Max Flow and Min Cuts: Basics

- Definitions: flow network, flow and its value, s-t cut and its capacity
- Max flow min cut theorem
- Ford-Fulkerson algorithm and its consequences
  - Residual graph, augmenting path; steps of the algorithm
  - Running time for integral capacity networks: $O(mC)$, $C :=$ sum of capacities on edges leaving s.
  - Existence of integral maximum flow on integral capacity networks.
  - Proof of max-flow-min-cut theorem. Computing min cut.
- Existence of polynomial time algorithm (e.g. Edmonds-Karp algorithm)

Max Flow and Min Cuts: Applications

- Bipartite matchings.
  - Maximum bipartite matchings.
  - Hall's theorem and its proof.
- Edge-disjoint paths. (Menger's theorem)
- Dilworth theorem. Partially ordered sets, chains, antichains.
- Vertex cover in bipartite graphs. (König-Egerváry theorem)
- Circulations (possibly with edge lower bounds).
- Image segmentation
- Project selection

NP and computational intractability

- Definitions: Polynomial time reductions, P, NP, NP-complete problems.
- Cook-Levin Theorem: SAT is NP-complete.
- Proving a problem to be NP-complete
  - Prove membership in NP (give a polynomial length certificate and poly-time verifier)
  - Reduce a known NP-complete problem A to the problem B at hand, showing that an instance of A has a solution if and only if the image instance of B does.

Classical NP-complete problems

- 3-SAT
- Packing: Independent Set
- Covering: Vertex Cover, Set Cover
- Sequencing: Hamiltonian Cycle/Path, Traveling Salesman Problem
- Partitioning: 3-Dimensional Matching, Graph Coloring
- Numerical: Subset Sum, Number Partitioning, Knapsack

Exact solutions to NP-hard problems

- Fixed parameter tractable problems: clever exhaustive search.
  - Bounding the running time for recursive algorithms.
  - Common technique: find a way to minimize the "branching" factor.
  - Example: Vertex cover
- Greedy and dynamic programming
  - Independent set on the trees
  - Coloring circular arcs

Approximation algorithms

- Definition: α-approximation algorithms, approximation ratio
- Example greedy approximation algorithms.
  - Load balancing.
  - Center selection (k-center)
  - Set cover
    - Harmonic function $H(n) = \sum_{i=1}^{n} \frac{1}{i} \approx \log n$.

Approximation algorithms

- Common methods
  - Bounding the optimal
  - Express greedy or termination condition
  - Minimum $\leq$ Average $\leq$ Maximum
  - Summing over conditions and noticing double (or multiple) counting
- (Fully) Polynomial Time Approximation Schemes.
  - Example: Knapsack.
  - Dynamic programming for small instances.
  - Round numbers to get a small instance.

Randomized Algorithms

- Probability events, independence.
- Union bound.
- Random variables, expectations.
  - Example: Geometric distribution: toss a (biased) random coin, expected number of tosses till a Heads vs. number of tosses so that a Heads shows up with high probability.
  - Example: Contention resolution: waiting time for one task and all tasks
- Calculating probability of the intersection of events using conditional probabilities
  - Example: Karger's min cut algorithm

Linearity of Expectations

- Linearity of expectations: independence not required!
  - Method: Decompose the count of a random object and use indicator variables.
- Examples:
  - Guessing cards
  - Coupon collection
  - Quicksort
  - Max SAT: $\frac{8}{7}$-approximation.