Sensitivity Analysis of the Maximum Matching Problem

Yuichi Yoshida

Samson Zhou

Average Sensitivity for Graph Algorithms

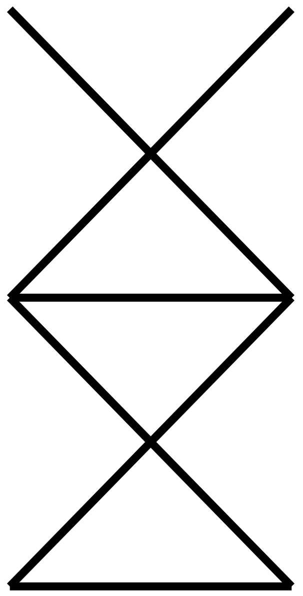
- Measure of how much a discrete algorithm changes when the input changes
- For a deterministic graph algorithm A on a graph G with n vertices, the average sensitivity is defined as

$$\mathop{\mathrm{E}}_{e \sim E(G)} [|A(G) \triangle A(G - e)|]$$

where $\triangle$ is the symmetric difference

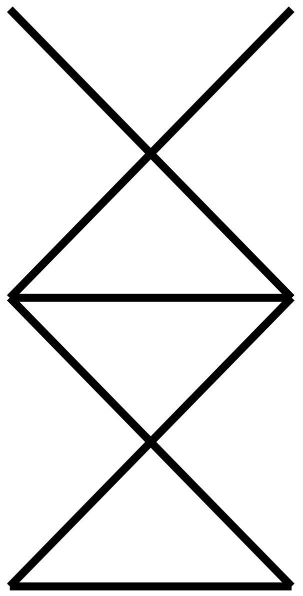
Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximal matching on graph with 6 nodes



Average Sensitivity for Graph Algorithms

- ❖ Example: Greedy algorithm for maximal matching on graph with 6 nodes

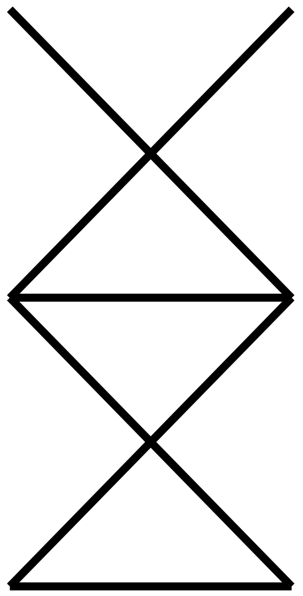


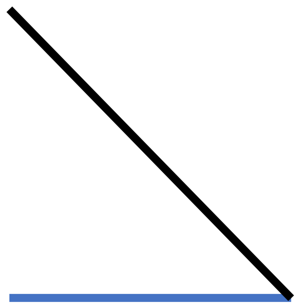
Average Sensitivity for Graph Algorithms

- ❖ Example: Greedy algorithm for maximal matching on graph with 6 nodes

Average Sensitivity for Graph Algorithms

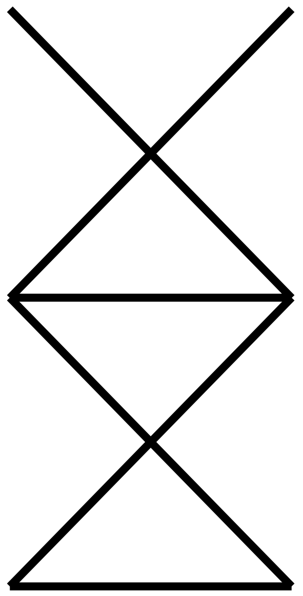
- Example: Greedy algorithm for maximal matching on graph with 6 nodes

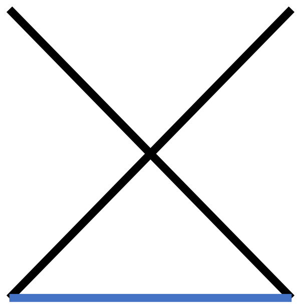




Average Sensitivity for Graph Algorithms

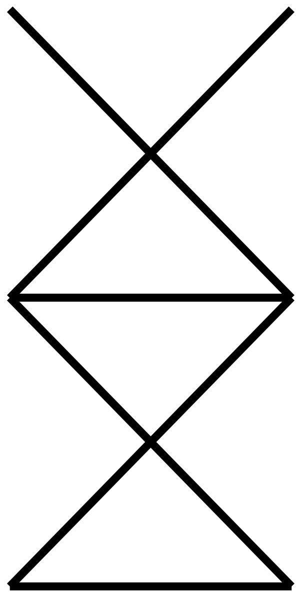
- Example: Greedy algorithm for maximal matching on graph with 6 nodes

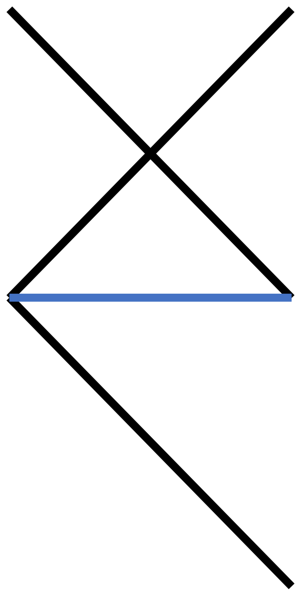




Average Sensitivity for Graph Algorithms

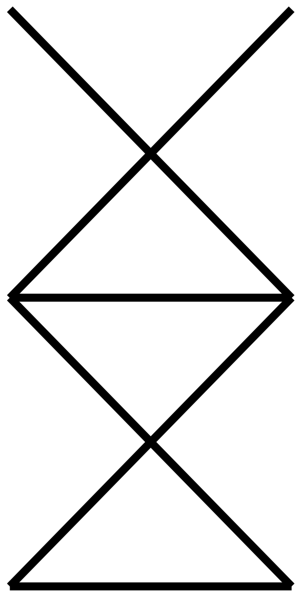
❖ Example: Greedy algorithm for maximal matching on graph with 6 nodes

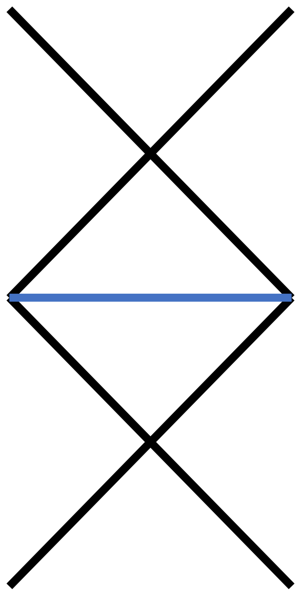




Average Sensitivity for Graph Algorithms

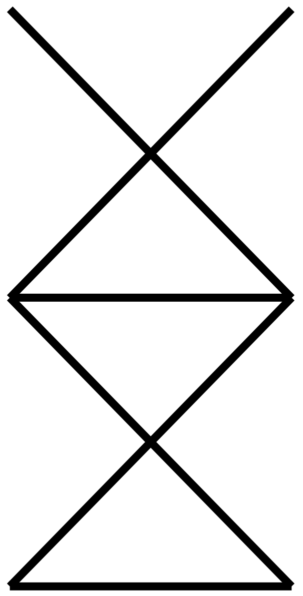
- ❖ Example: Greedy algorithm for maximal matching on graph with 6 nodes

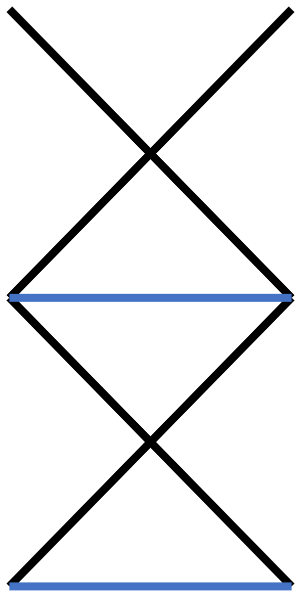


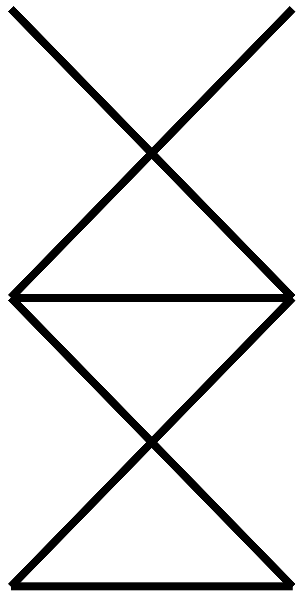


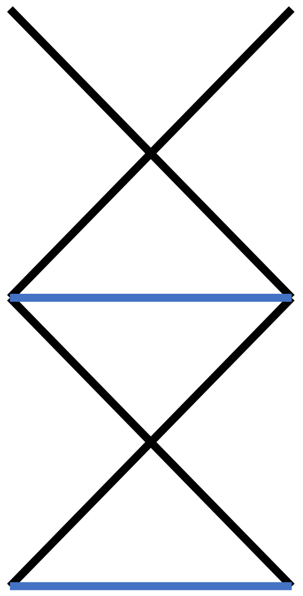
Average Sensitivity for Graph Algorithms

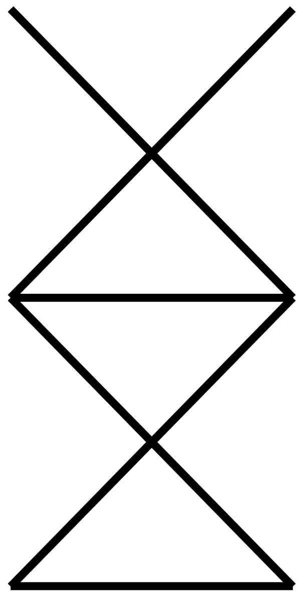
- Example: Greedy algorithm for maximal matching on graph with 6 nodes

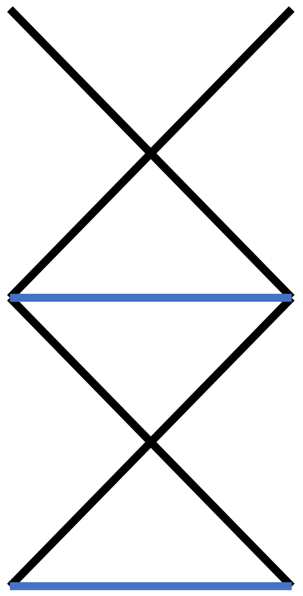


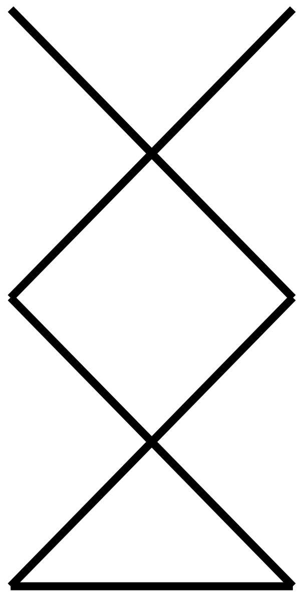


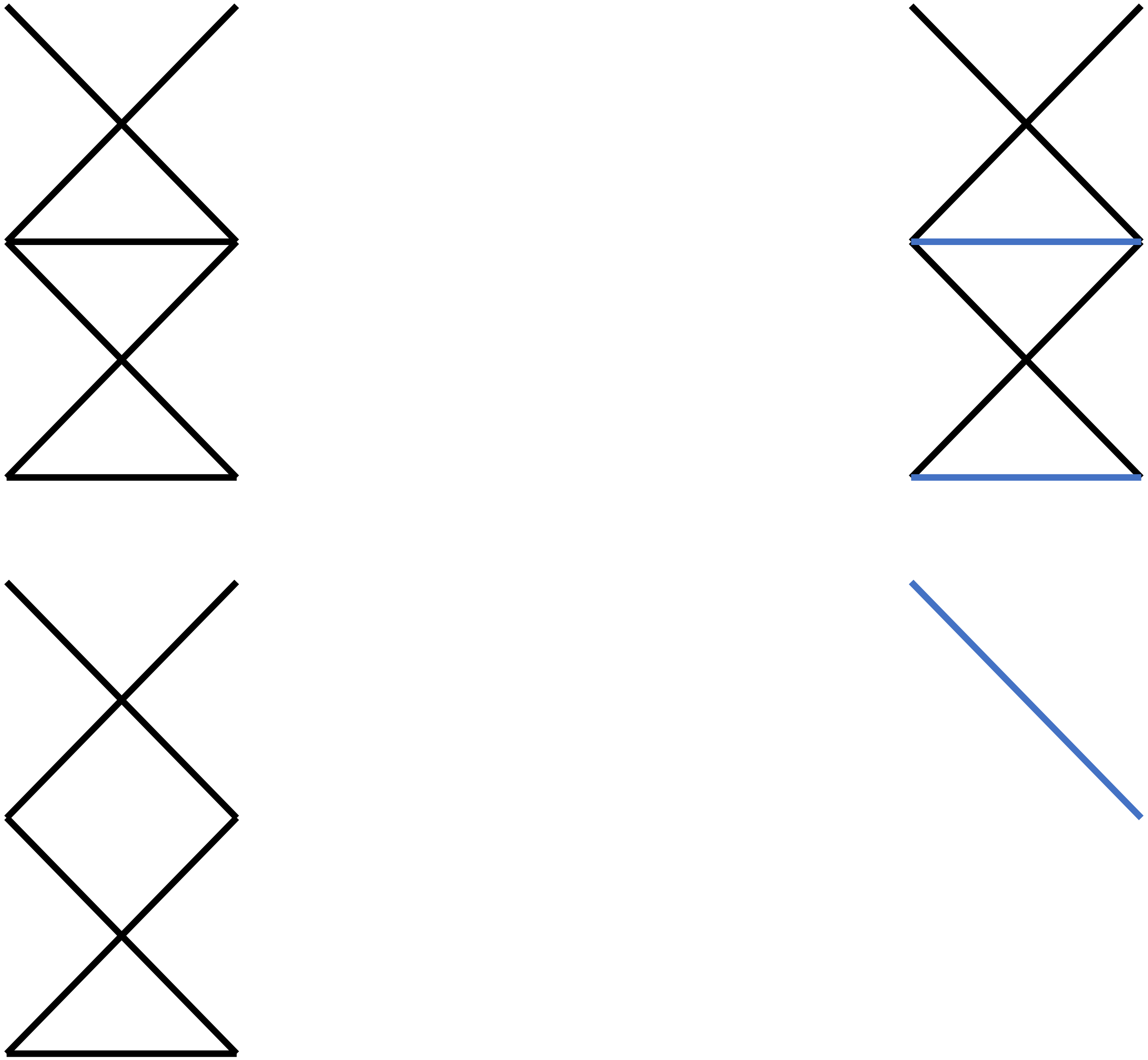


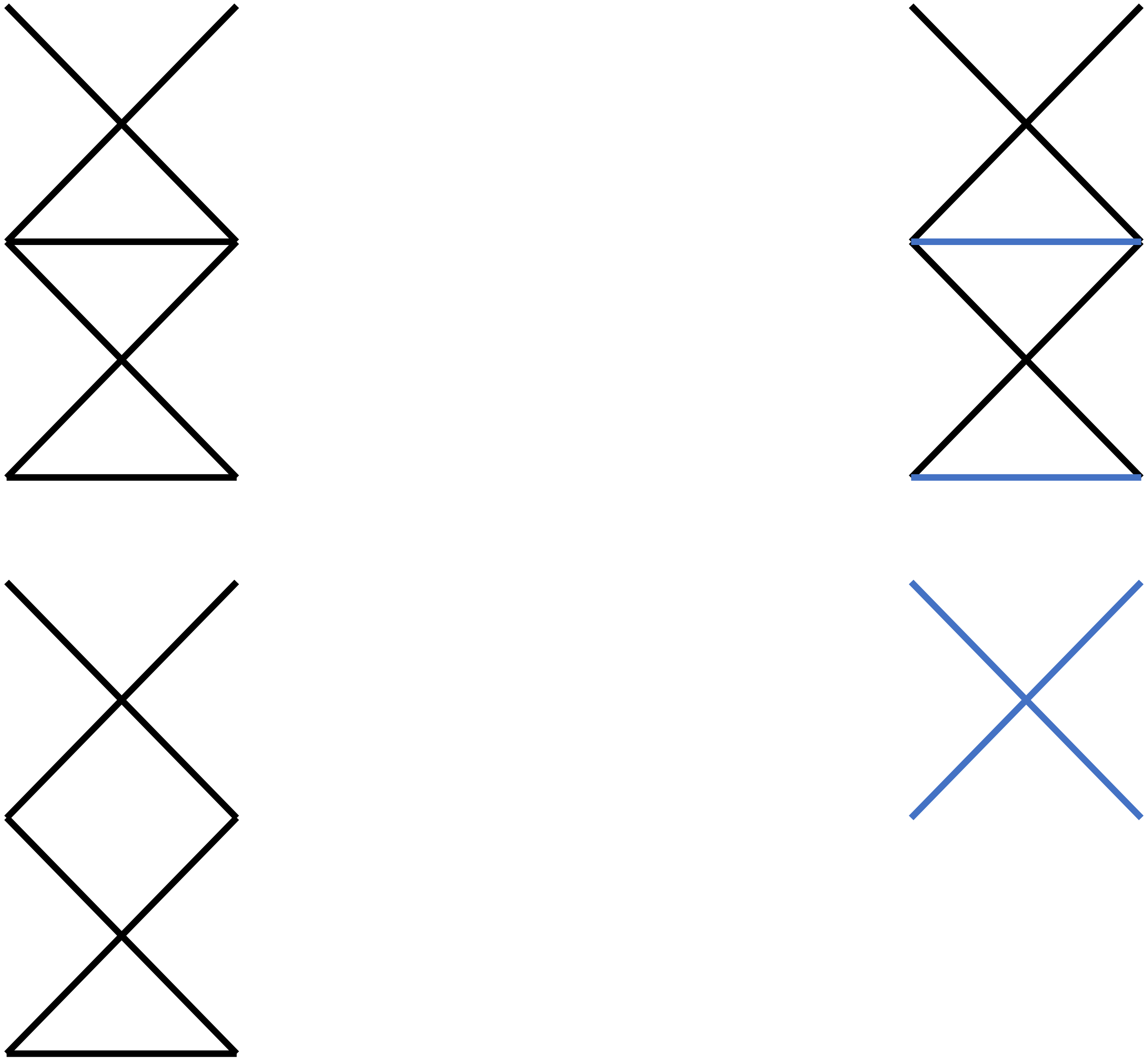


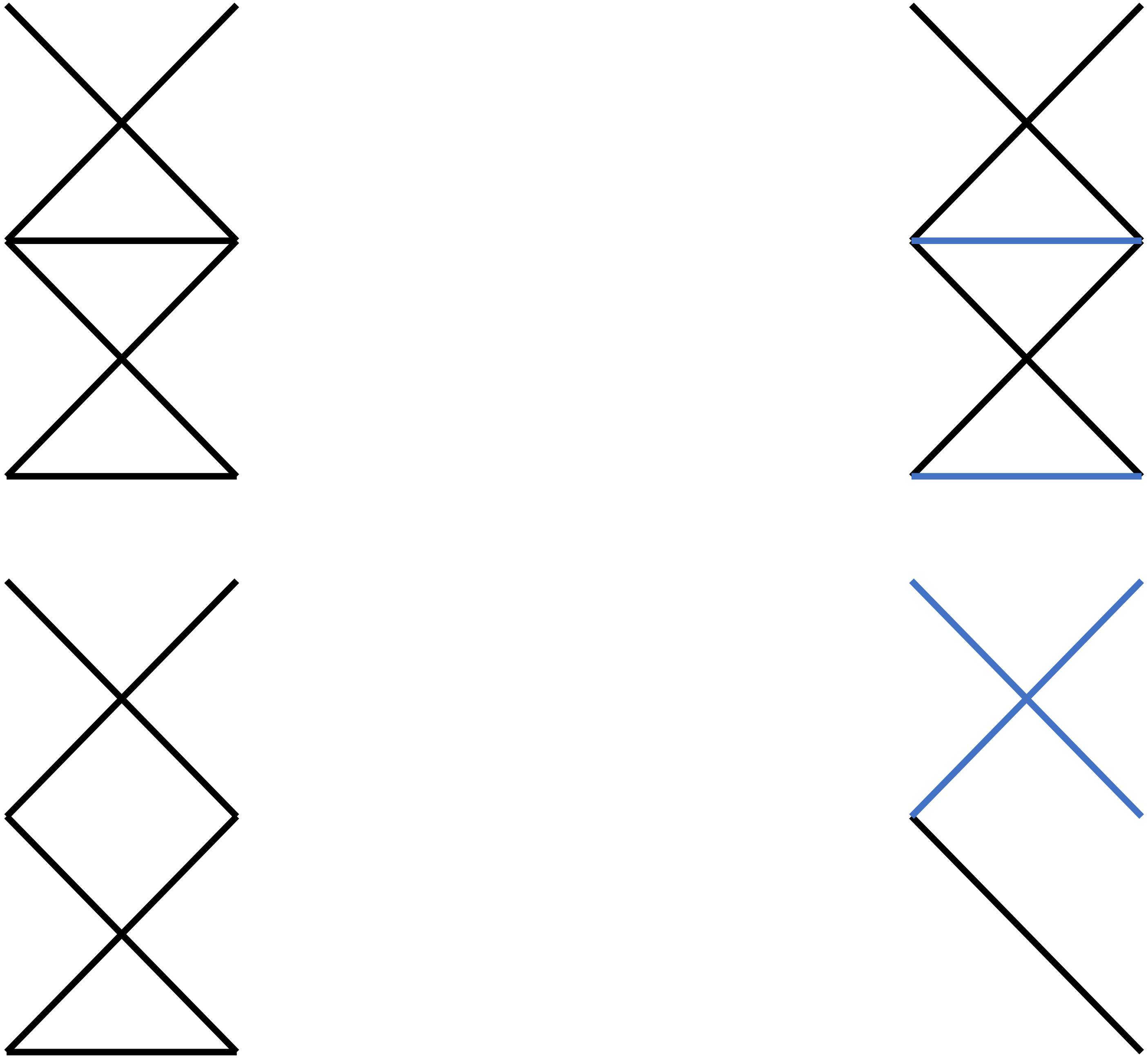


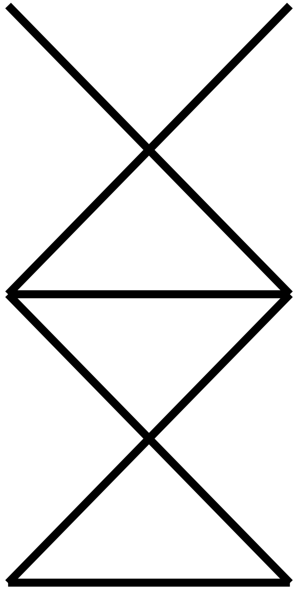


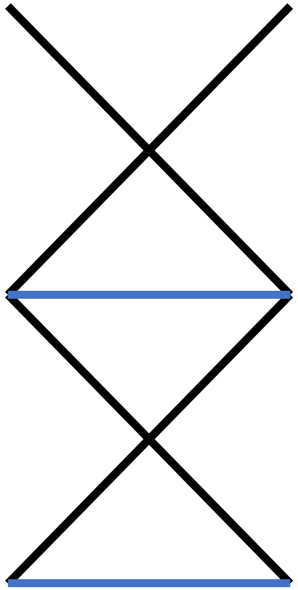


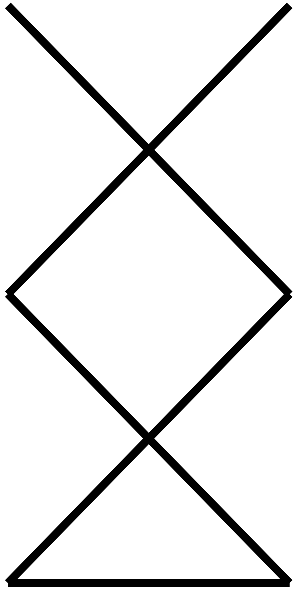


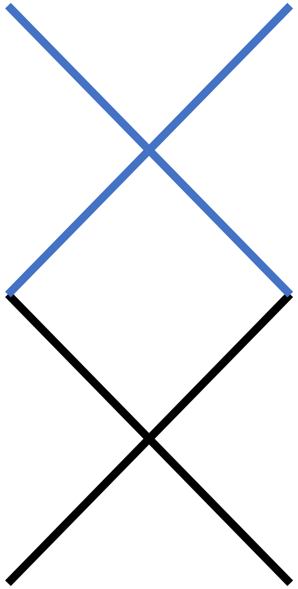


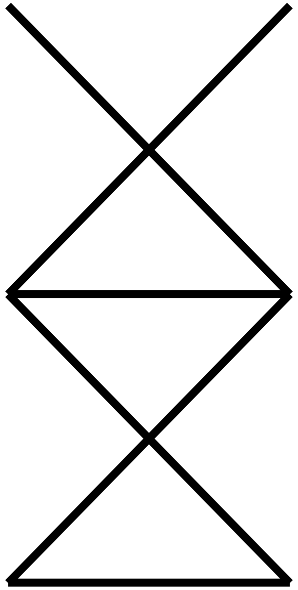
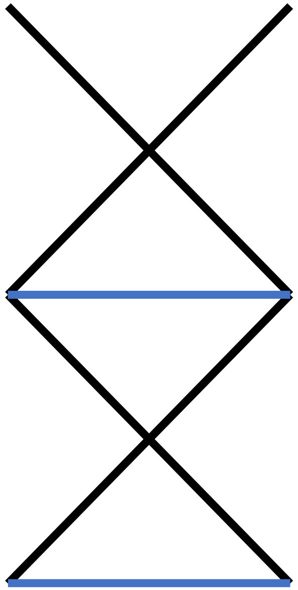
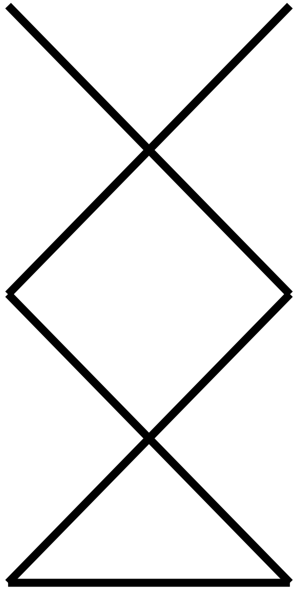
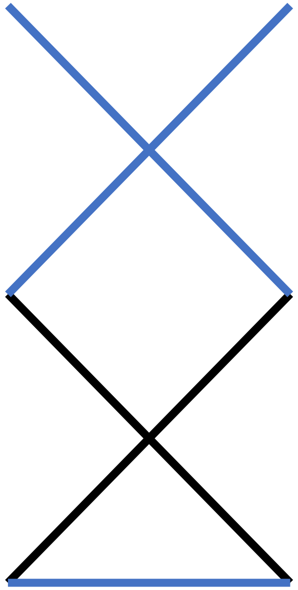


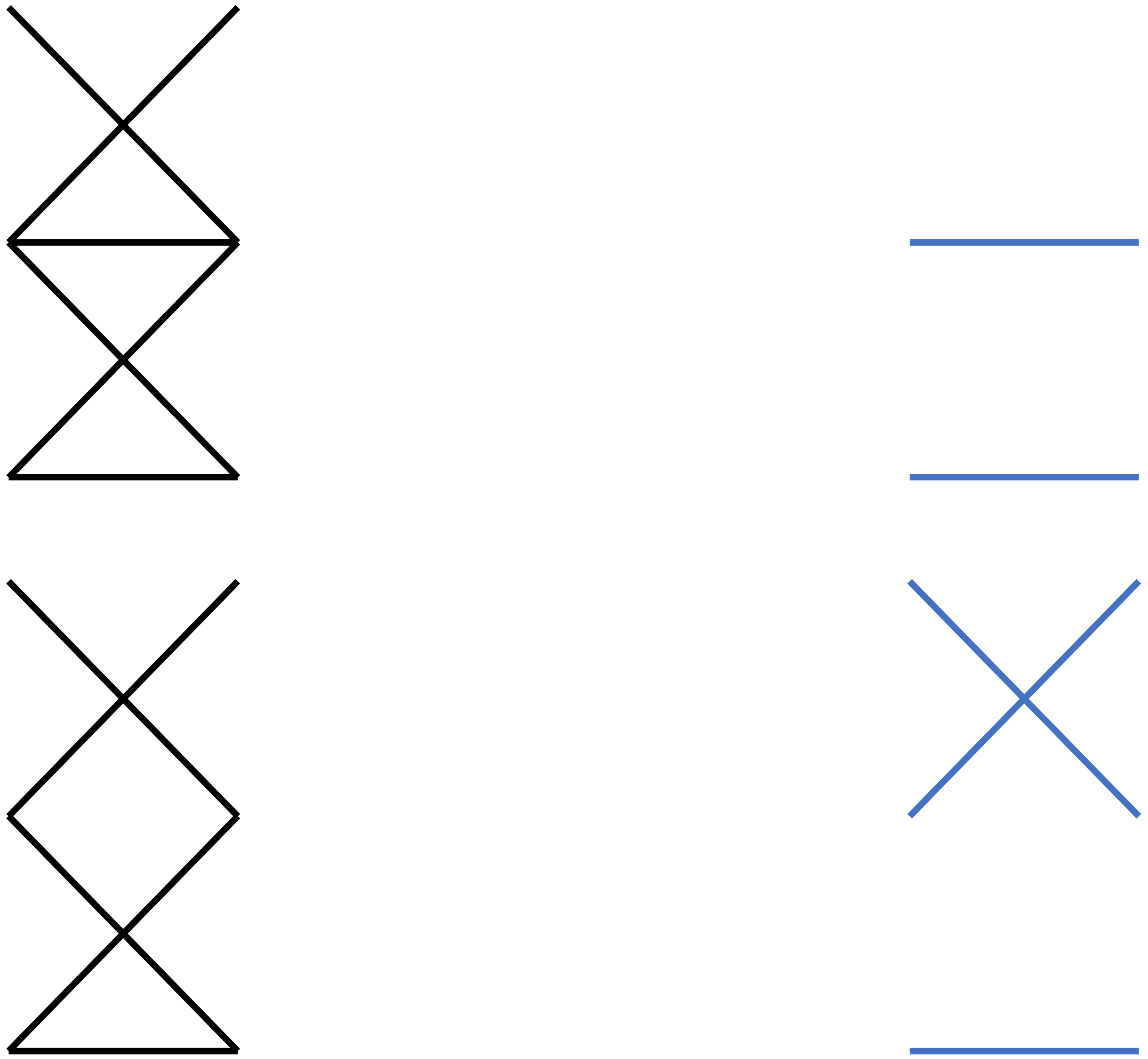


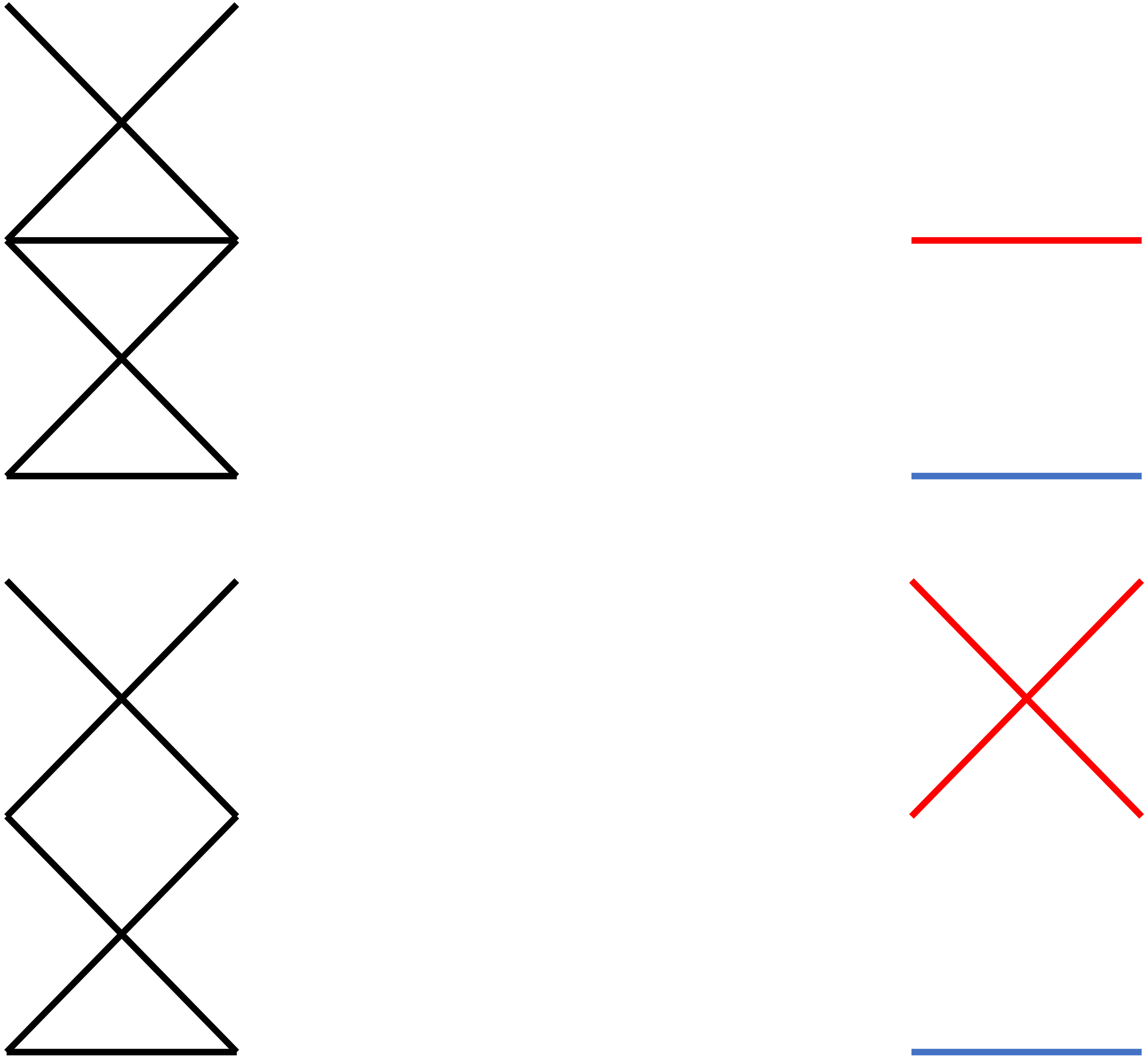










Average Sensitivity for Graph Algorithms

- Example: Greedy algorithm for maximum matching on graph with 6 nodes
- $|A(G) \triangle A(G - e)| = 3$

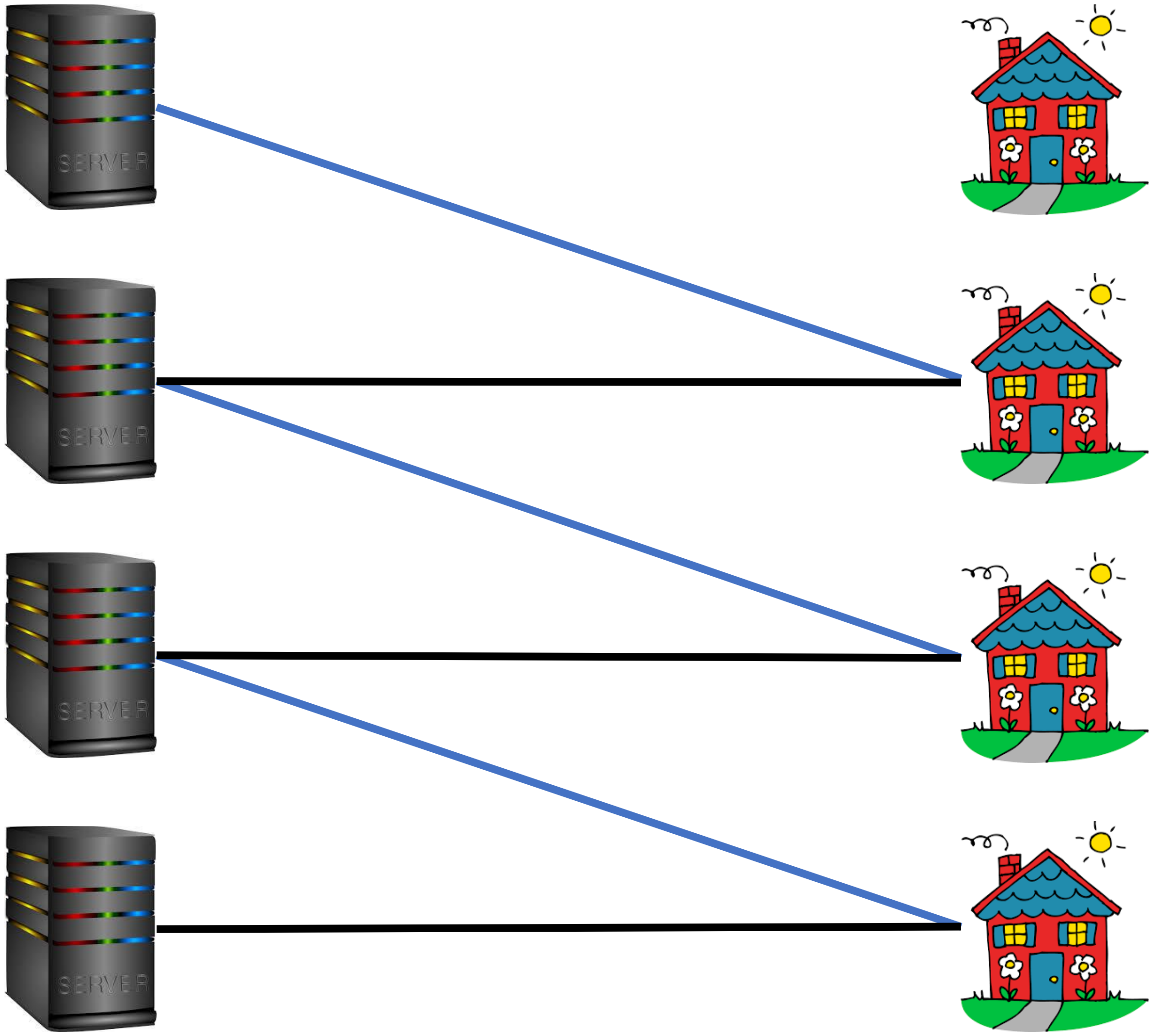
Average Sensitivity for Graph Algorithms

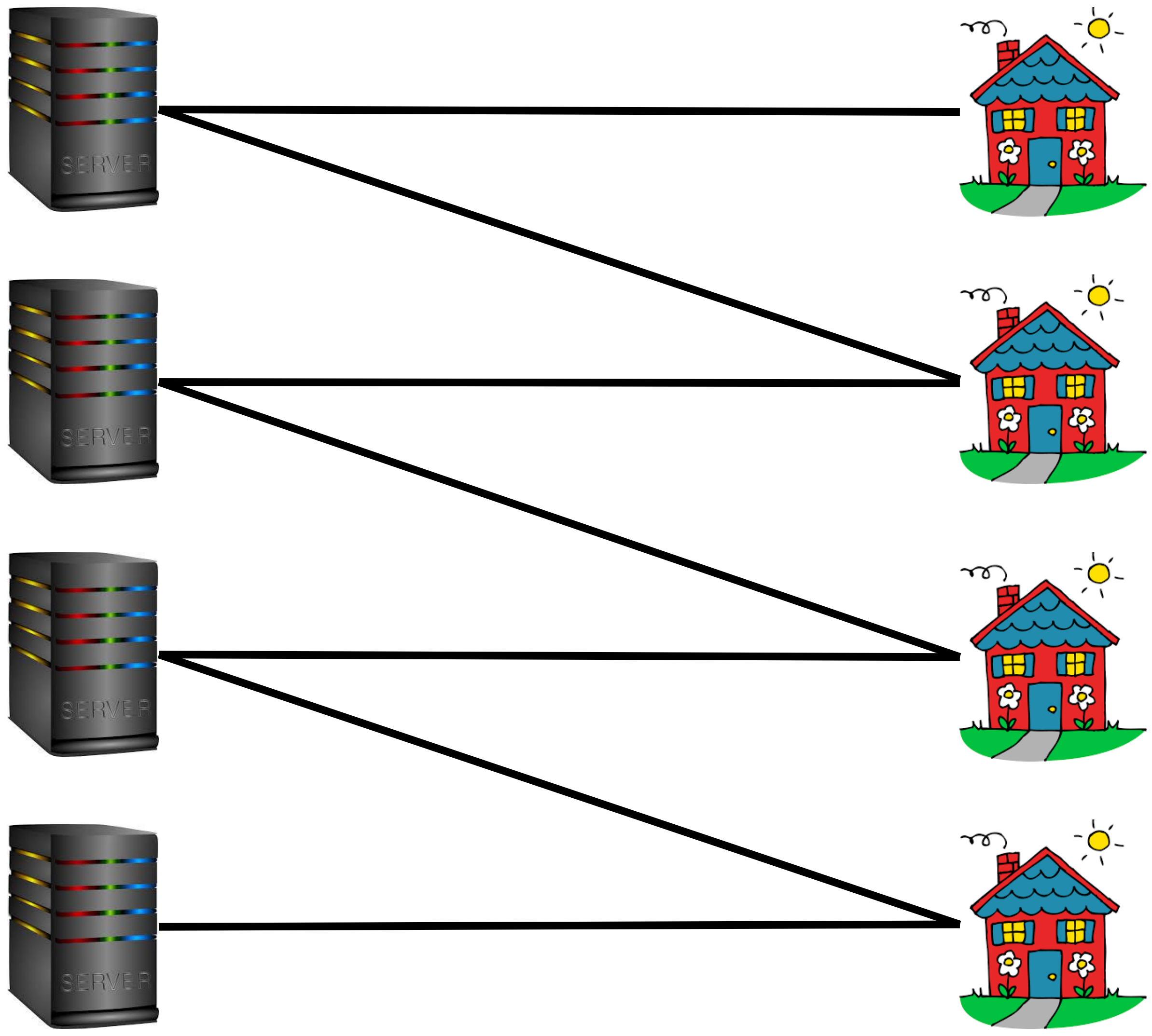
- Measure of how much a discrete algorithm changes when the input changes
- For a randomized graph algorithm A on a graph G with n vertices, the average sensitivity is defined as

$$\mathop{\mathrm{E}}_{e \sim E(G)}[d_{EMD}(A(G), A(G-e))]$$

where d_{EMD} is the Earth Mover Distance


SERVER
SERVER
SERVER
SERVER


SERVER
SERVER
SERVER
SERVER

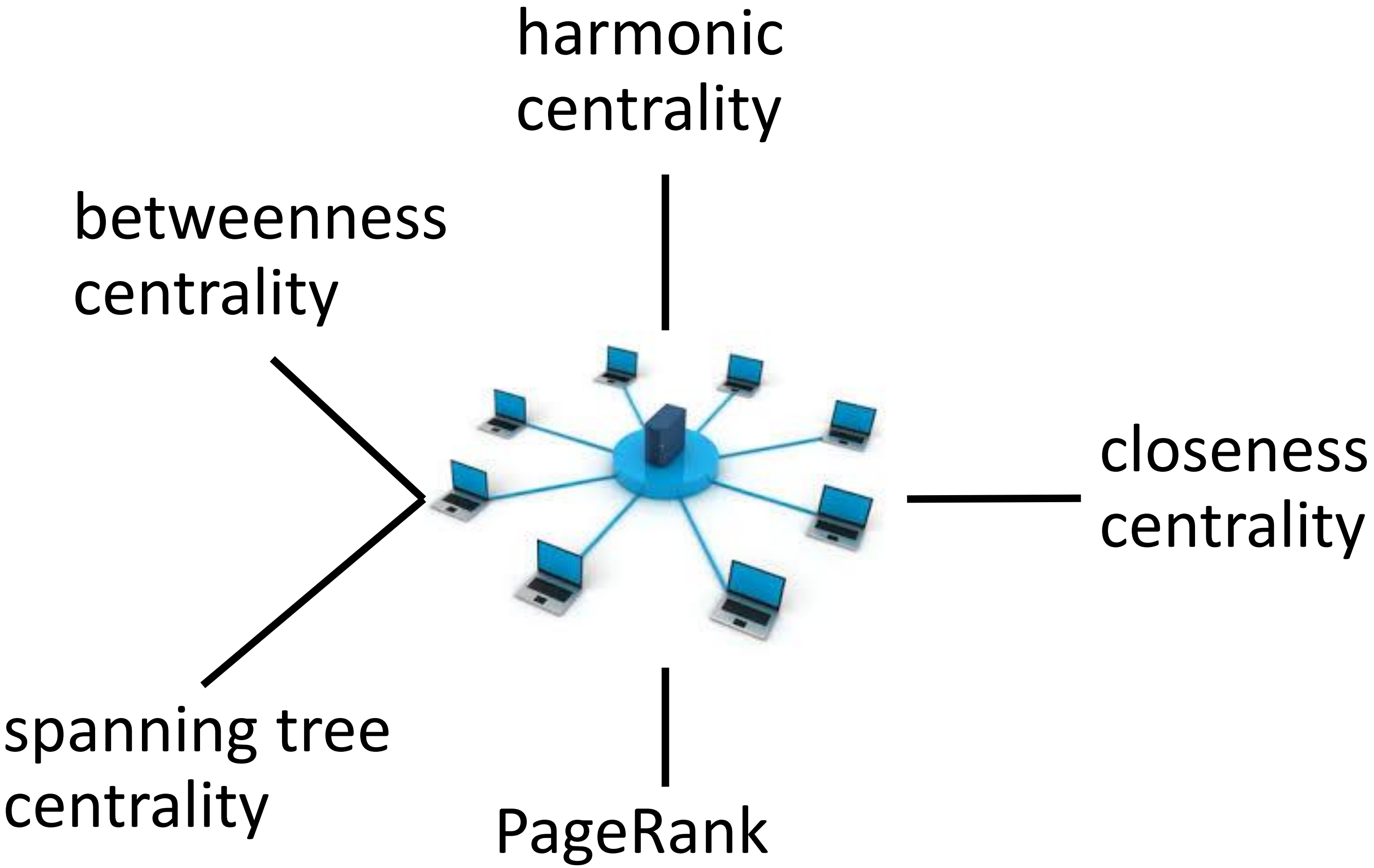

SERVER
SERVER
SERVER
SERVER


SERVER
SERVER
SERVER
SERVER


SERVER
SERVER
SERVER
SERVER

Average Sensitivity for Graph Algorithms

- Introduced by Murai and Yoshida [MY19] for centrality



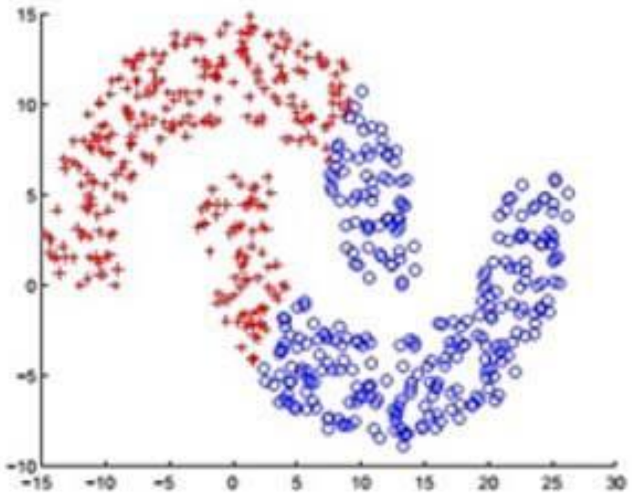
Average Sensitivity for Graph Algorithms

- Introduced by Murai and Yoshida [MY19] for centrality
- Varma and Yoshida [VY21] for graph algorithms that output edge/vertex sets

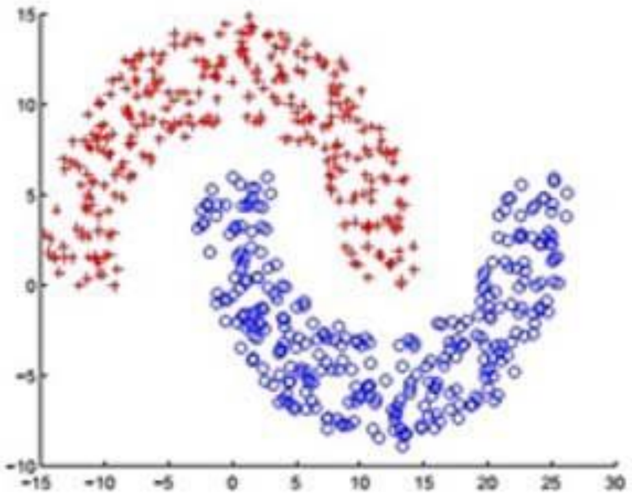
Problem	Output	Approximation Guarantee	Average Sensitivity
Minimum Spanning Forest	Edge set	1 $< \infty$	$O(\frac{n}{m})$ $\Omega(\frac{n}{m})$
Global Minimum Cut	Vertex set	$2+\epsilon$ 1 $< \infty$	$n^{O(\frac{1}{\epsilon \mathrm{OPT}})}$ $\Omega(n)$ $\Omega\left(\frac{n^{\frac{1}{\mathrm{OPT}}}}{\mathrm{OPT}^2}\right)$
Minimum s-t Cut	Vertex set	additive $O(n^{2/3})$	$O\left(n^{2/3}\right)$
Maximum Matching	Edge set	$1/2$ $1-\epsilon$ 1	1 $\tilde{O}\left(\left(\frac{\mathrm{OPT}}{\epsilon^3}\right)^{\frac{1}{1+\Omega(\epsilon^2)}}\right)$ $\Omega(n)$
Minimum Vertex Cover	Vertex set	2	2
2-Coloring	Vertex set	—	$\Omega(n)$

Average Sensitivity for Graph Algorithms

- Introduced by Murai and Yoshida [MY19] for centrality
- Varma and Yoshida [VY21] for graph algorithms that output edge/vertex sets
- Peng and Yoshida [PY20] for spectral clustering



(a) K-means



(b) Spectral Clustering

From Average Sensitivity to Worst-Case Sensitivity

- We introduce the definition of worst-case sensitivity
- For a randomized graph algorithm A on a graph G with n vertices, the average sensitivity is defined as

$$\mathop{\mathrm{E}}_{e \sim E(G)} [d_{EMD}(A(G), A(G-e))]$$

- For a randomized graph algorithm A on a graph G with n vertices, the worst-case sensitivity is defined as

$$\max_{e \sim E(G)} [d_{EMD}(A(G), A(G-e))]$$

Our Results (1)

- Theorem: $(1+\epsilon)$-approximation algorithm to the maximum matching problem with worst-case edge sensitivity $O_\epsilon(1)$.
- Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity Δ^x, where $x = O(6^\Delta + \log^* n)$, for graphs with vertex degree bounded by Δ.

Our Results (1)

- Theorem: $(1+\epsilon)$-approximation algorithm to the maximum matching problem with worst-case edge sensitivity $O_\epsilon(1)$.
- Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity Δ^x, where $x = O(6^\Delta + \log^* n)$, for graphs with vertex degree bounded by Δ.

Maximum Matching

- $(1+\epsilon)$-approximation algorithm for maximum matching with average sensitivity $O\left(\left(\frac{OPT}{\epsilon^3}\right)^{1/(1+\Omega(\epsilon^2))}\right)$ [VY21]

- What about worst-case sensitivity? Previous example shows greedy algorithm can have sensitivity $\Omega(n)$

Maximal Matching

- ❖ First idea: Randomized greedy for maximal matching
- ❖ The “expected” deleted/added edge appears halfway through the ordering of edges
- ❖ Intuition: Either most of the edges in the matching occur early in the ordering (dense graph) OR the deleted/added edge does not affect edges in the matching (sparse graph)

Randomized greedy for maximal matching

- Let $\pi(e)$ be the rank of e in a (fixed) random permutation
- $I_\pi(e)$:= Neighbors of e with smaller rank than e
- e is in the maximal matching M iff no edges of $I_\pi(e)$ are in M
- Consider the set S of edges that must be modified to maintain the invariant if e is moved to the beginning of π
- Unless $\pi(e) = \min \pi(S)$, then no edges need to be modified
- $\pi(e) = \min \pi(S)$ with probability $\frac{1}{|S|}$
- Expected number of modified edges $O(1)$

Randomized greedy for maximal matching

- Let π … e the rank of … (fixed) … permutation
- $I_{\pi}(e)$:= Ne…
- Consi… …intain the invariant i…
- Unless $\pi(e) =$ … odified
- $\pi(e) = \min \pi$ … with … ility $\frac{}{|S|}$
- Expected number of mo… ed edges $O(1)$

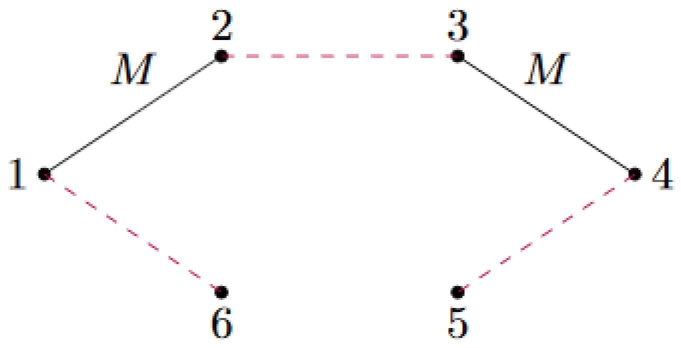
Similar idea used for maximal independent set in the dynamic distributed setting [CHK16]

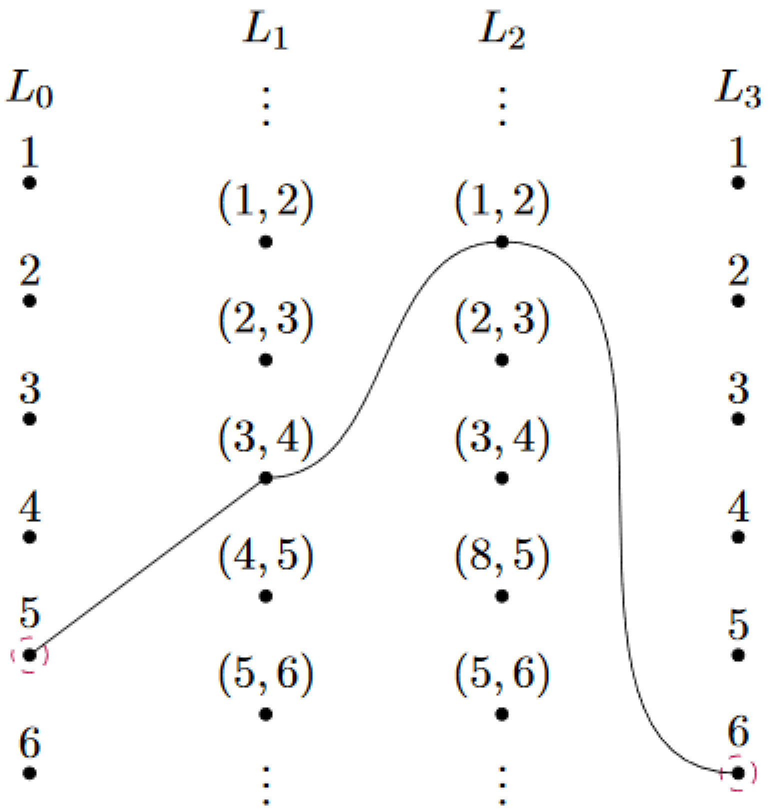
Step 1: Maximal Matching

- First idea: Randomized greedy for maximal matching with worst-case sensitivity $O(1)$
- From maximal matching to $(1+\epsilon)$-approximation algorithm for maximum matching?
- Modify the layered graph in the multi-pass streaming algorithm of McGregor [M05]

Step 2: Layered Graph

- Creates a graph with $\ell + 1$ layers to simultaneously sample a large number of augmenting paths of length ℓ





Step 2: Layered Graph

- Creates a graph with $\ell + 1$ layers to simultaneously sample a large number of augmenting paths of length ℓ
- Terminates in roughly $\frac{1}{\epsilon}$ iterations to achieve $(1+\epsilon)$-approximation algorithm for maximum matching
- Samples roughly $K = \left(\frac{1}{\epsilon}\right)^{2^{O\left(\frac{1}{\epsilon}\right)}}$ augmenting paths
- Worst-case sensitivity $O(3^K) = O_\epsilon(1)$
- Runtime: $O\big((n+m)K\big)$

Our Results (1)

- Theorem: $(1+\epsilon)$-approximation algorithm to the maximum matching problem with worst-case edge sensitivity $O_\epsilon(1)$.
- Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity Δ^{x}, where $x = O(6^\Delta + \log^* n)$, for graphs with vertex degree bounded by Δ.

Deterministic Algorithm

- Ingredient #1: Deterministic local computation algorithm (LCA) by Cole and Vishkin for 6^{Δ}-coloring of a graph with degree Δ using $O(\Delta \log^* n)$ queries [CV86]
- Partitions the graph into forests and assigns each node a color based on the LSB that differs between the node ID and its parent ID
- Fact #1: Deterministic LCA only queries vertices within distance $O(\log^* n)$

Deterministic Algorithm

- Ingredient #2: Framework by Parnas and Ron [PR07] that simulates local distributed algorithms using deterministic LCA using $\Delta^{O(c)}$ probes
- Iterate through colors and add each edge to maximal matching M if no adjacent edge is already in M
- Fact #2: Framework only queries vertices within distance $O(6^{\Delta})$

Deterministic Algorithm

- Fact #1: Deterministic LCA only queries vertices within distance $O(\log^* n)$
- Fact #2: Framework only queries vertices within distance $O(6^\Delta)$
- Sensitivity analysis: At most $O\left(\Delta^{6^\Delta + \log^* n}\right)$ queries in the LCA can be affected by a single deletion/addition

Our Results (1)

- Theorem: $(1+\epsilon)$-approximation algorithm to the maximum matching problem with worst-case edge sensitivity $O_\epsilon(1)$.
- Theorem: Deterministic algorithm for maximal matching with worst-case sensitivity Δ^x, where $x = O(6^\Delta + \log^* n)$, for graphs with vertex degree bounded by Δ.

Our Results (2)

- Theorem: Any deterministic constant-factor approximation algorithm for the maximum matching problem has worst-case edge sensitivity $\Omega(\log^* n)$.
- Theorem: For a graph with edge weights bounded by $W = poly(n)$ and a trade-off parameter α, there exists an algorithm that outputs a 4α-approximation to the maximum weighted matching in $O(m \log_\alpha n)$ time. For $\alpha = 2$, the algorithm has weighted sensitivity $O(W \log n)$ and normalized weighted sensitivity $O(\log n)$. For $\alpha > 2$, the algorithm has weighted sensitivity $O(W)$ and normalized weighted sensitivity $O(1)$.

Future Directions

- Worst-case analysis of monotone submodular maximization by McMeel and Yoshida [MY20]
- Worst-case (normalized) weighted sensitivity for other graph problems?
- Constant factor approximation algorithm for maximum weighted matching with “low” worst-case sensitivity?
- Can we get $(1 + \epsilon)$-approximation algorithm for maximum weighted matching with “low” worst-case (normalized) weighted sensitivity?


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
감사합니다
ευχαριστώ
arigatô
dakujem
grazie
takk
tapadh leat
dank je
спасибо
рахмат
Баярлалаа
mersi
kia ora
barka
welalin
tack
spas
mahalo
faafetai lava
vinaka
спасибі
blagodaram
misaotra
matondo
paldies
grazzi
хвала
kiitos
dankie
nanni
nandri
dhanyavad
maururu
hvala
köszönöm
gràçie
bayarlalaa
enkosi
dankon
aćiū
akun
asante
manana
obrigada
tenki
murakoze
djiere dieuf
tau
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
sobodi
děkuji
sagolun
mèsi
ありがとう
tanemirt
trugarez
najis tuke
didi madloba
kam sah hamnida
rahmat
rahmet
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
xièxie