Shortest Paths in Intersection Graphs of Unit Disks

Sergio Cabello
sergio.cabello@fmf.uni-lj.si
University of Ljubljana
Slovenija

Material based on joint work with
Miha Jejčič and Panos Giannopoulos
Setting

$P$: $n$ points in the plane
$G(P)$: connect two points when distance $\leq 1$
intersection graph congruent disks
Setting

$P$: $n$ points in the plane
$G(P)$: connect two points when distance $\leq 1$

intersection graph congruent disks

Objective: fast computation of sssp in $G(P)$
Motivation

Bounded communication range:

- minimize hops/links $\rightarrow$ unweighted $G(P)$
- minimize energy $\rightarrow$ weighted $G(P)$
Motivation

Bounded communication range:

- minimize hops/links $\rightarrow$ unweighted $G(P)$
- minimize energy $\rightarrow$ weighted $G(P)$

Separation in the plane:

- set $D$ of unit disks
- $s$ and $t$ in $\mathbb{R}^2 \setminus \bigcup D$
- $\min |D'|$ s.t. $D' \subseteq D$, $D'$ separates $s$ and $t$
Overview

- Setting/Motivation
- Related work for sssp
- Unweighted
  - $O(n \log n)$ time
  - Implementable: Delaunay, Voronoi, point location
- Weighted:
  - $O(n^{1+\varepsilon})$ time
  - Unimplementable: dynamic bichromatic closest pair, shallow cuttings
- Separation with unit disks:
  - $O(n^2 \text{ polylog } n)$ time
  - Implementable, but many ingredients
Overview

- Setting/Motivation
- Related work for sssp
- Unweighted
  - $O(n \log n)$ time
  - Implementable: Delaunay, Voronoi, point location
- Weighted:
  - $O(n^{1+\epsilon})$ time
  - Unimplementable: dynamic bichromatic closest pair, shallow cuttings
- Separation with unit disks:
  - $O(n^2 \text{polylog } n)$ time
  - Implementable, but many ingredients
Related work

exact SSSP

- Roditty and Segal, 2011
  - unweighted: $O(n \log^6 n)$ expected time via Chan’s dynamic NN DS
  - weighted: $O(n^{4/3+\varepsilon})$ time

- C. and Jejčič, 2014
  - unweighted: $O(n \log n)$ time; implementable
  - weighted: $O(n^{1+\varepsilon})$ time
More related work

- Roditty and Segal, 2011
  - \((1 + \varepsilon)\)-approximate distance oracles, improving Bose, Maheshwari, Narasimhan, Smid, and Zeh, 2004.

- Gao and Zhang, 2005
  - WSPD of size \(O(n \log n)\) for unit-disk metric
  - \((1 + \varepsilon)\)-approximate sssp distance in \(O(n \log n)\) time

- Chan and Efrat, 2001 (Fuel consumption)
  - distances \(\ell : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}_{>0}\)
  - \(O(n \log n)\) time when \(\ell(p, q) = f(|pq|) \cdot |pq|^2\), \(f\) increasing.
  - \(O(n^{4/3+\varepsilon})\) time when \(\ell\) has constant size description.
More related work

- Roditty and Segal, 2011
  - $(1 + \varepsilon)$-approximate distance oracles, improving Bose, Maheshwari, Narasimhan, Smid, and Zeh, 2004.

- Gao and Zhang, 2005
  - WSPD of size $O(n \log n)$ for unit-disk metric
  - $(1 + \varepsilon)$-approximate sssp distance in $O(n \log n)$ time

- Chan and Efrat, 2001 (Fuel consumption)
  - Distances $\ell : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}_{>0}$
  - $O(n \log n)$ time when $\ell(p, q) = f(|pq|) \cdot |pq|^2$, $f$ increasing.
  - $O(n^{4/3 + \varepsilon})$ time when $\ell$ has constant size description.

- Faster algorithms for geometric intersection graphs
Overview

- Setting/Motivation
- Related work for sssp
- Unweighted
  - $O(n \log n)$ time
  - Implementable: Delaunay, Voronoi, point location
- Weighted:
  - $O(n^{1+\varepsilon})$ time
  - Unimplementable: dynamic bichromatic closest pair, shallow cuttings
- Separation with unit disks:
  - $O(n^2 \text{ polylog } n)$ time
  - Implementable, but many ingredients
Unweighted

- BFS in $G(P)$ without building $G(P)$
- $W_i = \{p \in P \mid d_{G(P)}(s, p) = i\}$
- Build $W_0 = \{s\}$
- Iteratively build $W_i$ from $W_{i-1}$
- Edge connecting $p$ to $\text{NN}(p, W_{i-1})$ for all $p \in W_i$
- Until $W_i$ empty
Unweighted - Growing $W_i$
Unweighted - Growing $W_i$
Unweighted

- BFS in $G(P)$ without building $G(P)$
- $W_i = \{p \in P \mid d_{G(P)}(s, p) = i\}$
- Build $W_0 = \{s\}$
- Iteratively build $W_i$ from $W_{i-1}$
  - edge connecting $p$ to $\text{NN}(p, W_{i-1})$ for all $p \in W_i$
- Until $W_i$ empty
- Use $DT(P)$ to guide the search of candidate points for $W_i$
- Candidate points for $W_i$:
  - points adjacent to $W_{i-1}$ in $DT(P)$
  - points adjacent to $W_i$ in $DT(P)$
- Is this good enough?
Unweighted - Growing $W_i$
Unweighted - Growing $W_i$
Unweighted - Growing $W_i$
Lemma

Let \( p \in W_i \).

There exists a path \( q_1, \ldots, q_k = p \) in \( G(P) \cap DT(P) \) with \( q_1 \in W_{i-1} \) and \( q_2, \ldots, q_k \in W_i \).
Lemma

Let \( p \in W_i \).

There exists a path \( q_1, \ldots, q_k = p \) in \( G(P) \cap DT(P) \) with \( q_1 \in W_{i-1} \) and \( q_2, \ldots, q_k \in W_i \).
**Unweighted - Growing $W_i$**

**Lemma**

Let $p \in W_i$.

*There exists a path $q_1, \ldots, q_k = p$ in $G(P) \cap DT(P)$ with $q_1 \in W_{i-1}$ and $q_2, \ldots, q_k \in W_i$.*

- Data structure to decide whether candidate $q$ is $\in W_i$
  - DS for $NN(q, W_{i-1})$
  - check if distance $\leq 1$
- each edge of $DT(P)$ explored twice
- building $W_i$ takes time

$$O\left( (|W_{i-1}| + |W_i| + \sum_{p \in W_{i-1} \cup W_i} \deg_{DT(P)}(p)) \log n \right)$$
1. for $p \in P$ do
2. \hspace{1em} dist[$p$] $\leftarrow \infty$;
3. \hspace{1em} dist[$s$] $\leftarrow 0$
4. build the Delaunay triangulation $DT(P)$
5. $W_0 \leftarrow \{s\}$
6. $i \leftarrow 1$
7. while $W_{i-1} \neq \emptyset$ do
8. \hspace{1em} build data structure for nearest neighbour queries in $W_{i-1}$
9. \hspace{1em} $Q \leftarrow W_{i-1}$ (* generator of candidate points *)
10. $W_i \leftarrow \emptyset$
11. while $Q \neq \emptyset$ do
12. \hspace{2em} $q$ an arbitrary point of $Q$
13. \hspace{2em} remove $q$ from $Q$
14. \hspace{2em} for $qp$ edge in $DT(P)$ do
15. \hspace{3em} $w \leftarrow NN(W_{i-1}, q)$
16. \hspace{3em} if dist[$p$] $= \infty$ and $|pw| \leq 1$ then
17. \hspace{4em} dist[$p$] $\leftarrow i$
18. \hspace{4em} add $p$ to $Q$ and to $W_i$
19. \hspace{2em} $i \leftarrow i + 1$
20. return dist[.]
Overview

- Setting/Motivation
- Related work for sssp
- Unweighted ← Done
  - $O(n \log n)$ time
  - Implementable: Delaunay, Voronoi, point location
- Weighted:
  - $O(n^{1+\varepsilon})$ time
  - Unimplementable: dynamic bichromatic closest pair, shallow cuttings
- Separation with unit disks:
  - $O(n^2 \text{ polylog } n)$ time
  - Implementable, but many ingredients
Overview

- Setting/Motivation
- Related work for sssp
- Unweighted
  - $O(n \log n)$ time
  - Implementable: Delaunay, Voronoi, point location
- Weighted:
  - $O(n^{1+\varepsilon})$ time
  - Unimplementable: dynamic bichromatic closest pair, shallow cuttings
- Separation with unit disks:
  - $O(n^2 \text{polylog } n)$ time
  - Implementable, but many ingredients
Weighted - Ingredient - BCP

Bichromatic closest pair (BCP)

- weighted Euclidean
- red points $R$
- blue points $B$
- weights $w_b$ for each $b \in B$
- $\delta: B \times R \rightarrow \mathbb{R} \quad \delta(b, r) = w_b + |br|$
Weighted - Ingredient - BCP

Bichromatic closest pair (BCP)

- weighted Euclidean
- red points $R$
- blue points $B$
- weights $w_b$ for each $b \in B$
- $\delta: B \times R \to \mathbb{R}$
  \[ \delta(b, r) = w_b + |br| \]

- Eppstein 1995 + Agarwal, Efrat, Sharir 1999: dynamic BCP in $O(n^\varepsilon)$ amortized per operation
  - insertion/deletion
  - query for minima $\min_{r,b} \delta(r, b)$
Weighted - Idea

- Modification of Dijkstra’s algorithm

- **Standard** Dijsktra’s algorithm
  - keep an array dist[·]
  - dist[v] is an (over)estimate of $d_{G(P)}(s, v)$
  - keep partition $P$ into $S$ and $P \setminus S$
  - $S$ contains vertices with $\text{dist}[s] = d_{G(P)}(s, v)$
Weighted - Idea

- Modification of Dijkstra’s algorithm

- **Standard** Dijkstra’s algorithm
  - keep an array dist[\cdot]
  - dist[v] is an (over)estimate of \(d_{G(P)}(s, v)\)
  - keep partition \(P\) into \(S\) and \(P \setminus S\)
  - \(S\) contains vertices with \(\text{dist}[s] = d_{G(P)}(s, v)\)
  - an iteration: find a vertex

  \[
  q^* \in \arg \min_{q \in P \setminus S} \min_{p \in S, |pq| \leq 1} \text{dist}[p] + |pq|
  \]

  - move \(q^*\) from \(P \setminus S\) to \(S\)
  - usually we keep \(\text{dist}[q] = \min_{p \in S} \text{dist}[p] + |pq|\)
Weighted - Idea

- Modification of Dijkstra’s algorithm
  - array dist[], dist[v] is an (over)estimate of $d_{G(P)}(s, v)$
  - keep partition $P$ into $S$ and $R = P \setminus S$
  - partition $S$ into $D$ and $B$
  - $D$ are “dead” points, irrelevant when $\min \text{dist}[p] + |pq|$
  - an iteration: find a pair

\[
(b^*, r^*) \in \arg \min_{(b, r) \in B \times R} \text{dist}[b] + |br|
\]

- if $|b^* r^*| > 1$, move $b^*$ from $B$ to $D$
- else normal Dijsktra’s step
Weighted - Idea

- Modification of Dijkstra’s algorithm
1. for \( p \in P \) do
2. \( \text{dist}[p] \leftarrow \infty \)
3. \( \text{dist}[s] \leftarrow 0 \)
4. \( B \leftarrow \{ s \} \)
5. \( D \leftarrow \emptyset \)
6. \( R \leftarrow P \setminus \{ s \} \)
7. store \( R \cup B \) in a BCP dynamic DS wrt \( \delta(b, r) = \text{dist}[b] + |br| \)
8. while \( R \neq \emptyset \) do
9. \((b^*, r^*) \leftarrow \text{BCP}(B, R)\)
10. if \( |b^* r^*| > 1 \) then
11. \( \text{delete}(B, b^*) \)
12. \( D \leftarrow D \cup \{ b^* \} \)
13. else
14. \( \text{dist}[r^*] \leftarrow \text{dist}[b^*] + |b^* r^*| \)
15. \( \text{delete}(R, r^*) \)
16. \( \text{insert}(B, r^*) \)
17. return \( \text{dist}[\cdot] \)
Overview

- Setting/Motivation
- Related work for sssp
- Unweighted
  - $O(n \log n)$ time
  - Implementable: Delaunay, Voronoi, point location
- Weighted ← Done
  - $O(n^{1+\varepsilon})$ time
  - Unimplementable: dynamic bichromatic closest pair, shallow cuttings
- Separation with unit disks:
  - $O(n^2 \text{polylog } n)$ time
  - Implementable, but many ingredients
Overview

- Setting/Motivation
- Related work for sssp
- Unweighted
  - $O(n \log n)$ time
  - Implementable: Delaunay, Voronoi, point location
- Weighted
  - $O(n^{1+\varepsilon})$ time
  - Unimplementable: dynamic bichromatic closest pair, shallow cuttings
- Separation with unit disks:
  - $O(n^2 \text{ polylog } n)$ time
  - Implementable, but many ingredients
Setting

- set $D$ of unit disks
- $s, t$ points in $\mathbb{R}^2 \setminus \bigcup D$
- $P$ centers of the disks
- $G(P)$ as before, with distance 2
- C. and Giannopoulos
  - $O(n^2 + n \cdot |E(G(D))|)$
  - general objects
Setting

- set $D$ of unit disks
- $s, t$ points in $\mathbb{R}^2 \setminus \bigcup D$
- $P$ centers of the disks
- $G(P)$ as before, with distance 2
- C. and Giannopoulou$
  \quad \bullet \quad O(n^2 + n \cdot |E(G(D))|)$
  \quad \bullet \quad$general objects$
- today: $O(n^2 \log^4 n)$ for unit disks
- also easier to explain & understand
Algorithm of C. & Giannopoulos

- for a closed walk $\pi = p_1 \ldots p_k p_1$ in $G(P)$

$$N(\pi) = \pi \cap \overline{st} \pmod{2}$$
Algorithm of C. & Giannopoulous

- for a closed walk \( \pi = p_1 \ldots p_k p_1 \) in \( G(P) \)
  \[ N(\pi) = \pi \cap \overline{st} \pmod{2} \]

- if \( N(\pi) = 1 \) then \( \bigcup_{p \in V(\pi)} \text{disk}(p, 1) \) separates \( s \) and \( t \)
- shortest closed walk \( \pi \) with \( N(\pi) = 1 \) gives an optimal solution
- shortest closed walk \( \pi \) with \( N(\pi) = 1 \) is actually a cycle
- enough to restrict the search to fundamental cycles: defined by a BFS-tree and an additional edge

\[
\min |V(\text{cycle}(T_r, e))| \\
\text{s.t. } r \in P, \ T_r \text{ BFS cycle from } r \\
e \in E(G(P)) \setminus E(T_r) \\
N(\text{cycle}(T_r, e)) = 1
\]
Algorithm of C. & Giannopoulos

- for a closed walk $\pi = p_1 \ldots p_k p_1$ in $G(P)$
  
  $$N(\pi) = \pi \cap \overline{st} \pmod{2}$$
for each $r$ in $P$

- construct BFS tree $T_r$ from $r$
- attach to each $p \in P$ the label $d[p] = d_{G(P)}(s, p)$
- solve

$$\min d[p] + d[q]$$

s.t. $|pq| \leq 1$

$$N(\text{cycle}(T_r, pq)) = 1$$

- break $P$ into groups depending on $N(T_r[r, p])$
- use range searching & vertical shooting to solve the resulting problems
Adaptation

for each $r$ in $P$
Adaptation

for each $r$ in $P$
Resulting problem - Example

- vertical segment $st$
- points $A$ and $B$ with weights $(w_p)_{p \in A \cup B}$

\[
\begin{align*}
\min & \quad w_a + w_b \\
\text{s.t.} & \quad a \in A, b \in B \\
& \quad |ab| \leq 1 \\
& \quad ab \cap st \neq \emptyset
\end{align*}
\]

Solvable in $O(n \log^4 n)$
Conclusions

- shortest paths in unit disk graphs
  - $O(n \log n)$ for unweighted
  - $O(n^{1+\varepsilon})$ for weighted
- Improvement for separation with unit disks

Open problems:
- Can we compute efficiently a compact representation of the distances in all the graphs $G \leq \lambda(P)$?
- Given $s, t \in P$ and $k \in \mathbb{N}$, find minimum $\lambda$ such that $d_G \leq \lambda(P)(s,t) \leq k$.
  - Easy in $\tilde{O}(n^{4/3})$.
- Dual to separation problem – barrier resilience: find $(s,t)$-curve that touches as few disks as possible.
  - Polynomial? Hard? Or both?

Sergio Cabello
Conclusions

- shortest paths in unit disk graphs
  - $O(n \log n)$ for unweighted
  - $O(n^{1+\varepsilon})$ for weighted
- Improvement for separation with unit disks

- Open problems:
  - Can we compute efficiently a compact representation of the distances in all the graphs $G_{\leq \lambda}(P)$?
  - Given $s, t \in P$ and $k \in \mathbb{N}$, find minimum $\lambda$ such that $d_{G_{\leq \lambda}(P)}(s, t) \leq k$.
    Easy in $\tilde{O}(n^{4/3})$.
  - Dual to separation problem – barrier resilience: find $(s, t)$-curve that touches as few disks as possible. Polynomial? Hard?
Conclusions

- shortest paths in unit disk graphs
  - $O(n \log n)$ for unweighted
  - $O(n^{1+\varepsilon})$ for weighted
- Improvement for separation with unit disks

- Open problems:
  - Can we compute efficiently a compact representation of the distances in all the graphs $G_{\leq \lambda}(P)$?
  - Given $s, t \in P$ and $k \in \mathbb{N}$, find minimum $\lambda$ such that $d_{G_{\leq \lambda}(P)}(s, t) \leq k$.
    Easy in $\tilde{O}(n^{4/3})$.
  - Dual to separation problem – barrier resilience: find $(s, t)$-curve that touches as few disks as possible.
    Polynomial? Hard? Or both?

Sergio Cabello