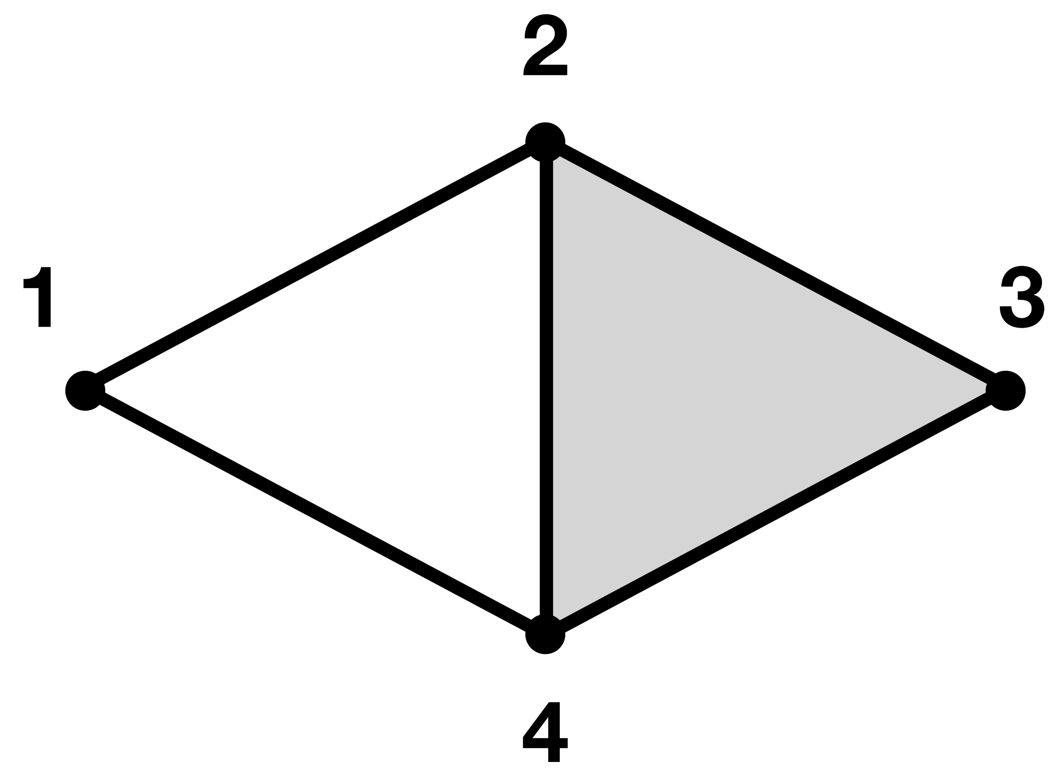
Topological Data Analysis

Lecture 10

Computational persistent homology

Oleg Kachan

Boundary matrix

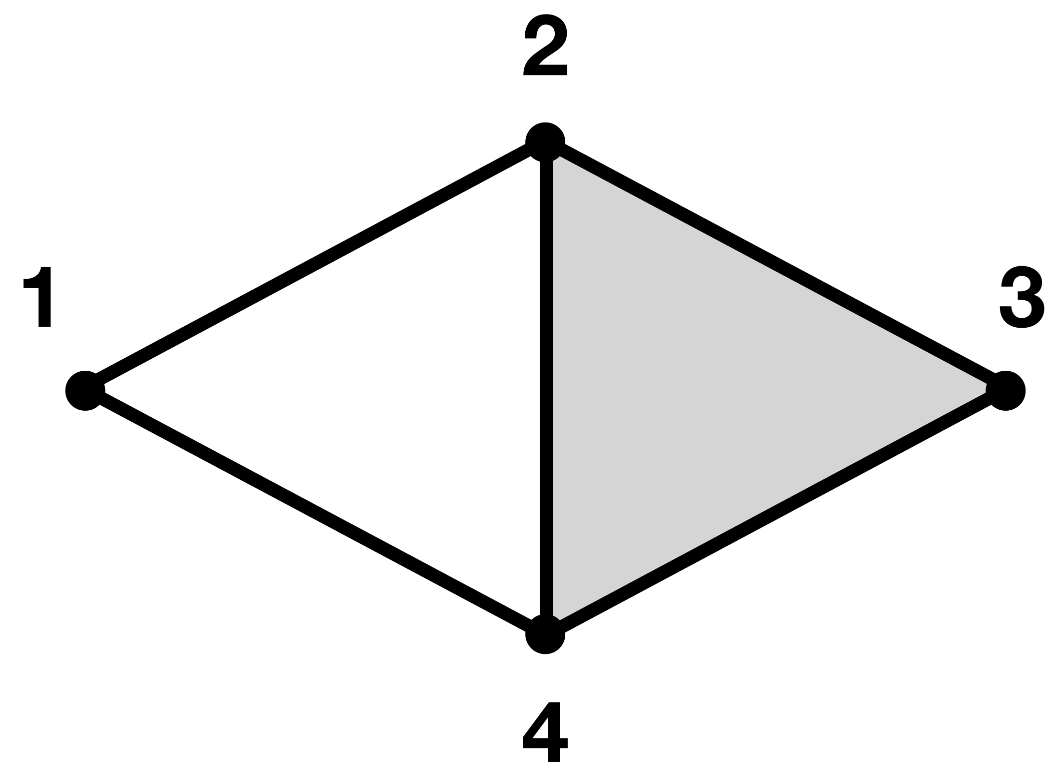


$\mathbf{B} =$

<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>12</th><th>14</th><th>23</th><th>24</th><th>34</th><th>234</th></tr>
<tr><td>0</td><td colspan="4">B_0</td><td colspan="5"></td><td></td></tr>
<tr><td>1</td><td colspan="4" rowspan="4"></td><td colspan="5" rowspan="4">B_1</td><td rowspan="4"></td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>12</td><td colspan="4" rowspan="5"></td><td colspan="5" rowspan="5"></td><td rowspan="5">B_2</td></tr>
<tr><td>14</td></tr>
<tr><td>23</td></tr>
<tr><td>24</td></tr>
<tr><td>34</td></tr>
</table>

$$K = \{\{0\}, \{1\}, \{2\}, \{3\}, \{12\}, \{14\}, \{23\}, \{24\}, \{34\}, \{234\}\}$$

Boundary matrix

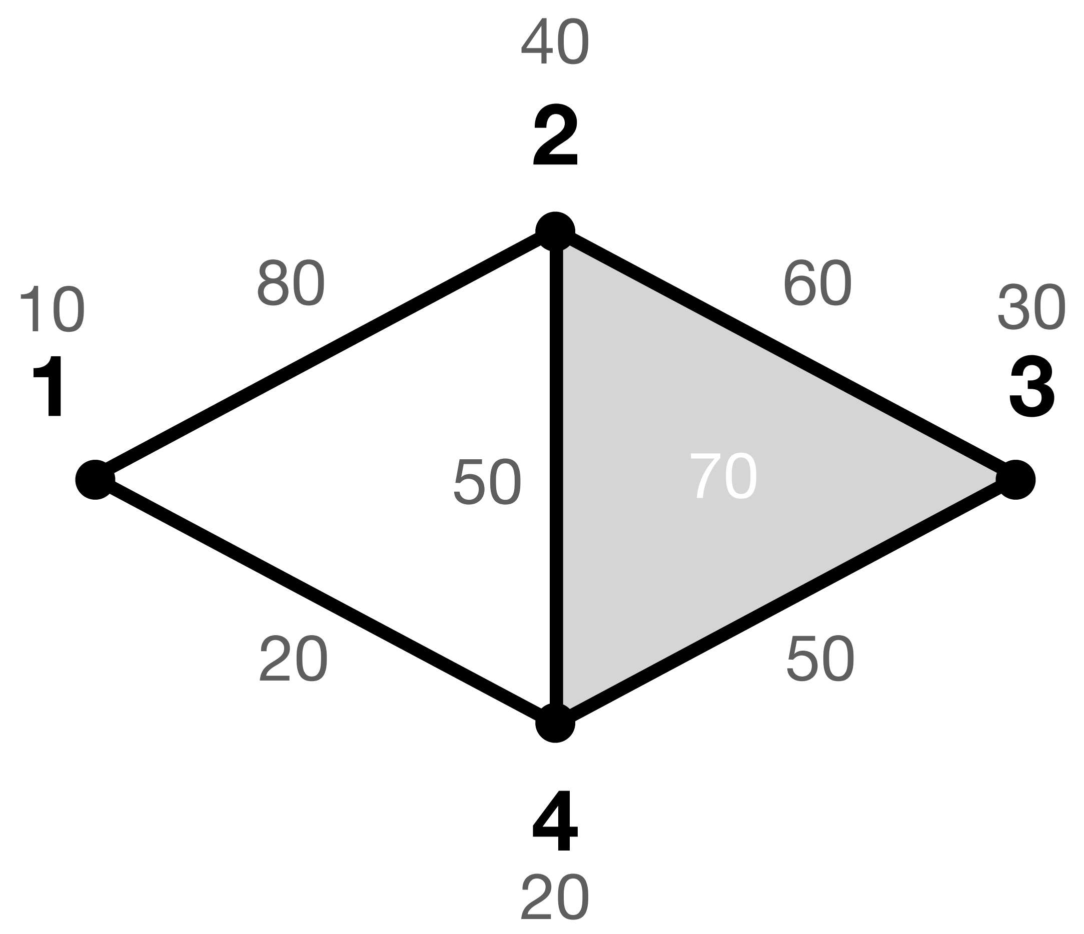


$\mathbf{B} =$

	1	2	3	4	12	14	23	24	34	234
0	1	1	1	1						
1					1	1				
2					1		1	1		
3							1		1	
4						1		1	1	
12										
14										
23										1
24										1
34										1

One may skip adding $\mathbf{B}_0$ to the full matrix or zero out it

Filtration function



A function $f : K \to \mathbb{R}$ is called a filtration function iff, either

$$f(\tau) \leq f(\sigma) \iff \tau \subseteq \sigma$$ (sublevel filtration)

$$f(\tau) \geq f(\sigma) \iff \tau \supseteq \sigma$$ (superlevel filtration)

$$K_t = \{\sigma \in K \mid f(\sigma) \leq t\}$$

sublevel set $t \in (-\infty, +\infty)$

$$K^t = \{\sigma \in K \mid f(\sigma) \geq t\}$$

superlevel set $t \in (+\infty, -\infty)$

A filtration is a sequence of sublevel (superlevel) sets s.t.

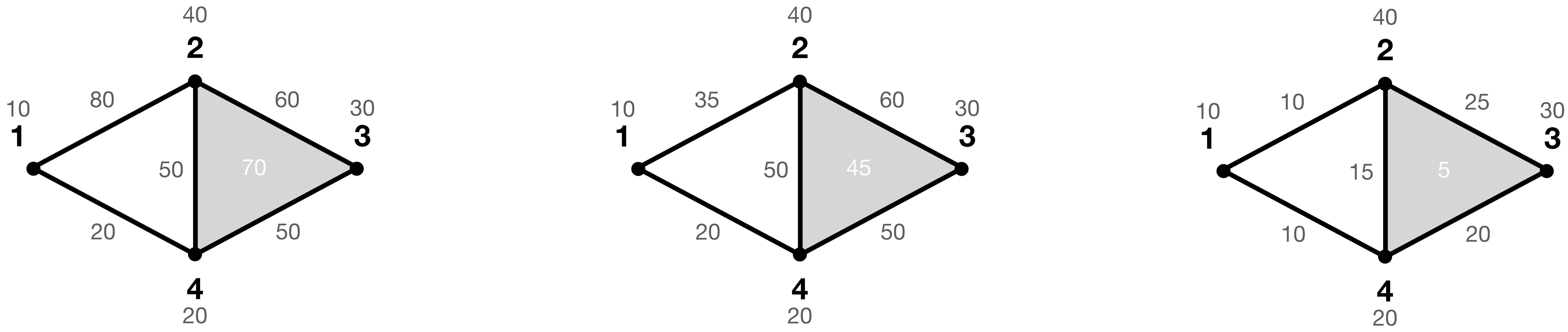
$$\varnothing \subset K_1 \subset K_2 \subset \ldots \subset K$$

sublevel filtration

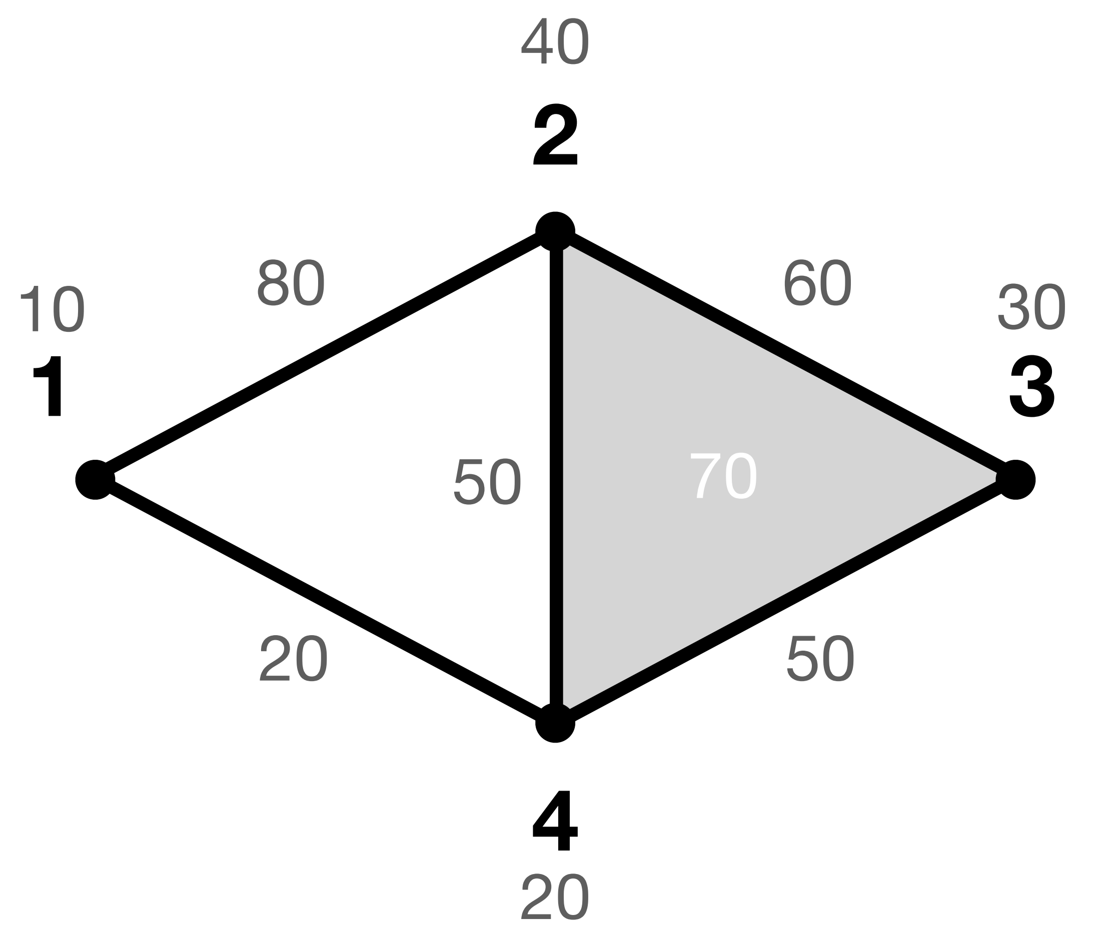
$$\varnothing \subset K^1 \subset K^2 \subset \ldots \subset K$$

superlevel filtration

Filtration function



Filtered simplicial complex



$\mathbf{B} =$

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
0										
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

Filtration function provides order on simplices, therefore on columns and rows of the filtration matrix. Ties are broken, first by simplex dimension, second by lexicographic order given by order on vertices.

Standard algorithm for computing persistent homology

Given a *n x n* matrix ∂ define a function mapping a column to the row index of its lowest nonzero element

$$\mathrm{low}(j) = \begin{cases} \max\{i \mid \partial_{ij} \neq 0\}, & \partial_j \neq 0, \\ -1, & \text{otherwise}\,. \end{cases}$$

Standard algorithm

- reduce ∂ by column additions
- select a current column ∂_j, moving from left to right
- for each *k* < *j* add columns ∂_k to current column ∂_j if $\mathrm{low}(k) = \mathrm{low}(j)$
- matrix is reduced when $\mathrm{low}(\,\cdot\,)$ is injective

Algorithm 3: Standard algorithm over $\mathbb{F}_2$

Input: An $n \times n$ boundary matrix ∂ over $\mathbb{F}_2$

for $j = 0, \ldots, n-1$ **do**
    while $\exists k < j$ *such that* $\mathrm{low}_\partial(k) = \mathrm{low}_\partial(j) > -1$ **do**
        $\partial_j \leftarrow \partial_j + \partial_k$
    end while
end for
return ∂

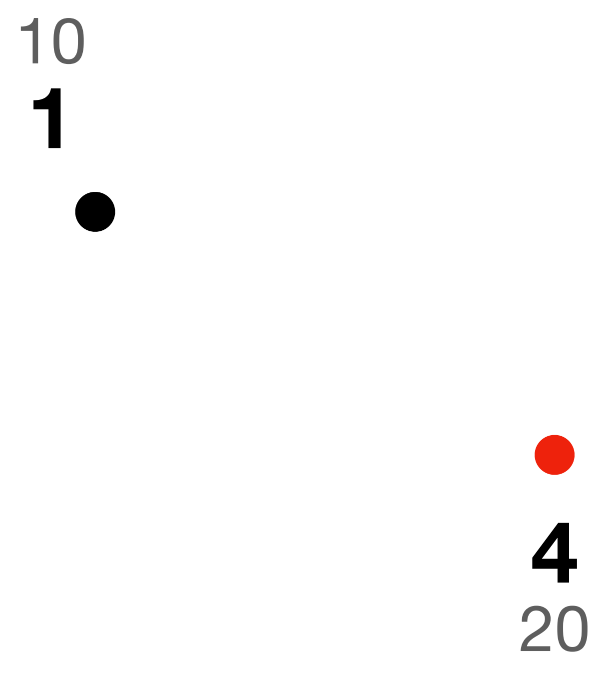
Topological Persistence and Simplification
Edelsbrunner el al. *Discrete & Computational Geometry (2002)*

Boundary matrix reduction

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=10

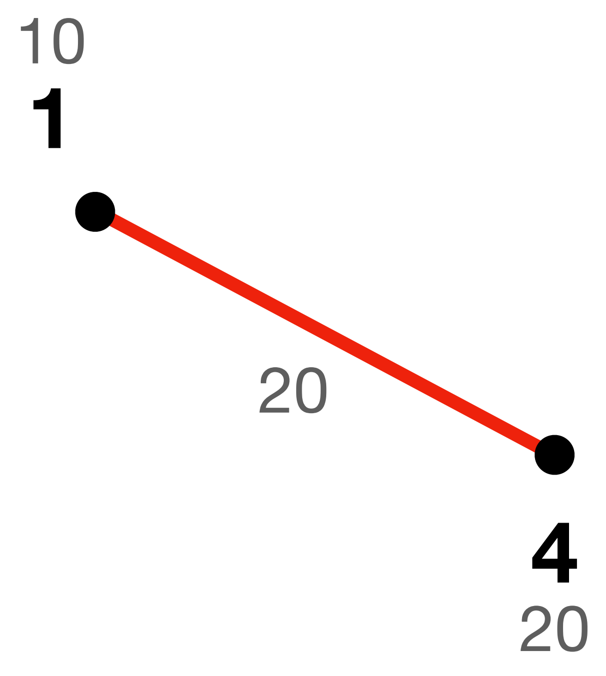
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=20

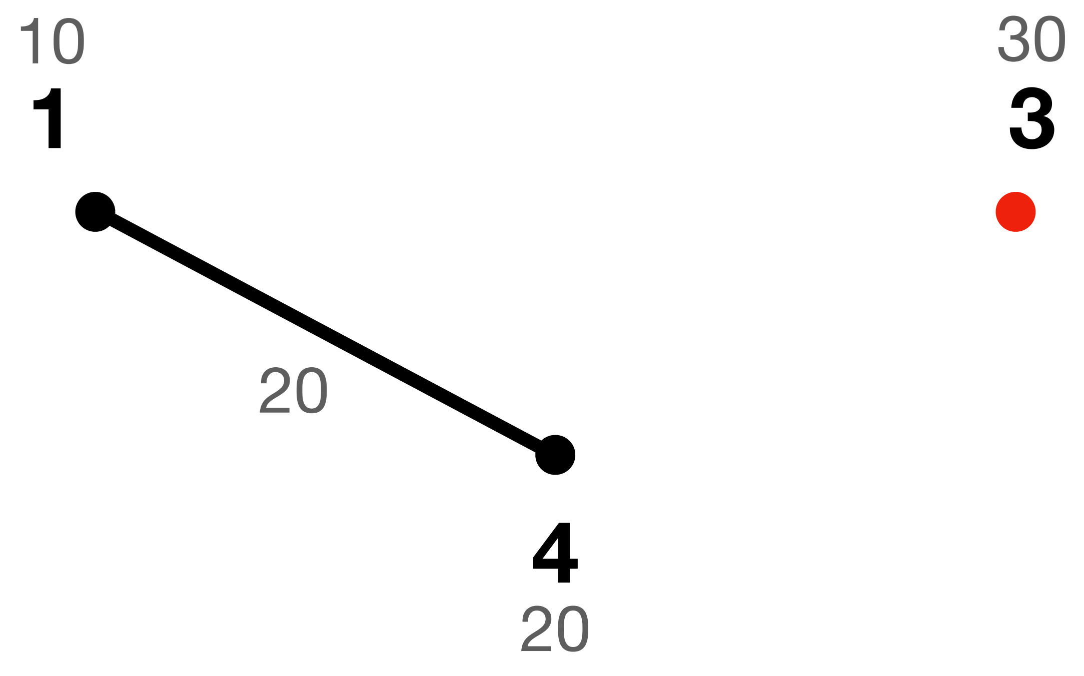
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=20

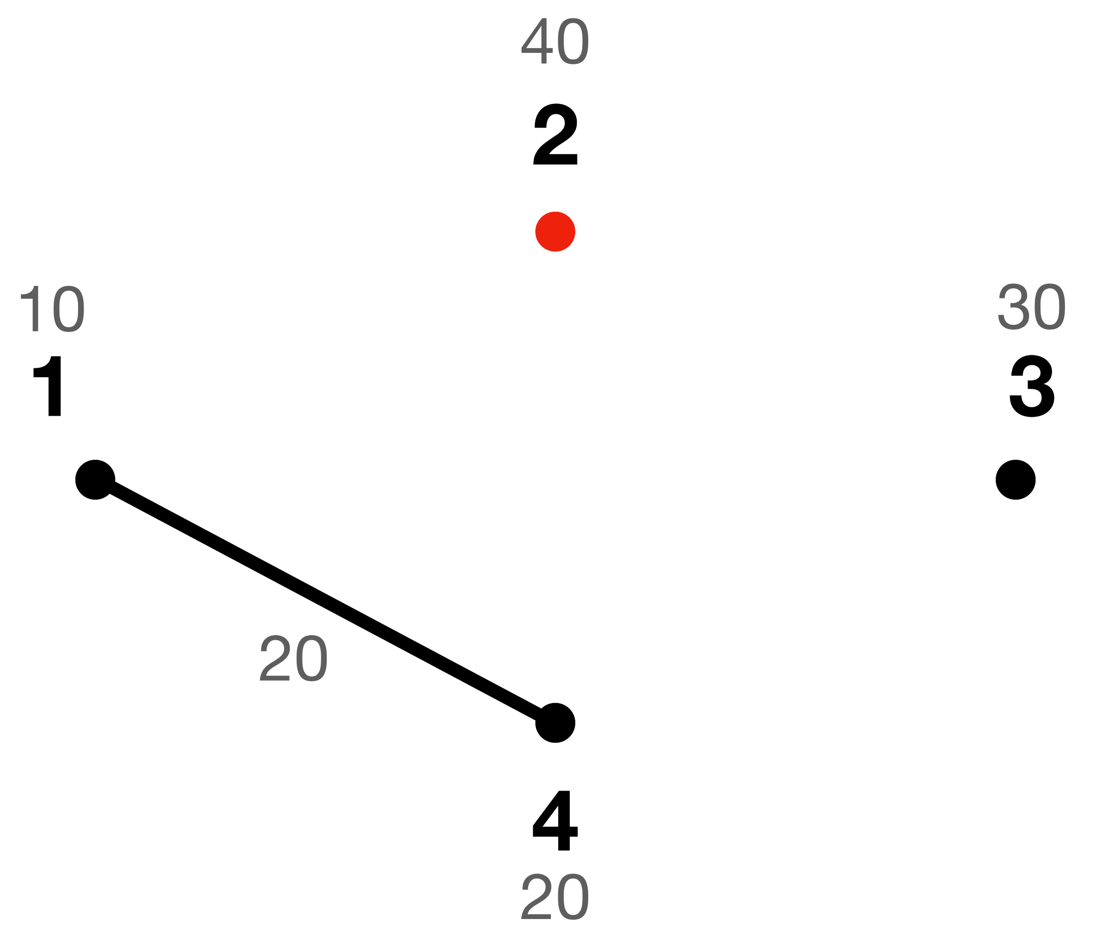
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=30

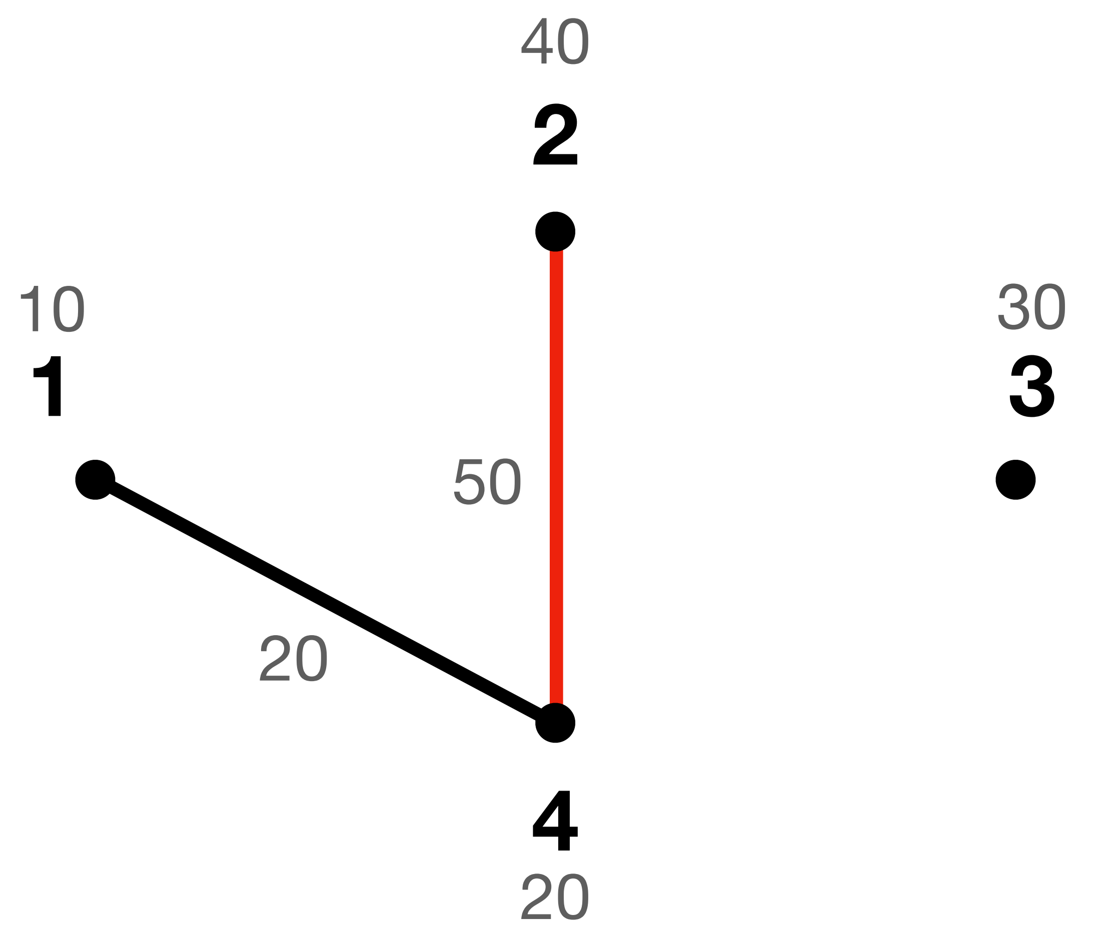
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=40

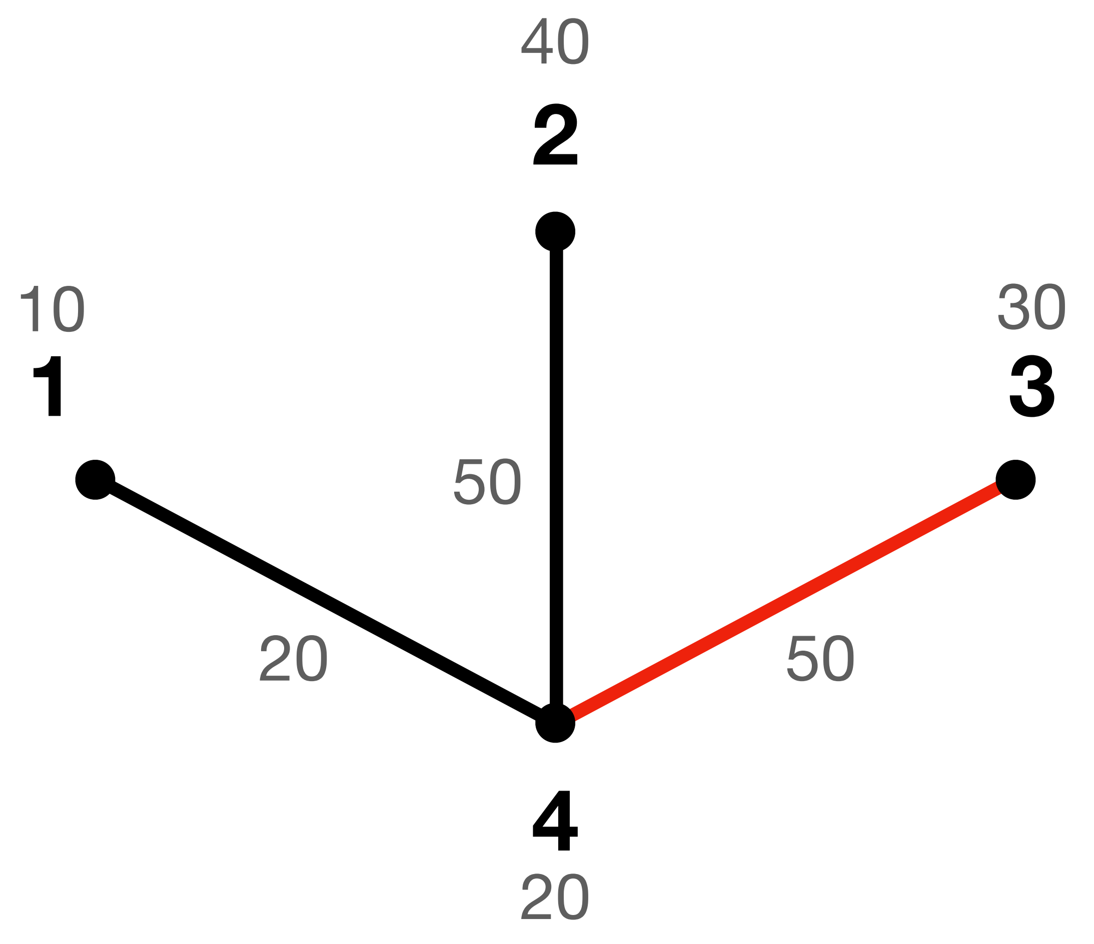
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=50

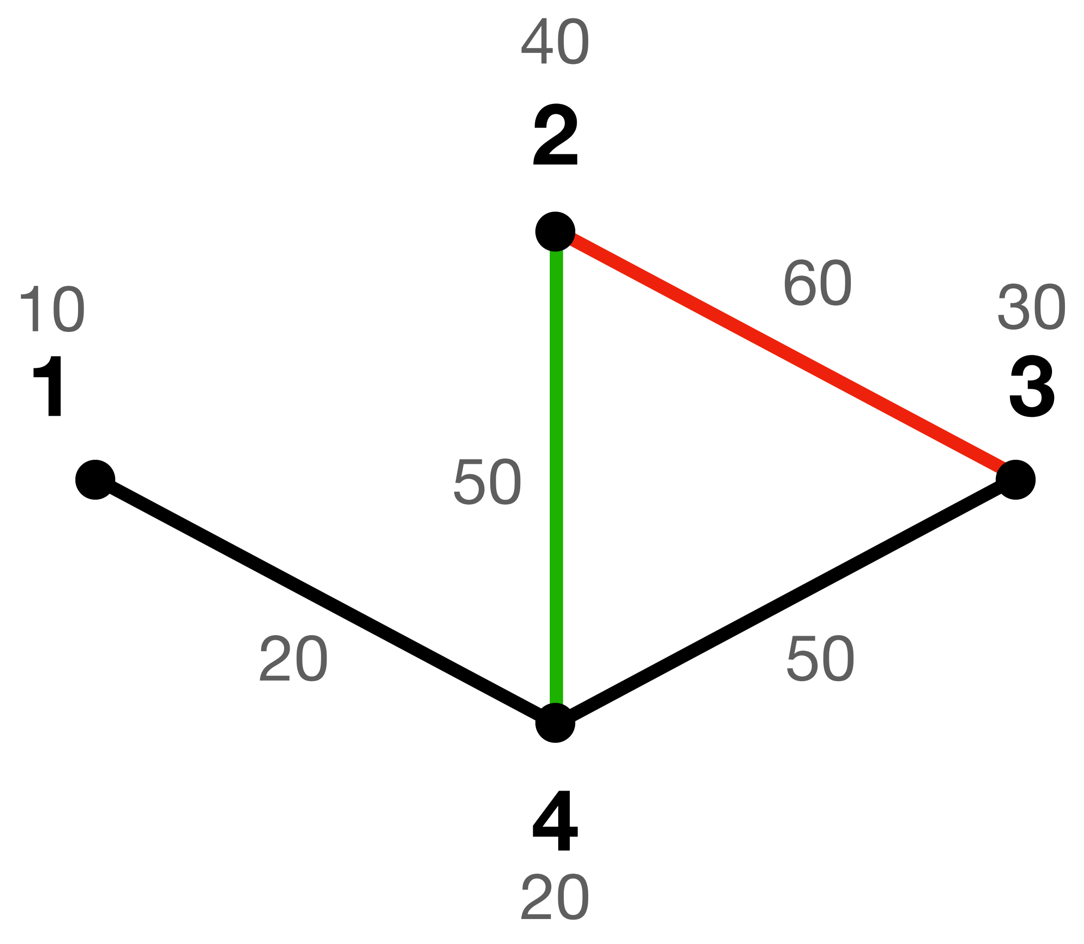
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=50

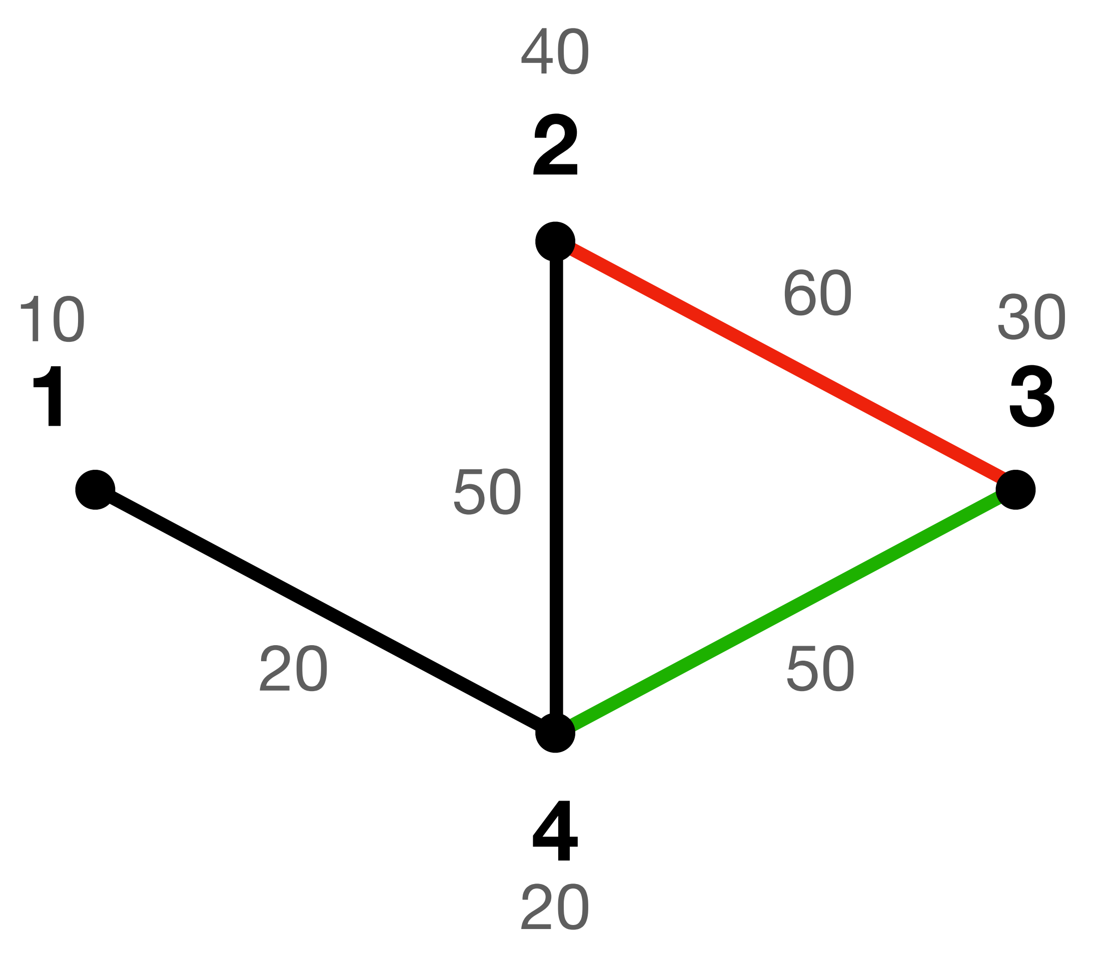
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

t=60

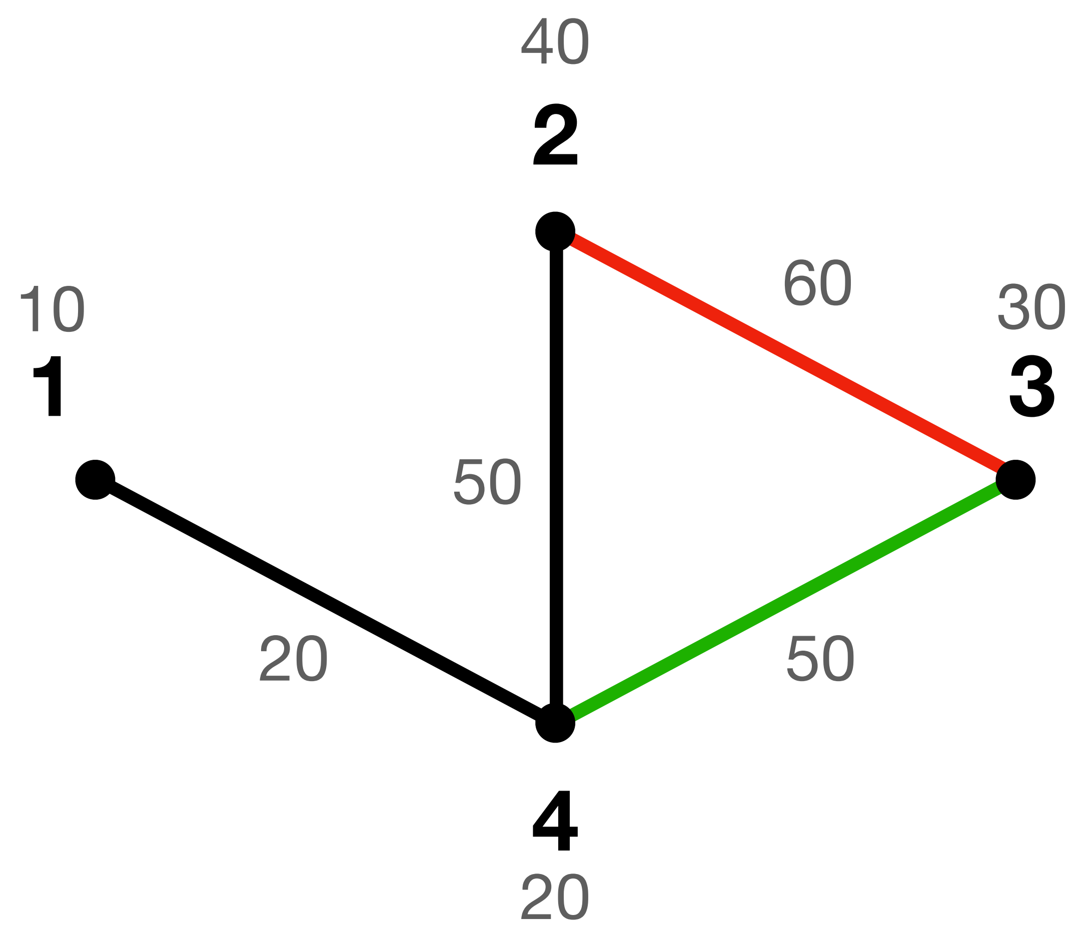
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1	1		
14										
3							1	1		
2						1				1
24									1	
34									1	
23									1	
12										

t=60

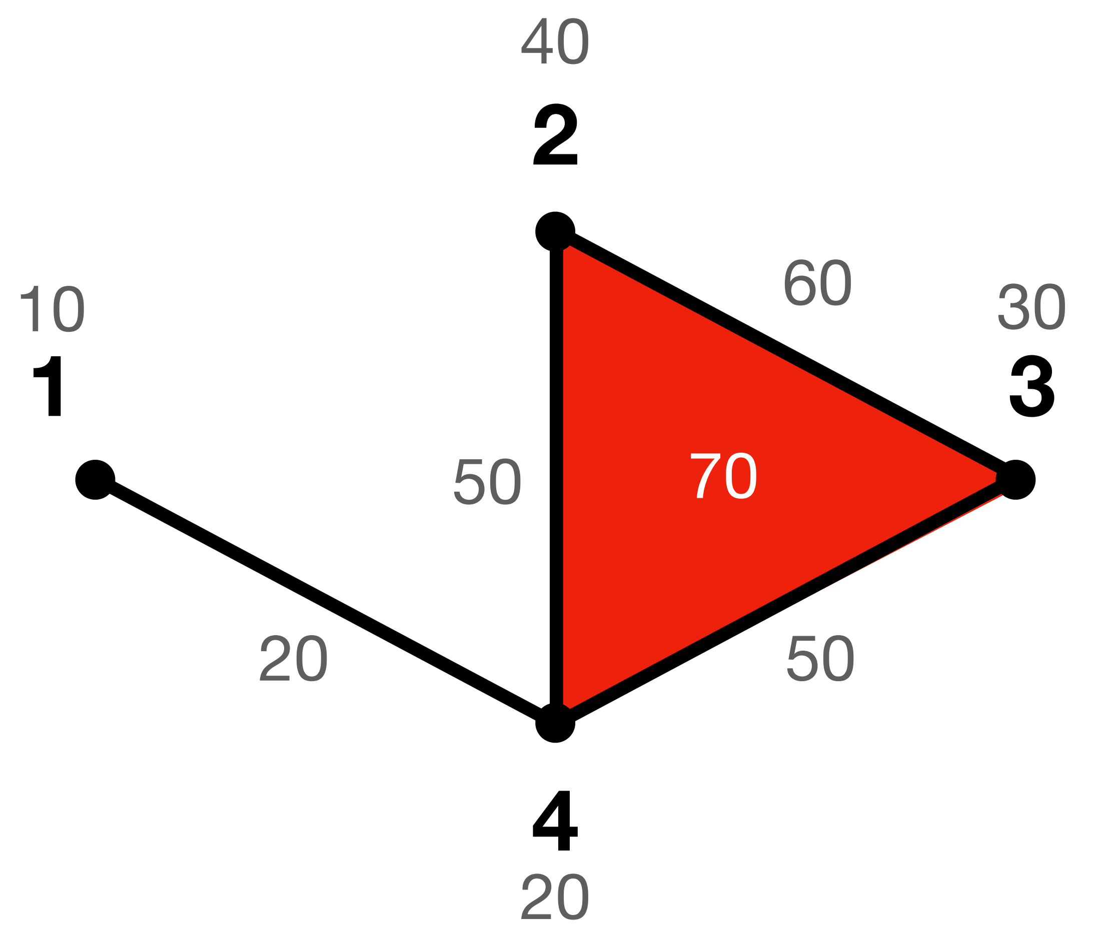
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1			
2						1				1
24									1	
34									1	
23									1	
12										

t=60

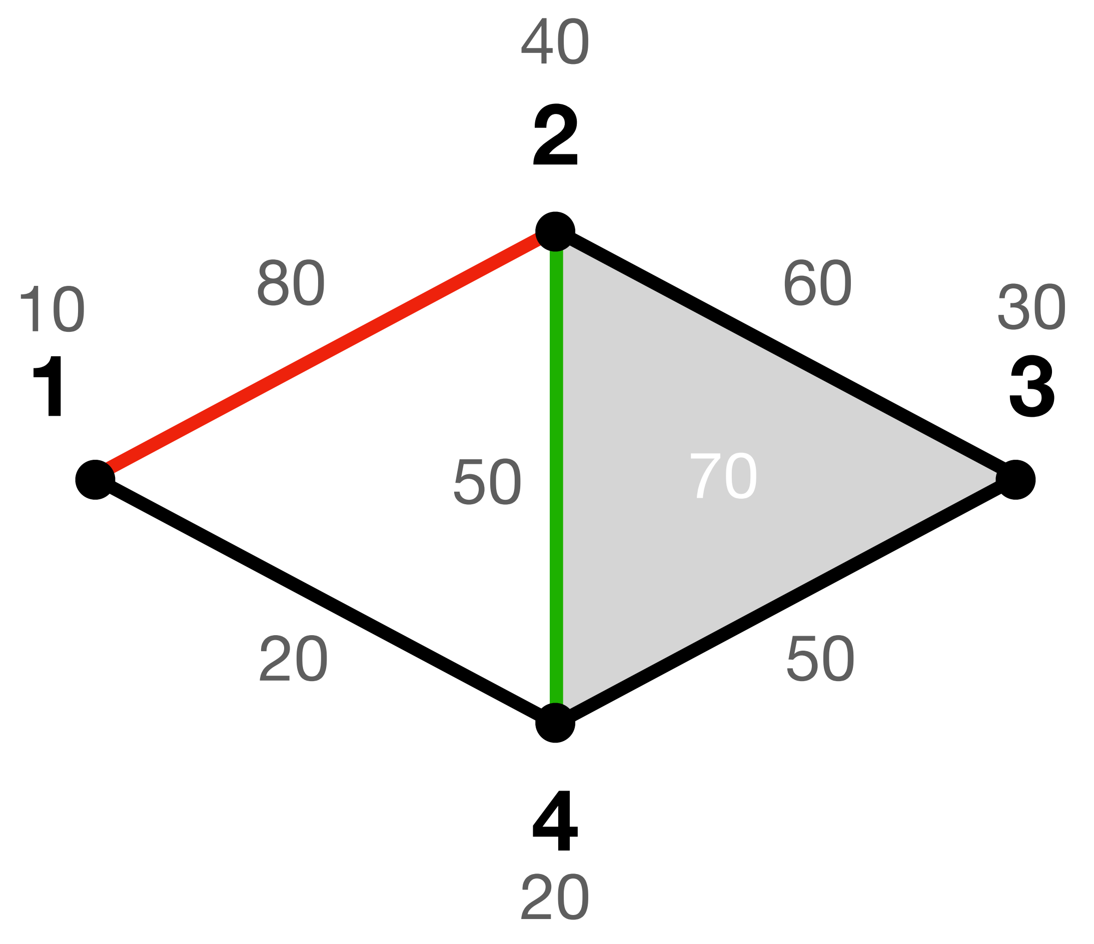
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1			
2						1				1
24									1	
34									1	
23									1	
12										

t=70

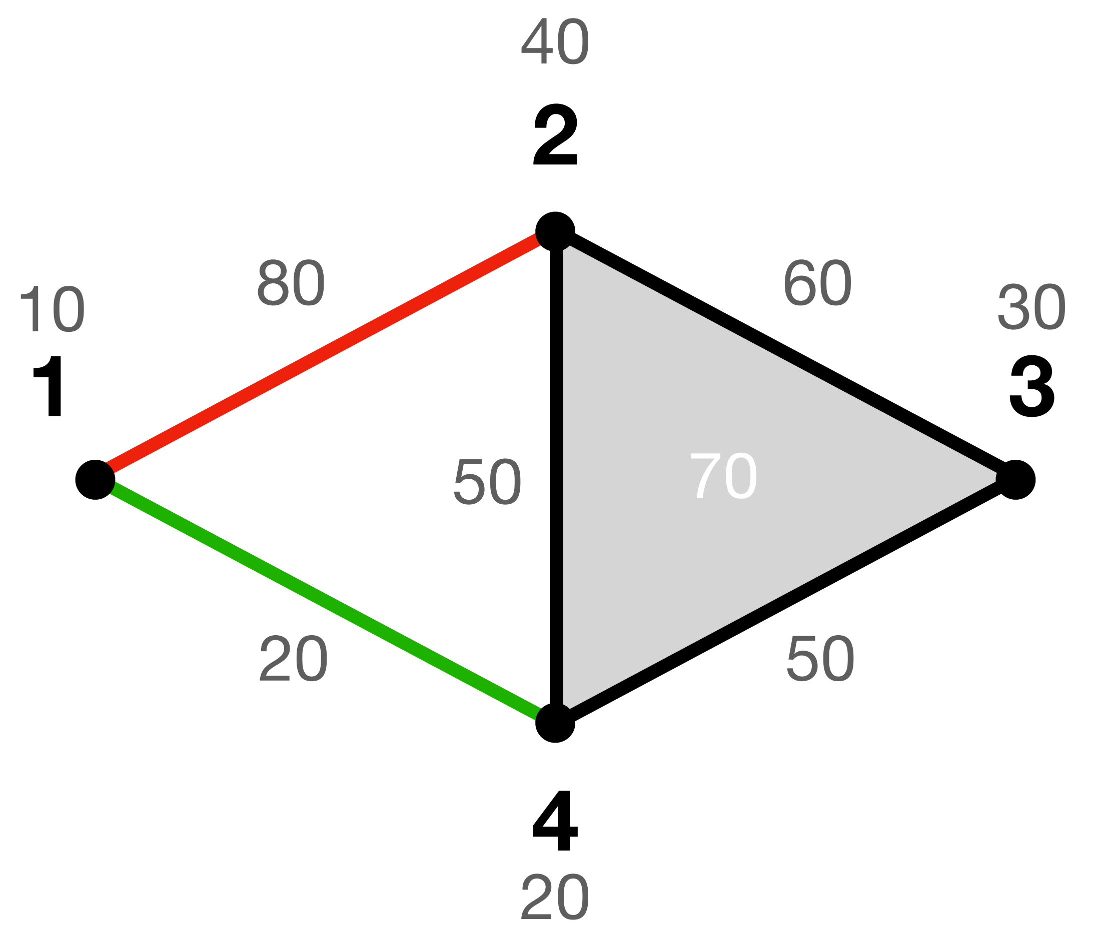
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1			
2						1				1
24									1	
34									1	
23									1	
12										

t=80

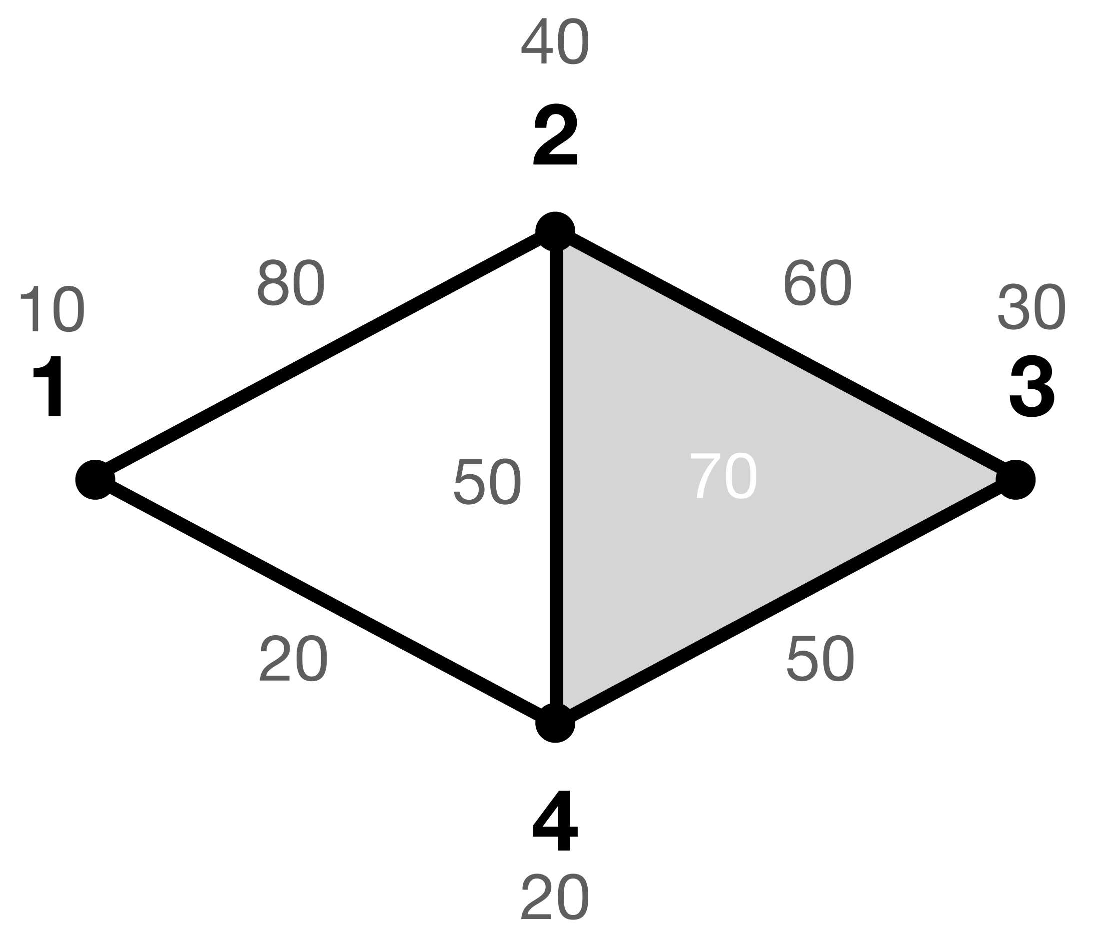
Boundary matrix reduction



	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			**1**			1	1			1
14										
3							**1**			
2						**1**				
24									1	
34									1	
23									**1**	
12										

t=80

Boundary matrix reduction

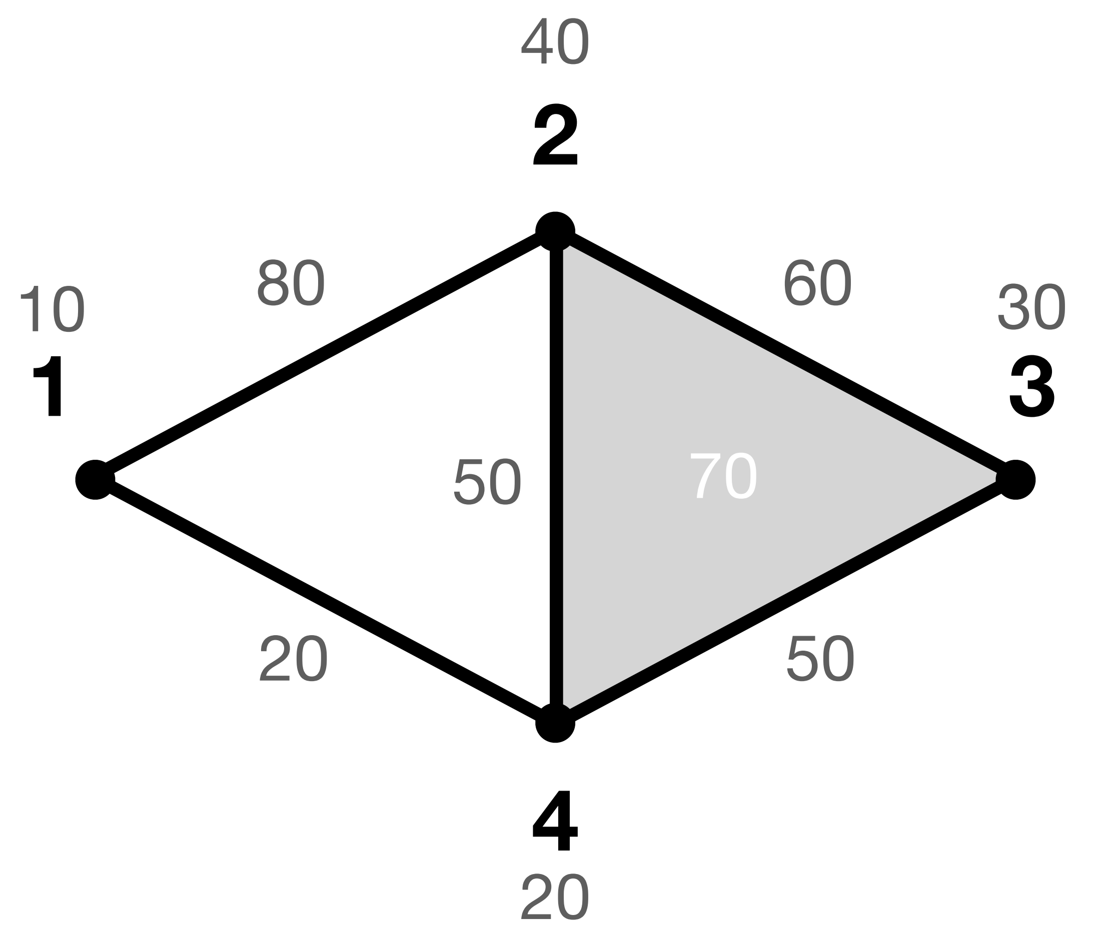


R =

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							
4			**1**			1	1			
14										
3							**1**			
2						**1**				
24									1	
34									1	
23									**1**	
12										

Matrix is called reduced if
all lowest nonzero elements are in unique rows

Persistence pairing



$\mathbf{R} =$

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							
4			1			1	1			
14										
3							1			
2						1				
24									1	
34									1	
23									1	
12										

Essential simplices correspond to unpaired empty columns

Persistence pairing

P

(4, 14) 0

(2, 24) 0

(3, 34) 0

(23, 234) 1

E

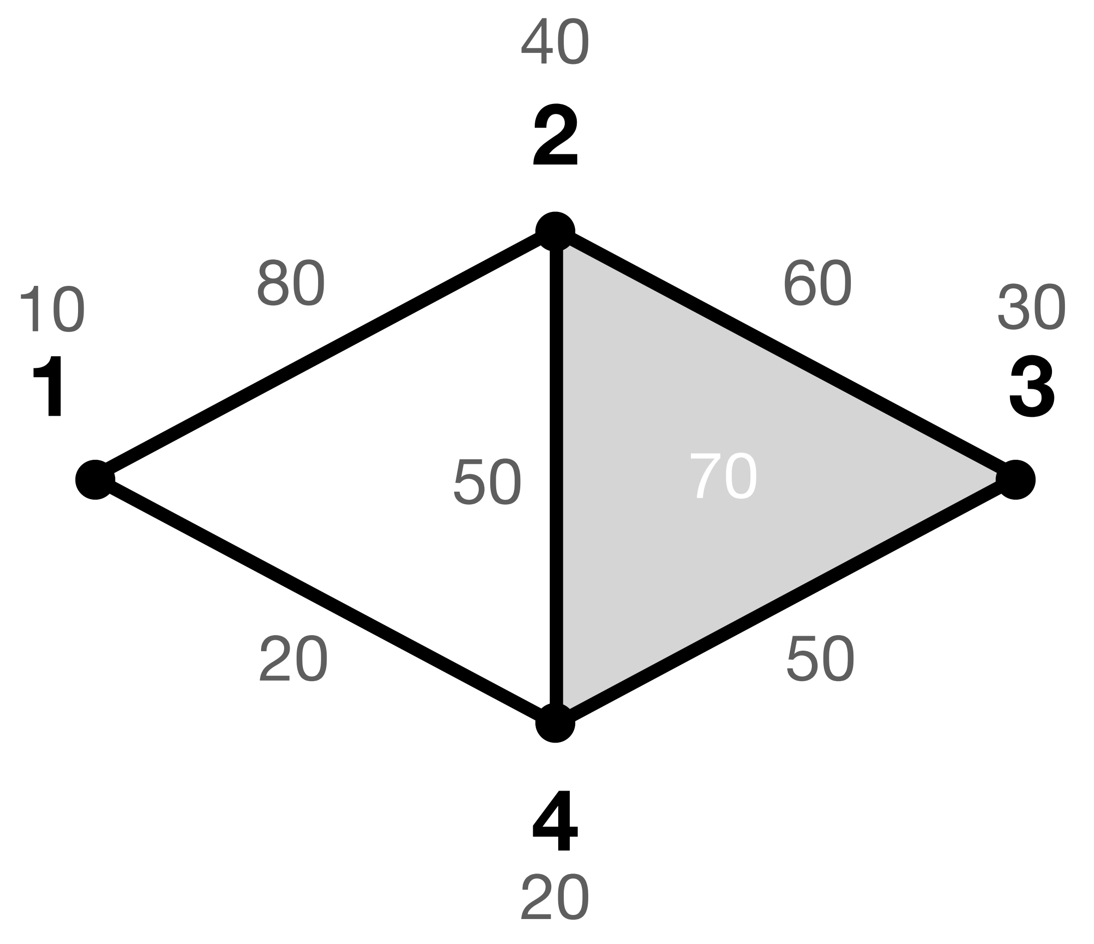
(1, $\emptyset$) 0

(12, $\emptyset$) 1

$$P = \{(i, j) \mid i = \mathrm{low}(\partial_j), \partial_j \neq 0\}$$

$$E = \{j \mid j \notin \mathrm{low}(\,\cdot\,), \partial_j = 0\}$$

Persistence diagram



$\mathbf{R} =$

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							
4			1			1	1			
14										
3							1			
2						1				
24									1	
34									1	
23									1	
12										

Persistence pairing

P

(4, 14) 0

(2, 24) 0

(3, 34) 0

(23, 234) 1

E

(1, $\varnothing$) 0

(12, $\varnothing$) 1

Persistence diagram

(20, 20) 0

(40, 50) 0

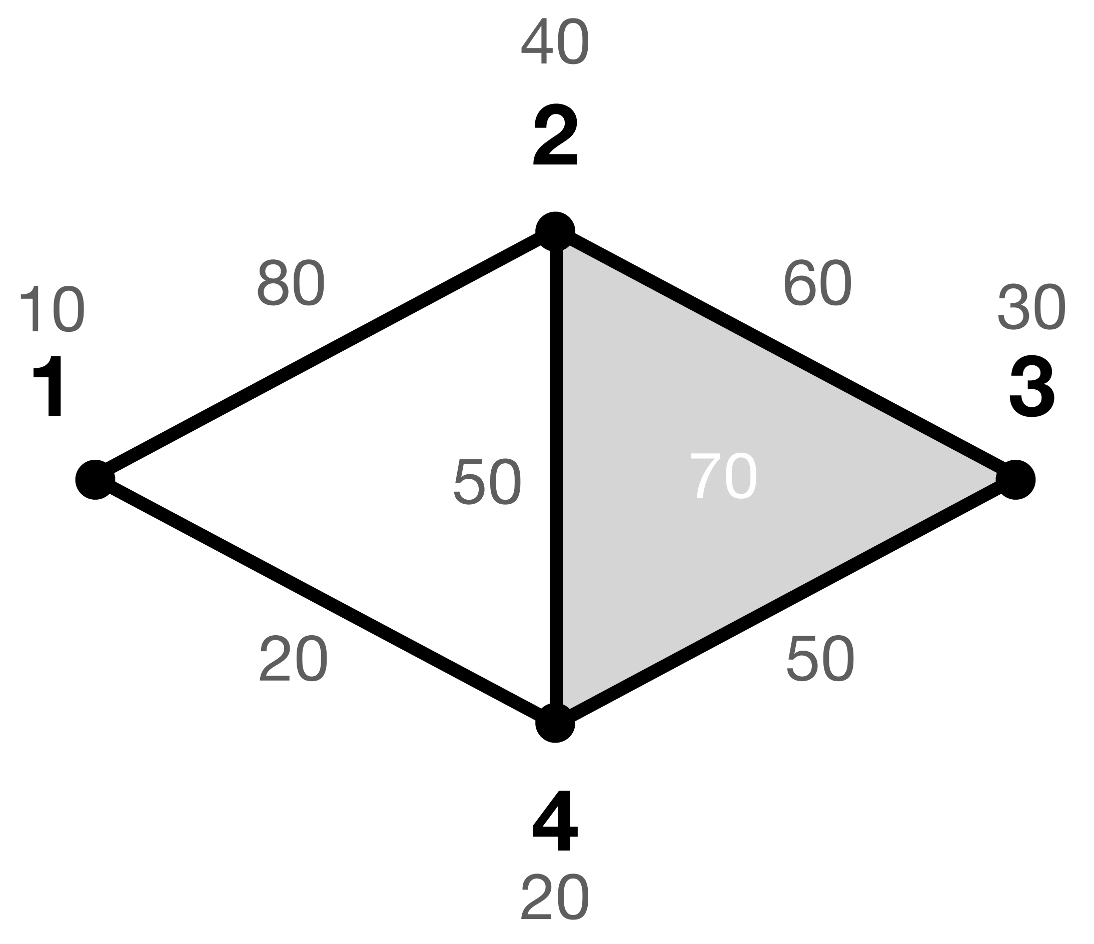
(30, 50) 0

(60, 70) 1

(10, ∞) 0

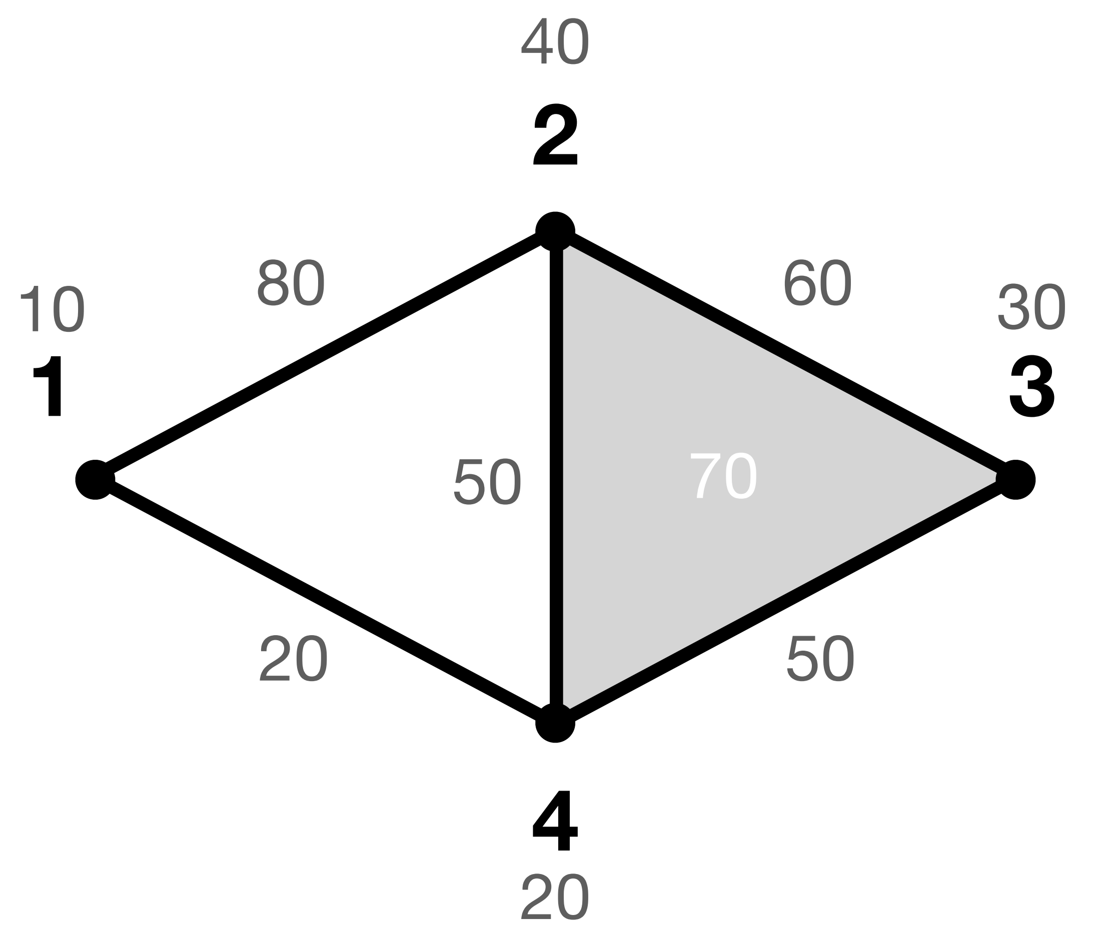
(80, ∞) 1

Coboundary matrix reduction



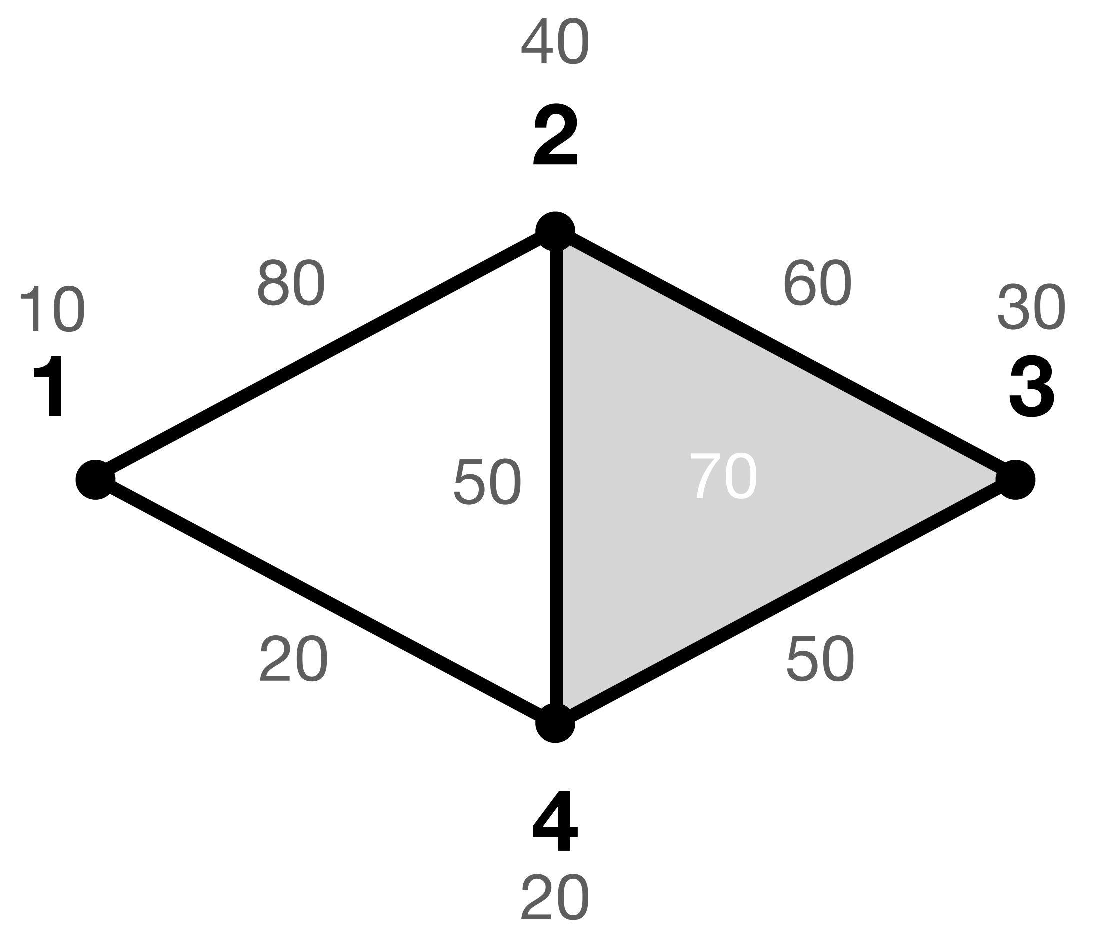
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			1	1	1					
23						1	1			
34							1		1	
24						1			1	
2										
3										
14									1	1
4										
1										

Coboundary matrix reduction



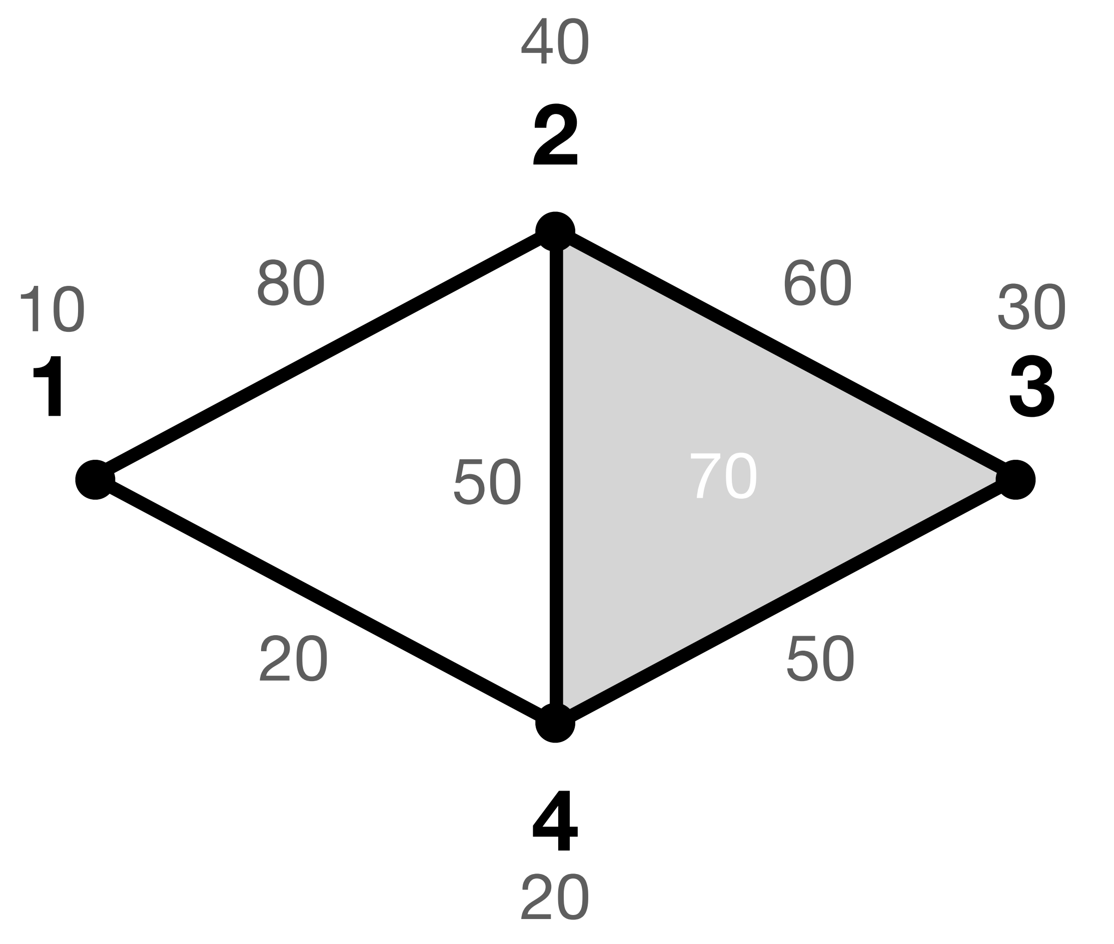
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			**1**	1	1					
23						1	1			
34							1		1	
24						1			1	
2										
3										
14									1	1
4										
1										

Coboundary matrix reduction



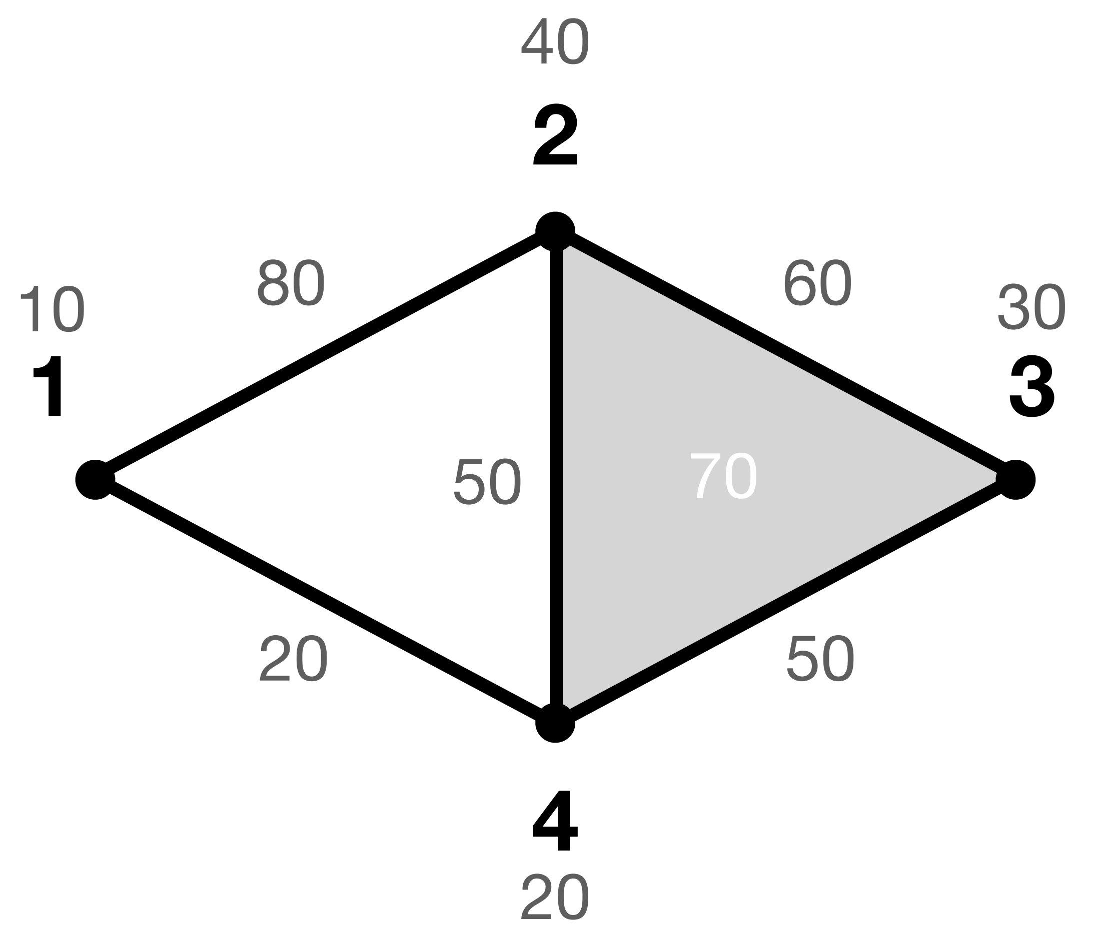
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			**1**	1	1					
23						1	1			
34							1		1	
24						1			1	
2										
3										
14									1	1
4										
1										

Coboundary matrix reduction



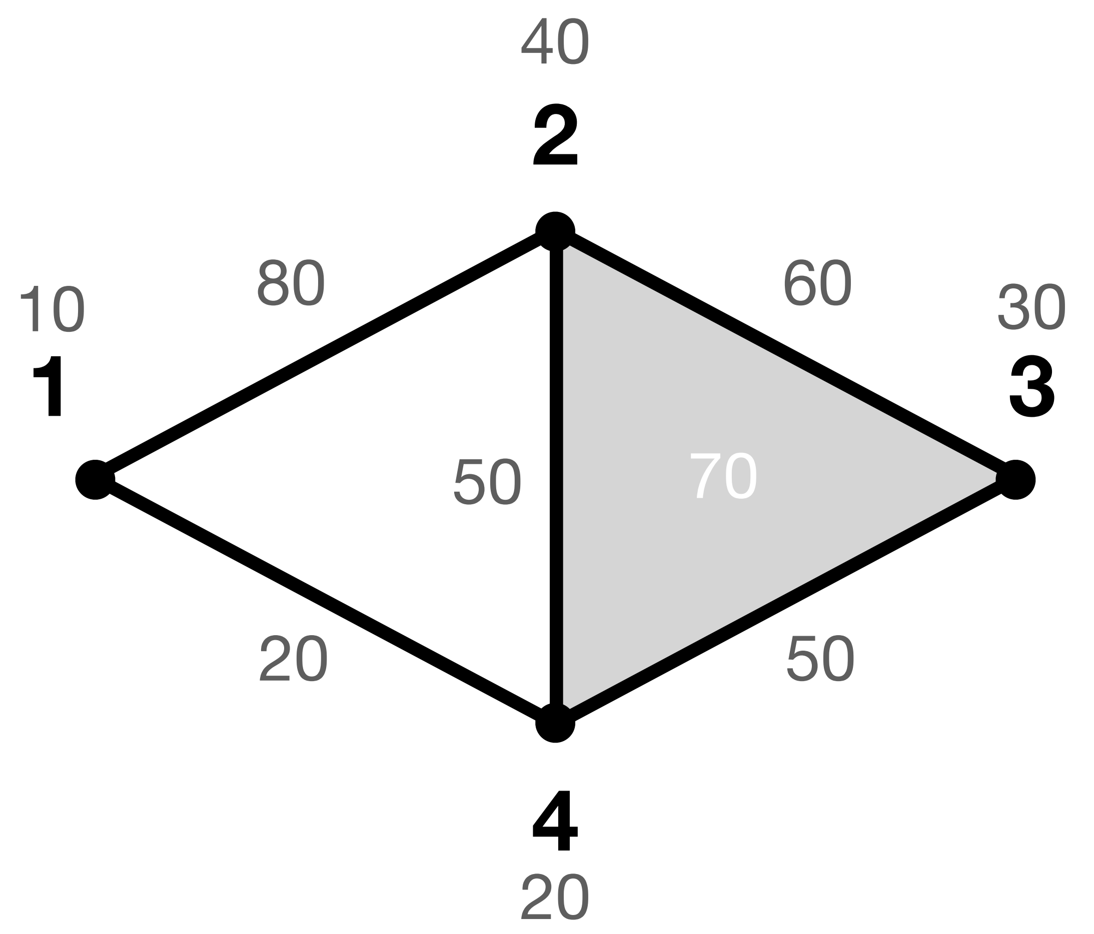
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			**1**		1					
23						1	1			
34							1		1	
24						1			1	
2										
3										
14									1	1
4										
1										

Coboundary matrix reduction



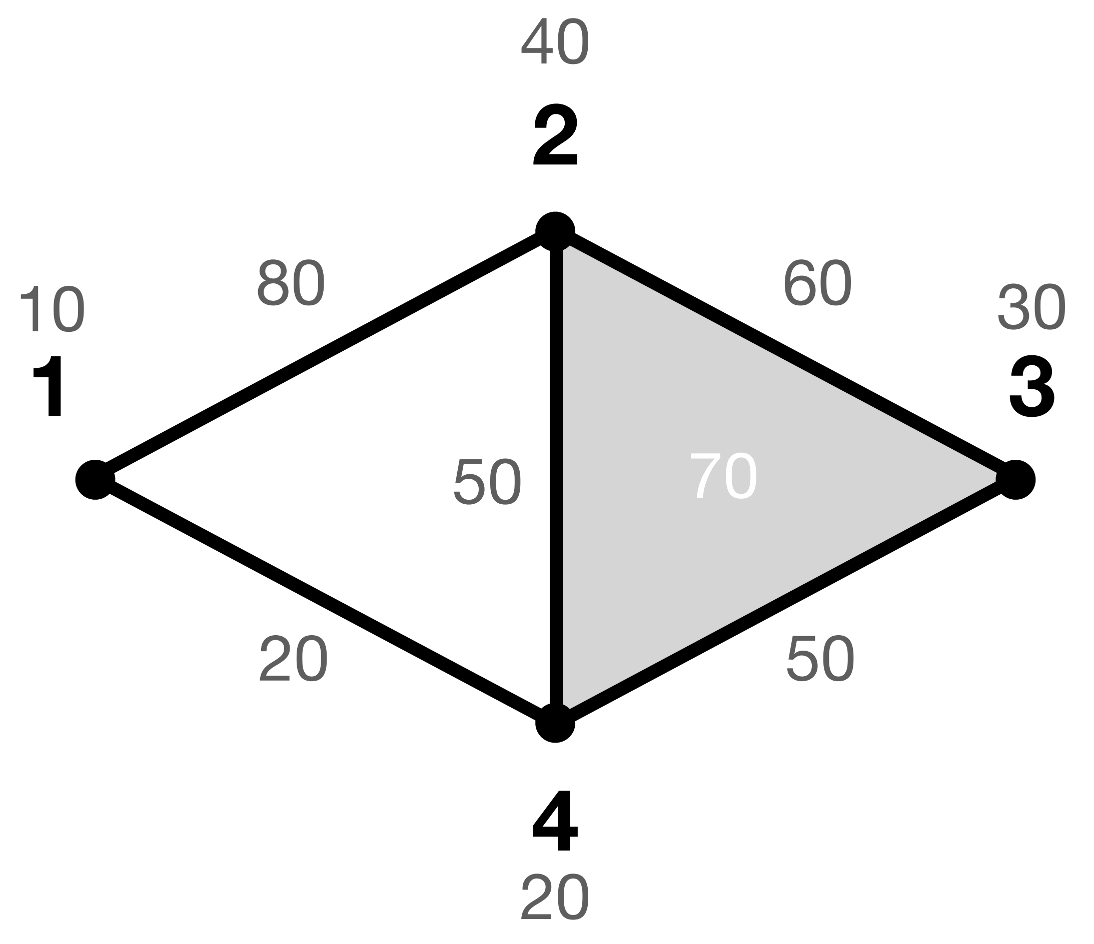
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			1							
23						1	1			
34							1		1	
24						1			1	
2										
3										
14									1	1
4										
1										

Coboundary matrix reduction



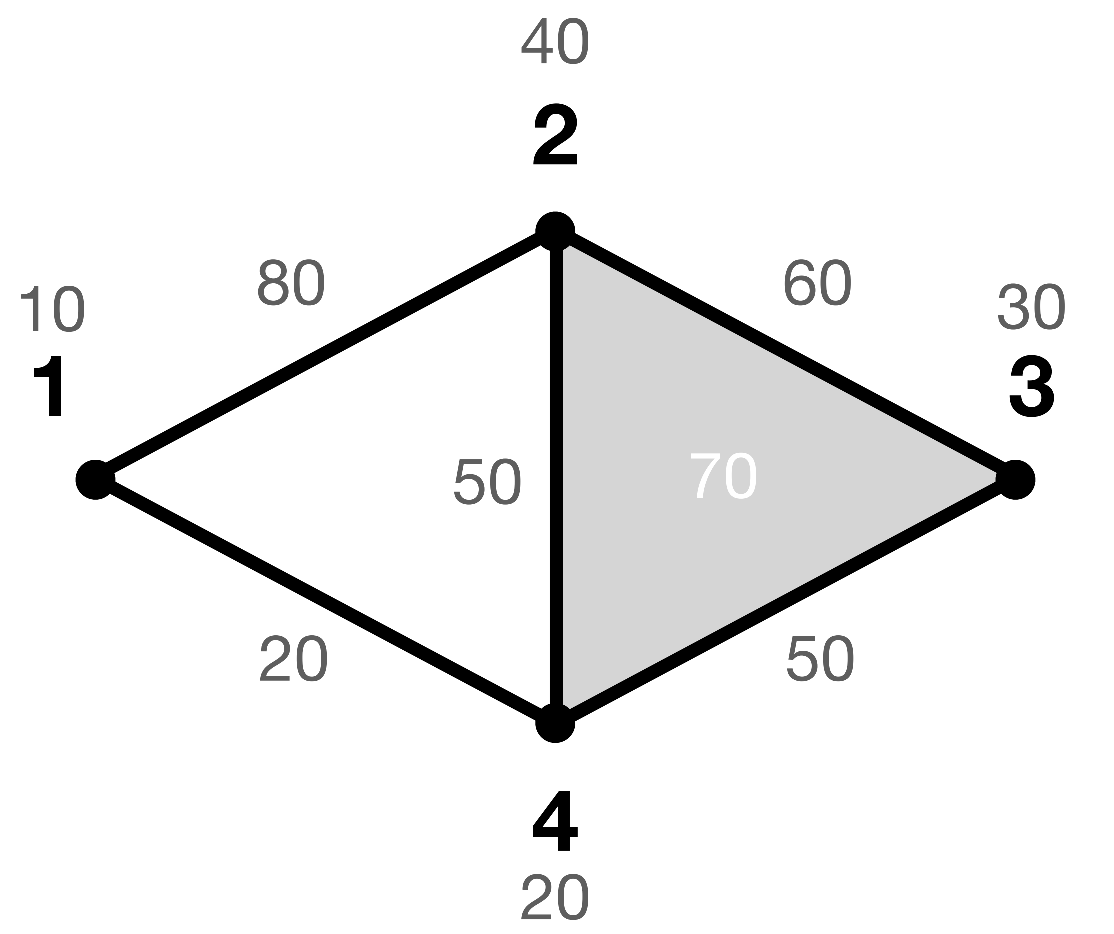
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			**1**							
23						1	1			
34							**1**		1	
24						**1**			1	
2										
3										
14									1	1
4										
1										

Coboundary matrix reduction



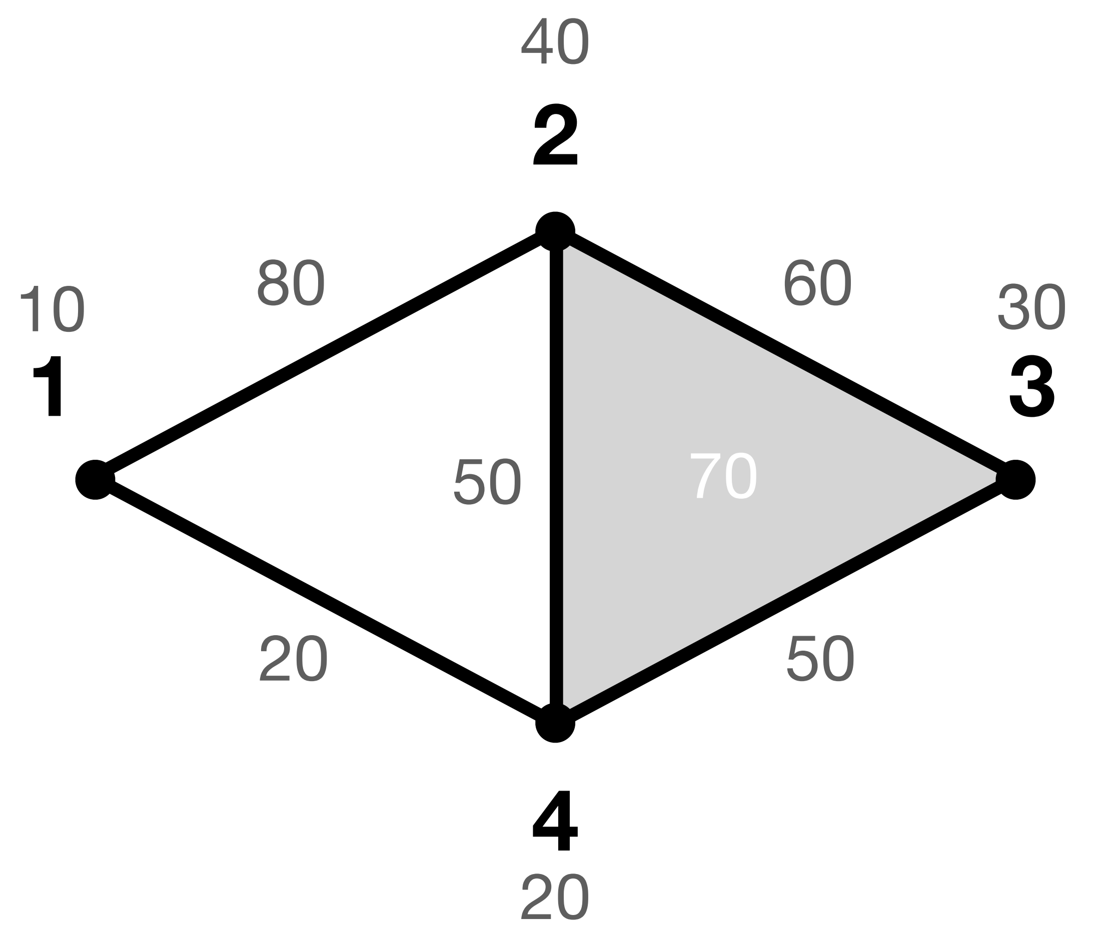
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			**1**							
23						1	1			
34							**1**		1	
24						**1**			1	
2										
3										
14									**1**	1
4										
1										

Coboundary matrix reduction



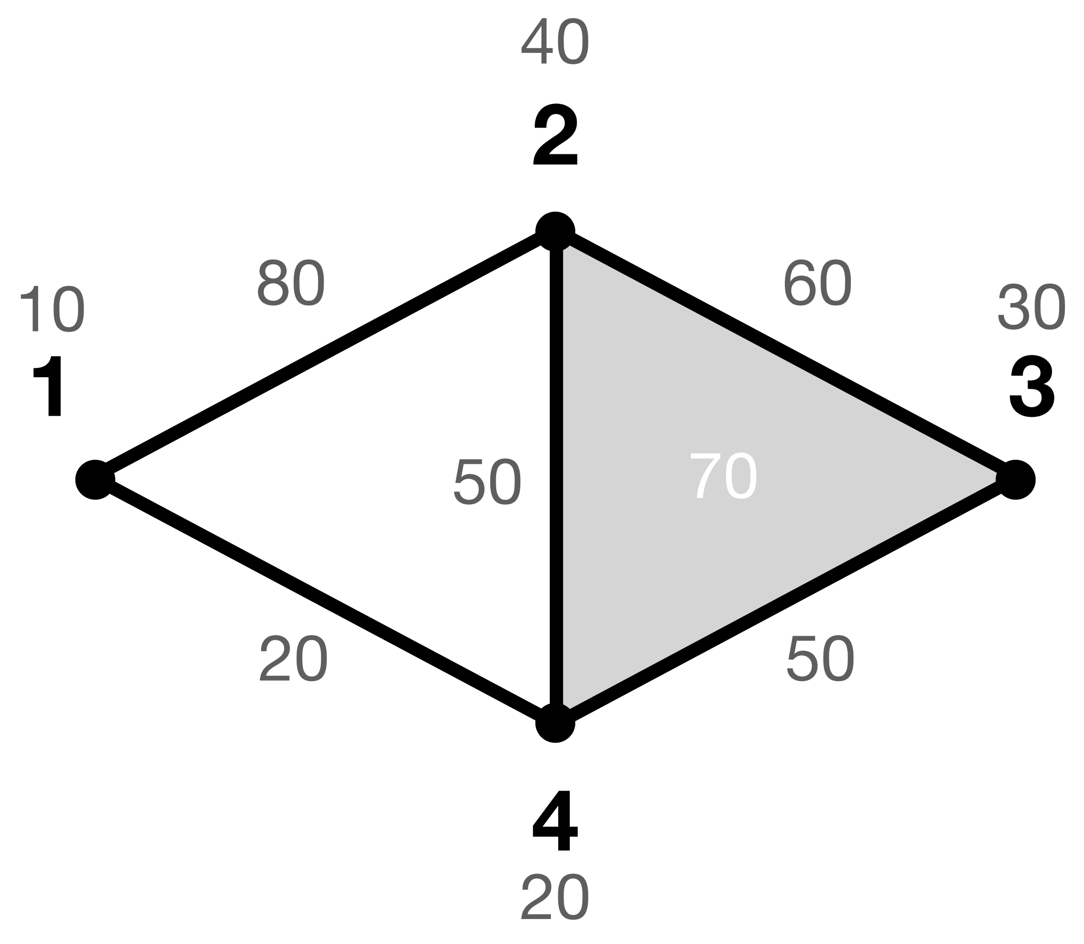
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			**1**							
23						1	1			
34							**1**		1	
24						**1**			1	
2										
3										
14									**1**	1
4										
1										

Coboundary matrix reduction



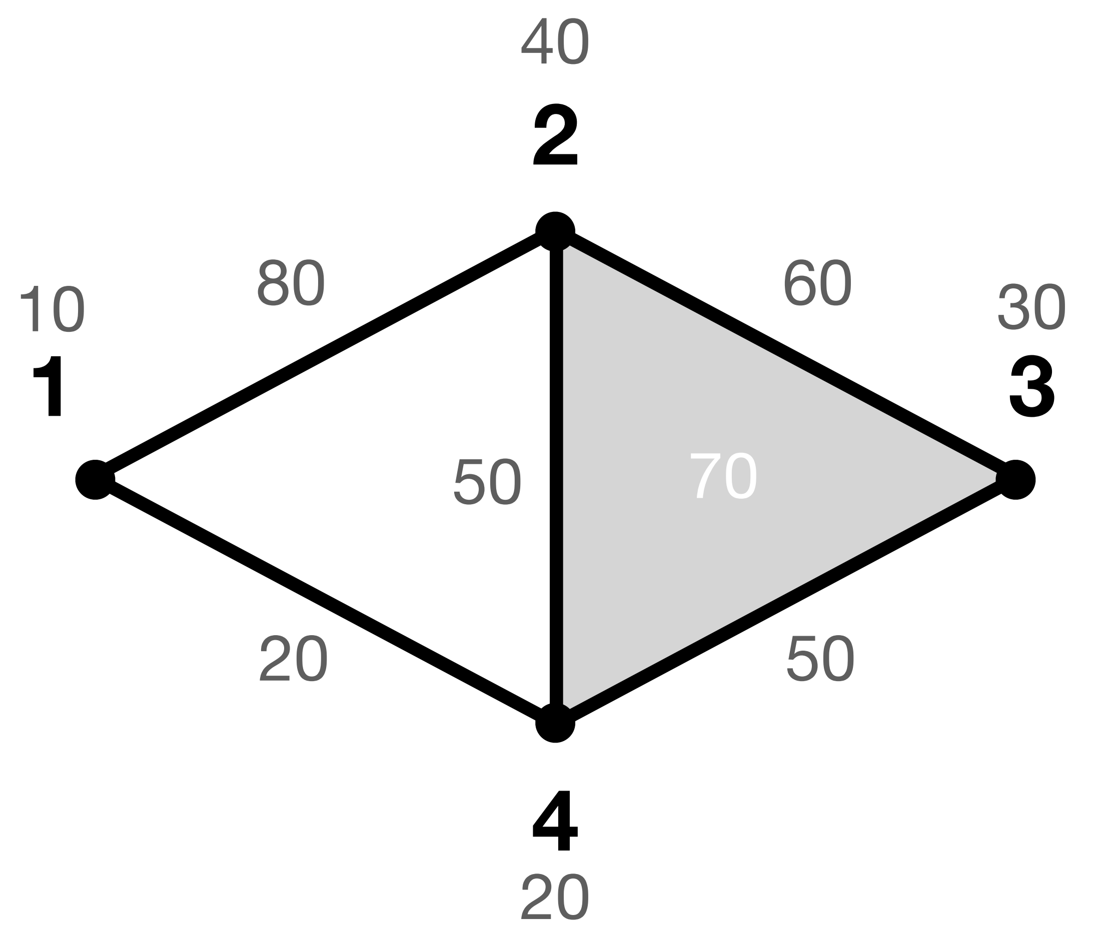
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				1
234			**1**							
23						1	1			
34							**1**		1	1
24						**1**			1	1
2										
3										
14									**1**	
4										
1										

Coboundary matrix reduction



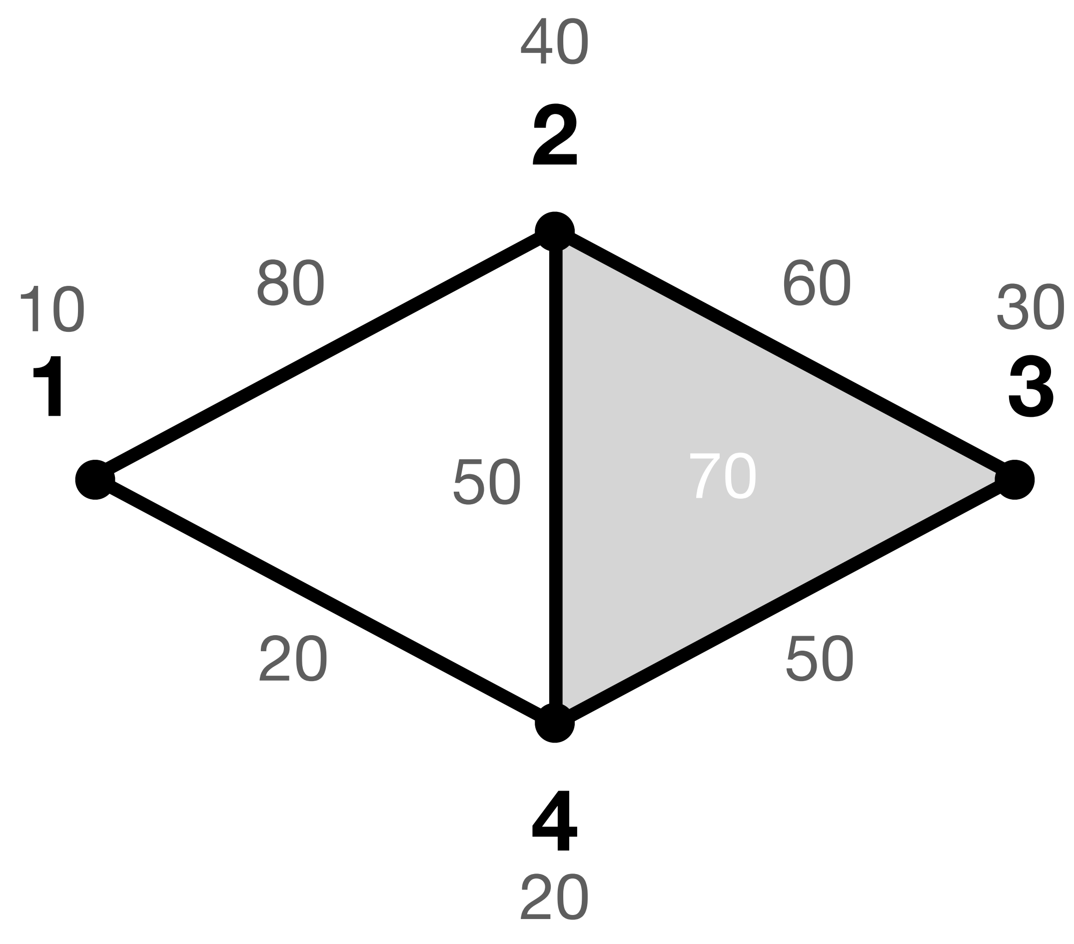
	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				
234			**1**							
23						1	1			1
34							**1**		1	1
24						**1**			1	
2										
3										
14									**1**	
4										
1										

Coboundary matrix reduction



	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				
234			**1**							
23						1	1			
34							**1**		1	
24						**1**			1	
2										
3										
14									**1**	
4										
1										

Persistence pairing



	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				
234			1							
23						1	1			
34							1		1	
24						1			1	
2										
3										
14									1	
4										
1										

Persistence pairing

P

(4, 14) 0

(2, 24) 0

(3, 34) 0

(23, 234) 1

E

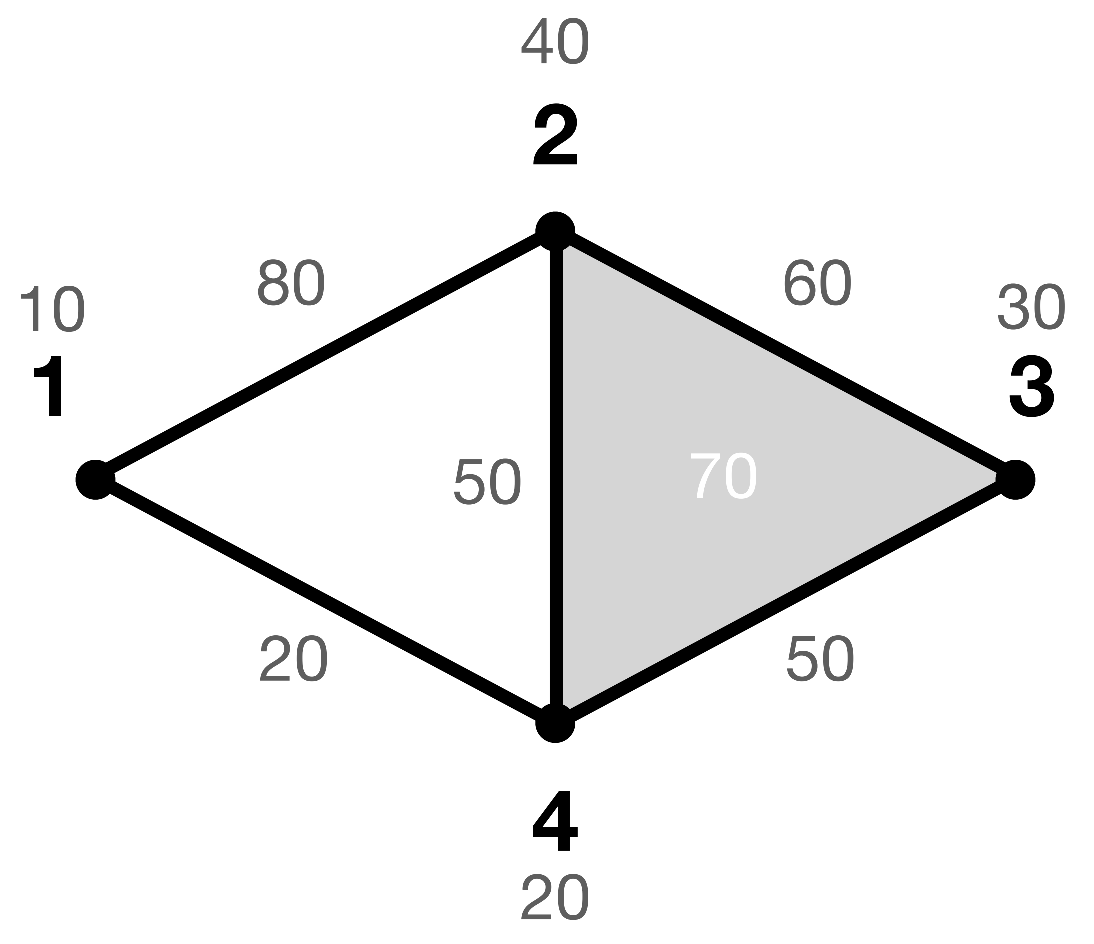
(1, Ø) 0

(12, Ø) 1

$$P = \{(j, i) \mid i = \mathrm{low}(\partial_j), \partial_j \neq 0\}$$

$$E = \{j \mid j \notin \mathrm{low}(\,\cdot\,), \partial_j = 0\}$$

Persistence diagram



	80	70	60	50	50	40	30	20	20	10
	12	**234**	**23**	**34**	**24**	**2**	**3**	**14**	**4**	**1**
12						1				
234			1							
23						1	1			
34							1		1	
24						1			1	
2										
3										
14									1	
4										
1										

Persistence pairing

P

(4, 14) 0

(2, 24) 0

(3, 34) 0

(23, 234) 1

E

(1, $\varnothing$) 0

(12, $\varnothing$) 1

Persistence diagram

(20, 20) 0

(40, 50) 0

(30, 50) 0

(60, 70) 1

(10, ∞) 0

(80, ∞) 1

Clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									**1**	
12										

Persistent Homology Computation with a Twist
Chen C., Kerber M. *European Workshop on Computational Geometry (2011)*

Clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									**1**	
12										

Persistent Homology Computation with a Twist
Chen C., Kerber M. *European Workshop on Computational Geometry (2011)*

Clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1			
2						1				1
24									1	
34									1	
23									1	
12										

Persistent Homology Computation with a Twist
Chen C., Kerber M. *European Workshop on Computational Geometry (2011)*

Clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			1
14										
3							1			
2						1				
24									1	
34									1	
23									1	
12										

Persistent Homology Computation with a Twist
Chen C., Kerber M. *European Workshop on Computational Geometry (2011)*

Clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							
4			**1**			1	1			
14										
3							**1**			
2						**1**				
24									1	
34									1	
23									**1**	
12										

Persistent Homology Computation with a Twist
Chen C., Kerber M. *European Workshop on Computational Geometry (2011)*

Parallel algorithm

Given a matrix , the leftmost columns are defined

$$L := \{\partial_j \mid \nexists k < j : \mathrm{low}(k) = \mathrm{low}(j) > -1\}$$

Given a leftmost column $\ell \in L$, its neighbors

$$N(\ell) := \{j > \ell \mid \mathrm{low}(j) = \mathrm{low}(\ell)\}$$

Algorithm 4: Parallel column additions

Input: An $n \times n$ boundary matrix ∂ over $\mathbb{Z}_2$

while ∂ *is not reduced* **do**
- $\mathcal{L} \leftarrow \{j \mid \nexists k < j : \mathrm{low}_\partial(k) = \mathrm{low}_\partial(j) > -1\}$
- **for** $\ell \in \mathcal{L}$ **do**
  - **for** $j \in \mathcal{N}(\ell)$ **do**
    - $\partial_j \leftarrow \partial_j + \partial_\ell$
  - **end for**
- **end for**

end while

return ∂

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									1	
12										

Leftmost, neighbors

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1	1		1
14										
3							1	1		
2						1				
24									1	
34									1	
23									1	
12										

Leftmost, **neighbors**

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1	1		1
14										
3							1	1		
2						1				
24									1	
34									1	
23									1	
12										

Leftmost, **neighbors**

Leftmost, **neighbors**

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							
4			1			1	1			
14										
3							1			
2						1				
24									1	
34									1	
23									1	
12										

Leftmost, **neighbors**

Leftmost, **neighbors**

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm with clearing optimization

Given a matrix , the leftmost columns are defined

$$L := \{\partial_j \mid \nexists k < j : \mathrm{low}(k) = \mathrm{low}(j) > -1\}$$

Given a leftmost column $\ell \in L$, its neighbors

$$N(\ell) := \{j > \ell \mid \mathrm{low}(j) = \mathrm{low}(\ell)\}$$

Algorithm 5: The final form of our implemented algorithm

Input: An $n \times n$ boundary matrix ∂ over $\mathbb{F}_2$

while ∂ *is not reduced* **do**
    for $j = 0, \ldots, n-1$ **do**
        Set column $\mathrm{low}_\partial(j)$ to zero
    end for
    $\mathcal{L} \leftarrow \{j \mid \nexists k < j : \mathrm{low}_\partial(k) = \mathrm{low}_\partial(j) > -1\}$
    for $\ell \in \mathcal{L}$ **do**
        for $j \in \mathcal{N}(\ell)$ **do**
            $\partial_j \leftarrow \partial_j + \partial_\ell$
        end for
    end for
end while
return ∂

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm with clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1	1		
2						1		1		1
24									1	
34									1	
23									**1**	
12										

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm with clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			
14										
3							1			
2						1				1
24									1	
34									1	
23									1	
12										

Leftmost, **neighbors**

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm with clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							1
4			1			1	1			1
14										
3							1			
2						1				
24									1	
34									1	
23									1	
12										

Leftmost, neighbors

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*

Parallel algorithm with clearing optimization

	10	20	20	30	40	50	50	60	70	80
	1	**4**	**14**	**3**	**2**	**24**	**34**	**23**	**234**	**12**
1			1							
4			**1**			1	1			
14										
3							**1**			
2						**1**				
24									1	
34									1	
23									**1**	
12										

Leftmost, neighbors

Computing Persistent Homology in Parallel in Functional Language
von Bromssen E., *Master Thesis (2021)*