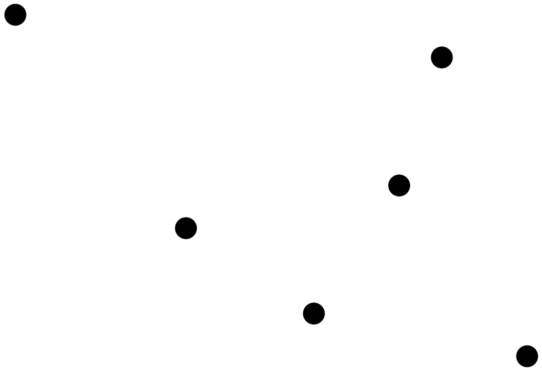
A Simple 3-Approximation of Minimum Manhattan Networks

Bernhard Fuchs and Anna Schulze

TU Braunschweig and Uni Köln

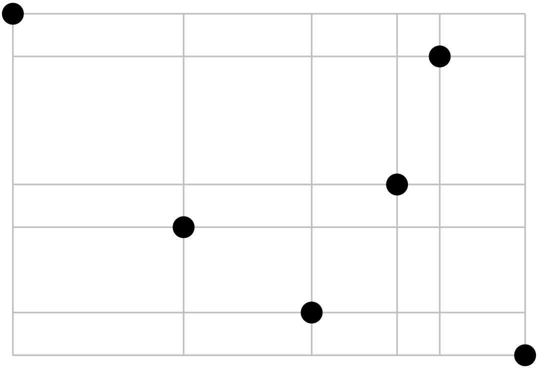
CTW 2008

Manhattan networks



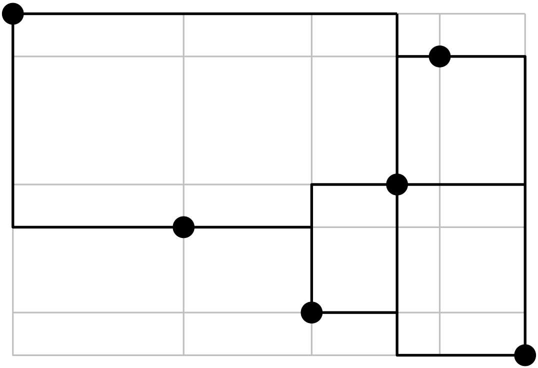
- Given a set of points in the plane,

Manhattan networks



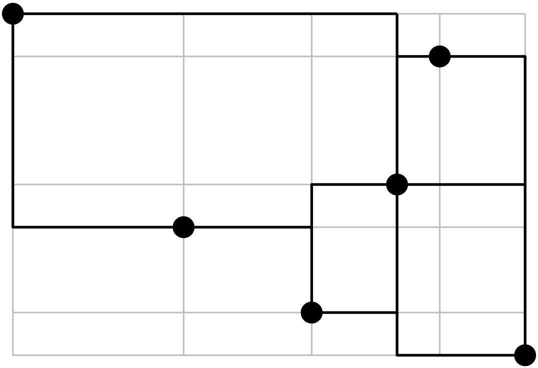
- Given a set of points in the plane,
- a **Manhattan network** contains all pairwise shortest rectilinear paths.

Manhattan networks



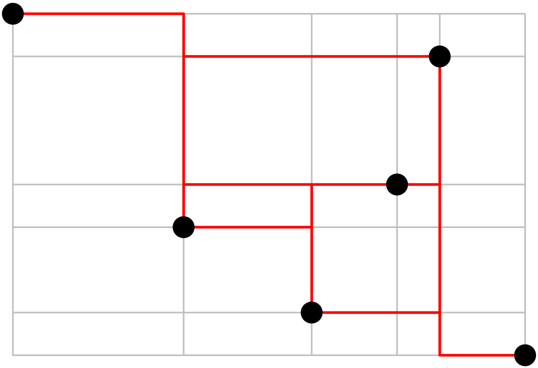
- Given a set of points in the plane,
- a **Manhattan network** contains all pairwise shortest rectilinear paths.

Manhattan networks



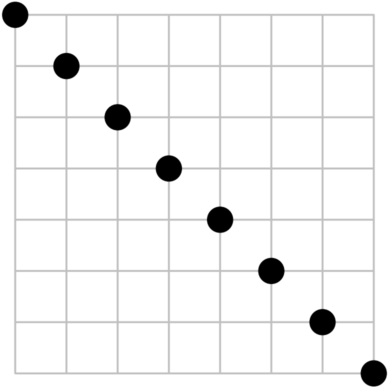
- Given a set of points in the plane,
- a **Manhattan network** contains all pairwise shortest rectilinear paths.
- Task: Find a **Minimum** **Manhattan Network** (MMN)!

Manhattan networks



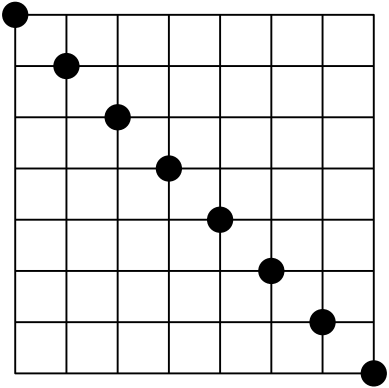
- Given a set of points in the plane,
- a **Manhattan network** contains all pairwise shortest rectilinear paths.
- Task: Find a **Minimum Manhattan Network** (MMN)!

Simple heuristics – Hannan grid



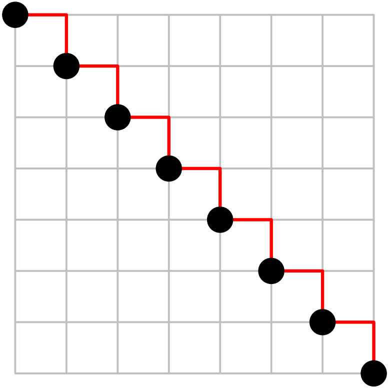
- First heuristic: Take the whole Hannan grid, length $\Omega(n^2)$.

Simple heuristics – Hannan grid



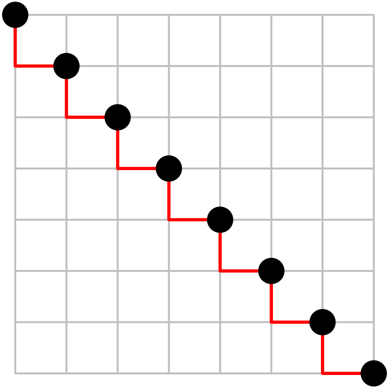
- First heuristic: Take the whole Hannan grid, length $\Omega(n^2)$.

Simple heuristics – Hannan grid



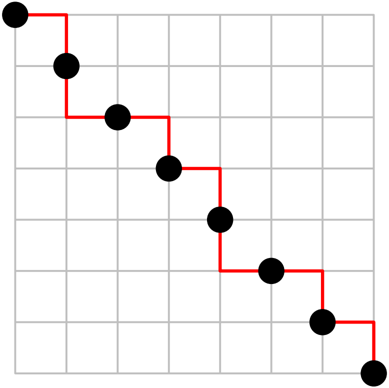
- First heuristic: Take the whole Hannan grid, length $\Omega(n^2)$.
- MMN has length $O(n)$.

Simple heuristics – Hannan grid



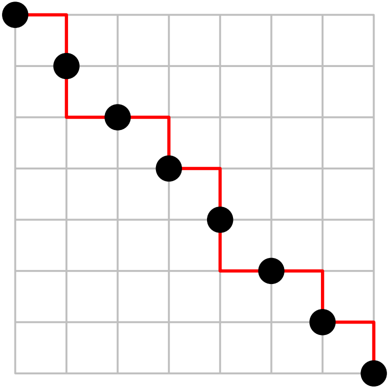
- First heuristic: Take the whole Hannan grid, length $\Omega(n^2)$.
- MMN has length $O(n)$.

Simple heuristics – Hannan grid



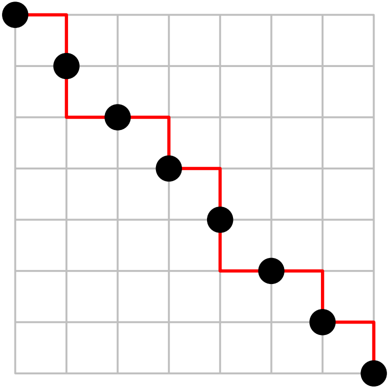
- First heuristic: Take the whole Hannan grid, length $\Omega(n^2)$.
- MMN has length $O(n)$.

Simple heuristics – Hannan grid



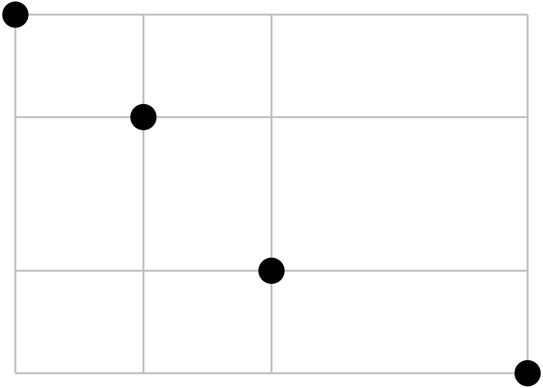
- First heuristic: Take the whole Hannan grid, length $\Omega(n^2)$.
- MMN has length $O(n)$.
- No constant factor approximation.

Simple heuristics – Hannan grid



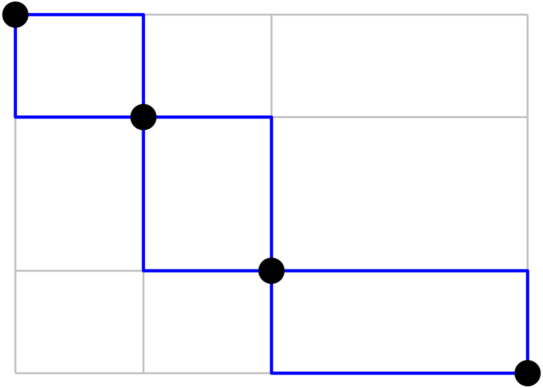
- First heuristic: Take the whole Hannan grid, length $\Omega(n^2)$.
- MMN has length $O(n)$.
- No constant factor approximation.
- Next idea: Just insert **critical rectangles**!

Critical rectangles



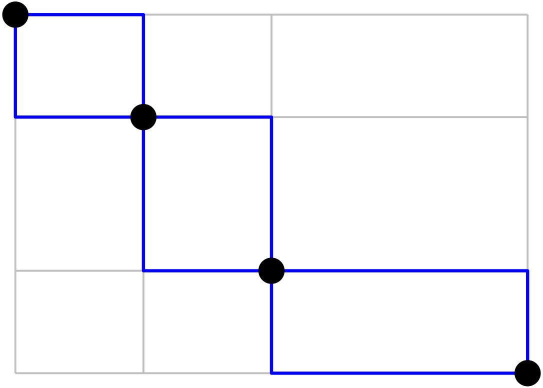
- A **critical rectangle** contains *exactly* the two points spanning it.

Critical rectangles



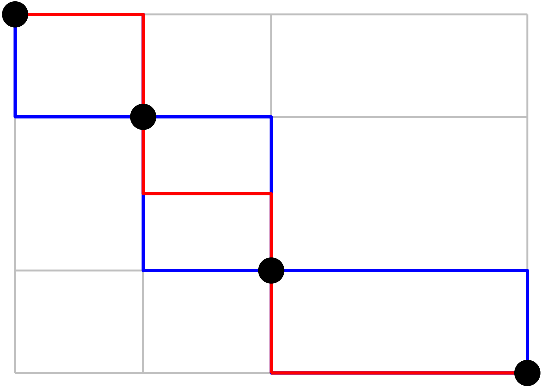
- A **critical rectangle** contains *exactly* the two points spanning it.

Critical rectangles



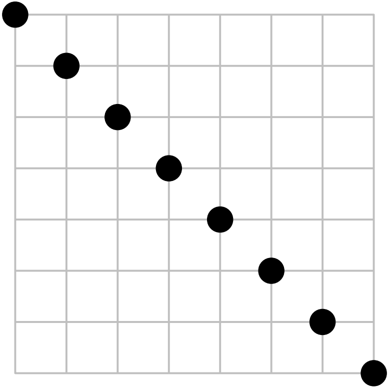
- A **critical rectangle** contains *exactly* the two points spanning it.
- It obviously suffices to consider critical rectangles.

Critical rectangles



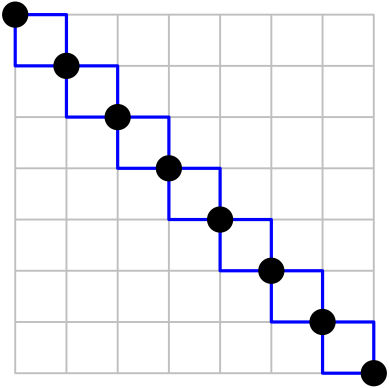
- A **critical rectangle** contains *exactly* the two points spanning it.
- It obviously suffices to consider critical rectangles.
- Note: Each Manhattan network has to cross such a rectangle!

Critical rectangles



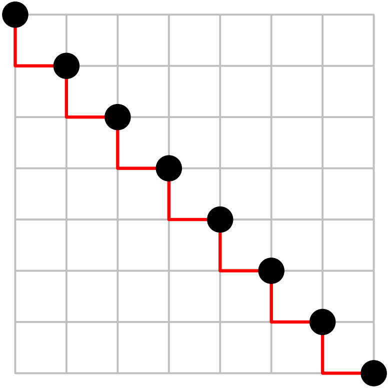
- A **critical rectangle** contains *exactly* the two points spanning it.
- It obviously suffices to consider critical rectangles.
- Note: Each Manhattan network has to cross such a rectangle!

Critical rectangles



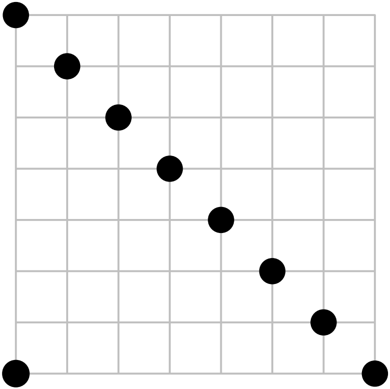
- A **critical rectangle** contains *exactly* the two points spanning it.
- It obviously suffices to consider critical rectangles.
- Note: Each Manhattan network has to cross such a rectangle!

Critical rectangles



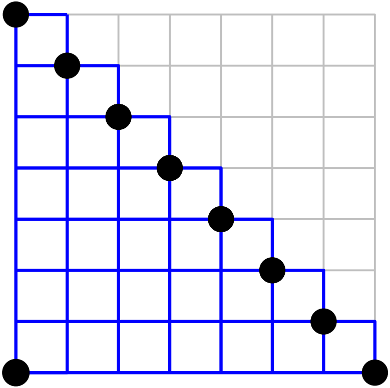
- A **critical rectangle** contains *exactly* the two points spanning it.
- It obviously suffices to consider critical rectangles.
- Note: Each Manhattan network has to cross such a rectangle!

Too many critical rectangles



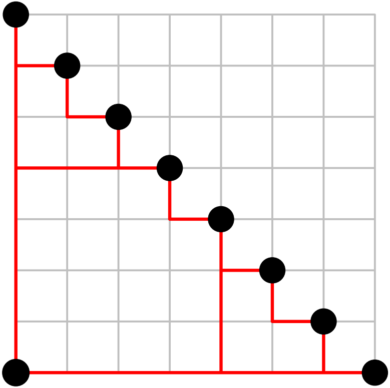
- Add one more point.

Too many critical rectangles



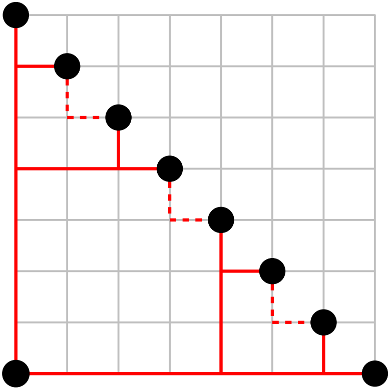
- Add one more point.
- Many more critical rectangles, total length $\Omega(n^2)$.

Too many critical rectangles



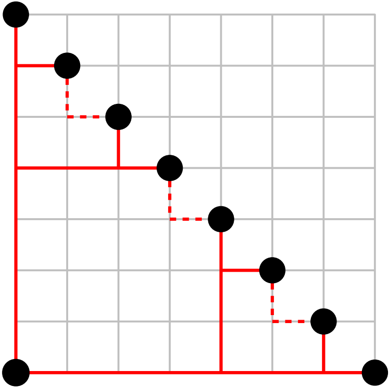
- Add one more point.
- Many more critical rectangles, total length $\Omega(n^2)$.
- MMN is basically binary tree, length $O(n \log n)$.

Too many critical rectangles



- Add one more point.
- Many more critical rectangles, total length $\Omega(n^2)$.
- MMN is basically binary tree, length $O(n \log n)$.

Too many critical rectangles



- Add one more point.
- Many more critical rectangles, total length $\Omega(n^2)$.
- MMN is basically binary tree, length $O(n \log n)$.
- Still no constant factor approximation.

Prior work

- Complexity of MMN: Still unknown!

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$
		4-approx.	in $O(n \log n)$

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$
		4-approx.	in $O(n \log n)$
[Kato et al.]	02	2-approx.	in $O(n^3)$

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	
[Seibert, Unger]	05	1.5-approx.	in $O(n^3)$	

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	
[Seibert, Unger]	05	1.5-approx.	in $O(n^3)$	incomplete

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	
[Seibert, Unger]	05	1.5-approx.	in $O(n^3)$	incomplete
[Nouioua]	—	2-approx.	in $O(n \log n)$	

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	
[Seibert, Unger]	05	1.5-approx.	in $O(n^3)$	incomplete
[Nouioua]	—	2-approx.	in $O(n \log n)$	unpublished

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	
[Seibert, Unger]	05	1.5-approx.	in $O(n^3)$	incomplete
[Nouioua]	—	2-approx.	in $O(n \log n)$	unpublished

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	
[Seibert, Unger]	05	1.5-approx.	in $O(n^3)$	incomplete
[Nouioua]	—	2-approx.	in $O(n \log n)$	unpublished

- Our algorithm: A new 3-approximation in $O(n \log n)$.

Prior work

- Complexity of MMN: Still unknown!
- Approximation algorithms so far:

[Gudmunsson et al.]	99	8-approx.	in $O(n^3)$	
		4-approx.	in $O(n \log n)$	
[Kato et al.]	02	2-approx.	in $O(n^3)$	incomplete
[Benkert et al.]	04	3-approx.	in $O(n \log n)$	
[Chepoi et al.]	05	2-approx.	via LP-rounding	
[Seibert, Unger]	05	1.5-approx.	in $O(n^3)$	incomplete
[Nouioua]	—	2-approx.	in $O(n \log n)$	unpublished

- Our algorithm: A new 3-approximation in $O(n \log n)$.
- Much simpler than [Benkert et al.], both algorithm and proof.

Algorithm outline

Our algorithm has two phases:

Algorithm outline

Our algorithm has two phases:

Phase I

A horizontal and a vertical sweep adding line segments 'on-the-fly'.

Algorithm outline

Our algorithm has two phases:

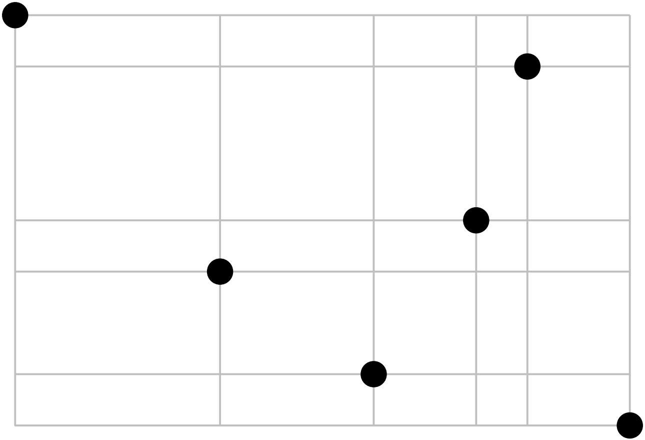
Phase I

A horizontal and a vertical sweep adding line segments 'on-the-fly'.

Phase II

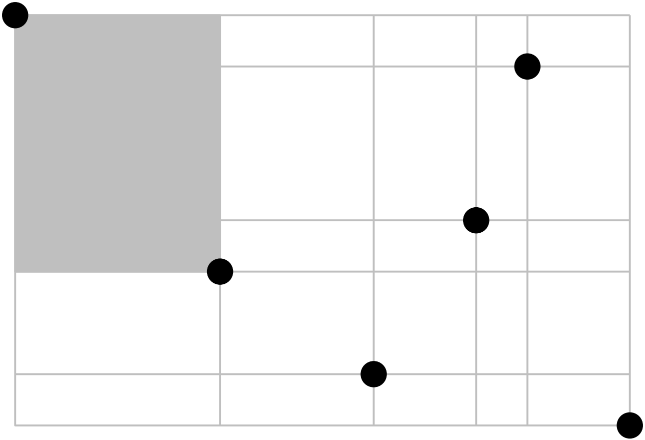
A standard 2-approximation algorithm inside so-called '*staircases*'.

Phase I - The sweep



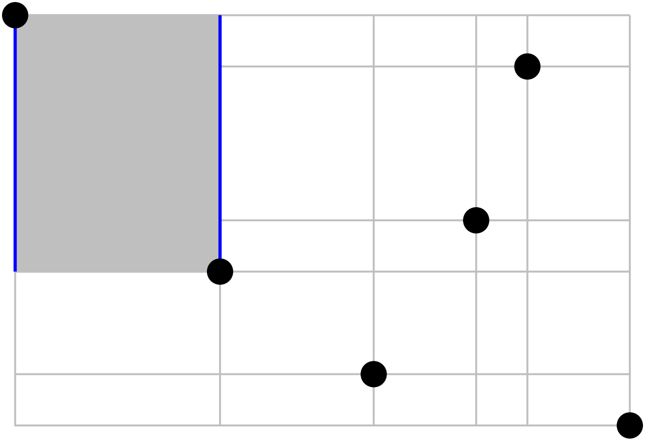
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.

Phase I - The sweep



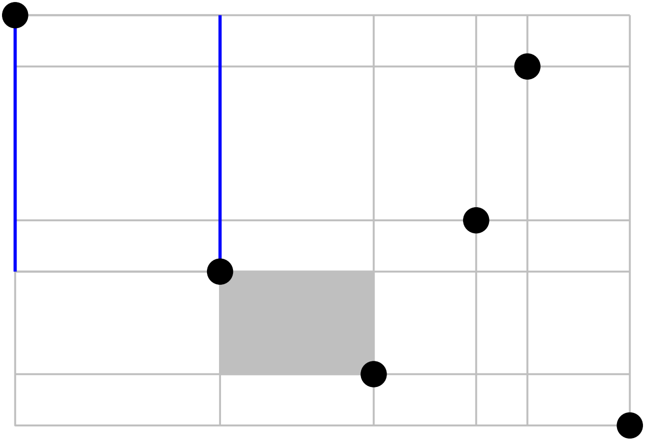
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.

Phase I - The sweep



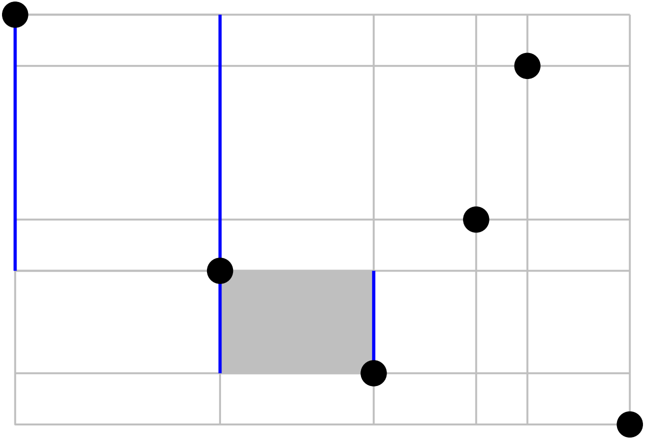
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.

Phase I - The sweep



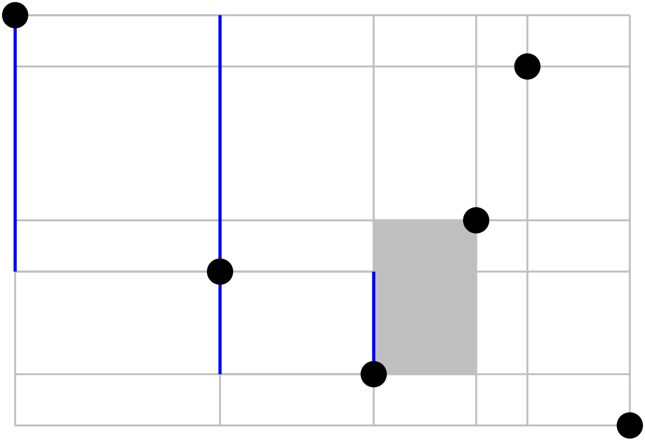
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



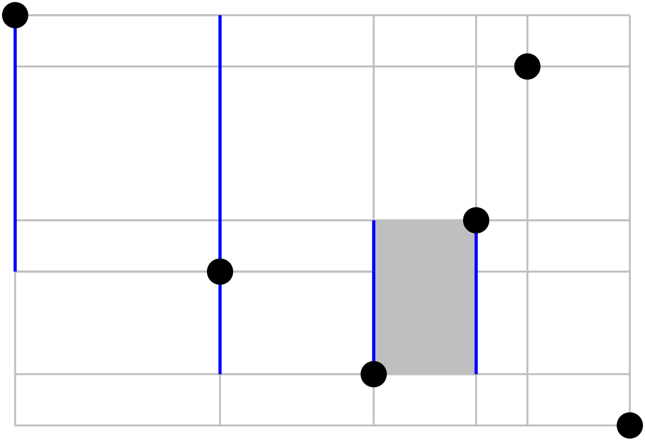
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



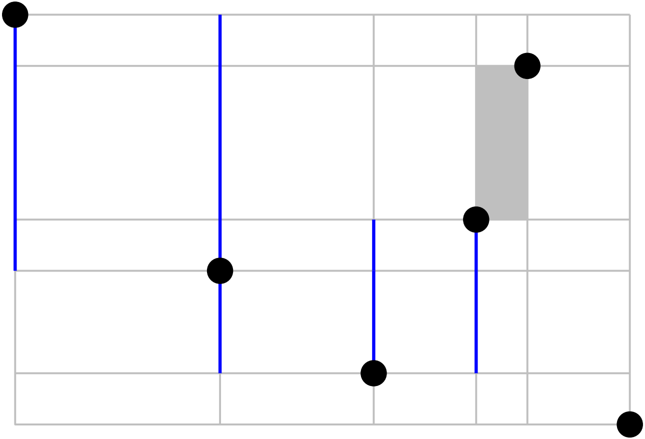
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



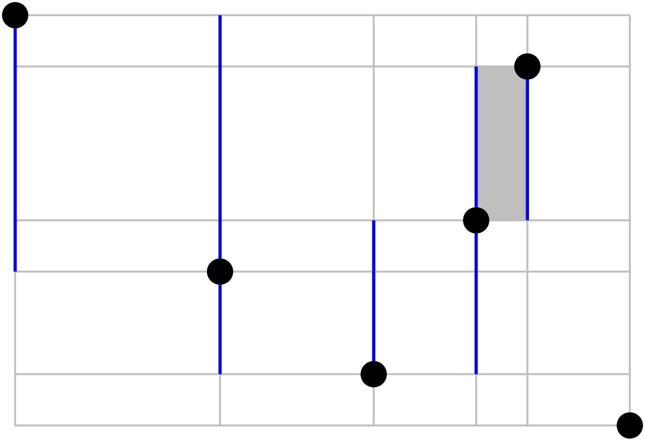
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



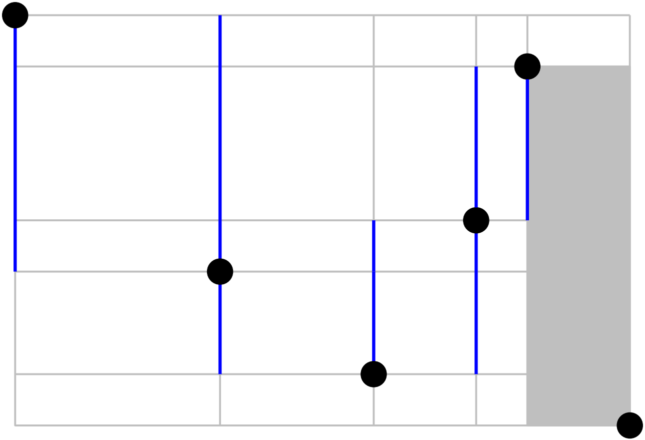
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



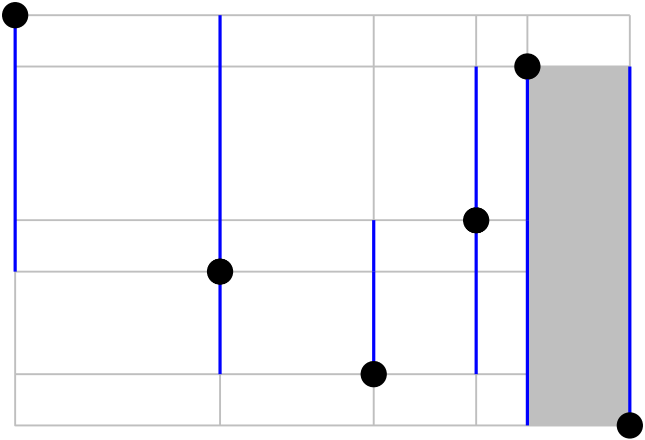
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



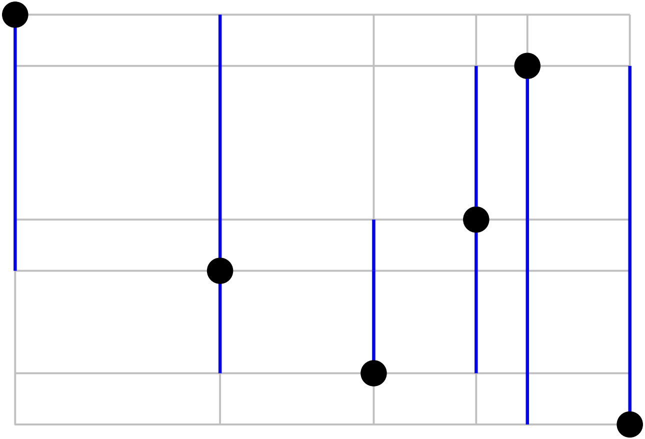
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



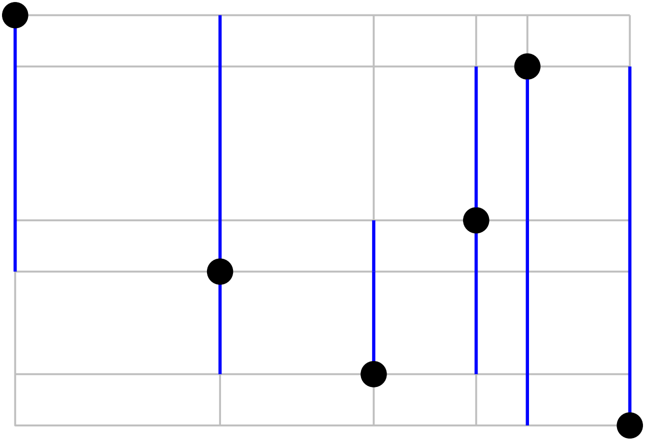
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



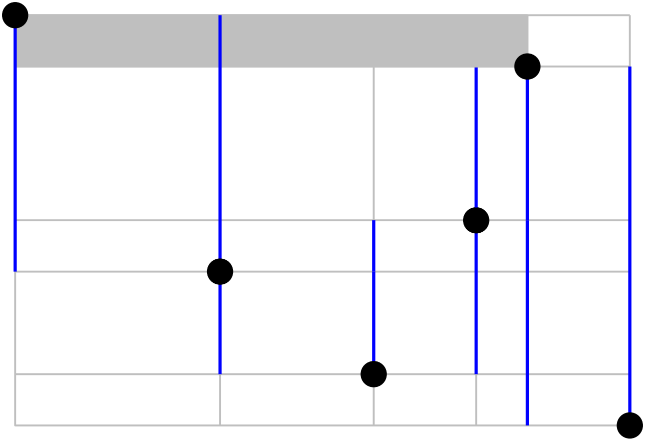
- Consider critical rectangles spanned by *horizontal*, or *x-neighbors*.
- Add vertical sides of rectangle.
- Iterate through rectangles from left to right.

Phase I - The sweep



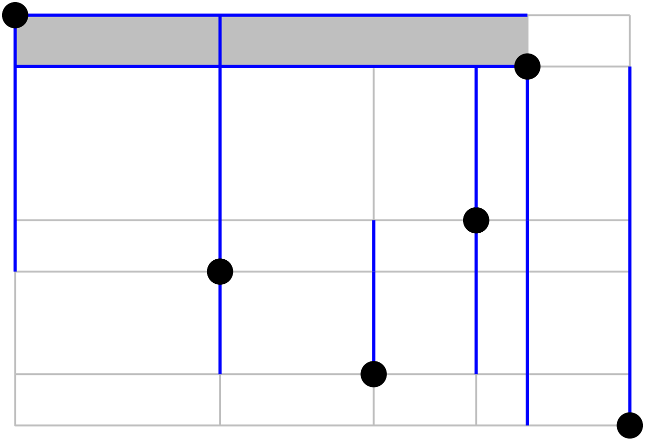
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



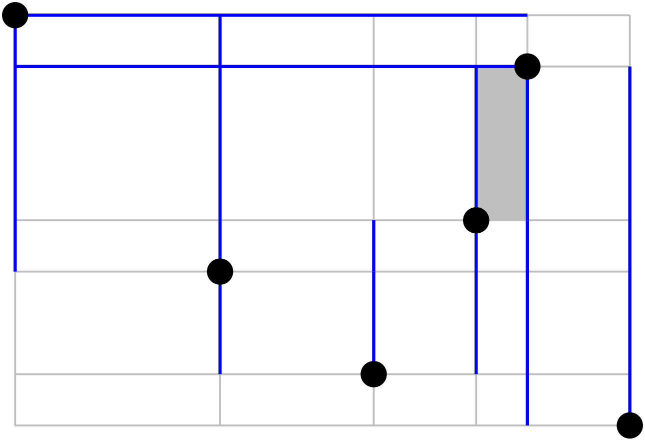
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



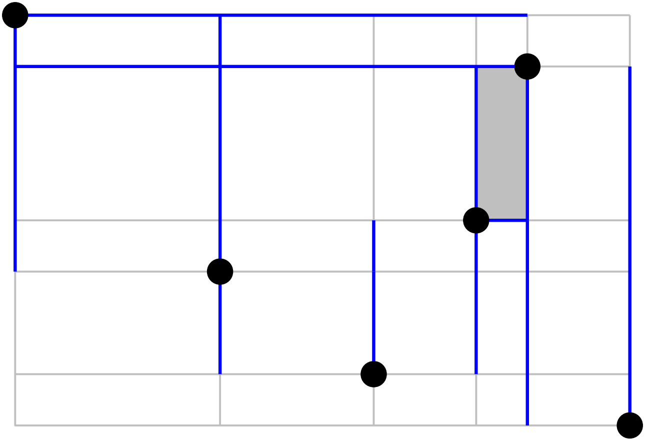
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



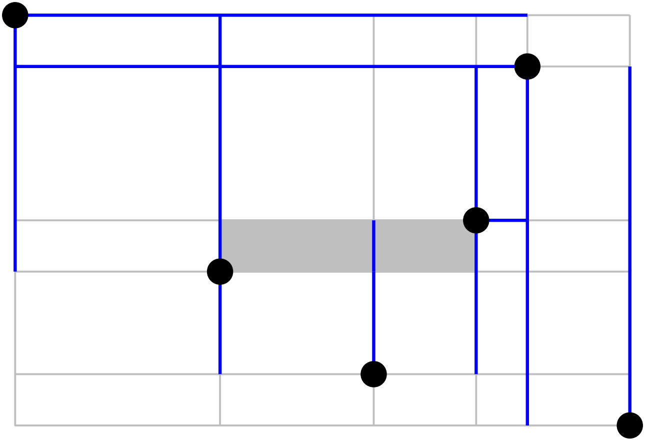
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



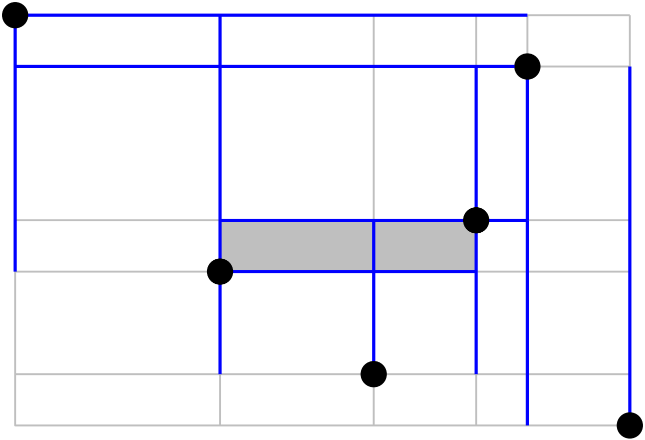
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



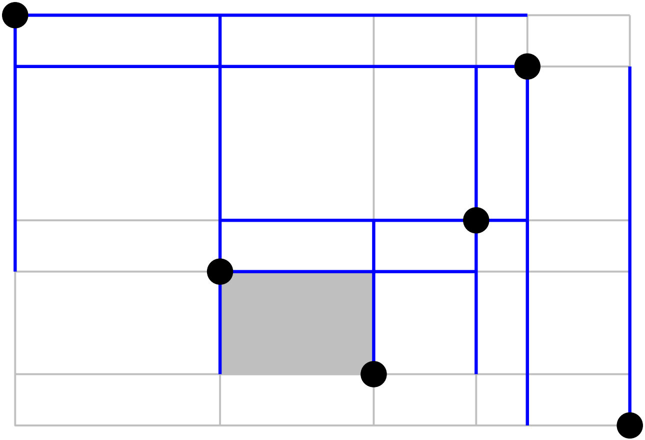
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



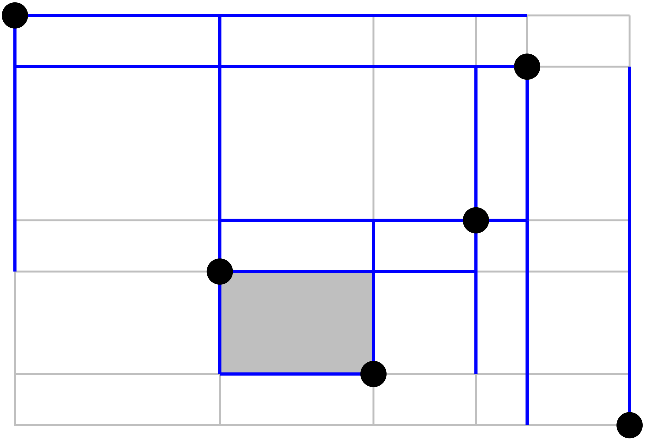
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



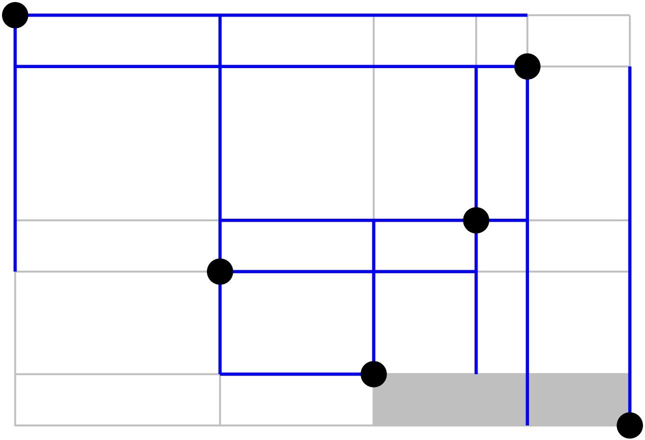
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



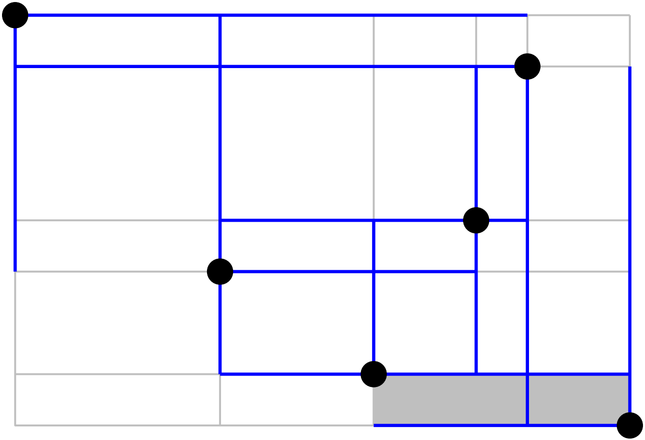
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



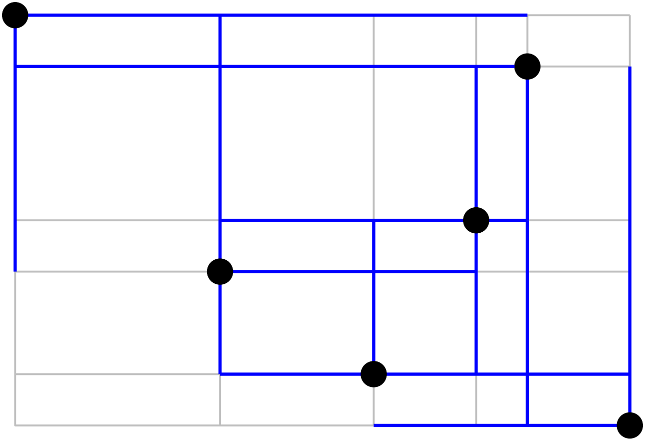
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



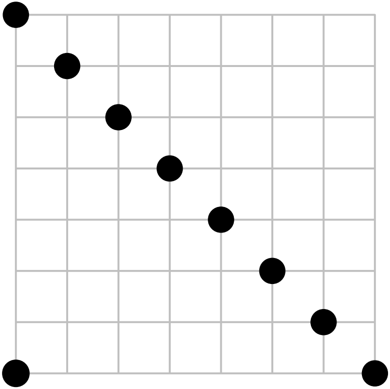
- Proceed likewise with *y-neighbors*.

Phase I - The sweep



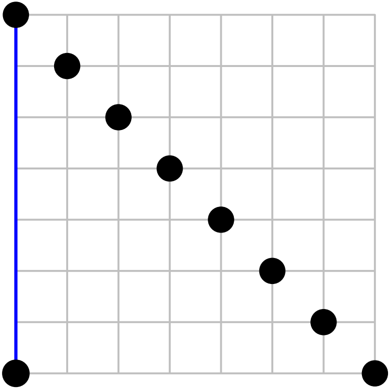
- Proceed likewise with *y-neighbors*.

After the sweep



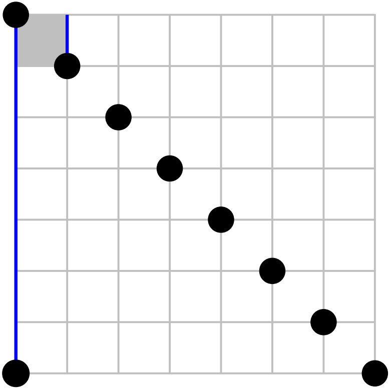
- In general, no Manhattan network after sweep.

After the sweep



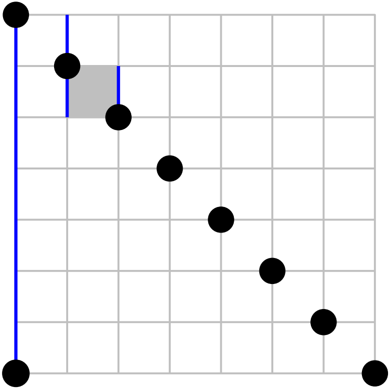
- In general, no Manhattan network after sweep.

After the sweep



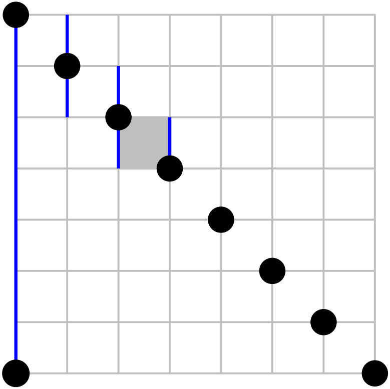
- In general, no Manhattan network after sweep.

After the sweep



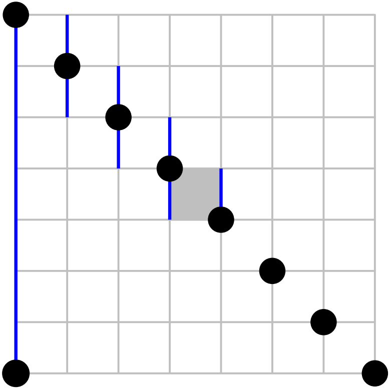
- In general, no Manhattan network after sweep.

After the sweep



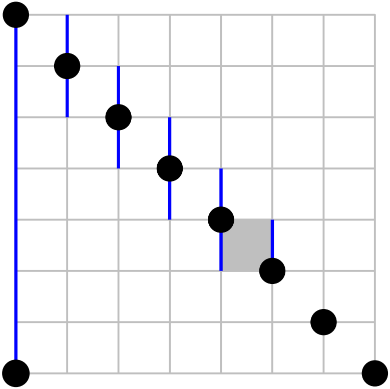
- In general, no Manhattan network after sweep.

After the sweep



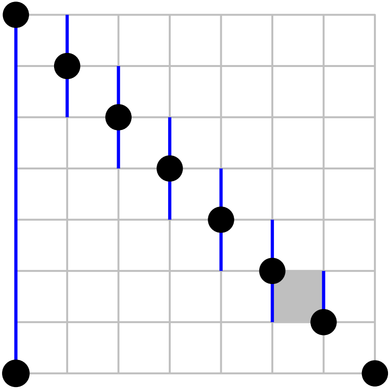
- In general, no Manhattan network after sweep.

After the sweep



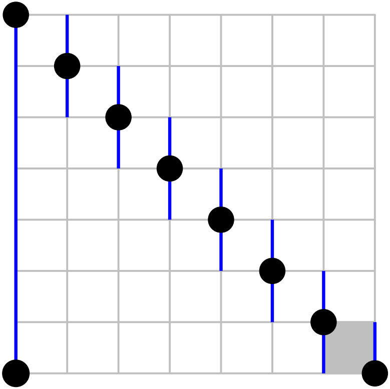
- In general, no Manhattan network after sweep.

After the sweep



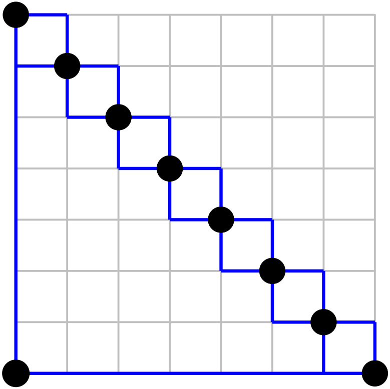
- In general, no Manhattan network after sweep.

After the sweep



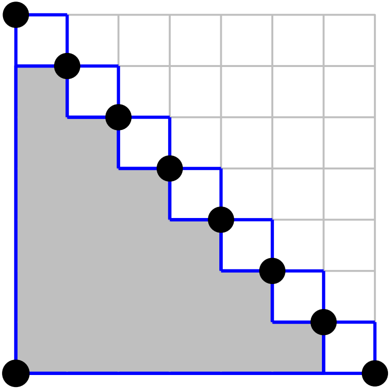
- In general, no Manhattan network after sweep.

After the sweep



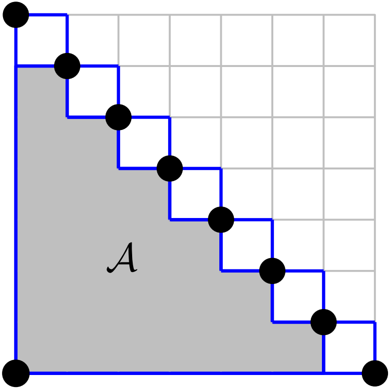
- In general, no Manhattan network after sweep.

After the sweep



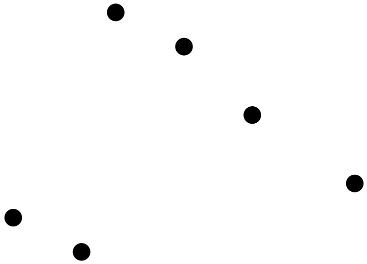
- In general, no Manhattan network after sweep.
- So-called **staircases** still empty.

After the sweep



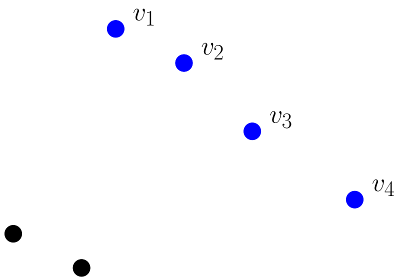
- In general, no Manhattan network after sweep.
- So-called **staircases** still empty.
- Call $\mathcal{A}$ the *staircase area*.

After the sweep



Definition (staircase)

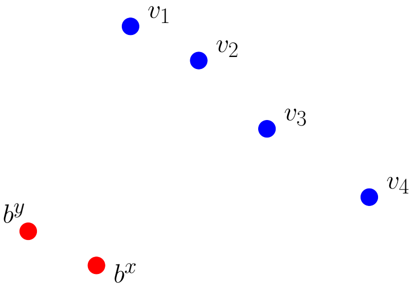
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.

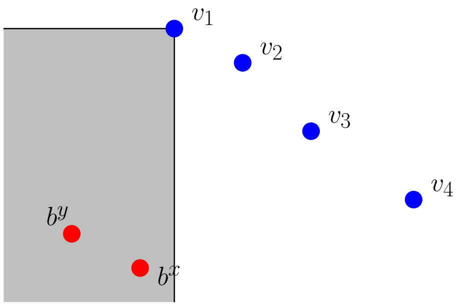
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)

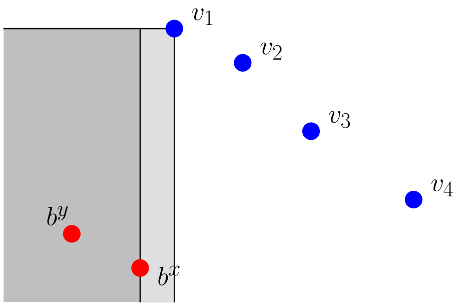
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)
- For all v_i, b^x is the x-neighbor of v_i in the third quadrant of v_i.

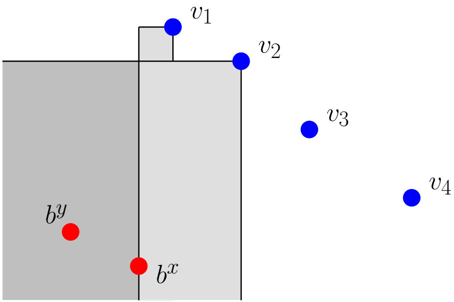
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)
- For all v_i, b^x is the x-neighbor of v_i in the third quadrant of v_i.

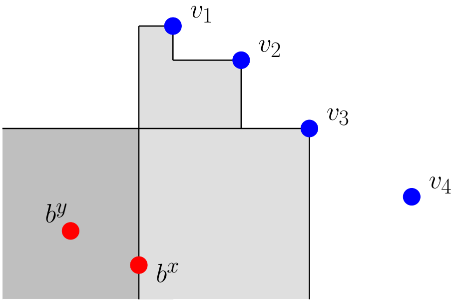
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)
- For all v_i, b^x is the x-neighbor of v_i in the third quadrant of v_i.

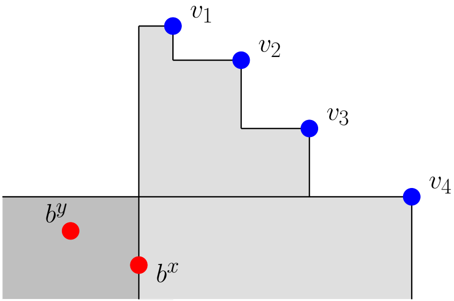
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)
- For all v_i, b^x is the x-neighbor of v_i in the third quadrant of v_i.

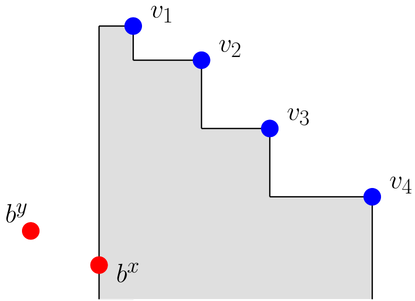
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)
- For all v_i, b^x is the x-neighbor of v_i in the third quadrant of v_i.

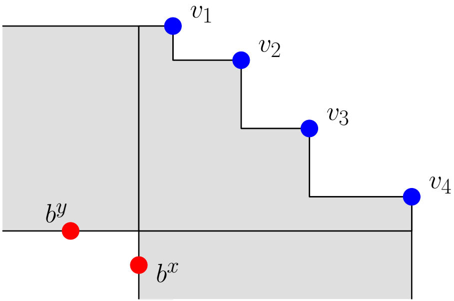
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)
- For all v_i, b^x is the x-neighbor of v_i in the third quadrant of v_i.

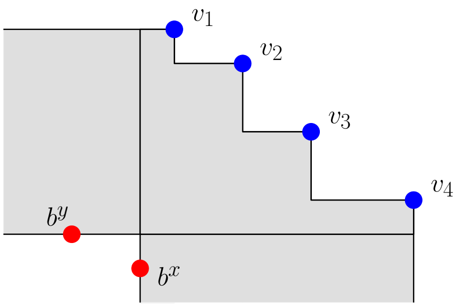
After the sweep



Definition (staircase)

- k sequence points $(v_1, \ldots, v_k)$.
- Two base points b^x, b^y. ($b^x = b^y$ possible.)
- For all v_i, b^x is the x-neighbor of v_i in the third quadrant of v_i.
- For all v_i, b^y is the y-neighbor of v_i in the third quadrant of v_i.

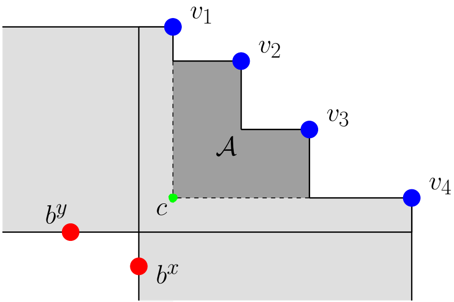
After the sweep



Observation

- The grey shaded areas contain no points.

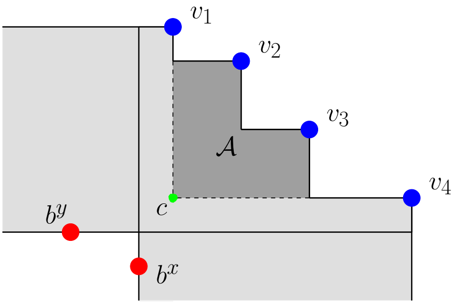
After the sweep



Observation

- The grey shaded areas contain no points.
- No sweep lines lie inside the staircase area $\mathcal{A}$.

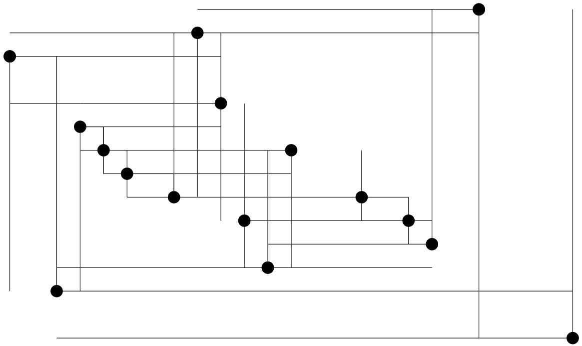
After the sweep



Observation

- The grey shaded areas contain no points.
- No sweep lines lie inside the staircase area $\mathcal{A}$.
- $\Rightarrow$ Points $v_3, \ldots, v_{k-2}$ need to be connected to *cross point* c.

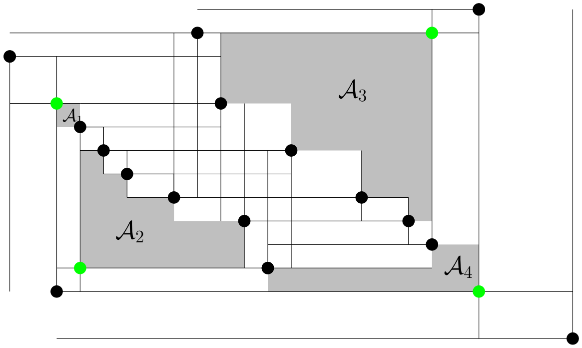
After the sweep



Lemma

- *Except for staircase areas, all critical pairs of points are connected via shortest paths after the sweep.*

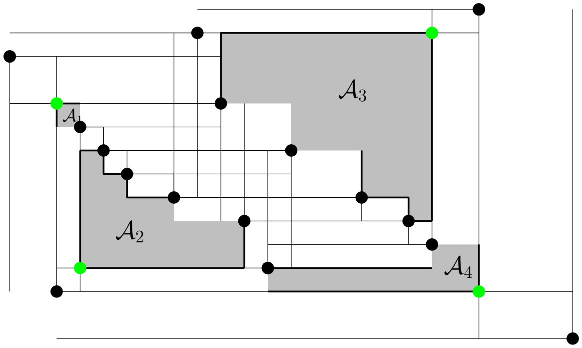
After the sweep



Lemma

- *Except for staircase areas, all critical pairs of points are connected via shortest paths after the sweep.*

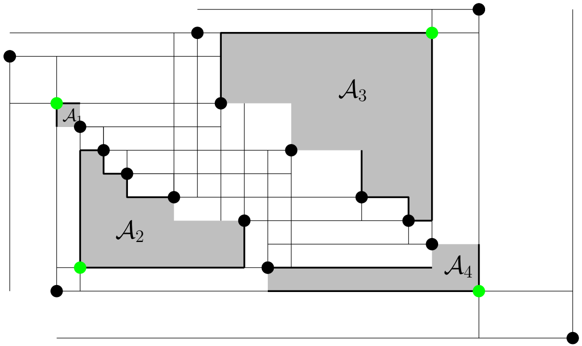
After the sweep



Lemma

- *Except for staircase areas, all critical pairs of points are connected via shortest paths after the sweep.*
- *The staircase areas $\mathcal{A}_i$ are bordered by as many line segments from the sweep step as possible,*

After the sweep



Lemma

- *Except for staircase areas, all critical pairs of points are connected via shortest paths after the sweep.*
- *The staircase areas $\mathcal{A}_i$ are bordered by as many line segments from the sweep step as possible, and are as small as possible.*

Approximation analysis

Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Approximation analysis

Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Phase I (Area $\overline{\mathcal{A}}$)

- By construction, one of the two lines inserted by the sweep is justified.

Approximation analysis

Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Phase I (Area $\overline{\mathcal{A}}$)

- By construction, one of the two lines inserted by the sweep is justified.

$\Rightarrow$ 2-Approximation.

Approximation analysis

Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Phase I (Area $\overline{\mathcal{A}}$)

- By construction, one of the two lines inserted by the sweep is justified.

$\Rightarrow$ *2-Approximation?!* Unfortunately not:

Approximation analysis

Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Phase I (Area $\overline{\mathcal{A}}$)

- By construction, one of the two lines inserted by the sweep is justified.

$\Rightarrow$ *2-Approximation?!* Unfortunately not:

Approximation analysis

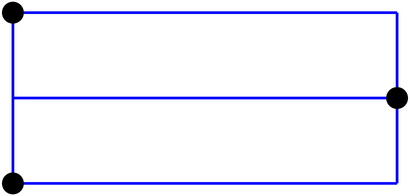
Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Phase I (Area $\overline{\mathcal{A}}$)

- By construction, one of the two lines inserted by the sweep is justified.

$\Rightarrow$ *2-Approximation?!* Unfortunately not:



Approximation analysis

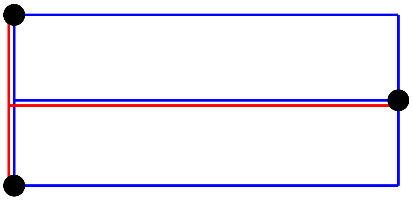
Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Phase I (Area $\overline{\mathcal{A}}$)

- By construction, one of the two lines inserted by the sweep is justified.

$\Rightarrow$ *2-Approximation?!* Unfortunately not:



Approximation analysis

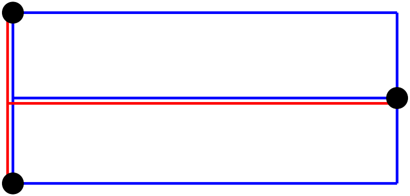
Consider the approximation seperately:

- inside staircases ($\mathcal{A} = \bigcup_i \mathcal{A}_i$, Phase II), and
- outside staircases ($\overline{\mathcal{A}} := \mathbb{R}^2 \setminus \mathcal{A}$, Phase I).

Phase I (Area $\overline{\mathcal{A}}$)

- By construction, one of the two lines inserted by the sweep is justified.

$\Rightarrow$ **3-Approximation.** ✓



Approximation analysis

Phase II (Area $\overline{\mathcal{A}}$)

- Use standard 2-approximation for staircases.

Approximation analysis

Phase II (Area $\overline{\mathcal{A}}$)

- Use standard 2-approximation for staircases.

- Let $\mathcal{A}^*$ be optimal staircase areas. Note: $\mathcal{A} \subseteq \mathcal{A}^*$.

Approximation analysis

Phase II (Area $\overline{\mathcal{A}}$)

- Use standard 2-approximation for staircases.

- Let $\mathcal{A}^*$ be optimal staircase areas. Note: $\mathcal{A} \subseteq \mathcal{A}^*$.
- Let $alg = alg_{\mathcal{A}} + alg_{\overline{\mathcal{A}}}$ and $opt = opt_{\mathcal{A}^*} + opt_{\overline{\mathcal{A}^*}}$.

Approximation analysis

Phase II (Area $\overline{\mathcal{A}}$)

- Use standard 2-approximation for staircases.

- Let $\mathcal{A}^*$ be optimal staircase areas. Note: $\mathcal{A} \subseteq \mathcal{A}^*$.
- Let $alg = alg_{\mathcal{A}} + alg_{\overline{\mathcal{A}}}$ and $opt = opt_{\mathcal{A}^*} + opt_{\overline{\mathcal{A}^*}}$.
- Phase I:

$$alg_{\overline{\mathcal{A}}} \leq 3 \cdot opt_{\overline{\mathcal{A}^*}}.$$

Approximation analysis

Phase II (Area $\overline{\mathcal{A}}$)

- Use standard 2-approximation for staircases.

- Let $\mathcal{A}^*$ be optimal staircase areas. Note: $\mathcal{A} \subseteq \mathcal{A}^*$.
- Let $alg = alg_{\mathcal{A}} + alg_{\overline{\mathcal{A}}}$ and $opt = opt_{\mathcal{A}^*} + opt_{\overline{\mathcal{A}^*}}$.
- Phase I:

$$alg_{\overline{\mathcal{A}}} \leq 3 \cdot opt_{\overline{\mathcal{A}^*}}.$$

- Phase II:

$$alg_{\mathcal{A}} \leq 2 \cdot opt_{\mathcal{A}} \leq 2 \cdot opt_{\mathcal{A}^*}.$$

Approximation analysis

Phase II (Area $\overline{\mathcal{A}}$)

- Use standard 2-approximation for staircases.

- Let $\mathcal{A}^*$ be optimal staircase areas. Note: $\mathcal{A} \subseteq \mathcal{A}^*$.
- Let $alg = alg_{\mathcal{A}} + alg_{\overline{\mathcal{A}}}$ and $opt = opt_{\mathcal{A}^*} + opt_{\overline{\mathcal{A}^*}}$.
- Phase I:

$$alg_{\overline{\mathcal{A}}} \leq 3 \cdot opt_{\overline{\mathcal{A}^*}}.$$

- Phase II:

$$alg_{\mathcal{A}} \leq 2 \cdot opt_{\mathcal{A}} \leq 2 \cdot opt_{\mathcal{A}^*}.$$

- Altogether:

$$alg \;=\; alg_{\mathcal{A}} + alg_{\overline{\mathcal{A}}} \;\leq\; 2 \cdot opt_{\mathcal{A}^*} + 3 \cdot opt_{\overline{\mathcal{A}^*}} \;\leq\; 3 \cdot opt. \qquad \square$$

Conclusion

We presented a simplified 3-approximation algorithm for MMNs.

Future work:

Conclusion

We presented a simplified 3-approximation algorithm for MMNs.

Future work:

- Design better approximation algorithms.

Conclusion

We presented a simplified 3-approximation algorithm for MMNs.

Future work:

- Design better approximation algorithms.
- Design PTAS.

Conclusion

We presented a simplified 3-approximation algorithm for MMNs.

Future work:

- Design better approximation algorithms.
- Design PTAS.
- Design efficient optimal algorithm or prove NP-hardness!

Conclusion

We presented a simplified 3-approximation algorithm for MMNs.

Future work:

- Design better approximation algorithms.
- Design PTAS.
- Design efficient optimal algorithm or prove NP-hardness!

Thank you!