More Network Flow Applications

Lecture
March 25, 2021

Most slides are courtesy Prof. Chekuri

Part I

Baseball Pennant Race

Pennant Race

TUESDAY, SEPTEMBER 10, 1996

San Francisco Chronicle

The Gate
Sports Online
▶ http://www.sfgate.com

SPORTING G

49ers, Young Get Big Brea

Quarterback m

By Gary Swan
Chronicle Staff Writer

The bye week has come at a perfect time for the 49ers and quarterback Steve Young. If they had a game next Sunday, there's a good chance Young would not play.

But the pulled groin muscle on his up-

Giants Officially Leave the NL West Race

By Nancy Gay
Chronicle Staff Writer

With the smack of another National League West bat 500 miles away, the Giants' run at the division title ended last night, just as they were handing the visiting St. Louis Cardinals an even bigger lead in the NL Central.

CARDINALS 6
GIANTS 2

In San Diego, Greg Vaughn's three-run homer in the eighth pushed the Padres over the Pirates and officially shoved the rest of the Giants' season into the background. On the heels of their tedious 6-2 loss before an announced crowd of 10,307 at Candlestick Park, the Giants fell 19½ games off the lead.

As it is, the worst the Padres (80-65) can finish is 80-82. The Giants have fallen to 59-83 with 20

Financing in Place For Giants' New Stadium
SEE PAGE B1, MAIN NEWS

games left; they cannot win 80 games. Coming off a miserable 2-8 mark on a three-city road trip that saw their road record drop to 27-47, the Giants were hoping to get off on the right foot in their longest homestand of the year (15 games, 14 days).

"Where we are, you're going to be eliminated sooner or later," Baker said quietly. "But it doesn't alter the fact that we've still got to play ball. You've still got to play hard, the fans come out to watch you play. You've got to play for the fact of loving to play, no matter where you are in the standings.

"You've got to play the role of spoiler, to not make it easier on

GIANTS: *Page D5 Col. 3*

Pennant Race: Example

Example

Team	Won	Remaining
New York	92	2
Baltimore	91	3
Toronto	91	3
Boston	89	2

Can Boston win the pennant?

Pennant Race: Example

Example

Team	Won	Remaining
New York	92	2
Baltimore	91	3
Toronto	91	3
Boston	89	2

Can Boston win the pennant?
No, because Boston can win at most 91 games.

Another Example

Example

Team	Won	Remaining
New York	92	2
Baltimore	91	3
Toronto	91	3
Boston	90	2

Can Boston win the pennant?

Another Example

Example

Team	Won	Remaining
New York	92	2
Baltimore	91	3
Toronto	91	3
Boston	90	2

Can Boston win the pennant?
Not clear unless we know what the remaining games are!

Refining the Example

Example

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	92	2	—	1	1	0
Baltimore	91	3	1	—	1	1
Toronto	91	3	1	1	—	1
Boston	90	2	0	1	1	—

Can Boston win the pennant?

Refining the Example

Example

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	92	2	—	1	1	0
Baltimore	91	3	1	—	1	1
Toronto	91	3	1	1	—	1
Boston	90	2	0	1	1	—

Can Boston win the pennant? Suppose Boston does

Refining the Example

Example

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	92	2	—	1	1	0
Baltimore	91	3	1	—	1	1
Toronto	91	3	1	1	—	1
Boston	90	2	0	1	1	—

Can Boston win the pennant? Suppose Boston does

1. Boston wins both its games to get 92 wins

Refining the Example

Example

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	92	2	—	1	1	0
Baltimore	91	3	1	—	1	1
Toronto	91	3	1	1	—	1
Boston	90	2	0	1	1	—

Can Boston win the pennant? Suppose Boston does

1. Boston wins both its games to get 92 wins
2. New York must lose both games

Refining the Example

Example

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	92	2	—	1	1	0
Baltimore	91	3	1	—	1	1
Toronto	91	3	1	1	—	1
Boston	90	2	0	1	1	—

Can Boston win the pennant? Suppose Boston does

1. Boston wins both its games to get 92 wins
2. New York must lose both games; now both Baltimore and Toronto have at least 92

Refining the Example

Example

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	92	2	—	1	1	0
Baltimore	91	3	1	—	1	1
Toronto	91	3	1	1	—	1
Boston	90	2	0	1	1	—

Can Boston win the pennant? Suppose Boston does

1. Boston wins both its games to get 92 wins
2. New York must lose both games; now both Baltimore and Toronto have at least 92
3. Winner of Baltimore-Toronto game has 93 wins!

Can Boston win the penant?

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	3	6	—	2	3	1
Baltimore	5	4	2	—	1	1
Toronto	4	6	3	1	—	2
Boston	2	4	1	1	2	—

(A) Yes.

(B) No.

Abstracting the Problem

Given

1. A set of teams S
2. For each $x \in S$, the current number of wins w_x
3. For any $x, y \in S$, the number of remaining games g_{xy} between x and y
4. A team z

Can z win the pennant?

Towards a Reduction

$\overline{z}$ can win the pennant if

1. $\overline{z}$ wins at least m games
2. no other team wins more than m games

Towards a Reduction

$\overline{z}$ can win the pennant if

1. $\overline{z}$ wins at least m games
   1. to maximize $\overline{z}$'s chances we make $\overline{z}$ win all its remaining games and hence $m = w_{\overline{z}} + \sum_{x \in S} g_{x\overline{z}}$
2. no other team wins more than m games

Towards a Reduction

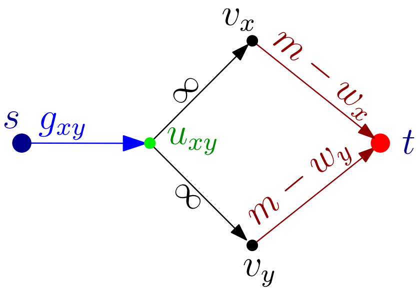
$\overline{z}$ can win the pennant if

1. $\overline{z}$ wins at least m games
   1. to maximize $\overline{z}$'s chances we make $\overline{z}$ win all its remaining games and hence $m = w_{\overline{z}} + \sum_{x \in S} g_{x\overline{z}}$
2. no other team wins more than m games
   1. for each $x, y \in S$ the g_{xy} games between them have to be *assigned* to either x or y.

Is there an assignment of remaining games to teams such that no team $x \neq \overline{z}$ wins more than $m - w_x$ games?

Flow Network: The basic gadget

1. s: source
2. t: sink
3. x, y: two teams
4. g_{xy}: number of games remaining between x and y.
5. w_x: number of points x has.
6. m: maximum number of points x can win before $\overline{z}$ starts loosing.

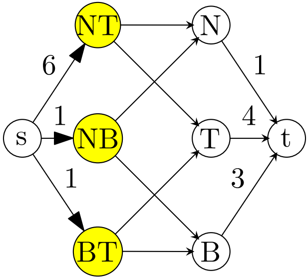


Flow Network: An Example

Can Boston win?

Team	Won	Remaining	NY	Bal	Tor	Bos
New York	90	11	—	1	6	4
Baltimore	88	6	1	—	1	4
Toronto	87	11	6	1	—	4
Boston	79	12	4	4	4	—

1. $m = 79 + 12 = 91$: Boston can get at most **91** points.



Constructing Flow Network

Notations

1. S: set of teams,
2. w_x wins for each team, and
3. g_{xy} games remaining between x and y.
4. m be the maximum number of wins for $\overline{z}$,
5. and $S' = S \setminus \{\overline{z}\}$.

Reduction

Construct the flow network G as follows

1. One vertex v_x for each team $x \in S'$, one vertex u_{xy} for each pair of teams x and y in S'
2. A new source vertex s and sink t
3. Edges (s, u_{xy}) of capacity g_{xy}
4. Edges (v_x, t) of capacity equal $m - w_x$
5. Edges (u_{xy}, v_x) and (u_{xy}, v_y) of capacity ∞

Correctness of reduction

Theorem

G' has a maximum flow of value $g^* = \sum_{x,y \in S'} g_{xy}$ if and only if $\overline{z}$ can win the most number of games (including possibly tie with other teams).

Proof of Correctness

Proof.

Existence of g^* flow $\Rightarrow$ $\bar{z}$ wins pennant

1. An integral flow saturating edges out of s ensures that each remaining game between x and y is played.
2. Capacity on (v_x, t) edges ensures that no team wins more than m games

Proof of Correctness

Proof.

Existence of g^* flow $\Rightarrow \bar{z}$ wins pennant

1. An integral flow saturating edges out of s ensures that each remaining game between x and y is played.
2. Capacity on (v_x, t) edges ensures that no team wins more than m games

Conversely, $\bar{z}$ wins pennant $\Rightarrow$ flow of value g^*

1. The game outcomes determines flow on edges; if x wins k of the games against y, then flow on (u_{xy}, v_x) edge is k and on (u_{xy}, v_y) edge is $g_{xy} - k$ □

Proof that $\overline{z}$ cannot win the pennant

1. Suppose $\overline{z}$ cannot win the pennant since $g^* < g$. How do we *prove* to some one *compactly* that $\overline{z}$ cannot win the pennant?

Proof that $\overline{z}$ cannot win the pennant

1. Suppose $\overline{z}$ cannot win the pennant since $g^* < g$. How do we *prove* to some one *compactly* that $\overline{z}$ cannot win the pennant?
2. Show them the min-cut in the reduction flow network!

Proof that $\overline{z}$ cannot win the pennant

1. Suppose $\overline{z}$ cannot win the pennant since $g^* < g$. How do we *prove* to some one *compactly* that $\overline{z}$ cannot win the pennant?
2. Show them the min-cut in the reduction flow network!
3. See Kleinberg-Tardos book for a natural interpretation of the min-cut as a certificate.

The biggest loser?

Given an input as above for the pennant competition, deciding if a team can come in the last place

- **(A)** Can be done using the same reduction as just seen.
- **(B)** Can not be done using the same reduction as just seen.
- **(C)** Can be done using flows but we need lower bounds on the flow, instead of upper bounds.
- **(D)** The problem is **NP-Hard** and requires exponential time.
- **(E)** Can be solved by negating all the numbers, and using the above reduction.
- **(F)** Can be solved efficiently only by running a reality show on the problem.

Part II

An Application of Min-Cut to Project Scheduling

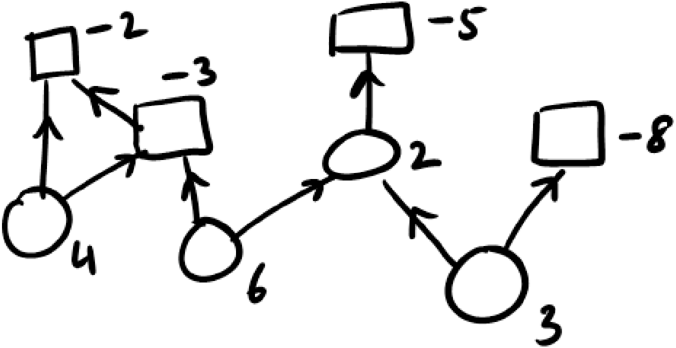
Project Scheduling

Problem:

1. n projects/tasks $1, 2, \ldots, n$
2. *dependencies* between projects: i depends on j implies i cannot be done unless j is done. dependency graph is *acyclic*
3. each project i has a cost/profit p_i
   1. $p_i < 0$ implies i requires a cost of $-p_i$ units
   2. $p_i > 0$ implies that i generates p_i profit

Goal: Find projects to do so as to *maximize* profit.

Example



Notation

For a set A of projects:

1. A is a *valid* solution if A is *dependency closed*, that is for every $i \in A$, all projects that i depends on are also in A.

Notation

For a set A of projects:

1. A is a *valid* solution if A is *dependency closed*, that is for every $i \in A$, all projects that i depends on are also in A.
2. $profit(A) = \sum_{i \in A} p_i$. Can be negative or positive.

Notation

For a set A of projects:

1. A is a *valid* solution if A is *dependency closed*, that is for every $i \in A$, all projects that i depends on are also in A.
2. $\textit{profit}(A) = \sum_{i \in A} p_i$. Can be negative or positive.

Goal: find valid A to maximize $\textit{profit}(A)$.

Idea: Reduction to Minimum-Cut

Finding a set of projects is partitioning the projects into two sets: those that are done and those that are not done.

Can we express this is a minimum cut problem?

Idea: Reduction to Minimum-Cut

Finding a set of projects is partitioning the projects into two sets: those that are done and those that are not done.

Can we express this is a minimum cut problem?

Several issues:

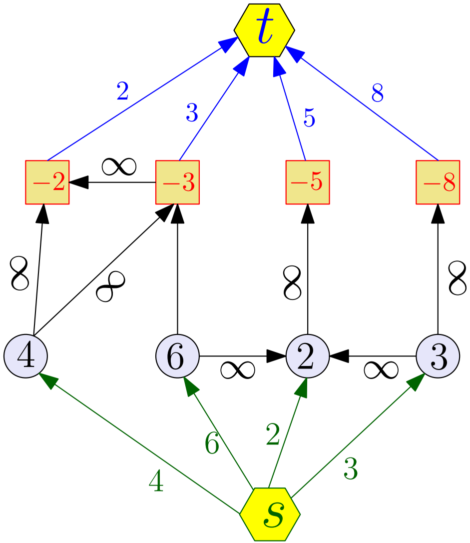
1. We are interested in maximizing profit but we can solve minimum cuts.
2. We need to convert negative profits into positive capacities.
3. Need to ensure that chosen projects is a valid set.
4. The cut value captures the profit of the chosen set of projects.

Reduction to Minimum-Cut

Note: We are reducing a *maximization* problem to a *minimization* problem.

1. projects represented as nodes in a graph
2. if i depends on j then (i, j) is an edge
3. add source s and sink t
4. for each i with $p_i > 0$ add edge (s, i) with capacity p_i
5. for each i with $p_i < 0$ add edge (i, t) with capacity $-p_i$
6. for each dependency edge (i, j) put capacity ∞ (more on this later)

Reduction: Flow Network Example



Reduction contd

Algorithm:

1. form graph as in previous slide
2. compute s-t minimum cut (A, B)
3. output the projects in $A - \{s\}$

Understanding the Reduction

Let $C = \sum_{i:p_i>0} p_i$: maximum possible profit.

Understanding the Reduction

Let $C = \sum_{i:p_i>0} p_i$: maximum possible profit.

Observation: The minimum s-t cut value is $\leq C$. Why?

Understanding the Reduction

Let $C = \sum_{i:p_i>0} p_i$: maximum possible profit.

Observation: The minimum s-t cut value is $\leq C$. Why?

Lemma

Suppose (A, B) is an s-t cut of finite capacity (no ∞) edges. Then projects in $A - \{s\}$ are a valid *solution.*

Understanding the Reduction

Let $C = \sum_{i:p_i>0} p_i$: maximum possible profit.

Observation: The minimum s-t cut value is $\leq C$. Why?

Lemma

Suppose (A, B) is an s-t cut of finite capacity (no ∞) edges. Then projects in $A - \{s\}$ are a valid *solution.*

Proof.

If $A - \{s\}$ is not a valid solution then there is a project $i \in A$ and a project $j \notin A$ such that i depends on j

Understanding the Reduction

Let $C = \sum_{i:p_i>0} p_i$: maximum possible profit.

Observation: The minimum s-t cut value is $\leq C$. Why?

Lemma

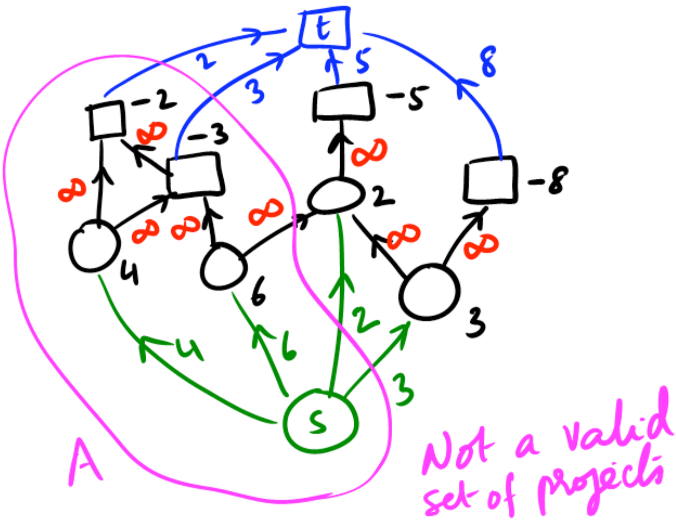
Suppose (A, B) is an s-t cut of finite capacity (no ∞) edges. Then projects in $A - \{s\}$ are a valid *solution.*

Proof.

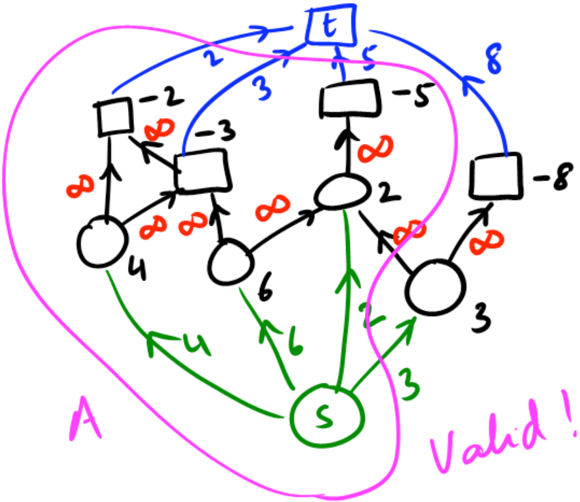
If $A - \{s\}$ is not a valid solution then there is a project $i \in A$ and a project $j \notin A$ such that i depends on j

Since (i, j) capacity is ∞, implies (A, B) capacity is ∞, contradicting assumption. □

Example



Example



Correctness of Reduction

Recall that for a set of projects X, $profit(X) = \sum_{i \in X} p_i$.

Correctness of Reduction

Recall that for a set of projects X, $profit(X) = \sum_{i \in X} p_i$.

Lemma

Suppose (A, B) *is an* s-t *cut of finite capacity (no* ∞*) edges. Then* $c(A, B) = C - profit(A - \{s\})$.

Correctness of Reduction

Recall that for a set of projects X, $profit(X) = \sum_{i \in X} p_i$.

Lemma

Suppose (A, B) is an s-t cut of finite capacity (no ∞) edges. Then $c(A, B) = C - profit(A - \{s\})$.

Proof.

Edges in (A, B):

1. (s, i) for $i \in B$ and $p_i > 0$: capacity is p_i
2. (i, t) for $i \in A$ and $p_i < 0$: capacity is $-p_i$
3. cannot have ∞ edges

□

Proof contd

For project set A let

1. $cost(A) = \sum_{i \in A: p_i < 0} -p_i$
2. $benefit(A) = \sum_{i \in A: p_i > 0} p_i$
3. $profit(A) = benefit(A) - cost(A)$.

Proof.

Let $A' = A \cup \{s\}$.

$$\begin{aligned} c(A', B) &= cost(A) + benefit(B) \\ &= cost(A) - benefit(A) + benefit(A) + benefit(B) \\ &= -profit(A) + C \\ &= C - profit(A) \end{aligned}$$

$\square$

Correctness of Reduction contd

We have shown that if (A, B) is an s-t cut in G with finite capacity then

1. $A - \{s\}$ is a valid set of projects
2. $c(A, B) = C - profit(A - \{s\})$

Correctness of Reduction contd

We have shown that if (A, B) is an s-t cut in G with finite capacity then

1. $A - \{s\}$ is a valid set of projects
2. $c(A, B) = C - profit(A - \{s\})$

Therefore a *minimum* s-t cut (A^*, B^*) gives a *maximum* profit set of projects $A^* - \{s\}$ since C is fixed.

Correctness of Reduction contd

We have shown that if (A, B) is an s-t cut in G with finite capacity then

1. $A - \{s\}$ is a valid set of projects
2. $c(A, B) = C - profit(A - \{s\})$

Therefore a *minimum* s-t cut (A^*, B^*) gives a *maximum* profit set of projects $A^* - \{s\}$ since C is fixed.

Question: How can we use ∞ in a real algorithm?

Correctness of Reduction contd

We have shown that if (A, B) is an s-t cut in G with finite capacity then

1. $A - \{s\}$ is a valid set of projects
2. $c(A, B) = C - profit(A - \{s\})$

Therefore a *minimum* s-t cut (A^*, B^*) gives a *maximum* profit set of projects $A^* - \{s\}$ since C is fixed.

Question: How can we use ∞ in a real algorithm?

Set capacity of ∞ arcs to $C + 1$ instead. Why does this work?