A Tight Lower Bound for Planar Multiway Cut with Fixed Number of Terminals

Dániel Marx[1]

[1]Computer and Automation Research Institute,
Hungarian Academy of Sciences (MTA SZTAKI)
Budapest, Hungary

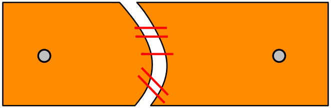
ICALP 2012
Warwick, UK
July 13, 2012

A classical problem

$s - t$ Cut

Input: A graph G, an integer p, vertices s and t

Output: A set S of at most p edges such that removing S separates s and t.



Fact

A minimum $s - t$ cut can be found in polynomial time.

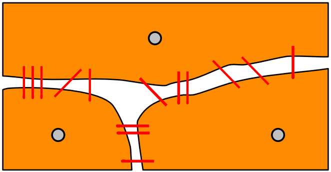
What about separating more than two terminals?

More than two terminals

Multiway Cut

Input: A graph G, an integer p, and a set T of terminals
Output: A set S of at most p edges such that removing S separates any two vertices of T

Note: Also called Multiterminal Cut or k-Terminal Cut.



Theorem [Dalhaus et al. 1994]

NP-hard already for $|T| = 3$.

Planar graphs

Theorem [Dalhaus et al. 1994] [Hartvigsen 1998] [Bentz 2012]

k-Terminal Cut can be solved in time $n^{O(k)}$ on **planar graphs.**

Can we improve the dependence on the number k of terminals?

- Is there a $c^k \cdot n^{O(1)}$ algorithm?
  (Asked by [Dalhaus et al. 1994])
- Is the problem fixed-parameter tractable?
  (Appears in the open problem list of [Downey-Fellows 1999])

[A problem is **fixed-parameter tractable (FPT)** parameterized by k if it can be solved in time $f(k) \cdot n^{O(1)}$ for some computable function $f(k)$ depending only on k.]

Results

Main result 1

k-Terminal Cut on planar graphs is W[1]-hard parameterized by the number k of terminals.

Lower bound on the exponent:

Main result 2

Assuming ETH, k-Terminal Cut on planar graphs cannot be solved in time $f(k) \cdot n^{o(\sqrt{k})}$ for any computable function $f(k)$.

[**Exponential Time Hypothesis (ETH):** n-variable 3-Sat cannot be solved in time $2^{o(n)}$.]

Bound on the exponent is tight:

Next talk [Klein and M.]

k-Terminal Cut on planar graphs can be solved in time $c^k \cdot n^{O(\sqrt{k})}$.

Results

Main result 1

k-Terminal Cut on planar graphs is W[1]-hard parameterized by the number k of terminals.

Lower bound on the exponent:

Main result 2

Assuming ETH, k-Terminal Cut on planar graphs cannot be solved in time $f(k) \cdot n^{o(\sqrt{k})}$ for any computable function $f(k)$.

[**Exponential Time Hypothesis (ETH):** n-variable 3-Sat cannot be solved in time $2^{o(n)}$.]

Bound on the exponent is tight:

Next talk [Klein and M.]

k-Terminal Cut on planar graphs can be solved in time $c^k \cdot n^{O(\sqrt{k})}$.

Results

Main result 1

k-Terminal Cut on planar graphs is W[1]-hard parameterized by the number k of terminals.

Lower bound on the exponent:

Main result 2

Assuming ETH, k-Terminal Cut on planar graphs cannot be solved in time $f(k) \cdot n^{o(\sqrt{k})}$ for any computable function $f(k)$.

[**Exponential Time Hypothesis (ETH):** n-variable 3-Sat cannot be solved in time $2^{o(n)}$.]

Bound on the exponent is tight:

Next talk [Klein and M.]

k-Terminal Cut on planar graphs can be solved in time $c^k \cdot n^{O(\sqrt{k})}$.

W[1]-hardness

Definition

A **parameterized reduction** from problem A to B maps an instance (x, k) of A to instance (x', k') of B such that

- $(x, k) \in A \iff (x', k') \in B$,
- $k' \leq g(k)$ for some computable function g.
- (x', k') can be computed in time $f(k) \cdot |x|^{O(1)}$.

Easy: If there is a parameterized reduction from problem A to problem B and B is FPT, then A is FPT as well.

Definition

A problem P is **W[1]-hard** if there is a parameterized reduction from k-Clique to P.

W[1]-hardness

Definition

A **parameterized reduction** from problem A to B maps an instance (x, k) of A to instance (x', k') of B such that

- $(x, k) \in A \iff (x', k') \in B$,
- $k' \leq g(k)$ for some computable function g.
- (x', k') can be computed in time $f(k) \cdot |x|^{O(1)}$.

Easy: If there is a parameterized reduction from problem A to problem B and B is FPT, then A is FPT as well.

Definition

A problem P is **W[1]-hard** if there is a parameterized reduction from k-Clique to P.

W[1]-hardness vs. NP-hardness

W[1]-hardness proofs are more delicate than NP-hardness proofs: we need to control the new parameter.

Example: k-Independent Set can be reduced to k'-Vertex Cover with $k' := n - k$. But this is **not** a parameterized reduction.

NP-hardness proof

Reduction from some graph problem. We build n vertex gadgets of constant size and m edge gadgets of constant size.

W[1]-hardness proof

Reduction from k-Clique. We build k large vertex gadgets, each having n states (and/or $\binom{k}{2}$ large edge gadgets with m states).

Planar problems

Another difference: Most problems remain NP-hard on planar graphs, but become FPT.

Algorithmic techniques for planar problems:

- Baker's shifting technique + treewidth
- Bidimensionality
- Protrusions

Very few W[1]-hardness results so far for planar problems.

Tight bounds

Theorem [Chen et al. 2004]

Assuming ETH, there is no $f(k) \cdot n^{o(k)}$ algorithm for k-Clique for any computable function f.

Transfering to other problems:
If there is a parameterized reduction from k-Clique to problem A mapping (x, k) to $(x', g(k))$, then an $f(k) \cdot n^{o(g^{-1}(k))}$ algorithm for problem A gives an $f(k) \cdot n^{o(k)}$ algorithm for k-Clique, contradicting ETH.

Bottom line:
To rule out $f(k) \cdot n^{o(\sqrt{k})}$ algorithms, we need a parameterized reduction that blows up the parameter at most quadratically.

Tight bounds

Theorem [Chen et al. 2004]

Assuming ETH, there is no $f(k) \cdot n^{o(k)}$ algorithm for k-Clique for any computable function f.

Transfering to other problems:

k-Clique (x, k)	$\Rightarrow$	Problem A $(x', g(k))$
$f(k) \cdot n^{o(k)}$ algorithm	$\Leftarrow$	$f(k) \cdot n^{o(g^{-1}(k))}$ algorithm

Bottom line:

To rule out $f(k) \cdot n^{o(\sqrt{k})}$ algorithms, we need a parameterized reduction that blows up the parameter at most quadratically.

Grid Tiling

Grid Tiling

Input: A $k \times k$ matrix and a set of pairs $S_{i,j} \subseteq [D] \times [D]$ for each cell.

Find: A pair $s_{i,j} \in S_{i,j}$ for each cell such that

- Horizontal neighbors agree in the first component.
- Vertical neighbors agree in the second component.

(1,1) (1,3) (4,2)	(1,5) (4,1) (3,5)	(1,1) (4,2) (3,3)
(2,2) (4,1)	(1,3) (2,1)	(2,2) (3,2)
(3,1) (3,2) (3,3)	(1,1) (3,1)	(3,2) (3,5)

$k = 3, \ D = 5$

Grid Tiling

Grid Tiling

Input: A $k \times k$ matrix and a set of pairs $S_{i,j} \subseteq [D] \times [D]$ for each cell.

Find: A pair $s_{i,j} \in S_{i,j}$ for each cell such that

- Horizontal neighbors agree in the first component.
- Vertical neighbors agree in the second component.

(1,1) (1,3) (4,2)	(1,5) (4,1) (3,5)	(1,1) (4,2) (3,3)
(2,2) (4,1)	(1,3) (2,1)	(2,2) (3,2)
(3,1) (3,2) (3,3)	(1,1) (3,1)	(3,2) (3,5)

$k = 3, \ D = 5$

Grid Tiling is W[1]-hard

Reduction from k-Clique

Definition of the sets:

- For $i = j$: $(x, y) \in S_{i,j} \iff x = y$
- For $i \neq j$: $(x, y) \in S_{i,j} \iff x$ and y are adjacent.

	(v_i, v_i)			

Each diagonal cell defines a value v_i . . .

Grid Tiling is W[1]-hard

Reduction from k-Clique

Definition of the sets:

- For $i = j$: $(x, y) \in S_{i,j} \iff x = y$
- For $i \neq j$: $(x, y) \in S_{i,j} \iff x$ and y are adjacent.

	$(., v_i)$			
$(v_i, .)$	(v_i, v_i)	$(v_i, .)$	$(v_i, .)$	$(v_i, .)$
	$(., v_i)$			
	$(., v_i)$			
	$(., v_i)$			

. . . which appears on a "cross"

Grid Tiling is W[1]-hard

Reduction from k-Clique

Definition of the sets:

- For $i = j$: $(x, y) \in S_{i,j} \iff x = y$
- For $i \neq j$: $(x, y) \in S_{i,j} \iff x$ and y are adjacent.

	$(., v_i)$			
$(v_i, .)$	(v_i, v_i)	$(v_i, .)$	$(v_i, .)$	$(v_i, .)$
	$(., v_i)$			
	$(., v_i)$		(v_j, v_j)	
	$(., v_i)$			

v_i and v_j are adjacent for every $1 \leq i < j \leq k$.

Grid Tiling is W[1]-hard

Reduction from k-Clique

Definition of the sets:

- For $i = j$: $(x, y) \in S_{i,j} \iff x = y$
- For $i \neq j$: $(x, y) \in S_{i,j} \iff x$ and y are adjacent.

	$(., v_i)$		$(., v_j)$	
$(v_i, .)$	(v_i, v_i)	$(v_i, .)$	(v_i, v_j)	$(v_i, .)$
	$(., v_i)$		$(., v_j)$	
$(v_j, .)$	(v_j, v_i)	$(v_j, .)$	(v_j, v_j)	$(v_j, .)$
	$(., v_i)$		$(., v_j)$	

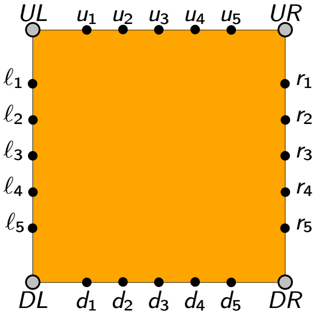
v_i and v_j are adjacent for every $1 \leq i < j \leq k$.

The gadget

For every set $S_{i,j}$, we construct a gadget such that

- for every $(x, y) \in S_{i,j}$, there is a minimum multiway cut that represents (x, y).
- every minimum cut represents some $(x, y) \in S_{i,j}$.

Main part of the proof: constructing these gadgets.



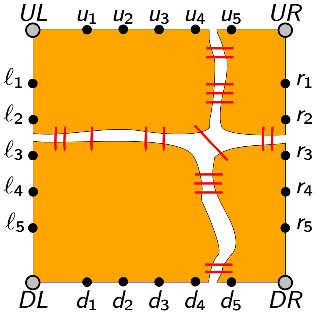
The gadget.

The gadget

For every set $S_{i,j}$, we construct a gadget such that

- for every $(x, y) \in S_{i,j}$, there is a minimum multiway cut that represents (x, y).
- every minimum cut represents some $(x, y) \in S_{i,j}$.

Main part of the proof: constructing these gadgets.



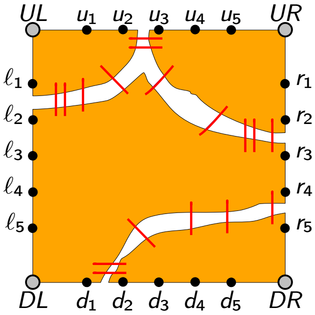
A cut representing (2,4).

The gadget

For every set $S_{i,j}$, we construct a gadget such that

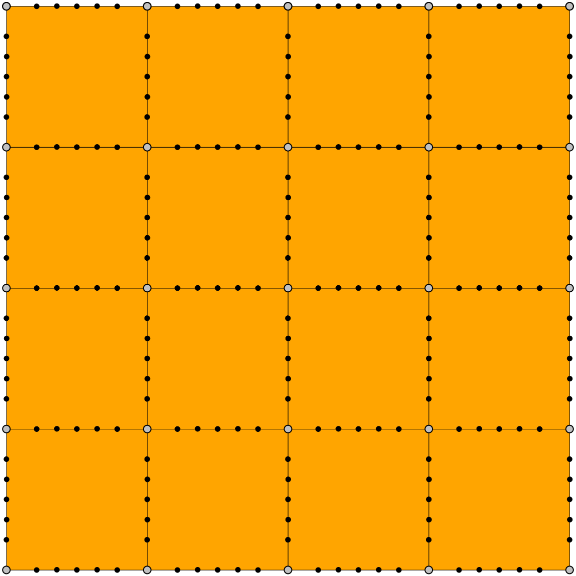
- for every $(x, y) \in S_{i,j}$, there is a minimum multiway cut that represents (x, y).
- every minimum cut represents some $(x, y) \in S_{i,j}$.

Main part of the proof: constructing these gadgets.

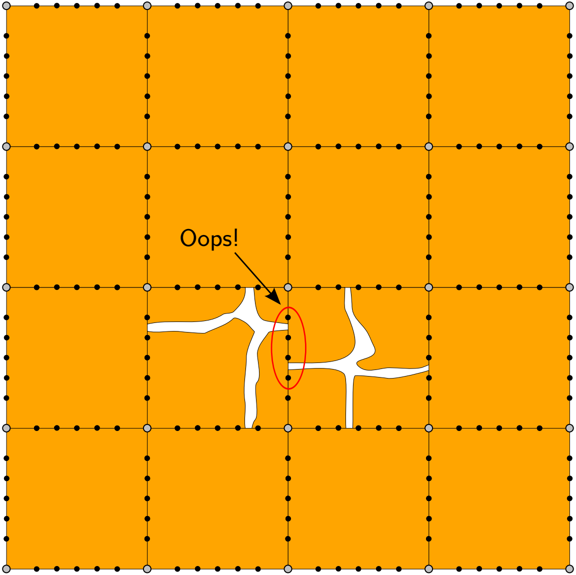


A cut not representing any pair.

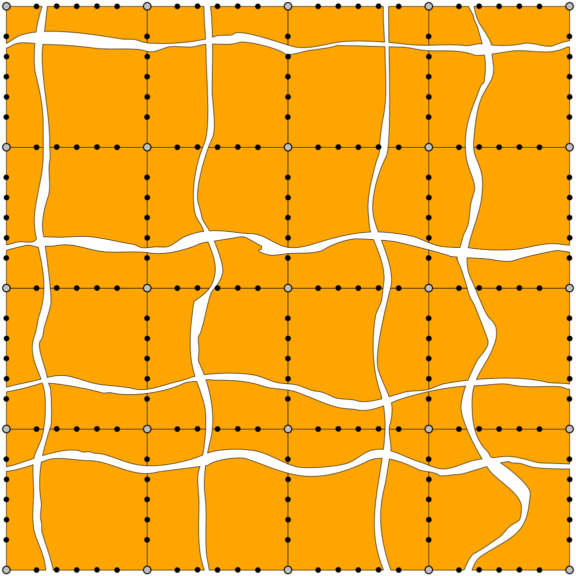
Putting together the gadgets



Putting together the gadgets

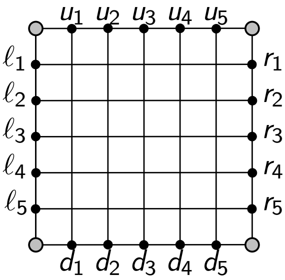


Putting together the gadgets



Constructing the gadget

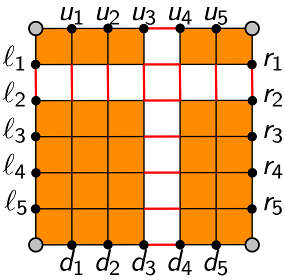
This is what we would like to have:



- We set up the weight of the grid edges such that every cheap cut is like this.
- Furthermore, we add something in the cells that ensures that the intersection of the horizontal and the vertical cut has to be a special cell.

Constructing the gadget

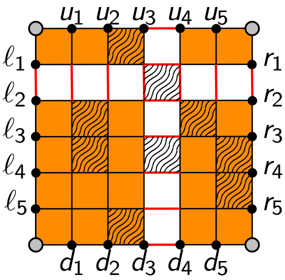
This is what we would like to have:



- We set up the weight of the grid edges such that every cheap cut is like this.
- Furthermore, we add something in the cells that ensures that the intersection of the horizontal and the vertical cut has to be a special cell.

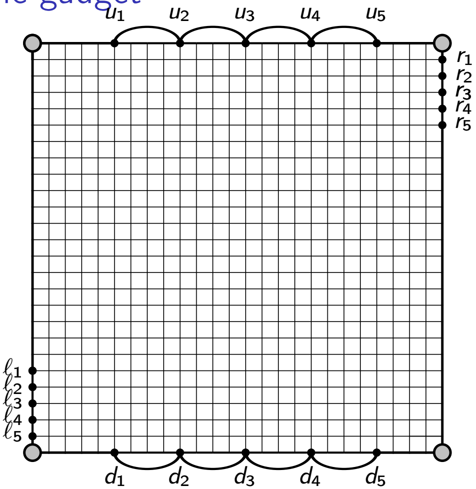
Constructing the gadget

This is what we would like to have:



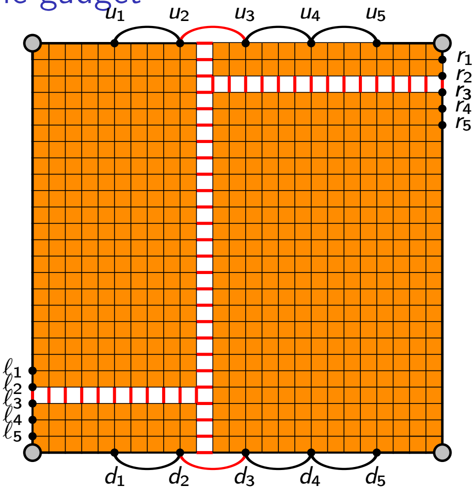
- We set up the weight of the grid edges such that every cheap cut is like this.
- Furthermore, we add something in the cells that ensures that the intersection of the horizontal and the vertical cut has to be a special cell.

Constructing the gadget



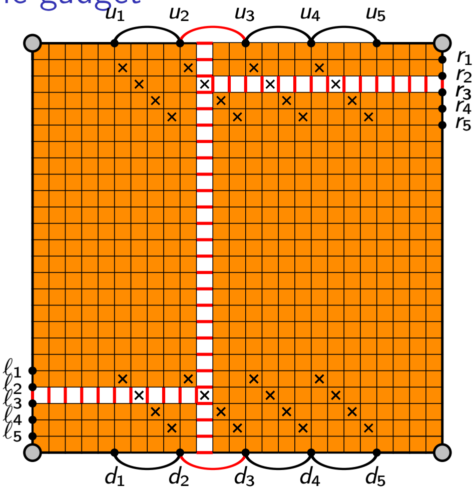
- We set up the weight of the grid edges such that every cheap cut is like this.
- We add something in the cells that ensures that the choice of the vertical cut determines the choice of the horizontal cuts.

Constructing the gadget



- We set up the weight of the grid edges such that every cheap cut is like this.
- We add something in the cells that ensures that the choice of the vertical cut determines the choice of the horizontal cuts.

Constructing the gadget



- We set up the weight of the grid edges such that every cheap cut is like this.
- We add something in the cells that ensures that the choice of the vertical cut determines the choice of the horizontal cuts.

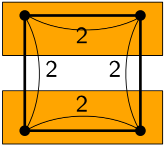
Special cells

Two different type of cells

normal cell special cell

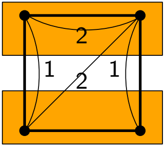
Special cells

They behave similarly with respect to horizontal cuts...



normal cell

4

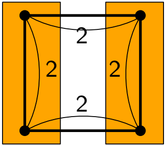


special cell

4

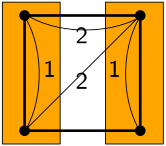
Special cells

They behave similarly with respect to vertical cuts. . .



normal cell

4

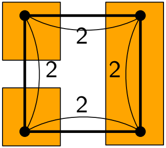


special cell

4

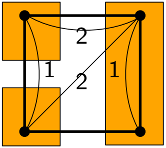
Special cells

. . . but they differ on 3-way cuts.



normal cell

6



special cell

5

Conclusions

- Main result: assuming ETH, there is no $f(k) \cdot n^{o(\sqrt{k})}$ time algorithm for planar k-terminal Multiway cut.
- (Almost) matches the $c^k \cdot n^{O(\sqrt{k})}$ time algorithm (next talk).
- Reduction from Grid Tiling (should be useful for other planar W[1]-hardness proofs).
- Main part: constructing the gadgets.