CS 580: Algorithm Design and Analysis

Jeremiah Blocki
Purdue University
Spring 2019

Recap

- Network Flow Problems
- Max-Flow Min Cut Theorem
- Ford Fulkerson
  - Augmenting Paths
  - Residual Flow Graph
  - Integral Solutions (given integral capacities)
- Capacity Scaling Algorithm
- Dinic's Algorithm
- Applications of Maximum Flow
- Maximum Bipartite Matching
- Marriage Theorem (Hall/Frobenius)
- Disjoint Paths [Menger's Theorem]
- Baseball Elimination
- Circulation with Demands
- Many Others…

2

7.12 Baseball Elimination

"See that thing in the paper last week about Einstein? . . . Some reporter asked him to figure out the mathematics of the pennant race. You know, one team wins so many of their remaining games, the other teams win this number or that number. What are the myriad possibilities? Who's got the edge?"
"The hell does he know?"
"Apparently not much. He picked the Dodgers to eliminate the Giants last Friday."

- Don DeLillo, Underworld

Baseball Elimination

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij} Atl	Phi	NY	Mon
Atlanta	83	71	8	-	1	6	1
Philly	80	79	3	1	-	0	2
New York	78	78	6	6	0	-	0
Montreal	77	82	3	1	2	0	-

Which teams have a chance of finishing the season with most wins?

- Montreal eliminated since it can finish with at most 80 wins, but Atlanta already has 83.
- $w_i + r_i < w_j \Rightarrow$ team i eliminated.
- Only reason sports writers appear to be aware of.
- Sufficient, but not necessary!

4

Baseball Elimination

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij} Atl	Phi	NY	Mon
Atlanta	83	71	8	-	1	6	1
Philly	80	79	3	1	-	0	2
New York	78	78	6	6	0	-	0
Montreal	77	82	3	1	2	0	-

Which teams have a chance of finishing the season with most wins?

- Philly can win 83, but still eliminated . . .
- If Atlanta loses a game, then some other team wins one.

Remark. Answer depends not just on how many games already won and left to play, but also on whom they're against.

5

Baseball Elimination

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij} Atl	Phi	NY	Mon
Atlanta	**83**	71	8	-	1	**6**	1
Philly	80	79	3	1	-	0	2
New York	**78**	78	6	**6**	0	-	0
Montreal	77	82	3	1	2	0	-

Which teams have a chance of finishing the season with most wins?

- Philly can win 83, but still eliminated . . .
- If Atlanta loses a game, then some other team wins one.

Remark. Answer depends not just on how many games already won and left to play, but also on whom they're against.

6

Baseball Elimination


TUESDAY, SEPTEMBER 10, 1996

San Francisco Chronicle

SPORTING G

The Gate

Sports Online

http://www.sfgate.com

49ers, Young Get Big Brea

Quarterback m

Giants Officially Leave the NL West Race

CARDINALS 6

GIANTS 2

Financing in Place For Giants' New Stadium

Baseball Elimination

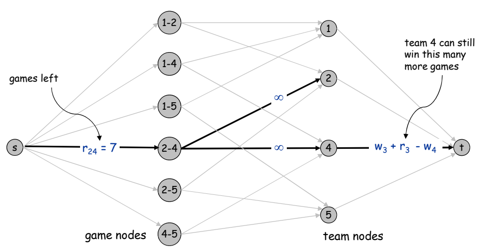
Baseball elimination problem.

- Set of teams S.
- Distinguished team $s \in S$.
- Team x has won w_x games already.
- Teams x and y play each other r_{xy} additional times.
- Is there any outcome of the remaining games in which team s finishes with the most (or tied for the most) wins?

Baseball Elimination: Max Flow Formulation

Can team 3 finish with most wins?

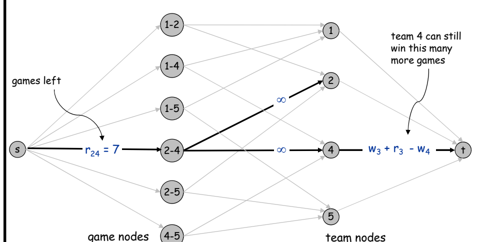
- Assume team 3 wins all remaining games $\Rightarrow$ $w_3 + r_3$ wins.
- Divvy remaining games so that all teams have $\leq$ $w_3 + r_3$ wins.



Baseball Elimination: Max Flow Formulation

Theorem. Team 3 is not eliminated iff max flow saturates all edges leaving source.

- Integrality theorem $\Rightarrow$ each remaining game between x and y added to number of wins for team x or team y.
- Capacity on (x, t) edges ensure no team wins too many games.



Baseball Elimination: Explanation for Sports Writers

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}				
				NY	Bal	Bos	Tor	Det
NY	75	59	28	-	3	8	7	3
Baltimore	71	63	28	3	-	2	7	4
Boston	69	66	27	8	2	-	0	0
Toronto	63	72	27	7	7	0	-	-
Detroit	49	86	27	3	4	0	0	-

AL East: August 30, 1996

Which teams have a chance of finishing the season with most wins?

- Detroit could finish season with 49 + 27 = 76 wins.

Baseball Elimination: Explanation for Sports Writers

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}				
				NY	Bal	Bos	Tor	Det
NY	75	59	28	-	3	8	7	3
Baltimore	71	63	28	3	-	2	7	4
Boston	69	66	27	8	2	-	0	0
Toronto	63	72	27	7	7	0	-	-
Detroit	49	86	27	3	4	0	0	-

AL East: August 30, 1996

Which teams have a chance of finishing the season with most wins?

- Detroit could finish season with 49 + 27 = 76 wins.

Certificate of elimination. R = {NY, Bal, Bos, Tor}

- Have already won w(R) = 278 games.
- Must win at least r(R) = 27 more.
- Average team in R wins at least 305/4 > 76 games.

Baseball Elimination: Explanation for Sports Writers

Certificate of elimination.

If $\overbrace{\frac{w(T)+g(T)}{|T|}}^{\text{LB on avg \# games won}} > w_z + g_z$ then z is eliminated (by subset T).

$$T \subseteq S,\ \ w(T) := \overbrace{\sum_{i \in T} w_i}^{\text{\# wins}},\ \ g(T) := \overbrace{\sum_{\{x,y\} \subseteq T} g_{xy}}^{\text{\# remaining games}},$$

Theorem. [Hoffman-Rivlin 1967] Team z is eliminated iff there exists a subset T* that eliminates z.

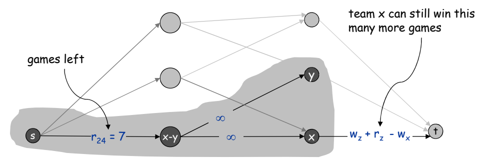
Proof idea. Let T* = team nodes on source side of min cut.

13

Baseball Elimination: Explanation for Sports Writers

Theorem. [Hoffman-Rivlin 1967] Team z is eliminated iff there exists a subset T* that eliminates z.

Proof idea. Let T* = team nodes on source side of min cut.

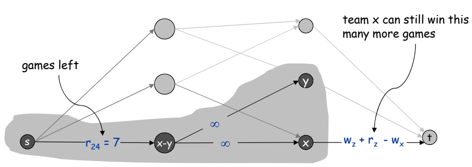


14

Baseball Elimination: Explanation for Sports Writers

Pf of theorem.

- Use max flow formulation, and consider min cut (A, B).
- Define T* = team nodes on source side of min cut.
- Observe x-y $\in$ A iff both x $\in$ T* and y $\in$ T*.
  - infinite capacity edges ensure if x-y $\in$ A then x $\in$ A and y $\in$ A
  - if x $\in$ A and y $\in$ A but x-y $\in$ T, then adding x-y to A decreases capacity of cut



15

Baseball Elimination: Explanation for Sports Writers

Pf of theorem.

- Use max flow formulation, and consider min cut (A, B).
- Define T* = team nodes on source side of min cut.
- Observe x-y $\in$ A iff both x $\in$ T* and y $\in$ T*.
- $g(S-\{z\}) \ > \ cap(A, B)$

$$= \ \underbrace{g(S-\{z\}) - g(T^*)}_{\text{capacity of game edges leaving s}} \ + \ \underbrace{\sum_{x \in T^*} (w_z + g_z - w_x)}_{\text{capacity of team edges leaving s}}$$

$$= \ g(S-\{z\}) - g(T^*) \ - \ w(T^*) \ + \ |T^*|(w_z + g_z)$$

- Rearranging terms: $w_z + g_z \ < \ \frac{w(T^*) + g(T^*)}{|T^*|}$ ▪

16

Linear Programming

- Even more general than Network Flow!
- Many Applications
  - Network Flow Variants
    - Taxation
    - Multi-Commodity Flow Problems
  - Supply-Chain Optimization
  - Operations Research
    - Entire Courses Devoted to Linear Programming!
- Our Focus
  - Using Linear Programming as a tool to solve algorithms problems
  - We won't cover algorithms to solve linear programs in any depth

Motivating Example: Time Allocation

168 Hours in Each Week to Allocate as Follows


Studying (S)


Partying (P)



Everything Else (E)

Constraints:

- [168 Hours] $S + P + E = 168$
- [Maintain Sanity] $P + E \geq 70$
- [Pass Courses 1] $S \geq 60$
- [Pass Courses 2] $2S + E - 3P \geq 150$ (too little sleep, and/or too much partying makes it more difficult to study)

18 Credit for Example: Avrim Blum

Motivating Example: Time Allocation

168 Hours in Each Week to Allocate as Follows




Studying (S) Partying (P) Everything Else (E)

Constraints:

- [168 Hours] $S + P + E = 168$
- [Maintain Sanity] $P + E \geq 70$
- [Survive] $E \geq 56$
- [Pass Courses 1] $S \geq 60$
- [Pass Courses 2] $2S + E - 3P \geq 150$ (too little sleep, and/or too much partying makes it more difficult to study)

Question 1: Can we satisfy all of the constraints?
(Maintain Sanity + Pass Courses)
Answer: Yes. One feasible solution is S=80, P=20, E=68

19 Credit for Example: Avrim Blum

Motivating Example: Time Allocation

168 Hours in Each Week to Allocate as Follows




Studying (S) Partying (P) Everything Else (E)

Constraints:

- [168 Hours] $S + P + E = 168$
- [Maintain Sanity] $P + E \geq 70$
- [Survive] $E \geq 56$
- [Pass Courses 1] $S \geq 60$
- [Pass Courses 2] $2S + E - 3P \geq 150$ (too little sleep, and/or too much partying makes it more difficult to study)

Objective Function: $2P + E$ [Maximize Happiness]

Question 2: Can we find a feasible solution which maximizes the objective function?

20

Linear Program Definition

- Variables: $x_1,\ldots,x_n$
- m linear inequalities in these variables (equalities are OK)
  - Examples
    - $0 \leq x_1 \leq 1$
    - $x_1 + x_4 + 3x_{10} - 7x_{11} \leq 4$
    - $2S + E - 3P \geq 150$
- [Optional] Linear Objective Function
  - Example:
    - maximize $4x_4 + 3x_{10}$
    - minimize $3x_1 + 3x_2$
    - maximize $2P + E$
- Goal
  - Find values for $x_1,\ldots,x_n$ satisfying all constraints, and
  - Maximize the objective
- Feasibility Problem
  - No objective function

21

Linear Program Definition

- **Variables:** $x_1,\ldots,x_n$
- **Constraints:** m linear inequalities in these variables (equalities are OK)
- [Optional] Linear Objective Function

Requirement:

- All the constrains are linear inequalities in variables (S,P,E)
- The objective function is also linear

Example Non-Linear Constraints:

$PE \geq 70$ $\quad E \in \{0,1\}$

$E(1-E) = 1$ $\quad \mathbf{Max}\{P,E\} \geq 20$

22

Linear Program Example

Goal: Maximize 2P+E
Subject to:

- [168 Hours] $S + P + E = 168$
- [Maintain Sanity] $P + E \geq 70$
- [Survive] $E \geq 56$
- [Pass Courses 1] $S \geq 60$
- [Pass Courses 2] $2S + E - 3P \geq 150$
- [Non-Negativity] $P \geq 0$

Requirement:

- All the constrains are linear inequalities in variables (S,P,E)
- The objective function is also linear

Example Non-Linear Constraints:

$PE \geq 70$ $\quad E \in \{0,1\}$

$E(1-E) = 1$ $\quad \mathbf{Max}\{P,E\} \geq 20$

23 Credit for Example: Avrim Blum

Network Flow as a Linear Program

Given a directed graph G with capacities c(e) on each edge e we can use linear programming to find a maximum flow from source **s** to sink **t**.

Variables: x_e for each directed edge e (represents flow on edge e)

Objective: Maximize $\sum_{e \text{ out of } s} x_e$

Constraints:

- (Capacity Constraints) For each edge e we have $0 \leq x_e \leq c(e)$
- (Flow Conservation) For each $v \notin \{s,t\}$ we have

$$\sum_{e \text{ out of } v} x_e = \sum_{e \text{ into } v} x_e$$

24

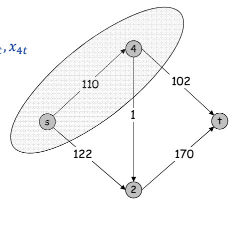
Network Flow as a Linear Program

Example:

Variables: $x_{s4}, x_{s2}, x_{42}, x_{2t}, x_{4t}$

Goal: maximize $x_{s4} + x_{s2}$
Subject to
- $0 \le x_{s4} \le 110$
- $0 \le x_{s2} \le 122$
- $0 \le x_{42} \le 1$
- $0 \le x_{2t} \le 170$
- $0 \le x_{4t} \le 102$

capacity



- $x_{s4} = x_{42} + x_{4t}$ [Flow Conservation at node 4]
- $x_{s4} + x_{42} = x_{2t}$ [Flow Conservation at node 2]

25

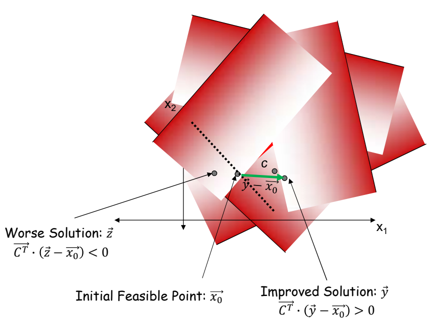
Solving a Linear Program

- Simplex Algorithm (1940s)
  - Not guaranteed to run in polynomial time
  - We can find bad examples, but…
  - The algorithm is efficient in practice!
- Ellipsoid Algorithm (1980)
  - Polynomial time (huge theoretical breakthrough), but ….
  - Slow in practice
- Newer Algorithms
  - Karmarkar's Algorithm
    - Competitive with Simplex
    - Polynomial Time

26

Algorithmic Idea: Direction of Goodness

Goal: Maximize $2x_1+3x_2$ c=(2,3)

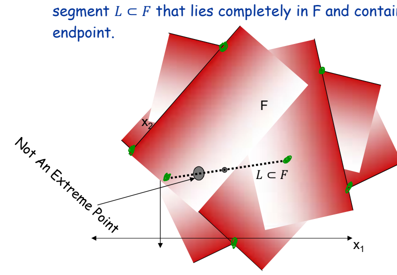


27

Linear Programming

Theorem: Maximum value achieved at vertex (extreme point)

Definition: Let F be the set of all feasible points in a linear program. We say that a point $p \in F$ is an extreme point (vertex) if *every* line segment $L \subset F$ that lies completely in F and contains p has p as an endpoint.

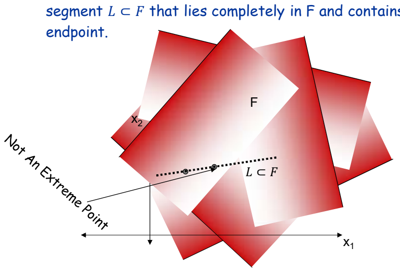


28

Linear Programming

Theorem: Maximum value achieved at vertex (extreme point)

Definition: Let F be the set of all feasible points in a linear program. We say that a point $p \in F$ is an extreme point (vertex) if *every* line segment $L \subset F$ that lies completely in F and contains p has p as an endpoint.

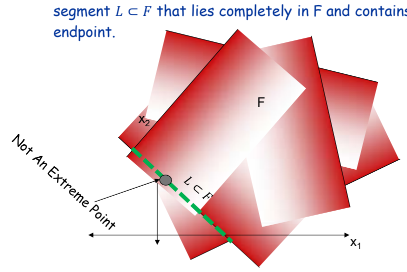


29

Linear Programming

Theorem: Maximum value achieved at vertex (extreme point)

Definition: Let F be the set of all feasible points in a linear program. We say that a point $p \in F$ is an extreme point (vertex) if *every* line segment $L \subset F$ that lies completely in F and contains p has p as an endpoint.

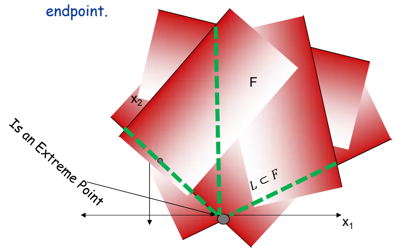


30

Linear Programming

Theorem: Maximum value achieved at vertex (extreme point)

Definition: Let F be the set of all feasible points in a linear program. We say that a point $p \in F$ is an extreme point (vertex) if *every* line segment $L \subset F$ that lies completely in F and contains p has p as an endpoint.



31

Linear Programming

Theorem: Maximum value achieved at vertex (extreme point)

Definition: Let F be the set of all feasible points in a linear program. We say that a point $p \in F$ is an extreme point (vertex) if *every* line segment $L \subset F$ that lies completely in F and contains p has p as an endpoint.

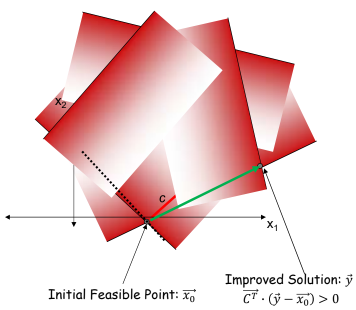
Observation: Each extreme point lies at the intersection of (at least) two constraints.

Theorem: a vertex is an optimal solution if there is no better *neighboring vertex.*

32

Algorithmic Idea: Vertex Walking

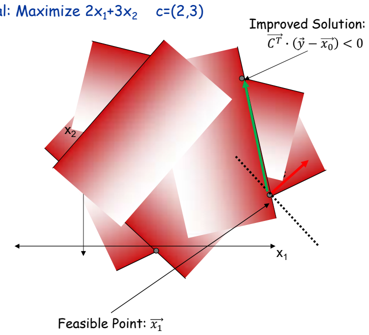
Goal: Maximize $2x_1+3x_2$ c=(2,3)



33

Algorithmic Idea: Vertex Walking

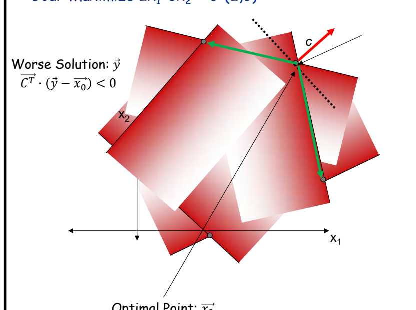
Goal: Maximize $2x_1+3x_2$ c=(2,3)



34

Algorithmic Idea: Vertex Walking

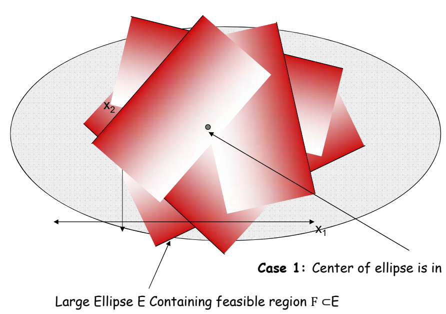
Goal: Maximize $2x_1+3x_2$ c=(2,3)



35

Ellipsoid Algorithm: Solves Feasibility Problem

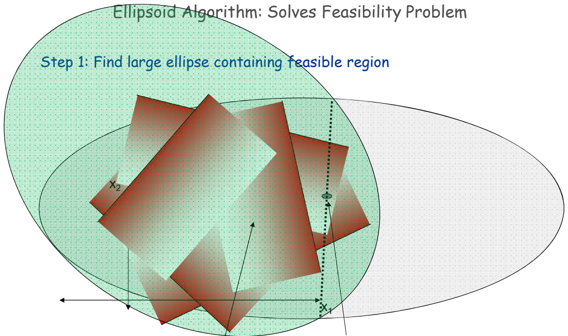
Step 1: Find large ellipse containing feasible region



36

Ellipsoid Algorithm: Solves Feasibility Problem

Step 1: Find large ellipse containing feasible region



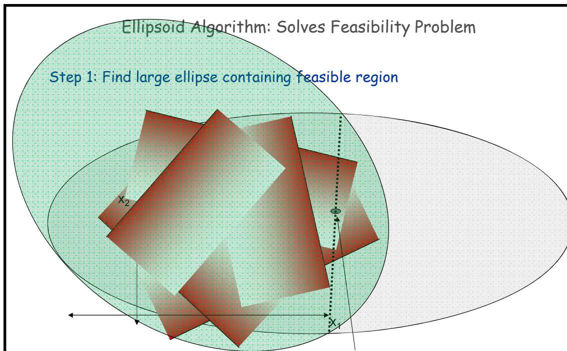
Case 2: Center of ellipse not in F

F contained in one half of the ellipsoid
➔ Can find smaller ellipsoid containing F
smaller by *at least* a $\left(1-\frac{1}{n}\right)$-factor.

37

Ellipsoid Algorithm: Solves Feasibility Problem

Step 1: Find large ellipse containing feasible region



Case 2: Center of ellipse not in F

smaller by *at least* a $\left(1-\frac{1}{n}\right)$-factor
➔ Every n steps volume drops by factor (1/e)
➔ poly(n) iterations to find feasible point (or reject)

38

Finding the Optimal Point with Ellipsoid Algorithm

Goal: maximize $\sum_i w_i x_i$ (where each w_i is a constant)

Key Idea: Binary Search for value of Optimal Solution!

- Add Constraint $\sum_i w_i x_i \geq B$
  - Infeasible?
    ➔Value of optimal solutions is less than B
  - Feasible?
    ➔Value of optimal solution is at least B

39

Linear Programming in Practice

Many optimization packages available

- Solver (in Excel)
- LINDO
- CPLEX
- GUROBI (free academic license available)
- Matlab, Mathematica

40

More Linear Programming Examples

Typical Operations Research Problem

Brewer's Problem: Maximize Profit

- (1 Barrel) of Ale sells for \$13, but recipe requires
  - 6 pounds corn,
  - 5 ounces of hops and
  - 33 pounds of malt.
- (1 Barrel) of Beer sells for \$23, but recipe requires
  - 16 pounds of corn
  - 4 ounces of hops and
  - 21 pounds of malt
- Suppose we start off with C= 480 pounds of corn, H=160 ounces of hops and M=1190 pounds of malt.
- Let A (resp. B) denote number of barrels of Ale (resp. Beer)

41

More Linear Programming Examples

Typical Operations Research Problem

Brewer's Problem: Maximize Profit

- (1 Barrel) of Ale sells for \$15, but recipe requires
  - 6 pounds corn,
  - 5 ounces of hops and
  - 33 pounds of malt.
- (1 Barrel) of Beer sells for \$27, but recipe requires
  - 16 pounds of corn
  - 4 ounces of hops and
  - 21 pounds of malt
- Suppose we start off with C= 480 pounds of corn, H=160 ounces of hops and M=1190 pounds of malt.
- Let A (resp. B) denote number of barrels of Ale (resp. Beer)
- **Goal:** maximize 15A+27B

42

More Linear Programming Examples

Brewer's Problem: Maximize Profit
- (1 Barrel) of Ale sells for $15, but recipe requires
  - 6 pounds corn, 5 ounces of hops and 33 pounds of malt.
- (1 Barrel) of Beer sells for $27, but recipe requires
  - 16 pounds of corn, 4 ounces of hops and 21 pounds of malt
- Suppose we start off with C= 480 pounds of corn, H=160 ounces of hops and M=1190 pounds of malt.
- Let A (resp. B) denote number of barrels of Ale (resp. Beer)
- **Goal:** maximize 15A+27B (subject to)
  - $A \geq 0, B \geq 0$ (positive production)
  - $6A + 16B \leq C$ (Must have enough CORN)
  - $5A + 4B \leq H$ (Must have enough HOPS)
  - $33A + 21B \leq M$ (Must have enough HOPS)

43

Solving in Mathematica

Maximize[{15 A + 27 B,A>= 0, B>= 0, 6A+16B <= 480, 5A + 4B <= 160, 33A+21 B <= 1190},{A,B}]

{6060/7,{A->80/7,B->180/7}}

Profit: $865.71

44

2-Player Zero-Sum Games

Example: Rock-Paper-Scissors

Alice/Bob	Rock	Paper	Scissors
Rock	(0,0)	(-1,1)	(1,-1)
Paper	(1,-1)	(0,0)	(-1,1)
Scissors	(1,-1)	(1,-1)	(0,0)

Alice wins ➔ Bob loses (and vice-versa)

Minimax Optimal Strategy (possibly randomized) best strategy you can find given that opponent is rational (and knows your strategy)

Minimax Optimal for Rock-Paper-Scissors: play each action with probability 1/3.

45

2-Player Zero-Sum Games

Example: Rock-Paper-Scissors

Alice's View of Rewards
(Bob's are reversed)

Alice/Bob	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Alice wins ➔ Bob loses (and vice-versa)

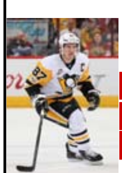
Minimax Optimal Strategy (possibly randomized) best strategy you can find given that opponent is rational (and knows your strategy)

Minimax Optimal for Rock-Paper-Scissors: play each action with probability 1/3.

46

2-Player Zero-Sum Games

Example: Shooter-Goalie



	Block Left	Block Right
Shoot Left	1/2	0.9
shoot Right	0.8	1/3

Shooter scores 80% of time when shooter aims right and goalie blocks left

Minimax Optimal Strategy (possibly randomized) best strategy you can find given that opponent is rational (and knows your strategy)

How can we find Minimax Optimal Strategy?

47

Finding Minimax Optimal Solution using Linear Programming

Variables: $p_1,\ldots,p_n$ and v (p_i is probability of action i)
Goal: Maximize v (our expected reward).

Constraints:
- $p_1, \ldots, p_n \geq 0$
- $p_1 + \ldots + p_n \geq 0$
- For all columns j we have

Expected reward when player 2 takes action j

$$\sum_i p_i m_{ij} \geq v$$

m_{ij} denotes reward when player 1 takes action i and player 2 takes action j.

48

Extra Slides

Circulation with Demands

Circulation with demands.
- Directed graph $G = (V, E)$.
- Edge capacities $c(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$.

demand if $d(v) > 0$; supply if $d(v) < 0$; transshipment if $d(v) = 0$

Def. A circulation is a function that satisfies:
- For each $e \in E$: $0 \le f(e) \le c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem: given (V, E, c, d), does there exist a circulation?

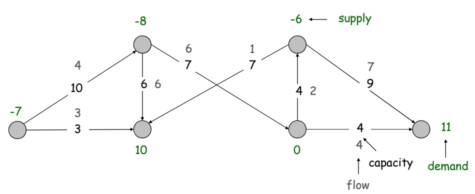
50

Circulation with Demands

Necessary condition: sum of supplies = sum of demands.

$$\sum_{v : d(v) > 0} d(v) = \sum_{v : d(v) < 0} -d(v) =: D$$

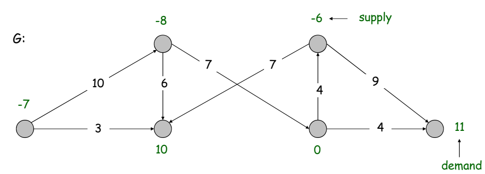
Pf. Sum conservation constraints for every demand node v.



51

Circulation with Demands

Max flow formulation.

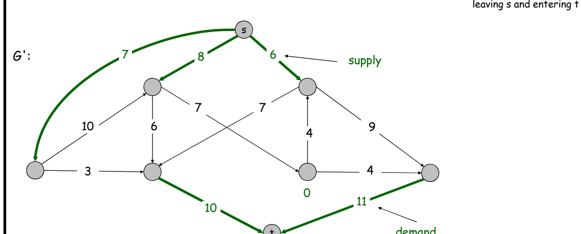


52

Circulation with Demands

Max flow formulation.
- Add new source s and sink t.
- For each v with $d(v) < 0$, add edge (s, v) with capacity $-d(v)$.
- For each v with $d(v) > 0$, add edge (v, t) with capacity $d(v)$.
- Claim: G has circulation iff G' has max flow of value D.

saturates all edges leaving s and entering t



53

Circulation with Demands

Integrality theorem. If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.

Pf. Follows from max flow formulation and integrality theorem for max flow.

Characterization. Given (V, E, c, d), there does not exists a circulation iff there exists a node partition (A, B) such that $\Sigma_{v \in B} d_v > cap(A, B)$

demand by nodes in B exceeds supply of nodes in B plus max capacity of edges going from A to B

Pf idea. Look at min cut in G'.

54

Circulation with Demands and Lower Bounds

Feasible circulation.
- Directed graph G = (V, E).
- Edge capacities c(e) and lower bounds ℓ (e), e $\in$ E.
- Node supply and demands d(v), v $\in$ V.

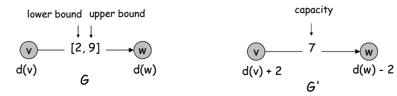
Def. A circulation is a function that satisfies:
- For each e $\in$ E: $\ell(e) \le f(e) \le c(e)$ (capacity)
- For each v $\in$ V: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem with lower bounds. Given (V, E, ℓ, c, d), does there exists a a circulation?

55

Circulation with Demands and Lower Bounds

Idea. Model lower bounds with demands.
- Send ℓ(e) units of flow along edge e.
- Update demands of both endpoints.



Theorem. There exists a circulation in G iff there exists a circulation in G'. If all demands, capacities, and lower bounds in G are integers, then there is a circulation in G that is integer-valued.

Pf sketch. f(e) is a circulation in G iff f'(e) = f(e) - ℓ(e) is a circulation in G'.

56

7.8 Survey Design

Survey Design

Survey design.
- Design survey asking n_1 consumers about n_2 products.
- Can only survey consumer i about product j if they own it.
- Ask consumer i between c_i and c_i' questions.
- Ask between p_j and p_j' consumers about product j.

one survey question per product

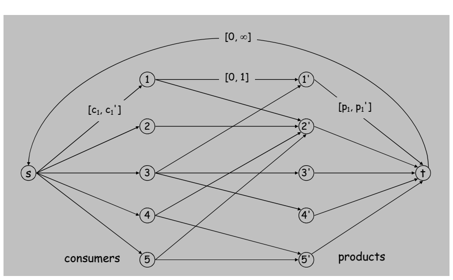
Goal. Design a survey that meets these specs, if possible.

Bipartite perfect matching. Special case when $c_i = c_i' = p_i = p_i' = 1$.

58

Survey Design

Algorithm. Formulate as a circulation problem with lower bounds.
- Include an edge (i, j) if consumer j owns product i.
- Integer circulation $\Leftrightarrow$ feasible survey design.



59

7.10 Image Segmentation

Image Segmentation

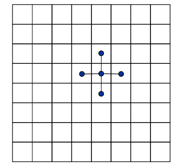
Image segmentation.
- Central problem in image processing.
- Divide image into coherent regions.

Ex: Three people standing in front of complex background scene. Identify each person as a coherent object.

61

Image Segmentation

Foreground / background segmentation.
- Label each pixel in picture as belonging to foreground or background.
- V = set of pixels, E = pairs of neighboring pixels.
- $a_i \geq 0$ is likelihood pixel i in foreground.
- $b_i \geq 0$ is likelihood pixel i in background.
- $p_{ij} \geq 0$ is separation penalty for labeling one of i and j as foreground, and the other as background.



Goals.
- Accuracy: if $a_i > b_i$ in isolation, prefer to label i in foreground.
- Smoothness: if many neighbors of i are labeled foreground, we should be inclined to label i as foreground.
- Find partition (A, B) that maximizes: $\underbrace{\sum_{i \in A} a_i}_{\text{foreground}} + \underbrace{\sum_{j \in B} b_j}_{\text{background}} - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

62

Image Segmentation

Formulate as min cut problem.
- Maximization.
- No source or sink.
- Undirected graph.

Turn into minimization problem.

- Maximizing $\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

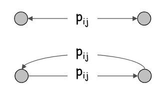
  is equivalent to minimizing $\underbrace{\left(\sum_{i \in V} a_i + \sum_{j \in V} b_j\right)}_{\text{a constant}} - \sum_{i \in A} a_i - \sum_{j \in B} b_j + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

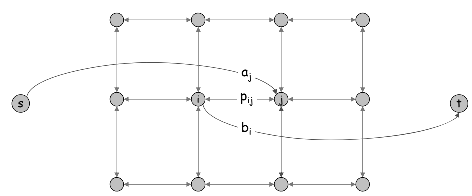
- or alternatively $\sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

63

Image Segmentation

Formulate as min cut problem.
- G' = (V', E').
- Add source to correspond to foreground; add sink to correspond to background
- Use two anti-parallel edges instead of undirected edge.





G'

64

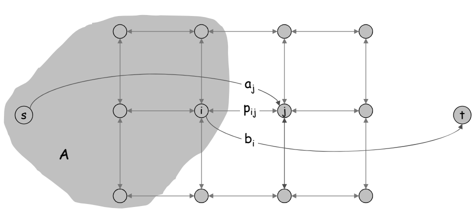
Image Segmentation

Consider min cut (A, B) in G'.
- A = foreground.

$$cap(A, B) = \sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{\substack{(i,j) \in E \\ i \in A,\ j \in B}} p_{ij}$$

if i and j on different sides, p_{ij} counted exactly once

- Precisely the quantity we want to minimize.



G'

65

7.11 Project Selection

Project Selection

can be positive or negative

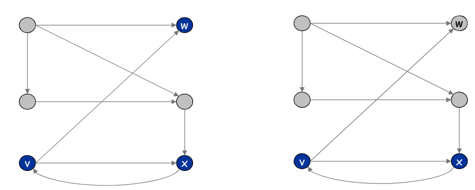
Projects with prerequisites.
- Set P of possible projects. Project v has associated revenue p_v.
  - some projects generate money: create interactive e-commerce interface, redesign web page
  - others cost money: upgrade computers, get site license
- Set of prerequisites E. If $(v, w) \in E$, can't do project v and unless also do project w.
- A subset of projects $A \subseteq P$ is feasible if the prerequisite of every project in A also belongs to A.

Project selection. Choose a feasible subset of projects to maximize revenue.

67

Project Selection: Prerequisite Graph

Prerequisite graph.
- Include an edge from v to w if can't do v without also doing w.
- {v, w, x} is feasible subset of projects.
- {v, x} is infeasible subset of projects.

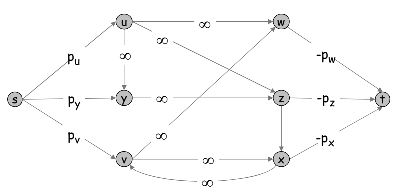


feasible infeasible

68

Project Selection: Min Cut Formulation

Min cut formulation.
- Assign capacity ∞ to all prerequisite edge.
- Add edge (s, v) with capacity p_v if $p_v > 0$.
- Add edge (v, t) with capacity $-p_v$ if $p_v < 0$.
- For notational convenience, define $p_s = p_t = 0$.

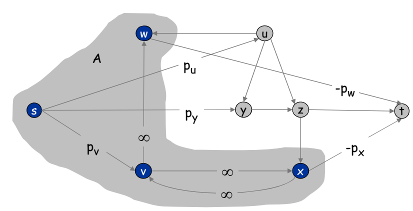


69

Project Selection: Min Cut Formulation

Claim. (A, B) is min cut iff $A - \{s\}$ is optimal set of projects.
- Infinite capacity edges ensure $A - \{s\}$ is feasible.
- Max revenue because:

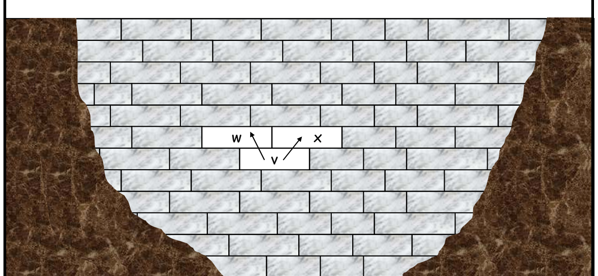
$$\begin{aligned} cap(A, B) &= \sum_{v \in B:\, p_v > 0} p_v + \sum_{v \in A:\, p_v < 0} (-p_v) \\ &= \underbrace{\sum_{v:\, p_v > 0} p_v}_{\text{constant}} - \sum_{v \in A} p_v \end{aligned}$$



70

Open Pit Mining

Open-pit mining. (studied since early 1960s)
- Blocks of earth are extracted from surface to retrieve ore.
- Each block v has net value p_v = value of ore - processing cost.
- Can't remove block v before w or x.

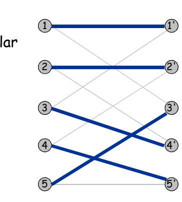


k-Regular Bipartite Graphs

Dancing problem.
- Exclusive Ivy league party attended by n men and n women.
- Each man knows exactly k women; each woman knows exactly k men.
- Acquaintances are mutual.
- Is it possible to arrange a dance so that each woman dances with a different man that she knows?

Mathematical reformulation. Does every k-regular bipartite graph have a perfect matching?

Ex. Boolean hypercube.



72

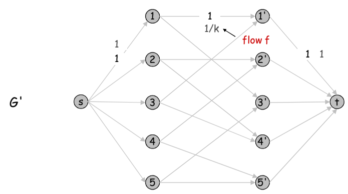
k-Regular Bipartite Graphs Have Perfect Matchings

Theorem. [König 1916, Frobenius 1917] Every k-regular bipartite graph has a perfect matching.

Pf. Size of max matching = value of max flow in G'. Consider flow:

$$f(u,v) = \begin{cases} 1/k & \text{if } (u,v) \in E \\ 1 & \text{if } u = s \text{ or } v = t \\ 0 & \text{otherwise} \end{cases}$$

- f is a flow and its value = n $\Rightarrow$ perfect matching. ▪



Census Tabulation (Exercise 7.39)

Feasible matrix rounding.

- Given a p-by-q matrix D = $\{d_{ij}\}$ of real numbers.
- Row i sum = a_i, column j sum b_j.
- Round each d_{ij}, a_i, b_j up or down to integer so that sum of rounded elements in each row (column) equals row (column) sum.
- Original application: publishing US Census data.

Goal. Find a feasible rounding, if one exists.

3.14	6.8	7.3	17.24
9.6	2.4	0.7	12.7
3.6	1.2	6.5	11.3
16.34	10.4	14.5	

original matrix

3	7	7	17
10	2	1	13
3	1	7	11
16	10	15	

feasible rounding

Census Tabulation

Feasible matrix rounding.

- Given a p-by-q matrix D = $\{d_{ij}\}$ of real numbers.
- Row i sum = a_i, column j sum b_j.
- Round each d_{ij}, a_i, b_j up or down to integer so that sum of rounded elements in each row (column) equals row (column) sum.
- Original application: publishing US Census data.

Goal. Find a feasible rounding, if one exists.
Remark. "Threshold rounding" can fail.

0.35	0.35	0.35	1.05
0.55	0.55	0.55	1.65
0.9	0.9	0.9	

original matrix

0	0	1	1
1	1	0	2
1	1	1	

feasible rounding

Census Tabulation

Theorem. Feasible matrix rounding always exists.

Pf. Formulate as a circulation problem with lower bounds.

- Original data provides circulation (all demands = 0).
- Integrality theorem $\Rightarrow$ integral solution $\Rightarrow$ feasible rounding. ▪

3.14	6.8	7.3	17.24
9.6	2.4	0.7	12.7
3.6	1.2	6.5	11.3
16.34	10.4	14.5