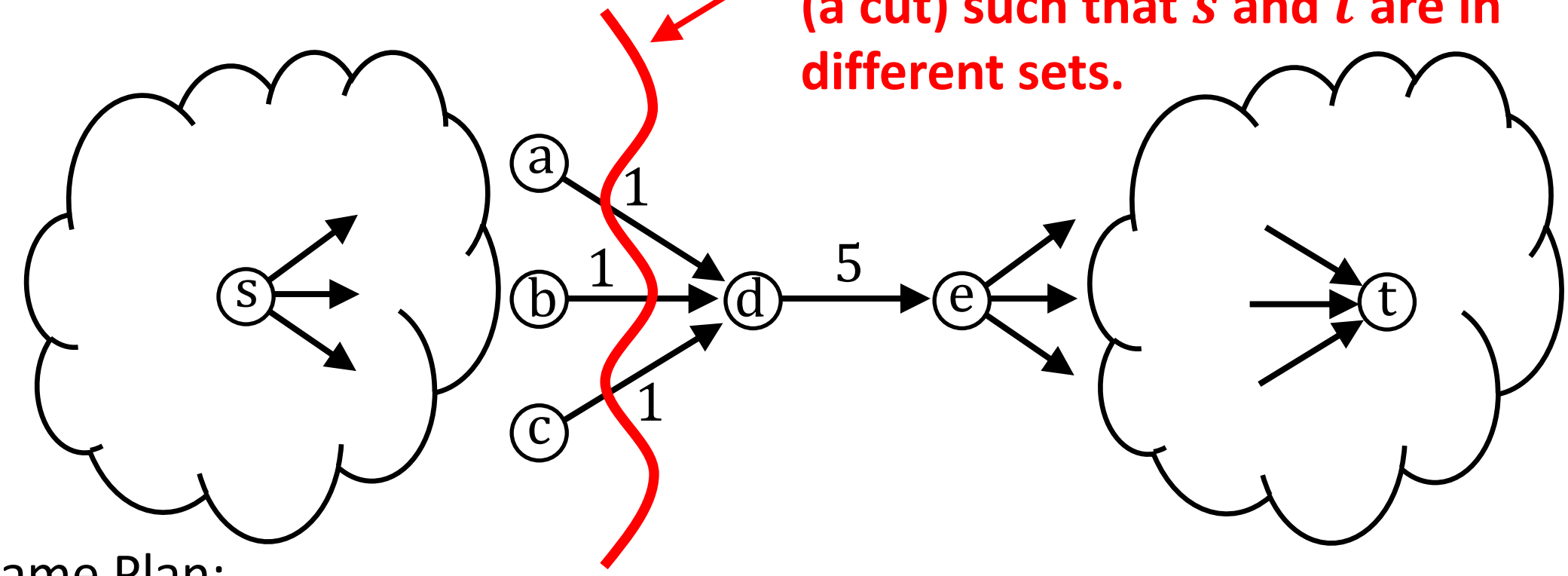
Flow Networks
CSCI 532

Optimality



Game Plan:

1. Show that value of every flow is $\leq$ capacity of every cut.
2. Given a flow where there are no $s-t$ paths left in the residual graph, there is a specific cut whose capacity = flow value.

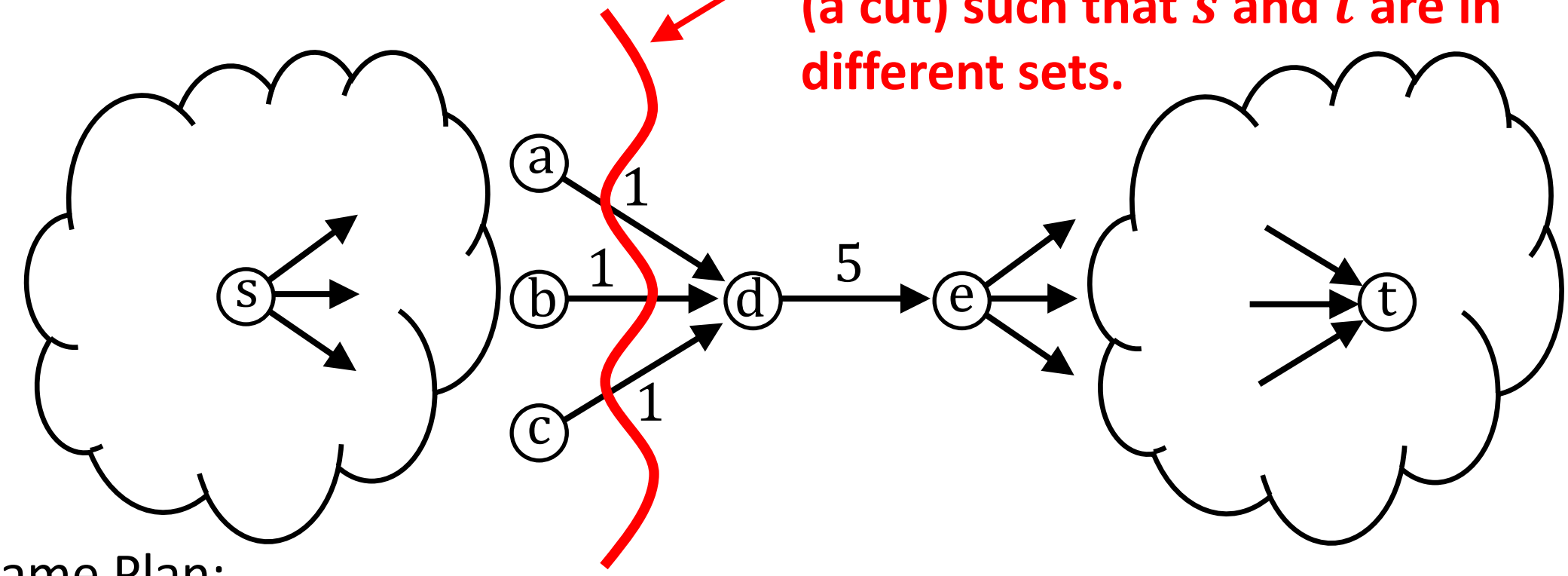
$\Rightarrow$ **The algorithm is optimal**

Optimality

Theorem 1: Let G be a flow network, (A, B) be an $s-t$ cut, and f be an $s-t$ flow. Then, $|f| = \sum_{e \in \text{out}(A)} f(e) - \sum_{e \in \text{in}(A)} f(e)$.

Corollary: Suppose G is a flow network, f is an $s-t$ flow on G, and (A, B) is an $s-t$ cut. Then, $|f| \leq c(A, B)$. (i.e. *every* flow is bounded by *any* $s-t$ cut)

Optimality



Game Plan:

1. Show that value of every flow is $\leq$ capacity of every cut.
2. Given a flow where there are no $s - t$ paths left in the residual graph, there is a specific cut whose capacity = flow value.

Optimality

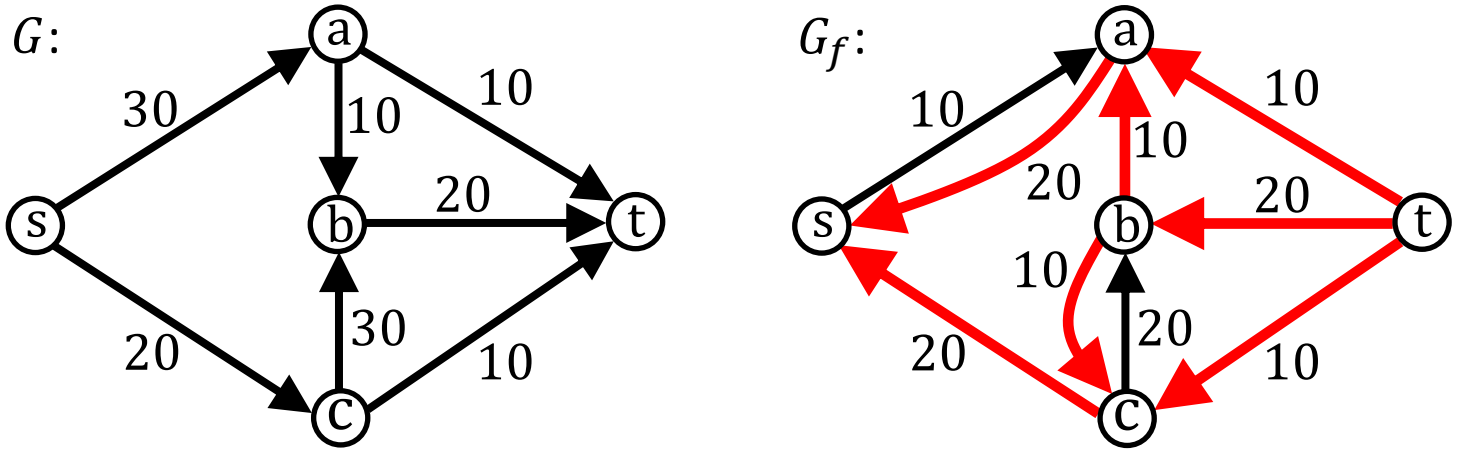
Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A, B) in $G = (V, E)$ for which $|f| = c(A, B)$.

Proof:

Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

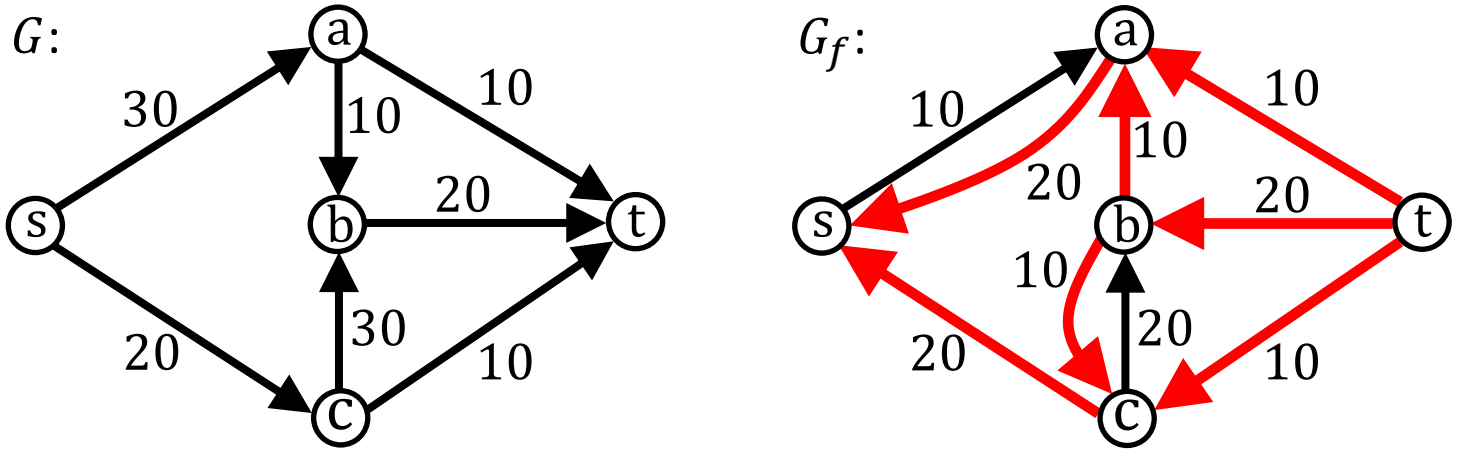
Proof:



Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

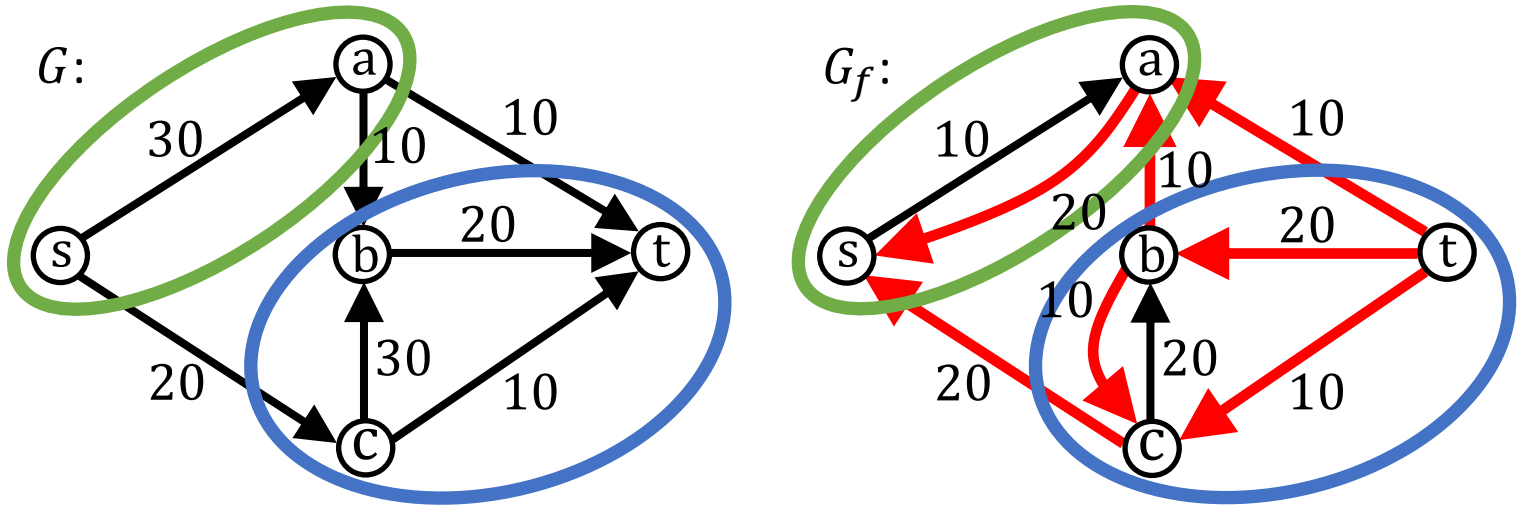
Proof: Let $A=\{v\in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V\backslash A$.



Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

Proof: Let $A=\{v\in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V\backslash A$.

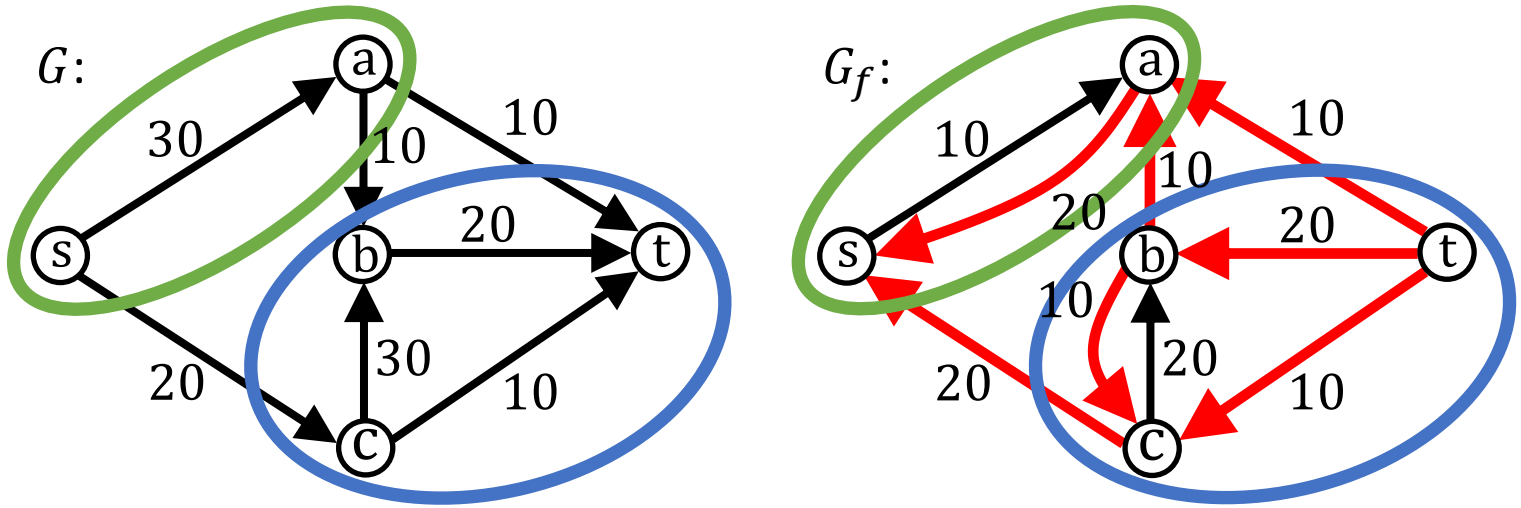


Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

Proof: Let $A=\{v\in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V\backslash A$.

(A,B) is an $s-t$ cut (because it partitions V, $s\in A$, and $t\in B$)



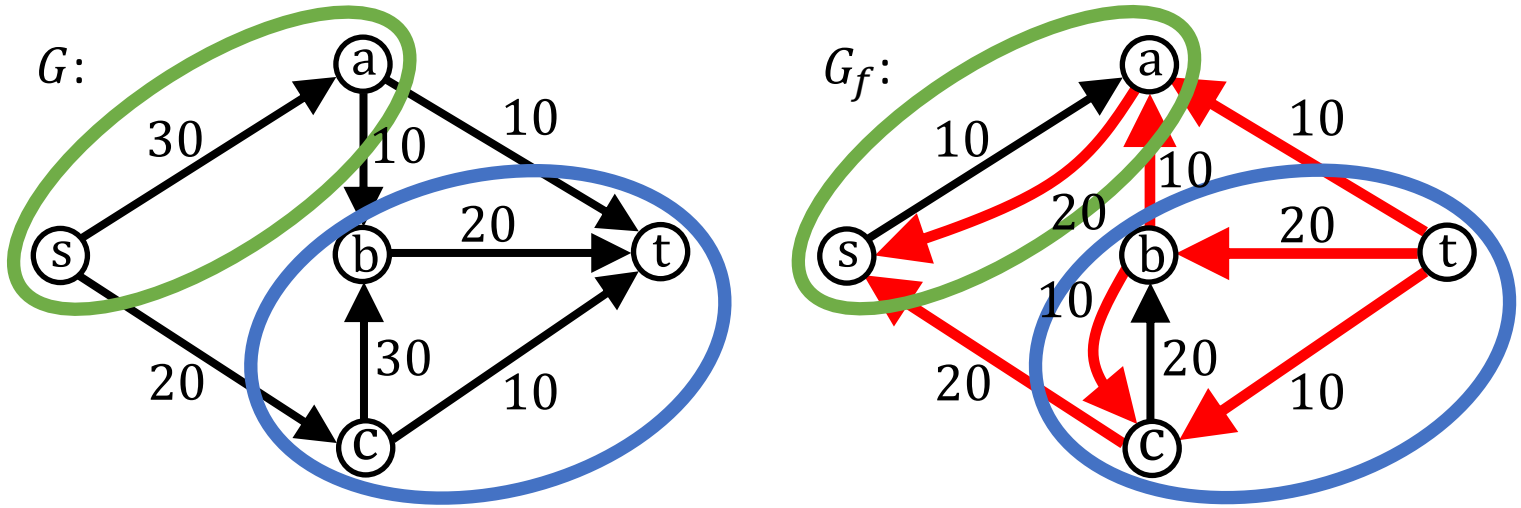
Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

Proof: Let $A=\{v\in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V\backslash A$.

(A,B) is an $s-t$ cut (because it partitions V, $s\in A$, and $t\in B$)

Need to compare flow across cut to capacity of cut.



Optimality

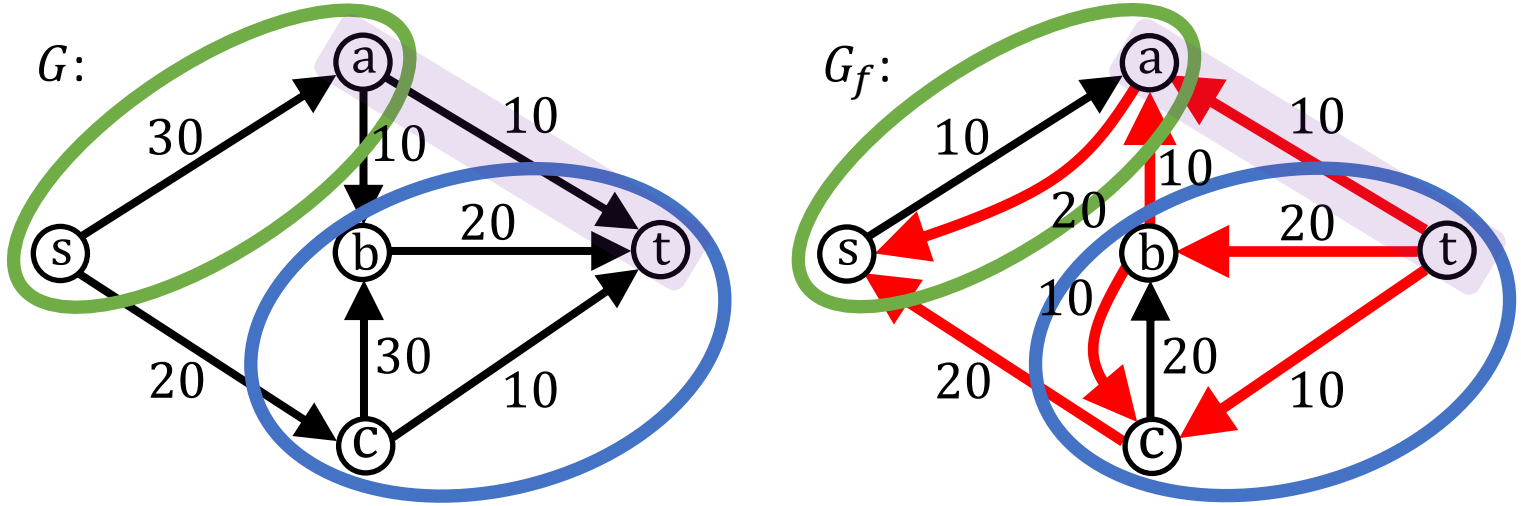
Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

Proof: Let $A=\{v\in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V\backslash A$.

(A,B) is an $s-t$ cut (because it partitions V, $s\in A$, and $t\in B$)

Let $e=(u,v)\in E$ (directed edge) such that $u\in A$ and $v\in B$.

What can we say about $f(e)$ related to its capacity?



Optimality

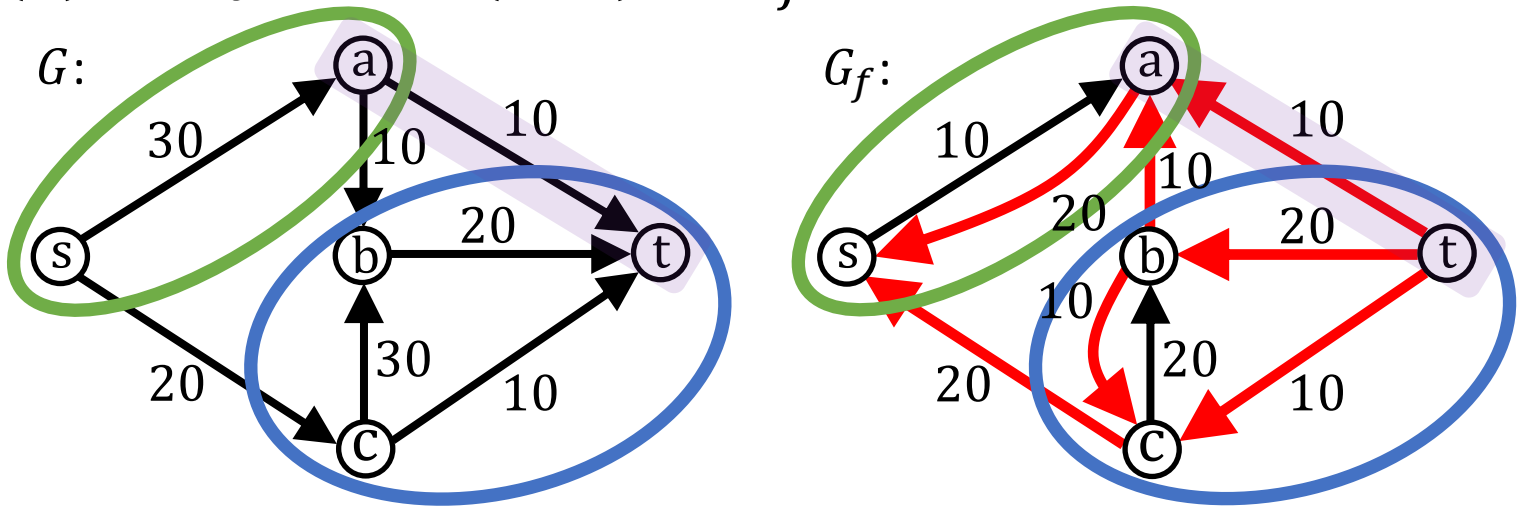
Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

Proof: Let $A=\{v \in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V \backslash A$.

(A,B) is an $s-t$ cut (because it partitions V, $s \in A$, and $t \in B$)

Let $e=(u,v) \in E$ (directed edge) such that $u \in A$ and $v \in B$.

$f(e)=c_e$ (since $(u,v) \notin G_f$, otherwise v would be in A)



Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

Proof: Let $A=\{v \in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V \backslash A$.

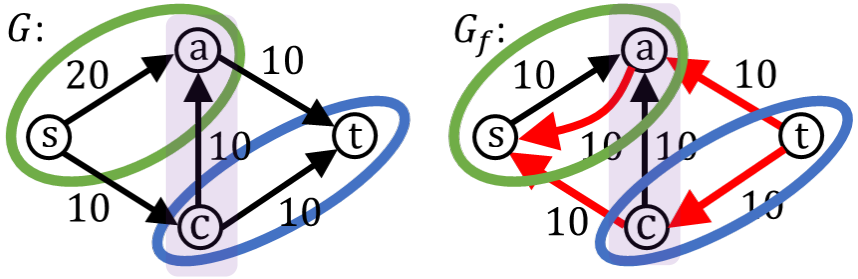
(A,B) is an $s-t$ cut (because it partitions V, $s \in A$, and $t \in B$)

Let $e=(u,v) \in E$ (directed edge) such that $u \in A$ and $v \in B$.

$f(e)=c_e$ (since $(u,v) \notin G_f$, otherwise v would be in A)

Let $e'=(u',v') \in E$ (directed edge) such that $u' \in B$ and $v' \in A$.

What can we say about $f(e')$?



Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f|=c(A,B)$.

Proof: Let $A=\{v \in V: \exists\ s-v \text{ path in } G_f\}$ and $B=V \backslash A$.

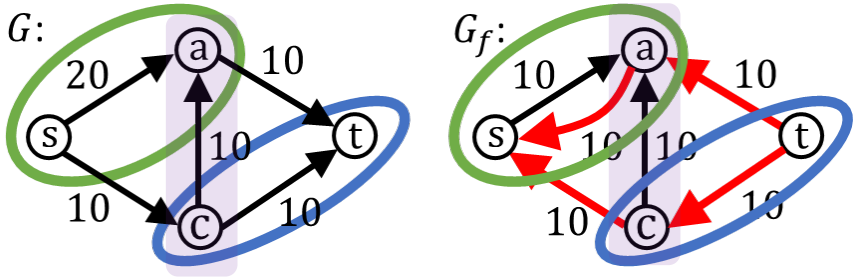
(A,B) is an $s-t$ cut (because it partitions V, $s \in A$, and $t \in B$)

Let $e=(u,v) \in E$ (directed edge) such that $u \in A$ and $v \in B$.

$f(e)=c_e$ (since $(u,v) \notin G_f$, otherwise v would be in A)

Let $e'=(u',v') \in E$ (directed edge) such that $u' \in B$ and $v' \in A$.

$f(e')=0$ (since $(v',u') \notin G_f$, otherwise u' would be in A)



Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A, B) in $G = (V, E)$ for which $|f| = c(A, B)$.

Proof: Let $A = \{v \in V : \exists\ s - v \text{ path in } G_f\}$ and $B = V \backslash A$.

(A, B) is an $s-t$ cut (because it partitions V, $s \in A$, and $t \in B$)

Let $e = (u, v) \in E$ (directed edge) such that $u \in A$ and $v \in B$.

$f(e) = c_e$ (since $(u, v) \notin G_f$, otherwise v would be in A)

Let $e' = (u', v') \in E$ (directed edge) such that $u' \in B$ and $v' \in A$.

$f(e') = 0$ (since $(v', u') \notin G_f$, otherwise u' would be in A)

Therefore, $|f| = \sum_{e \in \text{out}(A)} f(e) - \sum_{e \in \text{in}(A)} f(e)$ (Theorem 1)

Optimality

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A, B) in $G = (V, E)$ for which $|f| = c(A, B)$.

Proof: Let $A = \{v \in V : \exists\ s - v \text{ path in } G_f\}$ and $B = V \backslash A$.

(A, B) is an $s-t$ cut (because it partitions V, $s \in A$, and $t \in B$)

Let $e = (u, v) \in E$ (directed edge) such that $u \in A$ and $v \in B$.

$f(e) = c_e$ (since $(u, v) \notin G_f$, otherwise v would be in A)

Let $e' = (u', v') \in E$ (directed edge) such that $u' \in B$ and $v' \in A$.

$f(e') = 0$ (since $(v', u') \notin G_f$, otherwise u' would be in A)

Therefore, $|f| = \sum_{e \in \text{out}(A)} f(e) - \sum_{e \in \text{in}(A)} f(e)$ (Theorem 1)

$= \sum_{e \in \text{out}(A)} c_e - 0 = c(A, B)$

Optimality

Theorem: The flow returned by the Ford-Fulkerson algorithm is a maximum flow.

Proof:

??

Corollary: Suppose G is a flow network, f is an $s-t$ flow on G, and (A, B) is an $s-t$ cut. Then, $|f| \leq c(A, B)$. (i.e. *every* flow is bounded by *any* $s-t$ cut)

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A, B) in $G = (V, E)$ for which $|f| = c(A, B)$.

Optimality

Theorem: The flow returned by the Ford-Fulkerson algorithm is a maximum flow.

Proof: Since the Ford-Fulkerson algorithm finishes when no $s-t$ paths remain in G_f, Theorem 2 says there must be an $s-t$ cut such that the value of flow found equals the capacity of the cut.

Corollary: Suppose G is a flow network, f is an $s-t$ flow on G, and (A,B) is an $s-t$ cut. Then, $|f| \leq c(A,B)$. (i.e. *every* flow is bounded by *any* $s-t$ cut)

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f| = c(A,B)$.

Optimality

Theorem: The flow returned by the Ford-Fulkerson algorithm is a maximum flow.

Proof: Since the Ford-Fulkerson algorithm finishes when no $s-t$ paths remain in G_f, Theorem 2 says there must be an $s-t$ cut such that the value of flow found equals the capacity of the cut.

By the Corollary, there cannot be a flow with a larger value.

Corollary: Suppose G is a flow network, f is an $s-t$ flow on G, and (A,B) is an $s-t$ cut. Then, $|f| \leq c(A,B)$. (i.e. *every* flow is bounded by *any* $s-t$ cut)

Theorem 2: if f is an $s-t$ flow such that no $s-t$ path exists in residual graph G_f, then there is an $s-t$ cut (A,B) in $G=(V,E)$ for which $|f| = c(A,B)$.

Optimality

Theorem: The flow returned by the Ford-Fulkerson algorithm is a maximum flow.

Proof: Since the Ford-Fulkerson algorithm finishes when no $s - t$ paths remain in G_f, Theorem 2 says there must be an $s - t$ cut such that the value of flow found equals the capacity of the cut.

By the Corollary, there cannot be a flow with a larger value.

Therefore, the flow found by the Ford-Fulkerson algorithm is the maximum flow.

Corollary: Suppose G is a flow network, f is an $s - t$ flow on G, and (A, B) is an $s - t$ cut. Then, $|f| \leq c(A, B)$. (i.e. *every* flow is bounded by *any* $s - t$ cut)

Theorem 2: if f is an $s - t$ flow such that no $s - t$ path exists in residual graph G_f, then there is an $s - t$ cut (A, B) in $G = (V, E)$ for which $|f| = c(A, B)$.

Linear Programming
CSCI 532

Linear Programming

A very structured (details to follow) way to optimize a goal by turning knobs and following certain rules.

Linear Programming

A very structured (details to follow) way to optimize a goal by turning knobs and following certain rules.

Goal: Become as rich as possible

Knobs: Job Hunting, Education, Risk

Rules: Don’t rob a bank, get enough sleep to feel human, see your kids more than once a month

Linear Programming

A very structured (details to follow) way to optimize a goal by turning knobs and following certain rules.

Goal: Become as rich as possible

Knobs: Job Hunting, Education, Risk

Rules: Don’t rob a bank, get enough sleep to feel human, see your kids more than once a month

Linear Programming

A very structured (details to follow) way to optimize a goal by turning knobs and following certain rules.

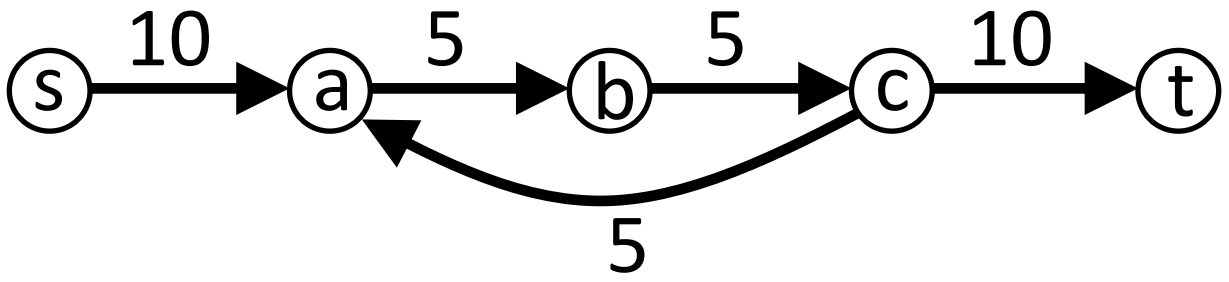
Vertex Cover:

Goal: Select the smallest subset of vertices

Knobs: Select vertex 1? Select vertex 2?...

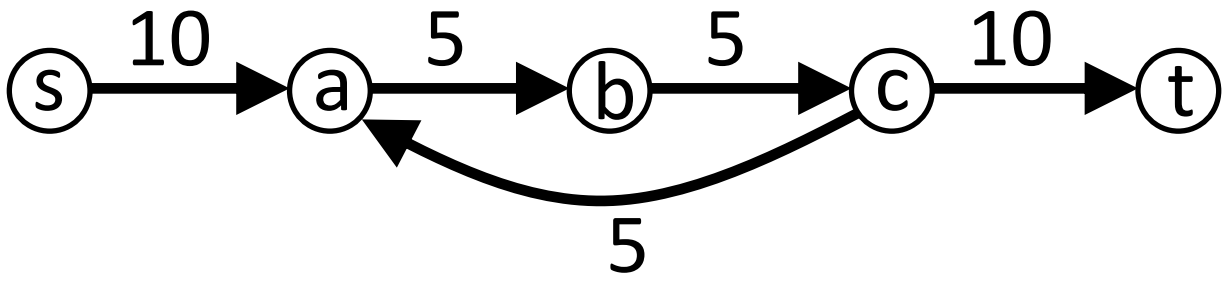
Rules: Every edge needs a selected vertex.

Max Flow



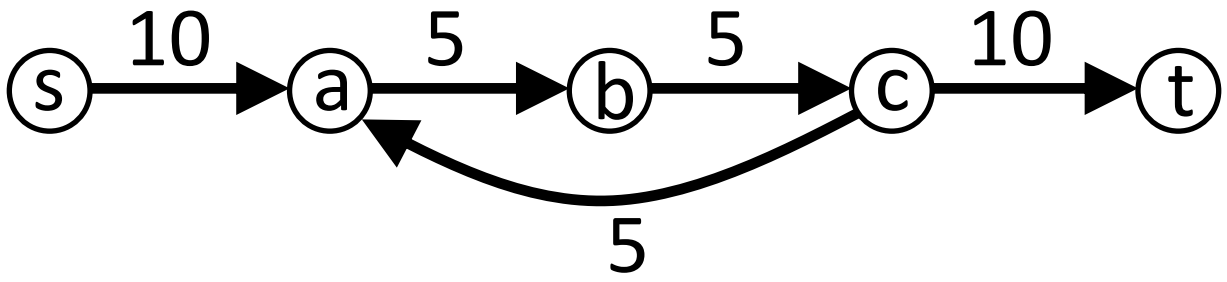
Max Flow = ?

Max Flow



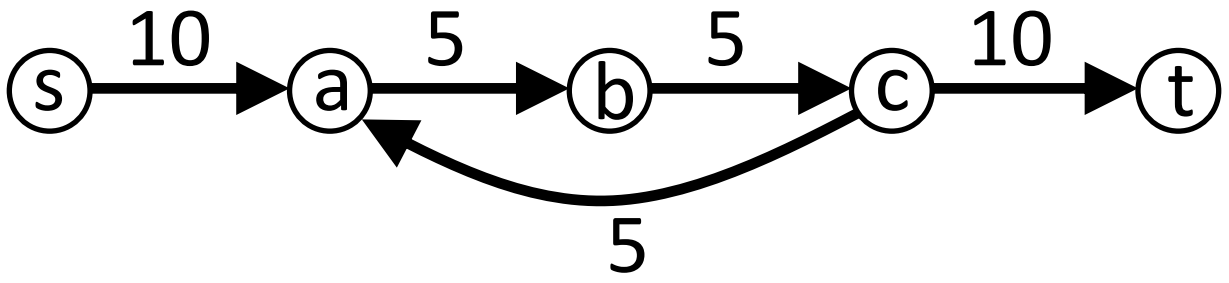
Max Flow = 5

Max Flow



Decision Variable (“knob”). The LP solver can change these values to better optimize the goal.

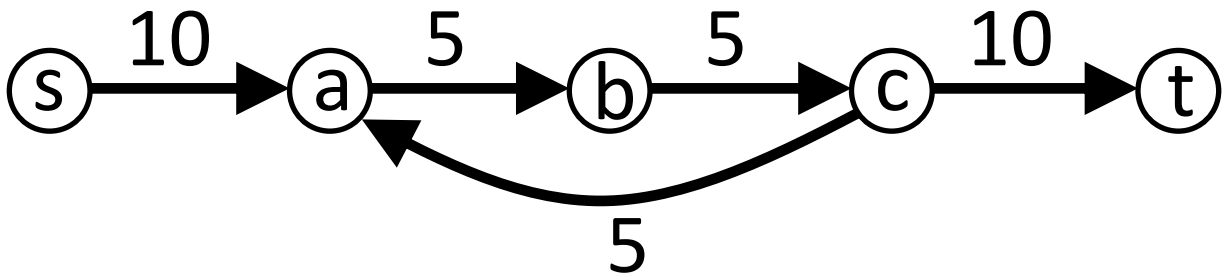
Max Flow



x_{sa} = Amount of flow on edge (s,a)

Decision Variable ("knob"). The LP solver can change these values to better optimize the goal.

Max Flow



x_{sa} = Amount of flow on edge (s,a)

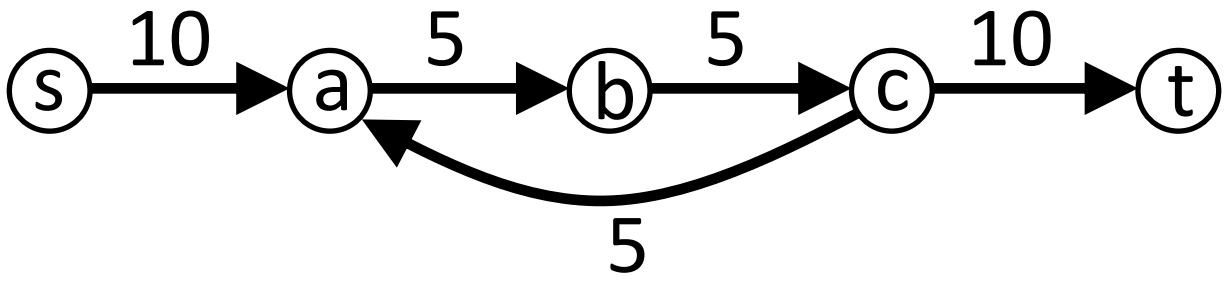
x_{ab} = Amount of flow on edge (a,b)

x_{bc} = Amount of flow on edge (b,c)

x_{ct} = Amount of flow on edge (c,t)

x_{ca} = Amount of flow on edge (c,a)

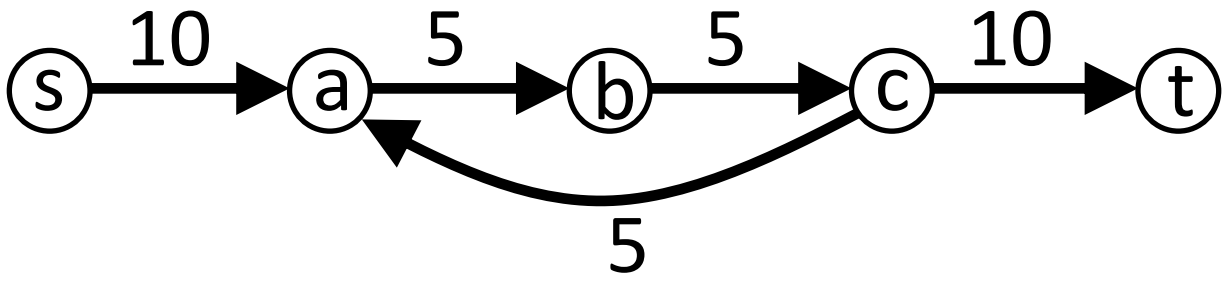
Max Flow



x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

c_{sa} = Capacity of edge (s,a)
c_{ab} = Capacity of edge (a,b)
c_{bc} = Capacity of edge (b,c)
c_{ct} = Capacity of edge (c,t)
c_{ca} = Capacity of edge (c,a)

Max Flow



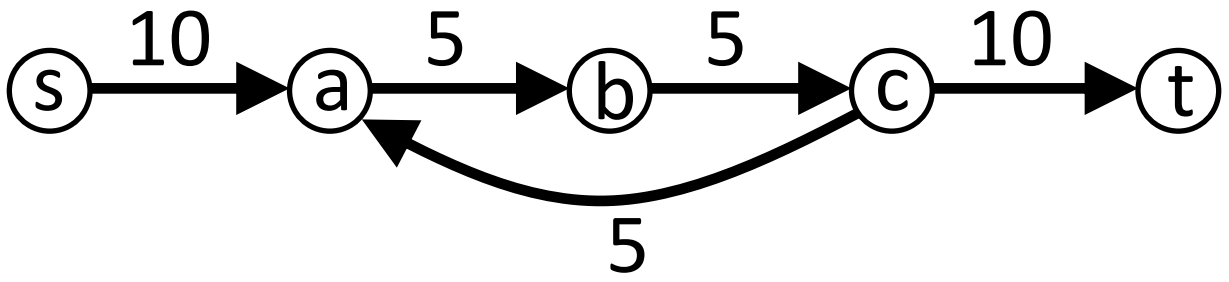
x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

~~c_{sa} = Capacity of edge (s,a)~~
~~c_{ab} = Capacity of edge (a,b)~~
~~c_{bc} = Capacity of edge (b,c)~~
~~c_{ct} = Capacity of edge (c,t)~~
~~c_{ca} = Capacity of edge (c,a)~~

Not decision variables!!

The solver is not allowed to modify capacities to influence the solution

Max Flow

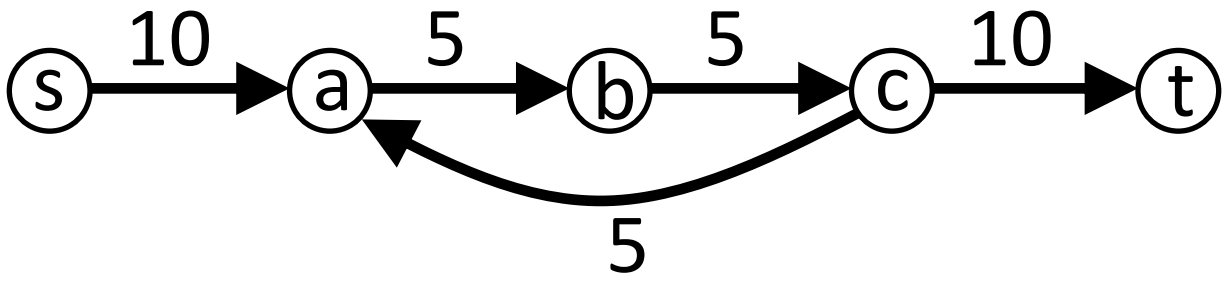


x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

Objective: ???

Objective ("goal"). The LP solver will do whatever it can to make this as good as possible.

Max Flow

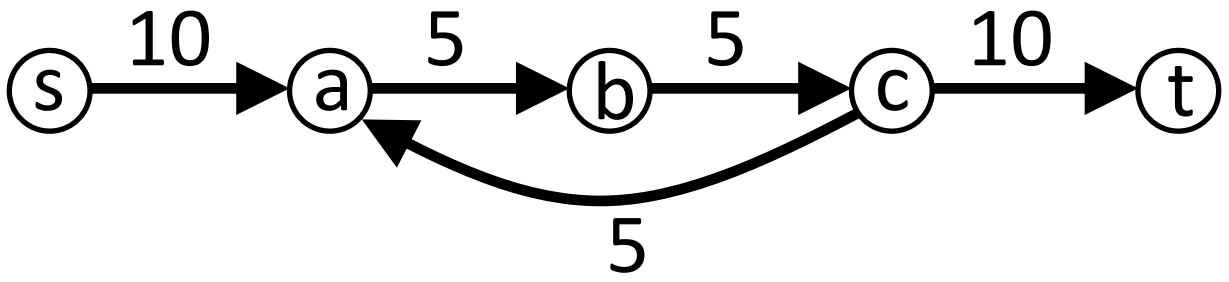


x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

Objective: $\max x_{ct}$

Objective ("goal"). The LP solver will do whatever it can to make this as good as possible.

Max Flow

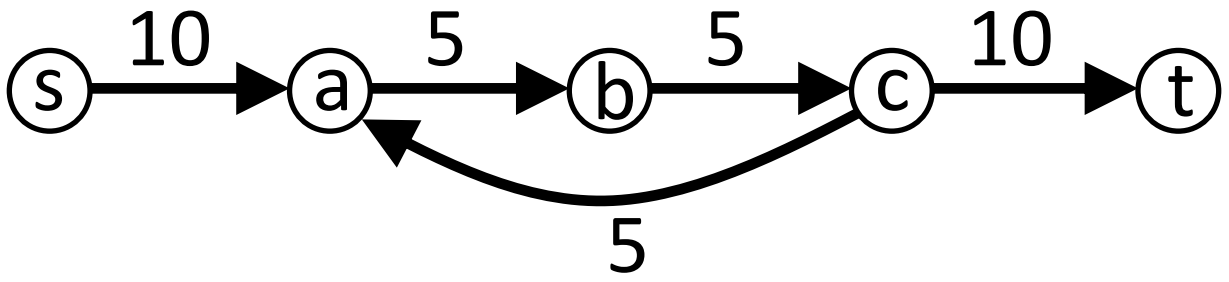


x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

Objective: $\max x_{ct}$

Good to go?

Max Flow

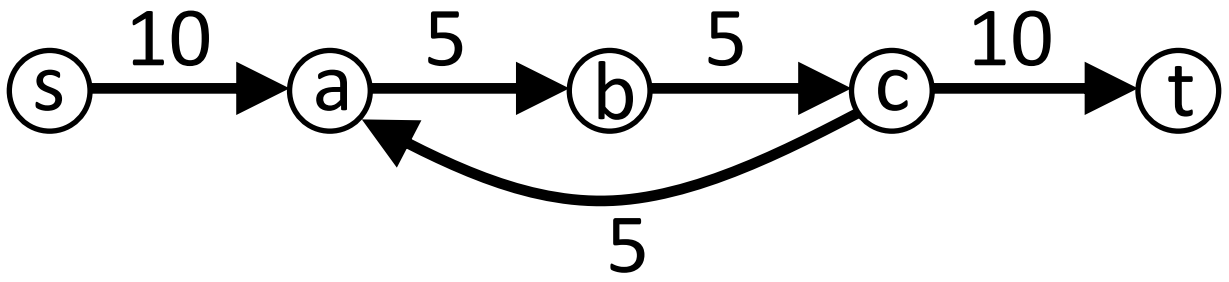


x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

Objective: $\max x_{ct}$

Good to go? No! We need rules or else the solver is going to become not helpful

Max Flow



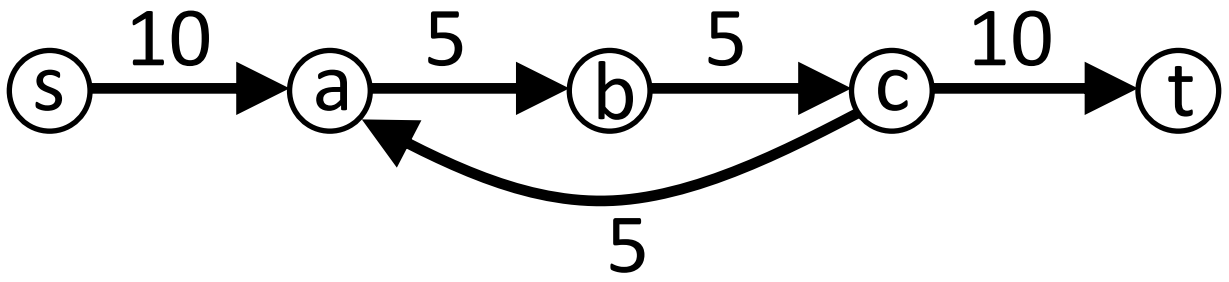
x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

Objective: $\max x_{ct}$

Subject to:

Constraints (“rules”). The solver is not allowed to break these rules.

Max Flow



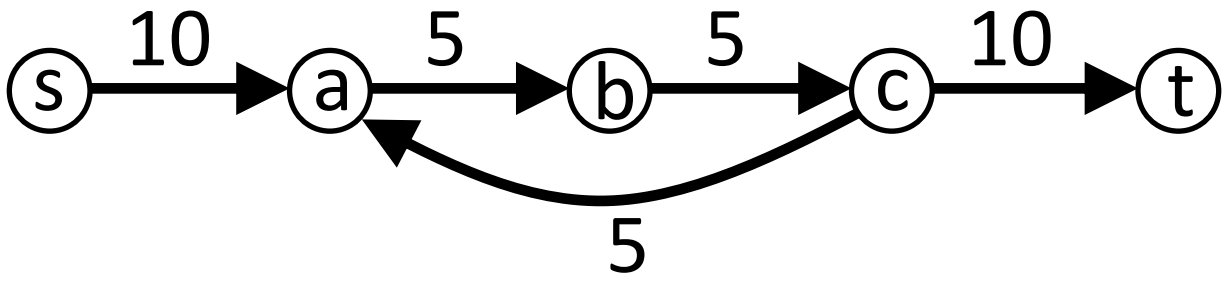
x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10$

Constraints ("rules"). The solver is not allowed to break these rules.

Max Flow



x_{sa} = Amount of flow on edge (s,a) x_{ct} = Amount of flow on edge (c,t)

x_{ab} = Amount of flow on edge (a,b) x_{ca} = Amount of flow on edge (c,a)

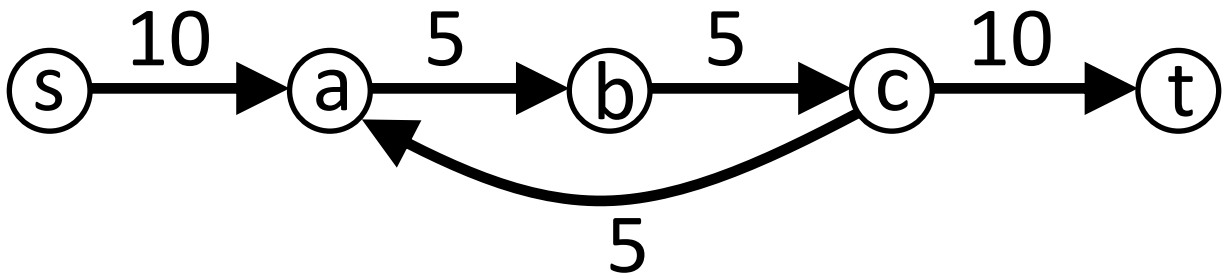
x_{bc} = Amount of flow on edge (b,c)

Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10, \quad x_{ab} \leq 5, \quad x_{bc} \leq 5, \quad x_{ct} \leq 10, \quad x_{ca} \leq 5$

Constraints ("rules"). The solver is not allowed to break these rules.

Max Flow



x_{sa} = Amount of flow on edge (s,a) x_{ct} = Amount of flow on edge (c,t)

x_{ab} = Amount of flow on edge (a,b) x_{ca} = Amount of flow on edge (c,a)

x_{bc} = Amount of flow on edge (b,c)

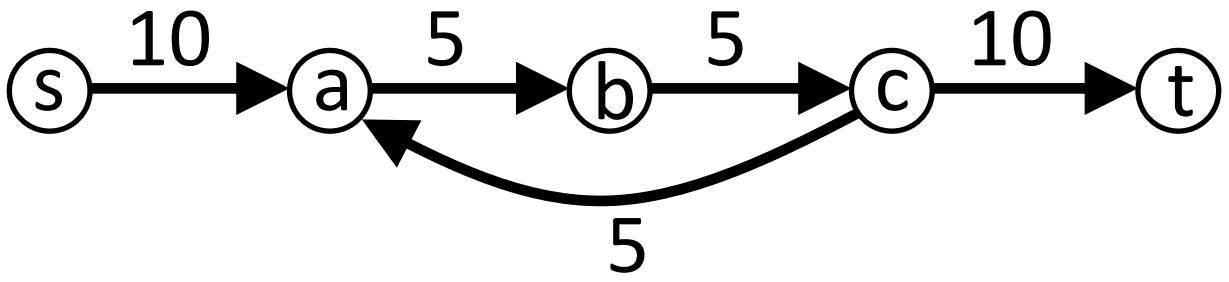
Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10, \quad x_{ab} \leq 5, \quad x_{bc} \leq 5, \quad x_{ct} \leq 10, \quad x_{ca} \leq 5$

$x_{sa} + x_{ca} - x_{ab} = 0$

Constraints ("rules"). The solver is not allowed to break these rules.

Max Flow



x_{sa} = Amount of flow on edge (s,a) x_{ct} = Amount of flow on edge (c,t)

x_{ab} = Amount of flow on edge (a,b) x_{ca} = Amount of flow on edge (c,a)

x_{bc} = Amount of flow on edge (b,c)

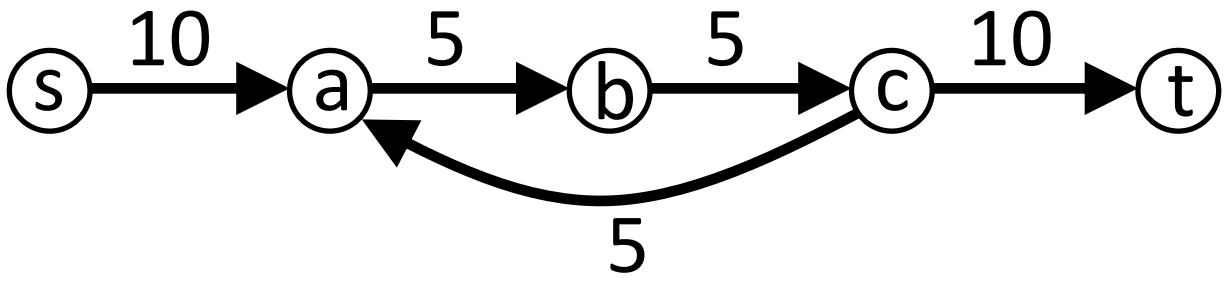
Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10, \quad x_{ab} \leq 5, \quad x_{bc} \leq 5, \quad x_{ct} \leq 10, \quad x_{ca} \leq 5$

$x_{sa} + x_{ca} - x_{ab} = 0, \quad x_{ab} - x_{bc} = 0, \quad x_{bc} - x_{ca} - x_{ct} = 0$

Constraints (“rules”). The solver is not allowed to break these rules.

Max Flow



Max Flow = 5

x_{sa} = Amount of flow on edge (s,a) x_{ct} = Amount of flow on edge (c,t)

x_{ab} = Amount of flow on edge (a,b) x_{ca} = Amount of flow on edge (c,a)

x_{bc} = Amount of flow on edge (b,c)

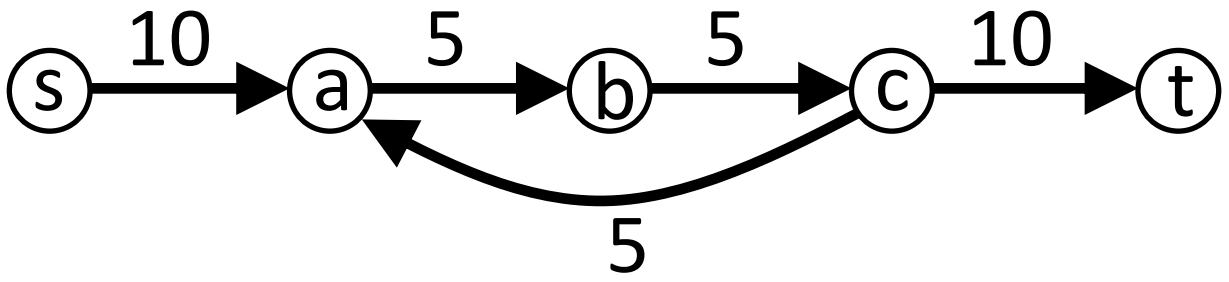
Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10, \quad x_{ab} \leq 5, \quad x_{bc} \leq 5, \quad x_{ct} \leq 10, \quad x_{ca} \leq 5$

$x_{sa} + x_{ca} - x_{ab} = 0, \quad x_{ab} - x_{bc} = 0, \quad x_{bc} - x_{ca} - x_{ct} = 0$

Any problems with this?

Max Flow



Max Flow = 5

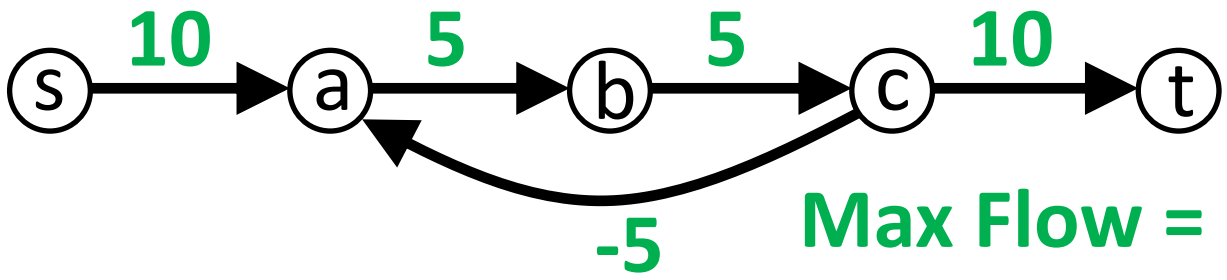
x_{sa} = Amount of flow on edge (s,a)
x_{ab} = Amount of flow on edge (a,b)
x_{bc} = Amount of flow on edge (b,c)
x_{ct} = Amount of flow on edge (c,t)
x_{ca} = Amount of flow on edge (c,a)

Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10, \quad x_{ab} \leq 5, \quad x_{bc} \leq 5, \quad x_{ct} \leq 10, \quad x_{ca} \leq 5$

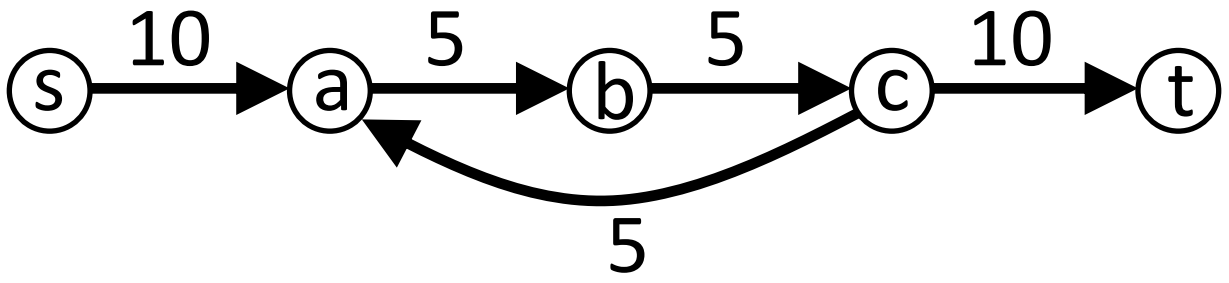
$x_{sa} + x_{ca} - x_{ab} = 0, \quad x_{ab} - x_{bc} = 0, \quad x_{bc} - x_{ca} - x_{ct} = 0$

Any problems with this? Yes! The solver will make some flow values negative to increase the max flow.



Max Flow = 10

Max Flow



Max Flow = 5

x_{sa} = Am… …dge (c,t)

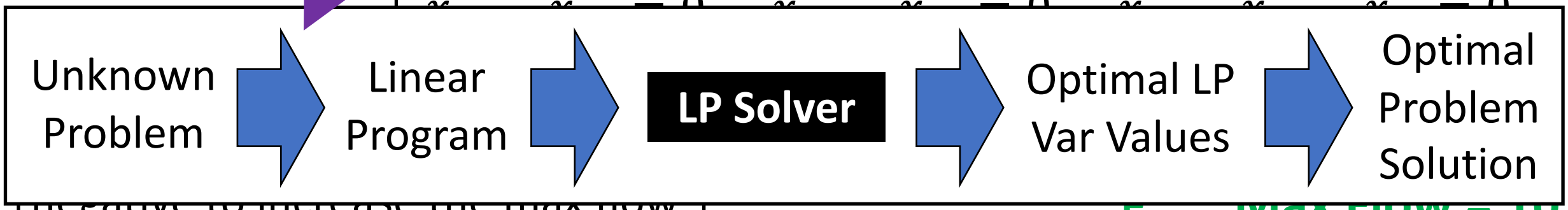
x_{ab} = Am… …dge (c,a)

x_{bc} = Am…

Our responsibility is to model the problem well enough that the solver has no choice but to give us the correct answer to the problem.

Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10, \quad x_{ab} \leq 5, \quad x_{bc} \leq 5, \quad x_{ct} \leq 10, \quad x_{ca} \leq 5$

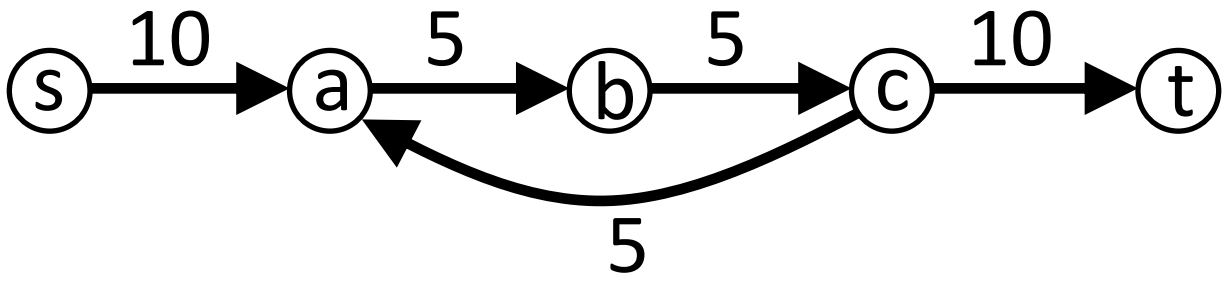


negative to increase the max flow.

-5

Max Flow = 10

Max Flow



x_{sa} = Amount of flow on edge (s,a) x_{ct} = Amount of flow on edge (c,t)
x_{ab} = Amount of flow on edge (a,b) x_{ca} = Amount of flow on edge (c,a)
x_{bc} = Amount of flow on edge (b,c)

Objective: $\max x_{ct}$

Subject to: $x_{sa} \leq 10, \quad x_{ab} \leq 5, \quad x_{bc} \leq 5, \quad x_{ct} \leq 10, \quad x_{ca} \leq 5$

$x_{sa} + x_{ca} - x_{ab} = 0, \quad x_{ab} - x_{bc} = 0, \quad x_{bc} - x_{ca} - x_{ct} = 0$

$x_{sa}, x_{ab}, x_{bc}, x_{ct}, x_{ca} \geq 0$

Max Flow

x_e = Amount of flow on edge e.

Objective: $\max \sum_{e \in \text{out}(s)} x_e$

Subject to: $x_e \leq \text{capacity}_e, \forall e \in E$

$\sum_{e \in \text{in}(v)} x_e - \sum_{e \in \text{out}(v)} x_e = 0, \forall v \in V \setminus \{s, t\}$

$x_e \geq 0, \forall e \in E$

Max Flow

Decision Variables:

- Real numbers = solvable in polynomial time (called LP).
- Integers = not (yet?) solvable in polynomial time (called integer linear program – ILP).

x_e = Amount of flow on edge e.

Objective: $\max \sum_{e \in \text{out}(s)} x_e$

Subject to: $x_e \leq \text{capacity}_e, \forall e \in E$

$\sum_{e \in \text{in}(v)} x_e - \sum_{e \in \text{out}(v)} x_e = 0, \forall v \in V \setminus \{s, t\}$

$x_e \geq 0, \forall e \in E$

Max Flow

Decision Variables:

- Real numbers = solvable in polynomial time (called LP).
- Integers = not (yet?) solvable in polynomial time (called integer linear program – ILP).

x_e = Amount of flow on edge e.

Objective:

- Can be minimization or maximization.
- Must be linear combinations of variables x_i (e.g. $a_1x_1 + \cdots + a_nx_n$ for constants a_i, not $a_ix_1x_2$).

Objective: $\max \sum_{e \in \text{out}(s)} x_e$

Subject to: $x_e \leq \text{capacity}_e, \forall e \in E$

$\sum_{e \in \text{in}(v)} x_e - \sum_{e \in \text{out}(v)} x_e = 0, \forall v \in V \setminus \{s, t\}$

$x_e \geq 0, \forall e \in E$

Max Flow

Decision Variables:

- Real numbers = solvable in polynomial time (called LP).
- Integers = not (yet?) solvable in polynomial time (called integer linear program – ILP).

x_e = Amount of flow on edge e.

Objective:

- Can be minimization or maximization.
- Must be linear combinations of variables x_i (e.g. $a_1x_1 + \cdots + a_nx_n$ for constants a_i, not $a_ix_1x_2$).

Objective: $\max \sum_{e \in \text{out}(s)} x_e$

Subject to:

$$x_e \leq \text{capacity}_e, \forall e \in E$$
$$\sum_{e \in \text{in}(v)} x_e - \sum_{e \in \text{out}(v)} x_e = 0, \forall v \in V \setminus \{s, t\}$$
$$x_e \geq 0, \forall e \in E$$

Constraints:

- Can be $\leq$, $\geq$, $=$.
- Must be linear combinations of variables.