Online Algos: Old and New

Lecture 4: Search Problems

Anupam Gupta (NYU)

NYU | COURANT

lecture plan

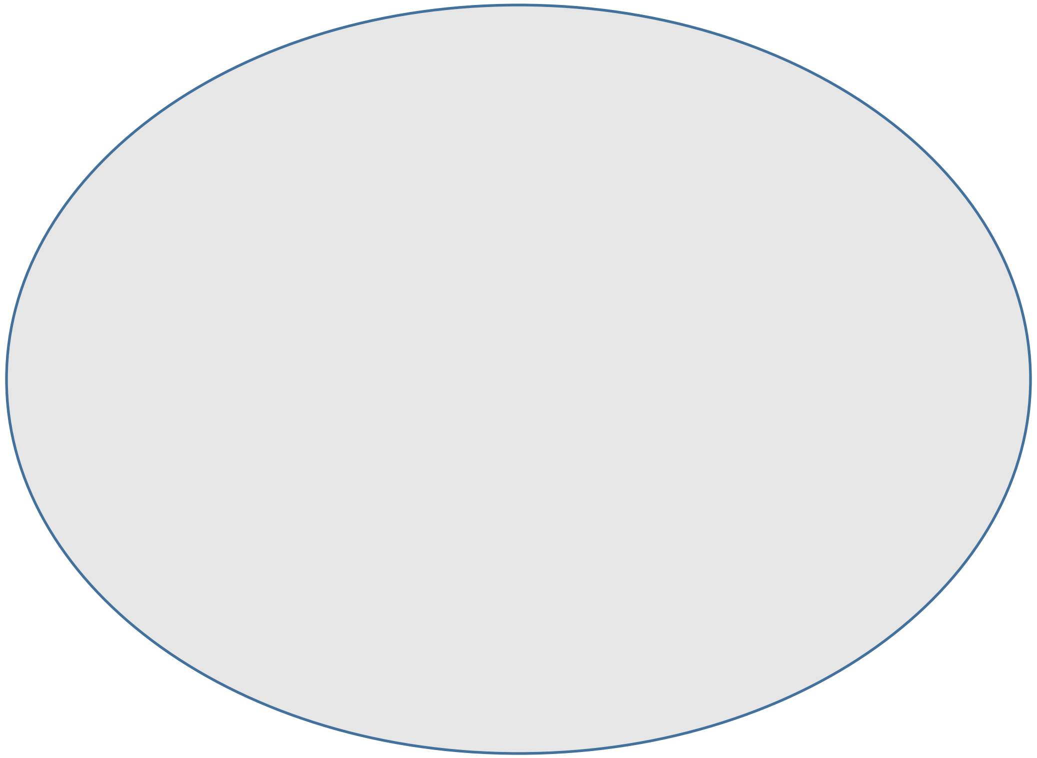
Lecture #1: Set Cover (worst case)

Lecture #2: Set Cover (beyond worst case), Network design (both)

Lecture #3: Resource Allocation (aka packing)

Lecture #4: Search Problems (aka chasing)

Metrical Task System = Function Chasing



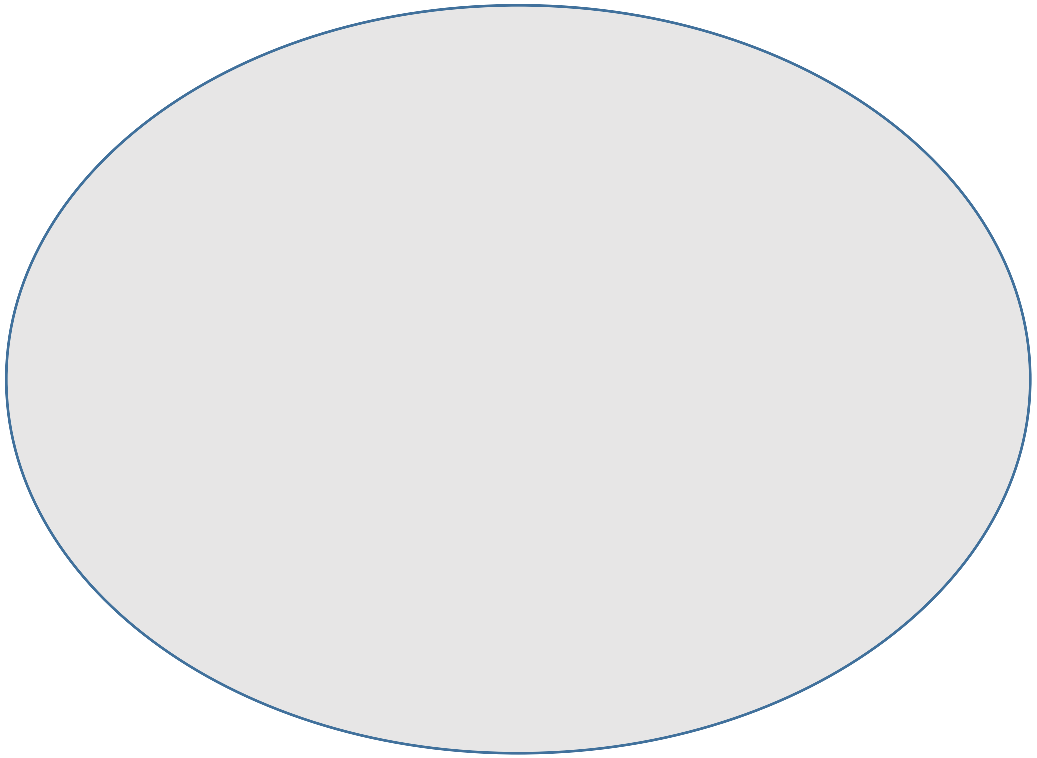
Metric space (X, d), $x_0 \in X$

@ time t:
- see cost function $f_t: X \to \mathbb{R}_{\geq 0}$
- play point $x_t \in X$

cost = $\sum_t f_t(x_t) + d(x_t, x_{t-1})$

[Borodin Linial Saks 92]

Metrical Service System = Set Chasing



Metric space (X, d), $x_0 \in X$

@ time t:
see subset $S_t \subseteq X$
play point $x_t \in S_t$

cost = $\sum_t d(x_t, x_{t-1})$

[Borodin Linial Saks 92]

Set Chasing

Uniform Metric

Uniform Metric Set Chasing

n points, unit distance from each other

see set $S_t \subseteq [n]$ at each timestep, must move to $x_t \in S_t$

minimize number of moves

Algo 0: Move to arbitrary point in S_t

$x_0 = 1$

$S_1 = \{2,3,4,5 \dots n\}$ $x_1 = 2$

$S_2 = \{1,3,4,5, \dots n\}$ $x_2 = 1$

$S_3 = \{2,3,4,5 \dots n\}$ $x_3 = 2$

$S_4 = \{1,3,4,5, \dots n\}$ $x_4 = 1$

⋮

OPT = 1

E[ALG] = $\Theta(n)$

Uniform Metric Set Chasing

n points, unit distance from each other

see set $S_t \subseteq [n]$ at each timestep, must move to $x_t \in S_t$

minimize number of moves

Algo 0: Move to arbitrary point in S_t

Algo 1: Move to random point in S_t

Adversary:

Pick random goal $g \in \{1 \ldots n\}$

for $t = 1\ to\ n^{10}$
    pick random h from $[n]$
    define $S_t = ([n] \setminus \{h\}) \cup \{g\}$

OPT = 1

E[ALG] = $\Theta(n)$

Uniform Metric Set Chasing

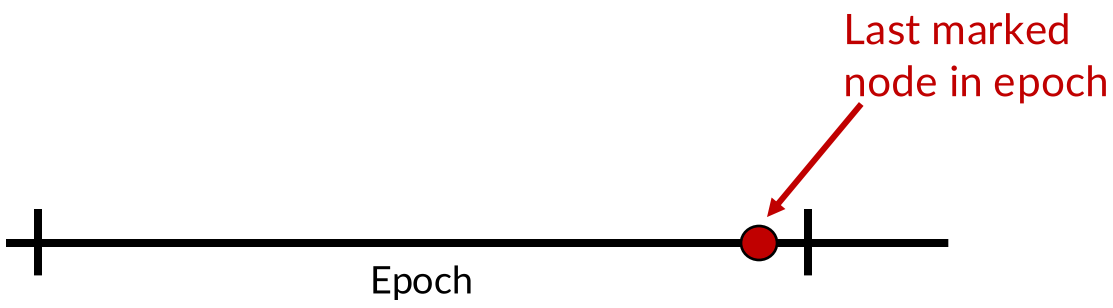
n points, unit distance from each other

see set $S_t \subseteq [n]$ at each timestep, must move to $x_t \in S_t$

minimize number of moves

Algo 2: Random marking

Thm 1: Random marking is $O(\log n)$ competitive



Random Marking

At time t:
  Mark all points not in S_t
  If $x_{t-1} \in S_t$ set $x_t \leftarrow x_{t-1}$
  else
    $x_t =$ random unmarked pt in S_t

If all points marked, unmark all

Epoch: times between two unmarks

OPT ≥ 1 in an epoch

ALG $\leq O(\log n)$

Uniform Metric Set Chasing

n points, unit distance from each other

see set $S_t \subseteq [n]$ at each timestep, must move to $x_t \in S_t$

minimize number of moves

Thm 1: Random marking is $O(\log n)$ competitive

Thm 2: Every (rand) algorithm is $\Omega(\log n)$ competitive

Yao's lemma

"antiset"

$S_t = [\mathrm{n}] \setminus \text{uniformly random node}$

Epoch ends when all pages have been antiset in this epoch

Epoch length = nH_n

OPT $\cong 1$ per epoch (move to last antiset)

E[ALG] pays $1/n$ per step

Uniform Metric Function Chasing

n points, unit distance from each other

see set $f_t: [n] \to R_+$ at each timestep, must play x_t

minimize $f_t(x_t) + 1(x_t \neq x_{t-1})$

Thm 1: variant of Random marking is $O(\log n)$ competitive

Thm 2: Every (rand) algorithm is $\Omega(\log n)$ competitive (follows from previous lower bound)

General Metric Function Chasing

General metric space (V, d) on n points

Each time function $f_t: V \to \mathbb{R}_+$

must play x_t, pay $f_t(x_t) + d(x_t, x_{t-1})$

Thm 3: $O(\log^2 n)$ competitive algo [Bubeck Cohen Lee Lee 17, Coester Lee 19]

Thm 4: Every (rand) algorithm is $\Omega(\log^2 n)$ competitive [Bubeck Coester Rabani 22]

Further directions

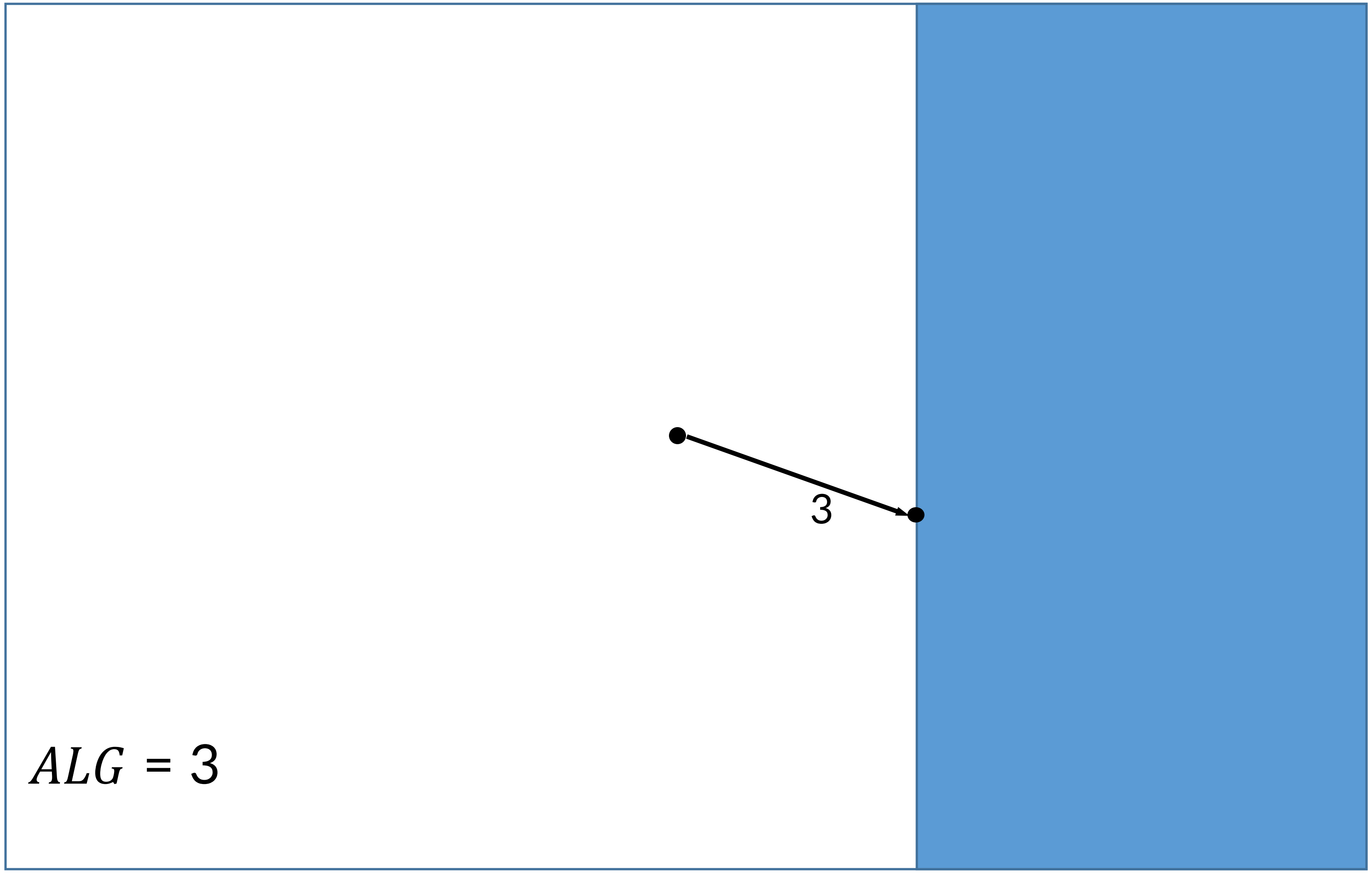
Many of these ideas extend to paging and k-server:

k servers moving in a metric space

requests arrive at location, must choose which server to move to it

Convex Function Chasing

Convex Body Chasing: definition



algorithm controls point in $\mathbb{R}^d$

at time t:

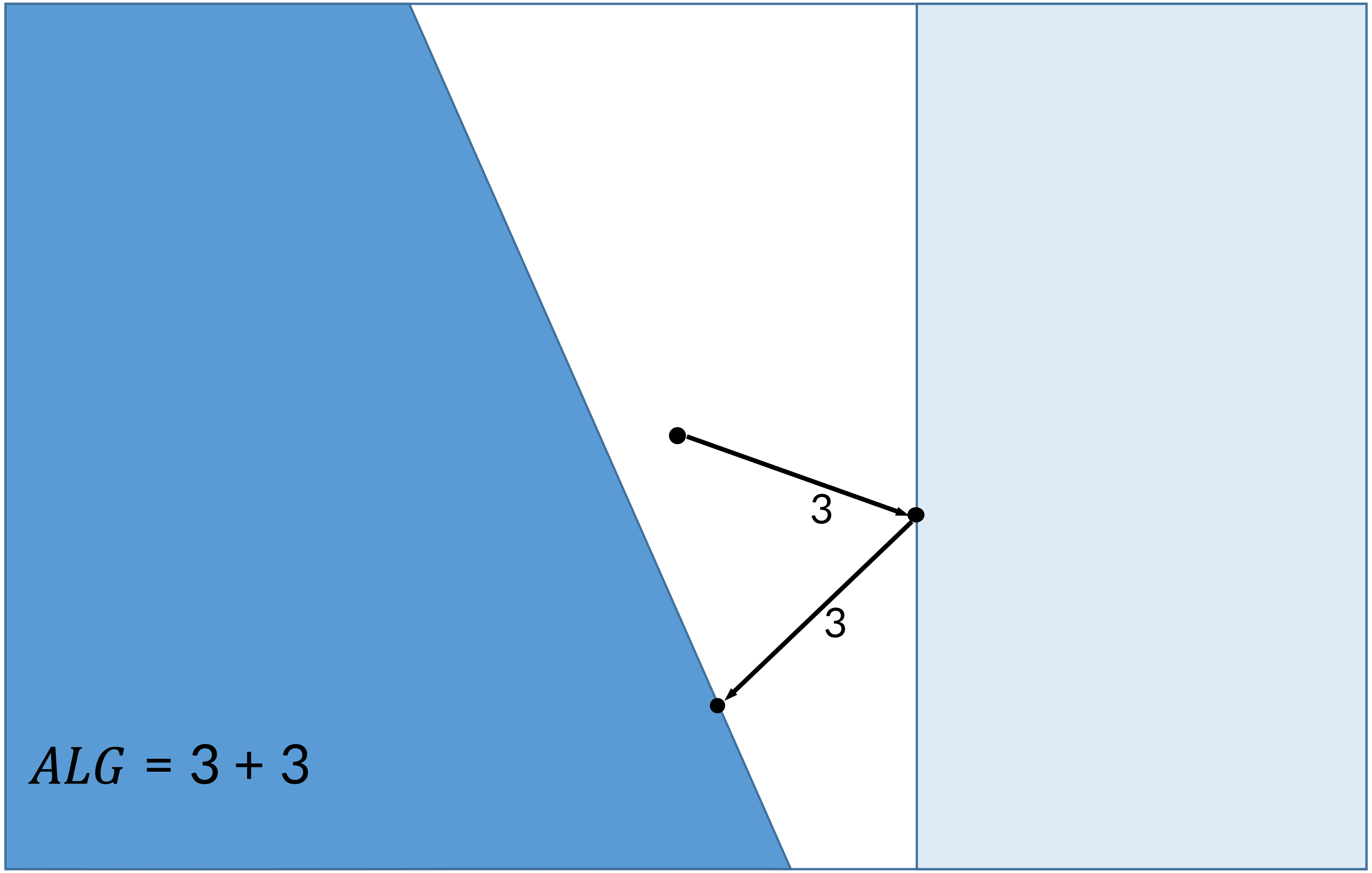
convex body K_t arrives

algorithm moves to x_t

pays $|| x_t - x_{t-1} ||$

[Friedman Linial 94]

Convex Body Chasing: definition



algorithm controls point in $\mathbb{R}^d$

at time t:

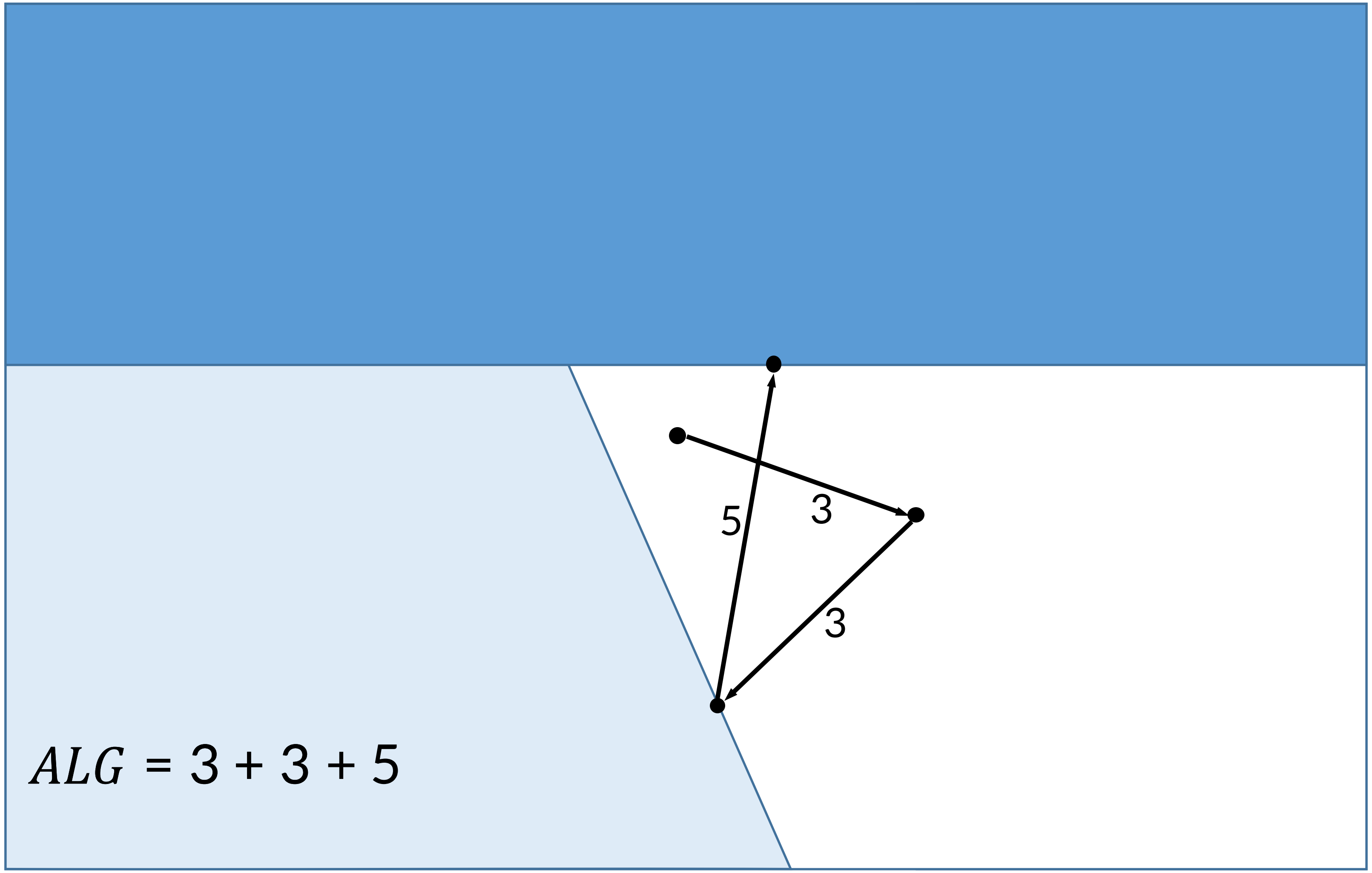
convex body K_t arrives

algorithm moves to x_t

pays $|| x_t - x_{t-1} ||$

[Friedman Linial 94]

Convex Body Chasing: definition



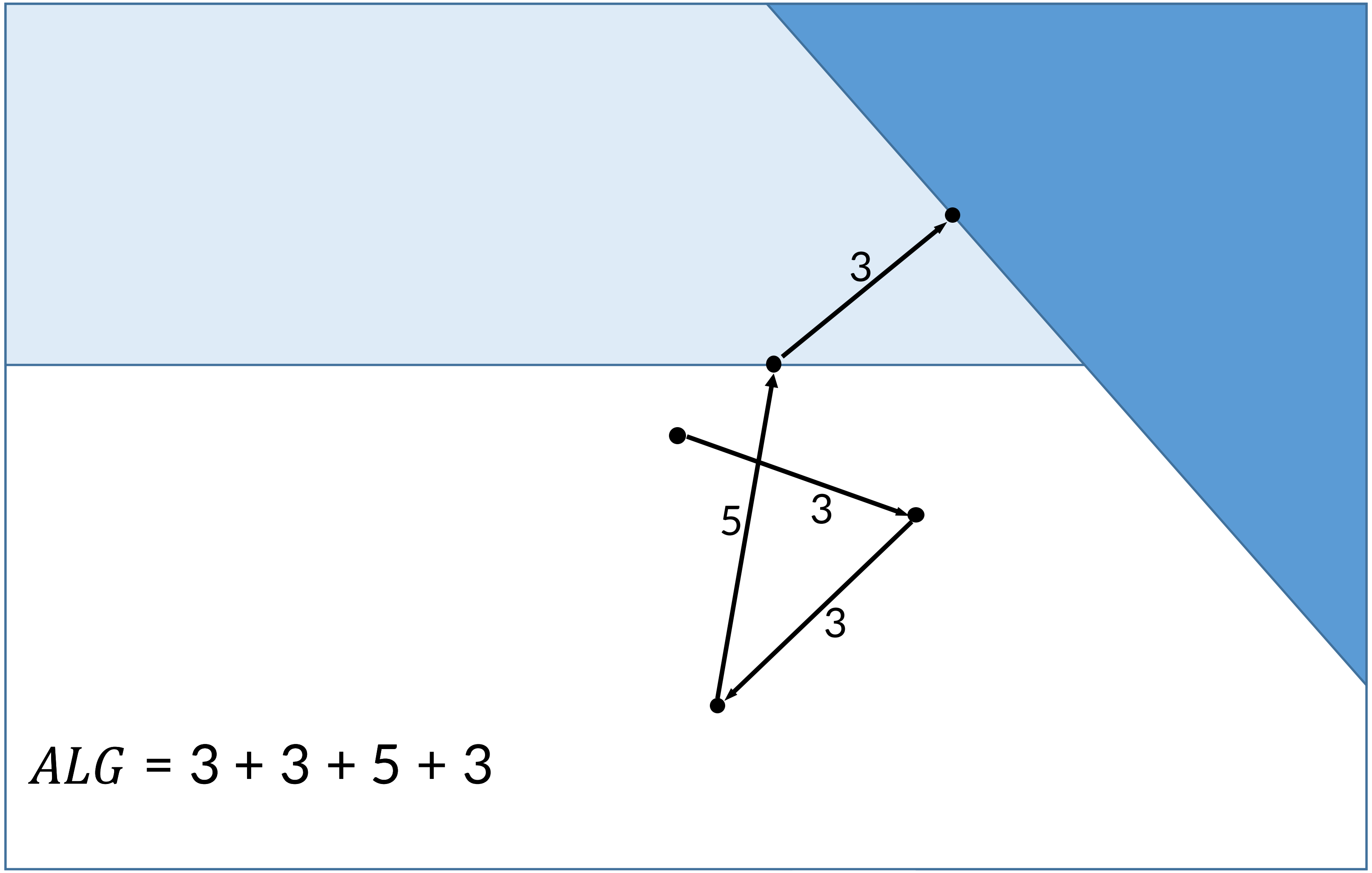
algorithm controls point in $\mathbb{R}^d$

at time t:

- convex body K_t arrives
- algorithm moves to x_t
- pays $|| x_t - x_{t-1} ||$

[Friedman Linial 94]

Convex Body Chasing: definition



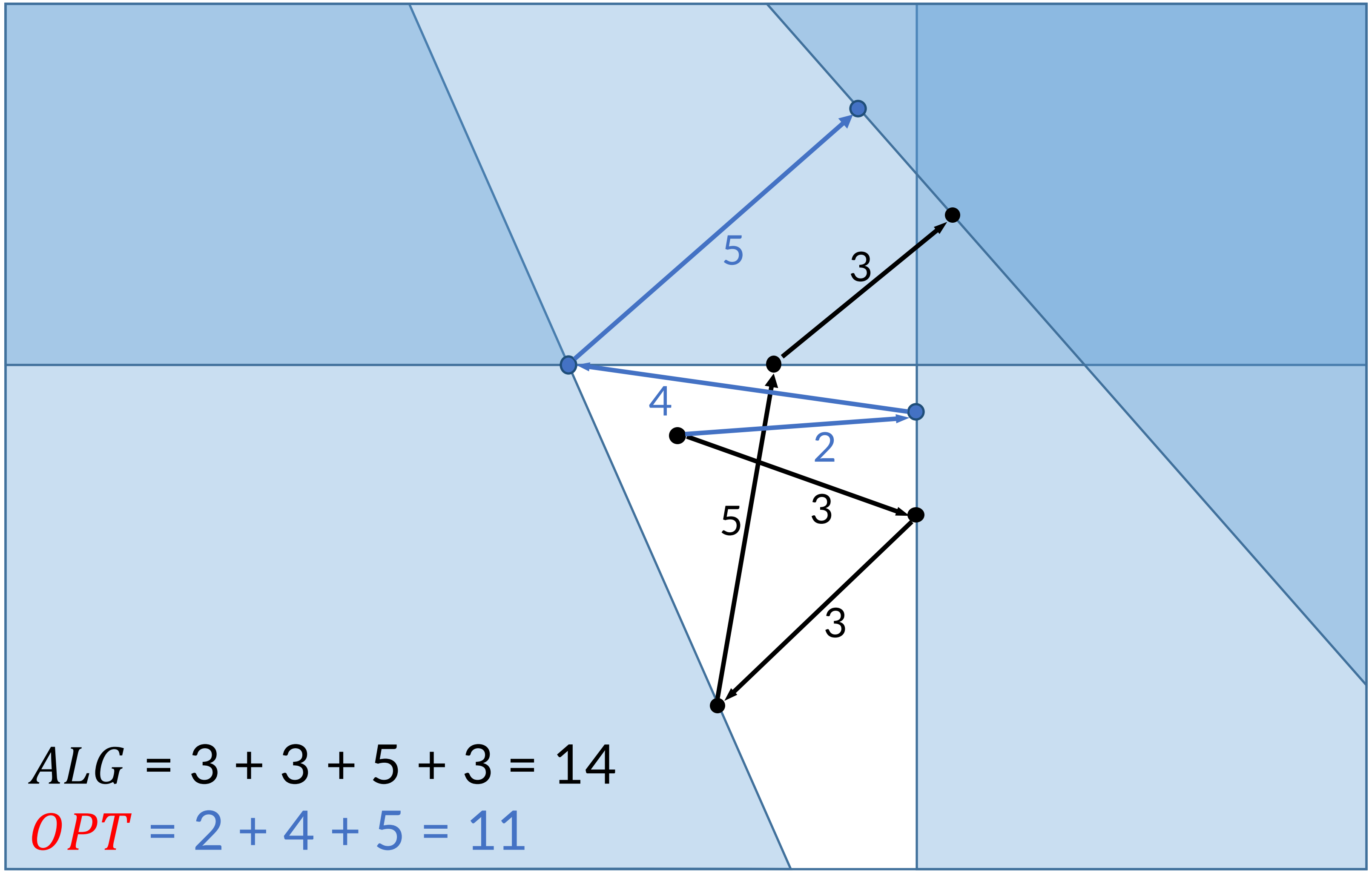
algorithm controls point in $\mathbb{R}^d$

at time t:

- convex body K_t arrives
- algorithm moves to x_t
- pays $|| \, x_t - x_{t-1} \, ||$

[Friedman Linial 94]

Convex Body Chasing: definition



algorithm controls point in $\mathbb{R}^d$

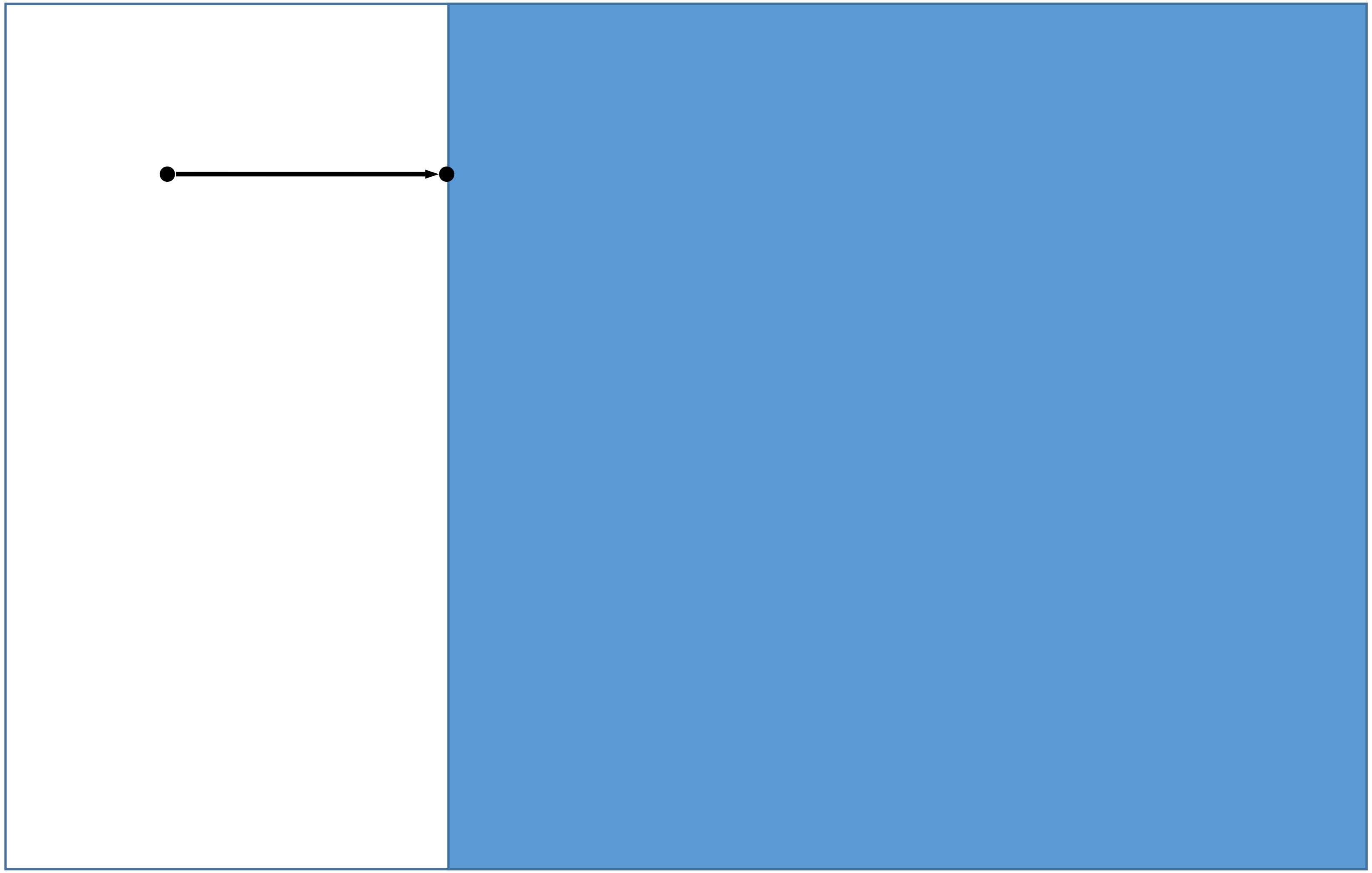
at time t:

- convex body K_t arrives
- algorithm moves to x_t
- pays $|| x_t - x_{t-1} ||$

$$\mathrm{cr}(ALG) \coloneqq \max_{\sigma} \frac{ALG(\sigma)}{OPT(\sigma)}$$

[Friedman Linial 94]

Nested Version



algorithm controls point in $\mathbb{R}^d$

at time t:

convex body K_t arrives

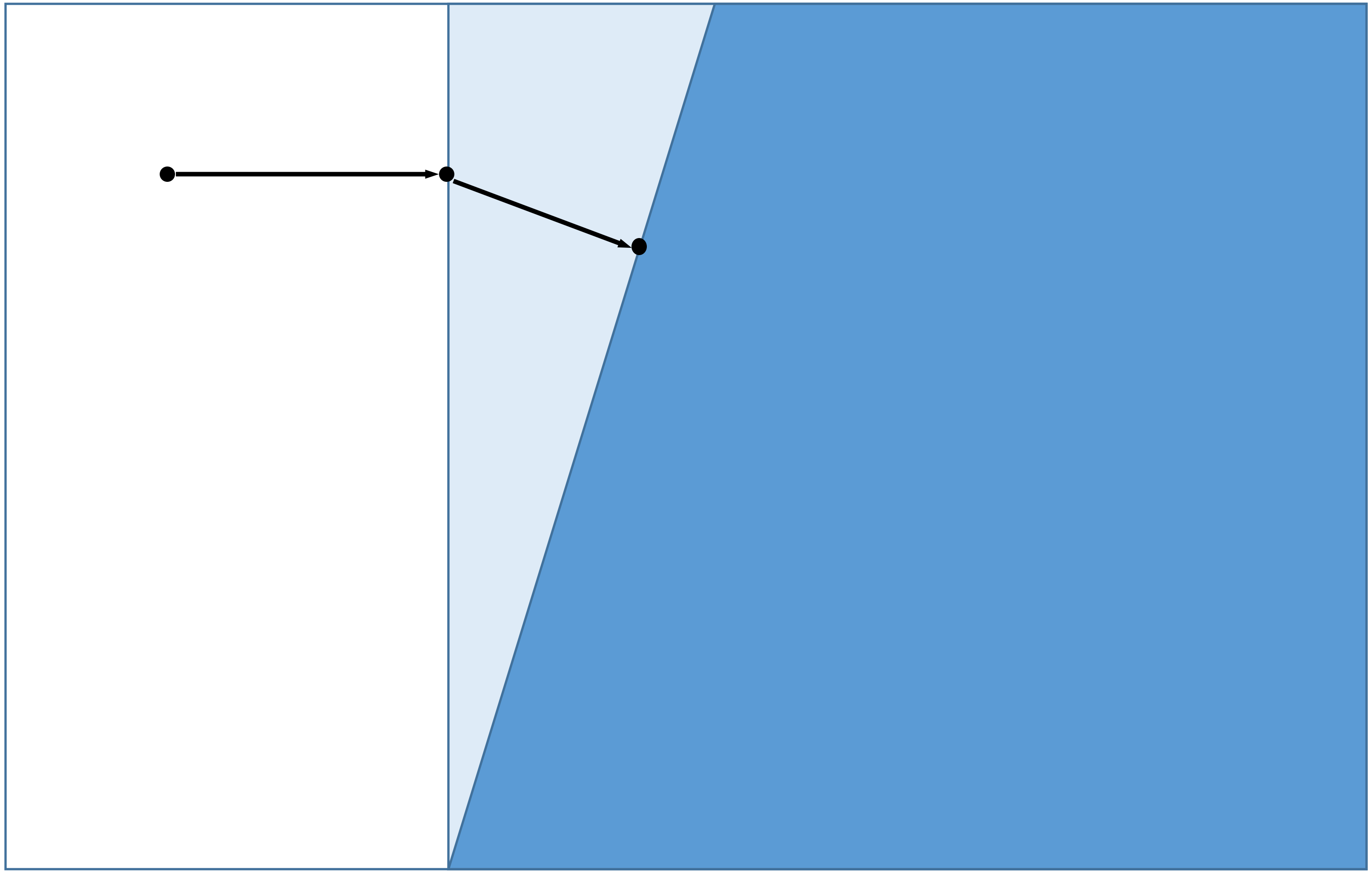
algorithm moves to x_t

pays $|| x_t - x_{t-1} ||$

nested: $K_t \subseteq K_{t-1}$

[Bansal Bohm Elias Koumoutsos Umboh SODA 2018]

Nested Version



algorithm controls point in $\mathbb{R}^d$

at time t:

convex body K_t arrives

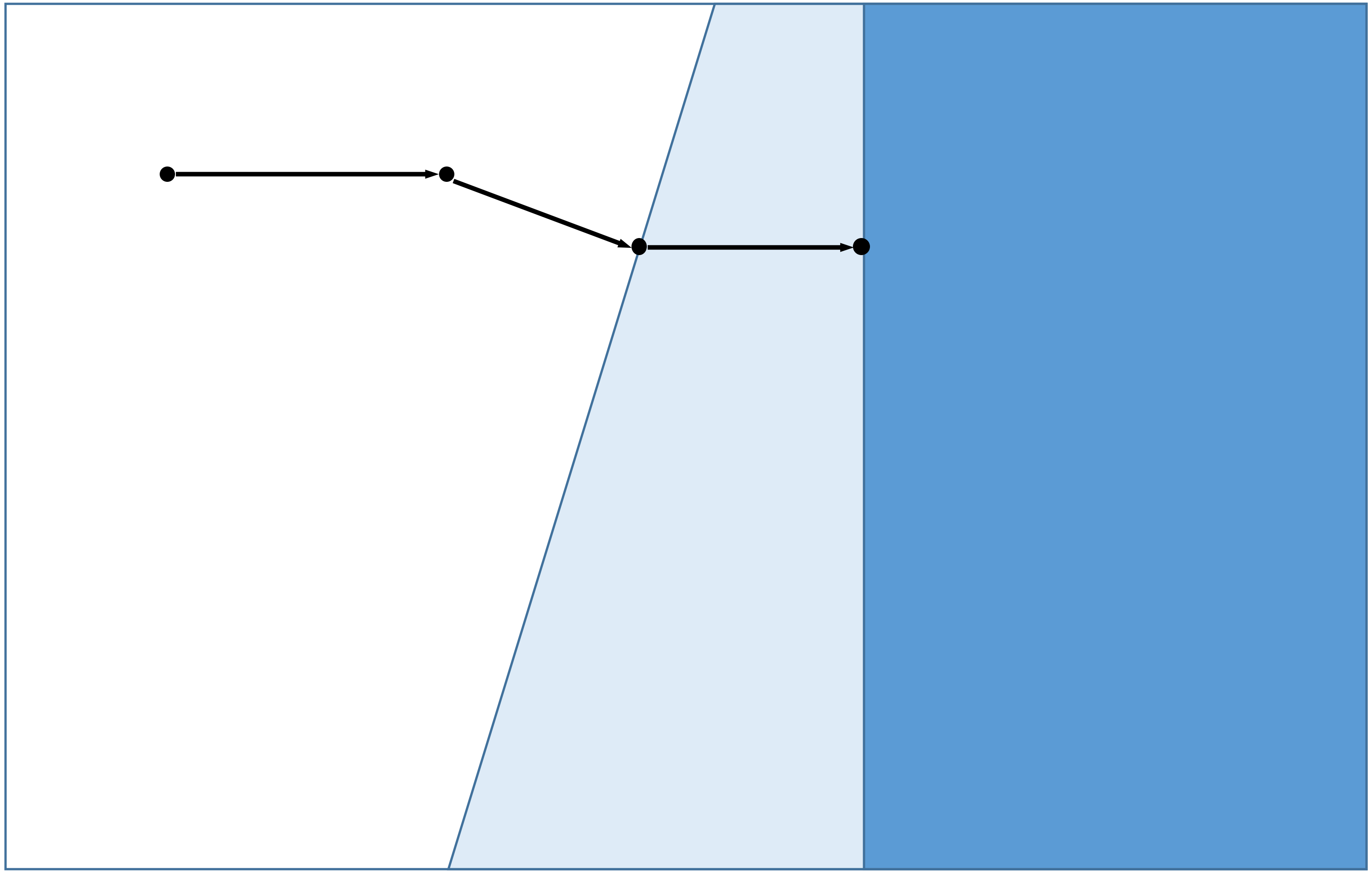
algorithm moves to x_t

pays $|| x_t - x_{t-1} ||$

nested: $K_t \subseteq K_{t-1}$

[Bansal Bohm Elias Koumoutsos Umboh SODA 2018]

Nested Version



algorithm controls point in $\mathbb{R}^d$

at time t:

convex body K_t arrives

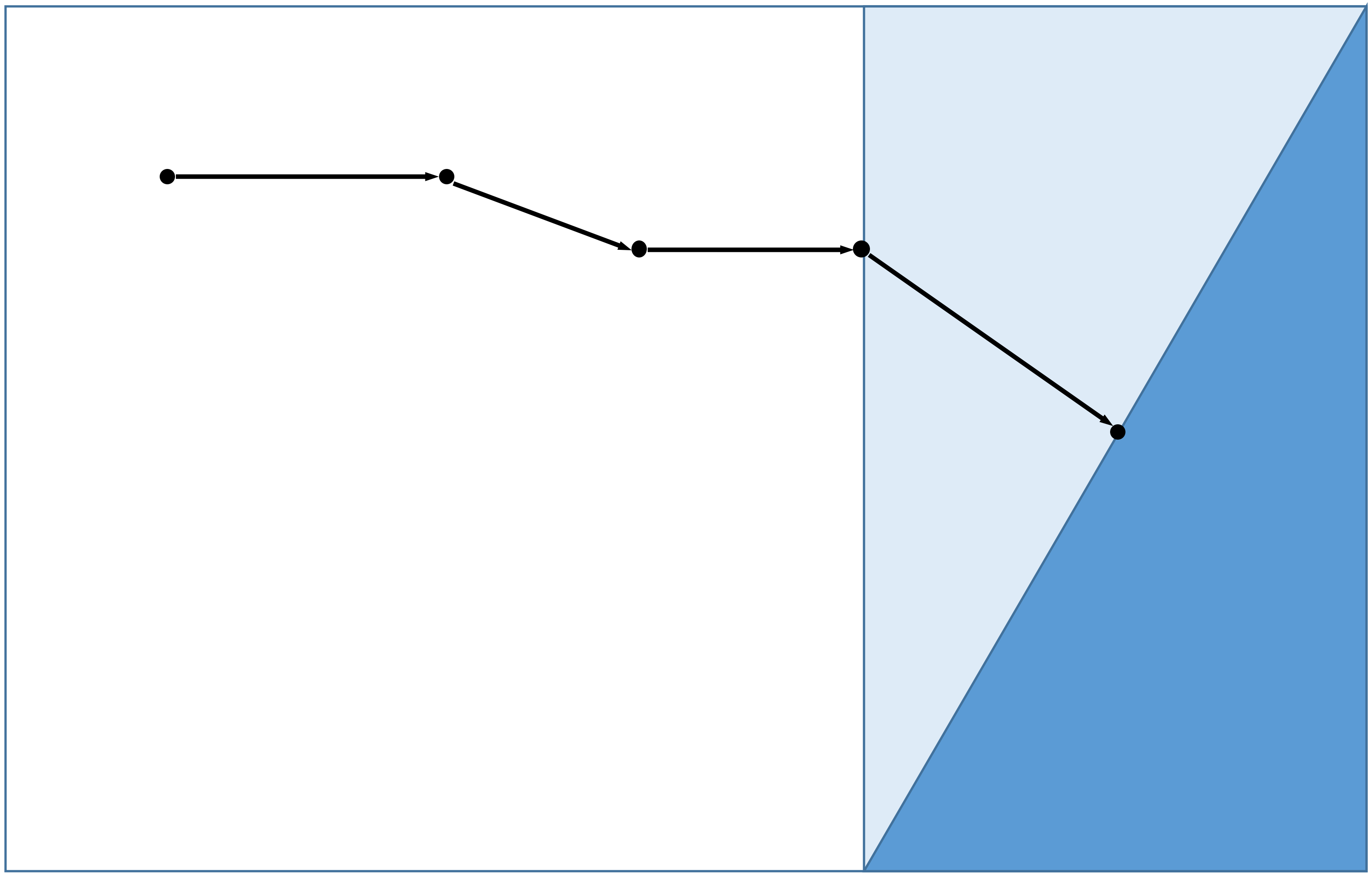
algorithm moves to x_t

pays $|| x_t - x_{t-1} ||$

nested: $K_t \subseteq K_{t-1}$

[Bansal Bohm Elias Koumoutsos Umboh SODA 2018]

Nested Version



algorithm controls point in $\mathbb{R}^d$

at time t:

convex body K_t arrives

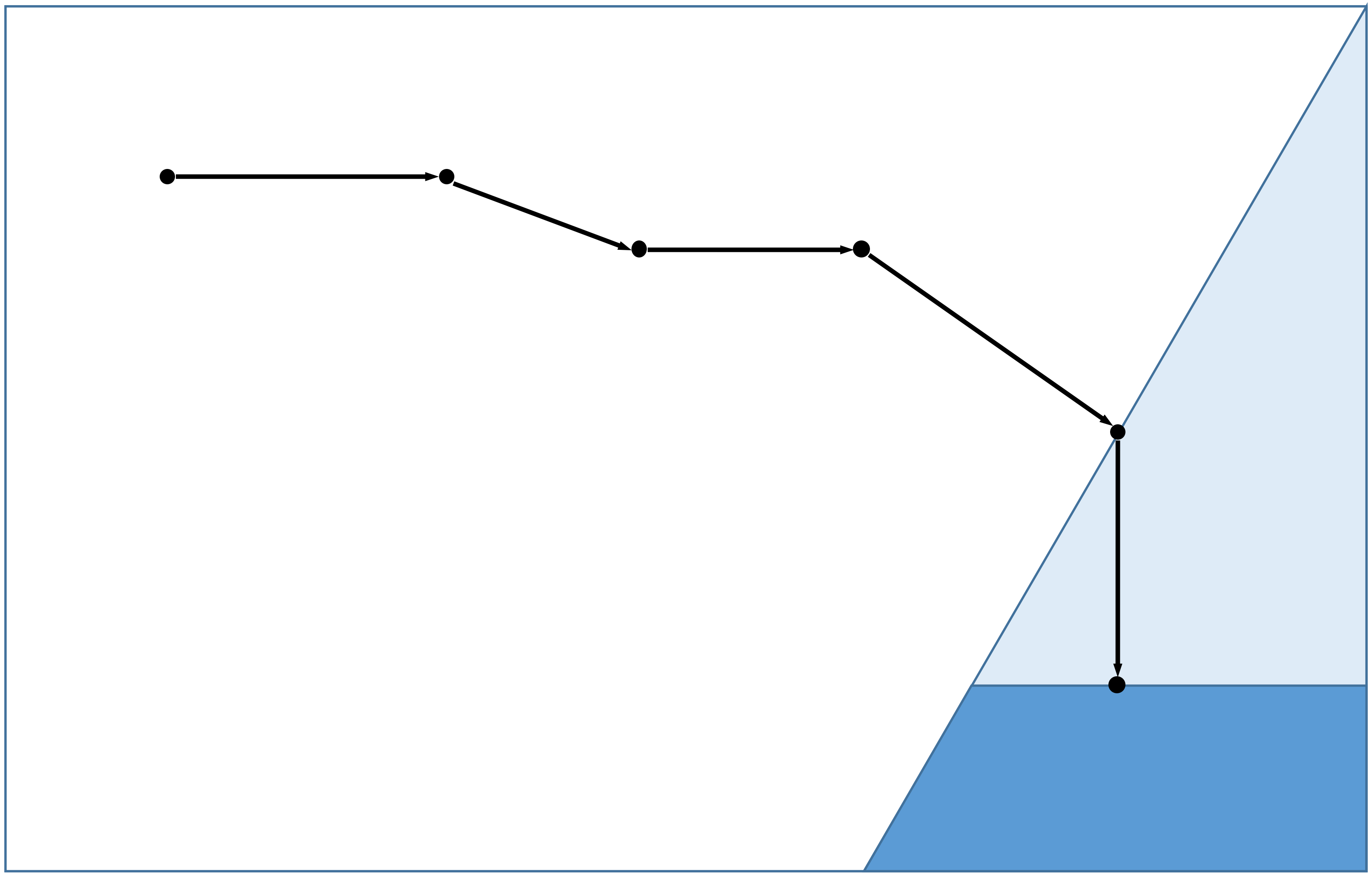
algorithm moves to x_t

pays $|| x_t - x_{t-1} ||$

nested: $K_t \subseteq K_{t-1}$

[Bansal Bohm Elias Koumoutsos Umboh SODA 2018]

Nested Version



algorithm controls point in $\mathbb{R}^d$

at time t:

convex body K_t arrives

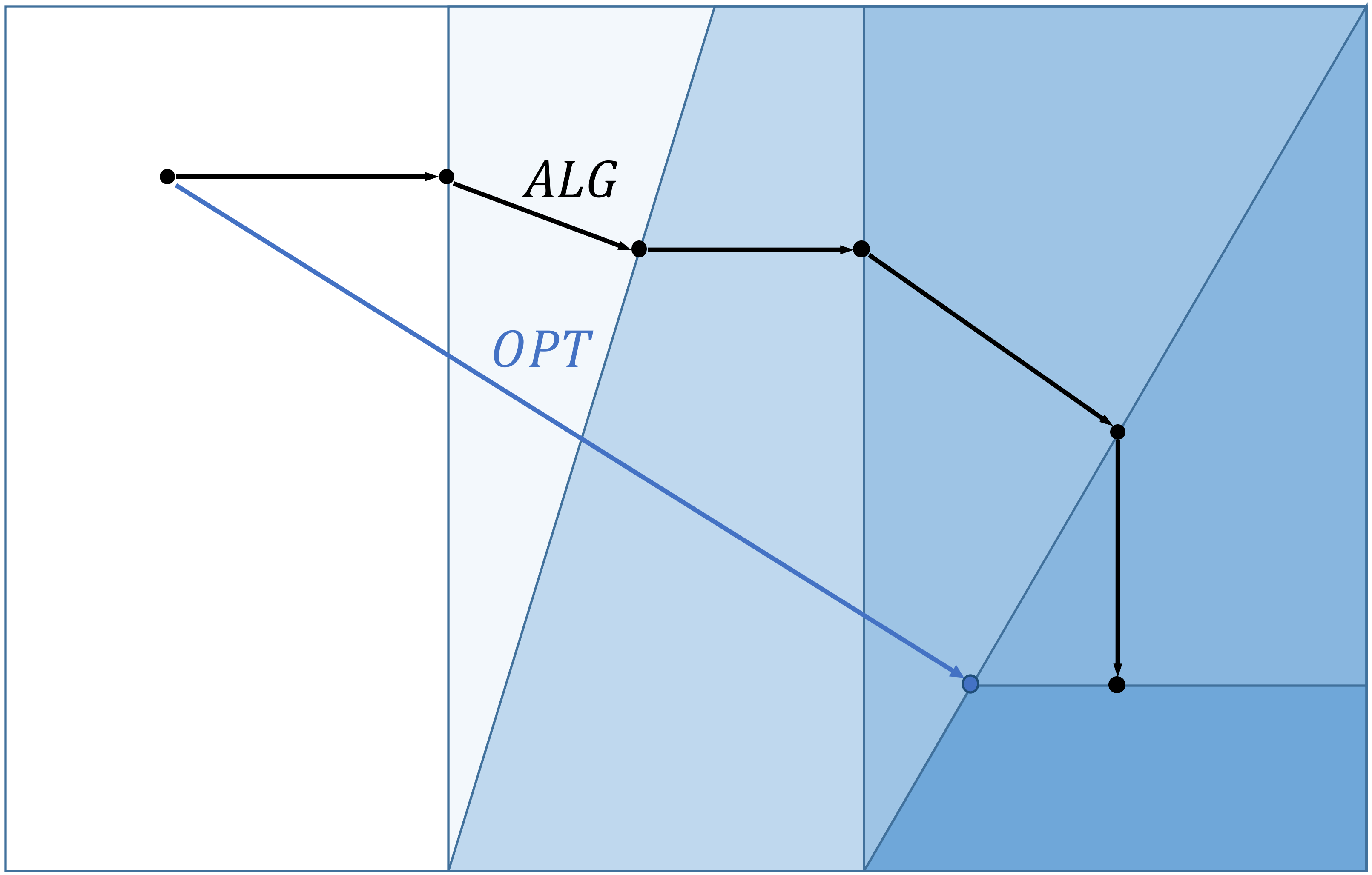
algorithm moves to x_t

pays $|| \; x_t - x_{t-1} \; ||$

nested: $K_t \subseteq K_{t-1}$

[Bansal Bohm Elias Koumoutsos Umboh SODA 2018]

Nested Version



algorithm controls point in $\mathbb{R}^d$

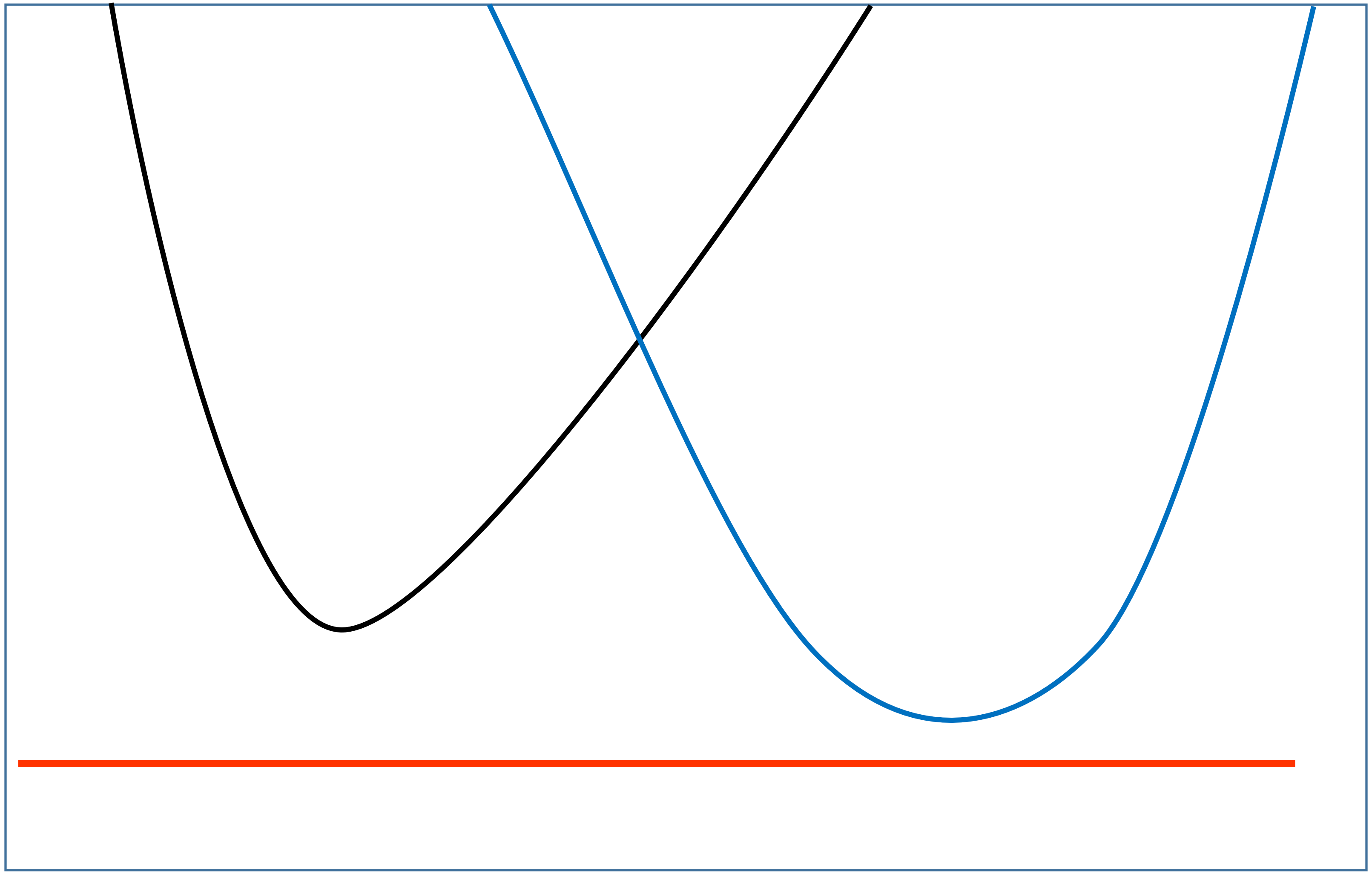
at time t:

- convex body K_t arrives
- algorithm moves to x_t
- pays $|| x_t - x_{t-1} ||$

nested: $K_t \subseteq K_{t-1}$

[Bansal Bohm Elias Koumoutsos Umboh SODA 2018]

a closely related problem: convex function chasing



algorithm controls point in $\mathbb{R}^d$

at time t:

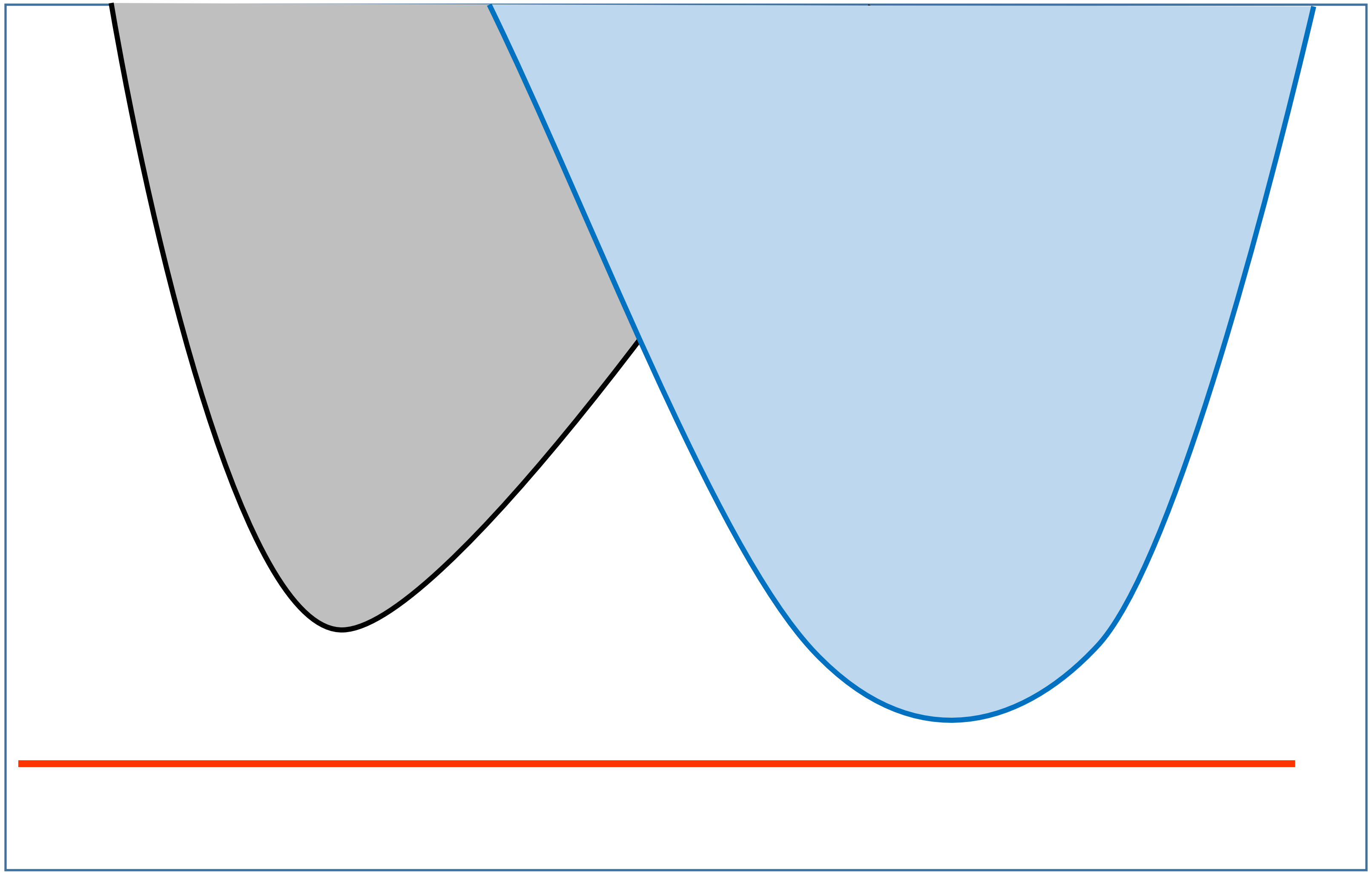
convex function f_t arrives

algorithm moves to x_t

pays $|| x_t - x_{t-1} || + f(x_t)$

[Friedman Linial 94]

Reductions: $\mathrm{CBC}_d \leq \mathrm{CFC}_d \leq \mathrm{CBC}_{d+1}$



algorithm controls point in $\mathbb{R}^d$

at time t:

- convex function f_t arrives
- algorithm moves to x_t
- pays $|| x_t - x_{t-1} || + f(x_t)$

within $O(1)$ of CBC in $\mathrm{d}+1$ dim

[Bubeck, Lee, Li, Sellke STOC 2019]

why do we care about convex body chasing?

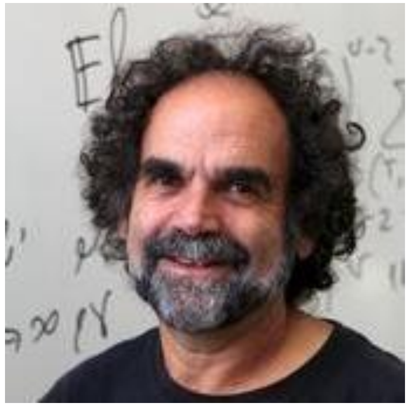
Generalization of (fractional version) of many online problems

- paging and k-server
- set cover and other packing/covering problems

get generic online convex program solvers?

get a unified algorithm for these problems?

a brief history



Discrete Comput Geom 9:293–321 (1993)

Discrete & Computational Geometry

On Convex Body Chasing*

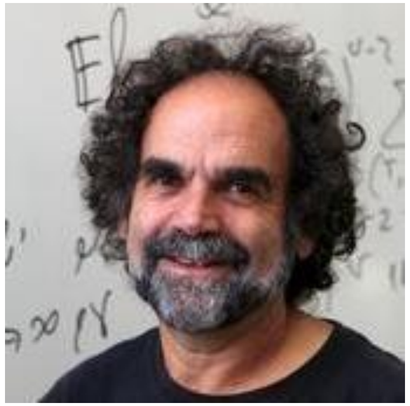
Joel Friedman[1] and Nathan Linial[2]

[1] Department of Computer Science, Princeton University, Princeton, NJ 08544, USA

[2] Department of Computer Science, Hebrew University, Givat Ram, Jerusalem 91904, Israel

Abstract. A player moving in the plane is given a sequence of instructions of the following type: at step i a planar convex set F_i is specified, and the player has to move to a point in F_i. The player is charged for the distance traveled. We provide a strategy for the player which is competitive, i.e., for any sequence F_i the cost to the player is within a constant (multiplicative) factor of the "off-line" cost (i.e., the least possible cost when all F_i are known in advance). We conjecture that similar strategies can be developed for this game in any Euclidean space and perhaps even in all metric spaces. The analogous statement where convex sets are replaced by more general families of sets in a metric space includes many on-line/off-line problems such as the k-server problem; we make some remarks on these more general problems.

contains, among many things: two algorithms



continuous or piecewise-continuous versions of the problem (with a very restricted set of discontinuities, so as not to include the discrete problem trivially!). For another example where the continuous version is simpler (for somewhat different reasons) see [3].

In Section 2 we given a **simple algorithm and analysis for line chasing in the plane,** and give some variants of the algorithm which are also competitive. In Section 3 we solve the half-plane-chasing problem in the plane. In Section 4 we make some general remarks about set-chasing problems, and in particular explain **that convex body chasing in the plane follows** from Section 3.

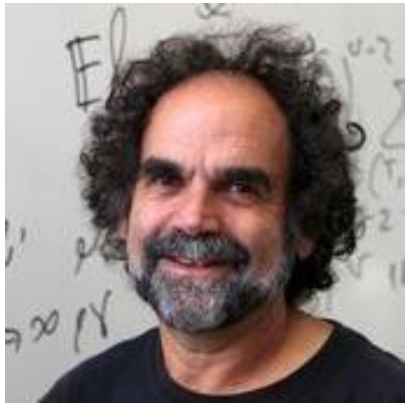
2. Line Chasing

In this section we discuss the problem of line chasing. For this problem we give a simple algorithm and analysis, and the technqies used here are built upon for the half-plane-chasing algorithm.

2.1. *Continuous Version*

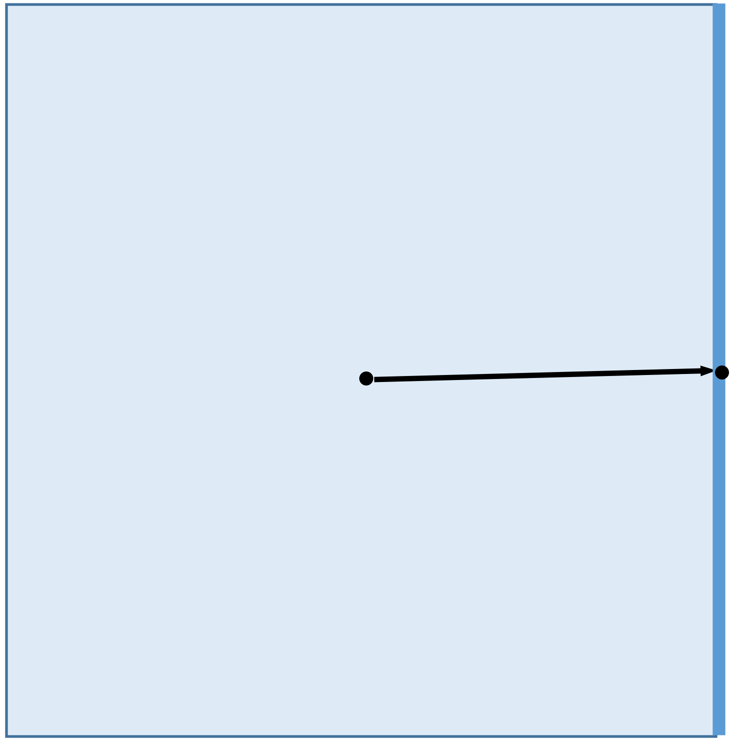
Consider the following continuous version of line chasing: we are given an initial point $p_0 \in \mathbf{R}^2$, and a family of lines in $\mathbf{R}^2$, l_t, where $t \in [0, T]$ for some T. In addition, $p_0 \in l_0$, and the lines vary continuously and piecewise differentiably in t; by the latter we mean that we can write the lines as

a lower bound



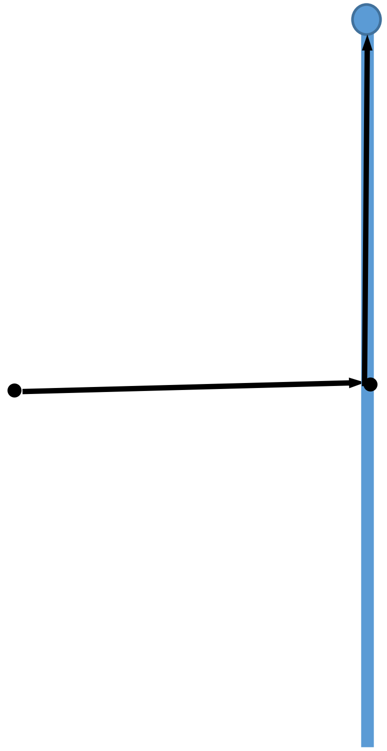
In the above, by a convex set in a metric space we mean a subset T which for any $x, y \in T$ contains all points z with $\rho(x, y) = \rho(x, z) + \rho(z, y)$. We remark that the condition of the second part of Question 1.2 also generalizes the k-server condition. We cannot really hope that the family of closed convex sets in every metric space is chaseable, for this would imply that there is a universal competitiveness ratio for all metric spaces (by gluing collections of metric spaces with bad ratios together); for example, **the competitiveness ratio of convex set chasing in $\mathbf{R}^n$ cannot be better than $\sqrt{n}$** on the problem instance $p_0 = 0$, and $F_i = \{x_i = \pm 1\}$ (with $\pm$ chosen according to which is further from the on-line player). Hence convex sets in $\mathbf{R}^\infty$ are not chaseable. One can ask for geometric properties on a metric space which imply chaseability of convex sets, such as a Helly-type property, etc. For example, the analysis of the "move-to-front" rule for maintaining a linear list in [11] shows that the family of convex sets in the symmetric group, S_n (with metric given by the number of transpositions), is chaseable, in fact with competitiveness ratio 1 using the greedy algorithm. More generally geometric constraints on the family of subsets may be written down so that the greedy

a lower bound



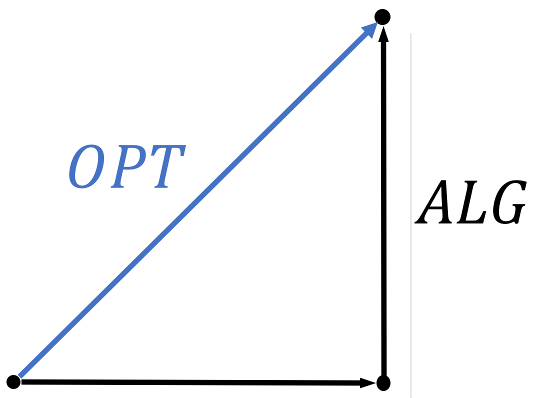
In the above, by a convex set in a metric space we mean a subset T which for any $x, y \in T$ contains all points z with $\rho(x, y) = \rho(x, z) + \rho(z, y)$. We remark that the condition of the second part of Question 1.2 also generalizes the k-server condition. We cannot really hope that the family of closed convex sets in every metric space is chaseable, for this would imply that there is a universal competitiveness ratio for all metric spaces (by gluing collections of metric spaces with bad ratios together); for example, **the competitiveness ratio of convex set chasing in $\mathbf{R}^n$ cannot be better than $\sqrt{n}$** on the problem instance $p_0 = 0$, and $F_i = \{x_i = \pm 1\}$ (with $\pm$ chosen according to which is further from the on-line player). Hence convex sets in $\mathbf{R}^\infty$ are not chaseable. One can ask for geometric properties on a metric space which imply chaseability of convex sets, such as a Helly-type property, etc. For example, the analysis of the "move-to-front" rule for maintaining a linear list in [11] shows that the family of convex sets in the symmetric group, S_n (with metric given by the number of transpositions), is chaseable, in fact with competitiveness ratio 1 using the greedy algorithm. More generally geometric constraints on the family of subsets may be written down so that the greedy

a lower bound



In the above, by a convex set in a metric space we mean a subset T which for any $x, y \in T$ contains all points z with $\rho(x, y) = \rho(x, z) + \rho(z, y)$. We remark that the condition of the second part of Question 1.2 also generalizes the k-server condition. We cannot really hope that the family of closed convex sets in every metric space is chaseable, for this would imply that there is a universal competitiveness ratio for all metric spaces (by gluing collections of metric spaces with bad ratios together); for example, **the competitiveness ratio of convex set chasing in $\mathbf{R}^n$ cannot be better than $\sqrt{n}$** on the problem instance $p_0 = 0$, and $F_i = \{x_i = \pm 1\}$ (with $\pm$ chosen according to which is further from the on-line player). Hence convex sets in $\mathbf{R}^\infty$ are not chaseable. One can ask for geometric properties on a metric space which imply chaseability of convex sets, such as a Helly-type property, etc. For example, the analysis of the "move-to-front" rule for maintaining a linear list in [11] shows that the family of convex sets in the symmetric group, S_n (with metric given by the number of transpositions), is chaseable, in fact with competitiveness ratio 1 using the greedy algorithm. More generally geometric constraints on the family of subsets may be written down so that the greedy

a lower bound

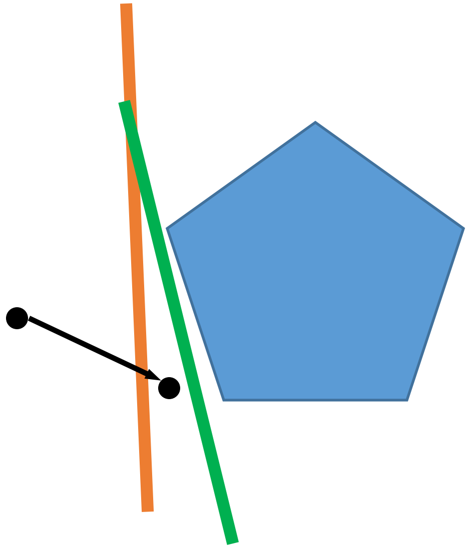


$$ALG \geq \sqrt{2} \cdot OPT$$

$$\boxed{ALG \geq \sqrt{d} \cdot OPT}$$

In the above, by a convex set in a metric space we mean a subset T which for any $x, y \in T$ contains all points z with $\rho(x, y) = \rho(x, z) + \rho(z, y)$. We remark that the condition of the second part of Question 1.2 also generalizes the k-server condition. We cannot really hope that the family of closed convex sets in every metric space is chaseable, for this would imply that there is a universal competitiveness ratio for all metric spaces (by gluing collections of metric spaces with bad ratios together); for example, **the competitiveness ratio of convex set chasing in $\mathbf{R}^n$ cannot be better than $\sqrt{n}$** on the problem instance $p_0 = 0$, and $F_i = \{x_i = \pm 1\}$ (with $\pm$ chosen according to which is further from the on-line player). Hence convex sets in $\mathbf{R}^\infty$ are not chaseable. One can ask for geometric properties on a metric space which imply chaseability of convex sets, such as a Helly-type property, etc. For example, the analysis of the "move-to-front" rule for maintaining a linear list in [11] shows that the family of convex sets in the symmetric group, S_n (with metric given by the number of transpositions), is chaseable, in fact with competitiveness ratio 1 using the greedy algorithm. More generally geometric constraints on the family of subsets may be written down so that the greedy

a reduction to half-spaces



Corollary 4.3. *If the family of affine half-spaces in* $\mathbf{R}^n$ *is chaseable, then so is the family of convex bodies in* $\mathbf{R}^n$.

Corollary 4.4. *The family of convex bodies in the plane is chaseable.*

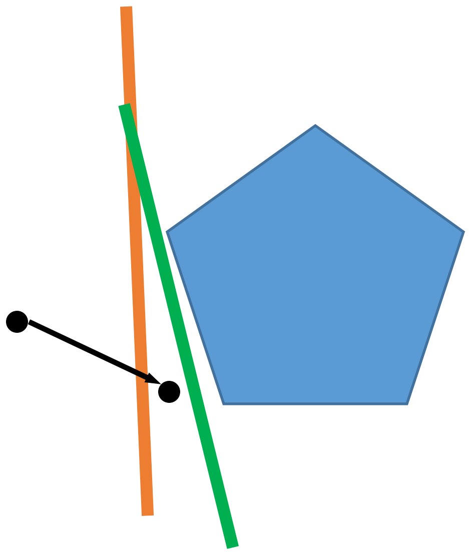
4.2. Plane Chasing in $\mathbf{R}^3$*, Lazy Line Chasing in* $\mathbf{R}^2$*, and Function Chasing*

At the time of writing we do not know whether or not convex bodies in $\mathbf{R}^3$ are chaseable. However, we define a "lazy set-chasing" problem and show that chasing planes in $\mathbf{R}^3$ is equivalent to the problem of lazy line chasing in $\mathbf{R}^2$.

The problem of lazy set chasing differs from the set-chasing problem in that a positive $\varepsilon \le 1$ is given as part of the input, and it is not required at time i to move to F_i. Instead, the cost of a solution $p_1, \ldots, p_n$ (here a solution is any collection of points in S) is

$$\sum_{i=1}^{n} \rho(p_{i-1}, p_i) + \varepsilon\rho(p_i, F_i),$$

a reduction to half-spaces



On Convex Body Chasing 317

Corollary 4.3. *If the family of affine half-spaces in* $\mathbf{R}^n$ *is chaseable, then so is the family of convex bodies in* $\mathbf{R}^n$.

Corollary 4.4. *The family of convex bodies in the plane is chaseable.*

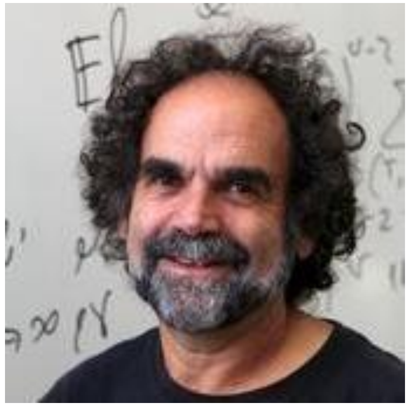
4.2. Plane Chasing in $\mathbf{R}^3$, *Lazy Line Chasing in* $\mathbf{R}^2$, *and Function Chasing*

At the time of writing we do not know whether or not convex bodies in $\mathbf{R}^3$ are chaseable. However, we define a "lazy set-chasing" problem and show that chasing planes in $\mathbf{R}^3$ is equivalent to the problem of lazy line chasing in $\mathbf{R}^2$.

The problem of lazy set chasing differs from the set-chasing problem in that a positive $\varepsilon \leq 1$ is given as part of the input, and it is not required at time i to move to F_i. Instead, the cost of a solution $p_1, \ldots, p_n$ (here a solution is any collection of points in S) is

$$\sum_{i=1}^{n} \rho(p_{i-1}, p_i) + \varepsilon\rho(p_i, F_i),$$

and a conjecture



The problem at hand is to find a solution whose cost is as small as possible. As usual, this problem has an *off-line* version, where we know the F_i in advance, and an *on-line version*, where the F_i are given one at a time and p_i must be chosen before knowing F_{i+1}; we seek to find a *competitive* on-line algorithm, i.e., one for which the cost is never more than a fixed constant times the cost of any (off-line) solution. A family $\mathscr{F}$ is said to be *chaseable* if there exists an on-line algorithm competitive with the off-line algorithm.

We wish to study what families are chaseable, and what geometric properties guarantee that a family is chaseable or not. At this level of generality these questions are probably difficult, and contain many on-line/off-line questions (as in [1]–[11]).

For example, this problem contains the k-server problem of [9]. More generally, we can form a k-server version of the set-chasing problem for $k > 1$, but clearly this is again a set-chasing problem for a family of subsets in the kth cartesian product of the original metric space. In fact, one motivation for the set-chasing problem is to put the chaseability of families such as those arising from k-server problems into a simple geometric framework.

From the geometric point of view, it seems natural to first consider set chasing in $\mathbf{R}^d$. The main goal of this paper is to prove that the collection of convex sets in $\mathbf{R}^2$ is chaseable. We more generally pose:

Conjecture 1.1. *For any d, the family of closed convex sets, in the metric space* $\mathbf{R}^d$, *is chaseable.*

Question 1.2. For which metric spaces is it true that the family of closed convex sets is chaseable? Same question for the family of unions of $\leq$ n closed convex sets, with n fixed.

rest of the talk

1. **Nested** Convex Set Chasing (part I)
   - Some failed algos
   - $O(d \log d)$ algo via recursive centroid

2. Nested Convex Set Chasing (part II)
   - Steiner point
   - move to Steiner point

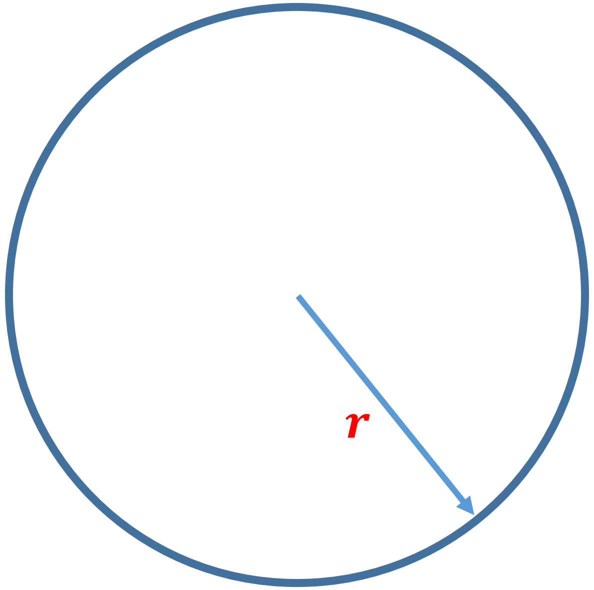
3. General Convex Set Chasing
   - reduction to nested Steiner point algo

simpler "bounded" version

Convex sets $B(0, r) = K_0$, followed by K_1, K_2 ... all subsets of $B(0, \boldsymbol{r})$

Promise: $OPT \approx \boldsymbol{r}$

Want: $ALG \leq \ f(d) \cdot \boldsymbol{r}$



Fact: implies $O\big(f(d)\big)$-competitive for CBC.

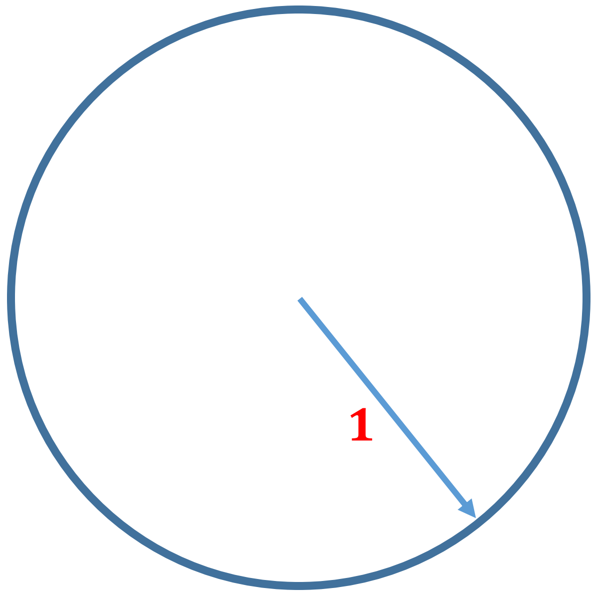
Proof: guess-and-double.

simpler "bounded" version

Convex sets $B(0, r) = K_0$, followed by K_1, K_2 ... all subsets of $B(0, \mathbf{1})$

Promise: $OPT \approx \mathbf{1}$

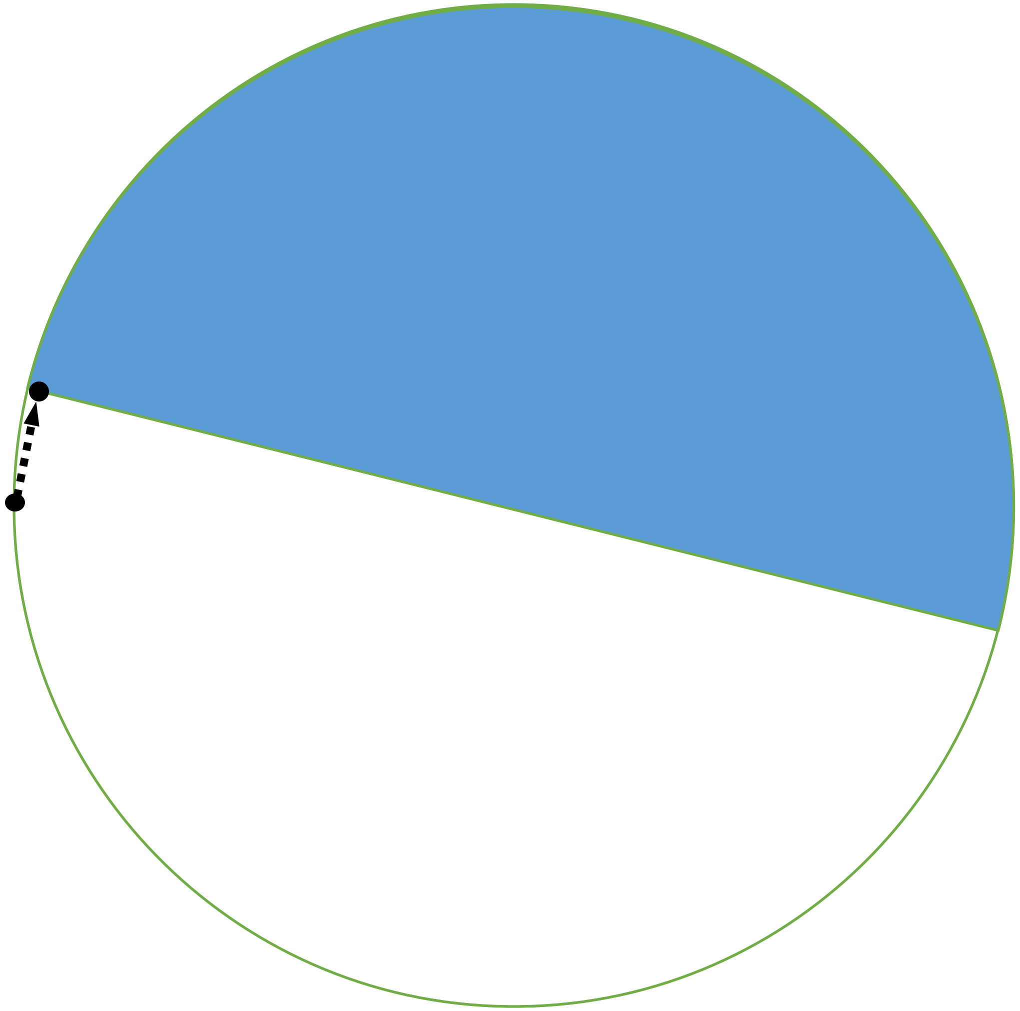
Want: $ALG \leq f(d) \cdot \mathbf{1}$



Fact: implies $O(f(d))$-competitive for CBC.

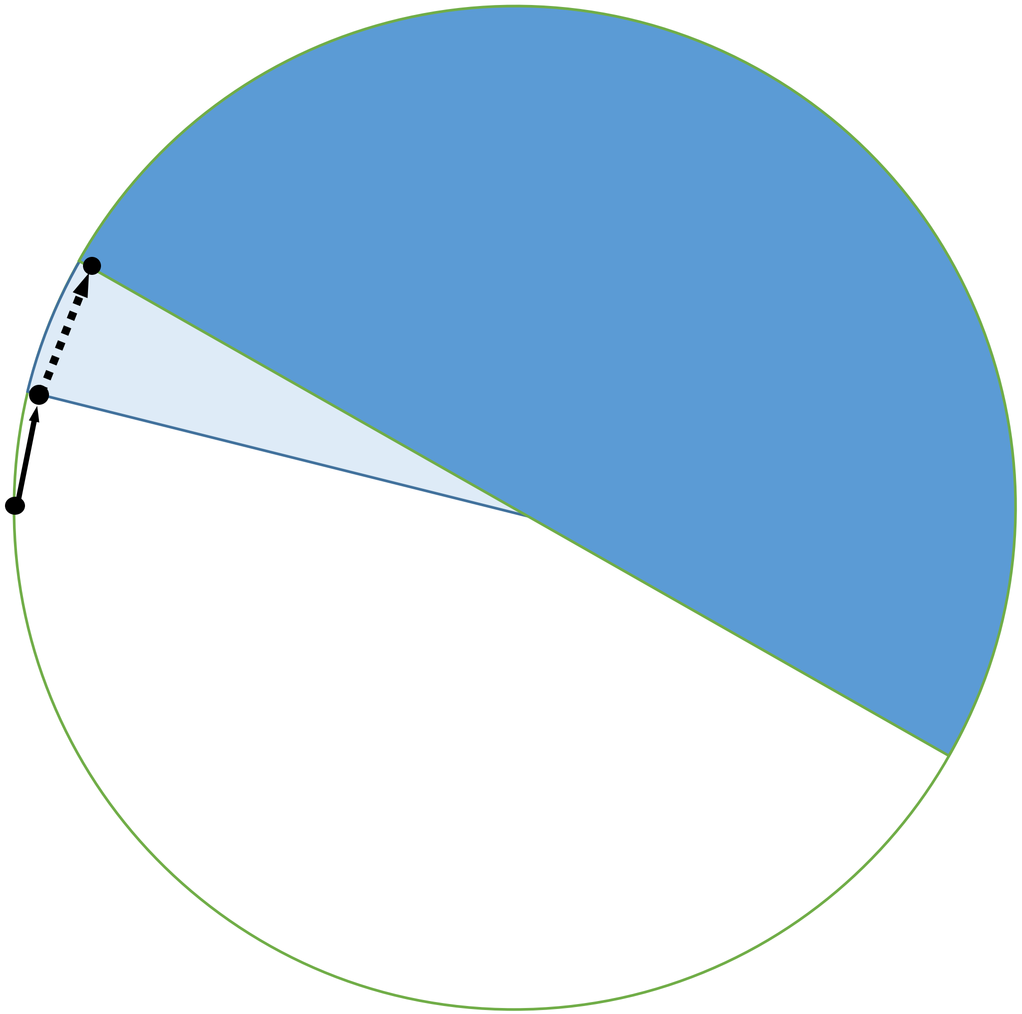
Proof: guess-and-double.

strawman #1: greedy



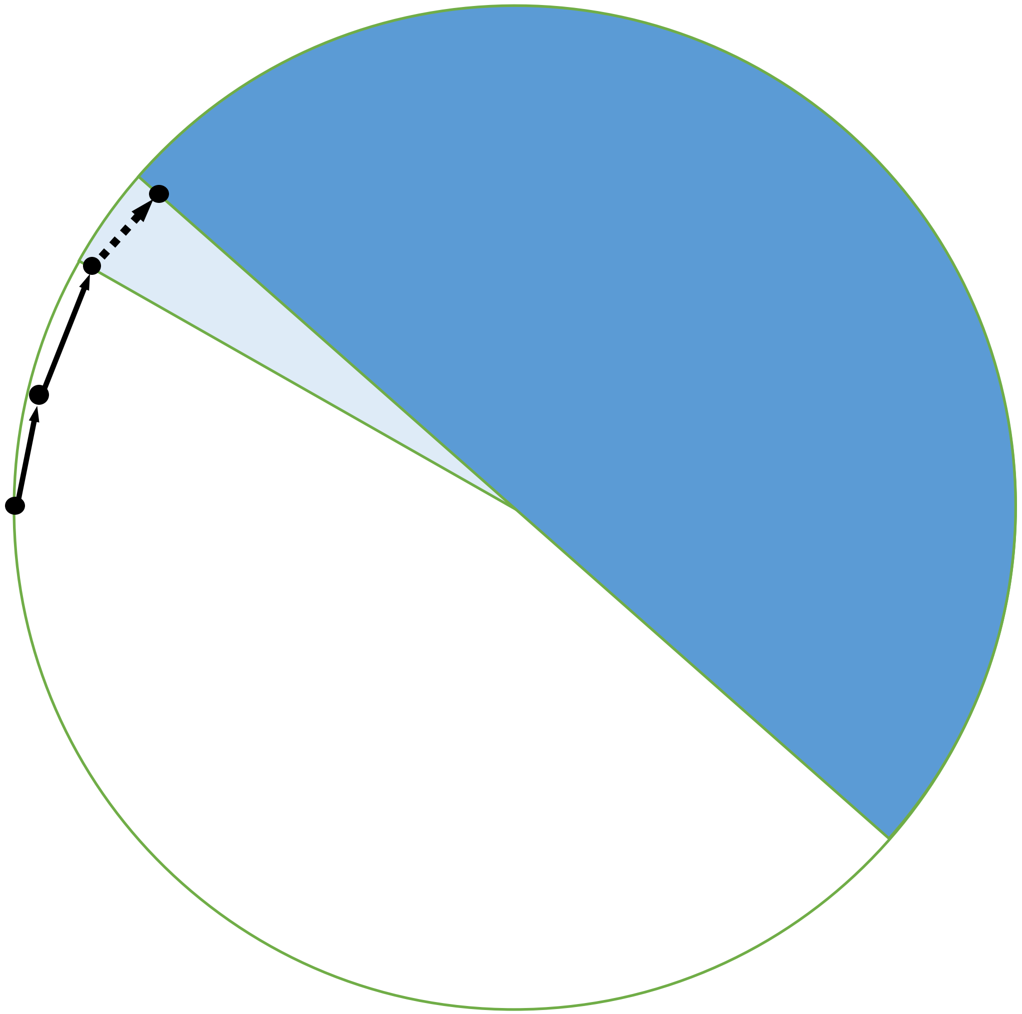
move to the closest point in K_t

strawman #1: greedy



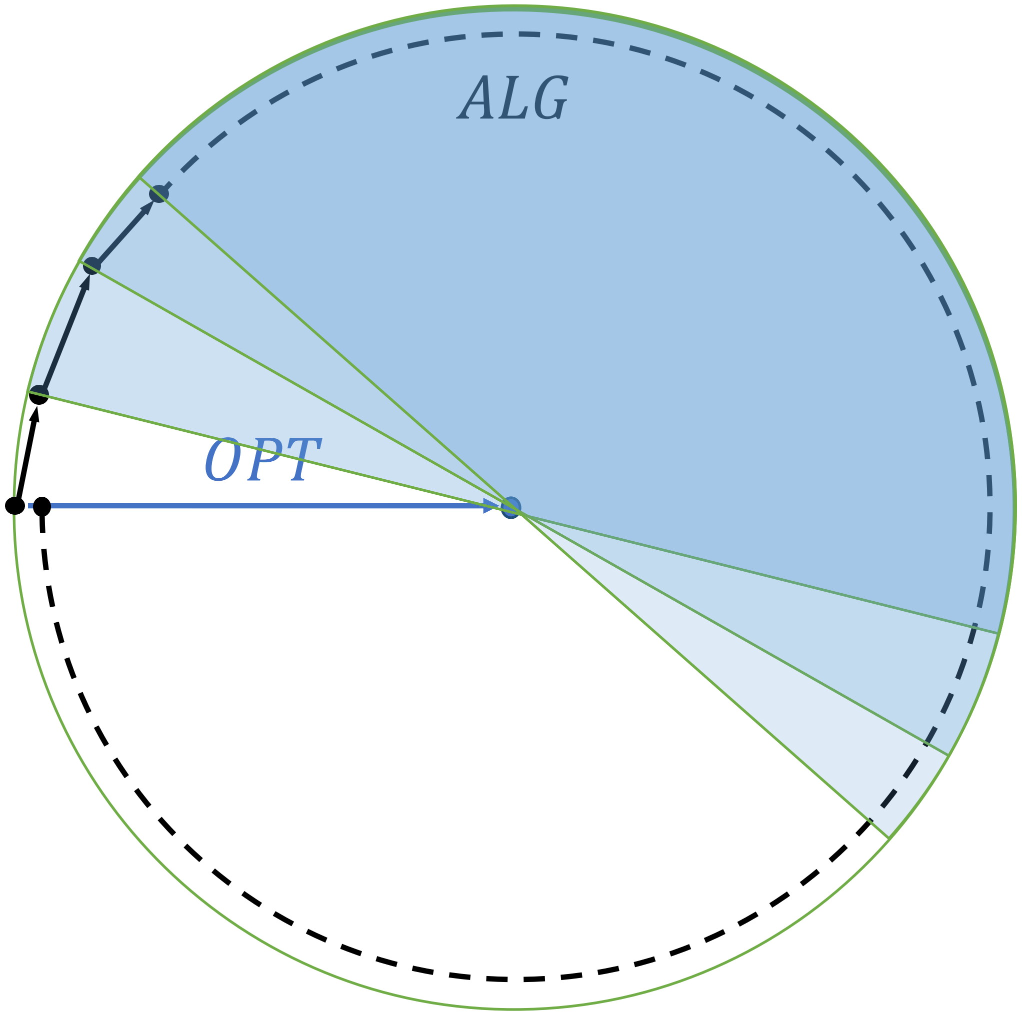
move to the closest point in K_t

strawman #1: greedy



move to the closest point in K_t

strawman #1: greedy



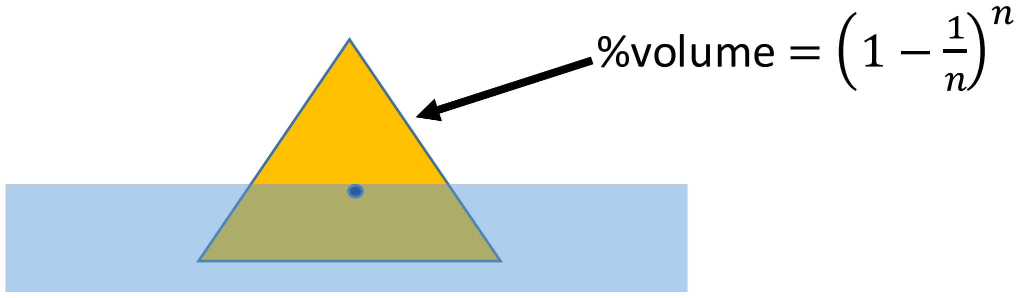
move to the closest point in K_t

unbounded competitive ratio!

Nested Case

Grunbaum's Inequality [1960]

For any convex body,
any half-space that cuts off the centroid
cuts volume by at least $(1 - 1/e)$



Branko Grunbaum, https://opc.mfo.de/detail?photo_id=7667

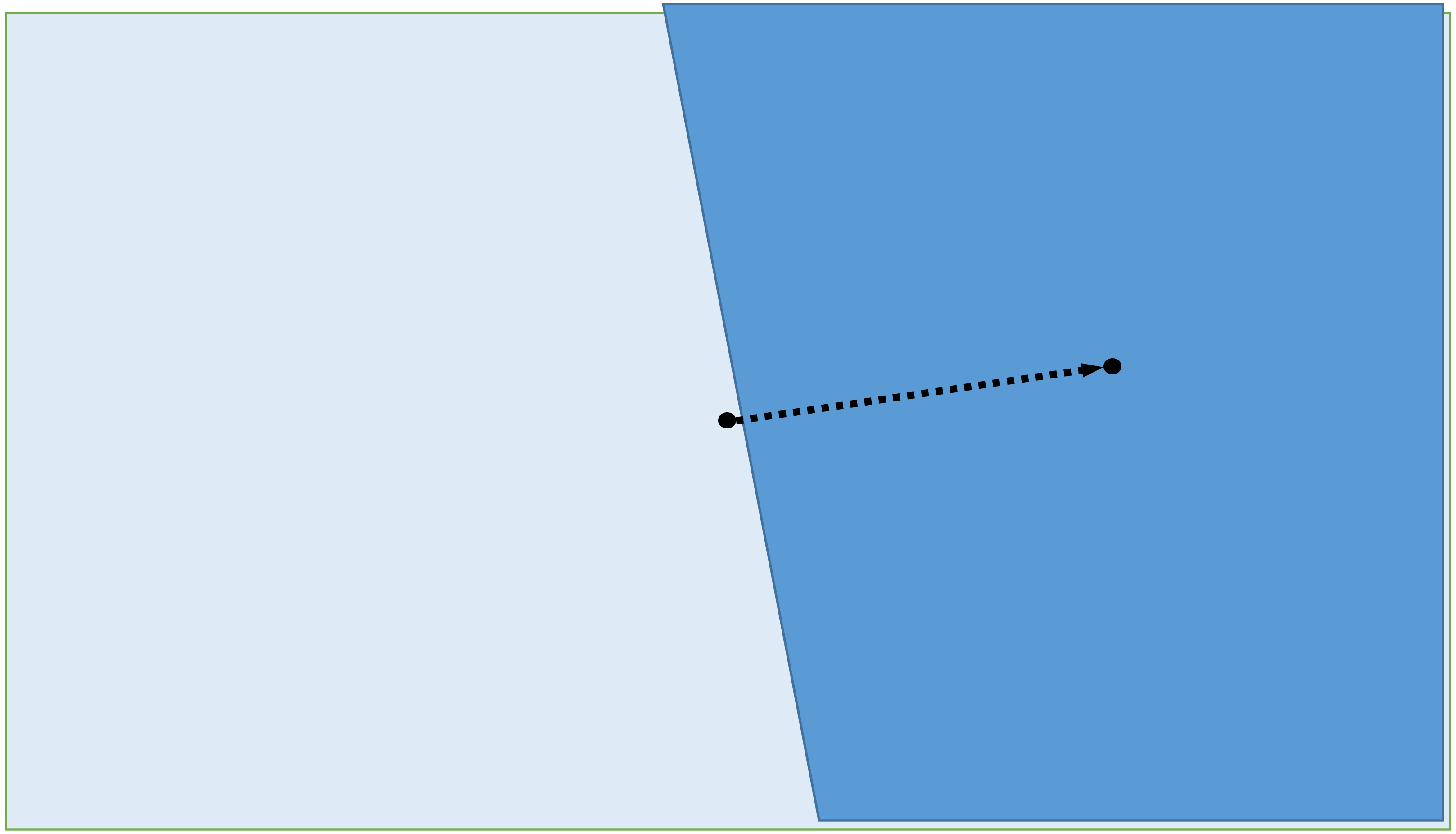
Algo for Nested Case:

Move to centroid of current body

Hope: each time volume decreases a lot

Maybe don’t need to move very often

strawman #2 (for nested case): centroid

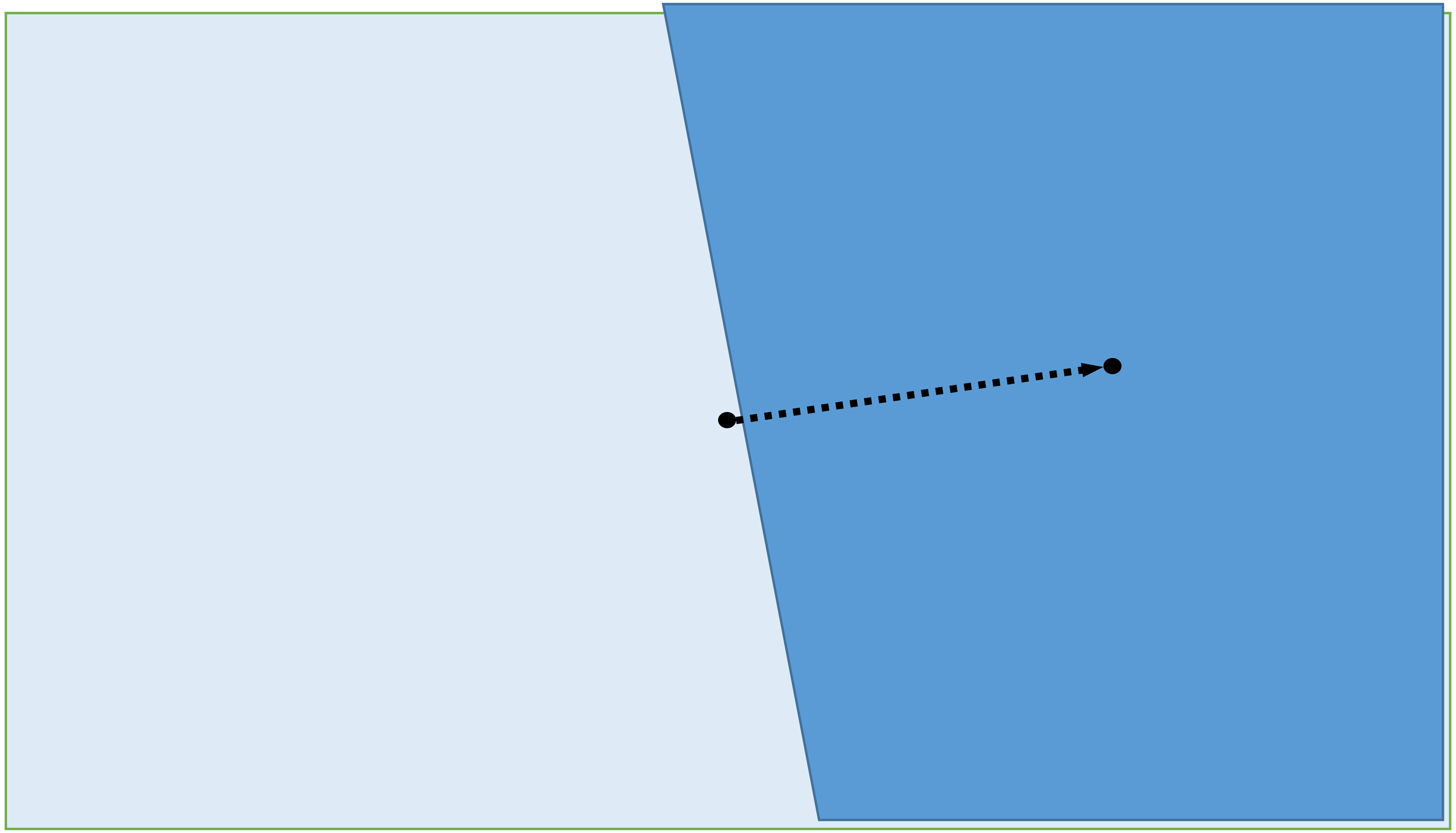


move to centroid of K_t

Grunbaum's Theorem
half-space cuts off centroid ⇒
volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

strawman #2 (for nested case): centroid



move to centroid of K_t

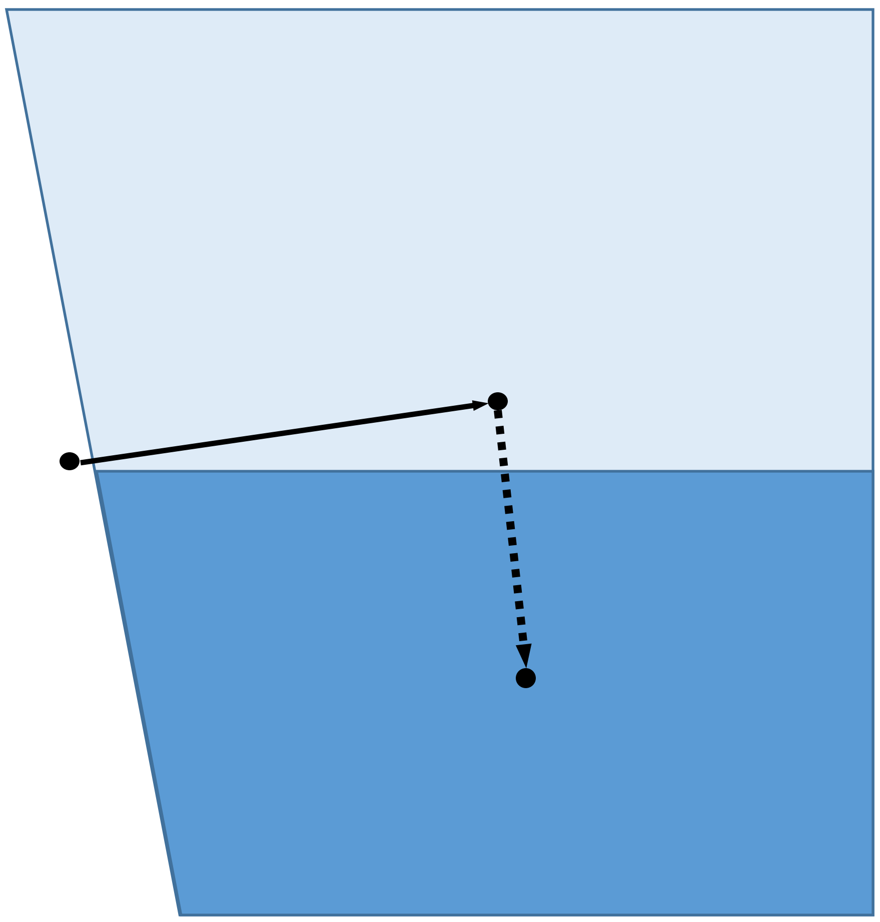
Grunbaum's Theorem

half-space cuts off centroid ⇒

volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

strawman #2 (for nested case): centroid

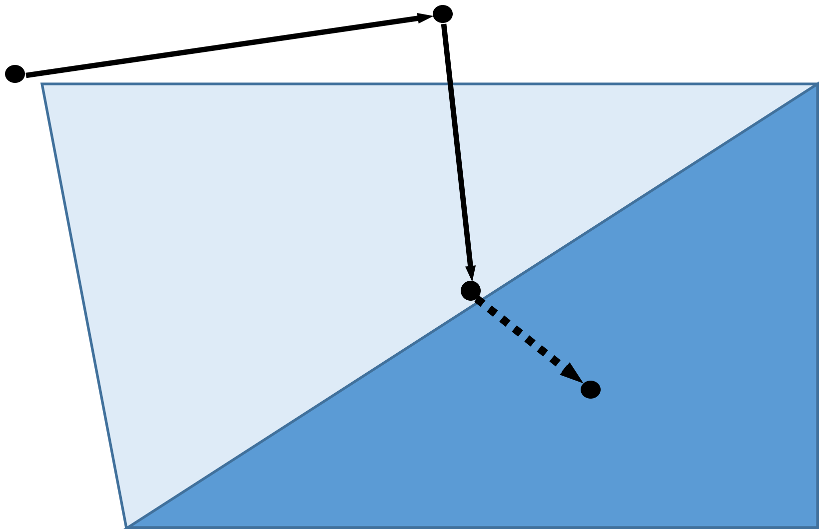


move to centroid of K_t

Grunbaum's Theorem
  half-space cuts off centroid ⇒
    volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

strawman #2 (for nested case): centroid

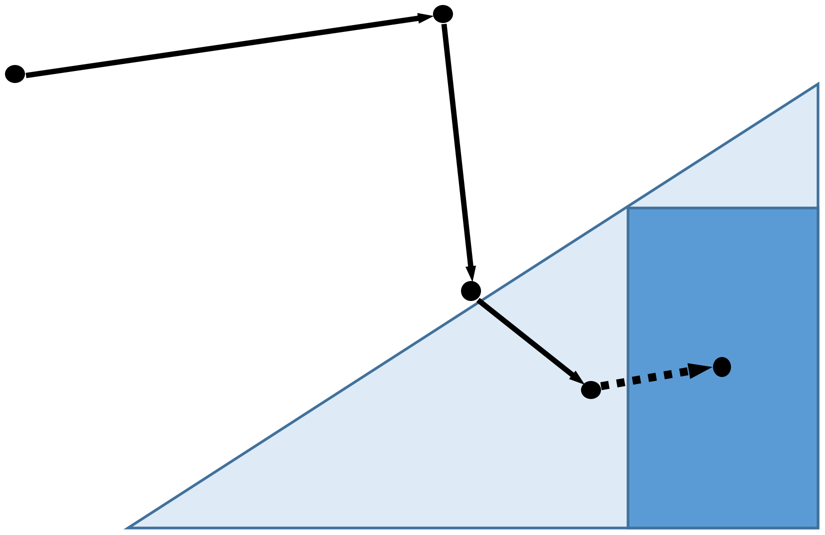


move to centroid of K_t

Grunbaum's Theorem
half-space cuts off centroid ⇒
volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

strawman #2 (for nested case): centroid

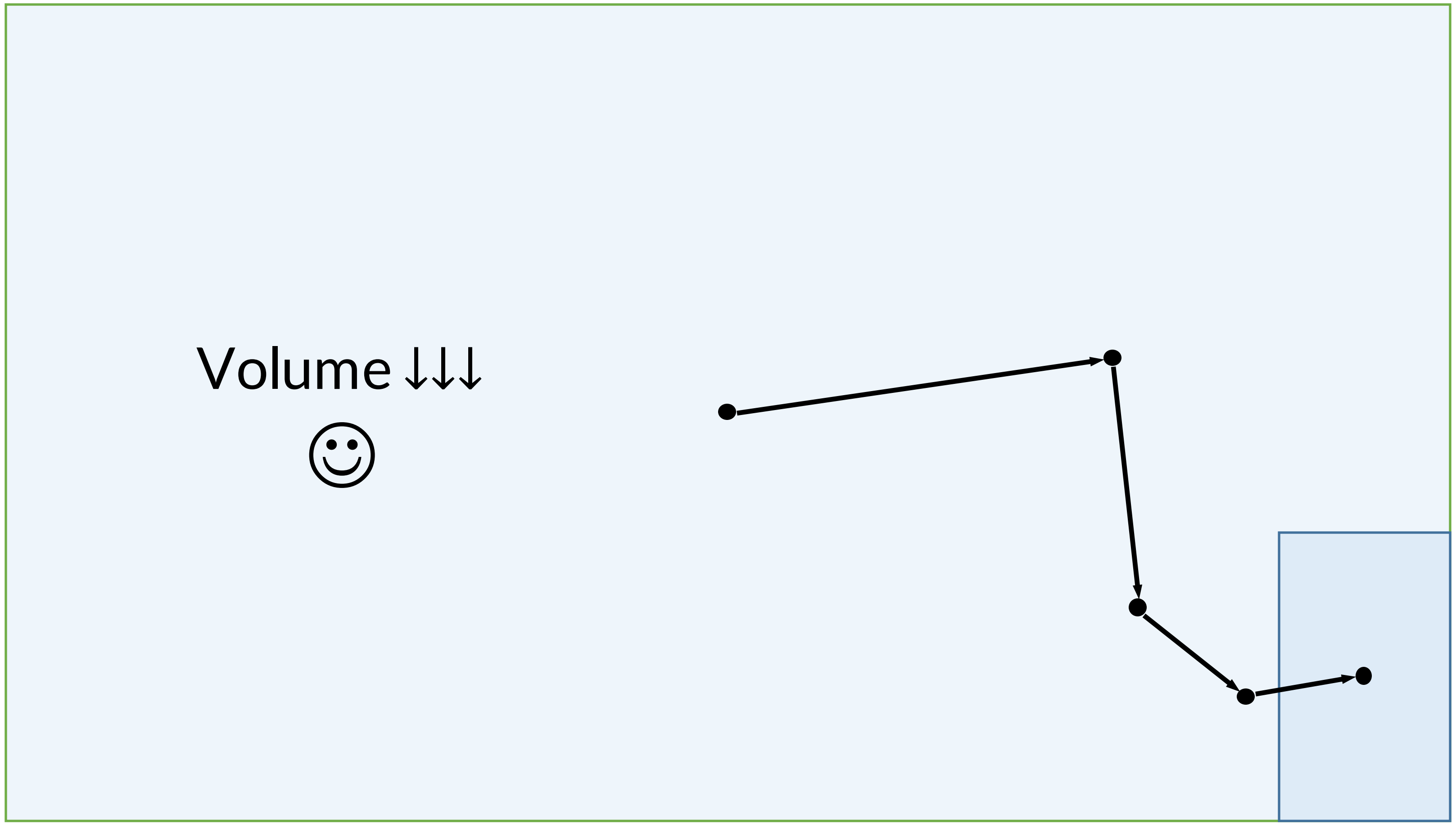


move to centroid of K_t

Grunbaum's Theorem
half-space cuts off centroid $\Rightarrow$
volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

strawman #2 (for nested case): centroid

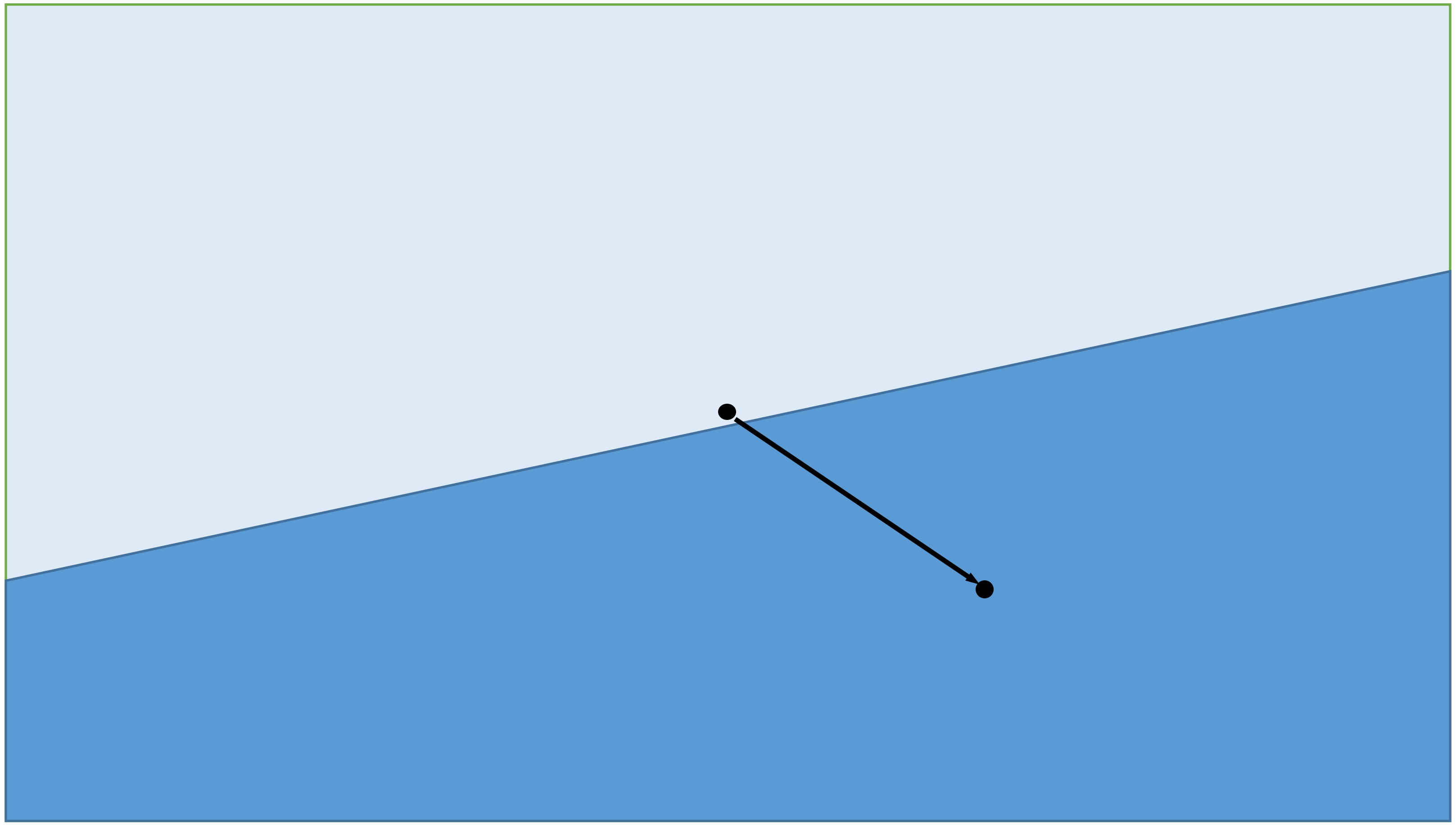


move to centroid of K_t

Grunbaum's Theorem
half-space cuts off centroid ⇒
volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

but alas…

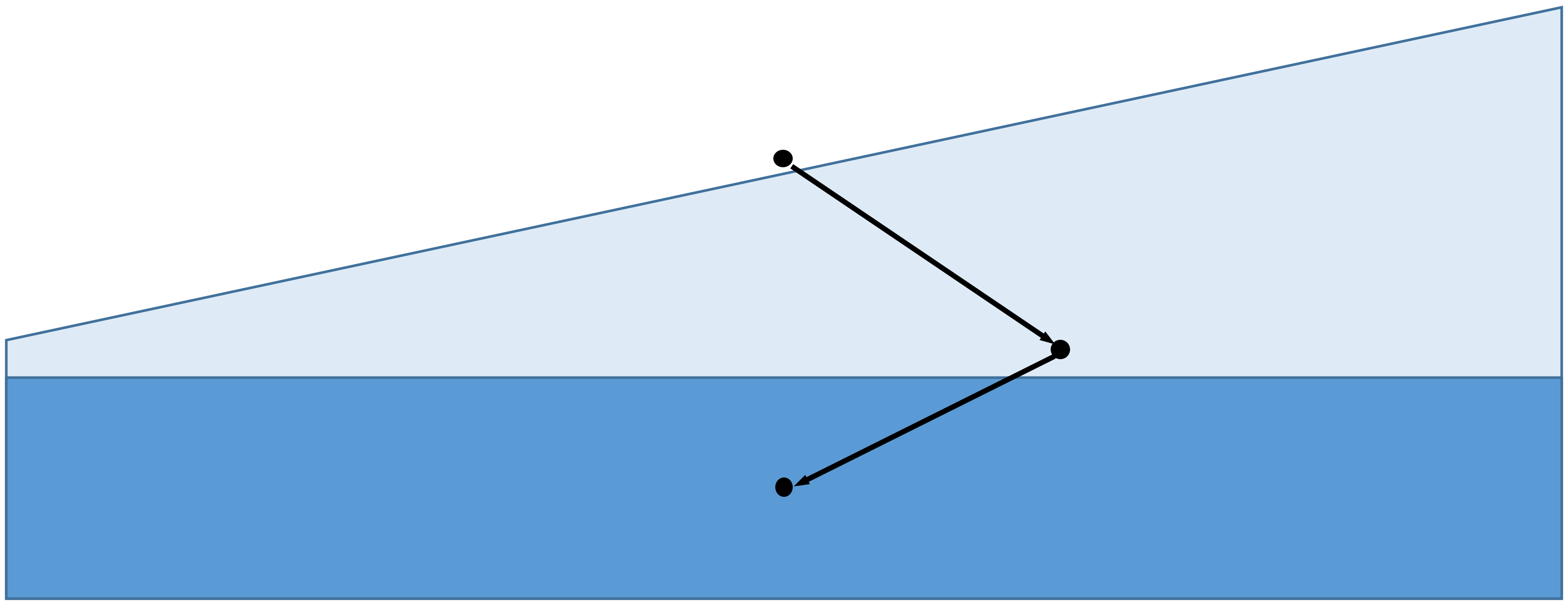


move to centroid of K_t

Grunbaum's Theorem
half-space cuts off centroid $\Rightarrow$
volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

but alas...

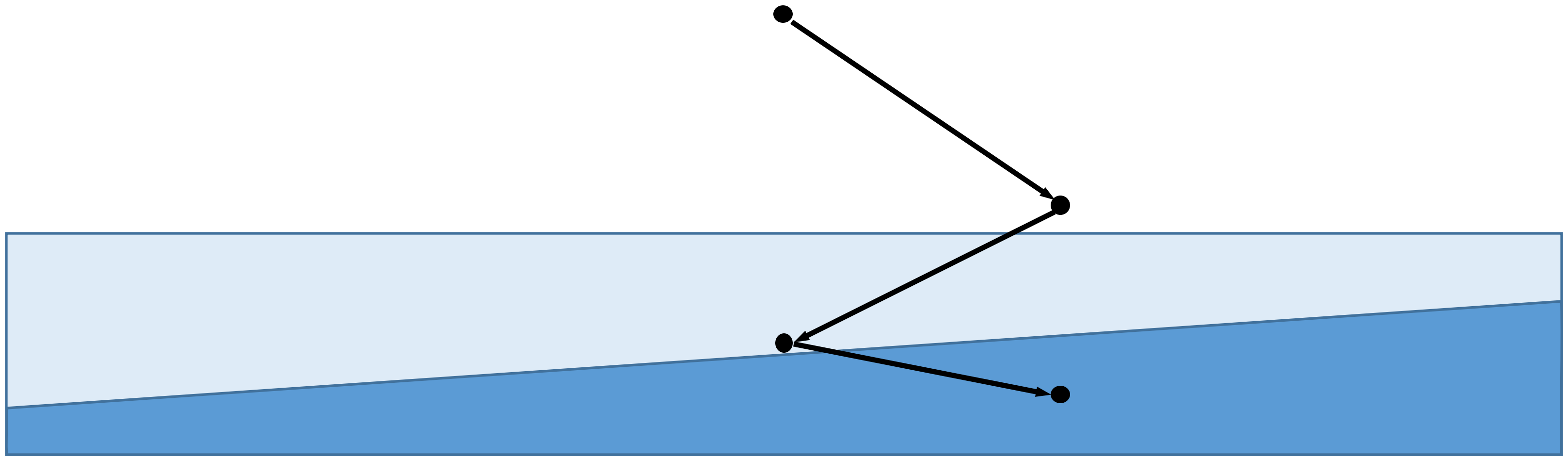


move to centroid of K_t

Grunbaum's Theorem
  half-space cuts off centroid $\Rightarrow$
    volume decreases $(1 - 1/e)$

[Argue Bubeck Cohen Gupta Lee 19

but alas...



move to centroid of K_t

Grunbaum's Theorem
half-space cuts off centroid ⇒
volume decreases $(1 - 1/e)$

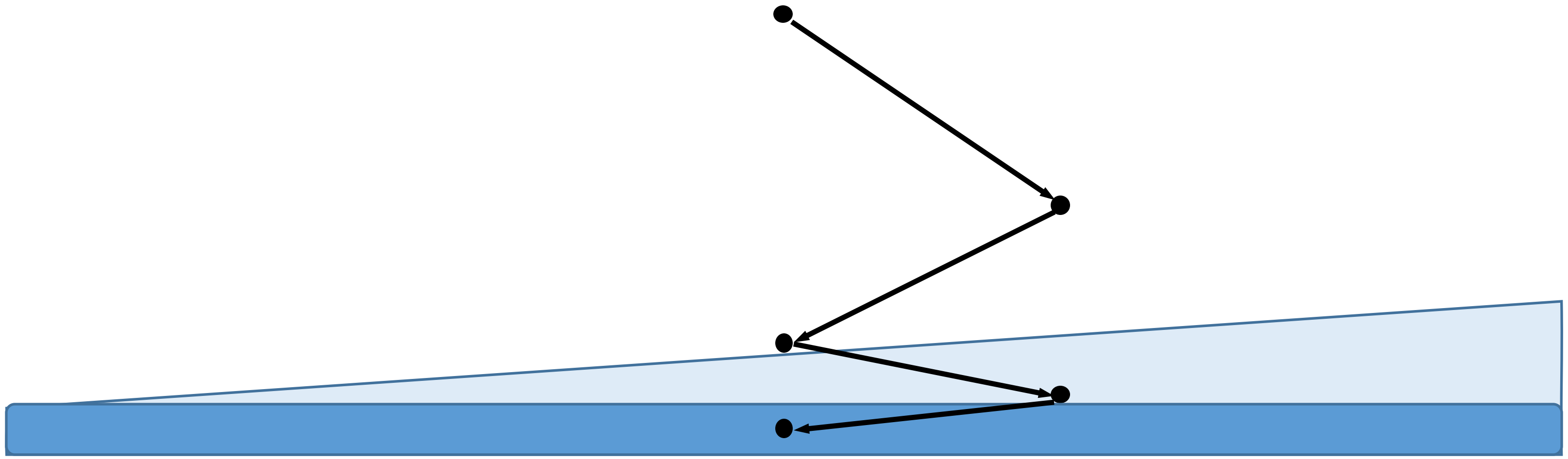
[Argue Bubeck Cohen Gupta Lee 19

but alas…

- ALG unbounded
- $OPT = O(1)$
- Not competitive

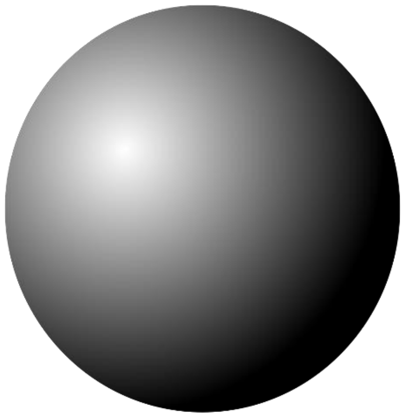
move to centroid of K_t

Grunbaum's Theorem
  half-space cuts off centroid ⇒
    volume decreases $(1 - 1/e)$



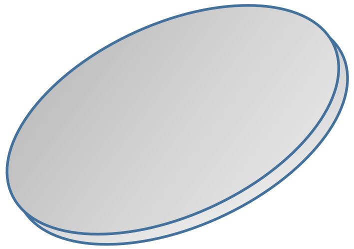
[Argue Bubeck Cohen Gupta Lee 19

fix using recursive centroid: a sketch



suppose recursive cuts don't reduce diameter ⇒ make body into "pancake"

fat directions: width more than $1/\mathrm{poly}(d)$



project out thin directions, run centroid algo on fat directions

thin directions are "thin enough" ⇒ "movement in them controlled"

$$\left.\begin{array}{l} O(\log d) \text{ steps to get another thin direction.} \\ d \text{ directions.} \end{array}\right\} \Rightarrow O(d \log d)$$

[Argue Bubeck Cohen Gupta Lee 19]

results for nested case

Theorem: [Argue Bubeck Cohen Gupta Lee 2019]

Recursive centroid algorithm is $O(d \log d)$ competitive.

Proof idea: use volume (projected onto "fat" directions) as potential function.
projections increase it, but Grunbaum cuts decrease it.

Theorem: [Bubeck Klartag Lee Li Sellke 2020]

Gaussian version of recursive centroid is $O(\sqrt{d \log d})$ competitive.

Almost tight for Euclidean norm

Algorithm idea: centroid with respect to Gaussian measure, dampens movement, retains volume drop.

how to generalize to the non-nested case?

a breakthrough for the general case…

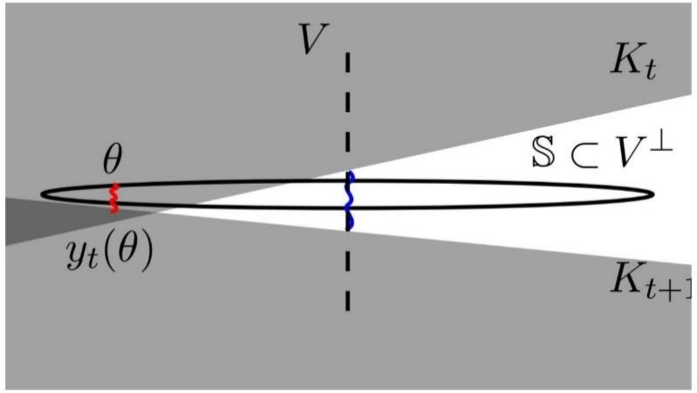
Theorem: [Bubeck Lee Li Sellke 2020]

(substantial extension of) recursive centroid algorithm is $2^{O(d)}$ competitive.

Proof:

It's complicated.

Contains several clever ideas, we'll discuss another day.



instead let's approach the problem via a different angle…

rest of the talk: a simpler, better result

Theorem: [Argue Gupta Guruganesh Tang 2020] [Sellke 2020]

the work function Steiner point algorithm is $O(d)$-competitive.

Towards the General Case

Another Algo for Nested Case

another algo for nested case

support function of convex body

Steiner point of convex body

⇒ new O(d)-competitive algo for nested convex bodies

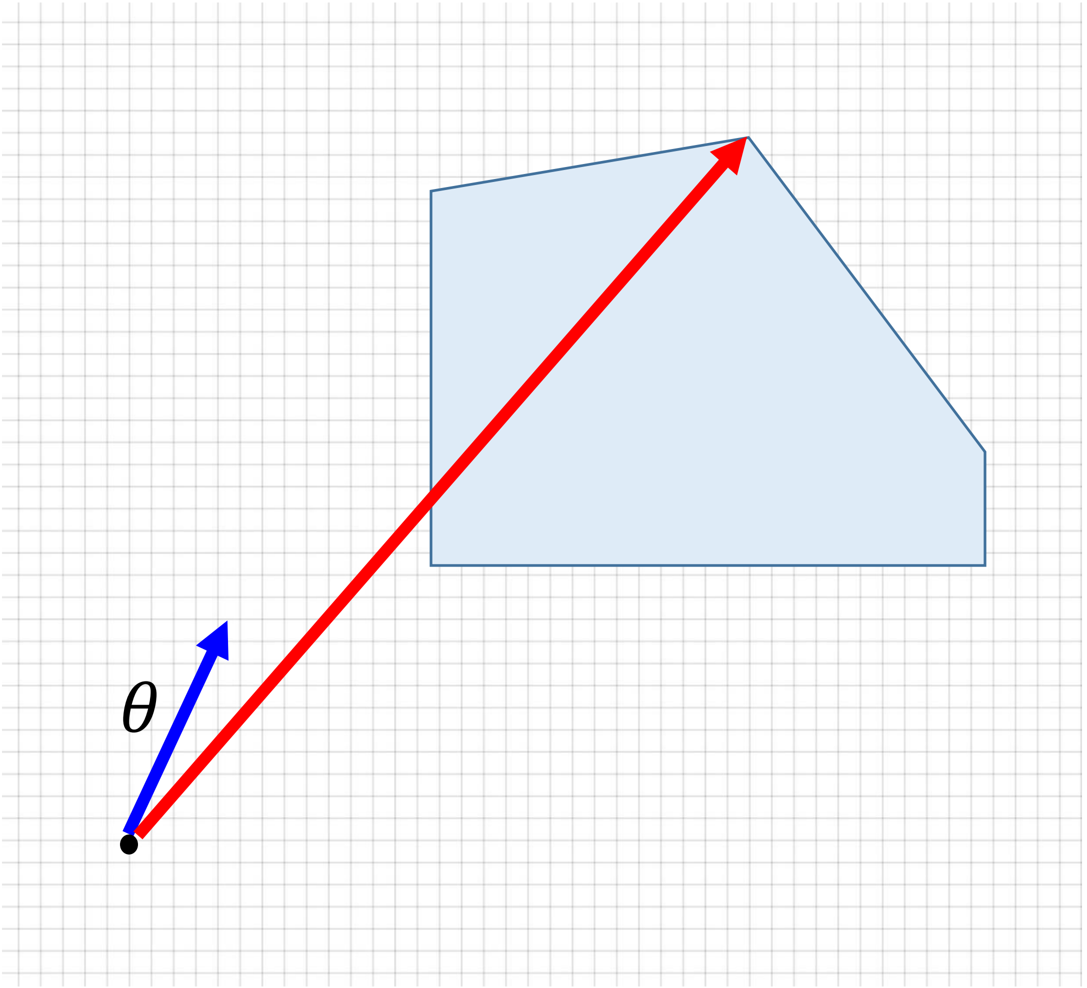
support function $h_K(\theta)$ of convex body

$$h_K(\theta) \coloneqq \max_{x \in K} \langle \theta, x \rangle$$

inner product with farthest point in K
in direction of θ

$$\nabla h_K(\theta) \coloneqq \arg\max_{x \in K} \langle \theta, x \rangle$$

farthest point in K in direction of θ



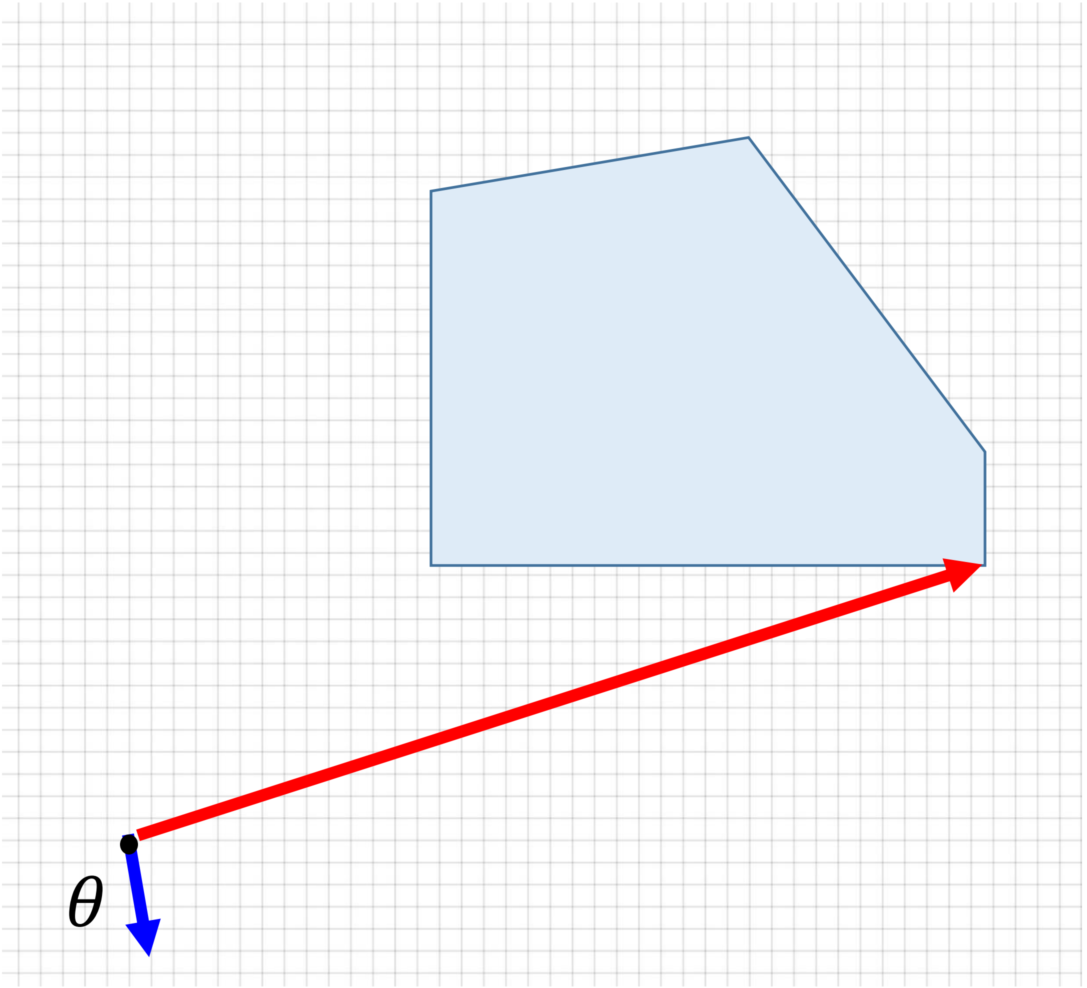
support function $h_K(\theta)$ of convex body

$$h_K(\theta) \coloneqq \max_{x \in K} \langle \theta, x \rangle$$

inner product with farthest point in K in direction of θ

$$\nabla h_K(\theta) \coloneqq \arg\max_{x \in K} \langle \theta, x \rangle$$

farthest point in K in direction of θ



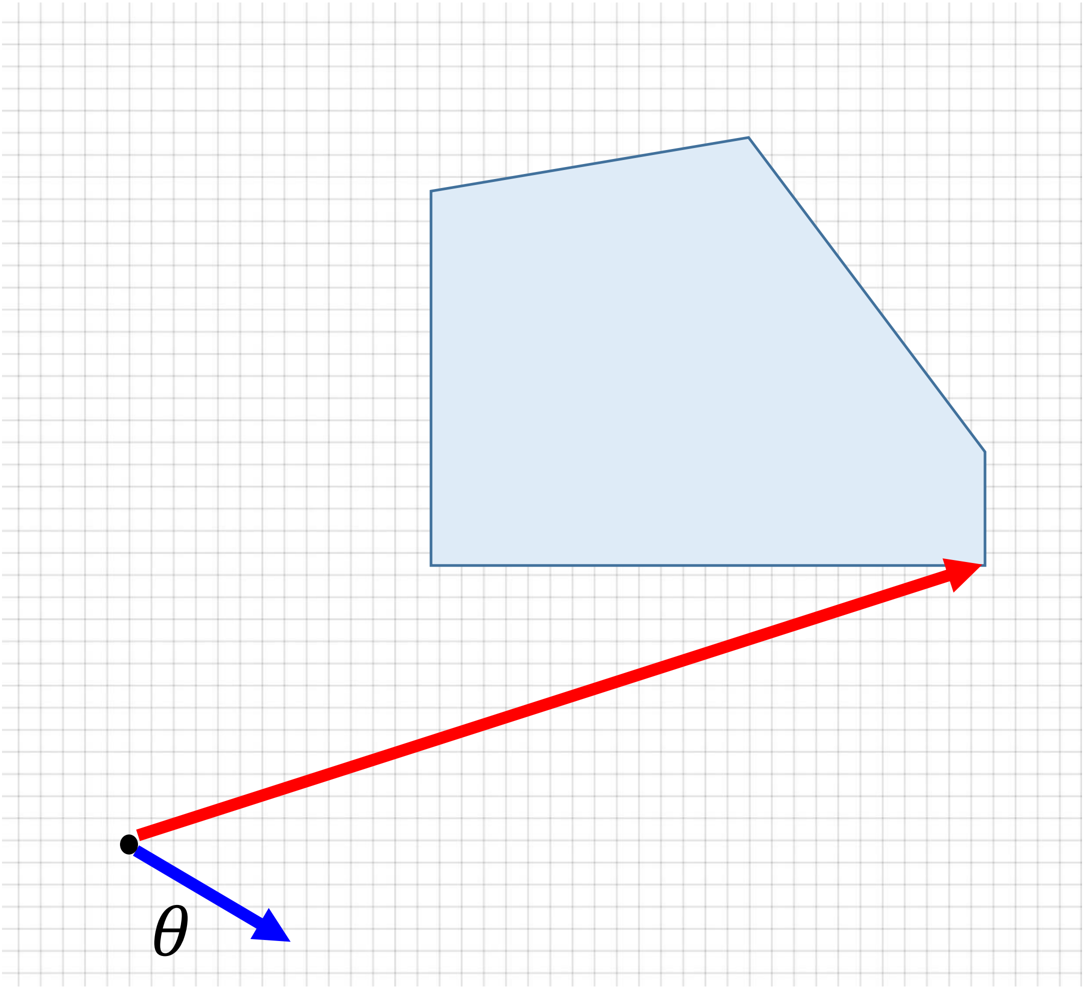
support function $h_K(\theta)$ of convex body

$$h_K(\theta) \coloneqq \max_{x \in K} \langle \theta, x \rangle$$

inner product with farthest point in K
in direction of θ

$$\nabla h_K(\theta) \coloneqq \arg\max_{x \in K} \langle \theta, x \rangle$$

farthest point in K in direction of θ



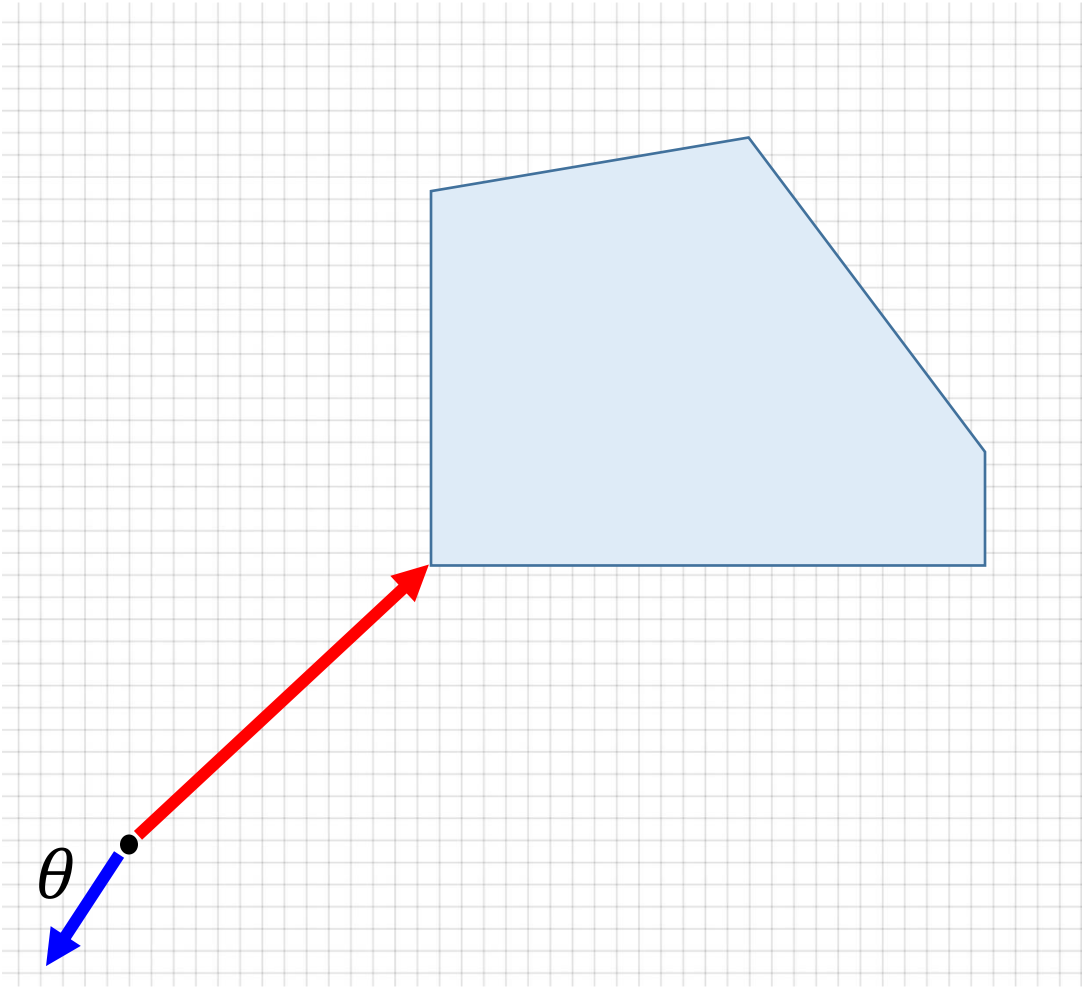
support function $h_K(\theta)$ of convex body

$$h_K(\theta) \coloneqq \max_{x \in K} \langle \theta, x \rangle$$

inner product with farthest point in K in direction of θ

$$\nabla h_K(\theta) \coloneqq \arg\max_{x \in K} \langle \theta, x \rangle$$

farthest point in K in direction of θ



another algo for nested case

support function of convex body

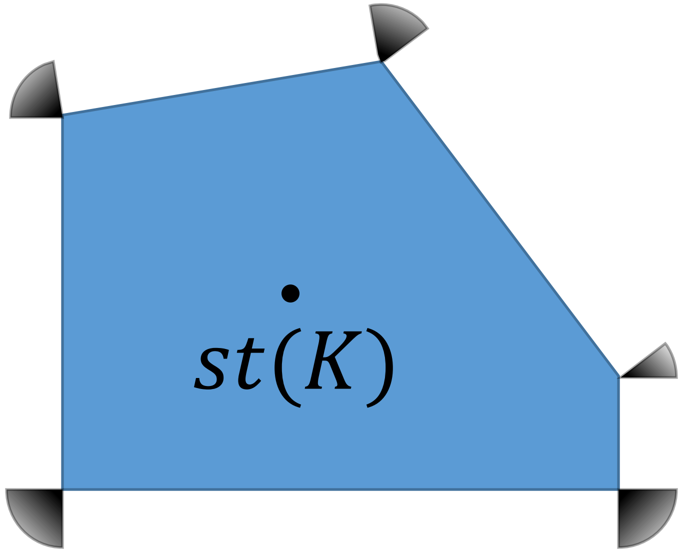
Steiner point of convex body

⇒ new O(d)-competitive algo for nested convex bodies

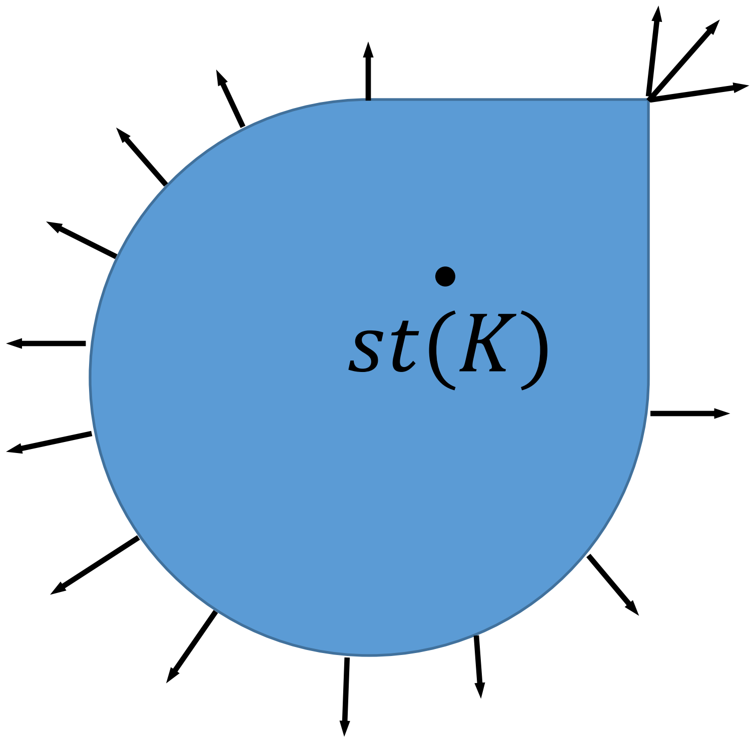
the Steiner point

Alternate "center" of convex body

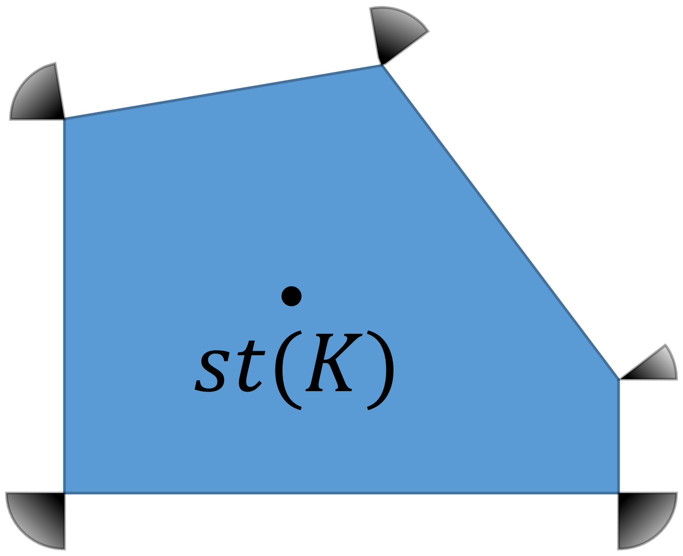
Introduced by Jakob Steiner in 1840



Average of extreme points
in all directions



⇒ Average of extreme points
weighted by size of normal cone



the Steiner point

$$h_K(\theta) := \max_{x \in K} \langle \theta, x \rangle$$

$$\nabla h_K(\theta) := \operatorname*{argmax}_{x \in K} \langle \theta, x \rangle$$

$$st(K) = \int_{\|\theta\|=1} \nabla h_K(\theta)\, d\theta$$

$$= \mathbb{E}_{G \sim N(0,1)^d}[\nabla h_K(G)]$$

$$= \int_g \nabla h_K(g)\, d\mu(g)$$

μ = density function for
d-dimensional standard Gaussian

an equivalent definition

$$h_K(\theta) \coloneqq \max_{x \in K} \langle \theta, x \rangle$$

$$\nabla h_K(\theta) \coloneqq \operatorname{argmax}_{x \in K} \langle \theta, x \rangle$$

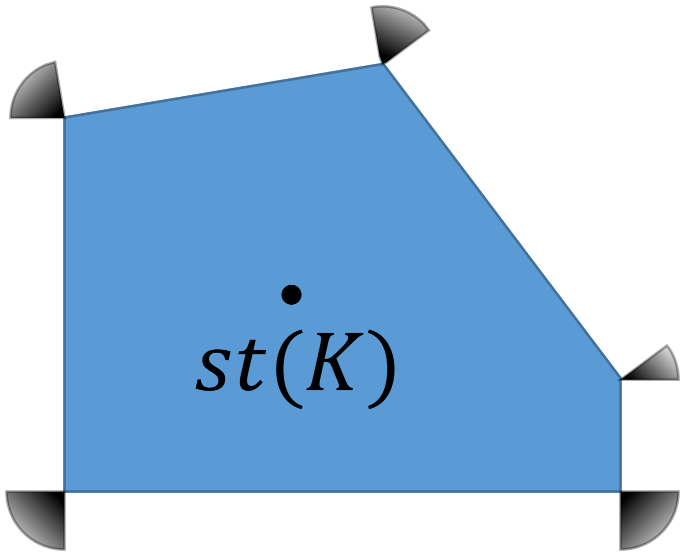
$$G \sim (N(0,1), N(0,1), \ldots, N(0,1))$$

Visually intuitive

$$st(K) = \mathbb{E}_G[\, \nabla h_K(G)]$$

$$= \mathbb{E}_G[\, G\, h_K(G)]$$

Algebraically useful

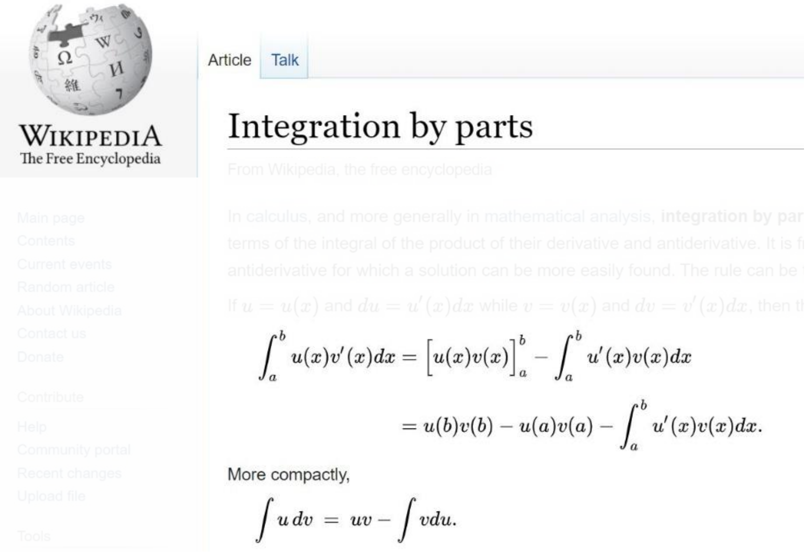


proving the equivalence

$$\mathbb{E}_G[\,\nabla f(G)] \;=\mathbb{E}_G[\,G\,f(G)]$$

In 1-dimension:

$$\mathbb{E}_G[\,f'(G)] \qquad = \int f'(x)\,\mu(x)\,dx$$



Integration by parts

$$\int_a^b u(x)v'(x)dx = \Big[u(x)v(x)\Big]_a^b - \int_a^b u'(x)v(x)dx$$

$$= u(b)v(b) - u(a)v(a) - \int_a^b u'(x)v(x)dx.$$

More compactly,

$$\int u\,dv \;=\; uv - \int v du.$$

proving the equivalence

$$\mathbb{E}_G[\,\nabla f(G)] = \mathbb{E}_G[\,G\,f(G)]$$

$\mu(x) \propto e^{-x^2/2}$

$\Rightarrow\ \mu'(x) \propto e^{-x^2/2} \cdot (-2x/2)$

In 1-dimension:

$$\mathbb{E}_G[\,f'(G)] = \int f'(x)\,\mu(x)\,dx$$

$$= f(\infty)\mu(\infty) - f(-\infty)\mu(-\infty) - \int f(x)\,\mu'(x)\,dx$$

$$= -\int f(x)\,(-x\,\mu(x))\,dx$$

$$= \int x\,f(x)\,\mu(x)\,dx$$

$$= \mathbb{E}_G[\,G\,f(G)]$$

simple case of
Gaussian integration-by-parts

Steiner point

$$h_K(\theta) \coloneqq \max_{x \in K} \langle \theta, x \rangle$$

$$\nabla h_K(\theta) \coloneqq \operatorname{argmax}_{x \in K} \langle \theta, x \rangle$$

Visually intuitive

$$st(K) = \mathbb{E}_G[\, \nabla h_K(G)]$$

$$= \mathbb{E}_G[\, G \; h_K(G)]$$

Algebraically useful

another algo for nested case

support function of convex body

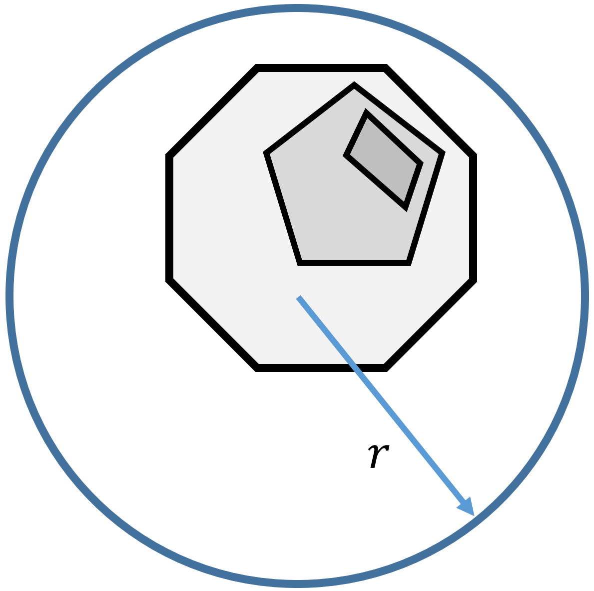
Steiner point of convex body

⇒ new O(d)-competitive algo for nested convex bodies

remember: suffices to solve bounded case (nested)

Given: nested convex sets $B(0, r) = K_0 \supset K_1 \supset \ldots \supset K_t$

Want: $x_t \in K_t$ and $ALG \leq f(d) \cdot r$

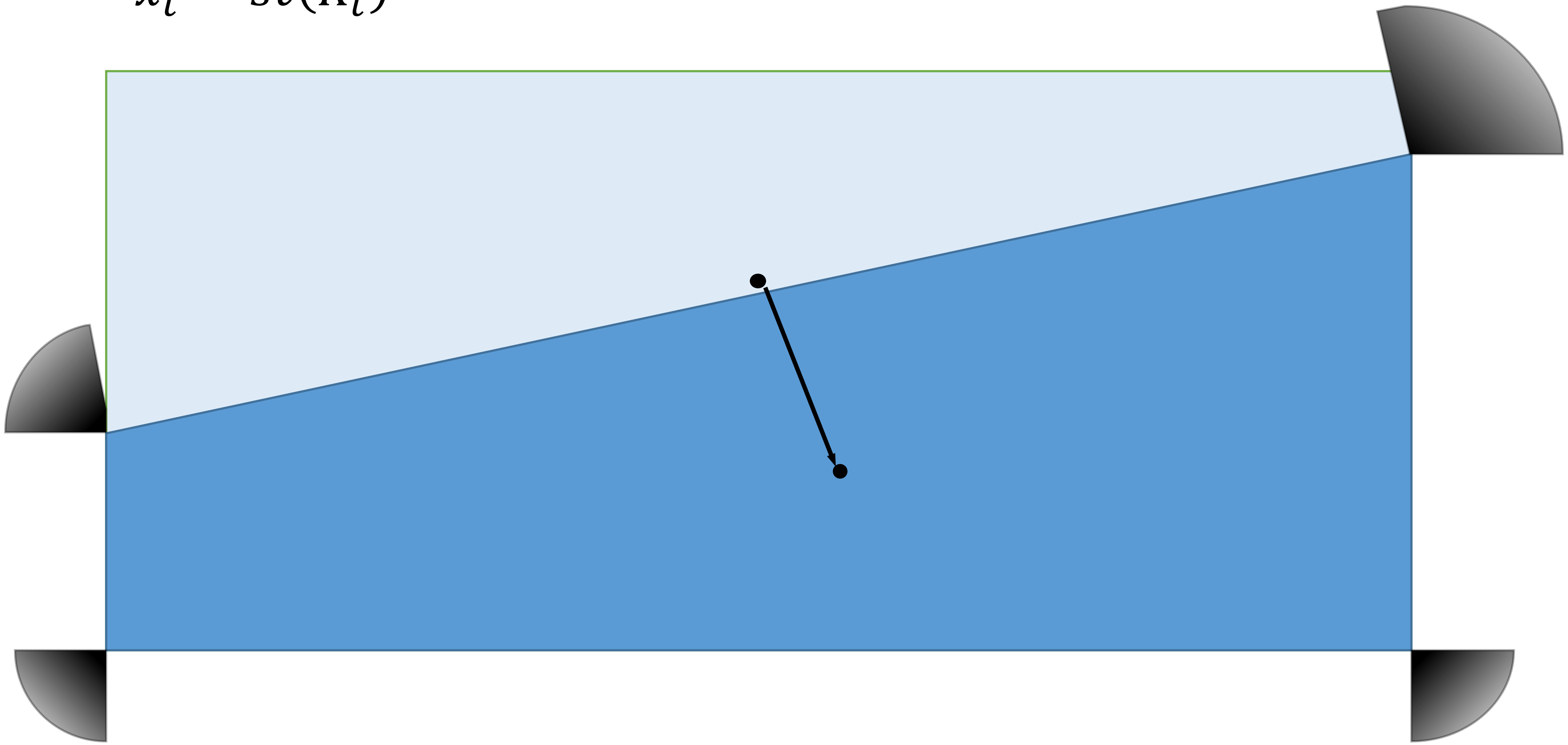


Move to the Steiner point of K_t

[Bubeck Klartag Lee Li Sellke 2020]

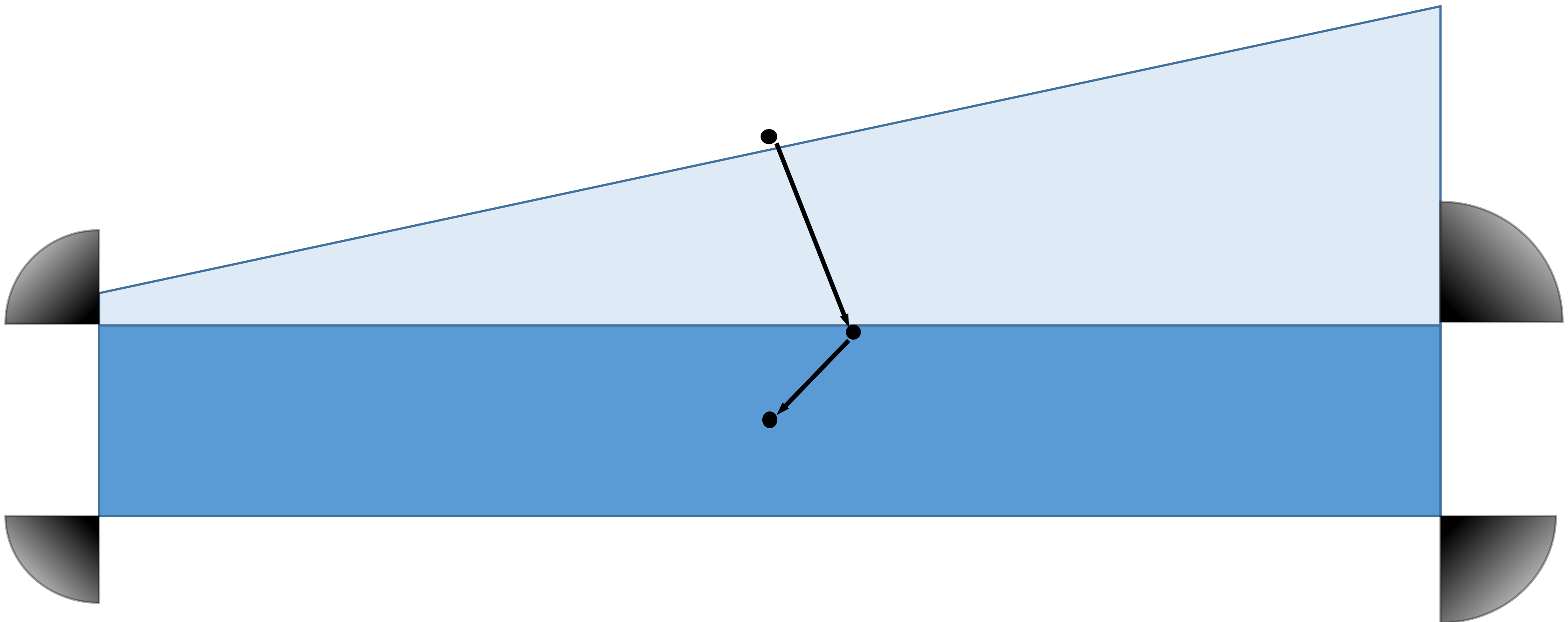
bad example (revisited)

$$x_t = st(K_t)$$



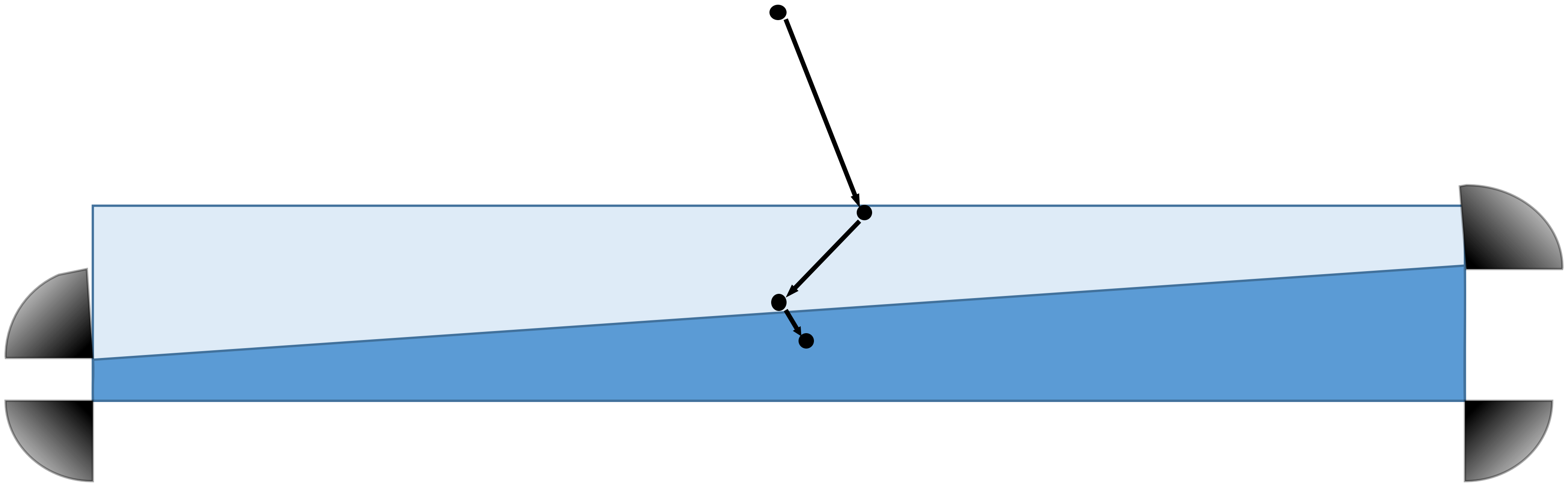
bad example (revisited)

$$x_t = st(K_t)$$



bad example (revisited)

$$x_t = st(K_t)$$

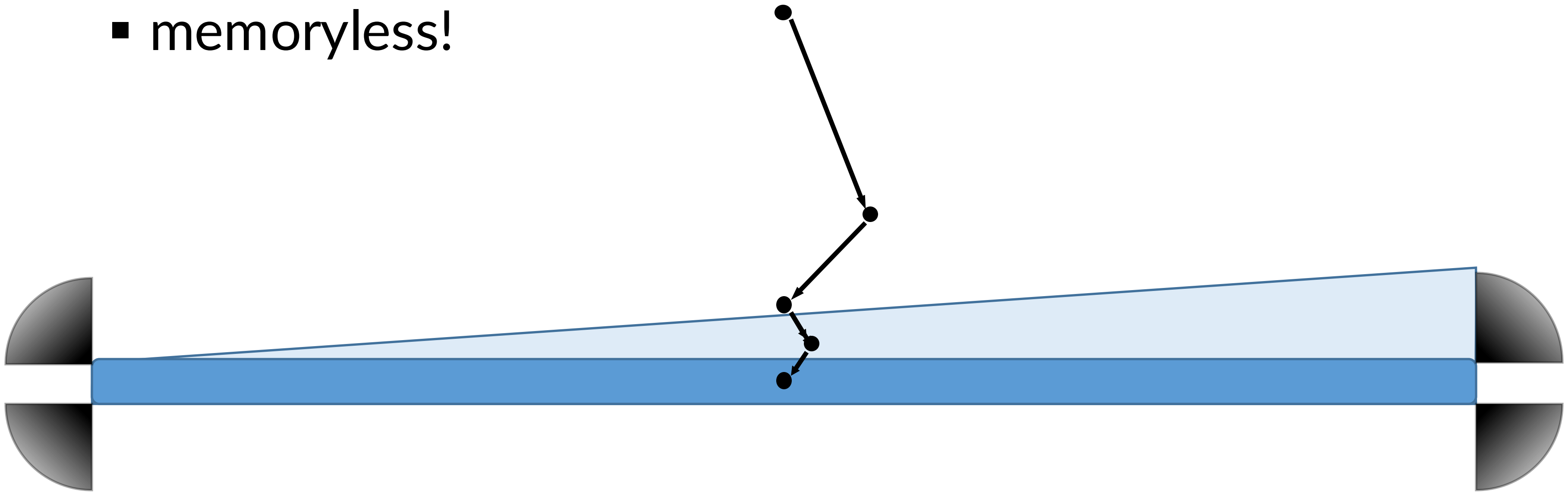


bad example (revisited)

$x_t = st(K_t)$

Steiner point algo: smoother version of recursive centroid

- $O(d)$ competitive!
- memoryless!



d-competitive for nested case

Move to the Steiner point of K_t

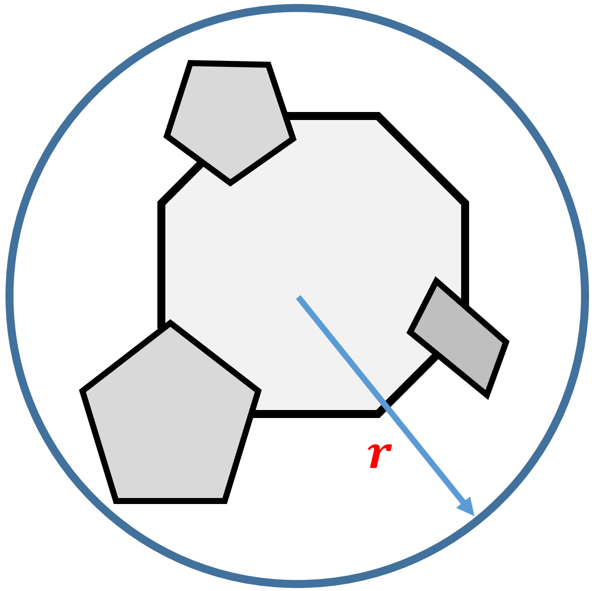
[Bubeck Klartag Lee Li Sellke 2020]

Finally: General Case

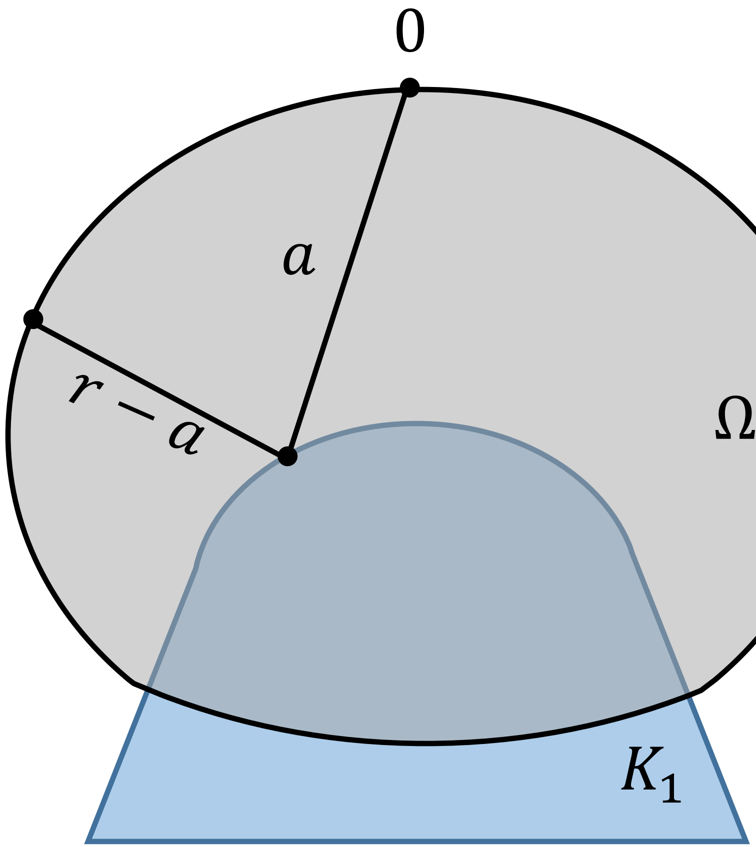
suffices to solve bounded case (non-nested)

Given: convex sets $B(0,\boldsymbol{r}) = K_0, K_1, K_2, \ldots, K_t$

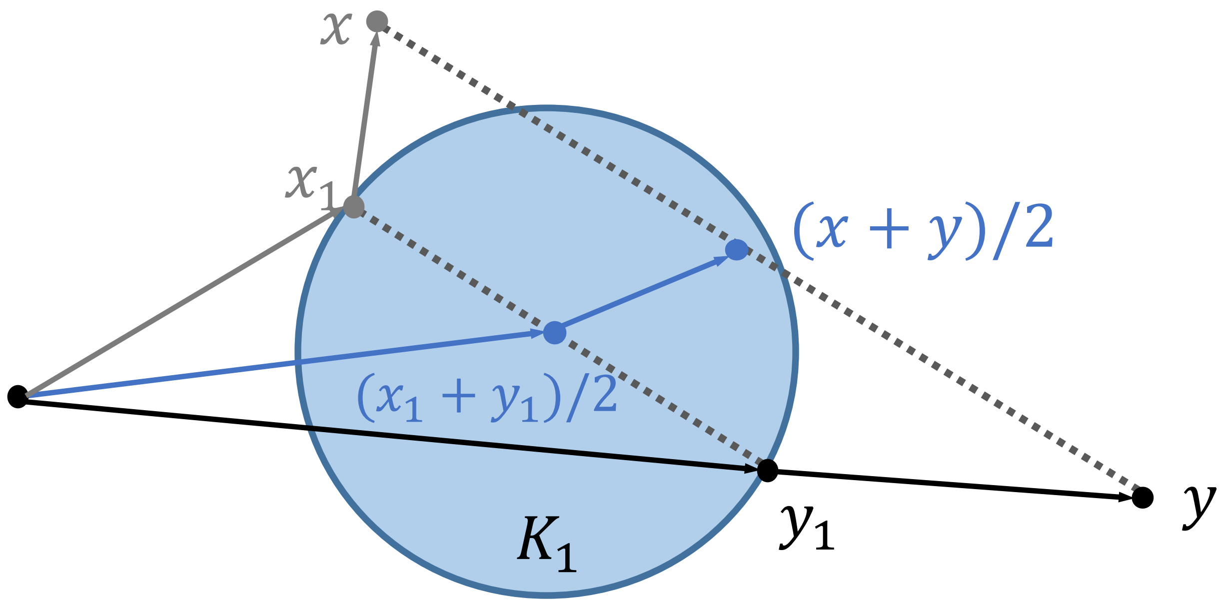
Want: $x_t \in K_t$ and $ALG \leq f(d) \cdot \boldsymbol{r}$

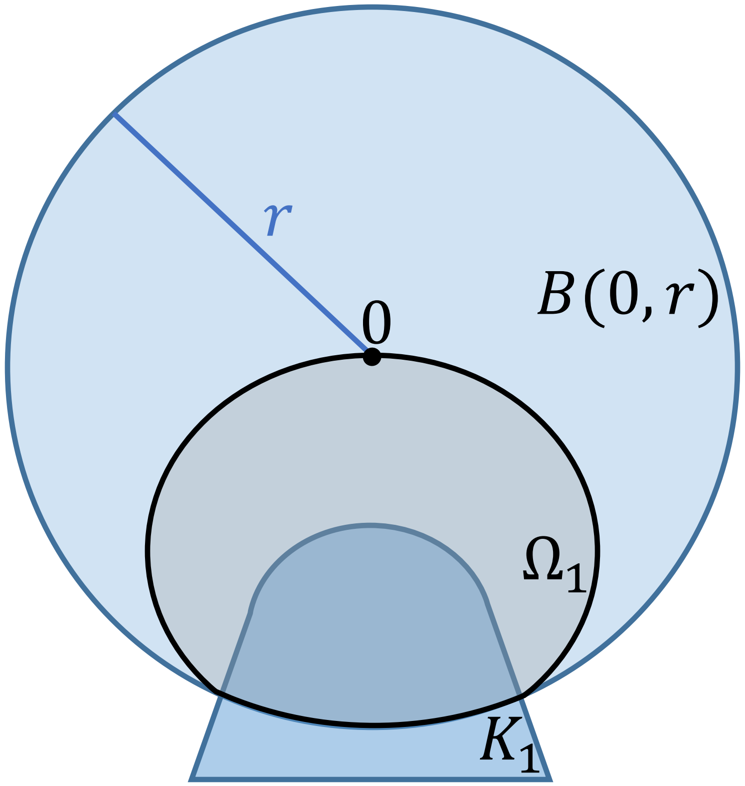


Define $\Omega_t :=$ {where OPT might be at time t, having paid $\leq r$}



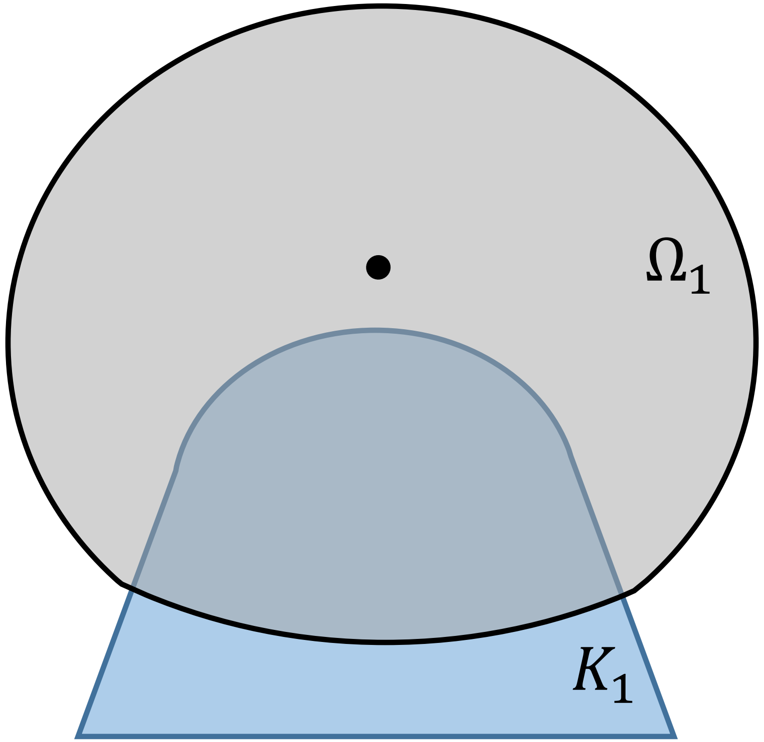
Ω_t form convex, nested sets





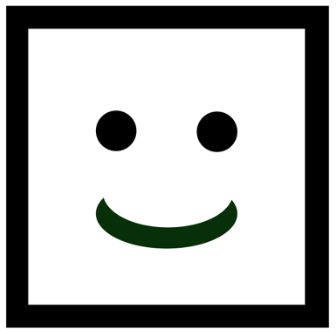
$\Rightarrow f(d)$-competitive algo for nested case on $\Omega_1, \ldots, \Omega_T$ pays $f(d) \cdot r$

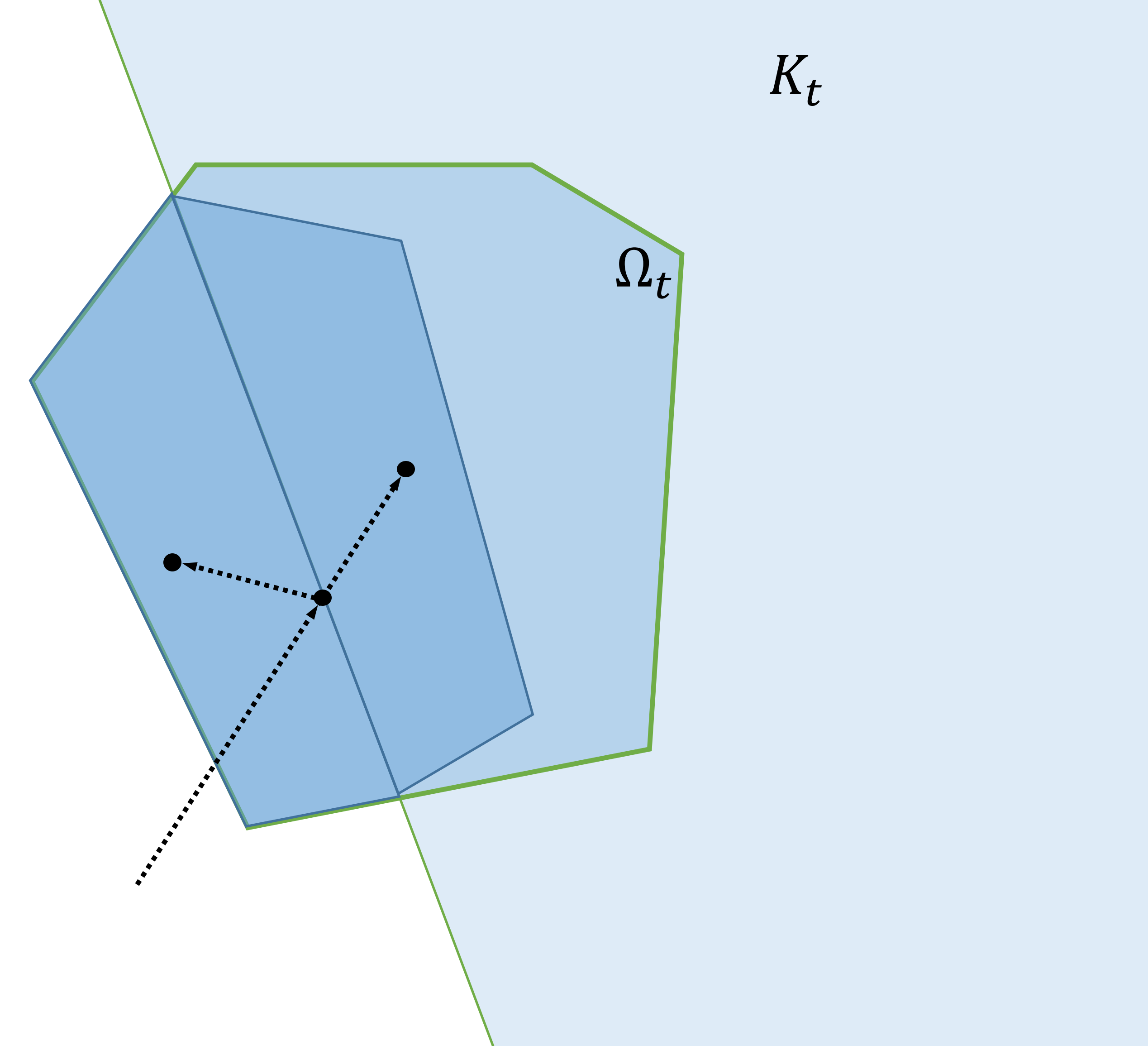
Black-box algorithm to chase Ω_t may be infeasible



$x_t \in \Omega_t$ but is $x_t \in K_t$?

Key lemma: $st(\Omega_t) \in K_t$





that proves

Theorem: [Argue Gupta Guruganesh Tang 2020] [Sellke 2020]

the work function Steiner point algorithm is $O(d)$ competitive.

functional Steiner point

Given a convex function f

$$st(f) \coloneqq \int_{\theta \in B^*} \nabla f^*(\theta) d\theta$$

where $f^*(\theta)$ = Fenchel dual, and $B^* \coloneqq$ dual space unit ball

Can define work function for function chasing
Algo: move to functional Steiner point of work function

- doesn't need guess-and-double
- works for all norms
- gives general unified view, potentially useful in other contexts

[Sellke SODA 20]

other directions

Chasing lines and subspaces

can reduce chasing k-dim affine subspaces in $\mathbb{R}^d$ to O(k)-dim chasing. [Argue Guruganesh Gupta]

see also [Antoniadis Barcelo Nugent Pruhs Schewior Scquizzato] [Bienkowski Byrka Coester Jez Koutsoupias]

Multi-server chasing

2-servers 2-d not chaseable, but special cases (k-median/means) doable [Bubeck Rabani Sellke]

Chasing well-conditioned functions

can chase fns having condition number κ with comp.ratio $O(\sqrt{\kappa})$

lower bd of $\kappa^{1/3}$. Close gap? [Argue Guruganesh Gupta]

can chase locally-polyhedral fns, other classes [Chen Goel Wierman, Goel Lin Sun Wierman]

wrap-up, and open questions

- $O(d)$ competitive algo for general convex body chasing algorithm
- $O(\sqrt{d \log d})$ competitive algo for nested bodies
- $\Omega(d^{1/2})$ lower bound

- better algorithms for special classes of convex body chasing?
  - e.g., faces of a given polytope (arises in k-server, paging)?
- broader classes of multiserver chasing?
- tight bounds for well-conditioned function-chasing?
- deeper understanding of connections to online learning?

lecture plan

Lecture #1: Set Cover (worst case)

Lecture #2: Set Cover (beyond worst case), Network design (both)

Lecture #3: Resource Allocation (aka packing)

Lecture #4: Search Problems (aka chasing)