Algorithm Design and Implementation

Principle of Algorithms XI

Network Flow II

Guoqiang Li

School of Software, Shanghai Jiao Tong University

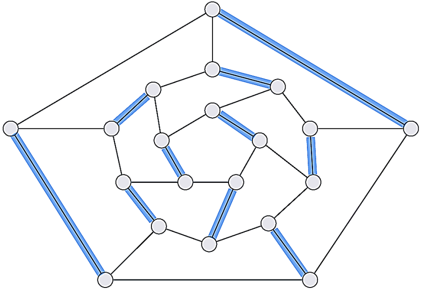
Bipartite Matching

Matching

Definition

Given an undirected graph $G = (V, E)$, subset of edges $M \subseteq E$ is a matching if each node appears in at most one edge in M.

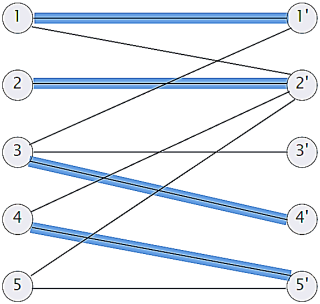
Max matching. Given a graph G, find a max-cardinality matching.



Definition

A graph G is bipartite if the nodes can be partitioned into two subsets L and R such that every edge connects a node in L with a node in R.

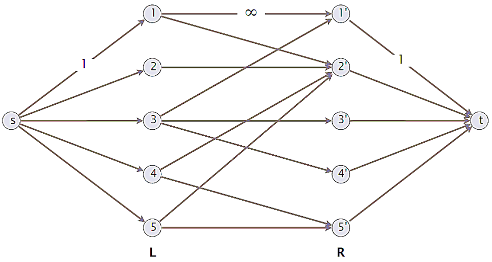
Bipartite matching. Given a bipartite graph $G = (L \cup R, E)$, find a max-cardinality matching.



Bipartite matching: max-flow formulation

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 unit-capacity edges from s to each node in L.
- Add unit-capacity edges from each node in R to t.



Max-flow formulation: proof of correctness

Theorem

1-1 correspondence between matchings of cardinality k in G and integral flows of value k in G'.

Proof. $\Rightarrow$

- Let M be a matching in G of cardinality k.
- Consider flow f that sends 1 unit on each of the k corresponding paths.
- f is a flow of value k.

Max-flow formulation: proof of correctness

Theorem

1-1 correspondence between matchings of cardinality k in G and integral flows of value k in G'.

Proof. $\Leftarrow$

- Let f be an integral flow in G' of value k.
- 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.
  - $|M| = k$: apply flow-value lemma to cut $(L \cup \{s\}, R \cup \{t\})$.

Max-flow formulation: proof of correctness

Theorem

1-1 correspondence between matchings of cardinality k in G and integral flows of value k in G'.

Corollary

Can solve bipartite matching problem via max-flow formulation.

Proof.

- Integrality theorem $\Rightarrow$ there exists a max flow f^* in G' that is integral.
- 1-1 correspondence $\Rightarrow$ f^* corresponds to max-cardinality matching.

Quiz 1

What is running time of Ford–Fulkerson algorithms to find a max-cardinality matching in a bipartite graph with $|L| = |R| = n$?

A. $O(m + n)$
B. $O(mn)$
C. $O(mn^2)$
D. $O(m^2 n)$

Perfect matchings in bipartite graphs

Definition

Given a graph $G = (V, E)$, a subset of edges $M \subseteq E$ is a perfect matching 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.

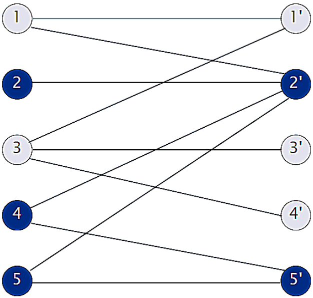
- Clearly, we must have $|L| = |R|$.
- Which other conditions are necessary?
- Which other conditions are sufficient?

Perfect matchings in bipartite graphs

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 $S \subseteq L$.

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



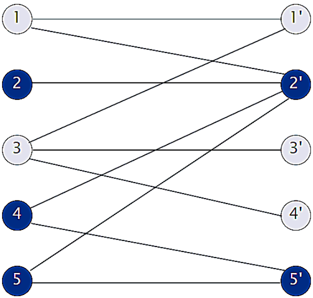
Hall's marriage theorem

Theorem (Frobenius 1917, Hall 1935)

Let $G = (L \cup R, E)$ be a bipartite graph with $|L| = |R|$. Then, graph G has a perfect matching iff $|N(S)| \geq |S|$ for all subsets $S \subseteq L$.

Proof. $\Rightarrow$

This was the previous observation.

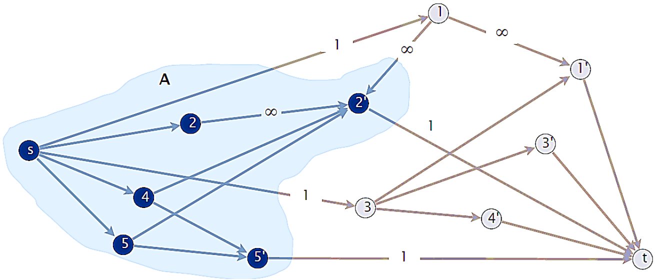


Hall's marriage theorem

Proof. $\Leftarrow$

Suppose G does not have a perfect matching.

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



Bipartite matching

Problem. Given a bipartite graph, find a max-cardinality matching.

year	worst case	technique	discovered by
1955	$O(mn)$	augmenting path	Ford–Fulkerson
1973	$O\left(mn^{1/2}\right)$	blocking flow	Hopcroft–Karp, Karzanov
2004	$O\left(n^{2.378}\right)$	fast matrix multiplication	Mucha–Sankowsi
2013	$\tilde{O}\left(m^{10/7}\right)$	electrical flow	Madry
20xx	???		

running time for finding a max-cardinality matching in a bipartite graph with n nodes and m edges

Quiz 2

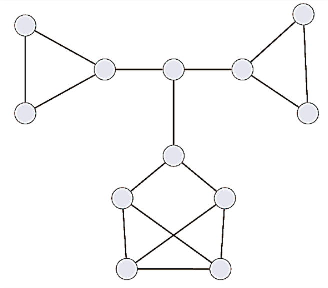
Which of the following are properties of the graph $G = (V, E)$?

A. G has a perfect matching.

B. Hall's condition is satisfied: $|N(S)| \geq |S|$ for all subsets $S \subseteq V$.

C. Both A and B.

D. Neither A nor B.



Nonbipartite matching

Problem. Given an undirected graph, find a max-cardinality matching.

- Structure of nonbipartite graphs is more complicated.
- But well understood. [Tutte–Berge formula, Edmonds–Gallai]
- Blossom algorithm: $O(n^4)$. [Edmonds 1965]
- Best known: $O(mn^{1/2})$. [Micali–Vazirani 1980, Vazirani 1994]

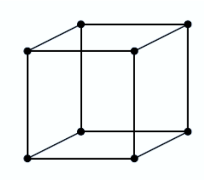
Hackathon problem

Hackathon problem.

- Hackathon attended by n Harvard students and n Princeton students.
- Each Harvard student is friends with exactly $k > 0$ Princeton students; each Princeton student is friends with exactly k Harvard students.
- Is it possible to arrange the hackathon so that each Princeton student pair programs with a different friend from Harvard?

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

Ex. Boolean hypercube.



Hackathon problem

Theorem

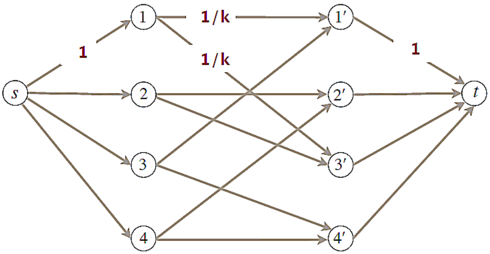
Every k-regular bipartite graph G has a perfect matching.

Proof.

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

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

- The value of flow f is $n \Rightarrow G'$ has a perfect matching.



Disjoint Paths

Edge-disjoint paths

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

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

Max-flow formulation. Assign unit capacity to every edge.

Edge-disjoint paths

Theorem

1-1 correspondence between k edge-disjoint $s \rightsquigarrow t$ paths in G and integral flows of value k in G'.

Proof. $\Rightarrow$

- Let $P_1, \ldots, P_k$ be k edge-disjoint $s \rightsquigarrow t$ paths in G.
- Set $f(e) = \begin{cases} 1 & \text{edge } e \text{ participates in some path } P_j \\ 0 & \text{otherwise} \end{cases}$
- Since paths are edge-disjoint, f is a flow of value k.

Edge-disjoint paths

Theorem

1-1 correspondence between k edge-disjoint $s \rightsquigarrow t$ paths in G and integral flows of value k in G'.

Proof. $\Leftarrow$

- Let f be an integral flow in G' of value k.
- Consider edge (s, u) with $f(s, u) = 1$.
  - by flow 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.

Edge-disjoint paths

Theorem

1-1 correspondence between k edge-disjoint $s \rightsquigarrow t$ paths in G and integral flows of value k in G'.

Corollary

Can solve edge-disjoint paths problem via max-flow formulation.

Proof.

- Integrality theorem $\Rightarrow$ there exists a max flow f^* in G' that is integral.
- 1-1 correspondence $\Rightarrow$ f^* corresponds to max number of edge-disjoint $s \rightsquigarrow t$ paths in G.

Network connectivity

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

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

Menger's theorem

Theorem (Menger 1927)

The max number of edge-disjoint $s \rightsquigarrow t$ paths equals the min number of edges whose removal disconnects t from s.

Proof. $\leq$

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

Menger's theorem

Theorem (Menger 1927)

The max number of edge-disjoint $s \rightsquigarrow t$ paths equals the min number of edges whose removal disconnects t from s.

Proof. $\geq$

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

Quiz 3

How to find the max number of edge-disjoint paths in an undirected graph?

A. Solve the edge-disjoint paths problem in a digraph (by replacing each undirected edge with two antiparallel edges).
B. Solve a max flow problem in an undirected graph.
C. Both A and B.
D. Neither A nor B.

Edge-disjoint paths in undirected graphs

Definition

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

Edge-disjoint paths problem in undirected graphs. Given a graph $G = (V, E)$ and two nodes s and t, find the max number of edge-disjoint s-t paths.

Edge-disjoint paths in undirected graphs

Max-flow formulation. Replace each edge with two antiparallel edges and assign unit capacity to every edge.

Observation. Two paths P_1 and P_2 may be edge-disjoint in the digraph but not edge-disjoint in the undirected graph.

Edge-disjoint paths in undirected graphs

Lemma

In any flow network, there exists a maximum flow f in which for each pair of antiparallel edges e and e' : either $f(e) = 0$ or $f(e') = 0$ or both. Moreover, integrality theorem still holds.

Proof. [by induction on number of such pairs]

- Suppose $f(e) > 0$ and $f(e') > 0$ for a pair of antiparallel edges e and e'.
- Set $f(e) = f(e) - \delta$ and $f(e') = f(e') - \delta$, where $\delta = \mathbf{min}\{f(e), f(e')\}$.
- f is still a flow of the same value but has one fewer such pair.

Edge-disjoint paths in undirected graphs

Lemma

In any flow network, there exists a maximum flow f in which for each pair of antiparallel edges e and e' : either $f(e) = 0$ or $f(e') = 0$ or both. Moreover, integrality theorem still holds.

Theorem

Max number of edge-disjoint $s \rightsquigarrow t$ paths = value of max flow.

Proof. Similar to proof in digraphs; use lemma.

More Menger theorems

Theorem

Given an undirected graph and two nodes s and t, the max number of edge-disjoint s-t paths equals the min number of edges whose removal disconnects s and t.

Theorem

Given an undirected graph and two nodes s and t, the max number of internally node-disjoint s-t paths equals the min number of internal nodes whose removal disconnects s and t.

Theorem

Given an directed graph with two nonadjacent nodes s and t, the max number of internally node-disjoint $s \rightsquigarrow t$ paths equals the min number of internal nodes whose removal disconnects t and s.

Extensions to Max Flow

Quiz 4

Which extensions to max flow can be easily modeled?

A. Multiple sources and multiple sinks.

B. Undirected graphs.

C. Lower bounds on edge flows.

D. All of the above.

Multiple sources and sinks

Definition

Given a digraph $G = (V, E)$ with edge capacities $c(e) \geq 0$ and multiple source nodes and multiple sink nodes, find max flow that can be sent from the source nodes to the sink nodes.

Multiple sources and sinks: max-flow formulation

- Add a new source node s and sink node t.
- For each original source node s_i add edge (s, s_i) with capacity ∞.
- For each original sink node t_j, add edge (t_j, t) with capacity ∞.

Claim

1–1 correspondence betweens flows in G and G'.

Circulation with supplies and demands

Definition

Given a digraph $G = (V, E)$ with edge capacities $c(e) \geq 0$ and node demands $d(v)$, a circulation is a function $f(e)$ that satisfies:

- For each $e \in E$: $0 \leq f(e) \leq c(e)$ capacity
- For each $v \in V$: $\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ flow conservation

Circulation with supplies and 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

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

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

Proof. Follows from max-flow formulation + integrality theorem for max flow.

Theorem

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

Proof sketch. Look at min cut in G'.

Circulation with supplies, demands, and lower bounds

Definition

Given a digraph $G = (V, E)$ with edge capacities $c(e) \geq 0$, lower bounds $\ell(e) \geq 0$, and node demands $d(v)$, a circulation $f(e)$ is a function that satisfies:

- For each $e \in E : \ell(e) \leq f(e) \leq c(e)$ capacity
- For each $v \in V : \sum_{e \text{ into } v} f(e) - \sum_{\text{out of } v} f(e) = d(v)$

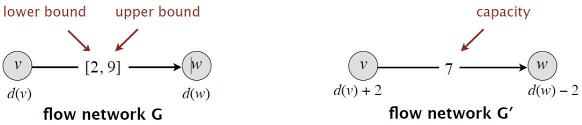
flow conservation

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

Circulation with supplies, demands, and lower bounds

Max-flow formulation. Model lower bounds as circulation with demands.

- Send $\ell(e)$ 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'. Moreover, if all demands, capacities, and lower bounds in G are integers, then there exists a circulation in G that is integer-valued.

Proof sketch. $f(e)$ is a circulation in G iff $f'(e) = f(e) - \ell(e)$ is a circulation in G'.

Survey Design

Survey design

- Design survey asking n_1 consumers about n_2 products.
- Can survey consumer i about product j only 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.

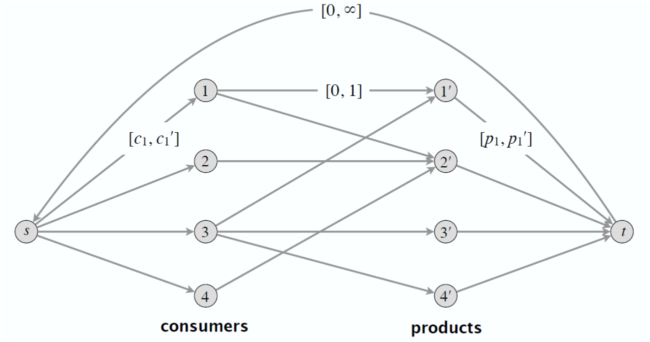
Goal. Design a survey that meets these specs, if possible.

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

Survey design

Max-flow formulation. Model as a circulation problem with lower bounds.

- Add edge (i, j) if consumer j owns product i.
- Add edge from s to consumer j.
- Add edge from product i to t.
- Add edge from t to s.
- All demands $= 0$.
- Integer circulation $\Leftrightarrow$ feasible survey design.



Airline Scheduling

Airline scheduling

Airline scheduling.

- Complex computational problem faced by airline carriers.
- Must produce schedules that are efficient in terms of equipment usage, crew allocation, and customer satisfaction.
- One of largest consumers of high-powered algorithmic techniques.

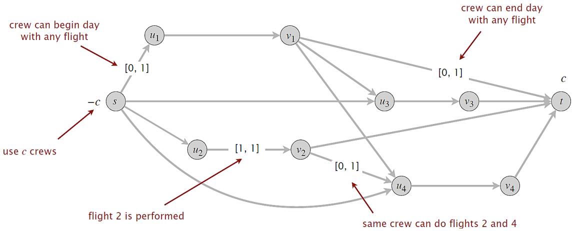
"Toy problem."

- Manage flight crews by reusing them over multiple flights.
- Input: set of k flights for a given day.
- Flight i leaves origin o_i at time s_i and arrives at destination d_i at time f_i.
- Minimize number of flight crews.

Airline scheduling

Circulation formulation. [to see if c crews suffice]

- For each flight i, include two nodes u_i and v_i.
- Add source s with demand $-c$, and edges (s, u_i) with capacity 1.
- Add sink t with demand c, and edges (v_i, t) with capacity 1.
- For each i, add edge (u_i, v_i) with lower bound and capacity 1.
- If flight j reachable from i, add edge (v_i, u_j) with capacity 1.



Airline scheduling: running time

Theorem

The airline scheduling problem can be solved in $O(k^3 \log k)$ time.

Proof.

- k = number of flights.
- c = number of crews (unknown).
- $O(k)$ nodes, $O(k^2)$ edges.
- At most k crews needed.
  $\Rightarrow$ solve $\log_2$ circulation problems. $\leftarrow$ binary search for min value c^*
- Value of any flow is between 0 and k
  $\Rightarrow$ at most k augmentations per circulation problem.
- Overall time = $O(k^3 \log k)$.

Remark

Can solve in $O(k^3)$ time by formulating as minimum-flow problem.

Airline scheduling: postmortem

Remark. We solved a toy version of a real problem.

Real-world problem models countless other factors:

- Union regulations: e.g., flight crews can fly only a certain number of hours in a given time window.
- Need optimal schedule over planning horizon, not just one day.
- Deadheading has a cost.
- Flights don't always leave or arrive on schedule.
- Simultaneously optimize both flight schedule and fare structure

Message.

- Our solution is a generally useful technique for efficient reuse of limited resources but trivializes real airline scheduling problem.
- Flow techniques useful for solving airline scheduling problems and are widely used in practice.
- Running an airline efficiently is a very difficult problem.