel [31]. Here the vertices are discovered in decreasing order with respect to a linear objective function  $c^T x$ . To understand the dual version of the Shelling Algorithm one should imagine the light of a lamp located at a point in a facet of  $\text{Conv}\{a_1, \dots, a_m\}$ . When we move that lamp away from the polyhedron (on a fictive straight line), then more and more facets will become illuminated. Seidel's algorithm computes the sequence of facets in the same order as they become illuminated. We can proceed in the corresponding way for the opposite side of the polyhedron. This algorithm has to spend much effort in storing, too.

A typical incremental algorithm is the Double Description Method [27] (see also [21] and [22]) originally introduced by Motzkin et al. in 1953. Here one calculates in sequence the primal polyhedra produced by  $n, n+1, \dots, m-1, m$  of the restrictions. In the original or standard form, one takes the restrictions in their incoming order. However, soon the question arises as to whether a permutation of that order would be advantageous.

The dual version of that algorithm is called the Beneath-Beyond Algorithm. (Sometimes we use this term also for the primal version, because beneath and beyond has an important geometrical meaning there, too). Its history goes back to Grünbaum [23]. Several times it was rediscovered, refined and used under different names. See for instance [28], [32], [19] and [26].

Here, the newly introduced point  $a_i$  is matched with every facet of  $\text{Conv}\{a_1, \dots, a_{i-1}\}$ . Now one has to check whether  $a_i$  is located above (beyond) or below (beneath) the facet under consideration. The "beyond" facets will disappear when we proceed to  $\text{Conv}\{a_1, \dots, a_i\}$ . On the other side new facets will appear. Each of them is the convex hull of  $a_i$  with a ridge between a formerly "beneath" and a formerly "beyond" facet.

The order of introducing the restrictions/points is quite different in the multidimensional version of the Quickhull Algorithm (see [3]). It works identically in the Update procedure, but the order of introduction is changed dynamically. So-called "deep cuts" are preferred. The purpose is that the structure of the final polyhedron shall be approximated very early.

There is a vast literature on these algorithms. The papers deal with quite different aspects. Here are some examples:

We mention, e.g. the work of Bremner [9], who constructs families of polyhedra where the computational effort for incremental primal algorithms must be extremely higher than the number of vertices in the final polyhedron, because the intermediate auxiliary polyhedra are much more complicated than the final one. So, incremental algorithms cannot be called "output-sensitive". This means that the computational effort for calculating the output should be polynomially bounded by the number of (binary) bits required to describe the input and the output of the problem (i.e.  $\text{Size}(\text{Input}) + \text{Size}(\text{Output})$ ) (see also [16]).

Avis et al. [1] show that every algorithm has at least one of several weaknesses:

- either an inability to solve degenerate problems,
- or the inability to hold the intermediate results as simple as the final output.

Bremner et al. [10] raise the question of whether there is a kind of primal–dual algorithm which can exploit the fact that there are two dual and almost equivalent problems to solve (as mentioned in the beginning of this section). The aim is to apply the “easier” version each time.

Joswig [25] deals with the Beneath-Beyond Algorithm and compares it with other convex-hull implementations. It turns out that each algorithm has its own “domain of superiority” in the huge set of polyhedra.

Many researchers deal with Quickhull. Barber et al. [3] present an approach for general dimensions motivated by  $2d$ -experience. For polyhedra showing a so-called balance condition, the behaviour is truly output-sensitive.

As far as we know, there is no truly satisfying theoretical result on the average-case behaviour of the Double Description Method or the Beneath-Beyond Algorithm. This is even worse for Quickhull, where we are (for reasons lying in stochastic theory) even far away from a rigorous probabilistic analysis.

## 1.2. *The Task of This Paper*

The aim of this paper is an average-case analysis of the behaviour of the most popular “incremental” algorithm for finding all vertices of the polyhedron

$$X = \{x \mid a_1^T x \leq 1, \dots, a_m^T x \leq 1\} \quad (1)$$

in a version that is appropriate for nondegenerate problems. The algorithm under consideration is called the “Double Description Method”.

Assume that we already have discovered (calculated) all vertices of the auxiliary polyhedron

$$X_l = \{x \mid a_1^T x \leq 1, \dots, a_l^T x \leq 1\}, \quad \text{where } 1 \leq l \leq m. \quad (2)$$

Then we may proceed in order to obtain the corresponding information for  $X_{l+1}$ . This explains the term incremental:  $X_{l+1}$  is calculated on the basis of the knowledge about  $X_l$ . The update from  $X_l$  to  $X_{l+1}$  consists of two stages:

1. In the first stage we have to decide whether a vertex of  $X_l$  satisfies  $a_{l+1}^T x \leq 1$  or not. In geometrical terms this is the question whether this vertex lies beneath or beyond the hyperplane  $\{x \in \mathbb{R}^n \mid a_{l+1}^T x = 1\}$ .
2. In the second stage all those vertices beyond the hyperplane have to be deleted, and the new vertices on the hyperplane have to be identified and appended to our list of vertices.

As explained in the last section, the Beneath-Beyond Algorithm does the same job indirectly by determining all facets of  $\text{Conv}(0, a_1, \dots, a_m)$  generated by  $n$  of the  $a_i$ -points.

This information can after that be re-translated into the primal question by calculation of the primal vertex which is associated with the discovered dual facet.

Let us have a look on both stages in the dual context. In stage 1 the Beneath-Beyond Algorithm checks every extant facet whether the additional point  $a_{l+1}$  lies beneath the facet hyperplane (i.e. on the side where the origin lies) or beyond (i.e. on the opposite side). In stage 2 those “beyond” facets are deleted, the “beneath” facets are kept and the new facets, which all have  $a_{l+1}$  as one of their generators, are constructed. The effort of doing these steps in the dual context is the same as in the primal context.

Note that sometimes—in a sloppy use of language—we use the term “Beneath-Beyond” also in the primal context, because it has a geometrical meaning even in the primal stage 1.

We base our analysis on the so-called

### Rotation-Symmetry Model (RSM).

$a_1, \dots, a_m$  are distributed independently, identically and symmetrically under rotations on  $\mathbb{R}^n \setminus \{0\}$ . (3)

This stochastic model has been used successfully for an average-case analysis of the Simplex Method in [6] and [8], for Inner-Point Algorithms in [24] and for Gift-Wrapping Algorithms for convex hulls, resp. for finding all vertices of  $X$ , in [7].

In our approach the term “averaging” results from randomly generating the data of problems (here the set of vectors  $\{a_1, \dots, a_m\}$ ) according to a certain distribution rule. After that, for any given data set we simulate a complete solution of the problem by application of the algorithm under consideration. Finally we count (sum up) all the necessary arithmetical operations for all generated data sets and calculate an average number per generated problem.

The resulting mean values (expected values) will be classified according to

1. the number of variables = the dimension  $n$ ,
2. the number  $m$  of restrictions and
3. the specific distribution under use (specified by a parameter  $k$  with  $-1 < k < \infty$ ).

In consequence, we want to achieve the value of  $E_{m,n}^k(s)$ , where  $s$  is the number of arithmetical operations (steps) required to solve a given problem.  $E$  stands for the expected value in the space with parameters  $m, n$  and  $k$ , where  $m$  is the number of restrictions,  $n$  is the number of variables and  $k$  denotes the special distribution parameter under consideration.

Under RSM the following condition is satisfied with probability 1 (almost surely):

### Condition of Nondegeneracy (CoN)

Each subset of  $\{a_1, \dots, a_m\}$  consisting of  $n$  vectors is linearly independent, and each subset consisting of  $(n + 1)$  vectors is in general position. (4)

Fortunately, the remaining complementary cases have no influence on the expected values because all operation numbers are finite and uniformly bounded from above. So we may solve degenerate problems even by extremely lengthy (but finite) enumeration

methods without deteriorating the average computation time. This effect avoids and saves complicated case studies. Hence it simplifies our analysis tremendously.

In the following we use the following notations for balls and spheres:

$$\omega_n := \{x \in \mathbb{R}^n \mid \|x\| = 1\} \quad \text{and} \quad \Omega_n := \{x \in \mathbb{R}^n \mid \|x\| \leq 1\} \quad (5)$$

for unit sphere  $\omega_n$ , resp. unit ball  $\Omega_n$ .

$\lambda_k$  denotes the  $k$ -dimensional Lebesgue-measure of a set, hence

$$\lambda_{n-1}(\omega_n) = \frac{2 \cdot \pi^{n/2}}{\Gamma(n/2)} \quad \text{and} \quad \lambda_n(\Omega_n) = \frac{\pi^{n/2}}{\Gamma((n+2)/2)}. \quad (6)$$

The convexity of the logarithm of the gamma-function admits several estimations, which will frequently be exploited in the calculations of this paper (see [6, Appendix]).

**Remark 1.** For  $x \geq 1$  and all  $0 < \alpha < 1$  we have

$$(x + \alpha - 1)^\alpha \leq \frac{\Gamma(x + \alpha)}{\Gamma(x)} \leq x^\alpha, \quad (7)$$

$$\frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-3}(\omega_{n-2})} = \frac{2\pi}{n-2} \quad \text{and} \quad \frac{\lambda_n(\Omega_n)}{\lambda_{n-2}(\Omega_{n-2})} = \frac{2\pi}{n}, \quad (8)$$

$$\sqrt{\frac{2\pi}{n-1}} \leq \frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-2}(\omega_{n-1})} \leq \sqrt{\frac{2\pi}{n-2}} \quad \text{and} \quad \sqrt{\frac{2\pi}{n+1}} \leq \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \leq \sqrt{\frac{2\pi}{n}}. \quad (9)$$

Rotation-symmetric distributions can be characterized uniquely by specifying their “radial distribution function”:

$$F(r) := P(x \mid \|x\| \leq r) \quad \text{for} \quad r \in [0, \infty). \quad (10)$$

$P(x \mid \|x\| \leq r)$  is the probability that a random point  $x$  has a Euclidean distance to the origin not greater than  $r$ . In our class of distributions we have  $F(r) = 1 \forall r \geq 1$ . These are the distributions with bounded support  $\Omega_n$ .

If the radial distribution has a density, then we denote it by  $f(r)$ , i.e.

$$F(r) = \int_0^r f(\rho) d\rho \quad \forall r \in [0, \infty). \quad (11)$$

Suppose that the original distribution over  $\mathbb{R}^n$  has a density  $\hat{f}$ . Then  $\hat{f}(x_1) = \hat{f}(x_2)$  as long as  $\|x_1\| = \|x_2\|$ , because of rotation symmetry.

Now we introduce a function induced by this property of  $\hat{f}$ , namely  $\tilde{f}: [0, \infty) \rightarrow \mathbb{R}$  such that  $\tilde{f}(\|x\|) = \hat{f}(x)$  for all  $x \in \mathbb{R}^n$ . Then we observe the relation

$$F(r) = \lambda_{n-1}(\omega_n) \int_0^r t^{n-1} \tilde{f}(t) dt,$$

where

$$\tilde{f}(r) = f(r) \frac{1}{r^{n-1} \lambda_{n-1}(\omega_n)}.$$

For the sake of better calculation and evaluation possibilities we restrict our interest to the following subfamily of rotation-symmetric distributions.

We deal only with radial distribution functions parametrized by a value  $k > -1$  of the following kind:

$$F_k(r) := \begin{cases} \frac{\int_0^r (1 - \tau^2)^k \tau^{n-1} d\tau}{\int_0^1 (1 - \tau^2)^k \tau^{n-1} d\tau} & \text{for } 0 \leq r \leq 1, \\ 1 & \text{for } r > 1. \end{cases} \quad (12)$$

The corresponding radial densities of these distributions are

$$f_k(r) := \begin{cases} \frac{(1 - r^2)^k r^{n-1}}{\int_0^1 (1 - \tau^2)^k \tau^{n-1} d\tau} & \text{for } 0 \leq r \leq 1, \\ 0 & \text{for } r > 1. \end{cases} \quad (13)$$

**Remark 2.** The parameter  $k$  gives the weight on the radii between 0 and 1.  $k \rightarrow -1$  puts extremely much weight at the boundary of the ball.  $k \rightarrow \infty$  puts more and more weight in the interior and less at outer spheres.

**Remark 3.** Interesting special cases are:

$$\begin{aligned} k = 0 & \Rightarrow \hat{f} \text{ constant on } \Omega_n && \equiv \text{uniform distribution on } \Omega_n, \\ k \rightarrow -1 & \Rightarrow \text{extremal dominance at } r = 1 && \equiv \text{uniform distribution on } \omega_n, \\ k \rightarrow \infty & \Rightarrow \text{extremal dominance at } r = 0 && \equiv \text{totally centralized,} \\ k = \frac{n-1}{2} & \Rightarrow \text{radial density symmetric about } r = \frac{1}{2}. \end{aligned}$$

The philosophy of this paper is very similar to that of the paper on Gift-Wrapping from 1997, see [7]. Both belong to a long-term project, where several algorithms will be analyzed theoretically under the same stochastic model (RSM).

There have been some rudimentary results on average-case analysis and even for RSM distributions. Compare the work of Dwyer from 1988 [17]. Mostly the results were given in terms of the expected number of vertices (and this number was not specified in  $m$  and  $n$ ). Moreover, they were given only for single distributions (as a “random sphere”) and they were applicable only to the situation where  $m$  was growing tremendously, while  $n$  was fixed. That means the dependency upon  $n$  was not rigorously evaluated.

Here we try to evaluate the average-case behaviour in formulas which specifically exhibit the dependency upon  $m$  and  $n$ . This is done for all configurations with  $m \geq n$ . In cases where it is too difficult to calculate the equations, we put the emphasis on deriving true upper bounds instead of using the order notation. In addition, we are able to study a whole variety of distributions via our parameter  $k$ . Thus we get a variation

of the nonredundancy rate of restrictions moving from 1 to 0 in the limit when  $k$  grows to infinity. Then the algorithms will behave quite differently, and our results can reflect these differences.

For this evaluation we are forced to apply methods of stochastic geometry. One of the most important tools for our analysis and one of the most “classical” problems of stochastic geometry is the calculation of the expected number of vertices  $E_{m,n}(\#(V))$  when the restriction vectors  $a_i$  are distributed according to a certain stochastic model (in our case the RSM distributions). There are many papers dealing with such questions, and the interested reader is referred to a survey by Buchta [11]. (See also [30] and [20].)

For two special distributions under consideration there are rather old results and derivations of that expected number: For the uniform distribution over the unit ball ( $\Omega_n$ ) we know a result of Raynaud [29], which gives an asymptotic approximation

$$E_{m,n}(\#(V)) = \text{Const.}^{\Omega}(n) \cdot m^{(n-1)/(n+1)}(1 + o(1)) \quad (\text{as } m \rightarrow \infty, \text{ and } n \text{ is fixed}).$$

Buchta and Müller [12] gave a formula for that number in the form of a combinatorial sum. A key and extremal role in our study is the uniform distribution over the unit sphere ( $\omega_n$ ). Compare in particular Section 7. Here Buchta et al. [13] again calculate a combinatorial sum and an asymptotic approximation of the form

$$E_{m,n}(\#(V)) = \text{Const.}^{\omega}(n) \cdot m(1 + o(1)) \quad (\text{as } m \rightarrow \infty, \text{ and } n \text{ is fixed}).$$

In this paper we rely on a more general method for deriving such expected numbers in a simple form of nonasymptotic bounds, which allows us to handle our whole family of distributions with varying parameter  $k$  uniformly. The above-mentioned distributions appear as special cases ( $k = 0$ , resp.  $k \rightarrow -1$ ). This related, but slightly different, evaluation method had been independently derived in [4]–[6], at that time only for the purpose of determining the expected number of vertices on a Simplex-Path. Later, it could be used for different geometrical figures as the number of vertices itself [7]. An advantage of this method is that it creates upper bounds for all values of  $m$  and  $n$  and for all  $k$  in a closed form suitable for inserting in the complexity analysis of algorithms (as, e.g. the Double Description Method). For the two mentioned special distributions our method arrives at the same terms with respect to the behaviour in  $n$  and we see that  $(m + 2/(n + 1))^{(n-1)/(n+1)}$ , resp.  $m$ , can in fact be used in the upper bounds without any correction term  $o(1)$ . Details are given in Section 5.2.

### 1.3. The Main Results

Here are our main results about discovering all vertices of  $X$  as in (1).

**Theorem 1.** *Suppose that  $m$  restrictions are randomly generated according to the RSM in dimension  $n$  with distribution-parameter  $k$ . Then the Double Description Method (resp. the Beneath-Beyond Algorithm) determines all vertices of the polyhedron  $X$  with*



an arithmetical effort of at most

$$\begin{aligned}
E_{m,n}^k(s) &\leq O(n^3) \\
&+ 2\pi^{(n/2)} \frac{[n!]^{1/2}}{\Gamma(n/2)} (n+1+2k)^{(n-1)/2} \\
&\cdot \left[ \frac{\Gamma(k+1+n/2)}{2\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \right]^{(n-1)/(n+1+2k)} \\
&\cdot O \left\{ n^{(2+2k)/(n+1+2k)} m^{(n-1)/(n+1+2k)+1} \frac{n+1+2k}{2n+2k} \right. \\
&\quad \left. + n^{2+(2+2k)/(n+1+2k)} m^{(n-1)/(n+1+2k)} \frac{n+1+2k}{n-1} \right. \\
&\quad \left. + n^{1+(2+2k)/(n+1+2k)} m^{(n-1)/(n+1+2k)+1} \frac{n+1+2k}{2n+2k} \right. \\
&\quad \left. + n^{1+(2+2k)/(n+1+2k)} m^{(n-1)/(n+1+2k)} \frac{n+1+2k}{n-1} (\ln m - 1) \right\}. \quad (14)
\end{aligned}$$

For two typical values of  $k$  we get the following (slightly sharper) upper bounds.

**Theorem 2.** *Suppose that  $m$  restrictions are randomly generated according to the RSM in dimension  $n$  with distribution-parameter  $k \rightarrow -1$  (uniform distribution on  $\omega_n$ ). Then the Double Description Method (resp. the Beneath-Beyond Algorithm) determines all vertices of the polyhedron  $X$  with an arithmetical effort of at most*

$$\begin{aligned}
E_{m,n}^{k \rightarrow -1}(s) &\leq O(n^3) + \frac{1}{n} (n-1)^{n-3} \left( \frac{1}{n-2} \right)^{(n-1)/2} (2\pi)^{(n-1)/2} \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2-2n+1}{2}} \\
&\cdot O \left\{ n^2 \frac{m(m-1)}{2} + mn^4 + n^3 \frac{m(m-1)}{2} + n^3 m (\ln(m) - 1) \right\}. \quad (15)
\end{aligned}$$

**Theorem 3.** *Suppose that  $m$  restrictions are randomly generated according to the RSM in dimension  $n$  with distribution-parameter  $k = 0$  (uniform distribution on  $\Omega_n$ ). Then the Double Description Method (resp. the Beneath-Beyond Algorithm) determines all vertices of the polyhedron with an arithmetical effort of at most*

$$\begin{aligned}
E_{m,n}^{k=0}(s) &\leq O(n^3) + \frac{1}{n} \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2+1}{2}} (n+1)^{n-2+2/(n+1)} \sqrt{\frac{2\pi}{n}}^{n-1+2/(n-1)} n^{-(n-1)/(n+1)} \\
&\cdot O \left\{ n^2 m^{1+(n-1)/(n+1)} \frac{2n}{n+1} + n^4 m^{(n-1)/(n+1)} \frac{n+1}{n-1} + n^3 m^{1+(n-1)/(n+1)} \frac{2n}{n+1} \right. \\
&\quad \left. + n^3 m^{(n-1)/(n+1)} \frac{n+1}{n-1} \cdot \left( \ln m - \frac{n+1}{n-1} \right) \right\}. \quad (16)
\end{aligned}$$

Finally, we can compare the general results on incremental algorithms according to RSMs with the results for Gift-Wrapping, resp. sequential algorithms.

**Theorem 4.** *For the special case of uniform distribution on  $\omega_n$  and  $m \gg n$ , the expected number of vertices is almost linear in  $m$ , precisely it is  $m(1 - \varepsilon(m, n)) \cdot C(n)$ , for a function  $\varepsilon(m, n) > 0$  and  $\varepsilon(m, n) \rightarrow 0$  for  $m \rightarrow \infty$ ,  $n$  fixed. In this case, on the average, incremental algorithms with random order have to delete in each stage an expected number of about  $(n - 1)C(n)$  and to produce an expected number of about  $(n)C(n)$  completely new vertices in order to gain an increment of about  $1 \cdot C(n)$  in the expected total number. So in any case we have an overproduction with factor  $n$ .*

*For all other RSM distributions the overproduction factor is even more dramatic, because the deletion share is the same, but the incremental share is less (the expected number here is sublinear in  $m$ ).*

*Hence under our stochastic model and average-case analysis our incremental algorithm has to deal with  $n$ -times more “working vertices” than the final polyhedron  $X_m$  will have on the average. As a consequence of our implementation details, under our criteria it cannot be as efficient as the pivoting algorithm studied in [7].*

## 2. The Algorithm

Before diving deeply into stochastic geometry and integral theory, we describe our algorithm in detail and think about the arithmetical effort for a given deterministic problem. The most important role in that complexity analysis has the size of the output, i.e. the number of vertices which have to be detected. We denote this figure by  $\#(V) := \#(V_m)$ .

Concerning the structure of  $X (= X_m)$  and of  $X_l$  with  $n \leq l \leq m$  and its vertices we should take the following into regard. Here nondegeneracy guarantees that:

- (a)  $X$  is simple, i.e. every vertex of  $X$  is given as the solution of a system

$$a_{\Delta^1}^T x = 1, \dots, a_{\Delta^n}^T x = 1 \quad \text{and} \quad a_j^T x < 1 \quad \forall j \notin \Delta,$$

where  $1 \leq \Delta^1 < \dots < \Delta^n \leq m$  with  $\Delta = \{\Delta^1, \dots, \Delta^n\} \subset \{1, \dots, m\}$ .

- (b) Every edge is a line segment, resp. the solution set of a system

$$a_{\Delta^1}^T x = 1, \dots, a_{\Delta^{i-1}}^T x = 1, \quad a_{\Delta^{i+1}}^T x = 1, \quad a_{\Delta^n}^T x = 1 \\ \text{and} \quad a_j^T x \leq 1 \quad \forall j \in \{1, \dots, m\}.$$

- (c) Edges are incident to one or (mostly) two vertices, where one additional restriction is active or tight, i.e. at one end

$$a_{\Delta^i}^T x = 1 \quad \forall i \in \Delta, \quad \text{and} \quad a_j^T x < 1 \quad \forall j \notin \Delta,$$

and at the other end

$$a_{\Delta^{\bar{i}}}^T x < 1 \quad \text{for one } \bar{i} \in \Delta;$$

$$a_{\Delta^i}^T x = 1 \quad \text{for the remaining } i \in \Delta \quad \text{and}$$

$$a_j^T x = 1 \quad \text{for one } \bar{j} \notin \Delta \quad \text{and} \quad a_j^T x < 1 \quad \text{for the remaining } j \notin \Delta.$$

(d) Every vertex is incident to exactly  $n$  edges defined by

$$a_{\Delta^1}^T x = 1, \dots, a_{\Delta^{i-1}}^T x = 1, \quad a_{\Delta^{i+1}}^T x = 1, \quad a_{\Delta^n}^T x = 1 \quad \text{and} \quad a_j^T x \leq 1, \\ \forall j \in \{1, \dots, m\}.$$

For a probabilistic analysis of these vertices we need to have a dual view on the problem, because the randomly given data are  $a_1, \dots, a_m$ —originating from the dual space.

**Remark 4.** The set  $\{a_{\Delta^1}, \dots, a_{\Delta^n}\} \subset \{a_1, \dots, a_m\}$  determines a vertex  $x_\Delta$  of  $X$  in the primal space via  $a_{\Delta^1}^T x = 1, \dots, a_{\Delta^n}^T x = 1$  if and only if in the dual space the hyperplane through  $a_{\Delta^1}, \dots, a_{\Delta^n}$  is a boundary hyperplane for  $\text{Conv}(a_1, \dots, a_m, 0)$ . That means that  $\text{Conv}(a_{\Delta^1}, \dots, a_{\Delta^n})$  is a facet of  $\text{Conv}(a_1, \dots, a_m, 0)$  and that the origin  $0$  and all points  $a_i$  with  $i \notin \Delta$  belong to the same halfspace (bounded by that hyperplane through  $a_{\Delta^1}, \dots, a_{\Delta^n}$ ).

As already mentioned, our algorithm is an incremental algorithm. We successively solve the following auxiliary problems:

$n$	restrictions	$(a_1, \dots, a_n)$	$(X_n)$
$n + 1$	restrictions	$(a_1, \dots, a_n, a_{n+1})$	$(X_{n+1})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n + r = l$	restrictions	$(a_1, \dots, a_n, a_{n+1}, \dots, a_l)$	$(X_l)$
$n + r + 1 = l + 1$	restrictions	$(a_1, \dots, a_n, a_{n+1}, \dots, a_l, a_{l+1})$	$(X_{l+1})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n + (m - n) = m$	restrictions	$(a_1, \dots, a_n, a_{n+1}, \dots, a_m)$	$(X_m = X)$

We determine the vertex (and edge) structures of  $X_n, X_{n+1}, X_{n+2}, \dots, X_l, X_{l+1}, \dots, X_m = X$  in such a way that each  $X_{l+1}$  is developed from  $X_l$ . As soon as we have calculated the structure of  $X_m = X$ , we are ready.

Informally, we proceed as follows for one update: Assume that the vertex–edge structure of  $X_l$  is already known and available. Then we have to check every vertex  $x_\Delta$  of  $X_l$  to see whether it satisfies  $a_{l+1}^T x_\Delta \leq 1$ . If YES, we keep that vertex. If NO, the vertex is dropped. In the latter case we have to pay attention to a set of just-generated brand new vertices (appearing the first time in  $X_{l+1}$ ). All such new vertices of  $X_{l+1}$  have in common that they are generated as the intersection point of an  $X_l$ -edge and the hyperplane  $\{x \mid a_{l+1}^T x = 1\}$ . So all these new vertices satisfy  $a_{l+1}^T x = 1$ . Therefore it pays to store all the edges of a preliminary polyhedron  $X_l$ .

Following is a more rigorous description (in pseudocode terminology).

#### Initialization

Calculate the only vertex and the  $n$  edges (resp. rays) of

$$X_n = \{x \mid a_1^T x \leq 1, \dots, a_n^T x \leq 1\}.$$

#### Typical Step

for  $l$  from  $n$  to  $m - 1$  do:

Consider the description of  $X_l$ , i.e. its list of triples [vertex—edge—vertex], resp. [vertex—edge— $\infty$ ), and calculate the corresponding list for  $X_{l+1}$ .

### Details of Calculation

- (a) Check all edges to see whether they intersect the hyperplane  $\{x \mid a_{l+1}^T x = 1\}$  or not. If there exists an intersection point, we call that  $x_0$ .
- (b) (i) Keep all edges which belong to the halfspace  $\{x \mid a_{l+1}^T x \leq 1\}$  completely.  
(ii) Drop all edges which belong to the open halfspace  $\{x \mid a_{l+1}^T x > 1\}$  completely. Reduce the degree of the incident vertices each time by one, and drop the vertex as soon as its degree has fallen to 0.  
(iii) Edges crossing the hyperplane are treated as follows:  
(A) If the edge is incident to two vertices  $x_1$  (with  $a_{l+1}^T x_1 \leq 1$ ) and  $x_2$  (with  $a_{l+1}^T x_2 > 1$ ), then replace the edge  $[x_1, x_2]$  by the edge  $[x_1, x_0]$ , i.e. eliminate the line segment  $(x_0, x_2]$ .  
(B) If the edge is incident only to a vertex like  $x_2$ , then shorten that edge (resp. ray) by eliminating the line segment  $(x_0, x_2]$ , reduce the degree of  $x_2$  by one and eliminate  $x_2$  as soon as its degree has reached 0.  
(C) If the edge is incident only to a vertex like  $x_1$ , then replace the edge by  $[x_1, x_0]$  and eliminate the rest.
- (c) Now we have the new vertex  $x_0$ . Calculate all edges emanating from  $x_0$  and keeping  $a_{l+1}^T x \leq 1$  tight. (As a result of nondegeneracy, there are  $n - 1$  such edges.) The direction of the  $n$  edges are given as the  $n$  columns of the negative inverse matrix of

$$A_\Delta = \begin{pmatrix} a_{\Delta^1}^T \\ \vdots \\ a_{\Delta^n}^T \end{pmatrix} \quad \text{in the form } A_\Delta^{-1}(-e_i). \quad (17)$$

Determine the vertex at the other end of the edge or the fact that this edge is unbounded.

**Remark 5.** Note that in degenerate cases, step (c) will become more complicated, since there may be many more potential neighbours of  $x_0$  in the hyperplane, and there may be many directions of the kind mentioned in (17). So we should use combinatorial, resp. enumerative, methods to match each pair of such vertices of the  $x_0$ -type and to decide whether they are adjacent or we should check every such direction to see whether it defines an edge.

In our stochastic model this can be done for all occurring degenerate problems without impact on the expected values. However, as soon as degenerate problems get a positive weight, the quality of the algorithm in comparison with others may change dramatically.

### 3. Deterministic Complexity Considerations

The initialization requires the solution of  $n + 1$  systems (one for the vertex,  $n$  for the edges). Altogether this can be done with  $O(n^3)$  arithmetical operations, and this workload is sufficient for inverting  $A_\Delta$  and for the complete determination of the edge-directions  $A_\Delta^{-1}(-e_i)$  for  $i = 1, \dots, n$ .

For the costs of the update assume that we have a description in the form vertex–edge–vertex or vertex–edge– $\infty$  of  $X_l$  completely.

For each stage  $l = n, \dots, m - 1$  we have to perform one update step. Internally in one update, steps (i) and (ii) require, for each of the  $\#(E_l)$  edges ( $\#$  denotes the number of elements of a set, here of the edges of  $E_l$ ), the calculation of the scalar products  $a_{l+1}^T x_1, a_{l+1}^T x_2$  and the evaluation of the restriction  $a_{l+1}^T x \leq 1$  for  $x = x_1$  and  $x = x_2$ . This causes an effort of  $O(n)$ .

So we need to carry out  $O(\#(E_l)n)$  arithmetical steps.

Step (iii) handles some of these edges in an additional manner. All the intersection points  $x_0$  correspond to vertices in  $V_{l+1} \setminus V_l$ . For each of these  $\#(V_{l+1} \setminus V_l)$  vertices we need  $O(n^3)$  steps to calculate the edge directions.

Now the task of step (c) is to get familiar with the endpoints (vertices) on those edges. Hence we have to apply a quotient criterion to each such edge-direction. Under nondegeneracy this is restricted to exactly  $n$  edges. (Under degeneracy this number may become much higher, which makes this situation much more troublesome.) This costs  $(l + 1 - n)$  scalar products (with effort  $O(n(l + 1 - n))$ ) and  $(l + 1 - n)$  divisions. In total we have  $(l + 1 - n)n$  arithmetic operations per edge.

Note that the difference set can be characterized equivalently as  $V_{l+1}^{\text{act}}(a_{l+1})$ ; this is the set of  $X_{l+1}$ -vertices, where  $a_{l+1}^T x \leq 1$  is tight.

In addition, we introduce the term activity-event of a certain restriction, e.g.  $A.E.(a_{l+1})$  or activity-event in general, i.e. with respect to any of the restrictions  $(A.E.)$ . Then it is clear that in each vertex (under nondegeneracy) we have exactly  $n$  activity-events caused by  $n$  different restrictions, and so we have  $\#(A.E.) = \#(V)$  and  $\#(V^{\text{act}}(a_{l+1})) = \#(A.E.(a_{l+1}))$ .

So step (iii) causes in one update an effort of

$$O(n^3 + (l + 1 - n)n) \cdot \#(V_{l+1}^{\text{act}}(a_{l+1})).$$

Finally, we have to store the new vertices, edges and triples. The cost for each edge is  $n \cdot \ln(l + 1)$ .

We summarize all those contributions:

$$n^3 \quad \text{for initialization,} \quad (18)$$

and for  $l = n, \dots, m - 1$

$$\#(E_l)n \leq \#(V_l) \cdot n \cdot n \quad \text{[for evaluation of all edges],} \quad (19)$$

$$\#(V_{l+1} \setminus V_l)n^3 = \#(V_{l+1}^{\text{act}}(a_{l+1}))n^3 \quad \text{[for all edge directions],} \quad (20)$$

$$\#(V_{l+1} \setminus V_l)n(l + 1 - n)n \quad \text{[for the adjacent vertices],} \quad (21)$$

$$\#(V_{l+1} \setminus V_l)n[n \cdot \ln(l + 1)] \quad \text{[for storing the new edges].} \quad (22)$$

In formula, this looks as follows:

$$s \leq O \left( n^3 + \sum_{l=n}^{m-1} \#(V_l)n^2 + \sum_{l=n}^{m-1} \#(V_{l+1}^{\text{act}}(a_{l+1})) [n^3 + n^2(l + 1 - n) + n^2 \ln(l + 1)] \right). \quad (23)$$

The aim of further evaluations is to get precise estimations or bounds for  $\#(V_l)$  and for  $\#(V_{l+1}^{\text{act}}(a_{l+1}))$ .

In an average-case analysis, we need estimations or bounds for the corresponding expected numbers.

#### 4. Marginal Distribution Functions

As determined in the first section (see (12) and (13)), we deal with radial distribution functions and radial densities of the kind

$$F_k(r) := \begin{cases} \frac{\int_0^r (1-\tau^2)^k \tau^{n-1} d\tau}{\int_0^1 (1-\tau^2)^k \tau^{n-1} d\tau} & \text{for } 0 \leq r \leq 1, \\ 1 & \text{for } r > 1, \end{cases}$$

and

$$f_k(r) := \begin{cases} \frac{(1-r^2)^k r^{n-1}}{\int_0^1 (1-\tau^2)^k \tau^{n-1} d\tau} & \text{for } 0 \leq r \leq 1, \\ 0 & \text{for } r > 1. \end{cases}$$

For these distributions, we can evaluate the marginal distribution and density:

$$G: [-1, 1] \rightarrow [0, 1] \quad \text{with } G(h) = P(x^n \leq h), \quad (24)$$

$$\text{and } g_0: [-1, 1] \rightarrow [0, \infty) \quad \text{with } \int_{-1}^h g_0(\zeta) d\zeta = G(h). \quad (25)$$

For our class of distributions with bounded support  $(\Omega_n)$ , knowledge of the radial distribution allows us to specify these two and two other (very useful) marginal functions:

$$G(h) = 1 - \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \int_h^1 \int_{h/r}^1 (1-\sigma^2)^{(n-3)/2} d\sigma dF(r), \quad (26)$$

$$g_0(h) = \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \int_h^1 \frac{(r^2 - h^2)^{(n-3)/2}}{r^{n-2}} dF(r), \quad (27)$$

$$\begin{aligned} g_1(h) &= g_0(h) \cdot E(|x^1| \mid x^n = h) \\ &= \frac{2\lambda_{n-3}(\omega_{n-2})}{(n-2)\lambda_{n-1}(\omega_n)} \int_h^1 \frac{(r^2 - h^2)^{(n-2)/2}}{r^{n-2}} dF(r) \\ &= \frac{1}{\pi} \int_h^1 \frac{(r^2 - h^2)^{(n-2)/2}}{r^{n-2}} dF(r), \end{aligned} \quad (28)$$

$$\begin{aligned} g_2(h) &= g_0(h) \cdot E(|x^1|^2 \mid x^n = h) \\ &= \frac{\lambda_{n-2}(\omega_{n-1})}{(n-1)\lambda_{n-1}(\omega_n)} \int_h^1 \frac{(r^2 - h^2)^{(n-1)/2}}{r^{n-2}} dF(r). \end{aligned} \quad (29)$$

For the evaluation of these formulas the following is very helpful:

$$\begin{aligned}
 \text{(a)} \quad \int_0^1 (1 - \tau^2)^k \tau^{n-1} d\tau &= \int_0^1 (1 - \tau^2)^k \tau \tau^{n-2} d\tau \quad (\text{substitution } u = \tau^2) \\
 &= \frac{1}{2} \int_0^1 (1 - u)^k \cdot u^{(n-2)/2} du = \frac{\Gamma(k+1)\Gamma(n/2)}{2\Gamma(k+1+n/2)} \\
 &= \frac{1}{2} B\left(k+1, \frac{n}{2}\right) \quad \forall k > -1, \quad n > -1, \quad (30)
 \end{aligned}$$

where  $B(\cdot)$  denotes the well-known beta-function

$$B(k, l) := \int_0^1 x^{k-1} (1-x)^{l-1} dx = \frac{\Gamma(k)\Gamma(l)}{\Gamma(k+l)} \quad \text{for } k > 0, \quad l > 0. \quad (31)$$

$$\begin{aligned}
 \text{(b)} \quad \int_h^1 (r^2 - h^2)^l (1 - r^2)^k r dr &\quad \left(\text{substitution } u = \frac{r^2 - h^2}{1 - h^2}\right) \\
 &= \int_0^1 u^l (1 - h^2)^l (1 - u)^k (1 - h^2)^k (1 - h^2)^{\frac{1}{2}} du \\
 &= \frac{1}{2} (1 - h^2)^{l+k+1} \int_0^1 u^l (1 - u)^k du \\
 &= \frac{1}{2} (1 - h^2)^{l+k+1} \frac{\Gamma(k+1)\Gamma(l+1)}{\Gamma(k+l+2)} \\
 &= \frac{1}{2} (1 - h^2)^{l+k+1} B(k+1, l+1) \quad \forall k > -1, \quad l > -1. \quad (32)
 \end{aligned}$$

Based on the inequalities in (7) we can derive bounds on beta-integrals like

$$\begin{aligned}
 \text{(c)} \quad \frac{1}{l} \left(\frac{k}{l-\delta}\right)^\delta &\leq \binom{k}{l} \int_0^1 x^{k-l} (1-x)^{l-1-\delta} dx \\
 &\leq \frac{1}{l} \left(\frac{k+1-\delta}{l-1}\right)^\delta \quad \forall k, l \in \mathbb{N}, \quad \delta > 0. \quad (33)
 \end{aligned}$$

The use of (a) and (b) allows us to specialize the  $G$ - and  $g$ -functions to the  $k$ -class cases. Here we exploit formulas (26)–(29) and (6):

$$G_k(h) = 1 - \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \frac{2\Gamma(k+1+n/2)}{\Gamma(k+1)\Gamma(n/2)} \int_h^1 \int_{h/r}^1 (1 - \sigma^2)^{(n-3)/2} d\sigma (1 - r^2)^k r^{n-1} dr, \quad (34)$$

$$\begin{aligned}
 g_{0,k}(h) &= \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \frac{2\Gamma(k+1+n/2)}{\Gamma(k+1)\Gamma(n/2)} \int_h^1 \frac{(r^2 - h^2)^{(n-3)/2}}{r^{n-2}} (1 - r^2)^k r^{n-1} dr \\
 &= \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \frac{2\Gamma(k+1+n/2)}{\Gamma(k+1)\Gamma(n/2)} \frac{1}{2} (1 - h^2)^{(n-3)/2+k+1} \frac{\Gamma((n-1)/2)\Gamma(k+1)}{\Gamma(k+1+(n-1)/2)}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \frac{\Gamma(k+1+n/2)}{\Gamma(n/2)} \frac{\Gamma((n-1)/2)}{\Gamma(k+1+(n-1)/2)} (1-h^2)^{(n-1)/2+k} \\
&= \frac{\Gamma(k+1+n/2)}{\sqrt{\pi}\Gamma(k+1+(n-1)/2)} (1-h^2)^{(n-1)/2+k}, \tag{35}
\end{aligned}$$

$$\begin{aligned}
g_{1,k}(h) &= \frac{1}{\pi} \frac{2\Gamma(k+1+n/2)}{\Gamma(k+1)\Gamma(n/2)} \int_h^1 \frac{(r^2-h^2)^{(n-2)/2}}{r^{n-2}} (1-r^2)^k r^{n-1} dr \\
&= \frac{1}{\pi} \frac{2\Gamma(k+1+n/2)}{\Gamma(k+1)\Gamma(n/2)} \frac{1}{2} (1-h^2)^{n/2+k} \frac{\Gamma(n/2)\Gamma(k+1)}{\Gamma(k+1+n/2)} \\
&= \frac{1}{\pi} (1-h^2)^{n/2+k}, \tag{36}
\end{aligned}$$

$$\begin{aligned}
g_{2,k}(h) &= \frac{\lambda_{n-2}(\omega_{n-1})}{(n-1)\lambda_{n-1}(\omega_n)} \frac{2\Gamma(k+1+n/2)}{\Gamma(k+1)\Gamma(n/2)} \int_h^1 \frac{(r^2-h^2)^{(n-1)/2}}{r^{n-2}} (1-r^2)^k r^{n-1} dr \\
&= \frac{\lambda_{n-2}(\omega_{n-1})}{(n-1)\lambda_{n-1}(\omega_n)} \frac{2\Gamma(k+1+n/2)}{\Gamma(k+1)\Gamma(n/2)} \frac{1}{2} (1-h^2)^{(n+1)/2+k} \\
&\quad \cdot \frac{\Gamma((n+1)/2)\Gamma(k+1)}{\Gamma(k+1+(n+1)/2)} \\
&= \frac{\lambda_{n-2}(\omega_{n-1})}{(n-1)\lambda_{n-1}(\omega_n)} \frac{\Gamma(k+1+n/2)}{\Gamma(n/2)} \frac{\Gamma((n+1)/2)}{\Gamma(k+1+(n+1)/2)} (1-h^2)^{(n+1)/2+k} \\
&= \frac{1}{\sqrt{\pi}} \frac{\Gamma(n/2)}{(n-1)\Gamma((n-1)/2)} \frac{\Gamma(k+1+n/2)}{\Gamma(n/2)} \\
&\quad \cdot \frac{\Gamma((n+1)/2)}{\Gamma(k+1+(n+1)/2)} (1-h^2)^{(n+1)/2+k} \\
&= \frac{1}{2\sqrt{\pi}} \frac{\Gamma(k+1+n/2)}{\Gamma(k+1+(n+1)/2)} (1-h^2)^{(n+1)/2+k}. \tag{37}
\end{aligned}$$

## 5. Expected Number of Vertices and Update Vertices

### 5.1. Calculations and Estimations for General $k$ 's

As we have seen before, the expected number of vertices  $\#(V_l)$  and the expected number of update vertices  $\#(V_{l+1} \setminus V_l)$  play the most crucial role in the estimation of the expected effort. If we have the numbers  $E_{l,n}(\#(V_l))$  and  $E_{l+1,n}(\#(V_{l+1}^{\text{act}}(a_{l+1})))$  then we can estimate

$$\begin{aligned}
E_{m,n}(s) &\leq O \left( n^3 + \sum_{l=n}^{m-1} E_{l,n}(\#(V_l)) n^2 \right. \\
&\quad \left. + \sum_{l=n}^{m-1} E_{l+1,n}(\#(V_{l+1}^{\text{act}}(a_{l+1}))) [n^3 + n^2(l+1) + n^2 \ln(l+1)] \right). \tag{38}
\end{aligned}$$



Let us first deal with  $E_{m,n}(\#(V))$ . For this figure we have the integral representation

$$\begin{aligned}
E_{m,n}(\#(V)) &= \binom{m}{n} \cdot \int_{\mathbb{R}^n} \cdots \int_{\mathbb{R}^n} P(a_{n+1}, \dots, a_m \text{ below hyperplane through } a_1, \dots, a_n) \\
&\quad \cdot \hat{f}(a_1) \cdots \hat{f}(a_n) da_1 \cdots da_n \\
&= \binom{m}{n} \cdot \int_{\mathbb{R}^n} \cdots \int_{\mathbb{R}^n} P(a_1, \dots, a_n \text{ induce a facet of } \text{Conv}(a_1, \dots, a_m)) \\
&\quad \cdot \hat{f}(a_1) \cdots \hat{f}(a_n) da_1 \cdots da_n \\
&= \lambda_{n-1}(\omega_n) \cdot \binom{m}{n} \\
&\quad \cdot \int_0^1 \int_{\mathbb{R}^{n-1}} \cdots \int_{\mathbb{R}^{n-1}} G(h)^{m-n} |\text{Det } B| \hat{f}(b_1) \cdots \hat{f}(b_n) d(\bar{b}_1) \cdots d(\bar{b}_n) dh.
\end{aligned} \tag{39}$$

Here

$$B := \begin{bmatrix} b_1^1 & \cdots & b_1^{n-1} & 1 \\ \vdots & & \vdots & \\ b_n^1 & \cdots & b_n^{n-1} & 1 \end{bmatrix}, \quad b_i := \begin{bmatrix} b_i^1 \\ \vdots \\ b_i^{n-1} \\ h \end{bmatrix} \quad \text{and} \quad \bar{b}_i := \begin{bmatrix} b_i^1 \\ \vdots \\ b_i^{n-1} \end{bmatrix}.$$

In certain special cases (as for uniform distribution on  $\omega_n \equiv k \rightarrow -1$  or uniform distribution on  $\Omega_n \equiv k = 0$ ) it is possible to calculate directly the value of

$$\Lambda_B(h) := \int_{\mathbb{R}^{n-1}} \cdots \int_{\mathbb{R}^{n-1}} |\text{Det } B| \hat{f}(b_1) \cdots \hat{f}(b_n) d(\bar{b}_1) \cdots d(\bar{b}_n). \tag{40}$$

We shall return to this point at a later stage.

For general distributions we must get along with upper bounds obtained by use of the Cauchy–Schwartz inequality. Therefore we define

$$\Lambda_0(h) := \int_{\mathbb{R}^{n-1}} \cdots \int_{\mathbb{R}^{n-1}} \hat{f}(b_1) \cdots \hat{f}(b_n) d(\bar{b}_1) \cdots d(\bar{b}_n), \tag{41}$$

$$\Lambda_{B_2}(h) := \int_{\mathbb{R}^{n-1}} \cdots \int_{\mathbb{R}^{n-1}} |\text{Det } B|^2 \hat{f}(b_1) \cdots \hat{f}(b_n) d(\bar{b}_1) \cdots d(\bar{b}_n). \tag{42}$$

Then it is clear that

$$\Lambda_B(h) \leq \sqrt{[\Lambda_{B_2}(h) \cdot \Lambda_0(h)]}. \tag{43}$$

It is rather easy to show that (compare [6])

$$\Lambda_0(h) = g_0(h)^n \quad \text{and} \quad \Lambda_{B_2}(h) = n! g_2(h)^{n-1} g_0(h). \tag{44}$$

Consequently,

$$\Lambda_B(h) \leq [n!]^{1/2} g_0(h)^n \left[ \frac{g_2(h)}{g_0(h)} \right]^{(n-1)/2}. \tag{45}$$

Inserting our  $g$ -formulas for special values of  $k$ , this reads

$$\begin{aligned}
\Lambda_{B,k}(h) &\leq [n!]^{1/2} g_{0,k}(h)^n \left[ \frac{g_{2,k}(h)}{g_{0,k}(h)} \right]^{(n-1)/2} \\
&= [n!]^{1/2} \left[ \frac{\Gamma(k+1+n/2)}{\sqrt{\pi}\Gamma(k+1+(n-1)/2)} (1-h^2)^{(n-1)/2+k} \right]^{n-1} \\
&\quad \cdot g_{0,k}(h) \left[ \frac{1}{2(k+1+(n-1)/2)} (1-h^2) \right]^{(n-1)/2} \\
&= [n!]^{1/2} g_{0,k}(h) \left[ \frac{1}{2\pi} \right]^{(n-1)/2} \\
&\quad \cdot \frac{\Gamma(k+1+n/2)^{n-1}}{\Gamma(k+1+(n-1)/2)^{n-1} (k+1+(n-1)/2)^{(n-1)/2}} (1-h^2)^{[n/2+k](n-1)}. \quad (46)
\end{aligned}$$

So we obtain from (39)

$$\begin{aligned}
E_{m,n}(\#(V)) &\leq \lambda_{n-1}(\omega_n) \cdot \left[ \frac{1}{2\pi} \right]^{(n-1)/2} [n!]^{1/2} \\
&\quad \cdot \frac{\Gamma(k+1+n/2)^{n-1}}{\Gamma(k+1+(n-1)/2)^{n-1} (k+1+(n-1)/2)^{(n-1)/2}} \\
&\quad \cdot \binom{m}{n} \int_0^1 G_k(h)^{m-n} g_{0,k}(h) (1-h^2)^{[n/2+k](n-1)} dh. \quad (47)
\end{aligned}$$

Now we exploit an approximation relation between  $G_k(h)$  and  $(1-h^2)$  from (35):

$$\begin{aligned}
(1-G_k(h)) &= \int_h^1 g_{0,k}(\zeta) d\zeta \\
&= \frac{\Gamma(k+1+n/2)}{\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \int_h^1 (1-\zeta^2)^{(n-1)/2+k} d\zeta \\
&\geq \frac{\Gamma(k+1+n/2)}{\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \int_h^1 (1-\zeta^2)^{(n-1)/2+k} \zeta d\zeta \\
&= \frac{\Gamma(k+1+n/2)}{\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \frac{1}{(n+1+2k)} (1-h^2)^{(n+1)/2+k}. \quad (48)
\end{aligned}$$

Since the upper bound for  $\Lambda_{B,k}(h)$  (compare (46)) is essentially a power of  $(1-h^2)$  too, we can substitute it also by something like a power of  $[1-G_k(h)]$ :

$$\begin{aligned}
E_{m,n}(\#(V)) &\leq \lambda_{n-1}(\omega_n) \\
&\quad \cdot \binom{m}{n} \int_0^1 G_k(h)^{m-n} g_{0,k}(h) [1-G_k(h)]^{([n/2+k](n-1))/[(n+1)/2+k]} dh \\
&\quad \cdot \left[ \frac{1}{2\pi} \right]^{(n-1)/2} [n!]^{1/2} \frac{\Gamma(k+1+n/2)^{n-1}}{\Gamma(k+1+(n-1)/2)^{n-1} (k+1+(n-1)/2)^{(n-1)/2}} \\
&\quad \cdot \left[ \frac{\Gamma(k+1+n/2)}{\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \frac{1}{(n+1+2k)} \right]^{-([n/2+k](n-1))/[(n+1)/2+k]}
\end{aligned}$$

$$\begin{aligned}
&= \lambda_{n-1}(\omega_n) \cdot \binom{m}{n} \int_0^1 G_k(h)^{m-n} g_{0,k}(h) [1 - G_k(h)]^{n-1-(n-1)/(n+1+2k)} dh \\
&\cdot \left[ \frac{1}{2\pi} \right]^{(n-1)/2} [n!]^{1/2} \frac{\Gamma(k+1+n/2)^{n-1}}{\Gamma(k+1+(n-1)/2)^{n-1} (k+1+(n-1)/2)^{(n-1)/2}} \\
&\cdot \left[ \frac{\sqrt{\pi} \Gamma(k+1+(n-1)/2)(n+1+2k)}{\Gamma(k+1+n/2)} \right]^{n-1-(n-1)/(n+1+2k)} \\
&= \lambda_{n-1}(\omega_n) \cdot \binom{m}{n} \cdot \int_{1/2}^1 G^{m-n} [1-G]^{n-1-(n-1)/(n+1+2k)} dG \\
&\text{(beta-integral, see (30), (32) and (33))} \\
&\cdot \left[ \frac{1}{2\pi} \right]^{(n-1)/2} [n!]^{1/2} \left[ \frac{n+1+2k}{(n+1)/2+k} \right]^{(n-1)/2} \\
&\cdot \pi^{(n-1)/2} \left[ \frac{2\sqrt{\pi} \Gamma(k+(n+1)/2)}{\Gamma(k+1+n/2)} \right]^{-(n-1)/(n+1+2k)} (n+1+2k)^{(n-1)/2} \\
&\leq \frac{1}{n} \cdot \left[ \frac{m+(2+2k)/(n+1+2k)}{n-1} \right]^{(n-1)/(n+1+2k)} \\
&\cdot \lambda_{n-1}(\omega_n) [n!]^{1/2} (n+1+2k)^{(n-1)/2} \tag{49} \\
&\cdot \left[ \frac{\Gamma(k+1+n/2)}{2\sqrt{\pi} \Gamma(k+1+(n+1)/2)} \right]^{(n-1)/(n+1+2k)} \\
&= \frac{1}{n} \cdot \left[ \frac{m+(2+2k)/(n+1+2k)}{n-1} \right]^{(n-1)/(n+1+2k)} \\
&\cdot \frac{2\pi^{n/2} [n!]^{1/2}}{\Gamma(n/2)} (n+1+2k)^{(n-1)/2} \\
&\cdot \left[ \frac{\Gamma(k+1+n/2)}{2\sqrt{\pi} \Gamma(k+1+(n+1)/2)} \right]^{(n-1)/(n+1+2k)}.
\end{aligned}$$

This result can be simplified by use of the Stirling formula. The approximation is

$$\begin{aligned}
E_{m,n}(\#(V)) &\leq \approx \frac{1}{n} \left[ \frac{m+(2+2k)/(n+1+2k)}{n-1} \right]^{(n-1)/(n+1+2k)} \\
&\cdot 2^{n/2} \pi^{n/2} n^{3/4} (n+1+2k)^{(n-1)/2} \left[ k+1+\frac{n}{2} \right]^{-(n-1)/(2(n+1+2k))} \\
&\cdot \left[ \frac{1}{2\sqrt{\pi}} \right]^{(n-1)/(n+1+2k)}. \tag{50}
\end{aligned}$$

### 5.2. Sharper Calculations for Two Popular Distributions

For two special realizations of  $k$ , namely  $k \rightarrow -1$  and  $k = 0$  (uniform distribution on  $\omega_n$ , resp. on  $\Omega_n$ ), we can avoid the inexactness caused by the Cauchy–Schwartz inequality. This comes from the fact that here we are able to calculate  $\Lambda_B(h)$  exactly for all  $h \geq 0$ . The reason for that is simply that the intersection of a ball (sphere) with the hyperplane  $\{x \mid x^n = h\}$  reproduces a ball or a sphere of dimension  $n - 1$  with radius  $\sqrt{1 - h^2}$ . Since this holds for any value of  $h$ , the internal/conditional distribution in this intersection is (except for the scaling with  $\sqrt{1 - h^2}$ ) the same for every  $h$ . Therefore we can conclude

$$\Lambda_B^\Omega(h) = \text{Const.}^{k=0} (1 - h^2)^{(n-1)/2} [g_0^{k=0}(h)]^n, \quad (51)$$

$$\Lambda_B^\omega(h) = \text{Const.}^{k \rightarrow -1} (1 - h^2)^{(n-1)/2} [g_0^{k \rightarrow -1}(h)]^n. \quad (52)$$

Now it is known that in both cases for  $m = n$  there is exactly one vertex:

$$\begin{aligned} 1 = E_{n,n}(\#(V)) &= \lambda_{n-1}(\omega_n) \binom{n}{n} \int_0^1 G(h)^{n-n} \Lambda_B(h) dh \\ &= \text{Const.} \lambda_{n-1}(\omega_n) \int_0^1 (1 - h^2)^{(n-1)/2} g_0(h)^n dh. \end{aligned} \quad (53)$$

We can calculate

$$\left[ \lambda_{n-1}(\omega_n) \int_0^1 (1 - h^2)^{(n-1)/2} g_0(h)^n dh \right]^{-1}$$

in both cases. This amounts for  $k \rightarrow -1$ , resp.  $\omega_n$ , to

$$\begin{aligned} \text{Const.}^{k \rightarrow -1} &= \left\{ \lambda_{n-1}(\omega_n) \int_0^1 (1 - h^2)^{(n-1)/2} \left[ \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \right]^n [(1 - h^2)^{(n-3)/2}]^n dh \right\}^{-1} \\ &= \left\{ \lambda_{n-1}(\omega_n) \left[ \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \right]^n \int_0^1 (1 - h^2)^{(n^2-2n-1)/2} dh \right\}^{-1} \\ &= \left\{ \lambda_{n-1}(\omega_n) \left[ \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \right]^n \frac{\sqrt{\pi} \Gamma((n^2 - 2n - 1)/2 + 1)}{2\Gamma((n^2 - 2n - 1)/2 + \frac{3}{2})} \right\}^{-1}. \end{aligned} \quad (54)$$

That means

$$\text{Const.}^{k \rightarrow -1} = \left\{ \frac{1}{\lambda_{n-1}(\omega_n)} \left[ \frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-2}(\omega_{n-1})} \right]^n \frac{2\Gamma((n^2 - 2n - 1)/2 + \frac{3}{2})}{\sqrt{\pi} \Gamma((n^2 - 2n - 1)/2 + 1)} \right\}. \quad (55)$$

In the same way we obtain for  $\Omega_n$ ,

$$\begin{aligned} \text{Const.}^{k=0} &= \left\{ \lambda_{n-1}(\omega_n) \int_0^1 (1 - h^2)^{(n-1)/2} \left[ \frac{\lambda_{n-1}(\Omega_{n-1})}{\lambda_n(\Omega_n)} \right]^n [(1 - h^2)^{(n-1)/2}]^n dh \right\}^{-1} \\ &= \left\{ \lambda_{n-1}(\omega_n) \left[ \frac{\lambda_{n-1}(\Omega_{n-1})}{\lambda_n(\Omega_n)} \right]^n \int_0^1 (1 - h^2)^{(n^2-1)/2} dh \right\}^{-1} \\ &= \left\{ \lambda_{n-1}(\omega_n) \left[ \frac{\lambda_{n-1}(\Omega_{n-1})}{\lambda_n(\Omega_n)} \right]^n \frac{\sqrt{\pi} \Gamma((n^2 + 1)/2)}{2\Gamma((n^2 + 2)/2)} \right\}^{-1}. \end{aligned} \quad (56)$$

That means

$$Const.^{k=0} = \left\{ \frac{1}{\lambda_{n-1}(\omega_n)} \left[ \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \right]^n \frac{2\Gamma((n^2+2)/2)}{\sqrt{\pi}\Gamma((n^2+1)/2)} \right\}. \quad (57)$$

Now we conclude

$$\begin{aligned} E_{m,n}^{k \rightarrow -1}(\#(V)) &= \lambda_{n-1}(\omega_n) \binom{m}{n} \int_0^1 G(h)^{m-n} \Lambda_B^\omega(h) dh \\ &= \lambda_{n-1}(\omega_n) \binom{m}{n} \int_0^1 G(h)^{m-n} g_0(h) g_0(h)^{n-1} (1-h^2)^{(n-1)/2} dh \\ &\quad \cdot \left\{ \frac{1}{\lambda_{n-1}(\omega_n)} \left[ \frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-2}(\omega_{n-1})} \right]^n \frac{2\Gamma((n^2-2n+2)/2)}{\sqrt{\pi}\Gamma((n^2-2n+1)/2)} \right\} \\ &= \binom{m}{n} \left[ \frac{\lambda_{n-2}(\omega_{n-1})}{\lambda_{n-1}(\omega_n)} \right]^{n-1} \\ &\quad \cdot \int_0^1 G(h)^{m-n} g_0(h) [(1-h^2)^{(n-3)/2}]^{n-1} (1-h^2)^{(n-1)/2} dh \\ &\quad \text{(compare (48))} \\ &\quad \cdot \left\{ \left[ \frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-2}(\omega_{n-1})} \right]^n \frac{2\Gamma((n^2-2n+2)/2)}{\sqrt{\pi}\Gamma((n^2-2n+1)/2)} \right\} \\ &\leq \binom{m}{n} \left[ \frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-2}(\omega_{n-1})} \right] \left\{ \frac{2\Gamma((n^2-2n+2)/2)}{\sqrt{\pi}\Gamma((n^2-2n+1)/2)} \right\} (n-1)^{n-2} \\ &\quad \cdot \left[ \frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-2}(\omega_{n-1})} \right]^{n-2} \\ &\quad \cdot \int_{1/2}^1 \Phi^{m-n} [1-\Phi]^{n-2} d\Phi \quad (58) \\ &\leq \frac{\Gamma(m+1)\Gamma(n-1)}{\Gamma(n+1)\Gamma(m)} (n-1)^{n-2} \left[ \frac{\lambda_{n-1}(\omega_n)}{\lambda_{n-2}(\omega_{n-1})} \right]^{n-1} \\ &\quad \cdot \left\{ \frac{2\Gamma((n^2-2n+2)/2)}{\sqrt{\pi}\Gamma((n^2-2n+1)/2)} \right\} \quad (59) \\ &\leq m \frac{1}{n} (n-1)^{n-3} \left( \frac{1}{n-2} \right)^{(n-1)/2} (2\pi)^{(n-1)/2} \\ &\quad \cdot \frac{2}{\sqrt{\pi}} \sqrt{(n^2-2n+1)/2}. \quad (60) \end{aligned}$$

Ignoring constants, we get an order of

$$m \cdot n^{n/2-5/2} \cdot 2^{n/2} \cdot \pi^{n/2}$$

(that means that we have saved  $n^{1/4}$  in comparison with Cauchy–Schwartz).

Correspondingly we get

$$\begin{aligned}
E_{m,n}^{k=0}(\#(V)) &= \lambda_{n-1}(\omega_n) \binom{m}{n} \int_0^1 G(h)^{m-n} \Lambda_B^\Omega(h) dh \\
&= \lambda_{n-1}(\omega_n) \binom{m}{n} \int_0^1 G(h)^{m-n} g_0(h) g_0(h)^{n-1} (1-h^2)^{(n-1)/2} dh \\
&\quad \cdot \left\{ \frac{1}{\lambda_{n-1}(\omega_n)} \left[ \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \right]^n \frac{2\Gamma((n^2+2)/2)}{\sqrt{\pi}\Gamma((n^2+1)/2)} \right\} \\
&= \binom{m}{n} \left[ \frac{\lambda_{n-1}(\Omega_{n-1})}{\lambda_n(\Omega_n)} \right]^{n-1} \\
&\quad \cdot \int_0^1 G(h)^{m-n} g_0(h) [(1-h^2)^{(n-1)/2}]^{n-1} (1-h^2)^{(n-1)/2} dh \\
&\quad \cdot \left\{ \left[ \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \right]^n \frac{2\Gamma((n^2+2)/2)}{\sqrt{\pi}\Gamma((n^2+1)/2)} \right\} \\
&\leq \binom{m}{n} \left[ \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \right] \left\{ \frac{2\Gamma((n^2+2)/2)}{\sqrt{\pi}\Gamma((n^2+1)/2)} \right\} \tag{61} \\
&\quad \cdot \int_{1/2}^1 \Phi^{m-n} [1-\Phi]^{n-2+2/(n+1)} d\Phi \\
&\quad \cdot (n+1)^{n-2+2/(n+1)} \left[ \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \right]^{n-2+2/(n+1)} \\
&\quad \text{(compare (48))} \\
&\leq \frac{\Gamma(m+1)\Gamma(m-n+1)\Gamma(n-1+2/(n+1))}{\Gamma(n+1)\Gamma(m-n+1)\Gamma(m+2/(n+1))} (n+1)^{n-2+2/(n+1)} \\
&\quad \cdot \left[ \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \right]^{n-1+2/(n+1)} \left\{ \frac{2\Gamma((n^2+2)/2)}{\sqrt{\pi}\Gamma((n^2+1)/2)} \right\} \\
&\quad \text{(bounding as in (33))} \tag{62} \\
&\leq \left( \frac{m+2/(n+1)}{n-1} \right)^{(n-1)/(n+1)} \frac{1}{n} (n+1)^{n-2+2/(n+1)} \\
&\quad \cdot \left[ \frac{\lambda_n(\Omega_n)}{\lambda_{n-1}(\Omega_{n-1})} \right]^{n-1+2/(n+1)} \left\{ \frac{2\Gamma((n^2+2)/2)}{\sqrt{\pi}\Gamma((n^2+1)/2)} \right\} \\
&\leq \left( \frac{m+2/(n+1)}{n-1} \right)^{(n-1)/(n+1)} \frac{1}{n} [n+1]^{n-2+2/(n+1)} \left[ \frac{\sqrt{2\pi}}{\sqrt{n}} \right]^{n-1+2/(n+1)} \\
&\quad \cdot \frac{2}{\sqrt{\pi}} \left( \frac{n^2+1}{2} \right)^{1/2}. \tag{63}
\end{aligned}$$

This is an approximate order of

$$\begin{aligned} & \left(m + \frac{2}{n+1}\right)^{(n-1)/(n+1)} n^{-2} n^{n-2} n^{-(n-1)/2} n^{2^{n/2}} \pi^{n/2} \\ &= \left(m + \frac{2}{n+1}\right)^{(n-1)/(n+1)} n^{n/2-5/2} 2^{n/2} \pi^{n/2}. \end{aligned}$$

Again, we are about  $n^{1/4}$  better.

Note that there remain only few (slight) reasons why these two last estimations do not deliver lower bounds simultaneously. One reason is the use of the approximation relation (48), where we had used

$$\int_h^1 (1 - \zeta^2)^{(n-1)/2+k} d\zeta \geq \int_h^1 (1 - \zeta^2)^{(n-1)/2+k} \zeta d\zeta.$$

The quotient can be bounded from below by  $h$ . This error is done with power  $n$  for each  $(1 - G_k(h))$ . However, since we have  $G^{m-n}$  in our integral, we have the impact of the asymptotic movement as  $(m \rightarrow \infty, n \text{ fixed})$ . This effect makes values with  $h \ll 1$  irrelevant. For  $h \rightarrow 1$  the error  $(1/h)^n$  is ignorable.  $h$  then has its gravity at 1.

The same arguments hold after (58) and after (61), where we start the integration at 0 instead of at  $\frac{1}{2}$ . Since we have  $G^{m-n}$ , the region between 0 and  $\frac{1}{2}$  is almost irrelevant for the asymptotic case. So, asymptotically, up to, e.g. (59) or (62) in our two special examples the results can even be regarded as lower bounds.

In addition we need the expected value of  $E_{m,n}(\#(V_{l+1} \setminus V_l))$ .

**Remark 6.**

$$E_{m,n}(\#(V_{l+1} \setminus V_l)) = \frac{n}{l+1} E_{m,n}(\#(V_{l+1})). \quad (64)$$

*Proof.* The expected value of vertices of  $V_{l+1}$  is known from the derivation above. Each vertex rises  $n$  activity events (for each of the tight restrictions exactly one).

That means that in stage  $l+1$ ,

$$E_{m,n}(\#(A.E.)) = n E_{m,n}(\#(V_{l+1})).$$

However, now we exploit the symmetry of the points  $a_1, \dots, a_m$  and the fact that

$$\begin{aligned} E_{m,n}(\#(V_{l+1}^{\text{act}}(a_1))) &= E_{m,n}(\#(V_{l+1}^{\text{act}}(a_2))) = \dots = E_{m,n}(\#(V_{l+1}^{\text{act}}(a_{l+1}))) \\ &= \frac{1}{l+1} E_{m,n}(\#(A.E.)). \end{aligned}$$

Therefore we know that

$$\begin{aligned} E_{m,n}(\#(V_{l+1} \setminus V_l)) &= E_{m,n}(V_{l+1}^{\text{act}}(a_{l+1})) = \frac{1}{l+1} E_{m,n}(\#(A.E.)) \\ &= \frac{1}{l+1} E_{m,n}(\#(V_{l+1})) \cdot n. \quad \square \end{aligned}$$

## 6. The Expected Complexity of the Algorithm

After having an upper bound for  $E_{l,n}(\#(V))$  and for  $E_{l,n}(\#(V_{l+1} \setminus V_l))$ , which could even be sharpened for the special cases of uniform distribution on  $\omega_n$ , resp. on  $\Omega_n$ , we can now summarize. Insertion of the upper bounds into the complexity formulas delivers—under ignorance of constants—

$$\begin{aligned}
E_{m,n}^k(s) &\leq n^3 + n^2 \sum_{l=n}^{m-1} E_{l,n}(\#(V_l)) & (65) \\
&+ \sum_{l=n}^{m-1} E_{l+1,n}(\#(V_{l+1} \setminus V_l)) \{n^3 + n^2(l+1) + n^2 \ln(l+1)\} \\
&\leq n^3 + n^2 \sum_{l=n}^{m-1} \frac{1}{n} \left[ \frac{l + (2+2k)/(n+1+2k)}{n-1} \right]^{(n-1)/(n+1+2k)} \\
&\quad \cdot 2\pi^{(n/2)} \frac{[n!]^{1/2}}{\Gamma(n/2)} (n+1+2k)^{(n-1)/2} \\
&\quad \cdot \left[ \frac{\Gamma(k+1+n/2)}{2\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \right]^{(n-1)/(n+1+2k)} \\
&\quad + \sum_{l=n}^{m-1} \frac{n}{l+1} \frac{1}{n} \left[ \frac{l+1+(2+2k)/(n+1+2k)}{n} \right]^{(n-1)/(n+1+2k)} \\
&\quad \cdot 2\pi^{(n/2)} \frac{[n!]^{1/2}}{\Gamma(n/2)} (n+1+2k)^{(n-1)/2} \\
&\quad \cdot \left[ \frac{\Gamma(k+1+n/2)}{\sqrt{2\pi}\Gamma(k+1+(n-1)/2)} \right]^{(n-1)/(n+1+2k)} \{n^3 + n^2(l+1) + n^2 \ln(l+1)\} \\
&= n^3 + 2\pi^{n/2} \frac{[n!]^{1/2}}{\Gamma(n/2)} (n+1+2k)^{(n-1)/2} \\
&\quad \cdot \left[ \frac{\Gamma(k+1+n/2)}{2\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \right]^{(n-1)/(n+1+2k)} \\
&\quad \cdot \left\{ n \sum_{l=n}^{m-1} \left[ \frac{l + (2+2k)/(n+1+2k)}{n-1} \right]^{(n-1)/(n+1+2k)} \right. \\
&\quad + n^3 \sum_{l=n}^{m-1} \left[ \frac{l+1+(2+2k)/(n+1+2k)}{n} \right]^{(n-1)/(n+1+2k)-1} \\
&\quad + n^2 \sum_{l=n}^{m-1} \left[ \frac{l+1+(2+2k)/(n+1+2k)}{n} \right]^{(n-1)/(n+1+2k)} \\
&\quad \left. + n^2 \sum_{l=n}^{m-1} \left[ \frac{l+1+(2+2k)/(n+1+2k)}{n} \right]^{(n-1)/(n+1+2k)-1} \ln(l+1) \right\}
\end{aligned}$$



$$\begin{aligned}
&\leq n^3 + 2\pi^{(n/2)} \frac{[n!]^{1/2}}{\Gamma(n/2)} (n+1+2k)^{(n-1)/2} \\
&\quad \cdot \left[ \frac{\Gamma(k+1+n/2)}{2\sqrt{\pi}\Gamma(k+1+(n-1)/2)} \right]^{(n-1)/(n+1+2k)} \\
&\quad \cdot \left\{ n^{(2+2k)/(n+1+2k)} \left( m + \frac{2+2k}{n+1+2k} \right)^{(n-1)/(n+1+2k)+1} \frac{n+1+2k}{2n+2k} \right. \\
&\quad + n^{2+(2+2k)/(n+1+2k)} \left( m + \frac{2+2k}{n+1+2k} \right)^{(n-1)/(n+1+2k)} \frac{n+1+2k}{n-1} \\
&\quad + n^{1+(2+2k)/(n+1+2k)} \left( m + \frac{2+2k}{n+1+2k} \right)^{(n-1)/(n+1+2k)+1} \frac{n+1+2k}{2n+2k} \\
&\quad + n^{1+(2+2k)/(n+1+2k)} (m + (2+2k)/(n+1+2k))^{(n-1)/(n+1+2k)} \\
&\quad \left. \cdot \frac{n+1+2k}{n-1} (\ln m - 1) \right\}. \tag{66}
\end{aligned}$$

If we are interested only in the order, we may forget about the first summand in brackets, because it is obviously dominated by the third.

**Remark 7.** With the use of the Stirling formula, the quotient  $[n!]^{1/2}/\Gamma(n/2)$  can be approximated by

$$\begin{aligned}
\frac{(n/e)^{n/2} (2\pi n)^{1/4}}{e^{(n/2-1/2)\ln(n/2)-n/2} (2\pi)^{1/2}} &= \frac{n^{n/2+1/4} (2\pi)^{1/4} e(n/2)}{e^{(n/2)(n/2-1/2)} (2\pi)^{1/2}} = \left(\frac{n}{2}\right)^{3/4} (2\pi)^{-1/4} 2^{n/2-1/2} \\
&= n^{3/4} (\pi)^{-1/4} 2^{n/2-3/2}
\end{aligned}$$

and  $\Gamma(k+1+n/2)/\Gamma(k+1+(n+1)/2)$  can be bounded from above by  $(k+(n+1)/2)^{-1/2}$ .

We have seen that for  $\omega_n$ - and  $\Omega_n$ -uniform distributions sharper bounds are available (approximately  $n^{1/4}$  can be saved). Here the results are

$$\begin{aligned}
E_{m,n}^{k \rightarrow -1}(s) &\leq n^3 + n^2 \sum_{l=n}^{m-1} \frac{1}{n} l (n-1)^{n-3} \left( \frac{1}{n-2} \right)^{(n-1)/2} \\
&\quad \cdot (2\pi)^{(n-1)/2} \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2-2n+1}{2}} \\
&\quad + \sum_{l=n}^{m-1} \frac{n}{l+1} \frac{1}{n} (l+1)(n-1)^{n-3} \left( \frac{1}{n-2} \right)^{(n-1)/2} \\
&\quad \cdot (2\pi)^{(n-1)/2} \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2-2n+1}{2}} \\
&\quad \cdot \{n^3 + n^2(l+1) + n^2 \ln(l+1)\}
\end{aligned}$$

$$\begin{aligned}
&\leq n^3 + \frac{1}{n}(n-1)^{n-3} \left(\frac{1}{n-2}\right)^{(n-1)/2} (2\pi)^{(n-1)/2} \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2-2n+1}{2}} \\
&\quad \cdot \left\{ \sum_{l=n}^{m-1} n^2 l + n^4 + n^3(l+1) + n^3 \ln(l+1) \right\} \\
&\leq n^3 + \frac{1}{n}(n-1)^{n-3} \left(\frac{1}{n-2}\right)^{(n-1)/2} (2\pi)^{(n-1)/2} \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2-2n+1}{2}} \\
&\quad \cdot \left\{ n^2 \frac{m(m-1)}{2} + mn^4 + n^3 \frac{m(m-1)}{2} + n^3 m(\ln(m)-1) \right\}. \quad (67)
\end{aligned}$$

Again, the third summand in brackets seems to dominate.

For the  $\Omega_n$ -distribution we obtain

$$\begin{aligned}
E_{m,n}^{k=0}(s) &\leq n^3 + n^2 \sum_{l=n}^{m-1} \left(\frac{l+2/(n+1)}{n-1}\right)^{(n-1)/(n+1)} \frac{1}{n} \frac{2}{\sqrt{\pi}} \\
&\quad \cdot \sqrt{\frac{n^2+1}{2}} [n+1]^{n-2+2/(n+1)} \sqrt{\frac{2\pi}{n}}^{n-1+2/(n-1)} \\
&\quad + \sum_{l=n}^{m-1} \frac{n}{l+1} \frac{1}{n} \left(\frac{l+1+2/(n+1)}{n-1}\right)^{(n-1)/(n+1)} \\
&\quad \cdot \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2+1}{2}} (n+1)^{n-2+2/(n+1)} \sqrt{\frac{2\pi}{n}}^{n-1+2/(n-1)} \\
&\quad \cdot \{n^3 + n^2(l+1) + n^2 \ln(l+1)\} \\
&\leq n^3 + \frac{1}{n} \frac{2}{\sqrt{\pi}} \sqrt{\frac{n^2+1}{2}} (n+1)^{n-2+2/(n+1)} \sqrt{\frac{2\pi}{n}}^{n-1+2/(n-1)} (n-1)^{-(n-1)/(n+1)} \\
&\quad \cdot \left\{ n^2 \left(m + \frac{2}{n+1}\right)^{1+(n-1)/(n+1)} \frac{2n}{n+1} \right. \\
&\quad + n^4 \left(m + \frac{2}{n+1}\right)^{(n-1)/(n+1)} \frac{n+1}{n-1} \\
&\quad + n^3 \left(m + \frac{2}{n+1}\right)^{1+(n-1)/(n+1)} \frac{2n}{n+1} \\
&\quad \left. + n^3 \left(m + \frac{2}{n+1}\right)^{(n-1)/(n+1)} \frac{n+1}{n-1} \cdot \left(\ln m - \frac{n+1}{n-1}\right) \right\}. \quad (68)
\end{aligned}$$

Here again, the third summand in brackets is dominating.

## 7. Conclusions

Let us compare this outcome with the result for Gift-Wrapping (analyzed in [7]). This was a sequential algorithm, which worked as follows for the primal problem:

After having one vertex available, perform Simplex-Steps to adjacent vertices and note the incident edges. If the adjacent vertex is already known, declare the connecting edge as saturated.

Try to leave the vertex via an unsaturated edge (which leads to a new vertex) and repeat the same at the new vertex. The edge will now be saturated. If there is no unsaturated edge to leave the vertex, then perform a reverse pivot step along the Simplex-Path, as being used so far, until you meet a vertex with at least one unsaturated edge.

For each new vertex we have to determine all edges and to decide whether they are saturated or unsaturated. We have to store that information for access at all time. The drawback of this algorithm is the need to store the Simplex-Path and to keep track over all edges found so far.

The complexity amounted to a formula

$$O(\#(V_m) + n)[mn + nm + nn \ln m]$$

(the first summand in the second bracket for the pivot steps, the second for edges/quotients and the last for storing the necessary information).

The expected complexity amounted to

$$E_{m,n}(\#(V_m)) \cdot O(nm + n^2 \ln m). \quad (69)$$

This should be compared with our result derived before (38), (65), which amounts to

$$n^3 + n^2 \cdot \sum_{l=n}^{m-1} E_{l,n}(\#(V_l)) + \sum_{l=n}^{m-1} \frac{n}{l+1} E_{l,n}(\#(V_l)) \{n^3 + n^2(l+1) + n^2 \ln(l+1)\}. \quad (70)$$

Crucial for that comparison is the behaviour in  $l$  and the need to sum up  $m - 1$  stages. This is a drawback for our incremental algorithm. This drawback becomes more and more dramatic when the growth of  $E_{l,n}(\#(V_l))$  gets slower (with respect to  $l$ ), as we shall see.

In all the distributions from our family, we have proven upper bounds on the behaviour of  $E_{l,n}(\#(V))$  in  $l$  of the kind  $(l + (2 + 2k)/(n + 1 + 2k))^{(n-1)/(n+1+2k)}$ , that means that the behaviour is extremely sublinear for large  $k$ . At the borderline, we have the  $\omega_n$ -distribution with  $k \rightarrow -1$  and almost linear behaviour as  $l \cdot C_{n,k} = l \cdot C_{n,-1}$ .

For the asymptotic case ( $l \rightarrow \infty$ ,  $n$  and  $k$  fixed) it can be shown that there are also lower bounds on the behaviour in  $l$  of the same order. We have discussed this only for the special uniform distributions on  $\omega_n$  and on  $\Omega_n$ . A proof for the case of general  $k$ 's would extend this paper too much. Even for the factors reflecting the behaviour in  $n$  we got precision on  $\omega_n$  and on  $\Omega_n$  and obviously a "slight" overestimation for the general case. If we rely on our upper bounds, interpreting them as the "true" behaviour, then a comparison gives interesting insights.

For the  $\omega_n$ -distribution, summing up leads to a quadratic term

$$\sum_{l=n}^{m-1} l C_{n,-1} \approx \frac{m(m-1)}{2} C_{n,-1}$$

combined with a factor  $n^2$ . In the second sum (over the vertices of the difference set) a linear term in  $m$  is combined with a factor  $n^4$ . In Gift-Wrapping that  $m^2$ -term was only combined with a factor  $n$ . So, even in the most harmless case, we have to spend  $n$  times more effort. This is the most harmless case, because for other distributions the growth in  $l$  will be sublinear.

The example of the  $\omega_n$ -distribution already explains very well why the idea of incremental solution does not pay per se. Here we have seen that the true behaviour of the expected number of vertices, seen as a function of  $m$ , approximates a linear function of  $m$  from below with increasing precision for growing  $m$ . So take every number given in the following paragraphs as “approximately”.

For  $\omega_n$ , in each stage (e.g. stage  $l + 1$ ), the expected number of vertices grows from  $lC_{n,-1}$  to  $(l + 1)C_{n,-1}$ , which means that after that we have  $C_{n,-1}$  vertices more. However, the truth is that among the  $(l + 1)C_{n,-1}$  vertices now available, we have even  $(l + 1)C_{n,-1}(n/(l + 1)) = nC_{n,-1}$  completely new vertices (of the  $x_0$ -kind, where  $a_{l+1}$  is active). To keep the balance (we have a total increment of only  $C_{n,-1}$  in stage  $l + 1$ ), an expected number of  $(n - 1)C_{n,-1}$  vertices had to be deleted in that stage (i.e. for each  $l + 1$ ). So all our work devoted to these  $(n - 1)C_{n,-1}$  deleted vertices (and  $(m - n)(n - 1)C_{n,-1}$  vertices summarized over all stages) was completely in vain.

Finally, we calculate the overproduction quotient for the  $\omega_n$ -distribution:

$$\begin{aligned} & \frac{E(\text{newly produced vertices})}{E(\text{net increment in the number of vertices})} \\ & \equiv \frac{E_{m,n}(\#(V_{l+1} \setminus V_l))}{E_{m,n}(\#(V_{l+1})) - E_{m,n}(\#(V_l))} = n. \end{aligned} \quad (71)$$

For general  $k$ -distributions this is even worse:

In stage  $l + 1$  an expected set of

$$(n/(l + 1))(l + 1 + (2 + 2k)/(n + 1 + 2k))^{(n-1)/(n+1+2k)} C_{n,k}$$

vertices is new, and the total expected number grows from

$$(l + (2 + 2k)/(n + 1 + 2k))^{(n-1)/(n+1+2k)} C_{n,k}$$

to

$$(l + 1 + (2 + 2k)/(n + 1 + 2k))^{(n-1)/(n+1+2k)} C_{n,k}.$$

This means that the overproduction factor amounts to

$$\begin{aligned} & \frac{(n/(l + 1))(l + 1 + (2 + 2k)/(n + 1 + 2k))^{(n-1)/(n+1+2k)} C_{n,k}}{(l+1+(2+2k)/(n+1+2k))^{(n-1)/(n+1+2k)} C_{n,k} - (l+(2+2k)/(n+1+2k))^{(n-1)/(n+1+2k)} C_{n,k}} \\ & = \frac{n/(l + 1)}{1 - (1 - 1/(l + 1 + (2 + 2k)/(n + 1 + 2k)))^{(n-1)/(n+1+2k)}}. \end{aligned}$$

The denominator is of the type  $1 - (1 - 1/(l + 1 + \kappa(k)))^{1-\kappa(k)}$ , with  $\kappa(k) := (2 + 2k)/(n + 1 + 2k)$  being an increasing function of  $k$ . Note that an increasing  $k$  increases  $\kappa$  and so the term in brackets increases, too. Simultaneously, the exponent decreases, while the bracket-value is between 0 and 1. This leads to an increment of that term. However, that

increment makes the global quotient larger. This global quotient becomes exactly  $n$  for  $k \rightarrow -1$  and it gets larger for growing  $k$ . Hence the overproduction factor or quotient becomes higher for larger values of  $k$ .

So, in the case of our Rotation-Symmetry Model (with absolute emphasis on nondegenerate problems) incremental algorithms lose the competition on the average behaviour against pivoting algorithms.

In degenerate problems, this advantage may get totally reversed, because there a certain danger of overproduction arises on the side of the pivoting algorithm. Since here one vertex may be associated with many bases, we observe a (perhaps) dramatic overproduction of bases. Because the pivoting algorithm actually has to generate a walk over bases (rather than over vertices) this will deteriorate its behaviour significantly.

The advantage may turn around even under nondegeneracy, if additional information about the vectors  $a_i$  is exploited (as for instance in Quickhull), where the  $a_i$ 's are sorted in such a way that with high probability the extremals of these  $a_i$ 's are among the first in the queue. Then it may be achievable that in a very early stage  $X$  is almost ready or in its final shape (all the vertices of  $X$  are already known, and very few are still to be deleted). However, from the view of a probabilistic analysis, such a change in the order of handling the vectors would bring dependencies between the  $a_i$ 's into the game. So far, we are not able to handle these dependencies sufficiently.

## References

1. Avis, D., Bremner, D., and Seidel, R.: How Good are Convex Hull Algorithms? *Computational Geometry* **7**, 265–301, 1997.
2. Avis, D., and Fukuda, K.: A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra, *Discrete & Computational Geometry* **8**, 295–313, 1992.
3. Barber, C.D., Dobkin, D., and Huhdanpaa, H.: The Quickhull Algorithm for Convex Hull, *ACM Transactions on Mathematical Software* **22**(4), 469–483, 1996.
4. Borgwardt, K.H.: Untersuchungen zur Asymptotik der mittleren Schrittzahl von Simplexverfahren in der linearen Optimierung, Dissertation, Universität Kaiserslautern, 1977.
5. Borgwardt, K.H.: Some Distribution-Independent Results About the Asymptotic Order of the Average Number of Pivot Steps of the Simplex Method, *Mathematics of Operations Research* **7**, 441–462, 1982.
6. Borgwardt, K.H.: *The Simplex Method, A Probabilistic Analysis*, Springer-Verlag, Heidelberg, 1987.
7. Borgwardt, K.H.: Average Complexity of a Gift-Wrapping Algorithm for Determining the Convex Hull of Randomly Given Points, *Discrete & Computational Geometry* **17**, 79–109, 1997.
8. Borgwardt, K.H.: A Sharp Upper Bound for the Expected Number of Shadow-Vertices in LP-Polyhedra Under Orthogonal Projection on Two-Dimensional Planes, *Mathematics of Operations Research* **24**(3), 544–603, 1999, and **24**(4), 925–984, 1999.
9. Bremner, D.: Incremental Convex Hull Algorithms Are Not Output Sensitive, *Discrete & Computational Geometry* **21**, 57–68, 1999.
10. Bremner, D., Fukuda, K., and Marzetta, A.: Primal–Dual Methods for Vertex and Facet Enumeration, *Discrete & Computational Geometry* **20**, 333–357, 1998.
11. Buchta, C.: Zufällige Polyeder: Eine Übersicht, in Hlawka, Ed., ed., *Zahlentheoretische Analysis*, Springer-Verlag, Berlin, pp. 1–13, 1985.
12. Buchta, C., and Müller, J.: Random Polytopes in a Ball, *Journal of Applied Probability* **21**(4), 753–762, 1984.
13. Buchta, C., Müller, F., and Tichy, R.F.: Stochastic Approximations of Convex Bodies, *Mathematische Annalen* **271**(2), 225–235, 1985.
14. Chand, P.L., and Kapur, S.S.: An Algorithm for Convex Polytopes, *Journal of the Association for Computing Machinery* **17**(1), 78–86, 1970.

15. Charnes, A., Cooper, W.W., and Henderson, A.: *An Introduction to Linear Programming*, Wiley, New York, 1953.
16. Clarkson, K., Mehlhorn, K., and Seidel, R.: Four Results on Randomized Incremental Constructions, *Computational Geometry* **3**(4), 185–212, 1993.
17. Dwyer, R.A.: Average-Case Analysis of Algorithms for Convex Hulls and Voronoi Diagrams, Thesis, Computer Science Department, Carnegie Mellon University, 1988.
18. Dyer, M.E.: The Complexity of Vertex Enumeration Methods, *Mathematics of Operations Research* **8**, 381–402, 1983.
19. Edelsbrunner, H.: *Algorithms in Combinatorial Geometry*, Springer-Verlag, New York, 1987.
20. Efron, B.: The Convex Hull of a Random Set of Points, *Biometrika* **52**(3–4), 331–345, 1965.
21. Fukuda, K., and Prodon, A.: Double Description Method Revisited, in Deza, M., Euler, R., and Mousakakis, I., eds., *Combinatorics and Computer Science*, volume 1120 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 91–111, 1996.
22. Goodman, J.E., and Rourke, J.O. (eds.): *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, FL, 1997.
23. Grünbaum, B.: *Convex Polytopes*, Wiley, New York, 1997.
24. Huhn, P., and Borgwardt, K.H.: Interior-Point-Methods: Worst-Case and Average-Case Analysis of a Phase-I-Algorithm and a Termination Procedure, *Journal of Complexity* **18**, 833–910, 2002.
25. Joswig, M.: Beneath-and-Beyond Revisited, in Joswig, M., et al. (eds.), *Algebra, Geometry, and Software Systems*, Berlin, pp. 1–21, 2003.
26. Mulmuley, K.: *Computational Geometry, An Introduction Through Randomized Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
27. Motzkin, T.S., Raiffa, H., Thompson, G.L., and Thrall, R.M.: The Double Description Method, in Kuhn, H.W., and Tucker, A.W., eds., *Contributions to the Theory of Games*, Vol. 2, Princeton University Press, Princeton, RI, pp. 51–73, 1953.
28. Preparata, F.P., and Shamos, M.I.: *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
29. Raynaud, H.: Sur le Comportement Asymptotique de l'Enveloppe Convexe d'un Nuage des Points Tirés au Hasard dans  $\mathbb{R}^n$ , *Comptes Rendus des Séances de l'Académie des Sciences Paris* **261**, 627–629, 1965.
30. Schneider, R., and Wieacker, J.A.: Random Polytopes in a Convex Body, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **52**, 69–73, 1980.
31. Seidel, R.: Constructing Higher-Dimensional Convex Hulls at Logarithmic Cost per Face, in *Proceedings of the 18th ACM Symposium on Theory of Computing*, pp. 404–413, 1986.
32. Seidel, R.: Small-Dimensional Linear Programming and Convex Hulls Made Easy, *Discrete & Computational Geometry* **6**, 423–434, 1991.
33. Swart, G.: Finding the Convex Hull Facet by Facet, *Journal of Algorithms* **6**, 17–48, 1985.

Received October 25, 2004, and in revised form March 20, 2006, and April 7, 2006.

Online publication February 9, 2007.