IRP in DCG: Geometric Local Search

Lecture 1: Partitioning—Combinatorial and Spatial

1.1 Balls in $\mathbb{R}^d$

The structure underlying most of the algorithms in these lecture is a separator. Over the past several years, it has been realized that 'small-sized' separators exist for not only planar graphs, but a wider variety of graphs generally derived from geometric configurations. Two key such graphs are intersection graphs of balls, and Voronoi diagrams. The material in this section is taken from the papers [3, 1, 2].

$\star \quad \star \quad \star$

Theorem 1.1. *Given a set $\mathcal{D}$ of n disjoint balls in $\mathbb{R}^d$, there exists a d-dimensional cube T such that $i)$ $O(n^{1-1/d})$ balls of $\mathcal{D}$ intersect the boundary of T, and $ii)$ at most $dn/(d+1)$ balls of $\mathcal{D}$ lie completely in the interior (resp. exterior) of T.*

Proof.

For simplicity, we prove a weaker bound, where the number of balls lying completely inside or outside T is at most $2^d n/(2^d+1)$. Note that it is trivial to find a cube satisfying each of these two criteria separately. The key idea in showing the existence of a cube satisfying both is to first find a 'large-enough' parameterized family $\mathcal{F}$ of cubes where each cube in $\mathcal{F}$ has at least $2^d n/(2^d+1)$ ball *centers* in its interior and exterior. Then a pigeonholing argument shows the existence of a cube in $\mathcal{F}$ intersecting few balls of $\mathcal{D}$.

For a cube T, let $r(T)$ and $c(T)$ denote the radius and center of T. The family $\mathcal{F}$ is constructed as follows. Given the set of balls $\mathcal{D}$, let T_0 be the smallest radius cube that contains at least $n/(2^d+1)$ centers of the balls in $\mathcal{D}$. By scaling everything, one can assume that $r(T_0)=1$. See Figure 1.1 for an illustration in $\mathbb{R}^2$. Then the family $\mathcal{F}$ consists of all cubes with center $c(T_0)$, and radius at most 2:

$$\mathcal{F} = \{T \ \ s.t. \ \ 1 \leq r(T) \leq 2, \ \ c(T) = c(T_0)\}$$

Note that the extremal property of T_0 implies that any cube T in $\mathbb{R}^d$ with radius 1 contains at most $n/(2^d+1)$ points; otherwise by slightly decreasing the radius of T, one gets a cube with radius less than 1 which still contains at least $n/(2^d+1)$ points, a contradiction to the minimality of T_0. This immediately implies the following:

Lemma 1.2. For any cube $T \in \mathcal{F}$, the number of centers of balls in $\mathcal{D}$ contained in the interior (resp. exterior) of T is at most $n/(2^d+1)$.

Proof. As each $T \in \mathcal{F}$ contains T_0, it contains at least $n/(2^d+1)$ centers in its interior. On the other hand, each such T can be covered by 2^d translates of T_0, as $r(T) \leq 2$; by the choice of T_0, each such translate must contain at most $n/(2^d+1)$ centers. Thus T can contain at

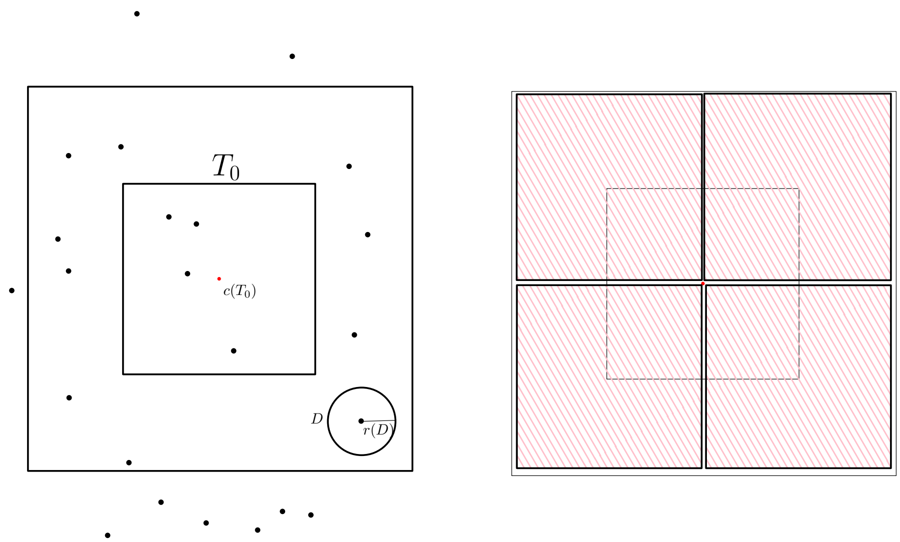


Figure 1.1: An illustration in $\mathbb{R}^2$. (left) The unit square T_0 with center $c(T_0)$, and (right) larger square of radius 2 can be tiled with 4 copies of T_0.

most $2^d n/(2^d+1)$ centers in its interior, implying at least $n - 2^d n/(2^d+1) = n/(2^d+1)$ centers lie in its exterior. $\square$

It remains to show that there exists a $T \in \mathcal{F}$ whose boundary intersects few balls of $\mathcal{D}$. To that end, we count small- and large-radius balls separately:

$$\mathcal{D}_1 = \left\{ D \in \mathcal{D} \;\middle|\; r(D) \le \frac{1}{n^{1/d}} \right\}, \qquad \mathcal{D}_2 = \left\{ D \in \mathcal{D} \;\middle|\; r(D) > \frac{1}{n^{1/d}} \right\}$$

As each $T \in \mathcal{F}$ has radius at most 2, a simple packing argument shows that the boundary of each $T \in \mathcal{F}$ intersects $O((n^{1/d})^{d-1}) = O(n^{1-1/d})$ balls in $\mathcal{D}_2$. Now pick a random cube $T_R \in \mathcal{F}$, with center $c(T_0)$, and radius a random value in $[1,2]$. Let X_D be a random variable which is 1 iff T_R intersects $D \in \mathcal{D}_1$. Then the expected number of balls of $\mathcal{D}_1$ intersected by T_R is:

$$\mathrm{E}\Big[\sum_{D\in\mathcal{D}_1} X_D\Big] = \sum_{D\in\mathcal{D}_1} \mathrm{E}[X_D] = \sum_{D\in\mathcal{D}_1} Pr[T_R \text{ intersects } D] \le \sum_{D\in\mathcal{D}_1} 2r(D) \le \sum_{D\in\mathcal{D}_1} \frac{2}{n^{1/d}} = O(n^{1-1/d})$$

where the last inequality follows from the fact that $r(D) \le 1/n^{1/d}$ for $D \in \mathcal{D}_1$.

Altogether, a random square $T_R \in \mathcal{F}$ intersects $O(n^{1-1/d})$ balls in $\mathcal{D}_2$, and expected $O(n^{1-1/d})$ balls in $\mathcal{D}_1$. Therefore there must exist a square $T \in \mathcal{F}$ that intersects at most $O(n^{1-1/d})$ balls of $\mathcal{D}$; and it contains at least $n/(2^d+1)$ centers of $\mathcal{D}$ in its interior and exterior. $\square$

Variants of the planar graph separator theorem

One can generalize the above proof to the case where each disk $D \in \mathcal{D}$ has a weight, and the goal is to remove $O(\sqrt{n})$ vertices from the planar graph such that the resulting two pieces have weight at most $2/3$-rds of the total weight.

Theorem 1.3. *Given a planar graph $G = (V, E)$, and a weight function $w : V \to \mathbb{R}$, one can partition V into three sets A, B, C such that $|C| = O(\sqrt{n})$, there are no edges between vertices of A and B, and $w(A), w(B) \leq 2/3 \cdot W(V)$.*

If the planar graph G is triangulated, the separator can be shown to have additional structure: from the Koebe-Andreev-Thurston circle packing theorem, the set of disks $\mathcal{D}$ intersected by any Jordan curve in the plane must form a cycle in G (otherwise one can add further edges to G among the intersected disks). This immediately implies the following:

Theorem 1.4. *Given a triangulated planar graph $G = (V, E)$, and a weight function $w : V \to \mathbb{R}$, there exists a cycle C of length $O(\sqrt{n})$ in G such that for the set A, B of vertices inside/outside C, we have $w(A), w(B) \leq 2/3 \cdot W(V)$.*

Lastly, separators are known to exist for broader category of graphs than just planar graphs; e.g., for K_h-minor free graphs one gets:

Theorem 1.5. *Let $G = (V, E)$ be a K_h-minor free graph on n vertices. Then G has a separator of size $O(h\sqrt{n})$.*

[1] Vijay V. S. P. Bhattiprolu and Sariel Har-Peled. Separating a voronoi diagram via local search. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, pages 18:1–18:16, 2016.

[2] Timothy M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2):178–189, 2003.

[3] Warren D. Smith and Nicholas C. Wormald. Geometric separator theorems & applications. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 232–243, 1998.

Questions

(solutions)

1. We will show that, given n disjoint disks $\mathcal{D}$ in $\mathbb{R}^2$, there exists a rectangle that intersects $O(\sqrt{n})$ disks of $\mathcal{D}$, and contains at most $2n/3$ centers of $\mathcal{D}$ on both sides. Consider the smallest axis-parallel rectangle, say T, with side-length ratios $4:3$ containing $2n/3$ centers of $\mathcal{D}$ in its interior.
   (a) Show there exists a $2:3$ ratio rectangle $T' \subset T$ containing at least $n/3$ centers.
   (b) By scaling and translation, assume T''s bottom-left and top-right coordinates are $(0,0)$ and $(2,3)$. Then show that each rectangle in the family $\mathcal{F}$ of rectangles with bottom-left coordinate $(-\delta,-\delta)$ and and top-right coordinate $(3-\delta, 4-\delta)$, $0 \le \delta \le 1$, contains least $n/3$ centers inside and outside.
   (c) Prove that one of these rectangles intersects $O(\sqrt{n})$ disks.
2. Extend the proof of Question 1 to show the existence, given n disjoint spheres in $\mathbb{R}^d$, of an axis-parallel cuboid in $\mathbb{R}^d$ with side-length ratios $(d+1):(d+2):\ldots:2d$ that intersects $O(n^{1-1/d})$ spheres, and contains at most $2n/3$ centers completely inside and outside.
3. Let P be a set of n points in $\mathbb{R}^2$. We will prove that there exists a set $Q \subset \mathbb{R}^2$ of $O(\sqrt{n})$ points which acts, in a sense, as a separator for the Voronoi diagram of P: that there exists a partition $P = P_1 \cup P_2$, $|P_1|, |P_2| \le 2n/3$, such that in the Voronoi diagram of $P \cup Q$, there is no edge between $p_i \in P_1$ and $p_j \in P_2$. Call Q a Voronoi separator for P.
   (a) Let T_0, T_R be as defined in the text, and let $P' \subset P$ be the set within distance $1/\sqrt{n}$ to T_R. Prove that the set Q_1 of $O(\sqrt{n})$ uniformly placed points on T_R will be a Voronoi separator for $P \setminus P'$.
   (b) Prove that $\mathrm{E}[|P'|] = O(\sqrt{n})$, and that one can pick an additional $O(\sqrt{n})$ points Q_2 such that $Q_1 \cup Q_2$ is a Voronoi separator for P.
4. Let $\mathcal{D}$ be a set of n balls in $\mathbb{R}^2$ such that no point in the plane is covered by more than κ balls of $\mathcal{D}$. Then prove that there exists a square intersecting $O(\sqrt{\kappa n})$ disks of $\mathcal{D}$, and containing at least $n/4$ disks completely inside and outside.
5. Let P be a set of n points in $\mathbb{R}^2$. Prove that there exists a partition of P into $R \cup S \cup B$ so that $i)$ $|R|, |B| \le 3n/4$, $ii)$ $|S| = O(\sqrt{n})$, and $iii)$ for any disk D in the plane, if D contains at least one point of R and B, then it contains at least one point of S.

1.2 Partitioning Planar Graphs

We now show that separators can be used iteratively to partition a graph into nearly equal-sized pieces with 'little' interaction among these pieces. The material in this section is taken from the papers [2, 1].

$\star \quad \star \quad \star$

The planar graph separator theorem partitions G, by removing a small-sized subset, into two roughly equal-sized subsets with no edges between them. In this section we show that by applying it repeatedly, for any integer r, one can get a partition of G into $\Theta(n/r)$ subsets. Specifically we will prove the following.

Theorem 1.6. *Given a planar graph $G = (V, E)$ on n vertices and an integer r, there exist subsets $V_1, \ldots, V_{n/r}$, where $V = \bigcup V_i$ and $|V_i| = O(r)$. Furthermore for each $i = 1 \ldots n/r$ $V_i = V_i^a \dot{\bigcup} V_i^b$, where:*

1. ***Interior vertices** V_i^a belong to only V_i, and $N(V_i^a) \subseteq V_i$.*
2. ***Boundary vertices** V_i^b could belong to many sets and may have edges to other sets. $|V_i^b| = O(\sqrt{r})$.*

Note that by setting $r = n/2$, we get back (within constants) the planar graph separator theorem: the set $V_1^b \cup V_2^b$ is a separator of size $O(\sqrt{n})$ for the two sets V_1^a and V_2^a. Similarly if the planar graph separator theorem gives the two sets A and B with separator C, the the sets $V_1^a = A, V_1^b = C, V_2^a = B, V_2^b = C$ satisfy the above theorem for $r = n/2$.

We first prove a slightly weaker statement where the size of the boundary vertices is only bounded on average, i.e., $\sum_i |V_i^b| = O(n/\sqrt{r})$. Then by further refining each set containing too many boundary vertices one gets the required statement. The proof is constructive: given $G = (V, E)$ and the parameter r, apply the planar graph separator theorem to G to get the two sets A, B with separator C. Now if $|A \cup C| \geq r$, recursively apply the procedure on the subgraph induced by $A \cup C$; same for $B \cup C$.

Clearly when the procedure ends, all the sets have $O(r)$ vertices, and together cover all the vertices. It remains to bound the total number of boundary vertices. Let $B_r(n)$ be the total number of boundary vertices created by the procedure, for a fixed parameter r. Then,

$$B_r(n) = \begin{cases} 0, & \text{if } n < r. \\ O(\sqrt{n}) + B_r(\alpha n + O(\sqrt{n})) + B_r((1-\alpha)n + O(\sqrt{n})) & \text{otherwise.} \end{cases}$$

where $1/3 \leq \alpha \leq 2/3$.

Solving this is complicated by the fact that the recursion does not create a partition of the vertices, i.e., the total number of vertices at each level is increasing. However the additional number of vertices added at each level is sub-linear, and so does not add up to a significant factor; we first upper-bound this increase in the overall number of vertices, and then sum up the recurrence.

As the recursion tree is not perfectly balanced, summing up the values in the i-th call of the recursion is not possible, as different subproblems will have different sizes. A clever trick is to sum up the terms according to the distance from the leaf nodes. Towards this, define the height of a node v of the recursion tree to be the length of the longest path from v to a descendant leaf node. Let N_i be the number of vertices (with multiplicities) over all nodes at height i, and h the height of the root node. Let l_i be the number of nodes at height i, and n_i^j the number of vertices in the j-th node at level i. Let $N_{\leq i}$ be the total number of vertices (again, counted with multiplicities) in all the nodes v such that $i)$ v has height at most i, and $ii)$ parent of v has height greater than i. Note that the set of such nodes forms a cut in the recursion tree; furthermore, $N_i \leq N_{\leq i}$, as all the nodes of height exactly i will be counted in $N_{\leq i}$. A parent node contains at least $3/2$ times more vertices than its child, and as the parent of each leaf node contains at least r vertices, a node at height i contains at least $(3/2)^i r$ vertices of P. Therefore $l_i \leq \frac{N_i}{(3/2)^i r}$.

We will bound $N_{\leq i}$ iteratively with decreasing values of i. A node v that was counted in $N_{\leq i}$ will be counted in $N_{\leq (i-1)}$ if and only if the height of v is strictly less than i. On the other hand, if v has height exactly i, it will be replaced in $N_{\leq (i-1)}$ by the vertices in its two children nodes, both of which will have height less than i and one of them must have height exactly $(i-1)$. Therefore all the additional vertices added in $N_{\leq (i-1)}$ come from all the nodes at height i:

$$N_{\leq (i-1)} \leq N_{\leq i} + \sum_{j=1}^{l_i} \sqrt{n_i^j} \leq N_{\leq i} + \sum_{j=1}^{l_i} \sqrt{N_i/l_i} = N_{\leq i} + \sqrt{N_{\leq i} l_i} \leq N_{\leq i}\Big(1 + \frac{(2/3)^{i/2}}{\sqrt{r}}\Big)$$

Recalling that $N_{\leq h} = n$, the total number of vertices at leaf nodes $N_{\leq 1}$:

$$N_{\leq 1} = n \prod_{i=1}^{h-1} \Big(1 + \frac{(2/3)^{i/2}}{\sqrt{r}}\Big) \leq n \prod_{i=1}^{h-1} e^{\frac{(2/3)^{i/2}}{\sqrt{r}}} \leq n e^{\sum \frac{(2/3)^{i/2}}{\sqrt{r}}} \leq n e^{4/\sqrt{r}} \leq 50n$$

Therefore $N_{\leq i} \leq 50n$ for all i. Therefore the number of boundary vertices added for all the nodes at height i is $\frac{N_i}{\sqrt{r}}(2/3)^{i/2} \leq \frac{50n}{\sqrt{r}}(2/3)^{i/2}$, and the total number over all heights is a geometric sum adding up to $O(n/\sqrt{r})$.

Let $W_1, \ldots, W_k$, $k = \Theta(n/r)$ be the total number of resulting sets. Note that $|W_i| \leq r$, and the total number of boundary vertices is $O(n/\sqrt{r})$, i.e., $\sum_i |W_i^b| = O(n/\sqrt{r})$. There are $\Theta(n/r)$ sets, and overall $O(n/\sqrt{r})$ boundary vertices. Therefore the average number of boundary vertices per set is $\frac{n/\sqrt{r}}{n/r} = O(\sqrt{r})$. We now show that this average case can be achieved for each set by further refinement.

To ensure that each set does not contain more than $\sqrt{r}$ boundary vertices, for each W_i, do the following: use the separator theorem on W_i to partition it into two sets where the number of boundary vertices in each set goes down by a constant factor, i.e., it is at most $2/3 \cdot |W_i^b|$ (using Theorem 1.3). As before, add the separator, of size $O(\sqrt{|W_i|}) = O(\sqrt{r})$, to both sets and recurse. We stop the recursion as soon as a set has at most $20\sqrt{r}$ boundary vertices.

Consider the sets $V_1, \dots, V_{t_i}$ created by a fixed W_i. As calculated earlier, the overall total number of new vertices added in the recursion is a fixed constant fraction of the initial size, so assume the total number of vertices at the end is $50|W_i| \leq 50r$. Therefore the number of new boundary vertices added at each node of the recursion tree is, in the worst case, $\sqrt{50r}$. As the number of internal nodes is at most the number of leaves (t_i in our case) the total number of boundary vertices in $V_1, \dots, V_{t_i}$ is at most $|W_i^b| + t_i\sqrt{50r}$. On the other hand, the $t_i/2$ parents of each leaf have at least $20\sqrt{r}$ boundary vertices for them to have been divided further, and so

$$\frac{t_i}{2} \cdot 20\sqrt{r} \leq \text{ Total boundary vertices } \leq |W_i^b| + t_i\sqrt{50r}$$

implying that $t_i = O(|W_i^b|/\sqrt{r})$. Summing up over all i to get the total number of sets:

$$\sum_i t_i = \sum_i O(|W_i^b|/\sqrt{r}) = \frac{\sum_i O(|W_i^b|)}{\sqrt{r}} = \frac{O(n/\sqrt{r})}{\sqrt{r}} = O(n/r)$$

This iterative use of a separator theorem to construct a partition into a number of sets with limited interaction is a general idea with broader applicability; it can be used whenever one knows the existence of a separator of sub-linear size. For example, using Theorem 1.5 in the same way immediately implies:

Theorem 1.7. *Given a K_h-minor free graph $G = (V, E)$ on n vertices and an integer r, there exist subsets $V_1, \dots, V_{n/r}$, where $V = \bigcup V_i$ and $|V_i| = O(r)$. Furthermore for each $i = 1 \dots n/r$ $V_i = V_i^a \dot{\bigcup} V_i^b$, where:*

1. ***Interior vertices*** *V_i^a belong to only V_i, and $N(V_i^a) \subseteq V_i$.*
2. ***Boundary vertices*** *V_i^b belong to many sets and have edges to other sets. $|V_i^b| = O(h\sqrt{r})$.*

[1] Sergio Cabello and David Gajser. Simple ptas's for families of graphs excluding a minor. *Discrete Applied Mathematics*, 189:41–48, 2015.

[2] Greg N. Frederickson. Planar graph decomposition and all pairs shortest paths. *J. ACM*, 38(1):162–204, 1991.

QUESTIONS

(solutions)

1. Given a planar graph $G = (V, E)$ and an integer r, show that there exists a subset X of size $O(n/\sqrt{r})$ such that $V \setminus X$ can be partitioned into subsets $V_1, \ldots, V_{n/r}$ such that $|V_i| = O(r)$, there are no edges between V_i and V_j for any $1 \leq i < j \leq n/r$, and each V_i has $O(\sqrt{r})$ neighbors in X.

2. Given a set $\mathcal{D}$ of n disjoint unit disks in the plane and a parameter r, show that there exists a partition of $\mathbb{R}^2$ into n/r disjoint rectangles such that each rectangle contains $O(r)$ centers of $\mathcal{D}$ in its interior, and intersects the boundary of $O(\sqrt{r})$ disks in $\mathcal{D}$.

3. Prove Theorem 1.7.

4. Given a planar graph $G = (V, E)$, prove that one can color the vertices of G with $O(\sqrt{n})$ colors (labelled $1 \ldots$) such that for any two vertices $u, v \in V$, any path in G from u to v contains a vertex of color higher than u or v.

5. Given a tree $T = (V, E)$ with maximum degree Δ, show the existence of an edge $e \in E$ such that $E \setminus \{e\}$ partitions T into two subtrees, each of size at least $\frac{|V|}{\Delta}$.

1.3 Local Expansion in Planar Graphs

The material in this section is taken from [1, 4, 2, 3].

⋆ ⋆ ⋆

In this section we consider a Helly-type question on planar graphs: by examining neighborhoods of vertices of a given bipartite graph $G = (R, B, E)$, can one determine, within some error, the relative sizes of R and B? For example, if every subset $R' \subseteq R$ of at most k vertices has 'many' neighbors in B, does that imply that $|B|$ is 'large' compared to $|R|$? More precisely, what is the function $f(k)$ such that for a given integer k, if every subset $R' \subseteq R$ of size at most k has at least $|R'|$ neighbors in B, then $|B| \geq f(k) \cdot |R|$? Given a graph $G = (V, E)$, call a subset $V' \subseteq V$ *locally expanding* if $|N(V')| \geq |V'|$. G is called k-*locally expanding* if every subset $V' \subseteq V$ with $|V'| \leq k$ is locally expanding.

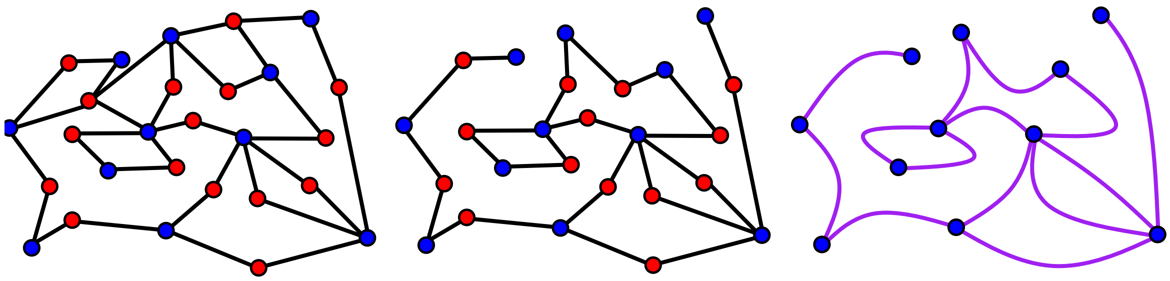
Trivially, one can deduce that $|B| \geq k$ and for general graphs, this cannot be improved. This is realized by a complete bipartite graph where $|R| = n$ and $|B| = k$; then every subset of R of size at most k has exactly k neighbors and yet B is arbitrarily small compared to R. So for general graphs, local expansion properties do not imply any global bounds. This is related to Hall's theorem: there exists a perfect matching in $G = (R, B, E)$ if and only if $|N(R')| \geq |R'|$ for *every* $R' \subseteq R$. A perfect matching implies $|B| \geq |R|$; however as the above counter-example shows, perfect matchings fail to exist if one restricts the expansion property to subsets of smaller sizes.

Note however that the counter-example is highly non-planar (many copies of $K_{3,3}$). It turns out that if we restrict ourselves to planar bipartite graphs, one can do considerably better:

Theorem 1.8. *If $G = (R, B, E)$ is a bipartite planar graph and $k \geq 3$ an integer such that for all $R' \subseteq R$ of size $\leq k$, $|N(R')| \geq |R'|$ then*

$$|R| \leq (1 + \frac{c}{\sqrt{k}})|B|$$

The theorem requires $k \geq 3$ since for $k = 1, 2$ the bipartite complete graph of $|R| = n$ and $|B| = k$ vertices is still planar. Before we give the proof of the main theorem, consider as a warm-up the proof for the next two smallest values of k.



THE CASE $k = 3$.

We are given a 3-locally expanding bipartite graph $G = (R, B, E)$; i.e., each subset of R of size at most 3 is locally expanding. Adding edges does not weaken the condition, so assume a maximal such graph. Each vertex in $R \cup B$ has degree at least 2 in such a graph.

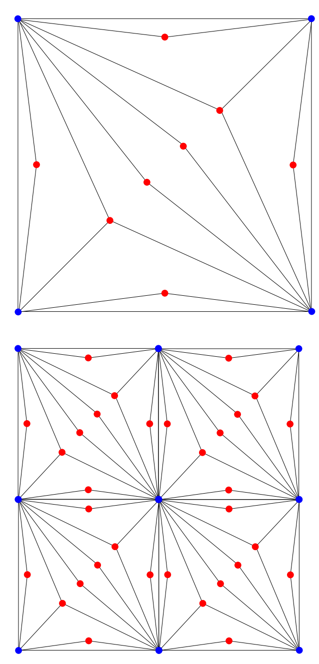
Let $R_{=2} \subseteq R$ be the set of vertices of R with degree exactly 2, and $R_{\geq 3} \subseteq R$ the ones with degree at least 3. Then as $|E| \leq 2|V| - 4$ for any bipartite planar graph (V, E),

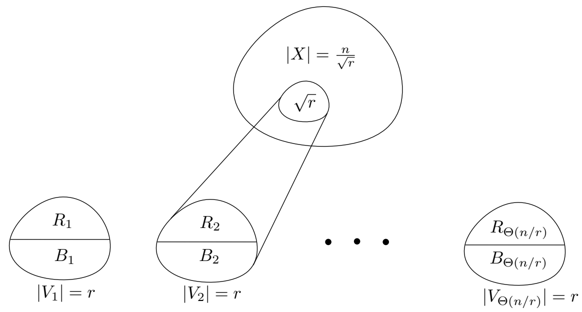
$$2|R_{=2}| + 3|R_{\geq 3}| \leq |E| \leq 2(|R_{=2}| + |R_{\geq 3}| + |B|) - 4$$

This bounds the number of vertices of degree at least 3: $|R_{\geq 3}| \leq 2|B|$.

Now let G' be the graph induced by the set $R_{=2} \cup B$. In this new graph, each vertex in $R_{=2}$ connects exactly two vertices in B, and likewise, every path of length two between $u, v \in B$ goes through a vertex $w \in R_{=2}$. From G', one can construct a multi-graph G_B on B where each path of length two between any $u, v \in B$ maps to a corresponding edge between u and v. Planarity of G_B follows immediately from the planarity of G'. Now crucially, note that any two vertices u, v in G_B can have at most two edges: otherwise three edges between u, v in G_B imply three degree-2 vertices of $R_{=2}$ connected to both u, v in G', a contradiction to the locally expansion condition. Therefore, $|R_{=2}| \leq 6|B|$, and we get $|R| \leq 8|B|$.

Tightness example. The tightness example follows from the observation that in a $k \times k$ grid, the number of squares (k^2) is equal, up-to lower order terms, to the number of vertices ($k^2 + 2k + 1$) of the grid. So the plan is to view the set of blue vertices as vertices of a grid, and then add 8 red vertices within each square of this grid and connect them up to the four blue boundary vertices of that square such that the resulting graph is 3-locally expanding. Then the grid formed by tiling these squares will have 8 times as many red vertices as the number of squares, which in turn is asymptotically equal to the number of blue vertices.





The figure on the right shows four blue vertices of a square connected to eight red vertices. This can then be tiled into a $k \times k$ grid to form the graph $G = (R, B, E)$. In this graph vertices in R have degree two or degree three. G is 3-locally expanding: clearly any subset of R containing a degree three vertex is locally expanding; on the other hand every pair of blue vertices have at most two degree-2 red vertices in common. Finally, the total number of red vertices is exactly $8k^2$ while the number of blue vertices is exactly $k^2 + k + 1$. So $|R| \geq 8|B| - o(|B|)$.

General k.

Let $n = |R| + |B|$, and apply Theorem 1.6 from section 1.2 with $r = k$. This implies that

$$R \cup B = X \uplus V_1 \uplus \cdots \uplus V_{\Theta(n/k)}, \text{ where } |X| = O(n/\sqrt{k}), \text{ and } |V_i| \leq k$$

Furthermore, $N(V_i) \subseteq V_i \cup X$, and $|N(V_i) \cap X| \leq \sqrt{r}$.

Let $R_i = V_i \cap R$ and $B_i = V_i \cap B$. As $|R_i| \leq k$, by locally expansion property, we can bound $|R_i|$ in terms of $|B_i|$, and then by summing it over all sets, we can bound the size of R in terms of B:

$$\begin{aligned}
|R| &= \sum_i |R_i| + |X| \leq \sum_i \Big(|B_i| + |N(R_i) \cap X|\Big) + O(n/\sqrt{r}) \\
&\leq \sum_i \Big(|B_i| + \sqrt{r}\Big) + O(n/\sqrt{r}) = \sum_i |B_i| + O(n/\sqrt{r}) \\
&\leq |B| + c \cdot \frac{|R| + |B|}{\sqrt{r}} \\
&\leq \Big(\frac{1 + c/\sqrt{r}}{1 - c/\sqrt{r}}\Big)|B| = \Big((1 + c/\sqrt{r})(1 + c/\sqrt{r} + (c/\sqrt{r})^2 + \dots\Big) \\
&\leq (1 + c/\sqrt{r})(1 + 2c/\sqrt{r})|B| \leq (1 + 4c/\sqrt{r})|B|
\end{aligned}$$

[1] Norbert Bus, Shashwat Garg, Nabil H. Mustafa, and Saurabh Ray. Limits of local search: Quality and efficiency. *Discrete & Computational Geometry*, 57(3):607–624, 2017.

[2] Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48(2):373–392, 2012.

[3] Bruno Jartoux and Nabil H. Mustafa. Optimality of geometric local search. In *34th International Symposium on Computational Geometry, SoCG 2018*, 2018.

[4] Nabil H. Mustafa and Saurabh Ray. Improved results on geometric hitting set problems. *Discrete & Computational Geometry*, 44(4):883–895, 2010.

QUESTIONS

1. Extend Theorem 1.8 to minor-free subgraphs using Theorem 1.7.
2. Let $G = (R, B, E)$ be a 4-locally expanding graph.
   (a) Prove that $|R| \leq 5|B|$.
   (b) By tiling a square with 5 vertices, show that this bound is tight.
3. We re-prove the local expansion theorem (with worse constants) for constant values of expansion in a purely geometric setting. Let $\mathcal{R}$ be a set of red disks, and $\mathcal{B}$ a set of blue disks in the plane such that $\mathcal{R} \cup \mathcal{B}$ is pairwise interior-disjoint. Assume the intersection graph induced by $\mathcal{R} \cup \mathcal{B}$ is 10-locally expanding for $\mathcal{R}$.
   (a) Let $\mathcal{R}_1$ be the set of red disks with a blue neighbor of smaller radius. Show that $|\mathcal{R}_1| \leq 6|\mathcal{B}|$.
   (b) Consider the smallest-radius blue circle in $\mathcal{B}$, say $B \in \mathcal{B}$, and let $\mathcal{R}' \subseteq \mathcal{R} \setminus \mathcal{R}_1$ be the disks adjacent to B. Then show that $|\mathcal{R}'| \leq 9$.
   (c) Conclude that $|\mathcal{R}| = O(|B|)$.
4. Prove that for a graph $G = (R, B, E)$ that is 2-locally planar bipartite graph and can be embedded without intersections in $\mathbb{R}$, we have $|R| \leq |B|$.
5. Let $G = (R, B, E)$ be a k-locally planar bipartite graph. Then prove that G has a matching of size at least $(1 - c/\sqrt{k}) \cdot |R|$.

Lectures 2–4: Local Search Algorithms—Combinatorial, Metric and Euclidean

2.1 Combinatorial

The material in this section is taken from [1, 3, 2].

⋆ ⋆ ⋆

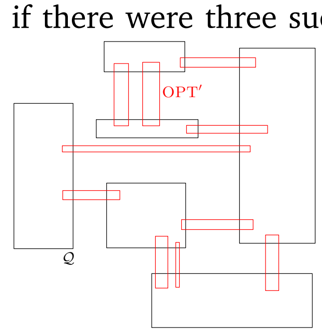
Consider the following algorithm for computing an independent set in the intersection graph of non-piercing rectangles. Let $\mathcal{R}$ be a set of n non-piercing rectangles. The algorithm starts with any independent set $\mathcal{Q}$, and at each step, considers the following local modification as long as the resulting set is still independent: i) add a new rectangle from $\mathcal{R} \setminus \mathcal{Q}$ to $\mathcal{Q}$, or ii) replace one rectangle of $\mathcal{Q}$ with two rectangles of $\mathcal{R} \setminus \mathcal{Q}$, or iii) replace two rectangles of $\mathcal{Q}$ with three rectangles of $\mathcal{R} \setminus \mathcal{Q}$. Let OPT be a maximum independent set in $\mathcal{R}$. Now follows the main claim:

```
Let Q = ∅
repeat
    foreach (R' ⊆ R, Q' ⊆ Q), |Q'| < |R'| ≤ 3 do
        if (Q \ Q') ∪ R' is independent then
            set Q = (Q \ Q') ∪ R'
until size of Q remains unchanged
return Q
```

Claim 2.1. $\mathcal{Q}$ is an 11-approximation; i.e., $|\mathcal{Q}| \geq |\mathrm{OPT}|/11$.

First note that each rectangle $R \in$ OPT must intersect some rectangle of $\mathcal{Q}$; otherwise the algorithm would have improved with $\mathcal{Q} \cup R$. Second, remove from OPT the set OPT_1 of rectangles that intersect exactly one rectangle of $\mathcal{Q}$. No two rectangles $R_1, R_2 \in \mathrm{OPT}_1$ can intersect the same rectangle $Q_1 \in \mathcal{Q}$; otherwise the algorithm would have improved with $(\mathcal{Q} \setminus \{Q_1\}) \cup \{R_1, R_2\}$. Therefore $|\mathrm{OPT}_1| \leq |\mathcal{Q}|$. Further remove the set of rectangles OPT_2, of size at most $4 \cdot |\mathcal{Q}|$, that contain some corner of a rectangle of $\mathcal{Q}$. Let $\mathrm{OPT}' = \mathrm{OPT} \setminus (\mathrm{OPT}_1 \cup \mathrm{OPT}_2)$ be the remaining rectangles.

Now observe two properties: i) each rectangle $R \in \mathrm{OPT}'$ intersects exactly two rectangles of $\mathcal{Q}$; any rectangle intersecting at least three rectangles of an independent set must contain a corner and, ii) between any two rectangles $Q_1, Q_2 \in \mathcal{Q}$, there can be at most two rectangles of OPT' intersecting both Q_1 and Q_2; otherwise if there were three such rectangles of OPT', the algorithm could locally improve $\mathcal{Q}$ by replacing Q_1, Q_2 with those three rectangles. The figure illustrates an example of $\mathcal{Q}$ and OPT'. It should remind the reader of planar graphs. Indeed one can construct a planar graph $G = (V, E)$ by mapping the rectangles of $\mathcal{Q}$ to vertices V while OPT' can be mapped to edges E. By the planar graph



theorem, $|\mathrm{OPT}'| \leq 6 \cdot |\mathcal{Q}|$, and so

$$|\mathrm{OPT}| \leq |\mathrm{OPT}_1| + |\mathrm{OPT}_2| + |\mathrm{OPT}'| \leq 11|\mathcal{Q}|$$

This should remind the reader of Theorem 1.8 in Section 1.3. Indeed, we are simply redoing that proof for $k = 3$! This should make it clear that this idea works in a more general setting than rectangles for any k-local search for a broader category of problems. So lets now formally state a more general condition needed for local search to work.

Given a set system $(X, \mathcal{R})$, where X is a base set of n elements, let $\Pi : 2^X \longrightarrow \mathbb{R}$ be an objective function that associates a 'value' to each subset of X (this value depends on the system $\mathcal{R}$). The goal then is to find a subset of X with minimum or maximum value. Consider a k-local search algorithm to compute a set optimizing Π: start with an initial solution $\mathcal{Q}$, and iteratively try to improve it by replacing a subset $\mathcal{Q}'$ of at most k points of $\mathcal{Q}$ with a set $\mathcal{R}'$ of at most k points of $X \setminus \mathcal{Q}$. If k is a constant, this search can be performed in time $n^{k^{O(1)}}$, assuming one can check if $(\mathcal{Q} \setminus \mathcal{Q}') \cup \mathcal{R}'$ improves $\mathcal{Q}$ in polynomial time. Call the output of this algorithm a k-locally optimal solution.

> Let $\mathcal{Q} \subseteq X$ be any feasible solution
> **repeat**
>   **foreach** $(\mathcal{R}' \subseteq X, \mathcal{Q}' \subseteq \mathcal{Q})$ *where* $|\mathcal{Q}'|, |\mathcal{R}'| \leq k$ **do**
>   **if** $\Pi((\mathcal{Q} \setminus \mathcal{Q}') \cup \mathcal{R}')$ *improves* $\Pi(\mathcal{Q})$ **then**
>   set $\mathcal{Q} = (\mathcal{Q} \setminus \mathcal{Q}') \cup \mathcal{R}'$
> **until** *value of* $\mathcal{Q}$ *remains unchanged*
> **return** $\mathcal{Q}$

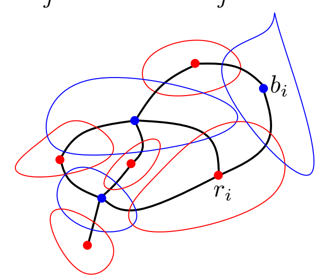
Let OPT be the globally optimal solution, and $\mathcal{Q}$ a k-locally optimal solution. Assume further that one can construct a planar bipartite graph on $\mathcal{Q} \cup \mathrm{OPT}$ that is k-locally expanding for $\mathcal{Q}$ or OPT. Then applying Theorem 1.8 on this graph yields that the sizes of the two sets $\mathcal{Q}$ and OPT are within $(1 \pm O(1/\sqrt{k}))$ factor of each other. Setting $k = O(1/\epsilon^2)$ implies that $\mathcal{Q}$ is a $(1+\epsilon)$-approximation, and the running time of the algorithm is $n^{O(1/\epsilon^2)}$.

Theorem 2.2. *Given* $(X, \mathcal{R})$ *and a function* Π*, let* $\mathcal{Q}$ *be a* k*-locally optimal solution, and* OPT *the optimal solution. Then if there exists a bipartite planar graph* $(\mathcal{Q}, \mathrm{OPT}, E)$ *that is* k*-locally expanding for* $\mathcal{Q}$ *or* OPT*, there is a* PTAS *for* Π.

Application: independent sets in pseudo-disks. Let $\mathcal{D}$ be a set of n pseudo-disks in the plane, and $\mathcal{Q} = \{Q_1, \ldots\}$ be a k-locally maximum independent set, and $\mathrm{OPT} = \{O_1, \ldots\}$ a maximum independent set. Now Theorem 2.2 together with the claim below implies the existence of a PTAS for this problem.

Claim 2.3. There exists a bipartite planar graph $G = (\mathcal{Q}, \mathrm{OPT}, E)$ that is k-locally expanding for OPT.

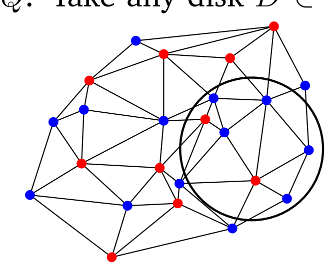
Proof. G will simply be the intersection graph of $\mathcal{Q} \cup \mathrm{OPT}$! It remains to argue that this graph is planar, and it is k-locally expanding for OPT. For each $Q_i \in \mathcal{Q}$ let $b_i \in Q_i$ be a point that is not contained in any disk of OPT. Similarly define r_j for each $O_j \in \mathrm{OPT}$. The vertex for each $Q_i \in \mathcal{Q}$ maps to b_i, vertex for each $O_j \in \mathrm{OPT}$ maps to r_j, and the edges from b_i to r_j can be drawn so that there are no intersections. It follows from the impossibility of further local improvement that this graph is k-locally expanding for OPT: take any set $\{O_1, \ldots O_t\}$, $t \leq k$, and let $\{Q_1, \ldots, Q_s\}$ be their neighbors in G. Crucially, $(\mathcal{Q} \setminus \{Q_1, \ldots, Q_s\}) \cup \{O_1, \ldots, O_t\}$ is an independent set. So $s < t$ would contradict the k-locally maximal property of $\mathcal{Q}$. $\square$



Application: hitting sets for disks. Let P be a set of n points in the plane, and let $\mathcal{D}$ be a set of m disks. Let $Q = \{q_1, \ldots\} \subseteq P$ be a k-locally minimum hitting set for the disks in $\mathcal{D}$, and $\mathrm{OPT} = \{o_1, \ldots\} \subseteq P$ be a minimum hitting set.

Claim 2.4. There exists a bipartite planar graph $G = (Q, \mathrm{OPT}, E)$ that is k-locally expanding for Q.

Proof. G is simply the Delaunay triangulation of $Q \cup \mathrm{OPT}$ (with the edges between Q and OPT removed). Delaunay triangulations are planar graphs, so it remains to prove that it is k-locally expanding for Q. So consider any subset Q' of size at most k, and let O' be the neighbors of Q' in G. Then we need to show that $(Q \setminus Q') \cup O'$ is a hitting set, and so $|O'| < |Q'|$ would contradict the k-locally minimal property of Q. Take any disk $D \in \mathcal{D}$. If it is hit by a point of Q not in Q', it is still being hit after the replacement. So assume D is only hit by a subset of Q'. Now crucially the subgraph of the Delaunay triangulation induced by the points of $D \cap (Q \cup \mathrm{OPT})$ is connected. As OPT is also a hitting set, $D \cap \mathrm{OPT} \neq \emptyset$. So there must be an edge in the Delaunay triangulation between a point $q \in Q'$ and some $o \in \mathrm{OPT} \cap D$. So $o \in O'$ and hits D. $\square$



[1] Norbert Bus, Shashwat Garg, Nabil H. Mustafa, and Saurabh Ray. Limits of local search: Quality and efficiency. *Discrete & Computational Geometry*, 57(3):607–624, 2017.

[2] Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48(2):373–392, 2012.

[3] Nabil H. Mustafa and Saurabh Ray. Improved results on geometric hitting set problems. *Discrete & Computational Geometry*, 44(4):883–895, 2010.

QUESTIONS

(solutions)

1. Let $\mathcal{R}$ be a set of n intervals in $\mathbb{R}$.
   (a) Show that a 1-locally maximum independent set is a 2-approximation to the maximum independent set, and this is tight.
   (b) Show that a 2-locally maximum independent set has size at most one less than the optimal independent set size.
2. Given a set P of n points in $\mathbb{R}^2$, let OPT be the size of the largest subset of P in general position. Show that one can compute a $\sqrt{\text{OPT}}$-approximation to OPT in polynomial time.
3. Show that $(2, 1)$ local improvement steps do not work for the hitting set algorithm for disks in the plane.

2.2 Metric

The material in this section is taken from [1, 3, 2].

We are given a set P of n points, together with a distance metric $d : P \times P \longrightarrow \mathbb{R}$ on P. Given an integer k, one would like to pick a subset $Q \subseteq P$ of k points such that the sum of distances of each point in P to its nearest point in Q is minimized.

More precisely, given P and d, define the *cost* of any $Q \subseteq P$ to be

$$\textsc{cost}(Q) = \sum_{p \in P} d(p, Q), \quad \text{where} \quad d(p, Q) = \min_{q \in Q} d(p, q)$$

Then given any integer k, the goal is to find a subset of size at most k of minimum cost.

The algorithm is simple: start with any set $S \subseteq P$ of size k and as long as the cost of the solution decreases, repeatedly replace a single point of S with a single point of $P \setminus S$. In general this will not give the optimal solution, and can get stuck in a local minima. It is easy to construct examples, even for $k = 2$, where the cost of the local-search solution is a constant factor far from the optimal solution.

Let $S \subseteq P$ be any subset of size k
repeat
  foreach $p \in P \setminus S$, $q \in S$ **do**
    if $\textsc{cost}((S \setminus \{q\}) \cup \{p\}) < \textsc{cost}(S)$ **then**
      set $S = (S \setminus \{q\}) \cup \{p\}$
until *size of* S *remains unchanged*
return S

Let OPT $= \{p_{o_1}, \dots, p_{o_k}\}$ denote the optimal solution, and $S = \{p_{s_1}, \dots, p_{s_k}\}$ the local-search solution. To show a relation between OPT and S, a first try is to iteratively transform S to OPT by swapping, in each iteration, a point of S with a point of OPT. Let S_i be the set formed by swapping the first i points of S with those of OPT, $S_i = \{p_{o_1}, \dots, p_{o_i}, p_{s_{i+1}}, \dots, p_{s_k}\}$. Now if one could construct an ordering of S and P such that

$$\textsc{cost}(S_{i+1}) \geq c \cdot \textsc{cost}(S_i)$$

it would imply that $\textsc{cost}(\text{OPT}) = \textsc{cost}(S_k) \geq c^k \textsc{cost}(S)$, and so give an upper-bound on the cost of S in terms of the cost of OPT. The problem is that after a few steps, the structure of S_i becomes arbitrary, and then S_i need not satisfy any locally optimal property. In particular, we only know the information $\textsc{cost}(S) \leq \textsc{cost}(S \setminus \{p_{s_i}\} \cup \{p_{o_j}\})$ for each $1 \leq i, j \leq k$. So one is unable to compare the costs of S_i with that of S_{i+1}.

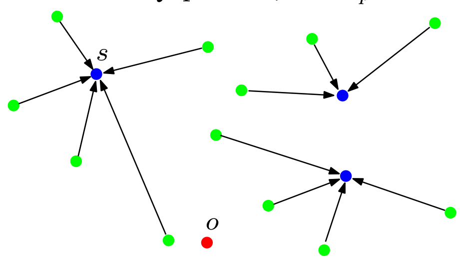
The main idea – much like the crossing lemma technique – is to do the swaps *in parallel*. So let $S_i = S \setminus \{p_{s_i}\} \cup \{p_{o_i}\}$. Now from the locally optimal property of S, we know that

$$\textsc{cost}(S_i) \geq \textsc{cost}(S) \ \text{ for all } \ i = 1 \dots k$$

On the other hand, we will charge the distances in $\textsc{cost}(S_i)$ to distances present in $\textsc{cost}(S)$ and $\textsc{cost}(\text{OPT})$ so that each such distance is charged only a constant number of times.

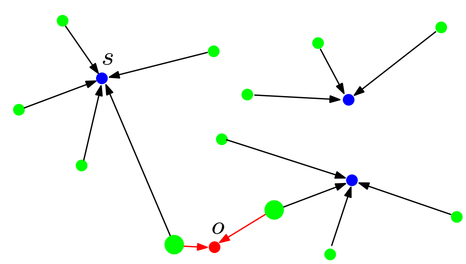
Summing up over all i, and putting these upper- and lower- bounds together will conclude the proof.

Our goal now is to upper-bound $\textsc{cost}(S_i)$. For brevity, denote $s = p_{s_i}$ and $o = p_{o_i}$, and for any $p \in P$, let o_p be the closest point in OPT to p, and s_p be the closest point in S to p. Consider $S_i = (S \setminus \{s\}) \cup \{o\}$. As s is no longer part of S_i, we will have to reassign points of P that were using s to some other point in S_i such that the total cost does not increase too much. Consider the different cases for a point $p \in P$:



1. The closest point in OPT to p is o . If p was mapping to o in the optimal solution, we will re-assign it to o in S_i (regardless of whether s was the closest point to p in S_i or not). Thus $d(p, S_i) - d(p, S) \leq d(p, o = o_p) - d(p, s_p)$.

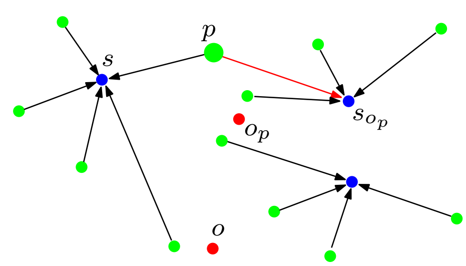
2. The closest point in S to p was not s . In this case, the closest point of p is still in S_i, and we need not change its mapping. Thus $d(p, S_i) - d(p, S) \leq 0$.



3. The closest point in S to p was s, and the closest point in OPT to p is not o . This is the key case, where p has lost its closest point $s_p = s$, and o might not be a good replacement for s. We will assign p to the closest point of o_p in S, the point s_{o_p}. Then

$$d(p, s_{o_p}) \leq d(p, o_p) + d(o_p, s_{o_p}) \leq d(p, o_p) + d(o_p, s_p) \leq d(p, o_p) + d(o_p, p) + d(p, s_p) \leq 2d(p, o_p) + d(p, s_p)$$

Thus $d(p, S_i) - d(p, S) \leq 2d(p, o_p)$.



Note that in each case, the additional cost for any p is just a function of the distance of p to either its closest point o_p in OPT or its closest point s_p in S. Thus one can now sum up the increase in the cost:

$$\begin{aligned}
\textsc{cost}(S_i) - \textsc{cost}(S) &\leq \sum_{p \text{ closest to } o} d(p, o_p) - d(p, s_p) + \sum_{\substack{p \text{ not closest to } o \\ p \text{ closest to } s}} 2d(p, o_p) \\
&\leq \sum_{p \text{ closest to } o} d(p, o_p) - d(p, s_p) + \sum_{p \text{ closest to } s} 2d(p, o_p)
\end{aligned}$$

We will now sum this up over all S_i, noting that then each p will be counted exactly once in both the first and the second term of the above summation,

$$\begin{aligned} 0 \leq \sum_i \textsc{cost}(S_i) - \textsc{cost}(S) &\leq \sum_{o \in \text{OPT}} \sum_{p \text{ closest to } o} d(p, o_p) - d(p, s_p) + \sum_{s \in S} \sum_{p \text{ closest to } s} 2d(p, o_p) \\ 0 &\leq \textsc{cost}(\text{OPT}) - \textsc{cost}(S) + 2\textsc{cost}(\text{OPT}) \end{aligned}$$

giving $\textsc{cost}(S) \leq 3\textsc{cost}(\text{OPT})$.

There is one problem in the above proof: in case 3., it could be that for some p, the nearest point to o_p is s, i.e., s_{o_p} is s. In that case we cannot re-assign p to s_{o_p}. To avoid this situation, the pairing for swaps that we create – p_{s_i} being swapped with p_{o_i} – must have the property that no point of OPT, other than p_{o_i}, can have p_{s_i} as its closest point in S. This fortunately can always be done, though at a slight cost: if there is a point $p_{s_i} \in S$ that is the closest point to exactly one point of OPT, then clearly we can pair them together. On the other hand, if a point $p_{s_i} \in S$ is the closest point to d points of OPT, then there must be $d-1$ other points of S that are the closest points to *no* point of OPT; in that case we pair up those $d-1$ points with the d points of OPT, although one of these $d-1$ points will appear in two pairings.

With this pairing, each point of OPT appears exactly once over all the k pairs, while a point of S can appear at most twice. Then the second term of the overall summation contributes $4\textsc{cost}(\text{OPT})$, and so we get a slightly weaker bound: $\textsc{cost}(S) \leq 5 \cdot \textsc{cost}(\text{OPT})$.

[1] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[2] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. Local search for max-sum diversification. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 130–142, 2017.

[3] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. In *Proceedings of the 18th Annual Symposium on Computational Geometry, Barcelona, Spain, June 5-7, 2002*, pages 10–18, 2002.

QUESTIONS

1. We will now apply the local search technique to the k-means problem, where instead one minimizes sum of squared distances. Given a set P of n points, and an integer k,

the goal is to find $Q \subseteq P$ of size k of minimum cost, where $\textsc{cost}(Q) = \sum_{p \in P} d^2(p, Q)$. Now triangle inequality does not hold for squared distances. Show that the local-search algorithm gives a set Q such that $\textsc{cost}(Q) \leq 25 \cdot \textsc{cost}(\mathsf{OPT})$.

2.3 Euclidean

The material in this section is taken from [1, 3, 2].

⋆ ⋆ ⋆

In the previous section, we considered the case where we are given a set P of n points, together with a distance metric $d : P \times P \longrightarrow \mathbb{R}$ on P. Given an integer k, one would like to pick a subset $Q \subseteq P$ of k points such that the sum of distances of each point in P to its nearest point in Q is minimized.

It turns out that if one assumes that points are in Euclidean space $\mathbb{R}^d$, then one can prove a stronger result—one can get a polynomial time approximating scheme for k-median clustering by the local search algorithm that searches for a local improvement by swapping $\frac{1}{\epsilon^{O(d)}}$ points.

This will be the main result covered in this lecture. It takes its material from the following papers.

[1] Sayan Bandyapadhyay and Kasturi R. Varadarajan. On variants of k-means clustering. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, pages 14:1–14:15, 2016.

[2] Vincent Cohen-Addad. A fast approximation scheme for low-dimensional *k*-means. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 430–440, 2018.

[3] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 353–364, 2016.

Lecture 5: Further Algorithms

3.1 Shifted Dissections

The main difficulty in showing the existence of small separators is that one needs to ensure two somewhat contradictory properties simultaneously: i) that the separator 'cuts' few objects, and ii) each of the resulting subproblems is of size at most a constant fraction of the original size. The first bounds the loss incured by dividing the problem into smaller ones at each step of the recursion, while the second ensures that the depth of the recursion is small. Separators with only one of these properties are typically much easier to construct.

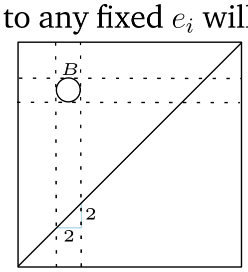
The key idea behind the Hochbaum-Maass technique is that when using separators for algorithmic purposes, it suffices to ensure weaker conditions for the separator: that it cuts few objects of $\Pi(\mathcal{O})$ (instead of $\mathcal{O}$), and that for each subproblem $\mathcal{O}_i$, $\Pi(\mathcal{O}_i)$ is at most a constant fraction of $\Pi(\mathcal{O})$, even if $|\mathcal{O}_i|$ is not much smaller that $|\mathcal{O}|$.

This is useful in cases where $\Pi(\mathcal{O}_i)$ can be computed directly, either approximately or exactly, because $\mathcal{O}_i$ satisfies certain spatial properties. Say $\Pi(\mathcal{O}') = \alpha(\mathcal{O}')$ is the size of the maximum independent set of $\mathcal{O}'$, where the given objects are unit balls in $\mathbb{R}^d$. Then if $\mathcal{O}_i$ lie in a small region, an upper-bound on $\Pi(\mathcal{O}_i)$ follows directly from packing arguments: one can only pack $O(k^d)$ disjoint unit balls in a cube T of side-length k in $\mathbb{R}^d$, and so if $\mathcal{O}_i$ is a set of unit balls lying inside T, we automatically get that $\alpha(\mathcal{O}_i) = O(k^d)$. Then one only has to worry about the separator cutting few unit balls, which by itself is a simpler problem. We illustrate this by now presenting a PTAS for the independent-set problem for a set $\mathcal{D}$ of n unit balls in $\mathbb{R}^d$.

Given an integer k, let $\mathcal{G}_k^d$ be a uniform grid of side-length k in $\mathbb{R}^d$, and for any $0 \le r < k\sqrt{d}$, define $\mathcal{G}_k^d(r)$ to be the grid obtained from $\mathcal{G}_k^d$ by translating it by r in all coordinates. As the diagonal of each cube of $\mathcal{G}_k^d$ has length $k\sqrt{d}$, this is equivalent to picking a random point q on the line $l : x_1 = \ldots = x_d$, and shifting $\mathcal{G}_k^d(r)$ by picking q as the origin.

Lemma 3.1. *Given a set of unit balls $\mathcal{B}$ in $\mathbb{R}^d$, and an integer k, there exists a uniform grid $\mathcal{G}$ of side-length k intersecting $O(d|\mathcal{B}|/k)$ balls of $\mathcal{B}$.*

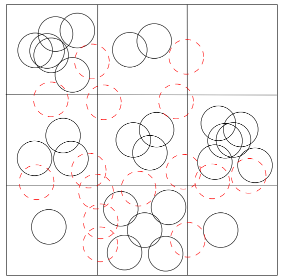
Proof. Pick a random number $r \in [0, k\sqrt{d}]$, and set $\mathcal{G} = \mathcal{G}_k^d(r)$. We show that the expected number of balls of $\mathcal{B}$ intersected by $\mathcal{G}$ is $O(d|\mathcal{B}|/k)$. For any fixed standard basis vector e_i and ball B, the probability that a hyperplane of $\mathcal{G}_k^d(r)$ perpendicular to e_i will intersect B is exactly equal to the length of l lying in the smallest strip perpendicular to e_i and containing B. For unit balls this is $2\sqrt{d}$, and so a hyperplane of $\mathcal{G}_k^d(r)$ perpendicular to any fixed e_i will intersect B with probability $2\sqrt{d}/k\sqrt{d} = 2/k$. When summed up over all d basis vectors, $\mathcal{G}_k^d(r)$ intersects B with probability at most $2d/k$. For a disk $B \in \mathcal{B}$, let $X_B = 1$ if and only if B intersects $\mathcal{G}_k^d(r)$. And so



$$\mathrm{E}[\text{Number of balls of } \mathcal{B} \text{ intersecting } \mathcal{G}_k^d(r)] = \mathrm{E}[\sum_{B \in \mathcal{B}'} X_B] \le \frac{2d|\mathcal{B}'|}{k}$$

$\square$

Given $\mathcal{D}$, apply the previous lemma with $\mathcal{B} = \text{OPT}$ and a parameter k to be fixed later to get a grid $\mathcal{G}$ of side-length k that intersects a set $\text{OPT}_s \subseteq \text{OPT}$ of $O(d|\text{OPT}|/k)$ balls. Let $\mathcal{D}_s$ be the balls of $\mathcal{D}$ intersected by $\mathcal{G}$. The remaining balls $\mathcal{D} \setminus \mathcal{D}_s$ are partitioned by $\mathcal{G}$ into the sets $\mathcal{D}_1, \dots, \mathcal{D}_t$, where $\mathcal{D}_j$ is the set of balls of $\mathcal{D}$ lying in the same cube of $\mathcal{G}$. Note the following properties:



- **Few balls removed by separator:** $\alpha(\mathcal{D} \setminus \mathcal{D}_s) \geq |\text{OPT}|(1 - d/k)$
- **Balls in different $\mathcal{D}_j$'s don't intersect:** $\alpha(\mathcal{D} \setminus \mathcal{D}_s) = \sum_j \alpha(\mathcal{D}_j)$
- **Packing argument:** $\alpha(\mathcal{D}_j) = O(k^d)$ for all $j = 1 \dots t$.

Altogether then $\alpha(\mathcal{D})$ can be approximated by throwing away $\mathcal{D}_s$ and returning $\alpha(\mathcal{D} \setminus \mathcal{D}_s)$. This can be computed by computing $\alpha(\mathcal{D}_j)$ independently, and returning the union of the t solutions. Each $\alpha(\mathcal{D}_j)$ can be computed by enumeration, to get the total time:

$$\sum_{j=1}^{t} |\mathcal{D}_j|^{O(k^d)} = n^{O(k^d)}$$

Setting $k = O(d/\epsilon)$ we get a PTAS in time $n^{O((d/\epsilon)^d)}$.

For the algorithm, a subtlety is that we have to construct $\mathcal{G}_k^d(r)$ without knowing OPT. Fortunately that is not a problem, as $\mathcal{G}_k^d(r)$ is simply a randomly shifted grid, which intersects few balls of OPT in expectation. This can be derandomized as outlined in the exercises.

The same idea works for the set-cover problem for unit balls in $\mathbb{R}^d$: given a set $P \subset \mathbb{R}^d$ of points, compute the smallest cardinality set of unit balls that cover P, its size denoted by $\gamma(P)$. Note that one can pick any set of balls in $\mathbb{R}^d$, and so there always exists a ball cover of size $|P|$. Let OPT be a minimum-sized set of unit balls covering P. Apply Lemma 3.1 with $\mathcal{B} = \text{OPT}$ to get a grid $\mathcal{G}$ of side-length k that intersects a set OPT_s of $O(d|\text{OPT}|/k)$ balls of OPT. Let $P_1, \dots, P_t$ be the disjoint subsets of P in the distinct cubes of $\mathcal{G}$. Crucially, the smallest subset of balls covering each P_i can be computed exactly in time $|P_i|^{O(k^d)}$: the cube of side-length k can be covered by $O(k^d)$ unit balls, so one has to enumerate all subsets of size $O(k^d)$ to find the smallest subset.

Let $\text{OPT}_i^b \subseteq \text{OPT}_s$ be the balls intersecting the boundary of the cell containing P_i, and $\text{OPT}_i^a \subseteq \text{OPT}_s$ be the balls lying in the interior of the cell containing P_i. Note that $\gamma(P_i) \leq |\text{OPT}_i^b| + |\text{OPT}_i^a|$, and as each unit ball (of OPT_s) intersects at most 2^d cells of $\mathcal{G}$, we get

$$\sum_i \gamma(P_i) \leq \sum_i |\text{OPT}_i^b| + \sum_i |\text{OPT}_i^a| \leq 2^d \cdot |\text{OPT}_s| + |\text{OPT}| \leq |\text{OPT}| \cdot (1 + \frac{2^d}{k})$$

Setting $k = 2^d/\epsilon$ gives a PTAS running in time $n^{O(2^{d^2}/\epsilon)}$.

These two applications of shifted dissection are very similar, using the same three properties: few objects of $\Pi(\mathrm{OPT})$ intersected by $\mathcal{G}$, the sub-problems in each cell can be solved independently, and each subproblem can be solved optimally by brute-force. However, there is a difference. In computing independent-sets, we threw away the few balls of OPT intersecting the separator $\mathcal{G}$; for the set-cover problem, the few balls of OPT were used to upper-bound the overlap across the different subproblems which were solved independently.

QUESTIONS

1. Let $\mathcal{R}$ be a set of n unit height rectangles in the plane.
   (a) Let $\mathcal{R}' \subseteq \mathcal{R}$ lie in a horizontal strip of width k. Show that one can compute $\alpha(\mathcal{R})$ exactly in time $n^{O(k)}$.
   (b) Use shifted dissection to show that there exists a PTAS for computing $\alpha(\mathcal{R})$.
2. Consider the $k/2$ grids $\mathcal{G}_k^d(2i\sqrt{d})$, for $i = 0, \ldots, k/2$. For any fixed set of balls $\mathcal{D}'$, show that there exists an i such that $\mathcal{G}_k^d(2i\sqrt{d})$ intersects at most $2d|\mathcal{D}'|/k$ balls of $\mathcal{D}'$.

3.2 Separator-based Algorithms

In this section we present algorithms for the more general problem of computing independent sets in intersection graphs of arbitrary-radius balls. As a warmup, consider a special case of this problem in $\mathbb{R}^2$ when the balls are interior-disjoint, though may intersect at the boundary. By the Koebe-Andreev-Thurston circle packing theorem, these graphs correspond exactly to planar graphs; so the problem asks for computing independent sets in planar graphs. We first address this problem; surprisingly, as we will see, the ideas for this special case can be generalized in a natural way to solve the more general problem for arbitrary-radius balls in $\mathbb{R}^d$.

A constant-factor approximation algorithm for computing independent sets in planar graphs follows immediately from the fact that there always exists a vertex of degree at most 5 in any planar graph. Given a planar graph $G = (V, E)$, add a vertex $v \in V$ of degree at most 5 in G to I, and recursively compute the independent set in the planar graph induced by $V \setminus N_G(v)$. At each step we add one vertex to I, and remove at most 5 vertices from V; therefore I has size at least $n/6$, and so is a 6-approximation to the maximum independent set in G.

This recursive algorithm can be viewed as partitioning the problem, by throwing away the set $N_G(v)$, into a constant-sized subset $\{v\}$ and a $\Omega(n)$-sized subset $V \setminus N_G(v)$. So the next natural step to improve this algorithm is recurse on two evenly balanced subsets using the planar graph separator theorem of Chapter **??**. Given $G = (V, E)$, let S be a separator of size $O(\sqrt{n})$ and A, B the two partitions of $V \setminus S$. Throw away S, recursively compute an independent set $IS(A)$ in A and an independent set $IS(B)$ in B. Now return $IS(A) \cup IS(B)$, which is an independent set as there are no edges between vertices of A and vertices of B. Given an input parameter $\epsilon > 0$, recurse until the problem size becomes only a function of ϵ (say a parameter t to be fixed later), at which point a brute-force computation can be done to compute the optimal independent set.

Algorithm: IS(Planar graph $G = (V, E)$)
if $|V| \leq t = O(1/\epsilon^2)$ **then**
  return *optimal independent set by brute-force computation.*
Let $S \subseteq V$ be a separator of size $O(\sqrt{|V|})$
Let A, B be the two partitions of $V \setminus S$
return *IS(A)* $\cup$ *IS(B)*

The key point is that in a planar graph on n vertices, the maximum independent set has size $\Omega(n)$, and so throwing away $O(\sqrt{n})$ vertices is not, relatively speaking, a big loss. The following recurrence for the independent set problem counts the total size of vertices thrown away over all recursive calls:

$$E(n) = \begin{cases} 0, & \text{if } n < t. \\ O(\sqrt{n}) + E(\alpha n) + E((1-\alpha)n) & \text{otherwise.} \end{cases}$$

where $1/3 \leq \alpha \leq 2/3$.

One can solve this by induction, but a more instructive way is to unfold and compute the resulting summation directly. Consider its recursion tree and let the height of each node v be the length of the longest path from v to a descendant leaf (leaves will have height 0). As each node of height 1 has problem size at least t, and the problem size increases at least by a factor of $3/2$ from a node to its parent, the problem size for a node of height i is at least $t \cdot (3/2)^i$.

Let $n_1, \ldots, n_l$ be the problem sizes for all the l nodes of height i. Note that $\sum n_i \leq n$, $n_i \geq t(3/2)^i$ and so $l \leq n/(3/2)^i t$. Then the total number of vertices thrown away by all nodes of height i is:

$$\sum_{i=1}^{l} O(\sqrt{n_i}) \leq \sum_{i=1}^{l} O(\sqrt{n/l}) = O(\sqrt{nl}) = O\Big(\sqrt{n \cdot \frac{n}{(3/2)^i t}}\Big) = O(n/\sqrt{t}) \cdot (2/3)^{i/2}$$

This becomes a geometric series when summed over all heights i, and we get $E(n) = O(n/\sqrt{t})$. So we have thrown away $O(n/\sqrt{t})$ vertices in total, and each remaining problem has size less than t. Setting $t = \Theta(1/\epsilon^2)$, we throw away $\epsilon n/4$ vertices, and each remaining subproblem has size $O(1/\epsilon^2)$.

Remark 1: Note that the exact function bounding the separator size – $O(\sqrt{n})$ in this case – is not crucial. The recurrence gives a PTAS as long as the separator size is sub-linear in n, so any function n^C would work for a fixed constant $C < 1$.

Remark 2: The reader may have noticed that a very similar form of recursion, using the planar graph separator theorem, has been seen before: in the proof of Theorem 1.6. In fact, we need not have re-done this proof – a PTAS for the independent set problem on planar graphs follows directly from Theorem 1.6 by setting $r = O(1/\epsilon^2)$: then $G = (V, E)$ is partitioned into $t = n/r = \epsilon^2 n$ subsets $V_1, \ldots, V_t$, where each $V_i = V_i^a \cup V_i^b$, $|V_i| = O(r) = O(1/\epsilon^2)$, $\sum_i |V_i^b| = O(n/\sqrt{r}) = O(\epsilon n)$, and for any distinct i, j, there no edges between vertices in V_i^a and in V_j^a. Throw away the vertices in the sets V_i^b for every i, for a total of $O(\epsilon n)$ vertices. Even if all of these belonged to a maximum independent set OPT of G, as $|\text{OPT}| \geq n/5$, we have thrown away only $O(\epsilon|\text{OPT}|)$ vertices. Now as each V_i^a has size a function of only $1/\epsilon$, a maximum independent set in V_i^a can be computed in time polynomial in n, and one can simply return the independent set $\bigcup_i IS(V_i^a)$.

We now turn our attention to the more general problem of computing independent sets in the intersection graphs of a set $\mathcal{D}$ of n arbitrary balls in $\mathbb{R}^d$. Surprisingly, the method for computing both constant-factor approximation and PTAS for the independent set problem for planar graphs apply naturally to this more general problem.

For getting a constant-factor approximation, add the smallest-radius ball $D \in \mathcal{D}$ to the independent set I, and recursively compute the independent set of $\mathcal{D} \setminus \mathcal{D}'$, where $\mathcal{D}'$ is the set of balls intersecting D. As all the balls in $\mathcal{D}'$ have larger radius than D, there can be only $O(2^d)$ balls of the optimal independent set in $\mathcal{D}'$. So this gives a $O(2^d)$-approximation algorithm.

So what is the generalization of the separator theorem for the case of intersecting balls? One cannot use the planar graph separator theorem as such, as $i)$ the balls are not disjoint; so if every pair of balls intersect, the intersection graph is a complete graph with no possible separator, and $ii)$ the optimal independent set in $\mathcal{D}$ can be much smaller than n, and so the separator size of $\Omega(\sqrt{n})$ is not comparatively small anymore. What is needed is the following generalization of the planar graph separator theorem in the geometric setting, following the idea of the shifted dissection technique:

Theorem 3.2. *Given a set $\mathcal{B}$ of k disjoint balls in $\mathbb{R}^d$, there exists a separator cube intersecting $O(k^{1-1/d})$ balls of $\mathcal{B}$.*

To prove this statement, one could follow exactly the proof of Theorem 1.1, and show the existence of a cube that intersects $O(k^{1-1/d})$ balls of $\mathcal{B}$. Indeed that can be done; however for the algorithm, there is a slight technical difficulty: we don't know OPT. So one cannot apply this theorem by giving it $\mathcal{B} = \text{OPT}$. We can, of course, compute a constant-factor approximation of OPT from our earlier discussion. But still, given a set of candidate cubes where one of them intersects $O(|\text{OPT}|^{1-1/d})$ balls of OPT, how do we recognize it? Let $\mathcal{D}_T$ be the set of balls intersecting a cube T. Now if $\alpha(\mathcal{D}_T) = O(|\text{OPT}|^{1-1/d})$ for some T, then clearly that T can only intersect $O(k^{1-1/d})$ balls of OPT. However the other direction is not true, as $\alpha(\mathcal{D}_T)$ could be large even if T intersects few balls of OPT. Fortunately the proof of Theorem 1.1 can be extended to this stronger statement:

Theorem 3.3. *Let $\mathcal{D}$ be a set of n balls in $\mathbb{R}^d$, and let* OPT *be a maximum independent set in the intersection graph of $\mathcal{D}$, where $k = |\text{OPT}|$. Then there exists a separator cube T such that $\alpha(\mathcal{D}_T) = O(k^{1-1/d})$. Furthermore, given $\mathcal{D}$ and the integer k, T can be computed in polynomial time.*

Proof. Let T_0 be the smallest cube in $\mathbb{R}^d$ containing at least $k/(2^d+1)$ centers of the balls in $\mathbb{R}^d$. Assume that $r(T_0) = 1$. Then as each cube in the family

$$\mathcal{F} = \{T \;\; s.t. \;\; 1 \le r(T) \le 2, \;\; c(T) = c(T_0)\}$$

can be covered with 2^d copies of T_0, it contains at least $k/(2^d+1)$ centers of balls in OPT on both sides. Thus it suffices to show the existence of a square $T \in \mathcal{F}$ such that $\alpha(\mathcal{D}_T) = O(k^{1-1/d})$. To that end, define:

$$\mathcal{D}_1 = \left\{ D \in \mathcal{D} \;\mid\; r(D) < \frac{1}{k^{1/d}} \right\}, \qquad \mathcal{D}_2 = \left\{ D \in \mathcal{D} \;\mid\; r(D) \ge \frac{1}{k^{1/d}} \right\}$$

By packing arguments, $\alpha(T \cap \mathcal{D}_2) = O((1/(1/k^{1/d}))^{d-1}) = O(k^{1-1/d})$ for any $T \in \mathcal{F}$. On the other hand, let $\mathcal{T} = \{T_1, \dots\}$ be the set of $k^{1/d}/4$ equally spaced cubes in $\mathcal{F}$. The distance between consecutive cubes in $\mathcal{T}$ is $4/k^{1/d}$, so a ball in $\mathcal{D}_1$ intersects at most one cube in $\mathcal{T}$. Furthermore, as the balls in $\mathcal{D}_1$ intersecting different cubes in $\mathcal{T}$ cannot intersect with each other, we have

$$\sum_{T \in \mathcal{T}} \alpha(\mathcal{D}_1 \cap T) \leq k$$

and so there exists $T \in \mathcal{T}$ with $\alpha(T \cap \mathcal{D}_1) \leq 4k/k^{1/d} = O(k^{1-1/d})$.

There are polynomial number of choices for T_0, and then $4k^{1/d}$ possible cubes in $\mathcal{T}$. For each cube in $\mathcal{T}$, use the constant-factor approximation algorithm to check if $\alpha(\mathcal{D}_T) = O(k^{1-1/d})$. □

Finally, the algorithm for computing an independent set in the intersection graph of balls in $\mathbb{R}^d$ follows as before, but using Theorem 3.3 instead of the planar graph separator theorem.

QUESTIONS

1. Let S be a set of n line segments in the plane, all having their left endpoint on the y-axis. Show that one can compute a maximum independent set in the intersection graph of S exactly in polynomial time.

2. Let S be a set of n line segments in the plane, all intersecting the y-axis. Let OPT be a maximum independent set of S.

   (a) Show that there exists a set $Q \subseteq \text{OPT}$ of size $\Omega(\sqrt{\text{OPT}})$ such that when the segments in Q are ordered by increasing value of intersection with the y-axis, either the slopes are monotonically increasing or monotonically decreasing.

   (b) Give a polynomial-time algorithm to compute one such Q exactly.

3. Let $\mathcal{R}$ be a set of n intervals in $\mathbb{R}$. Show that the following greedy algorithm computes a maximum independent set I: add the interval $R \in \mathcal{R}$ with the left-most right endpoint to I, delete all the segments intersecting R, and recursively compute an independent set of the remaining segments.

4. Let $\mathcal{R}$ be a set of n rectangles in the plane. Show that the following algorithm IS-R($\mathcal{R}$) computes a $\log n$-approximation to the maximum independent set of $\mathcal{R}$: let l be a vertical line that contains at most $n/2$ rectangles of $\mathcal{R}$ completely on either side, with $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$ be the set of rectangles intersecting/left/right of l. Then return the bigger of the two independent sets $\alpha(R_1)$, and $IS - R(\mathcal{R}_2) \cup IS - R(\mathcal{R}_3)$.

3.3 Algorithms via LP rounding

In this section we show a few different techniques to design approximation algorithms for packing and covering problems using the linear-programming relaxation of these problems.

Given a set of n geometric objects $\mathcal{O} = \{O_1, \ldots, O_n\}$, let w_O be the weight of object $O \in \mathcal{O}$. Then the IP for the independent set problem has a variable x_O for each object $O \in \mathcal{O}$, and an O is picked in a maximum independent set OPT if and only if $x_O = 1$. The linear programming relaxation is:

$$\text{Maximize } \sum_{O} w_O \cdot x_O$$

subject to:

$$\sum_{O \ni p} x_O \leq 1 \quad \forall p \in \mathbb{R}^2$$

$$0 \leq x_O \leq 1 \quad \forall O \in \mathcal{O}$$

Apply the above LP to the case where the geometric objects $\mathcal{D}$ are weighted disks in the plane. Let OPT* be the value of the linear program, and OPT a maximum independent set of $\mathcal{D}$. The key idea is to consider the disks in the order of decreasing radius: for each disk D under consideration, either add it to the current independent set or discard it. Once a disk is added, it will never be removed; likewise a discarded disk is never considered again.

Let $\mathcal{D} = \{D_1, \ldots, D_n\}$ be the set of disks in order of decreasing radius. Assume that we have processed the disks $D_1, \ldots, D_{i-1}$, and let $\mathcal{Q}_{i-1}$ be the current independent set. Then if D_i intersects a disk in $\mathcal{Q}_{i-1}$, discard it and set $\mathcal{Q}_i = \mathcal{Q}_{i-1}$. Otherwise add D_i to $\mathcal{Q}_{i-1}$ with probability $x_i/20$, and proceed to D_{i+1}.

```
Let Q_0 = ∅
foreach i = 1 ... n do
    Q_i = Q_{i-1}
    if Q_{i-1} ∪ D_i is independent then
        set Q_i = Q_{i-1} ∪ {D_i} with probability x_i/20
return Q_n
```

We now analyse the expected size of the final independent set $\mathcal{Q}_n$. Assume the set of disks $\{D_1, \ldots, D_{i-1}\}$ have already been considered, and let $\mathcal{D}_i$ be their subset which intersects D_i. Then for D_i to be added, no disk in $\mathcal{D}_i$ must have been added to $\mathcal{Q}_{i-1}$. The key idea is that all disks in $\mathcal{D}_i$ $i)$ have larger radius than D_i, and $ii)$ intersect D_i. By Section **??**, all such disks can be hit by a constant, 10, number of points. By the constraints of the LP, the sum of the variable values of all disks containing each such point is at most 1, and so the variable values of all disks in $\mathcal{D}_i$ is at most 10. Straightforward calculation now shows that each D is added to the independent set with probability linear in x_D:

Claim 3.4.

$$\Pr[D_i \in \mathcal{Q}_n] \geq \frac{x_{D_i}}{40}$$

Proof. First we compute the probability that none of the disks in $\mathcal{D}_i$ were picked. $\mathcal{D}_i$ can be hit by 10 points, and so $\sum_{D \in \mathcal{D}_i} x_D \leq 10$. Therefore

$$\Pr[\text{None of the disks in } \mathcal{D}_i \text{ were picked}] \geq \prod_{D \in \mathcal{D}_i} (1 - \frac{x_D}{20}) \geq 1 - \sum_{D \in \mathcal{D}_i} \frac{x_D}{20} \geq 1/2$$

Finally,

$$\begin{aligned} \Pr[D_i \text{ added to } \mathcal{Q}_{i-1}] &= \Pr[\text{None of the disks in } \mathcal{D}_i \text{ were picked}] \cdot \frac{x_{D_i}}{20} \\ &\geq \frac{1}{2} \cdot \frac{x_{D_i}}{20} \geq \frac{x_{D_i}}{40} \end{aligned}$$

□

The total expected size of the final independent set $\mathcal{Q}_n$ is:

$$\mathrm{E}[|\mathcal{Q}_n|] = \sum_{D \in \mathcal{D}} \Pr[D \in \mathcal{Q}_n] \geq \sum_{D \in \mathcal{D}} \frac{x_D}{40} = \frac{\text{OPT}^*}{40}$$

Therefore $|\text{OPT}| \leq \text{OPT}^* \leq 40\,\mathrm{E}[|\mathcal{Q}_n|]$, and the expected size of the independent set produced by the rounding algorithm is at least $|\text{OPT}|/40$.

Let $\mathcal{I} = \{I_1, \ldots, I_n\}$ be a set of n intervals in $\mathbb{R}$, and $P \subseteq \mathbb{R}$ a set of points. Consider the linear program for the hitting set problem, where each point $p \in P$ has a variable x_p:

$$\text{Minimize } \sum_i w_p \cdot x_p \qquad (3.1)$$

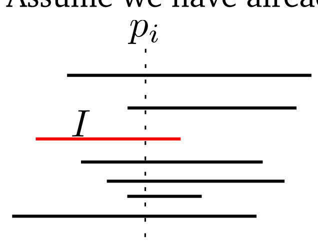
subject to:

$$\sum_{p \in I} x_p \geq 1 \quad \forall I \in \mathcal{I}$$

$$0 \leq x_p \leq 1 \quad \forall p \in P$$

Claim 3.5. The integrality gap of (3.1) is 1.

Proof. Let x be a solution to the LP. We now outline a procedure to iteratively change values of x to integral values without any increase in $\sum_p x_p$, or violating any of the LP constraints. Let $p_1, \ldots, p_n$ be the points sorted from left to the right. Assume we have already processed $p_1, \ldots, p_{i-1}$. Consider p_i, and let $\mathcal{I}_i$ be the set of intervals containing p_i. If $x_{p_i} = 1$, then remove all intervals of $\mathcal{I}_i$ from $\mathcal{I}$ and proceed to the next point. Otherwise let $I \in \mathcal{I}_i$ be the interval with the left-most right endpoint. Then as $\sum_{p \in I} x_p \geq 1$, I must contain some point p_j, $j > i$. Set $x_{p_i} = 0$ and $x_{p_j} = \min\{x_{p_j} + x_{p_i}, 1\}$. Clearly there is no increase in the overall sum, and that crucially, by the choice of I, each interval in $\mathcal{I}_i$ contains p_j, and so the constraints of the LP are preserved for all of them. $\square$



Let $\mathcal{R} = \{R_1, \ldots, R_n\}$ be a set of n axis-parallel rectangles in the plane, and $L = \{l_1, \ldots, l_m\}$ a set of m vertical and horizontal lines. Let x_l be the variable for each line $l \in L$, and OPT be the value of the optimal integral solution, and $\text{OPT}^* = \sum_l x_l$ the value of the fractional solution. Let $L_V \subseteq L$ be the set of vertical lines, and $L_H \subseteq L$ the set of horizontal lines.

The algorithm is simple: partition $\mathcal{R}$ into two sets $\mathcal{R}_H$ and $\mathcal{R}_V$ as

$$\mathcal{R}_H = \{R \in \mathcal{R} \text{ s.t. } \sum_{\substack{l \in L_H \\ l \text{ intersects } R}} x_l \geq 1/2\} \qquad \mathcal{R}_V = \{R \in \mathcal{R} \text{ s.t. } \sum_{\substack{l \in L_V \\ l \text{ intersects } R}} x_l > 1/2\}$$

Note that $\mathcal{R} = \mathcal{R}_H \cup \mathcal{R}_V$: the total value of the lines of L intersecting each rectangle $R \in \mathcal{R}$ is at least 1, and so either the total value of the horizontal lines of L_H or the vertical lines of L_V intersecting R is at least $1/2$. Now solve the hitting set problem for $\mathcal{R}$ by solving the one-dimensional problem for the rectangles of $\mathcal{R}_H$ and $\mathcal{R}_V$ separately, and taking the union of the resulting horizontal and vertical lines.

Intuitively this should work; but how to bound the size of the resulting hitting set in terms of the size of the optimal hitting set? Let $\text{OPT}_V, \text{OPT}_H$ be the optimal hitting sets for $\mathcal{R}_V, \mathcal{R}_H$ and let $\text{OPT}^*_V, \text{OPT}^*_H$ be the value of the corresponding linear programs with variables $\mathbf{x}_V, \mathbf{x}_H$. Note that setting $\mathbf{x}_V, \mathbf{x}_H$ variables to be twice the corresponding variables in $\mathbf{x}$ gives a feasible solution for the LPs of both $\mathcal{R}_V$ and $\mathcal{R}_H$. So we know the following

$$\begin{aligned} \text{OPT}^* &\leq |\text{OPT}| \\ \text{OPT}^*_V + \text{OPT}^*_H &\leq |\text{OPT}_V| + |\text{OPT}_H| \\ \text{OPT}^*_V + \text{OPT}^*_H &\leq 2 \cdot \text{OPT}^* \end{aligned}$$

This by itself does not suffice to bound $|\text{OPT}_V| + |\text{OPT}_H|$; the inequality is in the 'wrong direction'. Fortunately, from Claim 3.5 we know that $\text{OPT}^*_V = |\text{OPT}_V|$ and $\text{OPT}^*_H = |\text{OPT}_H|$, and we can conclude that

$$|\text{OPT}_V| + |\text{OPT}_H| = \text{OPT}^*_V + \text{OPT}^*_H \leq 2 \cdot \text{OPT}^* \leq 2 \cdot |\text{OPT}|$$

QUESTIONS

(solutions)

1. Extend the proof of separators to show an integrality gap of $O(1)$ for the independent-set LP for balls in $\mathbb{R}^d$.
2. Let P be a set of n points in the plane, and $\mathcal{R}$ a set system on P. A set $Q \subseteq P$ is an independent set with respect to $\mathcal{R}$ if each set in $\mathcal{R}$ contains at most one point of Q. Let OPT be the size of a maximum independent set with respect to $\mathcal{R}$, and $\text{OPT}^\star$ be the value of the LP for this problem.
   (a) Show that there exists a set $Q \subseteq P$ of size $\Omega(\text{OPT}^\star)$ such that $|R \cap Q| = O(\log |R|)$ for all $R \in \mathcal{R}$.
   (b) For any integer k, let $G_k(P, \mathcal{R})$ be the graph on P where $\{p, q\}$ is an edge iff there exists a set $R \in \mathcal{R}$ of size at most k and containing both p and q. Assume that there exists an independent set in G_k of size at least $f(n, k)$, for any k. Show that one can construct an independent set $Q' \subseteq P$ with respect to $\mathcal{R}$ of size $f(\text{OPT}^\star, \log |R|)$.

3.4 Discrete Independent Set for Shallow cell complexity

In this section we generalize the independent set algorithm for disks of the previous section to a more general setting for the *discrete* independent set problem.

Recall the setting: given a set of n disks $\mathcal{D} = \{D_1, \ldots, D_n\}$, with w_i the weight of $D_i \in \mathcal{D}$, the IP for the independent set problem has a variable x_i for each object $D_i \in \mathcal{D}$, and a disk D_i is picked in a maximum independent set OPT if and only if $x_i = 1$. For the discrete version, we are also given a set P of m points in the plane, and the goal is to find a maximum weight subset of $\mathcal{O}$ such that each point of P is covered at most once. The linear programming relaxation requires only a slight change:

$$\text{Maximize } \sum_i w_i \cdot x_i$$

subject to:

$$\sum_{D_i \ni p} x_i \leq 1 \quad \forall p \in \cancel{\mathbb{R}^2} P$$

$$0 \leq x_i \leq 1 \quad \forall i \in [1, n]$$

Let OPT* be the value of the linear program, and OPT a discrete maximum independent set of $\mathcal{D}$. For the independent set problem for disks of the last Section, we sorted all the disks of $\mathcal{D}$ by decreasing radius. The key idea was that when considering D_i for inclusion in an independent set, the total weight of the set of disks in $\{D_1, \ldots, D_{i-1}\}$ intersecting D_i was a constant: each D_j, $j < i$ has larger radius than D_i, and so all the disks in $\{D_1, \ldots, D_{i-1}\}$ intersecting D_i can be hit by a constant number of points in the plane. The total weight of the disks containing any such point, by the LP, was at most 1, and so the total weight of all the disks in $\{D_1, \ldots, D_{i-1}\}$ intersecting D_i was at most a constant.

Now it is useless to consider the ordering of the disks by radius, as a small radius disk can intersect many disks of high weight, as long as all these disks cannot be hit by a constant number of points of P. However, the basic idea remains the same: we will show that there is an ordering $\{D_1, \ldots, D_n\}$ of the disks such that for any D_i, the set of disks *intersecting D_i in the discrete sense* (i.e., disks intersecting D_i in some point of P) in $\{D_1, \ldots, D_{i-1}\}$ has total weight at most some constant. Then precisely the same algorithm and analysis works.

It remains to show the existence of the ordering; this will be shown by proving that there always exists a disk $D_i \in \mathcal{D}$ such that the total weight of the disks intersecting $\mathcal{D}_i$ in $\mathcal{D}$ in some point of P is $O(1)$, and by iteratively constructing the ordering on the remaining set $\mathcal{D} \setminus \{D_i\}$. Consider the sum of products of two variables $x_i \cdot x_j$, where the indices $\{i,j\}$ iterate over all pairs of disks D_i, D_j whose common intersection contains a point of P. We will bound this term, which together with the pigeonhole principle will show the existence of the required disk.

Lemma 3.6.

$$\sum_{\substack{\{i,j\}, i \neq j \\ D_i \cap D_j \cap P \neq \emptyset}} x_i \cdot x_j = O\Big(\sum_i x_i\Big)$$

Proof. Pick each disk D_i into a random sample S with probability $x_i/2$. Consider the random variable which is the number of pairs of disks of S which had a point in their common intersection not contained in any other object of S. First, for every pair of disks of $\mathcal{D}$ and any point p in their common intersection, the total weight of the disks of $\mathcal{D}$ containing p was at most 1, and so there is a lower bound on the probability of p being on the boundary of the union of S. On the other hand, union complexity implied that there are only $O(|S|)$ such pairs. Together this implied the statement.

In our current case, we only know that each point $p \in P$ is contained in disks of total weight at most 1. Thus we will count the number of pairs of disks $\{D_i, D_j\}$ in S such that a point $p \in P$ with $p \in D_i \cap D_j$ is not contained in any other disk of S. For a pair of disks D_i, D_j containing some point of P in their common intersection, let $\mathcal{E}_{ij}$ be the event that D_i, D_j are chosen in S and there is a point $p \in P$ in their common intersection for which none of the other disks containing p are chosen in S. Then

$$\Pr\left[\mathcal{E}_{ij}\right] \geq x_i/2 \cdot x_j/2 \cdot \sum_{D_k \ni p} \left(1 - \frac{x_k}{2}\right) \geq \frac{x_i x_j}{8}$$

And therefore we have

$$\mathrm{E}\Big[\sum_{\substack{i \neq j \\ D_i \cap D_j \cap P \neq \emptyset}} \mathcal{E}_{ij}\Big] = \sum_{\substack{i \neq j \\ D_i \cap D_j \cap P \neq \emptyset}} \mathrm{E}\left[\mathcal{E}_{ij}\right] \geq \sum_{\substack{i \neq j \\ D_i \cap D_j \cap P \neq \emptyset}} \frac{x_i x_j}{8}$$

On the other hand, if the event $\mathcal{E}_{ij}$ occurs, then $D_i \cap D_j$ must not be covered by the union of the other disks of S (since otherwise $p \in D_i \cap D_j$ cannot satisfy the conditions of $\mathcal{E}_{ij}$). Thus $\mathcal{E}_{ij}$ can only occur for those pairs which contribute an intersection vertex to $\mathcal{U}(S)$, and so we have

$$\mathrm{E}\Big[\sum_{\substack{i \neq j \\ D_i \cap D_j \cap P \neq \emptyset}} \mathcal{E}_{ij}\Big] \leq 3 \cdot \mathrm{E}[|S|] - 6 = O\Big(\sum_i x_i\Big)$$

Putting the upper and lower bounds gives the desired result. $\square$

Corollary 3.7. Given $\mathcal{D}$, there exists a disk D_i such that

$$X_i = \sum_{\substack{j \neq i \\ D_i \cap D_j \cap P \neq \emptyset}} x_j = O(1)$$

Proof. If each D_i had X_i greater than a large-enough constant c, then

$$\sum_{\substack{\{i,j\}, i \neq j \\ D_i \cap D_j \cap P \neq \emptyset}} x_i \cdot x_j = \sum_i x_i \cdot D_i \geq \sum_i c \cdot x_i$$

contradicting Lemma 3.6. $\square$

Now precisely the same analysis as previous section gives the required result.