On the computational complexity of Ham-Sandwich cuts, Helly sets and related problems

Christian Knauer (U Bayreuth)  
Hans Raj Tiwary (UL Bruxelles)  
Daniel Werner (FU Berlin)

March 12, 2011
Basics

The Ham-Sandwich Theorem
Our results
d-Sum

d-Ham-Sandwich

The idea
The construction
Correctness
Summary

Further results
The planar case

Let $P = R \cup B$. Then there is a line that *bisects* both sets simultaneously.
The planar case

Let $P = R \cup B$. Then there is a line that *bisects* both sets simultaneously.

Such a line can be found in linear time!

[Edelsbrunner, Waupotitsch; '86]
Theorem

For every $d$ point sets in $\mathbb{R}^d$ there exists a hyperplane that bisects them simultaneously.
**Theorem**

For every $d$ point sets in $\mathbb{R}^d$ there exists a hyperplane that bisects them simultaneously.

**Proof:** Borsuk-Ulam
General version

**Theorem**

For every $d$ point sets in $\mathbb{R}^d$ there exists a hyperplane that bisects them simultaneously.

**Proof:** Borsuk-Ulam

**known bounds:**

- trivial algorithm: $n^{d+1}$
General version

Theorem

For every \( d \) point sets in \( \mathbb{R}^d \) there exists a hyperplane that bisects them simultaneously.

Proof: Borsuk-Ulam

known bounds:

- trivial algorithm: \( n^{d+1} \)
- best known: \( O(n^{d-1}) \) [Lo, Matoušek, Steiger; ’92]
The Ham-Sandwich Theorem

For every $d$ point sets in $\mathbb{R}^d$ there exists a hyperplane that bisects them simultaneously.

**Proof:** Borsuk-Ulam

**known bounds:**

- trivial algorithm: $n^{d+1}$
- best known: $O(n^{d-1})$ [Lo, Matoušek, Steiger; ’92]
- recently: $O(n \log^d n)$ for well separated point sets [Bárány, Hubard, Jéronimo; ’08], [Steiger, Zhao; ’09]
The decision problem

Can we find a cut incrementally?
The decision problem

Can we find a cut incrementally?

\((d\text{-}HAM\text{-}SANDBWICH)\)

**Given:** Sets \(P_1, \ldots, P_d\) in \(\mathbb{R}^d\)
The decision problem

Can we find a cut incrementally?

\((d\text{-HAM-SANDWICH})\)

**Given:** Sets \(P_1, \ldots, P_d\) in \(\mathbb{R}^d\)

**Question:** Is there a ham-sandwich cut through the origin?
The decision problem

Can we find a cut incrementally?

\textbf{(d-Ham-Sandwich)}

\textbf{Given:} Sets $P_1, \ldots, P_d$ in $\mathbb{R}^d$

\textbf{Question:} Is there a ham-sandwich cut through the origin?

Alternatively:

\textbf{Given:} Sets $P_1, \ldots, P_{d+1}$ in $\mathbb{R}^d$

\textbf{Question:} Is there a ham-sandwich cut?
The decision problem

Can we find a cut incrementally?

\((d\text{-HAM-SANDWICH})\)

**Given:** Sets \(P_1, \ldots, P_d\) in \(\mathbb{R}^d\)

**Question:** Is there a ham-sandwich cut through the origin?

Alternatively:

**Given:** Sets \(P_1, \ldots, P_{d+1}\) in \(\mathbb{R}^d\)

**Question:** Is there a ham-sandwich cut?

No complexity results known so far.

Christian Knauer, Hans Raj Tiwary, Daniel Werner
If the dimension is part of the input, \( d\text{-HAM-SANDWICH} \) is
Our results

If the dimension is part of the input, \(d\text{-HAM-SANDWICH}\) is

- **NP-hard** (does not exclude \(O(n)\) for every fixed dimension)
If the dimension is part of the input, \textit{d-HAM-SANDWICH} is

- \textbf{NP-hard} (does not exclude $O(n)$ for every fixed dimension)
- \textbf{W[1]-hard} when parameterized with the dimension
Our results

If the dimension is part of the input, $d$-HAM-SANDWICH is

- **NP-hard** (does not exclude $O(n)$ for every fixed dimension)
- **W[1]-hard** when parameterized with the dimension
- requires $n^{\Omega(d)}$ time, unless 3-SAT can be solved in $2^{o(n)}$
The $d$-$\text{SUM}$ problem

$\text{(d-SUM)}$

**Given:** A set of integers $S = \{s_1, \ldots, s_n\}$. 
The \(d\text{-}\text{SUM}\) problem

\((d\text{-SUM})\)

**Given:** A set of integers \(S = \{s_1, \ldots, s_n\}\).

**Question:** Do \(d\) of them sum up to 0?
The \textbf{d-Sum} problem

\begin{itemize}
\item \textbf{(d-Sum)}
\item \textbf{Given:} A set of integers \( S = \{s_1, \ldots, s_n\} \).
\item \textbf{Question:} Do \( d \) of them sum up to 0?
\end{itemize}

\begin{itemize}
\item parameterized version of \textbf{Subset-Sum}
\end{itemize}
The $d$-SUM problem

$\text{(d-SUM)}$

**Given:** A set of integers $S = \{s_1, \ldots, s_n\}$.

**Question:** Do $d$ of them sum up to 0?

- parameterized version of $\text{SUBSET-SUM}$
- $\text{W}[1]$-hard [Fellows, Koblitz; ’93]
The $d$-\textsc{Sum} problem

($d$-\textsc{Sum})

\textbf{Given: } A set of integers $S = \{s_1, \ldots, s_n\}$.

\textbf{Question: } Do $d$ of them sum up to 0?

- parameterized version of \textsc{Subset-Sum}
- requires $n^{\Omega(d)}$ time, unless 3-\textsc{Sat} can be solved in $2^{o(n)}$ [Pătraşcu, Williams; ’10]
The idea

Reduction from $d$-SUM
The idea

Reduction from $d$-SUM

**General idea:** *Embed* the numbers as points into $\mathbb{R}^f(d)$ that have a certain property iff there are $d$ numbers that sum up to 0.
The idea

Reduction from $d$-SUM

**General idea:** *Embed* the numbers as points into $\mathbb{R}^f(d)$ that have a certain property iff there are $d$ numbers that sum up to 0.

**Here:** Construct point sets $P_1, \ldots, P_{d+1}$ in $\mathbb{R}^{d+1}$ such that
The idea

Reduction from $d$-SUM

**General idea:** Embed the numbers as points into $\mathbb{R}^{f(d)}$ that have a certain property iff there are $d$ numbers that sum up to 0.

**Here:** Construct point sets $P_1, \ldots, P_{d+1}$ in $\mathbb{R}^{d+1}$ such that there exists a *linear* ham-sandwich cut

\[ \Leftrightarrow \]

$d$ of the numbers sum up to 0.
Encoding the numbers

Let $S = \{s_1, \ldots, s_n\}$
Encoding the numbers

Let $S = \{s_1, \ldots, s_n\}$

**Goal:** Construct $d$ sets $P_1, \ldots, P_d$ in $\mathbb{R}^{d+1}$ from $S$

(and one extra set later)
Encoding the numbers

Let $S = \{s_1, \ldots, s_n\}$

**Goal:** Construct $d$ sets $P_1, \ldots, P_d$ in $\mathbb{R}^{d+1}$ from $S$
(and one extra set later)
such that number appears in solution $\iff$ linear cut goes through corresponding point.
Encoding the numbers

Let $S = \{s_1, \ldots, s_n\}$

**Goal:** Construct $d$ sets $P_1, \ldots, P_d$ in $\mathbb{R}^{d+1}$ from $S$

(and one extra set later)

such that number appears in solution $\iff$ linear cut goes through corresponding point.

In dimension $j$: add point $p_i^j := \frac{1}{s_i} \cdot e_j + e_{d+1}$ for $1 \leq i \leq n$
Encoding the numbers

Let \( S = \{s_1, \ldots, s_n\} \)

**Goal:** Construct \( d \) sets \( P_1, \ldots, P_d \) in \( \mathbb{R}^{d+1} \) from \( S \)

(and one extra set later)

such that number appears in solution \( \iff \) linear cut goes through corresponding point.

**In dimension** \( j \): add point \( p^j_i := \frac{1}{s_i} \cdot e_j + e_{d+1} \) for \( 1 \leq i \leq n \)
Encoding the numbers

Let $S = \{s_1, \ldots, s_n\}$

**Goal:** Construct $d$ sets $P_1, \ldots, P_d$ in $\mathbb{R}^{d+1}$ from $S$

(and one extra set later)

such that number appears in solution $\iff$ linear cut goes through corresponding point.

**In dimension $j$:** add point $p_i^j := \frac{1}{s_i} \cdot e_j + e_{d+1}$ for $1 \leq i \leq n$

Observe: if $h \cdot p_i^j = 0$ then $h_j = -h_{d+1}s_i$. 
**Problem:** Hyperplane through origin will *not* bisect the sets:

\[ x_{d+1} = 1 \]

Balancing points

Christian Knauer, Hans Raj Tiwary, Daniel Werner
Problem: Hyperplane through origin will not bisect the sets:

⇒ add balancing points
Balancing points

**Problem:** Hyperplane through origin will *not* bisect the sets:

\[ x_{d+1} = 1 \]

⇒ add *balancing* points

Christian Knauer, Hans Raj Tiwary, Daniel Werner
Balancing points

**Problem:** Hyperplane through origin will *not* bisect the sets:

\[ x_{d+1} = 1 \]

\[ x_{d+1} = 1 \]

⇒ add *balancing* points
Balancing points

**Problem:** Hyperplane through origin will *not* bisect the sets:

\[ x_{d+1} = 1 \]

⇒ add *balancing* points
Problem: Hyperplane through origin will \textit{not} bisect the sets:

\begin{align*}
\mathbf{x}_d + 1 &= 1 \\
\mathbf{x}_d + 1 &= \mathbf{x}_j
\end{align*}

$\implies$ add \textit{balancing} points
Balancing points

**Problem:** Hyperplane through origin will *not* bisect the sets:

\[ x_{d+1} = 1 \]

⇒ add *balancing* points
Balancing points

**Problem:** Hyperplane through origin will *not* bisect the sets:

\[ x_{d+1} = 1 \]

\[ x_j \]

\[ \Rightarrow \text{add balancing points} \]
The point $q$

One extra point will ensure that

- none of the balancing points can lie on a linear cut
The point $q$

One extra point will ensure that

- none of the balancing points can lie on a linear cut
- if points lie on linear cut $\Rightarrow$ corresponding numbers sum to 0
The point $q$

One extra point will ensure that

- none of the balancing points can lie on a linear cut
- if points lie on linear cut $\Rightarrow$ corresponding numbers sum to 0

Set

$$q = - \sum_{i=1}^{d} e_i$$

and $P_{d+1} = \{q\}$. 
Some facts

Every linear cut
  • must contain $q$
Some facts

Every linear cut

- must contain \( q \)
- contains \textit{exactly} one point from each \( P_i \)
Some facts

Every linear cut

- must contain $q$
- contains exactly one point from each $P_i$
- contains none of the balancing points
Why it works

Claim:

There are $d$ numbers that sum to 0.

$\Leftrightarrow$

There is a linear ham-sandwich cut.
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$. 
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}$, $1 \leq j \leq d$
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}$, $1 \leq j \leq d$ and $h_{d+1} = -1$. 
Why it works

⇒: Let \( \sum_{j=1}^{d} s_{ij} = 0 \).

Let \( h_j = s_{ij}, \ 1 \leq j \leq d \) and \( h_{d+1} = -1 \).

Then

\[
hp_{ij}^j
\]
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}$, $1 \leq j \leq d$ and $h_{d+1} = -1$.

Then

$$hp_{ij}^j = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right)$$
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}$, $1 \leq j \leq d$ and $h_{d+1} = -1$.

Then

$$hp_{ij} = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right) = s_{ij} \frac{1}{s_{ij}} - 1$$
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.
Let $h_j = s_{ij}$, $1 \leq j \leq d$ and $h_{d+1} = -1$.

Then

$$hp_{ij}^j = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right) = s_{ij} \frac{1}{s_{ij}} - 1 = 0,$$
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}$, $1 \leq j \leq d$ and $h_{d+1} = -1$.

Then

$$hp_{ij}^j = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right) = s_{ij} \frac{1}{s_{ij}} - 1 = 0,$$

so $h$ halves each $P_i$, $1 \leq i \leq d$. 
Why it works

⇒: Let \( \sum_{j=1}^{d} s_{ij} = 0 \).

Let \( h_j = s_{ij}, \ 1 \leq j \leq d \) and \( h_{d+1} = -1 \).

Then

\[
hp_{ij}^j = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right) = s_{ij} \frac{1}{s_{ij}} - 1 = 0,
\]

so \( h \) halves each \( P_i, \ 1 \leq i \leq d \).

Further, as

\[
hq
\]
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}, 1 \leq j \leq d$ and $h_{d+1} = -1$.

Then

$$ h p^j_{i,j} = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right) = \frac{1}{s_{ij}} s_{ij} - 1 = 0,$$

so $h$ halves each $P_i, 1 \leq i \leq d$.

Further, as

$$ h q = h \sum_{i=1}^{d} e_i $$
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}, 1 \leq j \leq d$ and $h_{d+1} = -1$.

Then

$$hp^j_{ij} = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right) = s_{ij} \frac{1}{s_{ij}} - 1 = 0,$$

so $h$ halves each $P_i, 1 \leq i \leq d$.

Further, as

$$hq = h \sum_{i=1}^{d} e_i = \sum_{j=1}^{d} s_{ij}$$
Why it works

⇒: Let $\sum_{j=1}^{d} s_{ij} = 0$.

Let $h_j = s_{ij}$, $1 \leq j \leq d$ and $h_{d+1} = -1$.

Then

$$hp_{ij}^j = h \left( \frac{1}{s_{ij}} \cdot e_j + e_{d+1} \right) = s_{ij} \frac{1}{s_{ij}} - 1 = 0,$$

so $h$ halves each $P_i$, $1 \leq i \leq d$.

Further, as

$$hq = h \sum_{i=1}^{d} e_i = \sum_{j=1}^{d} s_{ij} = 0$$

$q$ also lies on $h$. 
Why it works

⇐: Let $h$ be a linear cut.
Why it works

$\Leftarrow$: Let $h$ be a linear cut.

**Fact:** $h$ contains exactly one point from each $P_i$

(in particular, $h_{d+1} \neq 0$)
Why it works

⇐: Let $h$ be a linear cut.

Fact: $h$ contains exactly one point from each $P_i$

(in particular, $h_{d+1} \neq 0$)

Fact: each must be a point of the form $p_i = \frac{1}{s_i} \cdot e_j + e_{d+1}$
Why it works

\( \Leftarrow \): Let \( h \) be a linear cut.

**Fact:** \( h \) contains exactly one point from each \( P_i \)

\[ (\text{in particular, } h_{d+1} \neq 0) \]

**Fact:** each must be a point of the form \( p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1} \)

and wlog \( h_{d+1} = -1 \), thus \( h_j = s_{ij} \) for some \( i_j \).
Why it works

$\Leftarrow$: Let $h$ be a linear cut.

**Fact:** $h$ contains exactly one point from each $P_i$

(in particular, $h_{d+1} \neq 0$)

**Fact:** each must be a point of the form $p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1}$

and wlog $h_{d+1} = -1$, thus $h_j = s_{ij}$ for some $i_j$.

Further, as $q$ lies on $h$, we have
Why it works

⇐: Let $h$ be a linear cut.

Fact: $h$ contains exactly one point from each $P_i$ (in particular, $h_{d+1} 
eq 0$)

Fact: each must be a point of the form $p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1}$ and wlog $h_{d+1} = -1$, thus $h_j = s_{ij}$ for some $i_j$.

Further, as $q$ lies on $h$, we have

\[ 0 \]
Why it works

⇐: Let $h$ be a linear cut.

**Fact:** $h$ contains exactly one point from each $P_i$

(in particular, $h_{d+1} \neq 0$)

**Fact:** each must be a point of the form $p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1}$

and wlog $h_{d+1} = -1$, thus $h_j = s_{ij}$ for some $i_j$.

Further, as $q$ lies on $h$, we have

$$0 =hq$$
Why it works

$\Leftarrow$: Let $h$ be a linear cut.

**Fact:** $h$ contains exactly one point from each $P_i$ 
(in particular, $h_{d+1} \neq 0$)

**Fact:** each must be a point of the form $p_i^j = \frac{1}{s_i} \cdot e_j + e_{d+1}$

and wlog $h_{d+1} = -1$, thus $h_j = s_{ij}$ for some $i_j$.

Further, as $q$ lies on $h$, we have

$$0 = hq = h \sum_{i=1}^{d} e_i$$
Why it works

\[\Leftarrow: \text{Let } h \text{ be a linear cut.}\]

**Fact:** \( h \) contains exactly one point from each \( P_i \)

\[(\text{in particular, } h_{d+1} \neq 0)\]

**Fact:** each must be a point of the form

\[p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1}\]

and wlog \( h_{d+1} = -1 \), thus \( h_j = s_{i_j} \) for some \( i_j \).

Further, as \( q \) lies on \( h \), we have

\[0 =hq = h \sum_{i=1}^{d} e_i = \sum_{j=1}^{d} h_j\]
Why it works

⇐: Let $h$ be a linear cut.

**Fact:** $h$ contains exactly one point from each $P_i$

(in particular, $h_{d+1} \neq 0$)

**Fact:** each must be a point of the form $p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1}$

and wlog $h_{d+1} = -1$, thus $h_j = s_{ij}$ for some $i_j$.

Further, as $q$ lies on $h$, we have

$$0 = hq = h \sum_{i=1}^{d} e_i = \sum_{j=1}^{d} h_j = \sum_{j=1}^{d} s_{ij}$$
Why it works

\[ \Leftarrow: \text{Let } h \text{ be a linear cut.} \]

**Fact:** \( h \) contains exactly one point from each \( P_i \)

(in particular, \( h_{d+1} \neq 0 \))

**Fact:** each must be a point of the form

\[ p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1} \]

and \( \text{wlog } h_{d+1} = -1 \), thus \( h_j = s_{ij} \) for some \( i_j \).

Further, as \( q \) lies on \( h \), we have

\[ 0 = hq = h \sum_{i=1}^{d} e_i = \sum_{j=1}^{d} h_j = \sum_{j=1}^{d} s_{ij} \]

and thus \( d \) numbers sum up to 0.
Why it works

⇐: Let \( h \) be a linear cut.

Fact: \( h \) contains exactly one point from each \( P_i \)

(in particular, \( h_{d+1} \neq 0 \))

Fact: each must be a point of the form \( p^j_i = \frac{1}{s_i} \cdot e_j + e_{d+1} \)

and wlog \( h_{d+1} = -1 \), thus \( h_j = s_{i_j} \) for some \( i_j \).

Further, as \( q \) lies on \( h \), we have

\[
0 = hq = h \sum_{i=1}^{d} e_i = \sum_{j=1}^{d} h_j = \sum_{j=1}^{d} s_{i_j}
\]

and thus \( d \) numbers sum up to 0.
What we have done

\[ S = \{s_1, \ldots, s_n\} \text{ is a } d\text{-SUM instance} \]
What we have done

\[ S = \{s_1, \ldots, s_n\} \text{ is a } d\text{-SUM instance} \]
\[ \iff \]
\[ P_1, \ldots, P_{d+1} \text{ can be bisected by a linear cut} \]
What we have done

\[ S = \{s_1, \ldots, s_n\} \text{ is a } d\text{-SUM instance} \]

\[ \iff \]

\[ P_1, \ldots, P_{d+1} \text{ can be bisected by a linear cut} \]

- ptime-reduction \(\Rightarrow\) NP-hard
What we have done

\[ S = \{s_1, \ldots, s_n\} \text{ is a } d\text{-SUM instance} \]
\[ \iff \]
\[ P_1, \ldots, P_{d+1} \text{ can be bisected by a linear cut} \]

- ptime-reduction \implies NP-hard
- fpt-reduction \implies W[1]-hard
What we have done

\[ S = \{s_1, \ldots, s_n\} \text{ is a } d\text{-SUM instance} \]

\[ \iff \]

\[ P_1, \ldots, P_{d+1} \text{ can be bisected by a linear cut} \]

- ptime-reduction \Rightarrow NP-hard
- fpt-reduction \Rightarrow W[1]-hard
- linear parameter \Rightarrow n^{\Omega(d)} \text{ (conditional) lower bound}
What we have done

\[ S = \{s_1, \ldots, s_n\} \text{ is a } d\text{-SUM instance} \]

\[ \iff \]

\[ P_1, \ldots, P_{d+1} \text{ can be bisected by a linear cut} \]

- ptime-reduction \implies NP-hard
- fpt-reduction \implies W[1]-hard
- linear parameter \implies n^{\Omega(d)} (conditional) lower bound
In a similar spirit one can show $n^{\Omega(d)}$ lower bounds for
In a similar spirit one can show $n^{\Omega(d)}$ lower bounds for

- Carathéodory sets
In a similar spirit one can show $n^\Omega(d)$ lower bounds for

- Carathéodory sets
- Helly sets (via duality)
Further results

In a similar spirit one can show $n^{\Omega(d)}$ lower bounds for

- Carathéodory sets
- Helly sets (via duality)
- more specific: Minimum Infeasible Subsystem for LP