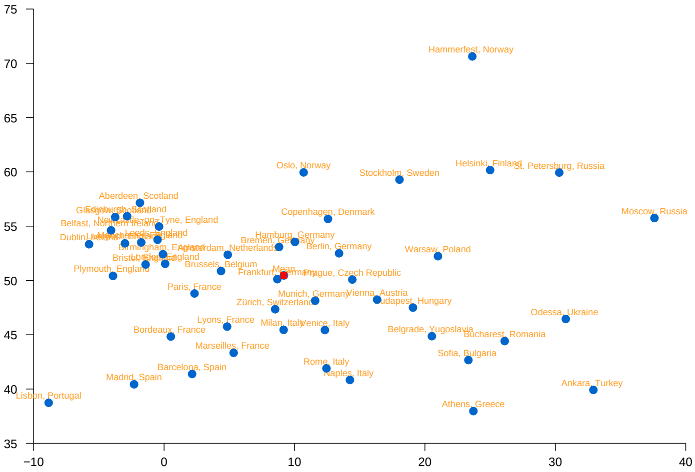
Primal–Dual Algorithms for Halfspace Depth

David Bremner   Komei Fukuda   Vera Rosta

March 17, 2006

Wir sind Zentrum



Estimators of Location

centre Given a set of vectors, return a vector which "best describes" the set.

City	halfplane depth-1
Frankfurt, Germany	19
Brussels, Belgium	17
Munich, Germany	16
Amsterdam, Netherlands	15
Zürich, Switzerland	13
London, England	13
Prague, Czech Republic	12

Estimators of Location

centre Given a set of vectors, return a vector which "best describes" the set.

depth measure Rank a set of vectors such that vectors of maximum rank define one or more centre vectors.

City	halfplane depth-1
Frankfurt, Germany	19
Brussels, Belgium	17
Munich, Germany	16
Amsterdam, Netherlands	15
Zürich, Switzerland	13
London, England	13
Prague, Czech Republic	12

Robustness

- The *breakdown point* of an estimator is the fraction of data that must be moved to infinity before the estimator is also moved to infinity.

Robustness

- The *breakdown point* of an estimator is the fraction of data that must be moved to infinity before the estimator is also moved to infinity.
- The breakdown point of the mean is $\frac{1}{n}$ (i.e. one error suffices to destroy the estimate).

Robustness

- The *breakdown point* of an estimator is the fraction of data that must be moved to infinity before the estimator is also moved to infinity.
- The breakdown point of the mean is $\frac{1}{n}$ (i.e. one error suffices to destroy the estimate).
- The median in $\mathbb{R}^1$ has breakdown $1/2$.

What makes a good depth measure?

Affine Invariant i.e. independant of coordinate system

What makes a good depth measure?

Affine Invariant i.e. independant of coordinate system

Robustness A high breakdown point. For affine invariant measures in $\mathbb{R}^d$,

$$\text{breakdown} \leq 1/d$$

What makes a good depth measure?

Affine Invariant i.e. independant of coordinate system

Robustness A high breakdown point. For affine invariant measures in $\mathbb{R}^d$,

$$\text{breakdown} \leq 1/d$$

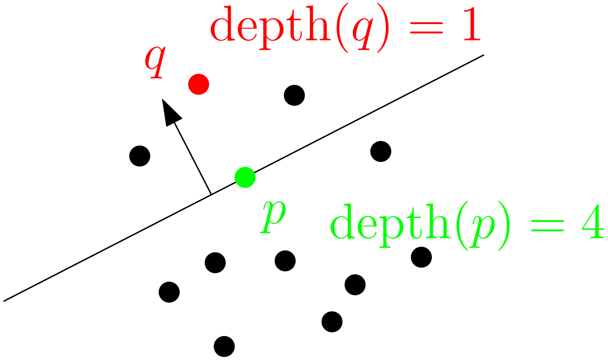
Nesting Let $D_{X,k}$ denote the points of $\mathbb{R}^d$ at depth k with respect to X. We want

$$k > j \implies D_{X,k} \subseteq \operatorname{conv} D_{X,j}$$

Halfspace Depth

The *halfspace depth* of a point q with respect to $S \subset \mathbb{R}^d$ is defined as

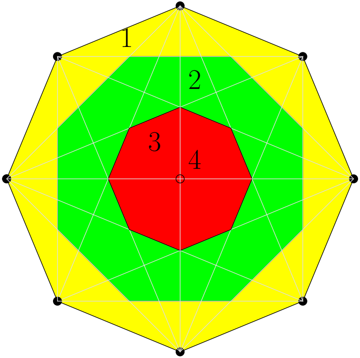
$$\mathrm{depth}_S(q) = \min_{a \in \mathbb{R}^d \setminus 0} |\{p \in S \mid \langle a, p \rangle \geq \langle a, q \rangle\}|$$



Halfspace Depth

The *halfspace depth* of a point q with respect to $S \subset \mathbb{R}^d$ is defined as

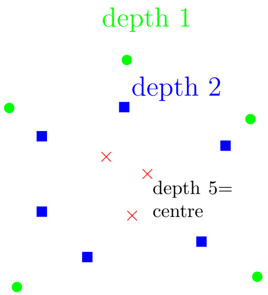
$$\text{depth}_S(q) = \min_{a \in \mathbb{R}^d \setminus 0} |\{p \in S \mid \langle a, p \rangle \geq \langle a, q \rangle\}|$$



Tukey Median

The *Tukey Median* $t(S)$ is defined as

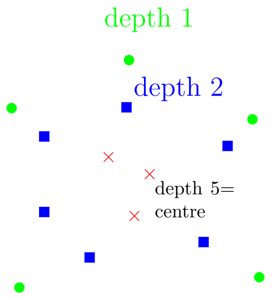
$$\{q \in S \mid \mathrm{depth}_S(q) = \max_{p \in S} \mathrm{depth}_S(p)\}$$



Tukey Median

The *Tukey Median* $t(S)$ is defined as

$$\{q \in S \mid \text{depth}_S(q) = \max_{p \in S} \text{depth}_S(p)\}$$

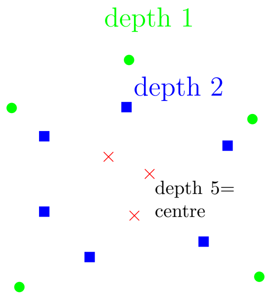


- Halfspace depth is nested

Tukey Median

The *Tukey Median* $t(S)$ is defined as

$$\{q \in S \mid \text{depth}_S(q) = \max_{p \in S} \text{depth}_S(p)\}$$



- Halfspace depth is nested
- The Tukey median has breakdown point at least $1/(d+1)$ for points in general position.

Complexity results

- Halfspace depth is NP-complete, Johnson and Preparata 1978
- Halfspace depth APX-hard Amaldi and Kahn, 1998

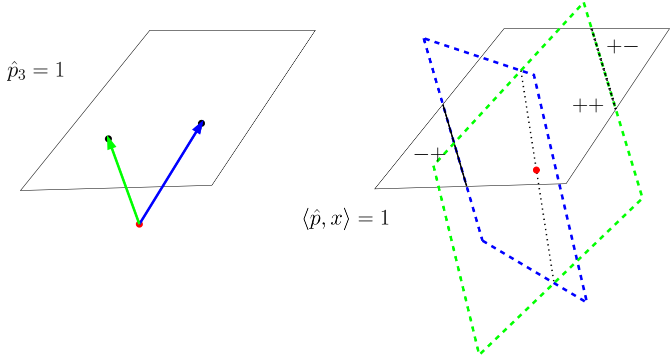
Complexity results

- Halfspace depth is NP-complete, Johnson and Preparata 1978
  - Densest Open and Closed Hemisphere problem
- Halfspace depth APX-hard Amaldi and Kahn, 1998

Complexity results

- Halfspace depth is NP-complete, Johnson and Preparata 1978
  - Densest Open and Closed Hemisphere problem
- Halfspace depth APX-hard Amaldi and Kahn, 1998
  - Maximum Feasible Subsystem
  - A 2-approximation of MFS is possible.

The Dual Arrangement

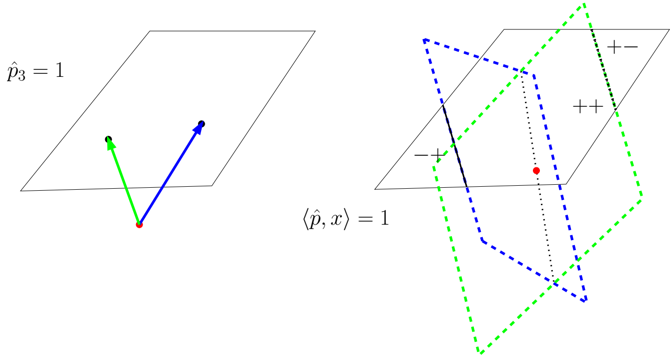


$$\hat{p} = (p, 1) \in \mathbb{R}^{d+1}$$

$$h(p) = \{x \in \mathbb{R}^{d+1} \mid \langle \hat{p}^i, x \rangle = 0\}$$

$$A_p = \{h(q) \mid q \in S \setminus \{p\}\} \cap \{x \mid \langle \hat{p}, x \rangle = 1\}$$

The Dual Arrangement



$$A_p = \{h(q) \mid q \in S \setminus \{p\}\} \cap \{x \mid \langle \hat{p}, x \rangle = 1\}$$

$$\sigma(x) = (\sigma_1 \ldots \sigma_n)$$

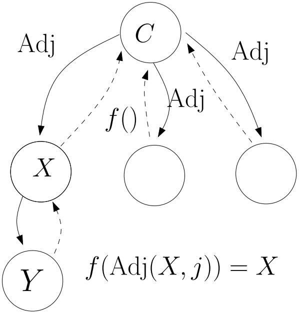
$$\text{where} \quad \sigma_i = \text{sign}(\langle \hat{p}, x \rangle)$$

Reverse Search

- *reverse search* requires two problem specific functions.

Reverse Search

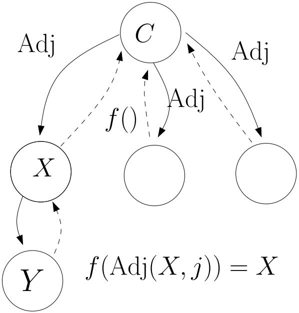
- *reverse search* requires two problem specific functions.
  - The *adjacency oracle* Adj() returns the neighbouring cells



Reverse Search

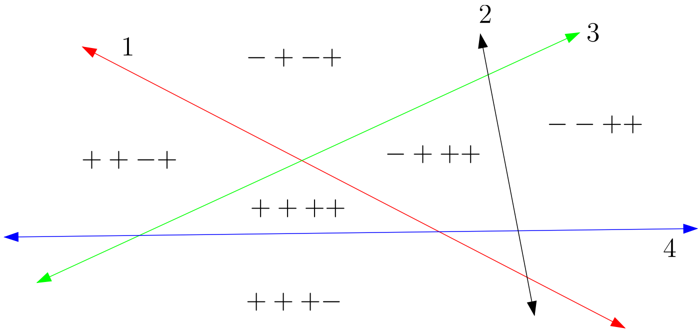
- *reverse search* requires two problem specific functions.
  - The *adjacency oracle* Adj() returns the neighbouring cells
  - The *local search function* $f(\cdot)$ satisfies

$$\exists C\ \forall X\ \exists k\ f^k(X) = C$$



Adjacency Oracle

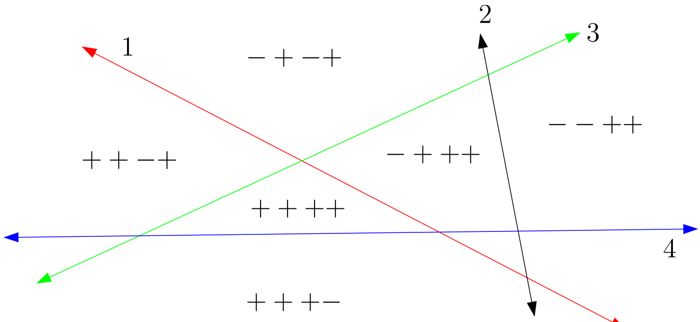
- $X \equiv \sigma(x)$ is *flippable* at position j if negating sign j yields a cell.



$\mathrm{Adj}(++++,1) =$ true $\quad \mathrm{Adj}(++++,2) =$ false

Adjacency Oracle

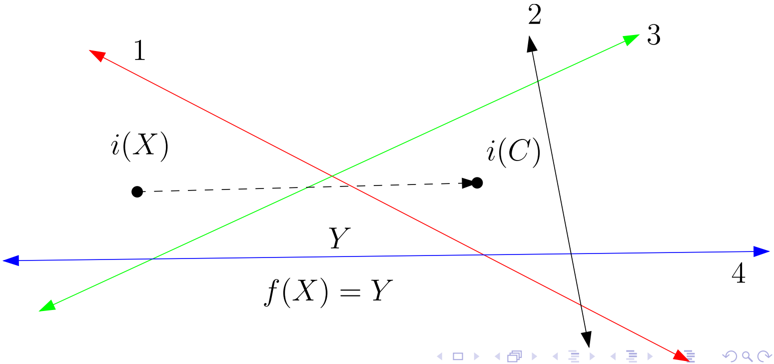
- $X \equiv \sigma(x)$ is *flippable* at position j if negating sign j yields a cell.
- $\mathrm{Adj}(X, j)$ is true iff X is flippable at position j. Solve via LP.



$\mathrm{Adj}(++++, 1) =$ true $\quad \mathrm{Adj}(++++, 2) =$ false

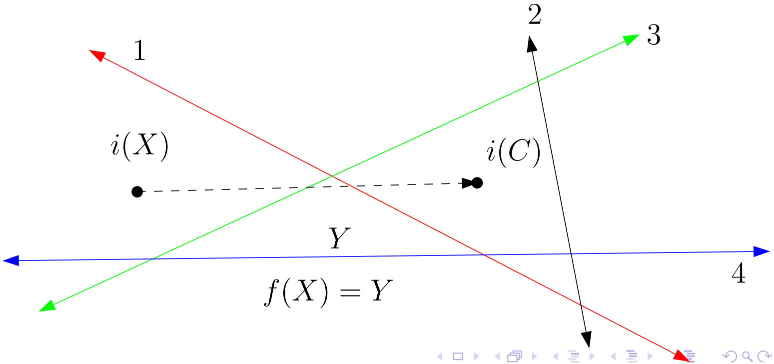
Local Search Function

- Define a canonical interior point $i(X)$ for each cell.



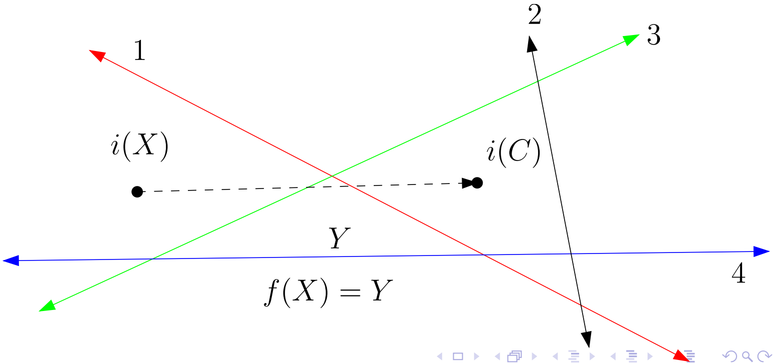
Local Search Function

- Define a canonical interior point $i(X)$ for each cell.
- Choose an arbitrary cell C.
- To find a closer cell to C "shoot a ray" from $i(X)$ to $i(C)$.



Local Search Function

- Define a canonical interior point $i(X)$ for each cell.
- Choose an arbitrary cell C.
- To find a closer cell to C "shoot a ray" from $i(X)$ to $i(C)$.
- Requires a single LP for $i(C)$. $i(X)$ can be computed by flipping test.



Reverse Search Summary

- Time Complexity $O(n \cdot \mathrm{LP}(n, d) \cdot |\mathrm{cells}|)$.

Reverse Search Summary

- Time Complexity $O(n \cdot \mathrm{LP}(n, d) \cdot |\mathrm{cells}|)$.
- Space complexity $O(nd)$.

Reverse Search Summary

- Time Complexity $O(n \cdot \mathrm{LP}(n, d) \cdot |\mathrm{cells}|)$.
- Space complexity $O(nd)$.
- Optimizations include
  - Choosing a shallow start cell
  - Pruning the search.

Reverse Search Summary

- Time Complexity $O(n \cdot \mathrm{LP}(n, d) \cdot |\mathrm{cells}|)$.
- Space complexity $O(nd)$.
- Optimizations include
  - Choosing a shallow start cell
  - Pruning the search.
- Implementation uses `ZRAM` for reverse-search framework, `cddlib` to solve small, dense LPs.
- Parallelization is no extra implementation effort with ZRAM, and speedup is linear.

Reverse Search Summary

- Time Complexity $O(n \cdot \mathrm{LP}(n, d) \cdot |\mathrm{cells}|)$.
- Space complexity $O(nd)$.
- Optimizations include
  - Choosing a shallow start cell
  - Pruning the search.
- Implementation uses `ZRAM` for reverse-search framework, `cddlib` to solve small, dense LPs.
- Parallelization is no extra implementation effort with ZRAM, and speedup is linear.
- Little information until enumeration terminates.

Primal–Dual Algorithms

- Update at a every step an upper bound and a lower bound for the depth.
- Terminate when (if) bounds are equal

Primal–Dual Algorithms

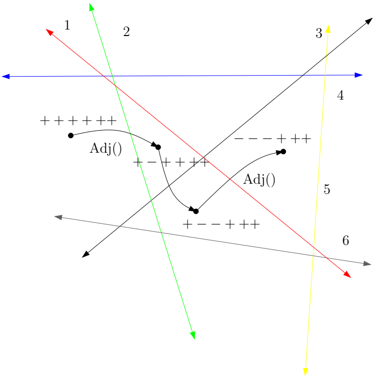
- Update at a every step an upper bound and a lower bound for the depth.
- Terminate when (if) bounds are equal
- To ensure termination, fall back on enumeration after a fixed time limit.

Primal–Dual Algorithms

- Update at a every step an upper bound and a lower bound for the depth.
- Terminate when (if) bounds are equal
- To ensure termination, fall back on enumeration after a fixed time limit.
- Generally, answers improve with time.

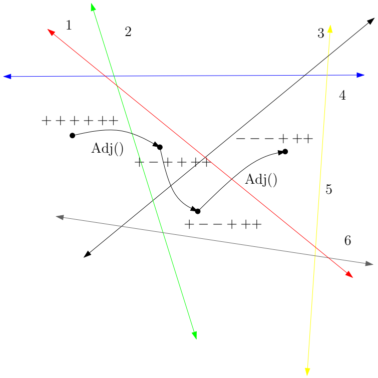
Upper Bounds via Random Walks

- Use Adj() oracle from enumeration algorithm



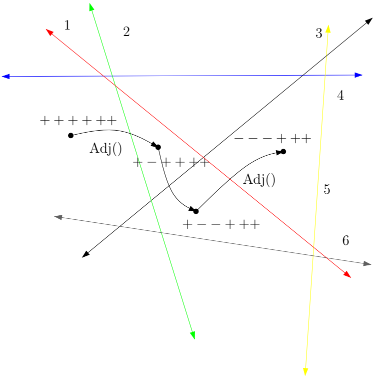
Upper Bounds via Random Walks

- Use Adj() oracle from enumeration algorithm
- Greedily try to reduce number of $+$ in σ until local minimum reached.



Upper Bounds via Random Walks

- Use Adj() oracle from enumeration algorithm
- Greedily try to reduce number of $+$ in σ until local minimum reached.
- Repeat several times choosing a random starting cell.

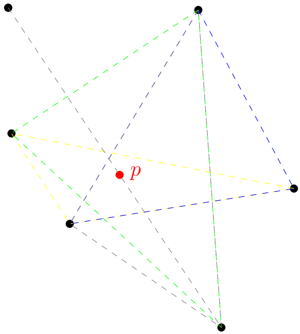


Minimal Dominating Sets

Definition

A *Minimal Dominating Set* (MDS) for $p \in \mathbb{R}^d$ with respect to $S \subset \mathbb{R}^d$ is $R \subseteq S$ such that

.



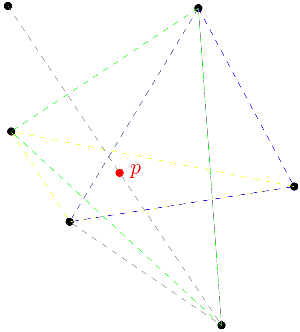
Minimal Dominating Sets

Definition

A *Minimal Dominating Set* (MDS) for $p \in \mathbb{R}^d$ with respect to $S \subset \mathbb{R}^d$ is $R \subseteq S$ such that

- $p \in \operatorname{conv} R$

.



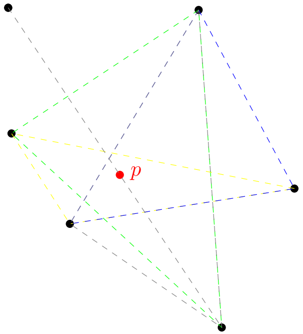
Minimal Dominating Sets

Definition

A *Minimal Dominating Set* (MDS) for $p \in \mathbb{R}^d$ with respect to $S \subset \mathbb{R}^d$ is $R \subseteq S$ such that

- $p \in \operatorname{conv} R$
- if $R' \subsetneq R$ then $p \notin \operatorname{conv} R'$.

An MDS might also be called a Charathéodory set.



Lower bounds via MDS's

Proposition

Let Δ be the set of all MDS's for p with respect to S. Let T be a minimum transversal (hitting set) of Δ.

$$|T| = \text{depth}(p)$$

Lower bounds via MDS's

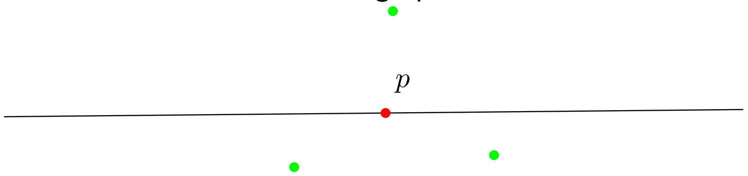
Proposition

Let Δ be the set of all MDS's for p with respect to S. Let T be a minimum transversal (hitting set) of Δ.

$$|T| = \text{depth}(p)$$

$|T| \leq \text{depth}(p)$

Each MDS intersects both closed sides of any hyperplane through p.



Lower bounds via MDS's

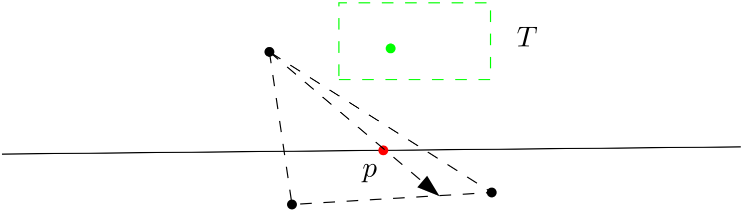
Proposition

Let Δ be the set of all MDS's for p with respect to S. Let T be a minimum transversal (hitting set) of Δ.

$$|T| = \text{depth}(p)$$

Suppose $|T| < \text{depth}(p)$

if T is separable from p, there is an uncovered MDS.



Lower bounds via MDS's

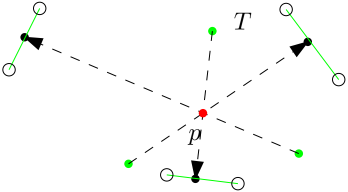
Proposition

Let Δ be the set of all MDS's for p with respect to S. Let T be a minimum transversal (hitting set) of Δ.

$$|T| = \text{depth}(p)$$

Suppose $|T| < \text{depth}(p)$

if T is not separable from P, there is a totally covered MDS.



Generating Missed MDSs (cuts)

Definition

Given a partial traversal T for the MDS's of p w.r.t. S Define $\bar{S} = S \setminus T$. Define the *auxiliary polytope* $Q(p, T)$ as λ satisfying:

$$\lambda \bar{S} = p$$

$$\sum_i \lambda_i = 1 \qquad \lambda_i \geq 0$$

- Each vertex (basic solution) of $Q(p, T)$ defines an MDS missed by T.

Generating Missed MDSs (cuts)

Definition

Given a partial traversal T for the MDS's of p w.r.t. S Define $\bar{S} = S \setminus T$. Define the *auxiliary polytope* $Q(p, T)$ as λ satisfying:

$$\lambda \bar{S} = p$$

$$\sum_i \lambda_i = 1 \qquad\qquad \lambda_i \geq 0$$

- Each vertex (basic solution) of $Q(p, T)$ defines an MDS missed by T.
- A single cut can be found by LP

Generating Missed MDSs (cuts)

Definition

Given a partial traversal T for the MDS's of p w.r.t. S Define $\bar{S} = S \setminus T$. Define the *auxiliary polytope* $Q(p, T)$ as λ satisfying:

$$\lambda \bar{S} = p$$

$$\sum_i \lambda_i = 1 \qquad \lambda_i \geq 0$$

- Each vertex (basic solution) of $Q(p, T)$ defines an MDS missed by T.
- A single cut can be found by LP
- k cuts can be found via reverse search (or other pivoting method).

Primal–Dual Algorithm

1. Walk in the dual arrangement to find a cell with minimal positive support

Primal–Dual Algorithm

1. Walk in the dual arrangement to find a cell with minimal positive support
2. Find obstructions (i.e. MDS's) to the optimality of this cell via the "auxiliary polytope"

Primal–Dual Algorithm

1. Walk in the dual arrangement to find a cell with minimal positive support
2. Find obstructions (i.e. MDS's) to the optimality of this cell via the "auxiliary polytope"
3. If none found, report optimal (we have solved the global minimum transversal problem).

Primal–Dual Algorithm

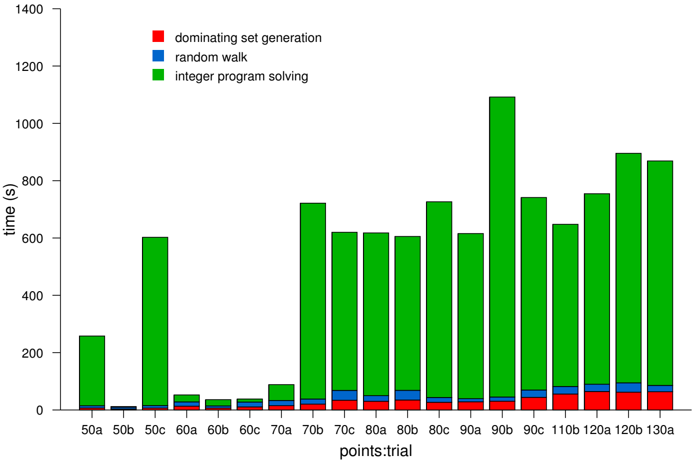
1. Walk in the dual arrangement to find a cell with minimal positive support
2. Find obstructions (i.e. MDS's) to the optimality of this cell via the "auxiliary polytope"
3. If none found, report optimal (we have solved the global minimum transversal problem).
4. Otherwise solve the resulting (partial) transversal problem via integer programming.

Primal–Dual Algorithm

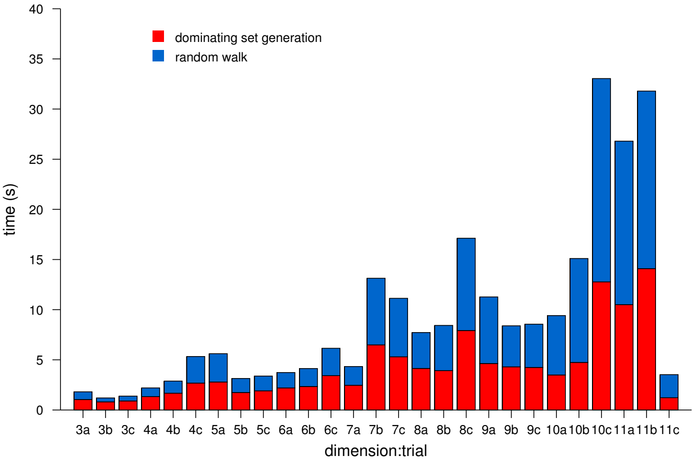
1. Walk in the dual arrangement to find a cell with minimal positive support
2. Find obstructions (i.e. MDS's) to the optimality of this cell via the "auxiliary polytope"
3. If none found, report optimal (we have solved the global minimum transversal problem).
4. Otherwise solve the resulting (partial) transversal problem via integer programming.
5. If bored, switch to enumeration.

Software

Software	Where	Purpose
cdd	http://www.ifor.math.ethz.ch/~fukuda/cdd_home	Solving Dense LPs
COIN	http://www.coin-or.org	Simplex Solver, cut generation
lrs	http://cgm.cs.mcgill.ca/~avis/	MDS generation.
SYMPHONY	http://www.branchandcut.org	Branch and Cut
ZRAM	http://www.cs.unb.ca/~bremner/zram	Parallel reverse search

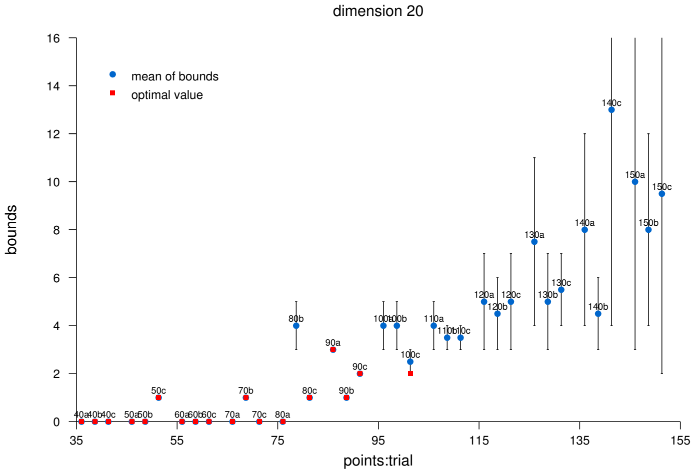


Dimension 10

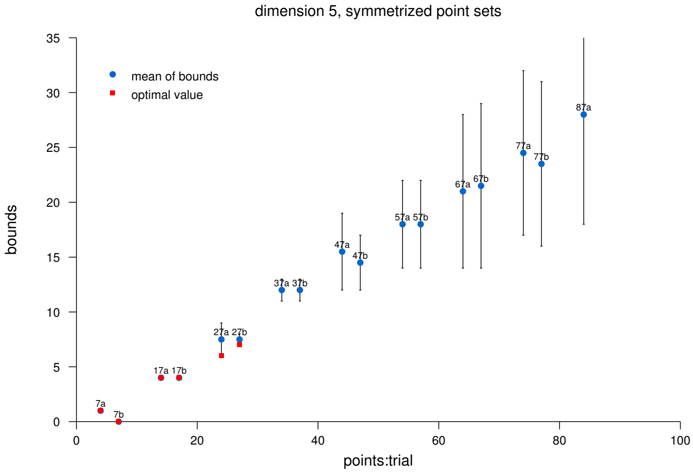


dominating set generation
random walk
time (s)
dimension:trial
0
5
10
15
20
25
30
35
40
3a 3b 3c 4a 4b 4c 5a 5b 5c 6a 6b 6c 7a 7b 7c 8a 8b 8c 9a 9b 9c 10a 10b 10c 11a 11b 11c

50 points



dimension 20
mean of bounds
optimal value
bounds
points:trial


dimension 5, symmetrized point sets
mean of bounds
optimal value
bounds
points:trial
7a
7b
17a 17b
27a 27b
37a 37b
47a
47b
57a 57b
67a 67b
77a
77b
87a
0
5
10
15
20
25
30
35
0
20
40
60
80
100

Conclusions

- Row generation is crucial to solve the IPs.

Conclusions

- Row generation is crucial to solve the IPs.
- Restarting IP solver should be avoided, integrate cut generation.

Conclusions

- Row generation is crucial to solve the IPs.
- Restarting IP solver should be avoided, integrate cut generation.
- Better upper bounds would nice.