SERIE B — INFORMATIK

Algorithms for Ham-Sandwich Cuts◇

Chi-Yuan Lo*
Jiří Matoušek**
William Steiger***

B 92–20
September 1992

Abstract

Given disjoint sets $P_1, P_2, \ldots, P_d$ in R^d with n points in total, a *ham-sandwich cut* is a hyperplane that simultaneously bisects the P_i. We present algorithms for finding ham-sandwich cuts in every dimension $d > 1$. When $d = 2$, the algorithm is optimal, having complexity $O(n)$. For dimension $d > 2$, the bound on the running time is proportional to the worst-case time needed for constructing a level in an arrangement of n hyperplanes in dimension $d-1$. This, in turn, is related to the number of k-sets in R^{d-1}. With the current estimates, we get complexity close to $O(n^{3/2})$ for $d = 3$, roughly $O(n^{8/3})$ for $d = 4$ and $O(n^{d-1-a(d)})$ for some $a(d) > 0$ (going to zero as d increases) for larger d. We also give a linear time algorithm for ham-sandwich cuts in R^3 when the three sets are suitably separated.

◇A preliminary version of the results of this paper appeared in [16] and [17]. Part of this research by J.M. was done while he was visiting the School of Mathematics, Georgia Institute of Technology, Atlanta, and part of his work on this paper was supported by Humboldt Research Fellowship. W.S. expresses gratitude to the NSF DIMACS Center at Rutgers, and his research was supported in part by NSF grants CCR-8902522 and CCR-9111491.
*AT&T Bell Laboratories, Murray Hill, NJ 07974, USA
**Department of Applied Mathematics, Charles University, Malostranské nám. 25, 118 00 Praha 1, Czechoslovakia, and Free University of Berlin, Arnimallee 2–6, W-1000 Berlin 33, Germany.
***Rutgers University, Dept. of Computer Science, New Brunswick, NJ 08903, USA.

1 Introduction and Summary

A hyperplane h is said to *bisect* a set P of n points in R^d if no more than $n/2$ points of P lie in either of the open halfspaces defined by h. It is no loss of generality to assume n odd since otherwise we may delete any point, x, and observe that any hyperplane that bisects $P \setminus \{x\}$ also bisects P.

If P is a disjoint union of d sets $P_1, \ldots, P_d$, a ham-sandwich cut is a hyperplane that simultaneously bisects all the P_i. The ham-sandwich theorem (see for example [12]) guarantees the existence of such a cut. Here we focus on the algorithmic question, which asks for efficient procedures for computing a cut, and for bounds on the complexity of this task. Throughout, we use a model of computation where any arithmetic operation or comparison is charged unit cost (the *real RAM* model).

In two dimensions, a ham-sandwich cut is a line h that bisects P_1 and P_2. For *the linearly separated* case, where the convex hulls of P_1 and P_2 do not intersect, Megiddo [21] gave an algorithm to compute h that runs in $O(n)$ steps. Megiddo's algorithm gives an optimal solution to a partitioning problem posed by Willard [23], namely to find lines ℓ_1 and ℓ_2 that separate n given points into "quadrants" containing at most n/4 points each. The first line may be any (say horizontal) line ℓ_1 partitioning the points evenly, easily obtained in $O(n)$ steps. The second line is a ham-sandwich cut for the points P_1 (above ℓ_1) and P_2 (below ℓ_1), obtained in linear time by Megiddo's algorithm.

Edelsbrunner and Waupotitsch [13] modified Megiddo's method for the general planar case. Their algorithm can compute h in time $O(n \log n)$. Earlier, Cole, Sharir and Yap [9] had described a procedure that may now be seen to have the same complexity, in view of the existence of a logarithmic depth sorting network [2].

In this paper we prove the following result (see also [16]).

Proposition 1 *Given two sets of points P_1 and P_2 in R^2, $|P_1| + |P_2| = n$, a ham-sandwich cut can be computed in $O(n)$ time.*

The proof consists of an optimal linear time algorithm which thus settles the complexity question for two dimensional ham sandwich cuts.

In three and higher dimensions much less was known. The brute-force approach has complexity $O(n^{d+1})$; the odd cardinality assumption forces a cut to contain a point from each P_i, and we can check the hyperplane corresponding to each possible d-tuple in linear time. It is also not too difficult to give an $O(n^d)$ algorithm, by constructing the arrangements of hyperplanes dual to the points of P (see Section 2 for the dual formulation of the problem).

Edelsbrunner [14] described a related problem of finding two planes that simultaneously divide each of two given sets of points in R^3 into four equal sized subsets; the points were required to satisfy a special separation condition. He gives an algorithm with running time $O(t(n)(\log n)^2)$, where $t(n)$ denotes the maximal number of $(n/2)$-sets possessed by any set of n points in R^3 (see also section 2).

In Section 4 we show how to generalize the ideas used in Proposition 1 to dimension $d > 2$ and describe an algorithm with complexity $O(n^{d-1})$. The running time can be further decreased using (relatively complicated) ray shooting methods for construction of levels in hyperplane arrangements. We prove the following.

Proposition 2 *Given n points in R^d which are partitioned into d sets $P_1, \ldots, P_d$ in R^d, a ham-sandwich cut can be computed in time proportional to the (worst-case) time needed to construct a given level in the arrangement of n given hyperplanes in R^{d-1}. The latter problem (i) requires at least $\Omega(n^{d-2})$ time; (ii) is easy to solve in $O(n^{d-1})$ time; (iii) can be solved within the following bounds:*

$O(n^{3/2} \log^2 n / \log^* n)$	*for $d = 3$,*
$O(n^{8/3+\varepsilon})$	*for $d = 4$,*
$O(n^{d-1-a(d)})$	*with certain (small) constant $a(d) > 0$ for $d \geq 5$.*

Finally for the case $d = 3$, if the sets are suitably separated, the general algorithm can be modified so that it finds a ham-sandwich cut in linear time. This extends Megiddo's result to R^3.

2 Preliminaries and Notation

We denote by S the coordinate hyperplane $x_d = 0$ (i.e. the x-axis for $d = 2$). For a subset $X \subseteq S$ we denote by $V(X)$ the vertical "cylinder" erected through X, i.e.

$$V(X) = \{(x_1, x_2, \ldots, x_d);\ x_d \in R,\ (x_1, \ldots, x_{d-1}, 0) \in X\}.$$

It is easier to look at a dual version of the ham-sandwich problem. We use the *duality transform* which maps the point $p = (x_1, \ldots, x_d)$ to the (nonvertical) hyperplane $\Pi = \{(w_1, \ldots, w_d) : w_d = 2x_1w_1 + \cdots + 2x_{d-1}w_{d-1} - x_d\}$ (see [12] for properties). The ham-sandwich cut problem then becomes the following:

> Given a set H of hyperplanes in R^d, partitioned into d classes $H_1, \ldots, H_d$, $|H_i|$ odd, find a point x which, for each $i = 1, \ldots, d$, has no more than $|H_i|/2$ of the hyperplanes of H_i below it, and no more than $|H_i|/2$ hyperplanes above.

To simplify our considerations, we make some general position assumptions. We suppose that every d-tuple of hyperplanes of H meets in a unique point (vertex) and that no point in R^d is incident with more than d of the hyperplanes. Also we assume that the vertical direction (the direction of the x_d-axis) is a "generic" one, i.e. that the vertical projections of all vertices on the coordinate hyperplane $x_d = 0$ are all distinct. This is no loss of generality, as one may use some variant of *simulation of simplicity* (see [12]) to handle the general case.

Given a set H of hyperplanes in R^d, they partition the space into a complex of convex cells, called the *arrangement* of H. An important concept for us will be the *p-level* in the arrangement of H, denoted by $L_p(H)$. This is defined as the closure of the set of all points which lie on a unique hyperplane of the arrangement and have exactly $p-1$ hyperplanes below it. In dimension 2, the p-level is a continuous, piecewise linear function whose segments always coincide with one of the lines in the arrangement. In higher dimensions, the p-level also consists of certain cells of the arrangement of H, and thus it is a piecewise linear hypersurface in R^d.

When $p = \lfloor (|H|+1)/2 \rfloor$, the $L_p(H)$ is called the *median-level* of the arrangement. The dual version of the ham-sandwich cut problem may be restated as follows:

> Given a set $H = \{\pi_1, \ldots, \pi_n\}$ of hyperplanes in R^d, partitioned into d classes $H_1, \ldots, H_d$, $|H_i|$ odd, find an intersection point of the median levels of the arrangements of $H_1, \ldots, H_d$.

Such an intersection point will be a vertex in the arrangement of H, whose d defining hyperplanes contain precisely one hyperplane of each H_i.

A key feature used by our algorithms is the *odd intersection property.* A set $X \subset S = \{(x_1, \ldots, x_{d-1}, 0)\}$ has the odd intersection property with respect to levels $\lambda_i \equiv L_{p_i}(H_i)$ if

$$|(\lambda_1 \cap \cdots \cap \lambda_d) \cap V(X)| \text{ is odd;} \tag{2.1}$$

i.e., the levels intersect an odd number of times in the cylinder erected through X (note that the set $\lambda_1 \cap \ldots \cap \lambda_d$ is finite by our general position assumption).

The running time of our algorithm will depend on the time needed for construction of levels in arrangements of hyperplanes; this time in turn depends on the combinatorial complexity of these levels. We review the known results:

Let $e_d(n,k)$ denote the maximum possible number of vertices of the k-level in an arrangement of n hyperplanes in R^d, and let $e_d(n) = \max\{e_d(n,k);\ k = 1, \ldots, n\}$. It is well-known that $e_d(n,k)$ is proportional to the maximum number of k-sets of an n point set in R^d. The k-set problem has been extensively studied (see [7], [10], and [12]).

It is known that $e_d(n) = \Omega(n^{d-1}\log n)$ and it is conjectured that this bound is close to the truth. The known upper bounds seem much weaker, however. It was shown that $e_2(n) = O(n^{3/2}/\log^* n)$ [22], that $e_3(n) = O(n^{8/3})$ [5], [10], and in general $e_d(n) = O(n^{d-\delta(d)})$ for some (small) positive constant $\delta(d)$ [4], [24].

Efficient output-sensitive algorithms for level construction are known in dimensions 2 and 3: a level of complexity b can be constructed in time $O(n \log n + b \log^2 n)$ for $d = 2$ [15] and in time $O(n^{1+\varepsilon} + bn^{\varepsilon})$ for $d = 3$, ε an arbitrarily small positive constant [1]. For $d > 3$ the efficiency of the algorithm of [1] gets worse; it guarantees that if the complexity of the level is $O(n^{d-\delta(d)})$ for some $\delta(d) > 0$, then the level can be constructed in $O(n^{d-\delta(d)\frac{d}{d+1}+\varepsilon})$ time.

3 The Planar Case

To elucidate the ideas used in our algorithms, we begin by explaining the planar case, and then show how these ideas may be extended to higher dimensions. To prove Proposition 1, namely that the planar ham-sandwich problem has linear complexity, we will present an algorithm for the task using the dual setting introduced in the previous section. Therefore we have two sets H_1 and H_2 of lines, and we want to find an intersection of the median levels μ_1 (of the lines in H_1) and μ_2. We suppose that both $n_1 = |H_1|$ and $n_2 = |H_2|$ are odd, and $n = n_1 + n_2$.

In this situation and with our general position assumptions we have

Lemma 3.1 *The median level of H_1 and the median level of H_2 intersect in an odd number of points.*

Proof: This conclusion may be deduced from a well-known elementary proof of the existence of the ham-sandwich cut in the plane. Here we give an elementary geometric proof. First we observe that the left unbounded ray and the right unbounded ray of the median level of H_1 lies on the same line $h_1 \in H_1$ (the one with the median slope). Similarly the unbounded rays of the median level of H_2 are parts of some line $h_2 \in H_2$. One of these lines, say h_1, has smaller slope. This means that far enough left, the median level of H_1 is below the median level of H_2, while far to the right, it is above the median level of H_2. By continuity, the median levels intersect an odd number of times. □

Remark. The lemma says that the whole x-axis has the odd intersection property with respect to the median levels of $H_1, \ldots, H_d$. In general, let $\lambda_i = L_{p_i}(H_i)$ denote the p_i level in the arrangement of the lines in H_i. Then an interval $T = (\ell, r)$ has the odd intersection property with respect to λ_1, λ_2 if and only if

$$(\lambda_1(\ell) - \lambda_2(\ell))(\lambda_1(r) - \lambda_2(r)) < 0, \tag{3.1}$$

where $(t, \lambda(t))$ denotes the point on the level λ at $x = t$.

Our algorithm will work in phases, and it will discard a constant fraction of the lines in each phase, until it reaches a situation with a small (constant) number of lines, where the ham-sandwich cut vertex can be found directly. At the beginning of each phase, the algorithm has the following data:

- an open interval T on the x-axis,
- current sets G_1, G_2 of lines, $G_i \subseteq H_i$, $|G_i| = m_i$,
- integers p_1, p_2, $1 \leq p_i \leq m_i$,

and the following invariant holds:

> The levels $\lambda_1 = L_{p_1}(G_1)$ and $\lambda_2 = L_{p_2}(G_2)$ have an odd number of intersections within $V(T)$, and each such intersection is an intersection of the median levels of the original sets H_1, H_2 of lines[1].

At the end of the phase, lines have been discarded so we now have new sets $G'_i \subseteq G_i$, $|G'_i| = m'_i$, integers $p'_i \leq m'_i$, and a new interval $T' \subset T$ on which the invariant holds

[1] In fact these intersections are *the only* ham-sandwich vertices in $V(T)$.

for the new data. To start the algorithm, T is the whole x-axis, $G_1 = H_1$, $G_2 = H_2$, $p_1 = \lfloor (n_1+1)/2 \rfloor$ and $p_2 = \lfloor (n_2+1)/2 \rfloor$. The invariant will be satisfied in view of Lemma 3.1.

Clearly the assertion of Proposition 1 holds if we can prove

Lemma 3.2 *Let T, G_1, G_2, p_1, p_2 be as above and satisfy the conditions of the invariant. Then in time $O(m_1+m_2)$, one can compute new $T' \subset T$, $G_1' \subseteq G_1$, $G_2' \subseteq G_2$, $p_1' \leq p_1$, $p_2' \leq p_2$, again satisfying the conditions of the invariant, and with the new value of $m' = |G_1'| + |G_2'| \leq 3m/4$; i.e., in linear time at least a quarter of the lines $G_1 \cup G_2$ that begin a phase may be discarded as candidates for ham-sandwich vertices.*

Proof: We first give an outline of the algorithm, and then we fill in the details. We suppose that $m_1 \geq m_2$ (renumbering the sets if necessary). The algorithm performs the following steps [the time for each step is indicated in square brackets]:

1. Divide the interval T into a constant number of subintervals $T_1, \ldots, T_C$, such that no $V(T_i)$ contains more than a prescribed (constant) fraction of the vertices of the arrangement of G_1 [$O(m_1)$].

2. Find one subinterval T_i with the odd intersection property [$O(m_1+m_2)$] .

3. Construct a trapezoid $\tau_i \subset V(T_i)$, such that

$$\lambda_1 \cap V(T_i) \subset \tau_i \tag{3.2}$$

$$\text{At most half of the lines of } G_1 \text{ intersect } \tau_i \tag{3.3}$$

   [$O(m_1)$].

4. Discard all the lines of G_1 which do not intersect τ_i (at least $m_1/2 \geq (m_1 + m_2)/4$ lines), and update p_1 accordingly ($p_1' \leftarrow p_1 - b$, b denoting the number of discarded lines of G_1 lying completely below τ_i). Then T_i becomes the new T, and we are ready for the next phase of the algorithm [$O(m_1+m_2)$].

Now we discuss the steps in greater details. The first result pertains to Step 1.

Lemma 3.3 *Let H be a set of n lines in the plane in general position, $\alpha < 1$ a prescribed positive constant, and T an interval on the x-axis. In $O(n)$ time, one can subdivide T into subintervals $T_1, T_2, \ldots, T_C$ ($C = C(\alpha)$ a constant), such that each $V(T_i)$ contains the at most αN of the $N = \binom{n}{2}$ vertices of the arrangement of H.*

Proof: We apply a theorem of [18] on approximate selection of the k-th leftmost intersection (which in turn uses a technique developed in [8]). Let $t_1 < \cdots < t_N$ denote the x-coordinates of the vertices of H, in order. It is proved in [18] that given a positive constant $\nu < 1$ and a number $k, 1 \leq k \leq N$, then in linear time one can find two lines of H whose intersection lies between $t_{k-\nu N}$ and $t_{k+\nu N}$. Using this selection procedure, we divide the x-axis into intervals guaranteed to contain no more than αN intersections each, as follows. Taking $\nu = \frac{\alpha}{5}$ and $k = \lceil i\frac{\alpha}{2}N \rceil$ we get, in linear time, an intersection with x-coordinate u_i that lies between $t_{\lceil (i-\frac{1}{2})\frac{\alpha}{2}N \rceil}$ and $t_{\lceil (i+\frac{1}{2})\frac{\alpha}{2}N \rceil}$. Carrying out such approximations for $i = 1, \ldots, \lfloor \frac{2}{\alpha} \rfloor$ we obtain intervals $T'_i = (u_{i-1}, u_i)$ and the non-empty intervals $T_i = T \cap T_i$ have the asserted properties (note that $C(\alpha) \leq 2/\alpha$). □

At the end of this section we will discuss more practical aspects of the algorithm and there we suggest another approach for constructing the subdivision in Step 1. A third possibility is to specialize the construction we use when subdividing in higher dimensions.

Lemma 3.3 shows how to do Step 1 in linear time. We will apply it to the m_1 lines in G_1. The value of α will be fixed later. For Step 2 (subinterval selection), we need the following lemma:

Lemma 3.4 *Given an interval $T = (\ell, r)$, the odd-intersection property for levels $\lambda_1 = L_{p_1}(G_1)$ and $\lambda_2 = L_{p_2}(G_2)$ may be tested in linear time; i.e., in time $O(|G_1| + |G_2|)$ we can find the parity of $|\lambda_1 \cap \lambda_2 \cap V(T)|$.*

Proof: The parity is odd iff the vertical order of the intersections of λ_1 and λ_2 with the line $x = \ell$ is opposite to the order of the intersections with $x = r$; i.e., (3.1) must hold on $T = (\ell, r)$. The intersection of the p_1-level with a vertical line $x = v$ can be found in $O(m_1)$ time, by computing the y-coordinates of the intersections of all lines of G_1 with $x = v$ and selecting the p_1-th smallest of these numbers, using a linear-time selection algorithm. □

Since T has the odd intersection property, so will at least one of the subintervals T_j from the subdivision. Testing them sequentially, we are guaranteed by Lemma 3.4 that in linear time we will discover a suitable subinterval $T_i = (\ell, r)$ with the odd intersection property. We now describe the construction of the trapezoid τ_i mentioned in Step 3 and verify its properties. Let D_l^- and D_l^+ be the intersections of the vertical line $x = \ell$ with the levels $L_{p_1 - \varepsilon m_1}(G_1)$ and $L_{p_1 + \varepsilon m_1}(G_1)$, respectively; similarly we define D_r^-, D_r^+. These four points define the trapezoid $\tau_i = D_l^- D_l^+ D_r^+ D_r^-$. With appropriate choice of ε it has the desired properties in view of

Lemma 3.5 *Let $\varepsilon = \frac{1}{8}$ and $\alpha = \frac{1}{32}$. Then (3.2) and (3.3) hold for any τ_i; i.e., at most half of the lines in G_1 meet τ_i, and within the strip $V(T_i)$, the level $L_{p_1}(G_1)$ remains within τ_i.*

Proof: The proof very much resembles the proof of a similar lemma in [18]. Consider the top of τ_i, the segment $\sigma = D_l^+ D_r^+$. The lines of G_1 that meet σ are partitioned into two classes, $\mathcal{S}$, the lines with slope smaller than that of σ, and $\mathcal{L}$, those with larger slope. Traversing σ from left to right, we keep count of the number of G_1 lines below. At the start, there are $p_1 + \varepsilon m_1$ lines below. When we meet a line in $\mathcal{S}$, the count increases by one, and when we meet an $\mathcal{L}$-line, it decreases by one. At the end there are again $p_1 + \varepsilon m_1$ lines below. Hence $|\mathcal{S}| = |\mathcal{L}|$.

Each $\mathcal{S}$-line intersects each $\mathcal{L}$-line within the vertical strip $V(T_i)$. Since this strip contains at most $\alpha\binom{m_1}{2} < \alpha m_1^2/2$ intersections, by the construction in Lemma 3.3, we have $|\mathcal{S}|^2 = |\mathcal{S}||\mathcal{L}| < \alpha m_1^2/2$, so $|\mathcal{S}| = |\mathcal{L}| < (\sqrt{\alpha/2})m_1$. Since σ is εm_1 lines above the p_1-level at both endpoints of the interval T_i, the p_1-level remains below σ as long as

$$\sqrt{\frac{\alpha}{2}} \leq \varepsilon.$$

The same argument will show that the p_1-level never breaks below the bottom of τ_i.

Now we count intersections of G_1 lines with the boundary of τ_i. There are exactly $2\varepsilon m_1$ such intersections on each of the vertical sides $D_\ell^- D_\ell^+$ and $D_r^- D_r^+$, by definition. Also, we have shown that at most $2(\sqrt{\alpha/2})m_1$ lines of G_1 meet the top side of τ_i; similarly the bottom side contributes the at most $2(\sqrt{\alpha/2})m_1$ intersections. The total is at most $4\varepsilon m_1 + 4(\sqrt{\alpha/2})m_1$ intersections which, using the above inequality, is less than $8\varepsilon m_1$. Since each G_1 line that meets τ_i intersects two sides, at most $4\varepsilon m_1$ lines can meet any trapezoid. So if $\varepsilon = \frac{1}{8}$ at least half the lines in G_1 miss τ_i as required by (3.3). If we now take $\alpha = \frac{1}{32}$, (3.2) is satisfied because the inequality, above, is. This finishes the proof of Lemma 3.5 and thus of Proposition 1 as well. □

We conclude this section by a remark concerning a practical implementation of the planar ham-sandwich cut algorithm. There are $\frac{2}{\alpha} = 64$ subintervals in the subdivision. In practice, it is wasteful to construct all of them and test them for the odd intersection property sequentially (although the asymptotic complexity is not affected). Instead, one may perform a binary search: start with T as the current interval, and select an intersection approximately in the middle among the intersections of the G_1-lines in the current interval. Subdivide the current interval into two subintervals by the selected intersection. At least one of them has the odd intersection property (one application of Lemma 3.4 suffices to determine which one) and it can be used as the current interval in the next step. This 'halving' is repeated until the number of intersections within the current interval becomes small enough, then one constructs τ_i and discards the G_1-lines, as described above. A relatively easy way to select an intersection approximately in the middle of the current interval is to choose a random intersection within that interval. For this purpose, one can use a modification of an algorithm for counting inversions of a permutation (or its approximate version, if one wants to stay within the asserted asymptotically linear time), see [8], [20] or [11]. With these modifications, the algorithm becomes relatively simple and (hopefully) practical.

4 The General Case

In this section we describe a generalization of the algorithm for an arbitrary fixed dimension, and prove the complexity assertions made in Proposition 2.

Proof of Proposition 2: The presentation is quite analogous to the one for the planar case. Let μ_i denote the median level of H_i. Let us call every point of $\mu_1 \cap \ldots \cap \mu_d$ a *ham-sandwich vertex* (with our general position assumptions, there are finitely many points in the intersection, each being a vertex of the arrangement of $H = H_1 \cup \ldots \cup H_d$).

We begin with an analog of Lemma 3.1 which shows that the odd intersection property (2.1) holds for the whole coordinate hyperplane S with respect to median levels.

Lemma 4.1 *The total number of ham-sandwich vertices is odd.*

Proof: This is, essentially, what one proves when establishing the existence of a ham-sandwich cut by topological arguments (from the Borsuk-Ulam theorem). A direct proof of showing the existence of odd number of ham-sandwich cuts along these lines was shown to us by I. Bárány. Here we give a somewhat different geometric proof, whose parts will also be useful later.

Using Lemma 3.1 as the base case in an induction we suppose the statement of Lemma 4.1 is true for dimensions smaller than d. Let $\mathcal{H}$ denote the arrangement of the n given hyperplanes. Consider the $N = \binom{n}{d-1}$ distinct 1-flats (lines) determined by $(d-1)$-tuples of hyperplanes of H and project each of these 1-flats vertically onto the coordinate hyperplane S (the general position assumption guarantees that no intersection projects to a point). Choose a unit vector $\delta \in S$, not orthogonal to any of the N projections. We can find two vertical hyperplanes π_{left} and π_{right}, both with normal δ such that *all* the vertices of $\mathcal{H}$ lie between them. By the choice of δ each of the given hyperplanes of H meets π_{left} and π_{right}. The intersections of the hyperplanes in $H_2, \ldots, H_d$ with π_{left} satisfy the induction hypothesis with $d-1$ and therefore $\mu_2 \cap \cdots \cap \mu_d$ meets π_{left} in an odd number of vertices (ham-sandwich vertices in π_{left}). Call them $\ell_1, \ldots, \ell_{2m+1}$. Similarly there are an odd number of $\mu_2 \cap \cdots \cap \mu_d$ vertices in π_{right}; call them $r_1, \ldots, r_{2k+1}$.

To complete the proof we describe the *skeleton*

$$\sigma = \mu_2 \cap \cdots \cap \mu_d,$$

the intersection of the median levels of the $H_2, \ldots, H_d$ hyperplanes. It consists of vertices connected by edges. A vertex is a point of the form

$$v = h_2 \cap \cdots \cap h_d \cap a' = (x_1, \ldots, x_d),$$

where $h_i \in H_i$ is in μ_i at $(x_1, \ldots, x_{d-1})$ and for some $q \in \{2, \ldots, d\}$, $a' \neq h_q$ is also in μ_q. The intersections

$$e = h_2 \cap \cdots \cap h_{q-1} \cap h_q \cap h_{q+1} \cap \cdots \cap h_d$$

and

$$e' = h_2 \cap \cdots \cap h_{q-1} \cap a' \cap h_{q+1} \cap \cdots \cap h_d$$

are both *edges* (1-flats) incident with v. The general position assumption guarantees that vertices have degree exactly two. Each vertex is in a connected component of σ which is either a chain $v_0, \ldots, v, \ldots, v_t$ of distinct vertices or a cycle $v_0, \ldots, v, \ldots, v_t$ of distinct vertices, except that $v_0 = v_t$. The terminal vertices v_0 and v_t in a chain are each incident with one edge which is an infinite halfline. If u and v are vertices on a chain and both above (w.r.t. x_d coordinate) or both below μ_1, the chain determines an *even* number of ham-sandwich cuts between u and v; otherwise it determines an odd number. Clearly cycles determine an even number of cuts.

Since all vertices of $\mathcal{H}$ are between π_{left} and π_{right}, no cycle of σ can meet either of these hyperplanes. On the other hand *both* terminal halflines of *each* chain must meet one of these hyperplanes, by the choice of δ. Thus each $\ell_i \in \pi_{left}$ and $r_j \in \pi_{right}$ is the intersection with a terminal halfline of some chain of σ.

In fact each ℓ_i is naturally matched with a unique r_j. Consider the line p containing the halfline meeting π_{left} at ℓ_i. The part of p to the left of π_{left} is in σ. Between π_{left} and π_{right}, p meets each of the $n-d+1$ hyperplanes in which it is not contained, and to the right of π_{right}, p has no vertices. Therefore the $n-d+1$ hyperplanes each reverse their "above/below" relation with p between π_{left} and π_{right}. This means that the part of p to the right of π_{right} is also in $\mu_2 \cap \cdots \cap \mu_d$, so it intersects π_{right} at some r_j. This establishes two facts: First, $2m+1$, the number of ℓ_j's, also equals the number of r_i's; second, amongst the ℓ_i's and r_j's, exactly half (or $2m+1$) are below μ_1. Now we are finished, because each chain has two terminal halflines that are either both above μ_1, both below it, or one of each. But since an odd number of the ℓ_i and r_j are below μ_1, an odd number of chains can have one terminal halfline above μ_1 and the other, below it, and this proves the lemma. □

Our algorithm uses simplices in the coordinate hyperplane S analogous to the interval T in the planar algorithm. It again works in phases, discarding a constant fraction of the hyperplanes in each phase.

At the beginning of each phase, the algorithm has the following data:

- an open simplex T in the coordinate hyperplane S,
- current sets $G_1, G_2, \ldots, G_d$ of hyperplanes, $G_i \subseteq H_i$, $|G_i| = m_i$, $m = m_1 + \cdots + m_d$,
- integers $p_1, p_2, \ldots, p_d$, $1 \le p_i \le m_i$.

The invariant is as follows:

> There are an odd number of intersections of $\lambda_1 \cap \ldots \lambda_d$ ($\lambda_i = L_{p_i}(G_i)$) in $V(T)$. These intersections are the ham-sandwich vertices in $V(T)$ for the original sets $H_1, \ldots, H_d$ of hyperplanes.

In the beginning, we let T be the whole coordinate hyperplane $S = \{x_d = 0\}$ (the word "simplex" is to be interpreted as an intersection of at most $d+1$ halfspaces), $G_i = H_i$ and $p_i = \lfloor (n_i+1)/2 \rfloor$. Then the invariant is then satisfied because of Lemma 4.1.

To establish Proposition 2, we prove an analog of Lemma 3.2:

Lemma 4.2 *Let T, G_i, p_i be as above and satisfy the conditions of the invariant. One can compute new $T' \subset T$, $G_i' \subset G_i$, $p_i' \le p_i$, $(i = 1, \ldots, d)$, again satisfying the conditions of the invariant, and with the new size $m' = |G_1'| + \cdots + |G_d'| \le (1 - 1/2d)m$. The running time is at most proportional to the worst-case running time needed to construct one level in a given arrangement of m hyperplanes in R^{d-1}.*

We again suppose that $m_1 \geq m_2, \ldots, m_d$. The outline of the algorithm is almost identical to the planar case:

1. Partition the simplex T into simplices $T_1, \ldots, T_C$ ($C = C(d)$ a constant) with suitable properties (to be described later).

2. Find one simplex T_i with the odd-intersection property; i.e., $|\lambda_1 \cap \ldots \cap \lambda_d \cap V(T_i)|$ is odd.

3. Construct a region $\tau_i \subset V(T_i)$, such that

$$\lambda_1 \cap V(T_i) \subset \tau_i \tag{4.1}$$

At most half of the hyperplanes of G_1 intersect τ_i. (4.2)

4. Discard all the hyperplanes of G_1 which do not intersect τ_i (at least $m_1/2 \geq m/(2d)$) and update p_1 accordingly ($p_1' \leftarrow p_1 - b$, b the number of G_1 planes lying below τ_i). Then T_i becomes the new T, and we are ready for the next phase of the algorithm.

To define the subdivision of T in Step 1 we need the notion of *ε-approximation.* Let H be a collection of hyperplanes in R^d, and consider the set system $(H, \mathcal{R})$, where $\mathcal{R}$ consists of all subsets of H definable by segments, i.e. of the form $\{h \in H;\ h \cap s \neq \emptyset\}$, where s is a segment in R^d. Given a parameter $\varepsilon > 0$, an ε-approximation for $(H, \mathcal{R})$ is a subset $A \subseteq H$ of hyperplanes with the property that

$$\left| \frac{|A \cap R|}{|A|} - \frac{|R|}{|H|} \right| < \varepsilon, \tag{4.3}$$

for every $R \in \mathcal{R}$. The following lemma is a particular case of a result of [19]:

Lemma 4.3 [18] *Given a set H of n hyperplanes in R^d and $\varepsilon > 0$, one can compute an ε-approximation for $(H, \mathcal{R})$ of size $O(\varepsilon^{-2} \log \frac{1}{\varepsilon})$ in time $O(f(\varepsilon)n)$, where $f(\varepsilon)$ is a factor depending on ε (and d) only; in particular, the running time is $O(n)$ for a fixed ε.* □

Let us remark that a random sample A of size $C\varepsilon^{-2} \log \frac{1}{\varepsilon}$ (for a suitable constant C) will, with high probability, be an ε-approximation for $(H, \mathcal{R})$. This again suggests a possible simplification for an implementation of the algorithm.

The partition in Step 1 of the algorithm is performed as follows: We let $\varepsilon > 0$ be a small enough constant (to be fixed later), and let A be an ε-approximation for the hyperplanes in G_1. We project all pairwise intersections of the hyperplanes of A into the coordinate hyperplane S, which gives a set Π, of $K = \binom{|A|}{2}$ $d-2$ dimensional projections (hyperplanes) in S. Note that the size of A and thus also K are bounded by a constant, as ε is a constant. We form the arrangement of Π (within S) and triangulate the part of it within T, obtaining the simplices $T_1, \ldots, T_C$ (this partitioning procedure, which may look rather mysterious, will be substantiated when discussing Step 3 of the algorithm).

The following lemma deals with Step 2 (selecting the appropriate simplex). This step will dominate the running time, as all other steps can be performed in linear time.

Lemma 4.4 *Given a simplex $T \subset S$, the parity of $|\lambda_1 \cap \ldots \cap \lambda_d \cap V(T)|$ can be determined in time proportional to the (worst-case) time needed to construct one given level for a collection of at most m hyperplanes in R^{d-1}.*

Proof: In each vertical face F of the infinite prism $V(T)$ consider the $d-1$ dimensional arrangement $\mathcal{A}_F$ of the hyperplanes of $G = G_1 \cup \ldots \cup G_d$ intersected with F. We call a vertex $v \in \mathcal{A}_F$ *good* if it is in $\sigma = \lambda_2 \cap \cdots \cap \lambda_d$ and below λ_1.

First, we claim that the parity of $|\lambda_1 \cap \ldots \cap \lambda_d \cap V(T)|$ is the same as the parity of the total number of good vertices within all faces F of $V(T)$. The argument is similar to the one used in Lemma 4.1: Consider a chain $v_0, \ldots, v_t$ in σ and traverse it continuously from the infinite halfline leading to v_0, along edges $v_i v_{i+1}$, and then through the infinite halfline leading from v_t. It meets faces of $V(T)$ an even number of times, say at points $u_1, u_2, \ldots, u_{2k}$, each point alternately an entrance and an exit of $V(T)$ (i.e., u_{2j-1}, u_{2j} denotes a part of the chain *in* $V(T)$ and u_{2k}, u_{2k+1} a part *not in*). Each u_i is a vertex in $\mathcal{A}_F$ for a face of $V(T)$. If u_{2j-1} and u_{2j} are both good or both bad, then the chain has an even number of ham-sandwich cuts in V between these points, and if one of them is good and the other bad, then there are an odd number of cuts. This proves that for each chain in σ the parity of its intersections with faces of $V(T)$ which are good vertices, is the same as the parity of its intersections in $V(T)$ with λ_1. Obviously the same argument can be made for any cycle $v_0, \ldots, v_t$, $v_0 = v_t$ in σ. This establishes our claim and it suffices to describe how the parity of the number of good vertices is found.

An easy way of counting the good vertices is to construct the arrangement $\mathcal{A}_F$, traverse its vertices and count the good ones. This requires $O(m^{d-1})$ time for each face F. But we can do better using level construction algorithms. Let π_F be the vertical hyperplane containing F, and let us put $\bar{G}_i = \{g \cap \pi_F;\ g \in G_i\}$, $\bar{\lambda}_i = L_{p_i}(G_i) = \lambda_i \cap \pi_F$. The problem is now to count the points of $F \cap (\bar{\lambda}_2 \cap \ldots \cap \bar{\lambda}_d)$ lying below $\bar{\lambda}_1$.

For each point of $\bar{\lambda}_i$, we know that the number of hyperplanes of $\bar{G}_i$ below it is p_i. Hence each point (vertex) of $\sigma = \bar{\lambda}_2 \cap \ldots \cap \bar{\lambda}_d$ is a vertex of the level $L_2 = L_{p_2 + \cdots + p_d}(\bar{G}_2 \cup \ldots \cup \bar{G}_d)$. If we have a suitable combinatorial representation of L_2, we can thus traverse it in time proportional to its complexity and find all the vertices of σ.

It remains to decide which vertices of σ are below $\bar{\lambda}_1$. An obvious method is to locate each vertex v of σ in a projection of $\bar{\lambda}_1$ onto a horizontal $(d-2)$-dimensional hyperplane. However, reasonably efficient point location structures in convex subdivisions are only known for dimensions at most 3 (which means $d \leq 5$). We outline an alternative method that works for any d.

We will determine the position of all vertices of the above defined level L_2 with respect to $\bar{\lambda}_1$, by traversing the 1-skeleton of L_2 (by a depth-first graph traversal, say). During this traversal, we remember whether we are below or above $\bar{\lambda}_1$, and we will update this information as we traverse an edge crossing $\bar{\lambda}_1$. To this end, we need to detect all intersections of the edges of σ with $\bar{\lambda}_1$. We observe that each such intersection is a vertex of the level $L_1 = L_{p_1+\cdots+p_d}(\bar{G}_1 \cup \ldots \cup \bar{G}_d)$. Hence all such intersections can be constructed in advance by constructing and traversing L_1. Knowing these intersection, we associate and store them along with the edges of L_2. With a suitable implementation of the traversal of the levels, the running time is dominated by the time needed to construct the levels L_1 and L_2. □

In Step 3, we define the polyhedron τ_i and establish its properties. Let $c > 0$ be a constant to be specified later. For each vertex v_j of the simplex $T_i \subset S$, we define the points D_j^-, D_j^+ as follows:

$$D_j^- = L_{p_1 - c\varepsilon m_1}(G_1) \cap V(v_j) \text{ and}$$

$$D_j^+ = L_{p_1 + c\varepsilon m_1}(G_1) \cap V(v_j);$$

i.e., D_j^- (resp. D_j^+) is the intersection of the $p_1 - c\varepsilon m_1$ (resp. $p_1 + c\varepsilon m_1$) level of the G_1 hyperplanes with the vertical line through v_j (these points can be found in $O(m_1)$ time by linear-time selection). Then we define τ_i as the convex hull of $\{D_1^-, D_1^+, \ldots, D_d^-, D_d^+\}$. It remains to prove that the constants c, ε can be chosen in such a way that τ_i has the required properties (4.1) and (4.2).

Lemma 4.5 *Choose $c \le 3(d-1)/2$ and $\varepsilon = 1/(6(d-1)+4c)$. Then (4.1) and (4.2) hold for any τ_i; i.e., at most half of the lines in G_1 meet τ_i, and within the prism $V(T_i)$, the level $L_{p_1}(G_1)$ remains within τ_i.*

Proof: Consider a pair D_j^+, D_k^+ of vertices of τ_i. We will estimate the number of hyperplanes of G_1 intersecting the segment $D_j^+ D_k^+$. The levels of D_j^+ and D_k^+ in the arrangement of G_1 are equal, and the definition of ε-approximation implies that their levels in the arrangement of A differ by at most $2\varepsilon|A|$.

Suppose that there are more than $2\varepsilon|A|$ of the A hyperplanes intersecting the segment $D_j^+ D_k^+$. It is easy to argue that there must be two hyperplanes of A intersecting inside the two-dimensional vertical strip erected through the segment $D_j^+ D_k^+$ (the argument is similar as in the planar case). If we project the intersection of such two hyperplanes into S, we get a hyperplane (within S) belonging to the set Π. But T_i was a simplex from a triangulation of the arrangement of Π, so its edge cannot be intersected by a hyperplane of Π. This contradiction shows that the segment $D_j^+ D_k^+$ is intersected by no more than $2\varepsilon|A|$ of the A-hyperplanes, and thus by at most $3\varepsilon m_1$ of the G_1 hyperplanes, by the ε-approximation property.

Since the top and bottom faces of τ_i have a total of $d(d-1)$ edges, there are at most $3d(d-1)\varepsilon m_1$ intersections of hyperplanes in G_1 with edges in the top or bottom of τ_i. By the definition of D_j^-, D_j^+, each of the d vertical edges of τ_i accounts for $2c\varepsilon m_1$ intersections with hyperplanes in G_1, giving a total of at most $(3d(d-1) + 2cd)\varepsilon m_1$ intersections. Because each hyperplane meeting τ_i intersects at least d edges, at most $(3(d-1)+2c)\varepsilon m_1$ of the hyperplanes in G_1 can meet τ_i. Whatever c is, we will take $\varepsilon \leq 1/(6(d-1)+4c)$ and satisfy (4.2). To fix c, we already showed that there are at most $3d(d-1)\varepsilon m_1/2$ intersections of hyperplanes in G_1 with edges in the top face of τ_i. The choice of $c \geq 3(d-1)/2$ guarantees that the top face of τ_i meets at most $c\varepsilon m_1$ hyperplanes in G_1. This means that the level of each point in the top face differs from the (common) level of the vertices D_j^+ by at most $c\varepsilon m_1$, and, in particular, it is not smaller than p_1. This implies that λ_1, the p_1-level of G_1, can never get above the top of τ_i. The argument for the bottom is the same. This finishes the proof of Lemma 4.5 and therefore of Proposition 2 as well. □

5 A Separated Case in R^3

Suppose we have three disjoint sets P_1, P_2, P_3 in R^3. A line ℓ is *a transversal* if it meets all three convex hulls $\text{conv}(P_1)$, $\text{conv}(P_2)$, $\text{conv}(P_3)$. Our separation condition is that the sets have no transversal. For this case we generalize Megiddo's result [21] and prove that the complexity of the separated ham-sandwich problem in R^3 is $O(n)$. Specifically we will show that the separation condition allows a modification of the general algorithm so it runs in linear time. In Step 2 of the algorithm — the only one requiring more than linear time — we will be able to replace level construction in a two dimensional vertical face by planar ham-sandwich computations and a few other linear time operations.

Let us begin with two equivalent formulations of the transversal condition.

Lemma 5.1 *The following statements about three convex sets $A_1, A_2, A_3 \subset R^3$ are equivalent:*

(i) *A_1, A_2, A_3 have no line transversal.*

(ii) *For every permutation (i, j, k) of $(1, 2, 3)$, A_i can be separated from $A_j \cup A_k$ by a plane.*

(iii) *For any plane ρ, at least one pair of sets among the orthogonal projections of A_1, A_2, A_3 on ρ has an empty intersection.*

Proof: (i)⇒(ii): If suffices to show that $\text{conv}(A_1 \cup A_2) \cap A_3 = \emptyset$. Any point $x \in \text{conv}(A_1 \cup A_2)$ lies on a segment $a_1 a_2$ with $a_1 \in A_1$ and $a_2 \in A_2$, so if also $x \in A_3$ then the line ℓ through a_1, x and a_2 is a transversal.

(ii)⇒(iii): Let σ_i denote a plane separating A_i from the union of the other two sets $(i = 1, 2, 3)$. For simplicity assume that $\sigma_1, \sigma_2, \sigma_3$ are in general position; place the origin of coordinates to the point $\sigma_1 \cap \sigma_2 \cap \sigma_3$ and let σ_i^+ denote the halfspace bounded by σ_i and containing A_i, and σ_i^- the opposite halfspace. We have $A_1 \subset \sigma_1^+ \cap \sigma_2^- \cap \sigma_3^-$, and similarly for A_2, A_3. Let ρ be a projection plane and let r be its normal. Let us place the vector r into the origin and discuss the position of its endpoint R with respect to the σ_i's. If R belongs, for instance, to $\sigma_1^- \cap \sigma_2^-$, then the plane passing through R and through the line $\sigma_1 \cap \sigma_2$ separates A_1 from A_2, and it projects to a line in ρ separating the projection of A_1 from the projection of A_2. Similarly for $R \in \sigma_1^+ \cap \sigma_2^+$, and generally we get a separating line for some pair of projections whenever the signs of the halfspaces containing R for some two indices coincide. But this is the case for any R.

(iii)⇒(i): The projection to a plane orthogonal to a line transversal ℓ violates the condition (iii). □

Note that the condition (ii) can be tested in $O(n)$ time (using a linear-time linear programming algorithm in dimension 3).

A dual formulation of (iii) yields the condition we will need in the algorithm.

Lemma 5.2 *Let P_1, P_2, P_3 be point sets satisfying the separation condition, and let H_1, H_2, H_3 be the dual sets of planes. Let π be a vertical plane. There exists a pair of indices $(i,j) \in \{(1,2),(2,3),(3,1)\}$ such that if λ_i is some level of H_i and λ_j some level of H_j, then λ_i and λ_j have a unique intersection within π. Given π, such a pair of indices can be determined in $O(n)$ time.*

Proof: Let π be described by the equation $cx + dy + e = 0$. The duality transform maps a point $x \in \pi$ to a plane $\mathcal{D}(x)$ parallel to the direction $r = (2c, 2d, -e)$, so the points in π correspond to lines in a plane ρ orthogonal to r. If another plane h intersects π in a line ℓ, then the points of ℓ dualize to planes parallel to r and passing through the point $\mathcal{D}(h)$ dual to h. Hence the corresponding lines in ρ all pass through the projection of $\mathcal{D}(h)$ on ρ, so a line if π corresponds to a point in ρ. It is not difficult to verify that the point in ρ does not depend on the choice of h, and that this correspondence between points and lines in π and lines and points in ρ has the properties of a duality transform.

Returning to our situation, we find (according to Lemma 5.1(iii)) a pair (i,j) of indices such that the projections of the (primal) sets P_i and P_j into the above defined plane ρ are linearly separated (this can be done in linear time by linear programming). The proof is concluded by showing that when $\bar{P}_i, \bar{P}_j$ are the linearly separated projections in the plane ρ, then any level of the arrangement of lines dual to $\bar{P}_i$ (in the plane π, under the above discussed dual correspondence between ρ and π) intersects any level of the arrangement of lines dual to $\bar{P}_j$ in a unique point. This is essentially a result of Megiddo. He proved the uniqueness of the ham-sandwich cut for linearly separated sets, but the idea applies to any pair of levels. Choose the system of coordinates in the primal plane so that the separating line is the y-axis, and the coordinates in the dual plane so that the duality is the "usual" one (introduced in Sec. 2). Then all the lines dual to $\bar{P}_i$ have (say) positive slopes while the ones dual to $\bar{P}_j$ have negative slopes, and the claim follows. □

Step 2 of the algorithm tests a triangle T_i for the odd-intersection property by computing the parity of good vertices in the vertical faces of $V(T_i)$. In the general case we constructed the relevant levels in a face and *counted* the good vertices. Using the separation condition, we may deduce the parity without constructing the levels.

Lemma 5.3 *Let n lines in general position in R^2 partitioned into sets H_1, H_2, H_3 be given, and let λ_i denote a level in the arrangement of H_i lines. Suppose that (i, j) is a given pair of indices such that $|\lambda_i \cap \lambda_j| = 1$. Then in time $O(n)$ we can compute the parity of good vertices in the strip $V = V((\ell, r)) = \{(x, y) : x \in (\ell, r)\}$ (a vertex $v = (x, y) \in \lambda_2 \cap \lambda_3$ is good if $\lambda_1(x) > y$).*

Proof: First suppose that the order of intersections of λ_i and λ_j with the vertical line $x = \ell$ is the same as the one for the vertical line $x = r$, that is,

$$(\lambda_i(\ell) - \lambda_j(\ell))(\lambda_i(r) - \lambda_j(r)) \geq 0. \tag{5.1}$$

Then λ_i and λ_j have an even number of intersections within V, and since this number is at most 1 at the same time, they are disjoint within V. So if $\{i, j\} = \{2, 3\}$ there are no good vertices and we are done. Otherwise, by symmetry, we may suppose $i = 1$, $j = 2$. In such case, either all vertices of $\lambda_2 \cap \lambda_3$ in V are good (if λ_2 is below λ_1 within V) or none is (if λ_2 is above λ_1). In the former case, the parity of $|\lambda_2 \cap \lambda_3 \cap V|$ can be deduced from the ordering of the intersections of λ_2 and λ_3 with the verticals bounding V.

It remains to deal with the case when (5.1) does not hold. In such case, we know that the (unique) intersection of λ_i and λ_j is contained in V, and we can find it in $O(n)$ time by the algorithm of Section 3; let c be its x-coordinate. We then replace the interval $T = (\ell, r)$ by two intervals $T' = (\ell, c)$ and $T'' = (c, r)$ and observe that (5.1) already holds for both of them. Thus we can determine the parity of good vertices within $V(T')$, within $V(T'')$ and account appropriately for the potential good vertices lying on the vertical line $x = c$ (for $(i, j) = (2, 3)$). $\square$

We can return to the algorithm from the previous section. We have a triangle $T = PQR$ in $x_3 = 0$. Lemma 5.2 shows that in the vertical plane containing its side (say PQ) at least one pair of the considered levels has a unique intersection, and that we can find such a pair in linear time. Lemma 5.3 then shows that the parity of good vertices in the vertical strip $V(PQ)$ may be found in linear time. This shows that Step 2 of the algorithm has linear complexity and proves

Proposition 3 *Given n points in R^3 partitioned into sets P_1, P_2, P_3, each with an odd number of points, and having no transversal, the complexity of the ham-sandwich problem is $O(n)$.*

6 Final Remarks

• **Approximation.** In various applications of the ham-sandwich cut construction, one sometimes does not really need an exact bisection of every set, but an approximate bisection suffices instead. Let us say that a hyperplane h is an *ε-approximate ham-sandwich cut* for sets $P_1, \ldots, P_d$ if there are no more than $(\varepsilon + 1/2)|P_i|$ points of P_i in either of the open halfspaces defined by h, $i = 1, \ldots, d$. For a fixed $\varepsilon > 0$, one can obtain such an approximate ham-sandwich cut in linear time, in any fixed dimension. First we compute A_i, an ε-approximation for P_i with respect to halfspaces, this means that for every halfspace γ,

$$\left| \frac{|P_i \cap \gamma|}{|P_i|} - \frac{|A_i \cap \gamma|}{|A_i|} \right| < \varepsilon \, .$$

Such an A_i of size depending on d and ε only can be computed deterministically and in $O(|P_i|)$ time, see [19]. Then we compute a ham-sandwich cut for $A_1, \ldots, A_d$ by some algorithm; since the size of A_i is bounded by a constant, this only takes a constant time. It is easy to see that a ham-sandwich cut for $A_1, \ldots, A_d$ is also an ε-approximate ham-sandwich cut for $P_1, \ldots, P_d$. In practice, one may take random samples of suitable size for the A_i's, and is guaranteed to find an ε-approximate ham-sandwich cut with probablility close to 1.

• **Applications.** Willard's partitioning problem, initially solved by Cole, Sharir, and Yap, admitted an optimal solution when Megiddo's ham-sandwich algorithm for the separated case was applied. There are some other problems to which the algorithms of the present paper may be applied so the current solutions can be improved. For example Atallah [6] considered the problem of matching n given red points $r_1, \ldots, r_n$ in the plane with n given blue points, $b_1, \ldots, b_n$ in such a way that the segments joining matched pairs do not intersect. He gave an $O(n(\log n)^2)$ algorithm for this task. If we used the ham-sandwich algorithm of Section 2 for the *divide* step of a recursive algorithm, after $O(\log n)$ levels we would have n trivial matching problems, each with one red and one blue point, and the segments will not intersect. This gives an extremely simple, $O(n \log n)$ solution to the matching problem which, by reduction to sorting, is easily seen to be optimal [the red points are $(1, 1), \ldots, (1, n)$; the blue points are $(0, a_1), \ldots, (0, a_n)$, the a_i the inputs to the sorting problem; the matching gives the ranks of the a_i's]. The approach easily extends to a higher dimensional version where, with d sets of n points each (each set of a certain color), the matching is an assignment of each point to a distinct, multicolored d-simplex; the geometric requirement is that the n simplices are pairwise disjoint. The algorithm we described here can be used to find such a matching in $O(n^{d-1-\gamma})$ time. This matching problem was discussed by Akiyama and Alon [3] but no algorithm was mentioned.

Acknowledgment: We thank Robin Thomas for valuable conversations, and Imre Bárány for showing us an elegant alternative proof of Lemma 4.1.

References

[1] P. Agarwal and J. Matoušek. Dynamic half-space range reporting and its applications. Tech. Report CS–91–43, Dept. Comput. Science, Duke University, 1991. The results combined with results of D. Eppstein appear in *Proc. 33. IEEE Symposium on Foundations of Computer Science*, 1992.

[2] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in cLog n parallel steps. *Combinatorica 3*, 1-19, 1983.

[3] J. Akiyama and N. Alon. Disjoint simplices and geometric hypergraphs. *Annals of the New York Academy of Sciences, vol 555*, 1-3, 1989.

[4] N. Alon, I. Bárány, Z. Füredi, and D. Kleitman. Point selections and weak ε-nets for convex hulls. Manuscript, 1991.

[5] B. Aronov, B. Chazelle, H. Edelsbrunner, L. Guibas, M. Sharir, and R. Wenger. Points and triangles in the plane and halving planes in the space. *Discrete & Computational Geometry 6*, 435–442, 1991.

[6] M. Atallah. A matching problem in the plane. *J. Comput. Syst. Sciences 3*, 63–70, 1985.

[7] I. Bárány and W. Steiger. On the expected number of k-sets Tech. report, Rutgers University, 1992.

[8] R. Cole, J.S. Salowe, W.L. Steiger and E. Szemerédi. An optimal-time algorithm for slope selection. *SIAM J. Computing 18*, 792–810, 1989.

[9] R. Cole, M. Sharir, and C. Yap. On k-hulls and related topics. *SIAM J. Computing 16*, 61–77, 1987.

[10] T. Dey and H. Edelsbrunner. Counting simplex crossings and halving hyperplanes. Preprint, U. Illinois. 1992.

[11] M. B. Dillencourt, D. M. Mount, and N. S. Netanyahu. A randomized algorithm for slope selection. *International Journal of Computational Geometry and Applications*, 2, 1992, 1–27.

[12] H. Edelsbrunner. *ALGORITHMS IN COMBINATORIAL GEOMETRY.* Springer-Verlag, Berlin, 1987.

[13] H. Edelsbrunner and R. Waupotitsch. Computing a ham sandwich cut in two dimensions. *J. Symbolic Computing 2*, 171–178, 1986.

[14] H. Edelsbrunner. Edge-skeletons in arrangements with applications. *Algorithmica 1*, 93–109, 1986.

[15] H. Edelsbrunner and E. Welzl. Constructing belts in two-dimensional arrangements with applications. *SIAM J. Computing 15*, 271–284, 1986.

[16] C.-Y. Lo and W.L. Steiger. An optimal time algorithm for ham-sandwich cuts in the plane. *Proc. 2. Canadian Conference on Computational Geometry*, 5–9, 1990.

[17] C.-Y. Lo, J. Matoušek and W.L. Steiger. Ham-sandwich cuts in R^d. *Proc. 24. ACM Symposium on Theory of Computing*, 539–545, 1992.

[18] J. Matoušek. Construction of ε-nets. *Discrete & Computational Geometry 5*, 427–448, 1990.

[19] J. Matoušek. Approximations and optimal geometric divide-and-conquer. *Proc. 23rd Ann. ACM Symposium on Theory of Computing*, 505–511, 1991.

[20] J. Matoušek. Randomized optimal algorithm for slope selection. *Information Processing Letters*, 183–187, 1991.

[21] N. Megiddo. Partitioning with two lines in the plane. *J. Algorithms 6*, 430–433, 1985.

[22] J. Pach, W. Steiger, and E. Szemerédi. An upper bound on the number of planar k-sets. *Proc. 30th IEEE Symposium on Foundations of Computer Science*, 72–79, 1989.

[23] D. Willard. Polygon retrieval. *SIAM J. Computing 11*, 149–165, 1982.

[24] R.T. Živaljević and S.T. Vrećica. The colored Tverberg problem and complexes of injective functions. *Israel J. Math.*, to appear.