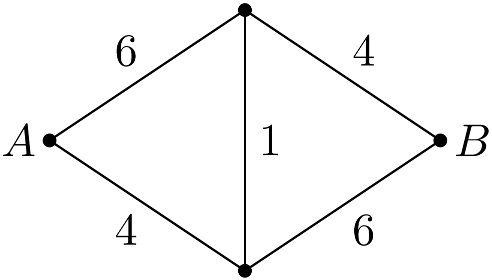
Networks, Flows and Cuts

Alexander Shen

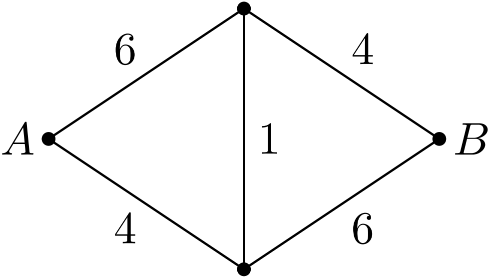
LIRMM / CNRS, University of Montpellier. France

Outline

Network: Pipes and Capacities

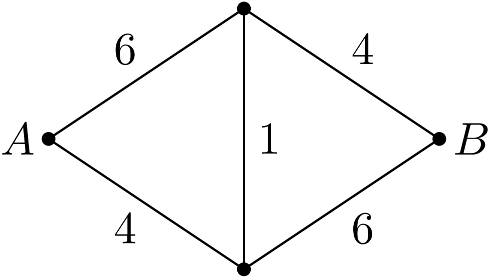


Network: Pipes and Capacities



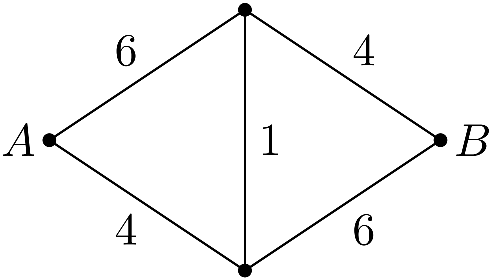
- edges = pipes

Network: Pipes and Capacities



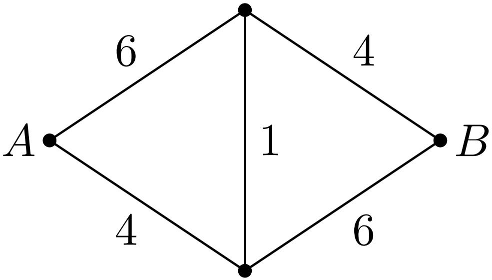
- edges = pipes
- numbers = capacities

Network: Pipes and Capacities



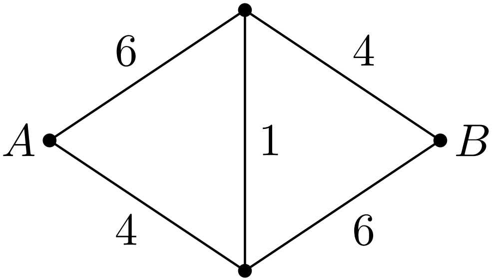
- edges = pipes
- numbers = capacities
- A: source, B: destination

Network: Pipes and Capacities



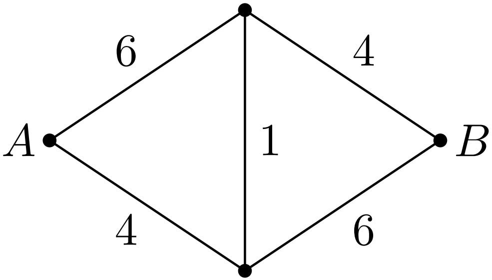
- edges = pipes
- numbers = capacities
- A: source, B: destination
- maximum flow?

Network: Pipes and Capacities



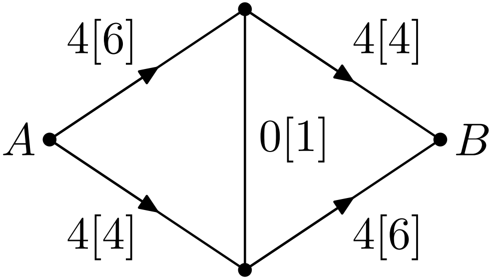
- edges = pipes
- numbers = capacities
- A: source, B: destination
- maximum flow? 10?

Network: Pipes and Capacities

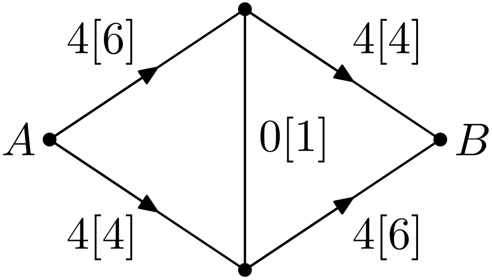


- edges = pipes
- numbers = capacities
- A: source, B: destination
- maximum flow? 10? not really

Flow: 8 Is Possible

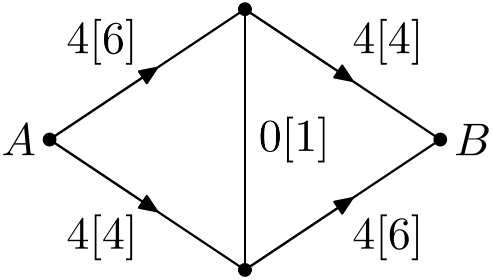


Flow: 8 Is Possible



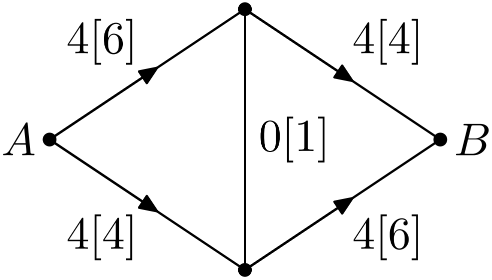
- flow[capacity]

Flow: 8 Is Possible



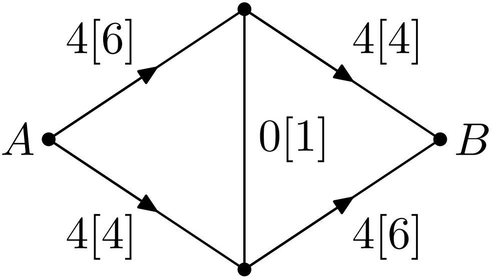
- flow[capacity]
- no oil is spilled (in = out)

Flow: 8 Is Possible



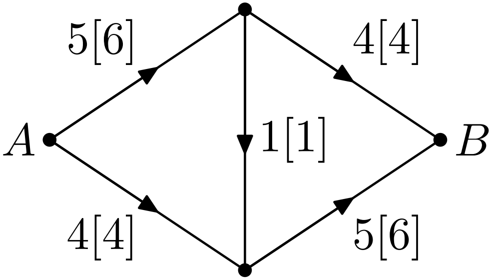
- flow[capacity]
- no oil is spilled (in = out)
- flow (from *A* to *B*): 8

Flow: 8 Is Possible

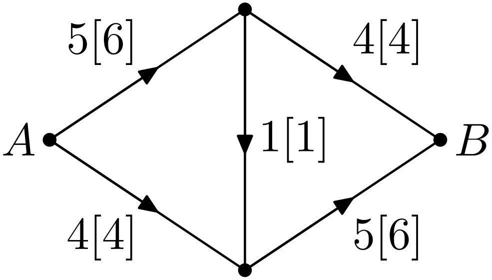


- flow[capacity]
- no oil is spilled (in = out)
- flow (from *A* to *B*): 8
- maximum flow?

Flow: 9 Is Possible

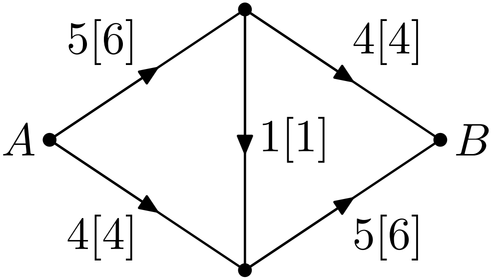


Flow: 9 Is Possible



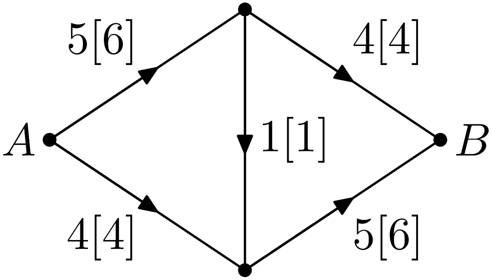
- flow (from *A* to *B*): 9

Flow: 9 Is Possible



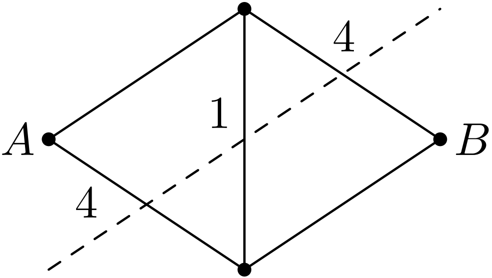
- flow (from A to B): 9
- maximum flow?

Flow: 9 Is Possible

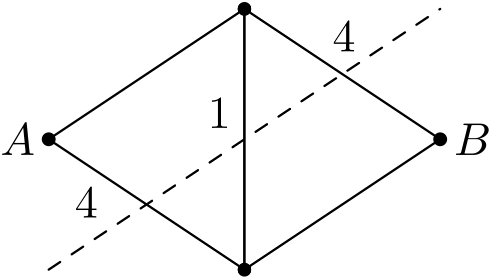


- flow (from A to B): 9
- maximum flow?
- 9 is possible, not more than 10

Why 9 Is Maximal

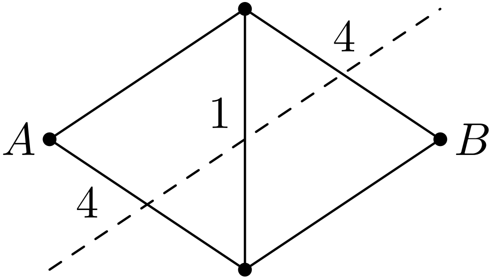


Why 9 Is Maximal



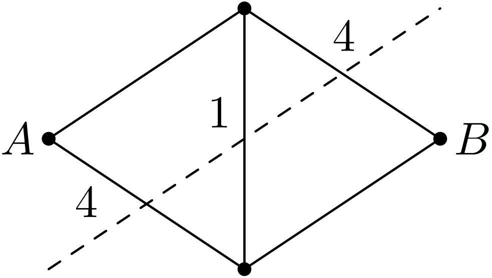
- dashed line is a *cut*

Why 9 Is Maximal



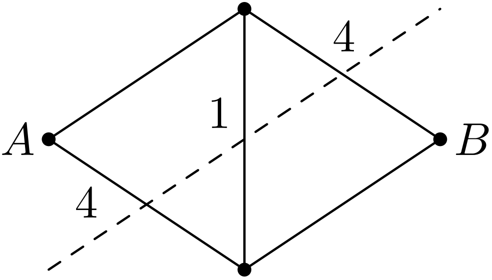
- dashed line is a *cut*
- pipes of capacities $4, 1, 4$ cross it

Why 9 Is Maximal



- dashed line is a *cut*
- pipes of capacities $4, 1, 4$ cross it
- no more than $4 + 1 + 4 = 9$

Why 9 Is Maximal



- dashed line is a *cut*
- pipes of capacities $4, 1, 4$ cross it
- no more than $4 + 1 + 4 = 9$
- 9 is indeed maximal

Our Goal

Our Goal

- network
  (=graph + edge capacities +
  source/destination)

Our Goal

- network
  (=graph + edge capacities +
  source/destination)
- maximal flow?

Our Goal

- network
  (=graph + edge capacities +
  source/destination)
- maximal flow?
- example

Our Goal

- network
  (=graph + edge capacities +
  source/destination)
- maximal flow?
- example
- maximality proof

Our Goal

- network
  (=graph + edge capacities +
  source/destination)
- maximal flow?
- example
- maximality proof (using cuts)

Our Goal

- network
  (=graph + edge capacities +
  source/destination)
- maximal flow?
- example
- maximality proof (using cuts)
- Ford and Fulkerson (1956)

Our Goal

- network
  (=graph + edge capacities +
  source/destination)
- maximal flow?
- example
- maximality proof (using cuts)
- Ford and Fulkerson (1956)
- simplification: integer capacities

Outline

Network: Capacities

Network: Capacities

- vertices $1, 2, \ldots n$

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$
  integers; $c[i, j] \geq 0$

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$
  integers; $c[i, j] \geq 0$; (no pipe = 0)

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$
  integers; $c[i, j] \geq 0$; (no pipe = 0)
- $c[i, j] \neq c[j, i]$ allowed

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$
  integers; $c[i, j] \geq 0$; (no pipe = 0)
- $c[i, j] \neq c[j, i]$ allowed
  generalization needed for the analysis

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$

  integers; $c[i, j] \geq 0$; (no pipe = 0)
- $c[i, j] \neq c[j, i]$ allowed

  generalization needed for the analysis
- two edges $i \to j$?

Network: Capacities

- vertices $1, 2, \dots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$
  integers; $c[i, j] \geq 0$; (no pipe = 0)
- $c[i, j] \neq c[j, i]$ allowed
  generalization needed for the analysis
- two edges $i \to j$? not allowed,
  but does not matter

Network: Capacities

- vertices $1, 2, \ldots n$
- source A, destination B
- capacities $c[i, j]$ for $i \to j$
  integers; $c[i, j] \geq 0$; (no pipe = 0)
- $c[i, j] \neq c[j, i]$ allowed
  generalization needed for the analysis
- two edges $i \to j$? not allowed,
  but does not matter
- $c[i, i] = 0$ for convenience

Network: Flows

Network: Flows

- flow $f[i, j]$ for $i \to j$

Network: Flows

- flow $f[i,j]$ for $i \to j$
- $f[i,j] \leq c[i,j]$: restriction

Network: Flows

- flow $f[i,j]$ for $i \to j$
- $f[i,j] \leq c[i,j]$: restriction
- $f[i,j] = -f[j,i]$

Network: Flows

- flow $f[i, j]$ for $i \to j$
- $f[i, j] \leq c[i, j]$: restriction
- $f[i, j] = -f[j, i]$ (electric current)

Network: Flows

- flow $f[i, j]$ for $i \to j$
- $f[i, j] \leq c[i, j]$: restriction
- $f[i, j] = -f[j, i]$ (electric current)
- implies $f[i, i] = 0$

Network: Flows

- flow $f[i,j]$ for $i \to j$
- $f[i,j] \leq c[i,j]$: restriction
- $f[i,j] = -f[j,i]$ (electric current)
- implies $f[i,i] = 0$
- no spill condition

Network: Flows

- flow $f[i, j]$ for $i \to j$
- $f[i, j] \leq c[i, j]$: restriction
- $f[i, j] = -f[j, i]$ (electric current)
- implies $f[i, i] = 0$
- no spill condition

  $\sum_j f[i, j] = 0$ for every i

Network: Flows

- flow $f[i, j]$ for $i \to j$
- $f[i, j] \leq c[i, j]$: restriction
- $f[i, j] = -f[j, i]$ (electric current)
- implies $f[i, i] = 0$
- no spill condition
  $\sum_j f[i, j] = 0$ for every i
- …except for A and B

Network: Flows

- flow $f[i,j]$ for $i \to j$
- $f[i,j] \leq c[i,j]$: restriction
- $f[i,j] = -f[j,i]$ (electric current)
- implies $f[i,i] = 0$
- no spill condition
  $\sum_j f[i,j] = 0$ for every i
- …except for A and B
- total flow:
  $\sum_j f[A,j] = \sum_i f[i,B]$

Network: Cuts

Network: Cuts

- how to prove that no flow exceeds *c*?

Network: Cuts

- how to prove that no flow exceeds c?
- find a cut C of total capacity c

Network: Cuts

- how to prove that no flow exceeds c?
- find a cut C of total capacity c
- C: some set of nodes that contains A but not B

Network: Cuts

- how to prove that no flow exceeds c?
- find a cut C of total capacity c
- C: some set of nodes that contains A but not B
- *total capacity* of a cut: sum of capacities of all outgoing edges ($C \to$ not C)

Network: Cuts

- how to prove that no flow exceeds c?
- find a cut C of total capacity c
- C: some set of nodes that contains A but not B
- *total capacity* of a cut: sum of capacities of all outgoing edges ($C \rightarrow$ not C)
- obvious: any total flow never exceeds the total capacity of any cut

Network: Cuts

- how to prove that no flow exceeds c?
- find a cut C of total capacity c
- C: some set of nodes that contains A but not B
- *total capacity* of a cut: sum of capacities of all outgoing edges ($C \rightarrow$ not C)
- obvious: any total flow never exceeds the total capacity of any cut
- Ford–Fulkerson: the equality happens for some flow and some cut

What Ford and Fulkerson Say

What Ford and Fulkerson Say

obvious:

What Ford and Fulkerson Say

obvious:

What Ford and Fulkerson Say

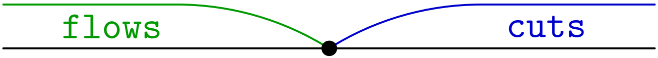
obvious:

Ford–Fulkerson:

What Ford and Fulkerson Say

obvious:

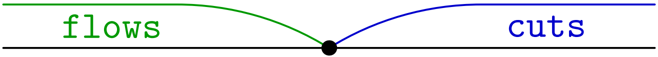
Ford–Fulkerson:



What Ford and Fulkerson Say

obvious:

Ford–Fulkerson:



Theorem: maximal flow = minimal cut