Some Geometric Applications of Anti-Chains

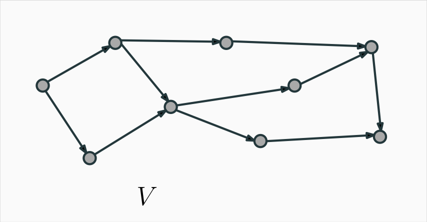
Sariel Har-Peled[1] Mitchell Jones[1]

CCCG 2020, August 5–7

[1]University of Illinois at Urbana-Champaign

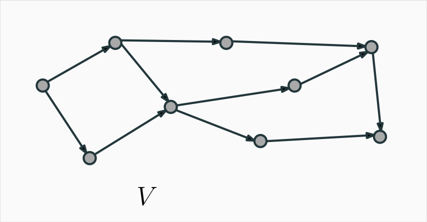
Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)



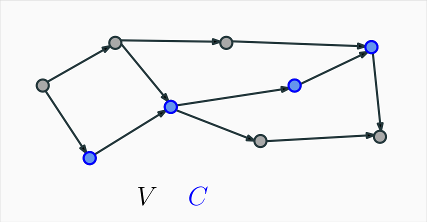
Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)
- Chain: Subset $C \subseteq V$ s.t. all elements are comparable



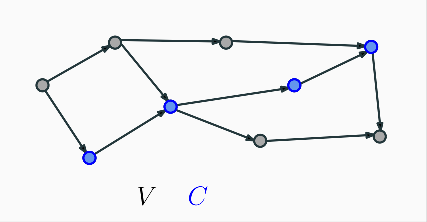
Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)
- Chain: Subset $C \subseteq V$ s.t. all elements are comparable



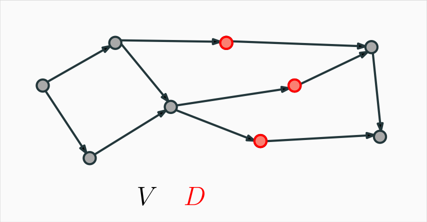
Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)
- Chain: Subset $C \subseteq V$ s.t. all elements are comparable
- Anti-chain: Subset $D \subseteq V$ s.t. all elements are incomparable



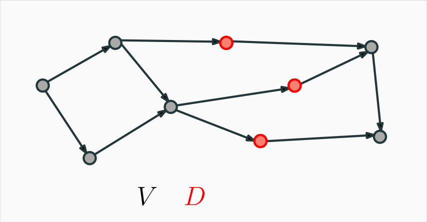
Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)
- Chain: Subset $C \subseteq V$ s.t. all elements are comparable
- Anti-chain: Subset $D \subseteq V$ s.t. all elements are incomparable



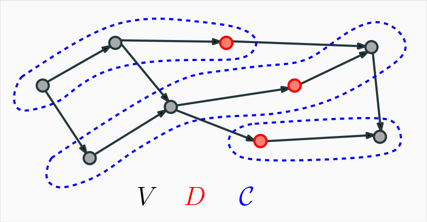
Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)
- Chain: Subset $C \subseteq V$ s.t. all elements are comparable
- Anti-chain: Subset $D \subseteq V$ s.t. all elements are incomparable
- Chain cover: collection of chains covering V



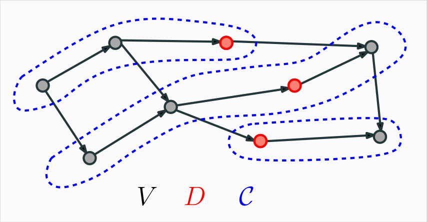
Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)
- Chain: Subset $C \subseteq V$ s.t. all elements are comparable
- Anti-chain: Subset $D \subseteq V$ s.t. all elements are incomparable
- Chain cover: collection of chains covering V



Partial orderings (posets)

- $(V, \prec)$: partially ordered set (poset)
- Chain: Subset $C \subseteq V$ s.t. all elements are comparable
- Anti-chain: Subset $D \subseteq V$ s.t. all elements are incomparable
- Chain cover: collection of chains covering V
- Largest anti-chain = smallest chain cover [Dilworth, 1950]

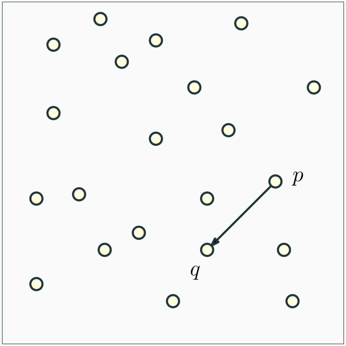


Implicit posets I

- This talk: implicitly defined posets on geometric objects

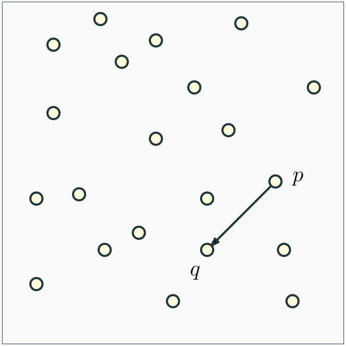
Implicit posets I

- This talk: implicitly defined posets on geometric objects
- Point set P: $(P, \prec)$, $p \prec q \iff p$ dominates q



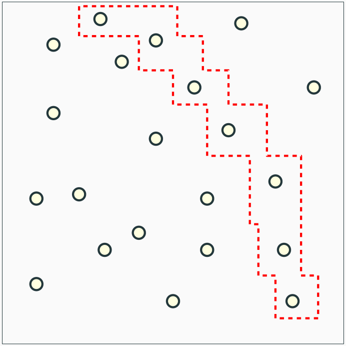
Implicit posets I

- This talk: implicitly defined posets on geometric objects
- Point set P: $(P, \prec)$, $p \prec q \iff p$ dominates q
- Anti-chain: downward "staircase" or Pareto-optimal subset



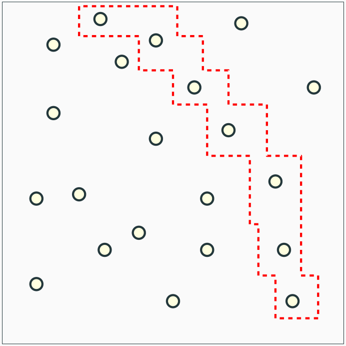
Implicit posets I

- This talk: implicitly defined posets on geometric objects
- Point set P: $(P, \prec)$, $p \prec q \iff p$ dominates q
- Anti-chain: downward "staircase" or Pareto-optimal subset



Implicit posets I

- This talk: implicitly defined posets on geometric objects
- Point set P: $(P, \prec)$, $p \prec q \iff p$ dominates q
- Anti-chain: downward "staircase" or Pareto-optimal subset
- Quickly find largest Pareto-optimal subset for $(P, \prec)$?

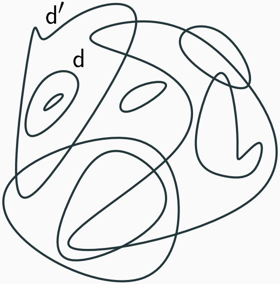


Implicit posets II

- $\mathcal{D}$: set of regions in $\mathbb{R}^d$

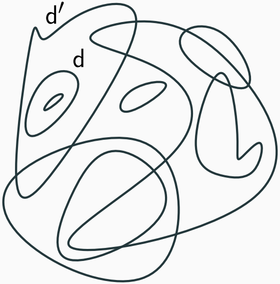
Implicit posets II

- $\mathcal{D}$: set of regions in $\mathbb{R}^d$
- $(\mathcal{D}, \prec)$, $\mathrm{d}' \prec \mathrm{d} \iff \mathrm{d} \subseteq \mathrm{d}'$



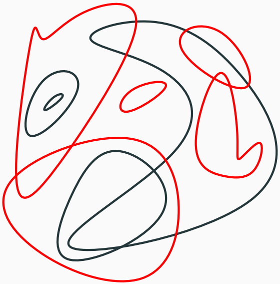
Implicit posets II

- $\mathcal{D}$: set of regions in $\mathbb{R}^d$
- $(\mathcal{D}, \prec)$, $\mathrm{d}' \prec \mathrm{d} \iff \mathrm{d} \subseteq \mathrm{d}'$
- Anti-chain: $S \subseteq \mathcal{D}$ s.t. $\forall\, \mathrm{d}_1, \mathrm{d}_2 \in S$, $\mathrm{d}_1 \not\subseteq \mathrm{d}_2$ and $\mathrm{d}_2 \not\subseteq \mathrm{d}_1$ or loose subset



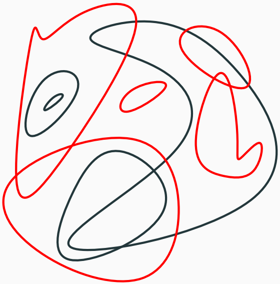
Implicit posets II

- $\mathcal{D}$: set of regions in $\mathbb{R}^d$
- $(\mathcal{D}, \prec)$, $\mathrm{d}' \prec \mathrm{d} \iff \mathrm{d} \subseteq \mathrm{d}'$
- Anti-chain: $S \subseteq \mathcal{D}$ s.t. $\forall\, \mathrm{d}_1, \mathrm{d}_2 \in S$, $\mathrm{d}_1 \not\subseteq \mathrm{d}_2$ and $\mathrm{d}_2 \not\subseteq \mathrm{d}_1$ or loose subset



Implicit posets II

- $\mathcal{D}$: set of regions in $\mathbb{R}^d$
- $(\mathcal{D}, \prec)$, $d' \prec d \iff d \subseteq d'$
- Anti-chain: $S \subseteq \mathcal{D}$ s.t. $\forall d_1, d_2 \in S$, $d_1 \not\subseteq d_2$ and $d_2 \not\subseteq d_1$ or loose subset
- Quickly find largest loose subset for $(\mathcal{D}, \prec)$?



Computing anti-chains

- Largest anti-chain can be found in $O(n^{2.5})$ time

Computing anti-chains

- Largest anti-chain can be found in $O(n^{2.5})$ time
- Reduces to max matching in bipartite graph G [Hopcroft and Karp, 1973]

Computing anti-chains

- Largest anti-chain can be found in $O(n^{2.5})$ time
- Reduces to max matching in bipartite graph G [Hopcroft and Karp, 1973]
- Implicit posets $\implies$ edges of G are implicit

Computing anti-chains

- Largest anti-chain can be found in $O(n^{2.5})$ time
- Reduces to max matching in bipartite graph G [Hopcroft and Karp, 1973]
- Implicit posets $\implies$ edges of G are implicit
- Possible speedup?

The framework

- Insight: algorithmic framework for implicit posets

The framework

- Insight: algorithmic framework for implicit posets
- $(V, \prec)$: poset of size n

The framework

- Insight: algorithmic framework for implicit posets
- $(V, \prec)$: poset of size n
- **Goal:** compute largest anti-chain for $(V, \prec)$

The framework

- Insight: algorithmic framework for implicit posets
- $(V, \prec)$: poset of size n
- **Goal:** compute largest anti-chain for $(V, \prec)$

For $P \subseteq V$, $m = |P|$, we have data structure $\mathcal{D}(P)$:

The framework

- Insight: algorithmic framework for implicit posets
- $(V, \prec)$: poset of size n
- **Goal:** compute largest anti-chain for $(V, \prec)$

For $P \subseteq V$, $m = |P|$, we have data structure $\mathscr{D}(P)$:

(i) Query $v \in V$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

The framework

- Insight: algorithmic framework for implicit posets
- $(V, \prec)$: poset of size n
- **Goal:** compute largest anti-chain for $(V, \prec)$

For $P \subseteq V$, $m = |P|$, we have data structure $\mathcal{D}(P)$:

(i) Query $v \in V$, $\mathcal{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

(ii) Delete an element from $\mathcal{D}(P)$ in $T(m)$ time

The framework

- Insight: algorithmic framework for implicit posets
- $(V, \prec)$: poset of size n
- **Goal:** compute largest anti-chain for $(V, \prec)$

For $P \subseteq V$, $m = |P|$, we have data structure $\mathscr{D}(P)$:

(i) Query $v \in V$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

(ii) Delete an element from $\mathscr{D}(P)$ in $T(m)$ time

(iii) Construct $\mathscr{D}(P)$ in $O(m \cdot T(m))$ time

The framework

- Insight: algorithmic framework for implicit posets
- $(V, \prec)$: poset of size n
- **Goal:** compute largest anti-chain for $(V, \prec)$

For $P \subseteq V$, $m = |P|$, we have data structure $\mathscr{D}(P)$:

(i) Query $v \in V$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

(ii) Delete an element from $\mathscr{D}(P)$ in $T(m)$ time

(iii) Construct $\mathscr{D}(P)$ in $O(m \cdot T(m))$ time

Our result

Can find largest anti-chain for $(V, \prec)$ in $O\left(n^{1.5} \cdot T(n)\right)$ time

The framework: remarks

Suppose for $P \subseteq V$, $m = |P|$, we have data structure $\mathscr{D}(P)$:

(i) Query $v \in V$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

(ii) Delete an element from $\mathscr{D}(P)$ in $T(m)$ time

(iii) Construct $\mathscr{D}(P)$ in $O(m \cdot T(m))$ time

Our result

Can find largest anti-chain for $(V, \prec)$ in $O\big(n^{1.5} \cdot T(n)\big)$ time

The framework: remarks

Suppose for $P \subseteq V$, $m = |P|$, we have data structure $\mathscr{D}(P)$:

(i) Query $v \in V$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

(ii) Delete an element from $\mathscr{D}(P)$ in $T(m)$ time

(iii) Construct $\mathscr{D}(P)$ in $O(m \cdot T(m))$ time

Our result

Can find largest anti-chain for $(V, \prec)$ in $O\big(n^{1.5} \cdot T(n)\big)$ time

---

- Idea: Simulate Hopcroft-Karp using $\mathscr{D}$

The framework: remarks

Suppose for $P \subseteq V$, $m = |P|$, we have data structure $\mathscr{D}(P)$:

(i) Query $v \in V$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

(ii) Delete an element from $\mathscr{D}(P)$ in $T(m)$ time

(iii) Construct $\mathscr{D}(P)$ in $O(m \cdot T(m))$ time

Our result

Can find largest anti-chain for $(V, \prec)$ in $O\big(n^{1.5} \cdot T(n)\big)$ time

---

- Idea: Simulate Hopcroft-Karp using $\mathscr{D}$
- Based on framework of Efrat et al., 2001

Suppose for $P \subseteq V$, $m = |P|$, we have data structure $\mathscr{D}(P)$:

(i) Query $v \in V$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$ in $T(m)$ time

(ii) Delete an element from $\mathscr{D}(P)$ in $T(m)$ time

(iii) Construct $\mathscr{D}(P)$ in $O(m \cdot T(m))$ time

Our result

Can find largest anti-chain for $(V, \prec)$ in $O\big(n^{1.5} \cdot T(n)\big)$ time

---

- Idea: Simulate Hopcroft-Karp using $\mathscr{D}$
- Based on framework of Efrat et al., 2001
- Recently: Similar framework for minimum cuts in disk graphs [Cabello and Mulzer, 2020]

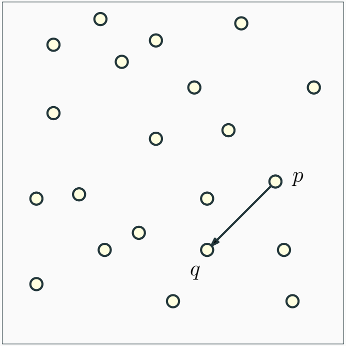
Applications

Largest Pareto-optimal subset

- P: point set

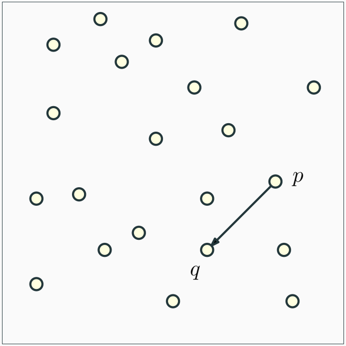
Largest Pareto-optimal subset

- P: point set
- p dominates $q \iff p \geq q$ coordinate wise



Largest Pareto-optimal subset

- P: point set
- p dominates $q \iff p \geq q$ coordinate wise
- $Q \subseteq P$ is Pareto-optimal if no point in Q dominates any other point in Q



Largest Pareto-optimal subset: the data structure

Largest Pareto-optimal subset $\equiv$ anti-chain in $(P, \prec)$

Query $v \in P$, $\mathscr{D}(P)$ returns $u \in P$ with $v \prec u$

Largest Pareto-optimal subset: the data structure

Largest Pareto-optimal subset $\equiv$ anti-chain in $(P, \prec)$

Query $v \in P$, $\mathscr{D}(P)$ returns $u \in P$ with u dominating v

Largest Pareto-optimal subset: the data structure

Largest Pareto-optimal subset $\equiv$ anti-chain in $(P, \prec)$

Query $v \in P$, $\mathscr{D}(P)$ returns $u \in P$ with u dominating v

u dominates $v \iff u \in [v_1, \infty) \times \ldots \times [v_d, \infty)$

Largest Pareto-optimal subset: the data structure

Largest Pareto-optimal subset $\equiv$ anti-chain in $(P, \prec)$

Query $v \in P$, $\mathscr{D}(P)$ returns $u \in P$ with u dominating v

u dominates $v \iff u \in [v_1, \infty) \times \ldots \times [v_d, \infty)$, d-sided orthogonal range query!

Largest Pareto-optimal subset: the data structure

Largest Pareto-optimal subset $\equiv$ anti-chain in $(P, \prec)$

Query $v \in P$, $\mathscr{D}(P)$ returns $u \in P$ with u dominating v

u dominates $v \iff u \in [v_1, \infty) \times \ldots \times [v_d, \infty)$, d-sided orthogonal range query!

Queries and deletions in time $O((\log n / \log\log n)^{d-1})$ [Chan and Tsakalidis, 2017].

Largest Pareto-optimal subset: the data structure

Largest Pareto-optimal subset $\equiv$ anti-chain in $(P, \prec)$

Query $v \in P$, $\mathcal{D}(P)$ returns $u \in P$ with u dominating v

u dominates $v \iff u \in [v_1, \infty) \times \ldots \times [v_d, \infty)$, d-sided orthogonal range query!

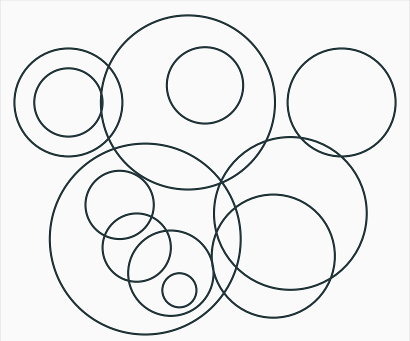
Queries and deletions in time $O((\log n / \log\log n)^{d-1})$ [Chan and Tsakalidis, 2017].

Our result

Largest Pareto-optimal subset in $\mathbb{R}^d$ in time $O(n^{1.5}(\log n / \log\log n)^{d-1})$

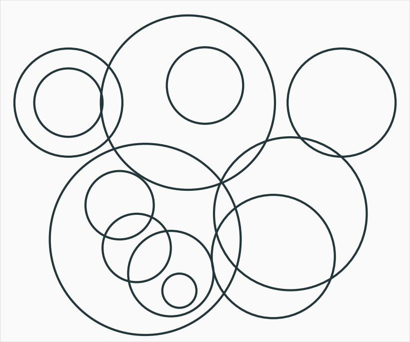
Largest loose subset of disks

- $\mathcal{D}$: set of disks in $\mathbb{R}^2$



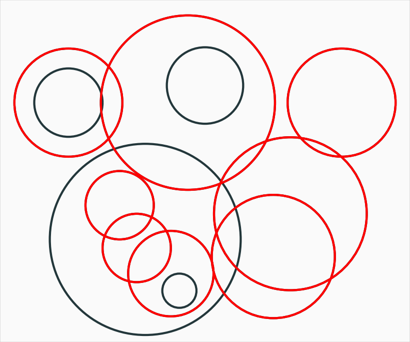
Largest loose subset of disks

- $\mathcal{D}$: set of disks in $\mathbb{R}^2$
- $(\mathcal{D}, \prec)$, $\mathrm{d}' \prec \mathrm{d} \iff \mathrm{d} \subseteq \mathrm{d}'$



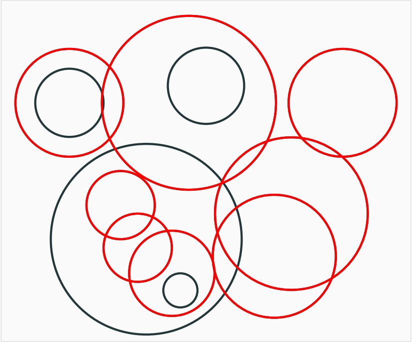
Largest loose subset of disks

- $\mathcal{D}$: set of disks in $\mathbb{R}^2$
- $(\mathcal{D}, \prec)$, $\mathrm{d}' \prec \mathrm{d} \iff \mathrm{d} \subseteq \mathrm{d}'$
- $S \subseteq \mathcal{D}$ is loose if no disk in S contains another disk in S



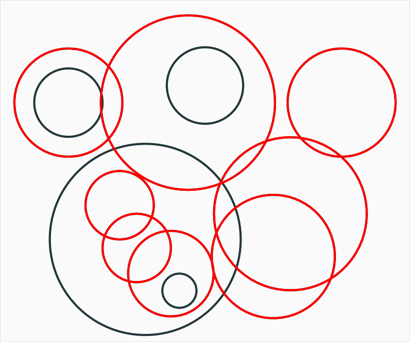
Largest loose subset of disks

- $\mathcal{D}$: set of disks in $\mathbb{R}^2$
- $(\mathcal{D}, \prec)$, $d' \prec d \iff d \subseteq d'$
- $S \subseteq \mathcal{D}$ is loose if no disk in S contains another disk in S
- Loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$



Largest loose subset of disks

- $\mathcal{D}$: set of disks in $\mathbb{R}^2$
- $(\mathcal{D}, \prec)$, $d' \prec d \iff d \subseteq d'$
- $S \subseteq \mathcal{D}$ is loose if no disk in S contains another disk in S
- Loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$
- Compare to Independent set which is NP-hard



Largest loose subset: the data structure

Largest loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$

Query $q \in \mathcal{D}$, $\mathscr{D}(\mathcal{D})$ returns $d \in \mathcal{D}$ with $q \prec d$

Largest loose subset: the data structure

Largest loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$

Query q $\in \mathcal{D}$, $\mathscr{D}(\mathcal{D})$ returns d $\in \mathcal{D}$ with d $\subseteq$ q

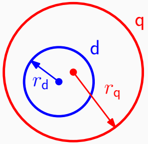
Largest loose subset: the data structure

Largest loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$

Query $\mathsf{q} \in \mathcal{D}$, $\mathscr{D}(\mathcal{D})$ returns $\mathsf{d} \in \mathcal{D}$ with $\mathsf{d} \subseteq \mathsf{q}$

For each $\mathsf{x} \in \mathcal{D}$: $\delta_{\mathsf{x}}(p) = \|c_{\mathsf{x}} - p\| + r_{\mathsf{x}}$.

$\mathsf{d} \subseteq \mathsf{q} \iff \delta_{\mathsf{d}}(c_{\mathsf{q}}) \leq r_{\mathsf{q}}$.

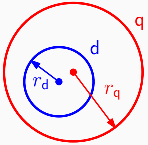


Largest loose subset: the data structure

Largest loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$

Query $\mathsf{q} \in \mathcal{D}$, $\mathscr{D}(\mathcal{D})$ returns $\mathsf{d} \in \mathcal{D}$ with $\mathsf{d} \subseteq \mathsf{q}$

For each $\mathsf{x} \in \mathcal{D}$: $\delta_{\mathsf{x}}(p) = \|c_{\mathsf{x}} - p\| + r_{\mathsf{x}}$.
$\mathsf{d} \subseteq \mathsf{q} \iff \delta_{\mathsf{d}}(c_{\mathsf{q}}) \leq r_{\mathsf{q}}$.



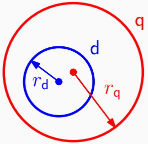
Dynamically maintain $F(p) = \min_{\mathsf{d} \in \mathcal{D}} \delta_{\mathsf{d}}(p)$

Largest loose subset: the data structure

Largest loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$

Query $\mathsf{q} \in \mathcal{D}$, $\mathscr{D}(\mathcal{D})$ returns $\mathsf{d} \in \mathcal{D}$ with $\mathsf{d} \subseteq \mathsf{q}$

For each $\mathsf{x} \in \mathcal{D}$: $\delta_{\mathsf{x}}(p) = \|c_{\mathsf{x}} - p\| + r_{\mathsf{x}}$.
$\mathsf{d} \subseteq \mathsf{q} \iff \delta_{\mathsf{d}}(c_{\mathsf{q}}) \leq r_{\mathsf{q}}$.



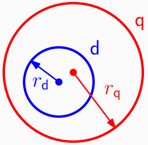
Dynamically maintain $F(p) = \min_{\mathsf{d} \in \mathcal{D}} \delta_{\mathsf{d}}(p)$ — lower envelope of surfaces $\{\delta_{\mathsf{d}} \mid \mathsf{d} \in \mathcal{D}\}$ in $\mathbb{R}^3$

Largest loose subset: the data structure

Largest loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$

Query $q \in \mathcal{D}$, $\mathscr{D}(\mathcal{D})$ returns $d \in \mathcal{D}$ with $d \subseteq q$

For each $x \in \mathcal{D}$: $\delta_x(p) = \|c_x - p\| + r_x$.
$d \subseteq q \iff \delta_d(c_q) \leq r_q$.



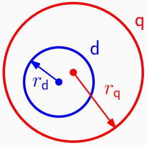
Dynamically maintain $F(p) = \min_{d \in \mathcal{D}} \delta_d(p)$ — lower envelope of surfaces $\{\delta_d \mid d \in \mathcal{D}\}$ in $\mathbb{R}^3$ — in time $O(\log^{10+\varepsilon} n)$ [Kaplan et al., 2017].

Largest loose subset: the data structure

Largest loose subset $\equiv$ anti-chain in $(\mathcal{D}, \prec)$

Query $q \in \mathcal{D}$, $\mathscr{D}(\mathcal{D})$ returns $d \in \mathcal{D}$ with $d \subseteq q$

For each $x \in \mathcal{D}$: $\delta_x(p) = \|c_x - p\| + r_x$.
$d \subseteq q \iff \delta_d(c_q) \leq r_q$.



Dynamically maintain $F(p) = \min_{d \in \mathcal{D}} \delta_d(p)$ — lower envelope of surfaces $\{\delta_d \mid d \in \mathcal{D}\}$ in $\mathbb{R}^3$ — in time $O(\log^{10+\varepsilon} n)$ [Kaplan et al., 2017].

Our result

Largest loose subset of disks in $\mathbb{R}^2$ in $O\left(n^{1.5} \log^{10+\varepsilon} n\right)$ time

Conclusion

- Framework for computing anti-chains in implicit posets

Conclusion

- Framework for computing anti-chains in implicit posets
- Similar frameworks using dynamic data structures [Efrat et al., 2001, Cabello and Mulzer, 2020]

Conclusion

- Framework for computing anti-chains in implicit posets
- Similar frameworks using dynamic data structures [Efrat et al., 2001, Cabello and Mulzer, 2020]
- Other results: Largest subset of non-crossing rectangles, isolated points, ...

Conclusion

- Framework for computing anti-chains in implicit posets
- Similar frameworks using dynamic data structures [Efrat et al., 2001, Cabello and Mulzer, 2020]
- Other results: Largest subset of non-crossing rectangles, isolated points, ...
- More applications?

Conclusion

- Framework for computing anti-chains in implicit posets
- Similar frameworks using dynamic data structures [Efrat et al., 2001, Cabello and Mulzer, 2020]
- Other results: Largest subset of non-crossing rectangles, isolated points, ...
- More applications?

Thank you!

Robert P. Dilworth. *A decomposition theorem for partially ordered sets*. *Annals of Mathematics*, 51(1): 161–166, 1950.

John E. Hopcroft and Richard M. Karp. *An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs*. *SIAM J. Comput.*, 2(4): 225–231, 1973.

Alon Efrat, Alon Itai, and Matthew J. Katz. *Geometry helps in bottleneck matching and related problems*. *Algorithmica*, 31(1): 1–28, 2001.

Sergio Cabello and Wolfgang Mulzer. *Minimum cuts in geometric intersection graphs*. *CoRR*, abs/2005.00858, 2020. arXiv: `2005.00858`.

References ii

Timothy M. Chan and Konstantinos Tsakalidis. *Dynamic orthogonal range searching on the ram, revisited*. *33rd Symp. on Comput. Geom.* (SoCG), 28:1–28:13, 2017.

Haim Kaplan, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, and Micha Sharir. *Dynamic planar voronoi diagrams for general distance functions and their algorithmic applications*. *28th Symp. on Discrete Algorithms* (SODA), 2495–2504, 2017.