Solving k-SUM using few linear queries

Jean Cardinal, John Iacono and Aurélien Ooms

1

Definition. Let k be a positive integer. Given n numbers, decide whether any k of them sum to zero.

Example. $(k = 5, n = 11)$

$\{ -975, -505, -430, -237, -178, -29, 67, 439, 660, 674, 898 \}$

Time complexity.

k even — $O(n^{\frac{k}{2}} \log n)$

k odd — $O(n^{\frac{k+1}{2}})$

Hardness. *assuming ETH holds

Pătrașcu & Williams '10 *No $n^{o(k)}$ algorithm for k-SUM!**

Abboud et al. '14 *k-SUM is $W[1]$-complete!*

2

s-linear decision trees.

$\sum_{j=1}^{s} \alpha_j x_{i_j} \leq^? t$ s-linear query

$s = S(n)$ is the "query size"

Two-track complexity.

query complexity $= Q(n)$

"# of times we make a linear query"

"other operations"

word-RAM time complexity $= T(n)$

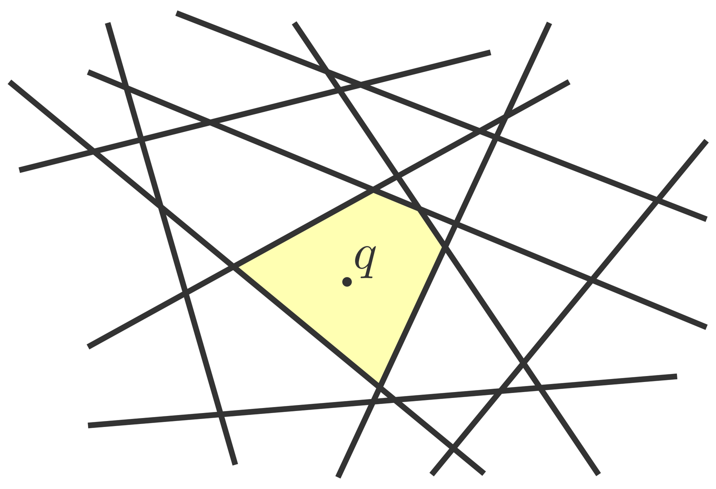
⚠ No manipulation of the input!

3

Goal. With queries as small as possible, $f(k)n^{O(1)}$ queries and $n^{\lceil \frac{k}{2} \rceil + O(1)}$ word-RAM running time.

	Results.	$S(n)$	$Q(n)$	$T(n)$
	Folklore	k	$n^{\lceil \frac{k}{2} \rceil}$	$n^{\lceil \frac{k}{2} \rceil}$
LB	Erickson '99	k	$\Omega(n^{\lceil \frac{k}{2} \rceil})$	$\Omega(n^{\lceil \frac{k}{2} \rceil})$
	Meiser '93	n	n^3	2^n
	New	$o(n)$	n^3	$n^{\lceil \frac{k}{2} \rceil + 8}$

Reduction to a point location problem. Given an input point $q \in \mathbb{R}^n$ and m hyperplanes, decide whether q lies on any of the hyperplanes.



n input numbers $(q_1, q_2, \ldots, q_n)$

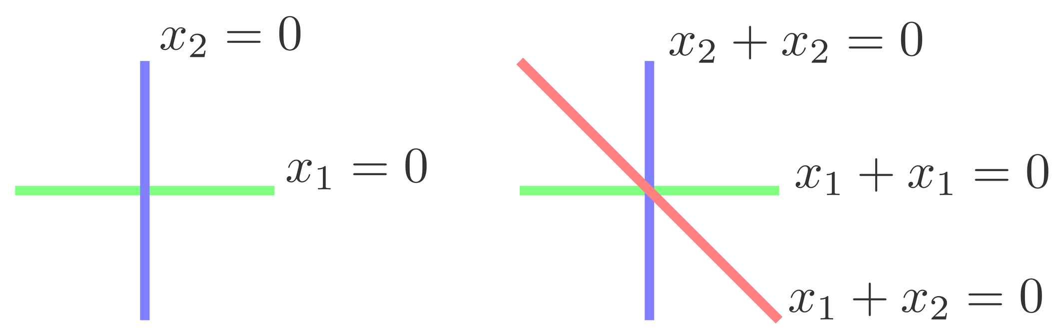
↳ an input point in $\mathbb{R}^n$

$O(n^k)$ k-tuple sums $\sum_{j=1}^{k} x_{i_j} = 0$

↳ $O(n^k)$ hyperplanes

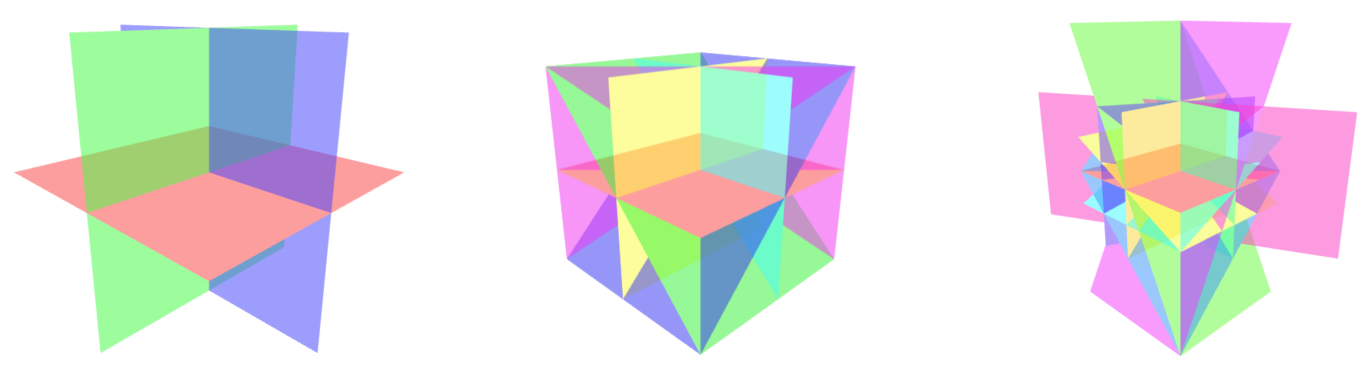
Examples of k-SUM arrangements.

$n = 2$



1SUM 2SUM

$n = 3$

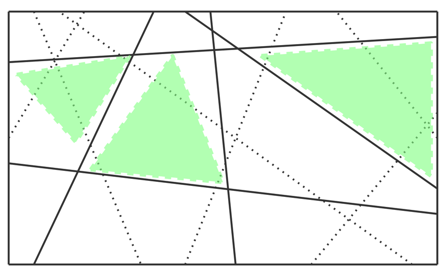


1SUM 2SUM 3SUM

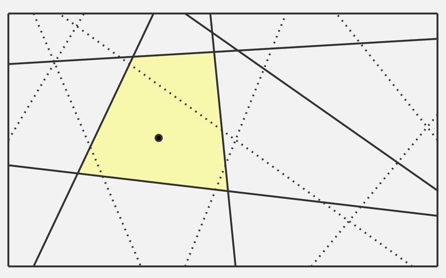
New Algorithm. Divide the problem into a few $o(n)$-sized subproblems for which we prune and search using ε-nets. We first pick a large ε-net to filter out most of the hyperplanes then fall back to smaller ε-nets.

Meiser's algorithm.*

1. Pick an ε-net $\mathcal{N}$ of size $O(n^2 \log^2 n)$.



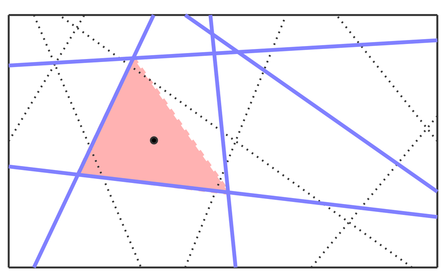
$T(n) = O(n^2 \log^3 n)$
no queries



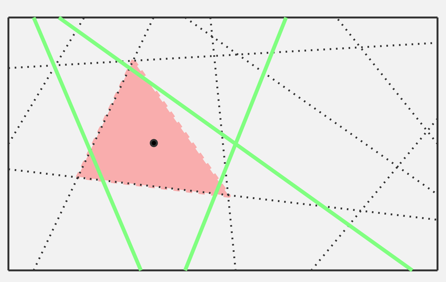
2. Compute the cell C of the arrangement $A(\mathcal{N})$ that contains q using linear queries.

$T(n) = O(n^2 \log^2 n)$
$Q(n) = O(n^2 \log^2 n)$
$S(n) = k$

3. Compute a simplex S inscribed in C and containing q using LP and ray-shooting.



$T(n) = n^{O(1)}$
$Q(n) = O(n^3 \log^2 n)$
$S(n) = o(n)$

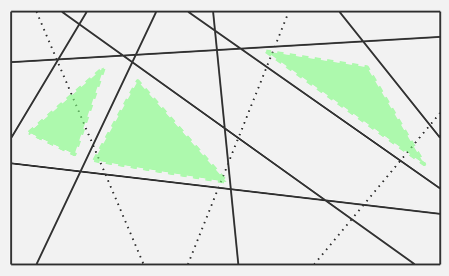


4. Filter then recurse on at most εm hyperplanes meeting S.

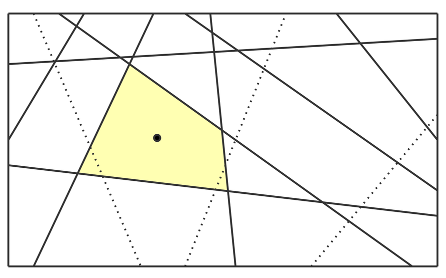
$T(n) = \tilde{O}(mn^2)$
$O(\log m)$ cumulated recursive calls.

Improved algorithm.

1. Pick an ε-net $\mathcal{N}$ of size $O(n^{k/2+2} \log^2 n)$.



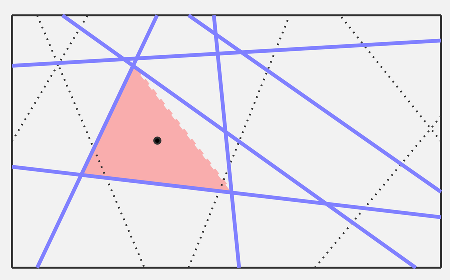
$T(n) = O(n^{k/2+2} \log^3 n)$
no queries



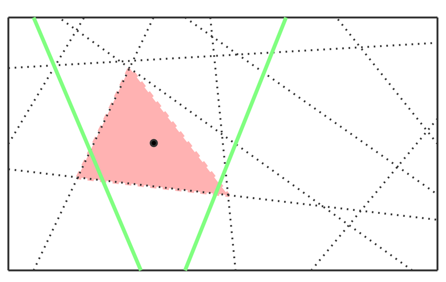
2. Compute the cell C of the arrangement $A(\mathcal{N})$ that contains q using Meiser's algorithm.

$T(n) = \tilde{O}(n^{k/2+4})$
$Q(n) = O(n^3 \log^2 n)$
$S(n) = o(n)$

3. Compute a simplex S inscribed in C and containing q using simplices of step 2.



$T(n) = n^{O(1)}$
$Q(n) = O(n^2 \log n)$
$S(n) = o(n)$



4. Filter then call Meiser's algorithm on $O(n^{\frac{k}{2}})$ hyperplanes meeting S.

$T(n) = \tilde{O}(n^{\lceil \frac{k}{2} \rceil + 8})$
$Q(n) = O(n^3 \log^3 n)$
$S(n) = o(n)$

*updated with our results