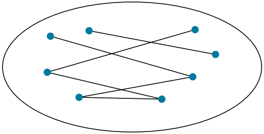
Fixed-Parameter Sensitivity Oracles

Davide Bilò, Katrin Casel, Keerti Choudhary, Sarel Cohen, Tobias Friedrich,
J.A. Gregor Lagodzinski, Martin Schirneck, and Simon Wietheger

13th Innovations in Theoretical Computer Science Conference
January 31–February 3, 2022

Sensitivity Oracles

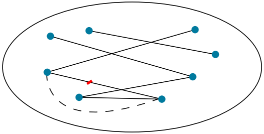
a.k.a. fault-tolerant data structures, algorithms for emergency planning, failure-prone graphs



Maintain graph property $P(G)$ (distances, connectivity, ...) under edge failures.

Sensitivity Oracles

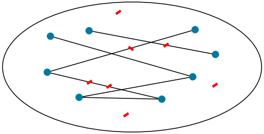
a.k.a. fault-tolerant data structures, algorithms for emergency planning, failure-prone graphs



Maintain graph property $P(G)$ (distances, connectivity, ...) under edge failures.

Sensitivity Oracles

a.k.a. fault-tolerant data structures, algorithms for emergency planning, failure-prone graphs

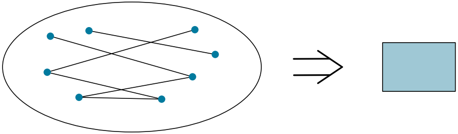


Maintain graph property $P(G)$ (distances, connectivity, ...) under edge failures.

- Sensitivity: failures in batches, maximum number f known.

Sensitivity Oracles

a.k.a. fault-tolerant data structures, algorithms for emergency planning, failure-prone graphs

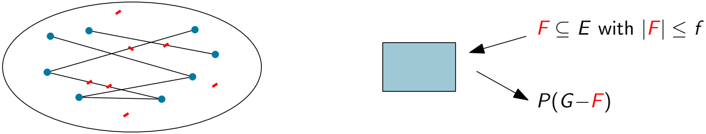


Maintain graph property $P(G)$ (distances, connectivity, ...) under edge failures.

- Sensitivity: failures in batches, maximum number f known.
- Data structure: preprocess once, query when needed.

Sensitivity Oracles

a.k.a. fault-tolerant data structures, algorithms for emergency planning, failure-prone graphs

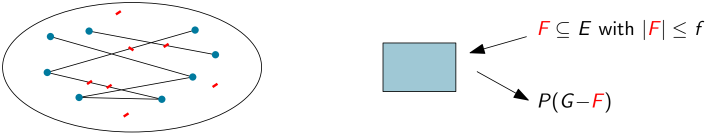


Maintain graph property $P(G)$ (distances, connectivity, ...) under edge failures.

- Sensitivity: failures in batches, maximum number f known.
- Data structure: preprocess once, query when needed.

Sensitivity Oracles

a.k.a. fault-tolerant data structures, algorithms for emergency planning, failure-prone graphs



Maintain graph property $P(G)$ (distances, connectivity, ...) under edge failures.

- Sensitivity: failures in batches, maximum number f known.
- Data structure: preprocess once, query when needed.

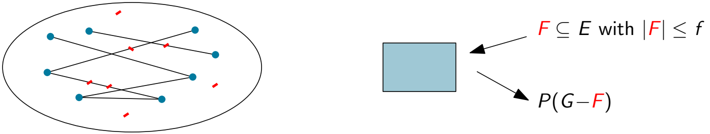
Fixed-Parameter Tractability

Fixed-parameter tractable (FPT): on n-vertex G with parameter k

$P(G,k)$ computable in time $O(g(k) \cdot n^c)$ for some function g and constant c.

Sensitivity Oracles

a.k.a. fault-tolerant data structures, algorithms for emergency planning, failure-prone graphs



Maintain graph property $P(G)$ (distances, connectivity, ...) under edge failures.

- Sensitivity: failures in batches, maximum number f known.
- Data structure: preprocess once, query when needed.

Fixed-Parameter Tractability

Fixed-parameter tractable (FPT): on n-vertex G with parameter k

$P(G, k)$ computable in time $O(g(k) \cdot n^c)$ for some function g and constant c.

- Textbook: Compute a vertex cover of size k in time $O(2^k k \cdot n)$.

Fixed-Parameter Sensitivity Oracles

Combine both areas: Oracle with sensitivity f for FPT property $P(G, k)$.

Fixed-Parameter Sensitivity Oracles

Combine both areas: Oracle with sensitivity f for FPT property $P(G, k)$.

- Answer: $P(G{-}F, k)$ with **same** parameter k.

Fixed-Parameter Sensitivity Oracles

Combine both areas: Oracle with sensitivity f for FPT property $P(G,k)$.

- Answer: $P(G-F,k)$ with **same** parameter k.
- Preprocessing time: $O(g(f,k) \cdot n^c)$ - **not** $g(k) \cdot n^{O(f)}$.

Fixed-Parameter Sensitivity Oracles

Combine both areas: Oracle with sensitivity f for FPT property $P(G, k)$.

- Answer: $P(G-F, k)$ with **same** parameter k.
- Preprocessing time: $O(g(f, k) \cdot n^c)$ - **not** $g(k) \cdot n^{O(f)}$.
- Query time: "significantly faster than computing $P(G-F, k)$ from scratch".

Fixed-Parameter Sensitivity Oracles

Combine both areas: Oracle with sensitivity f for FPT property $P(G, k)$.

- Answer: $P(G-F, k)$ with **same** parameter k.
- Preprocessing time: $O(g(f, k) \cdot n^c)$ - **not** $g(k) \cdot n^{O(f)}$.
- Query time: "significantly faster than computing $P(G-F, k)$ from scratch".

FPT for dynamic graphs:

- Iwata & Oka [SWAT 2014]
- Alman, Mnich & Vassilevska Williams [TALG 2020]
- Chen, Czerwinski, Disser, Feldmann, Hermelin, Nadara, Pilipczuk, Pilipczuk, Sorge, Wróblewski & Zych-Pawlewicz [SODA 2021]

Fixed-Parameter Sensitivity Oracles

Combine both areas: Oracle with sensitivity f for FPT property $P(G, k)$.

- Answer: $P(G{-}F, k)$ with **same** parameter k.
- Preprocessing time: $O(g(f, k) \cdot n^c)$ - **not** $g(k) \cdot n^{O(f)}$.
- Query time: "significantly faster than computing $P(G{-}F, k)$ from scratch".

FPT for dynamic graphs:

- Iwata & Oka [SWAT 2014]
- Alman, Mnich & Vassilevska Williams [TALG 2020]
- Chen, Czerwinski, Disser, Feldmann, Hermelin, Nadara, Pilipczuk, Pilipczuk, Sorge, Wróblewski & Zych-Pawlewicz [SODA 2021]

Fault tolerance on special graph class & take insights back to FPT:

- Lochet, Lokshtanov, Misra, Saurabh, Sharma & Zehavi [ITCS 2020]

Vertex Cover

Does $G-F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Vertex Cover

Does $G-F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Vertex Cover

Does $G-F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Trade-offs between space, preprocessing time, and query time

depending on f and k.

Vertex Cover

Does $G - F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Trade-offs between space, preprocessing time, and query time

depending on f and k. **This talk:** $f \leq k$ for simplicity.

Vertex Cover

Does $G - F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Trade-offs between space, preprocessing time, and query time

depending on f and k. **This talk:** $f \leq k$ for simplicity.

Solution 1:

- Space: $O(3^{f+k})$, prep. time: $O(3^{f+k}\,\mathrm{poly}(f,k)) + \mathrm{poly}(n)$, query time: $O(2^f)$.

Vertex Cover

Does $G-F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Trade-offs between space, preprocessing time, and query time

depending on f and k. **This talk:** $f \leq k$ for simplicity.

Solution 1:

- Space: $O(3^{f+k})$, prep. time: $O(3^{f+k}\,\mathrm{poly}(f,k)) + \mathrm{poly}(n)$, query time: $O(2^f)$.
- Add "this edge fails" option to standard branching.

Vertex Cover

Does $G-F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Trade-offs between space, preprocessing time, and query time

depending on f and k. **This talk:** $f \leq k$ for simplicity.

Solution 1:

- Space: $O(3^{f+k})$, prep. time: $O(3^{f+k}\,\mathrm{poly}(f,k)) + \mathrm{poly}(n)$, query time: $O(2^f)$.
- Add "this edge fails" option to standard branching.

Solution 2:

- Space: $O(2^{f+k^2+k})$, prep. time: $O(2^{f+k^2} 2.548^k) + \mathrm{poly}(n)$, query time: $O(k^2)$.

Vertex Cover

Does $G-F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Trade-offs between space, preprocessing time, and query time

depending on f and k. **This talk:** $f \leq k$ for simplicity.

Solution 1:

- Space: $O(3^{f+k})$, prep. time: $O(3^{f+k}\,\mathrm{poly}(f,k)) + \mathrm{poly}(n)$, query time: $O(2^f)$.
- Add "this edge fails" option to standard branching.

Solution 2:

- Space: $O(2^{f+k^2+k})$, prep. time: $O(2^{f+k^2} 2.548^k) + \mathrm{poly}(n)$, query time: $O(k^2)$.
- Combine branch & bound with fault-tolerant lookup trees.

Vertex Cover

Does $G - F$ have a vertex cover of size at most k?

Decidable in time $O(1.274^k + kn)$ [Chen, Kanj & Xia, TCS 2010].

Kernelization: equivalent graph with $O(k^2 + kf)$ vertices in polynomial time.

Trade-offs between space, preprocessing time, and query time

depending on f and k. **This talk:** $f \leq k$ for simplicity.

Solution 1:

- Space: $O(3^{f+k})$, prep. time: $O(3^{f+k}\,\mathrm{poly}(f,k)) + \mathrm{poly}(n)$, query time: $O(2^f)$.
- Add "this edge fails" option to standard branching.

Solution 2:

- Space: $O(2^{f+k^2+k})$, prep. time: $O(2^{f+k^2} 2.548^k) + \mathrm{poly}(n)$, query time: $O(k^2)$.
- Combine branch & bound with fault-tolerant lookup trees.

Solution 3:

- Space: $O(k^2)$, preprocessing time: $\mathrm{poly}(n)$, query time: $O(1.274^k)$.

[Alman, Mnich & Vassilevska Williams, TALG 2020]

k-Path

Does $G-F$ have a simple (directed) path of length k?

Deterministic: $O(2.554^k \cdot \text{poly}(n))$ [Tsur, TCS 2019]; randomized: $O(1.657^k \cdot \text{poly}(n))$ [Björklund, Husfeldt, Kaski & Koivisto, JCSS 2010].

k-Path

Does $G-F$ have a simple (directed) path of length k?

Deterministic: $O(2.554^k \cdot \text{poly}(n))$ [Tsur, TCS 2019]; randomized: $O(1.657^k \cdot \text{poly}(n))$ [Björklund, Husfeldt, Kaski & Koivisto, JCSS 2010].

Reminder: $f \leq k$.

Solution 1:

- Space: $O(k^{f+1})$, rand. prep. time: $O(k^f 1.657^k \cdot \text{poly}(n))$, query time: $O(f^2)$.

k-Path

Does $G - F$ have a simple (directed) path of length k?

Deterministic: $O(2.554^k \cdot \text{poly}(n))$ [Tsur, TCS 2019]; randomized: $O(1.657^k \cdot \text{poly}(n))$ [Björklund, Husfeldt, Kaski & Koivisto, JCSS 2010].

Reminder: $f \leq k$.

Solution 1:

- Space: $O(k^{f+1})$, rand. prep. time: $O(k^f 1.657^k \cdot \text{poly}(n))$, query time: $O(f^2)$.
- Fault-tolerant lookup tree of depth f, each node holds a k-path.

k-Path

Does $G-F$ have a simple (directed) path of length k?

Deterministic: $O(2.554^k \cdot \text{poly}(n))$ [Tsur, TCS 2019]; randomized: $O(1.657^k \cdot \text{poly}(n))$ [Björklund, Husfeldt, Kaski & Koivisto, JCSS 2010].

Reminder: $f \leq k$.

Solution 1:

- Space: $O(k^{f+1})$, rand. prep. time: $O(k^f 1.657^k \cdot \text{poly}(n))$, query time: $O(f^2)$.
- Fault-tolerant lookup tree of depth f, each node holds a k-path.

Solution 2:

- Space: $O(s\,k)$, rand. prep. time: $O(s\;1.657^k \cdot \text{poly}(n))$, query time: $O(s\,f)$, where $s = O((\frac{f+k}{f})^f (\frac{f+k}{k})^k f \cdot \log n)$. (Worst case: $s = O(2^{f+k} f \cdot \log n)$)

k-Path

Does $G-F$ have a simple (directed) path of length k?

Deterministic: $O(2.554^k \cdot \text{poly}(n))$ [Tsur, TCS 2019]; randomized: $O(1.657^k \cdot \text{poly}(n))$ [Björklund, Husfeldt, Kaski & Koivisto, JCSS 2010].

Reminder: $f \leq k$.

Solution 1:

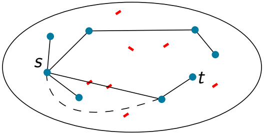
- Space: $O(k^{f+1})$, rand. prep. time: $O(k^f 1.657^k \cdot \text{poly}(n))$, query time: $O(f^2)$.
- Fault-tolerant lookup tree of depth f, each node holds a k-path.

Solution 2:

- Space: $O(s\,k)$, rand. prep. time: $O(s\ 1.657^k \cdot \text{poly}(n))$, query time: $O(s\,f)$, where $s = O((\frac{f+k}{f})^f(\frac{f+k}{k})^k f \cdot \log n)$. (Worst case: $s = O(2^{f+k} f \cdot \log n)$)
- Sample s subgraphs of G, discarding each edge with probability $f/(f+k)$.

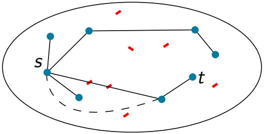
[Weimann & Yuster, TALG 2013]

Introduce Parameterization to Fault Tolerance



Fault-tolerant distance preservers:

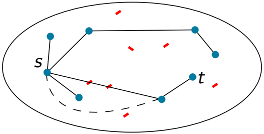
Introduce Parameterization to Fault Tolerance



Fault-tolerant distance preservers:

Graph G with source vertex s, compute subgraph $H \subseteq G$ s.t. for all vertices t and at most f edge failures F, $d_{G-F}(s,t) = d_{H-F}(s,t)$.

Introduce Parameterization to Fault Tolerance

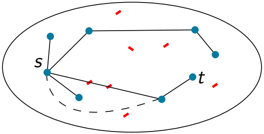


Fault-tolerant distance preservers:

Graph G with source vertex s, compute subgraph $H \subseteq G$ s.t. for all vertices t and at most f edge failures F, $d_{G-F}(s,t) = d_{H-F}(s,t)$.

Constructions with $\widetilde{O}(f\, n^{2-1/2^f})$ edges known, $\Omega(n^{2-1/(f+1)})$ edges required.
[Parter, PODC 2015] [Parter & Peleg, TALG 2016] [Bodwin, Grandoni, Parter & Vassilevska Williams, ICALP 2017]

Introduce Parameterization to Fault Tolerance



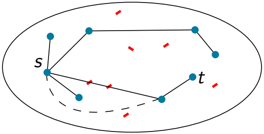
Fault-tolerant distance preservers:

Graph G with source vertex s, compute subgraph $H \subseteq G$ s.t. for all vertices t and at most f edge failures F, $d_{G-F}(s,t) = d_{H-F}(s,t)$.

Constructions with $\widetilde{O}(f\, n^{2-1/2^f})$ edges known, $\Omega(n^{2-1/(f+1)})$ edges required.
[Parter, PODC 2015] [Parter & Peleg, TALG 2016] [Bodwin, Grandoni, Parter & Vassilevska Williams, ICALP 2017]

Only preserve distances with $d_{G-F}(s,t) \leq d_G(s,t) + k$ for every $F \subseteq E$, $|F| \leq f$.

Introduce Parameterization to Fault Tolerance



Fault-tolerant distance preservers:

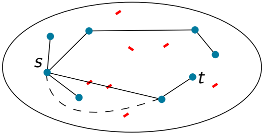
Graph G with source vertex s, compute subgraph $H \subseteq G$ s.t. for all vertices t and at most f edge failures F, $d_{G-F}(s,t) = d_{H-F}(s,t)$.

Constructions with $\widetilde{O}(f\, n^{2-1/2^f})$ edges known, $\Omega(n^{2-1/(f+1)})$ edges required.
[Parter, PODC 2015] [Parter & Peleg, TALG 2016] [Bodwin, Grandoni, Parter & Vassilevska Williams, ICALP 2017]

Only preserve distances with $d_{G-F}(s,t) \leq d_G(s,t) + k$ for every $F \subseteq E$, $|F| \leq f$.

- Distance preserver with $O(2^{fk+f+k} k \cdot n)$ edges in time $O(2^{fk+f+k} k^2 \cdot mn)$.

Introduce Parameterization to Fault Tolerance



Fault-tolerant distance preservers:

Graph G with source vertex s, compute subgraph $H \subseteq G$ s.t. for all vertices t and at most f edge failures F, $d_{G-F}(s,t) = d_{H-F}(s,t)$.

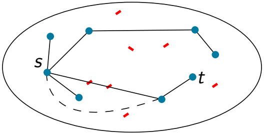
Constructions with $\widetilde{O}(f\, n^{2-1/2^f})$ edges known, $\Omega(n^{2-1/(f+1)})$ edges required.
[Parter, PODC 2015] [Parter & Peleg, TALG 2016] [Bodwin, Grandoni, Parter & Vassilevska Williams, ICALP 2017]

Only preserve distances with $d_{G-F}(s,t) \leq d_G(s,t) + k$ for every $F \subseteq E$, $|F| \leq f$.

- Distance preserver with $O(2^{fk+f+k} k \cdot n)$ edges in time $O(2^{fk+f+k} k^2 \cdot mn)$.

Randomized oracle reporting whether $d_{G-F}(s,t) \leq d_G(s,t) + k$.

Introduce Parameterization to Fault Tolerance



Fault-tolerant distance preservers:

Graph G with source vertex s, compute subgraph $H \subseteq G$ s.t. for all vertices t and at most f edge failures F, $d_{G-F}(s,t) = d_{H-F}(s,t)$.

Constructions with $\widetilde{O}(f\, n^{2-1/2^f})$ edges known, $\Omega(n^{2-1/(f+1)})$ edges required.
[Parter, PODC 2015] [Parter & Peleg, TALG 2016] [Bodwin, Grandoni, Parter & Vassilevska Williams, ICALP 2017]

Only preserve distances with $d_{G-F}(s,t) \leq d_G(s,t) + k$ for every $F \subseteq E$, $|F| \leq f$.

- Distance preserver with $O(2^{fk+f+k} k \cdot n)$ edges in time $O(2^{fk+f+k} k^2 \cdot mn)$.

Randomized oracle reporting whether $d_{G-F}(s,t) \leq d_G(s,t) + k$.

- Space: $O(2^k k^2 \cdot n^2)$, prep. time: $O(2^k k^\omega \cdot n^\omega)$, query time: $O(2^k k^\omega f^\omega)$.

  [Brand & Saranurak, FOCS 2019] ($\omega < 2.3729$ matrix multiplication exponent)

Conclusion

1. Fixed-parameter tractable algorithms are often already fault-tolerant.

Conclusion

1. Fixed-parameter tractable algorithms are often already fault-tolerant.
2. Sensitivity f fits naturally to the parameterized setting.

Conclusion

1. Fixed-parameter tractable algorithms are often already fault-tolerant.
2. Sensitivity f fits naturally to the parameterized setting.
3. Sensitivity oracles require careful trade-offs between space, preprocessing time, and query time.

Conclusion

1. Fixed-parameter tractable algorithms are often already fault-tolerant.
2. Sensitivity f fits naturally to the parameterized setting.
3. Sensitivity oracles require careful trade-offs between space, preprocessing time, and query time.
4. Parameterization can help relax bottlenecks in fault tolerance.

Conclusion

1. Fixed-parameter tractable algorithms are often already fault-tolerant.
2. Sensitivity f fits naturally to the parameterized setting.
3. Sensitivity oracles require careful trade-offs between space, preprocessing time, and query time.
4. Parameterization can help relax bottlenecks in fault tolerance.

Open Problems

- Fixed-parameter sensitivity oracles for more FPT properties.
- Bring down the (exponential) dependencies on f and k.
- More fault-tolerant problems that benefit from parameterization.

Conclusion

1. Fixed-parameter tractable algorithms are often already fault-tolerant.
2. Sensitivity f fits naturally to the parameterized setting.
3. Sensitivity oracles require careful trade-offs between space, preprocessing time, and query time.
4. Parameterization can help relax bottlenecks in fault tolerance.

Open Problems

- Fixed-parameter sensitivity oracles for more FPT properties.
- Bring down the (exponential) dependencies on f and k.
- More fault-tolerant problems that benefit from parameterization.

Thank you.