

CS 580: Algorithm Design and Analysis

Jeremiah Blocki
Purdue University
Spring 2018

No Class March 1

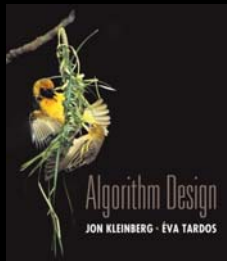
Network Flow Problem

- Directed Graph G with capacities $c(e)$ on each edge
- Source Node: s
- Sink Node: t
- (Max Flow) How much flow can we push from source to sink?
- (Min Cut) Find a minimum capacity $s-t$ cut
 - An $s-t$ cut is a partition (A, B) of V with $s \in A$ and $t \in B$.
- Theorem:** The maximum $s-t$ flow is equal to the minimum $s-t$ cut
- Algorithms to compute maximum $s-t$ flow
 - Ford-Fulkerson
 - Residual Graphs and augmenting paths
 - Can run in exponential time.
 - Capacity Scaling Algorithm $O(m^2 \log C)$
 - Dinic's Algorithm: $O(mn^2)$
- Integrality:** If all capacities $c(e)$ are integral then we can find a max flow $f(e)$ in which the flow $f(e)$ on every edge is integral.

7.5 Bipartite Matching

Midterm Exam 1

- Anticipate having midterm 1 graded at this point
- Look for comments on Piazza
 - Common Mistakes
 - Average, Max, Min, etc...
- Grading Mistake?
 - Contact course staff to ask for regrade
 - Grade could go up or down

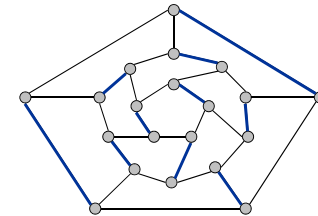


Chapter 7 Network Flow

Slides by Kevin Wayne, Copyright © 2005 Pearson-Addison Wesley. All Rights Reserved.

Matching

- Matching.**
- Input: undirected graph $G = (V, E)$.
 - $M \subseteq E$ is a **matching** if each node appears in at most edge in M .
 - Max matching: find a max cardinality matching.



Bipartite Matching

Bipartite matching.

- Input: undirected, **bipartite** graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a **matching** if each node appears in at most edge in M .
- Max matching: find a max cardinality matching.

7

Bipartite Matching

Max flow formulation.

- Create digraph $G' = (L \cup R \cup \{s, t\}, E')$.
- Direct all edges from L to R, and assign infinite (or unit) capacity.
- Add source s, and unit capacity edges from s to each node in L.
- Add sink t, and unit capacity edges from each node in R to t.

9

Bipartite Matching: Proof of Correctness

Theorem. Max cardinality matching in G = value of max flow in G' .

Pf. \geq

- Let f be a max flow in G' of value k .
- Integrality theorem $\Rightarrow k$ is integral and can assume f is 0-1.
- Consider $M =$ set of edges from L to R with $f(e) = 1$.
 - each node in L and R participates in at most one edge in M
 - $|M| = k$: consider cut $(L \cup s, R \cup t)$

11

Bipartite Matching

Bipartite matching.

- Input: undirected, **bipartite** graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a **matching** if each node appears in at most edge in M .
- Max matching: find a max cardinality matching.

8

Bipartite Matching: Proof of Correctness

Theorem. Max cardinality matching in G = value of max flow in G' .

Pf. \leq

- Given max matching M of cardinality k .
- Consider flow f that sends 1 unit along each of k paths.
- f is a flow, and has cardinality k .

10

Perfect Matching

Def. A matching $M \subseteq E$ is **perfect** if each node appears in exactly one edge in M .

Q. When does a bipartite graph have a perfect matching?

Structure of bipartite graphs with perfect matchings.

- Clearly we must have $|L| = |R|$.
- What other conditions are necessary?
- What conditions are sufficient?

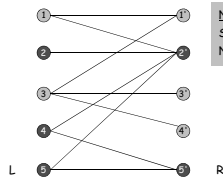
12

Perfect Matching

Notation. Let S be a subset of nodes, and let $N(S)$ be the set of nodes adjacent to nodes in S .

Observation. If a bipartite graph $G = (L \cup R, E)$ has a perfect matching, then $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.

Pf. Each node in S has to be matched to a different node in $N(S)$.

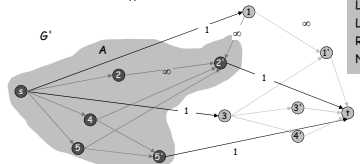


No perfect matching:
 $S = \{2, 4, 5\}$
 $N(S) = \{2', 5'\}$.

Proof of Marriage Theorem

Pf. \Leftarrow Suppose G does not have a perfect matching.

- Formulate as a max flow problem and let (A, B) be min cut in G' .
- By max-flow min-cut, $\text{cap}(A, B) < |L|$.
- Define $L_A = L \cap A$, $L_B = L \cap B$, $R_A = R \cap A$.
- $\text{cap}(A, B) = |L_B| + |R_A|$.
- Since min cut can't use ∞ edges: $N(L_A) \subseteq R_A$.
- $|N(L_A)| \leq |R_A| = \text{cap}(A, B) - |L_B| < |L| - |L_B| = |L_A|$.
- Choose $S = L_A$.



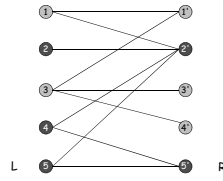
$L_A = \{2, 4, 5\}$
 $L_B = \{1, 3\}$
 $R_A = \{2', 5'\}$
 $N(L_A) = \{2', 5'\}$

7.6 Disjoint Paths

Marriage Theorem

Marriage Theorem. [Frobenius 1917, Hall 1935] Let $G = (L \cup R, E)$ be a bipartite graph with $|L| = |R|$. Then, G has a perfect matching iff $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.

Pf. \Rightarrow This was the previous observation.



No perfect matching:
 $S = \{2, 4, 5\}$
 $N(S) = \{2', 5'\}$.

Bipartite Matching: Running Time

Which max flow algorithm to use for bipartite matching?

- Generic augmenting path: $O(m \text{val}(f^*)) = O(mn)$.
- Capacity scaling: $O(m^2 \log C) = O(m^2)$.
- Shortest augmenting path: $O(m n^{1/2})$.

Non-bipartite matching.

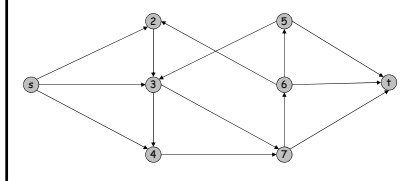
- Structure of non-bipartite graphs is more complicated, but well-understood. [Tutte-Berge, Edmonds-Galai]
- Blossom algorithm: $O(n^3)$. [Edmonds 1965]
- Best known: $O(m n^{1/2})$. [Micali-Vazirani 1980]

Edge Disjoint Paths

Disjoint path problem. Given a digraph $G = (V, E)$ and two nodes s and t , find the max number of edge-disjoint s - t paths.

Def. Two paths are **edge-disjoint** if they have no edge in common.

Ex: communication networks.



Edge Disjoint Paths

Disjoint path problem. Given a digraph $G = (V, E)$ and two nodes s and t , find the max number of edge-disjoint s - t paths.

Def. Two paths are **edge-disjoint** if they have no edge in common.

Ex: communication networks.

Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge.

Theorem. Max number edge-disjoint s - t paths equals max flow value.

Pf. \geq

- Suppose max flow value is k .
- Integrality theorem \Rightarrow there exists 0-1 flow f of value k .
- Consider edge (s, u) with $f(s, u) = 1$.
 - by conservation, there exists an edge (u, v) with $f(u, v) = 1$
 - continue until reach t , always choosing a new edge
- Produces k (not necessarily simple) edge-disjoint paths. •

can eliminate cycles to get simple paths if desired

Edge Disjoint Paths and Network Connectivity

Theorem. [Menger 1927] The max number of edge-disjoint s - t paths is equal to the min number of edges whose removal disconnects t from s .

Pf. \leq

- Suppose the removal of $F \subseteq E$ disconnects t from s , and $|F| = k$.
- Every s - t path uses at least one edge in F .

Hence, the number of edge-disjoint paths is at most k .

Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge.

Theorem. Max number edge-disjoint s - t paths equals max flow value.

Pf. \leq

- Suppose there are k edge-disjoint paths P_1, \dots, P_k .
- Set $f(e) = 1$ if e participates in some path P_i ; else set $f(e) = 0$.
- Since paths are edge-disjoint, f is a flow of value k . •

Network Connectivity

Network connectivity. Given a digraph $G = (V, E)$ and two nodes s and t , find min number of edges whose removal disconnects t from s .

Def. A set of edges $F \subseteq E$ **disconnects t from s** if every s - t path uses at least one edge in F .

Disjoint Paths and Network Connectivity

Theorem. [Menger 1927] The max number of edge-disjoint s - t paths is equal to the min number of edges whose removal disconnects t from s .

Pf. \geq

- Suppose max number of edge-disjoint paths is k .
- Then max flow value is k .
- Max-flow min-cut \Rightarrow cut (A, B) of capacity k .
- Let F be set of edges going from A to B .
- $|F| = k$ and disconnects t from s . •

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?

- 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.

27

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?

29

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!

26

Baseball Elimination

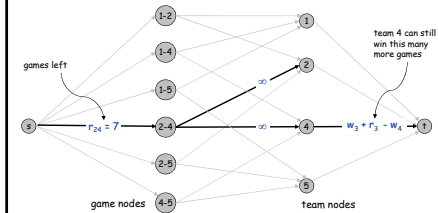


28

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.



30

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 adds to number of wins for team x or team y .
- Capacity on (x, t) edges ensure no team wins too many games.

AL East: August 30, 1996

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

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

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

Certificate of elimination.

$$T \subseteq S, w(T) = \sum_{i \in T} w_i, g(T) = \frac{\sum_{\{x,y\} \subseteq T} g_{xy}}{\sum_{\{x,y\} \subseteq T} 1}$$

LB on avg # games won

If $\frac{w(T)+g(T)}{|T|} > w_c + g_c$ then z is eliminated (by subset T).

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.

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 $y \in T^*$.
- $g(S - \{z\}) > cap(A, B)$

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

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

- Rearranging terms: $w_c + g_c < \frac{w(T^*) + g(T^*)}{|T^*|}$

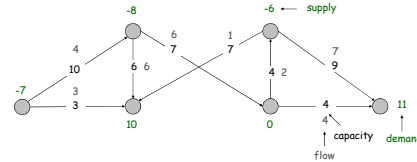
7.7 Extensions to Max Flow

Circulation with Demands

Necessary condition: sum of supplies = sum of demands.

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

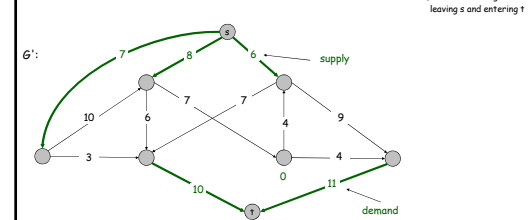
Pf. Sum conservation constraints for every demand node v .



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 .



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 $d(v) = 0$

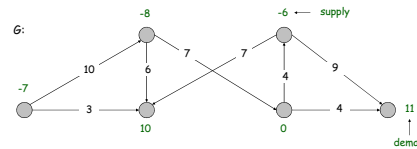
Def. A **circulation** is a function that satisfies:

- For each $e \in E$: $0 \leq f(e) \leq c(e)$ (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?

Circulation with Demands

Max flow formulation.



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** exist a circulation iff there exists a node partition (A, B) such that $\sum_{v \in B} d_v > \text{cap}(A, B)$

Pf idea. Look at min cut in G' .

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

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 $e \in E$: $l(e) \leq f(e) \leq c(e)$ (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 exist a circulation?

43

7.8 Survey Design

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.

47

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' .

44

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 .

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

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

one survey question per product

45

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.

49

Image Segmentation

Formulate as min cut problem.

- Maximization.
- No source or sink.
- Undirected graph.

Turn into minimization problem.

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

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

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

50

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.

51

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 ≥ 0 is likelihood pixel i in foreground.
- b_i ≥ 0 is likelihood pixel i in background.
- p_{ij} ≥ 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: $\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij}$ where $|A \cap \{i,j\}| = 1$

50

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.

50

7.11 Project Selection

Project Selection

can be positive or negative
↓

- Projects with prerequisites.
 - 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.

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 $p_s = p_t = 0$.

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 .

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.

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:

$$cap(A, B) = \sum_{v \in B, p_v > 0} p_v + \sum_{v \in A, p_v < 0} (-p_v)$$

$$= \sum_{v \in B, p_v > 0} p_v - \sum_{v \in A, p_v < 0} p_v$$

constant

Extra Slides

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.

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	

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

original matrix 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. •

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

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	

0	0	1	1
1	1	0	2
1	1	1	

original matrix feasible rounding