CSE 417
Algorithms and Complexity
Autumn 2023
Lecture 27
Network Flow Applications
NP-Completeness

Announcements

• Homework 9
• Exam practice problems on course homepage
• Final Exam: Monday, December 11, 8:30 AM
  – One Hour Fifty Minutes

Problem Reduction

• Reduce Problem A to Problem B
  – Convert an instance of Problem A to an instance of Problem B
  – Use a solution of Problem B to get a solution to Problem A
• Practical
  – Use a program for Problem B to solve Problem A
• Theoretical
  – Show that Problem B is at least as hard as Problem A

Minimum Cut Applications

• Image Segmentation
• Open Pit Mining / Task Selection Problem
• Reduction to Min Cut problem

S, T is a cut if S, T is a partition of the vertices with s in S and t in T
The capacity of an S, T cut is the sum of the capacities of all edges going from S to T

Image Segmentation

• Separate foreground from background

12/4/2023  CSE 417  2
Image analysis

- $a_i$: value of assigning pixel $i$ to the foreground
- $b_i$: value of assigning pixel $i$ to the background
- $p_{ij}$: penalty for assigning $i$ to the foreground, $j$ to the background or vice versa
- $A$: foreground, $B$: background
- $Q(A,B) = \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, i \in A, j \in B} p_{ij}$

Pixel graph to flow graph

Min-cut Construction

Open Pit Mining

- Each unit of earth has a profit (possibly negative)
- Getting to the ore below the surface requires removing the dirt above
- Test drilling gives reasonable estimates of costs
- Plan an optimal mining operation

Open Pit Mining (Task selection)

Mine Graph
Determine an optimal mine

Generalization

- Precedence graph $G=(V,E)$
- Each $v$ in $V$ has a profit $p(v)$
- A set $F$ is feasible if when $w$ in $F$, and $(v,w)$ in $E$, then $v$ in $F$.
- Find a feasible set to maximize the profit

Min cut algorithm for profit maximization

- Construct a flow graph where the minimum cut identifies a feasible set that maximizes profit

Precedence graph construction

- Precedence graph $G=(V,E)$
- Each edge in $E$ has infinite capacity
- Add vertices $s$, $t$
- Each vertex in $V$ is attached to $s$ and $t$ with finite capacity edges

Find a finite value cut with at least two vertices on each side of the cut

The sink side of a finite cut is a feasible set

- No edges permitted from $S$ to $T$
- If a vertex is in $T$, all of its ancestors are in $T$
Setting the costs

- If \( p(v) > 0 \),
  - \( \text{cap}(v,t) = p(v) \)
  - \( \text{cap}(s,v) = 0 \)
- If \( p(v) < 0 \)
  - \( \text{cap}(s,v) = -p(v) \)
  - \( \text{cap}(v,t) = 0 \)
- If \( p(v) = 0 \)
  - \( \text{cap}(s,v) = 0 \)
  - \( \text{cap}(v,t) = 0 \)

Minimum cut gives optimal solution
Why?

Computing the Profit

- \( \text{Cost}(W) = \sum_{w \in W; p(w) < 0} -p(w) \)
- \( \text{Benefit}(W) = \sum_{w \in W; p(w) > 0} p(w) \)
- \( \text{Profit}(W) = \text{Benefit}(W) - \text{Cost}(W) \)

Express \( \text{Cap}(S,T) \) in terms of \( B, C, \text{Cost}(T), \text{Benefit}(T), \) and \( \text{Profit}(T) \)

\[
\text{Cap}(S,T) = \text{Cost}(T) + \text{Ben}(S) = \text{Cost}(T) + \text{Ben}(S) + \text{Ben}(T) - \text{Ben}(T) = B + \text{Cost}(T) - \text{Ben}(T) - \text{Profit}(T)
\]

NP-Completeness
Algorithms vs. Lower bounds

- Algorithmic Theory
  - What we can compute
    - I can solve problem X with resources R
  - Proofs are almost always to give an algorithm that meets the resource bounds
- Lower bounds
  - How do we show that something can’t be done?

Theory of NP Completeness

The Universe

NP-Complete

NP

Polynomial Time

- P: Class of problems that can be solved in polynomial time
  - Corresponds with problems that can be solved efficiently in practice
  - Right class to work with “theoretically”

Decision Problems

- Theory developed in terms of yes/no problems
  - Independent set
    - Given a graph G and an integer K, does G have an independent set of size at least K
  - Shortest Path
    - Given a graph G with edge lengths, a start vertex s, and end vertex t, and an integer K, does the graph have a path between s and t of length at most K

What is NP?

- Problems solvable in non-deterministic polynomial time . . .
- Problems where “yes” instances have polynomial time checkable certificates
Certificate examples

- Independent set of size $K$
  - The Independent Set
- Satisfiable formula
  - Truth assignment to the variables
- Hamiltonian Circuit Problem
  - A cycle including all of the vertices
- $K$-coloring a graph
  - Assignment of colors to the vertices

Certifiers and Certificates: 3-Satisfiability

SAT: Does a given CNF formula have a satisfying formula
Certificate: An assignment of truth values to the $n$ boolean variables
Certifier: Check that each clause has at least one true literal.

$$\begin{align*}
(x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_1) \land (x_1 \lor x_2 \lor x_4) \land (x_1 \lor x_3 \lor x_4)
\end{align*}$$

instance $s$

$$x_1 = 1, \ x_2 = 1, \ x_3 = 0, \ x_4 = 1$$

Polynomial time reductions

- $Y$ is Polynomial Time Reducible to $X$
  - Solve problem $Y$ with a polynomial number of computation steps and a polynomial number of calls to a black box that solves $X$
  - Notations: $Y \leq_p X$

- Usually, this is converting an input of $Y$ to an input for $X$, solving $X$, and then converting the answer back

Composability Lemma

- If $X \leq_p Y$ and $Y \leq_p Z$ then $X \leq_p Z$

Lemmas

- Suppose $Y \leq_p X$. If $X$ can be solved in polynomial time, then $Y$ can be solved in polynomial time.

- Suppose $Y \leq_p X$. If $Y$ cannot be solved in polynomial time, then $X$ cannot be solved in polynomial time.
NP-Completeness

- A problem X is NP-complete if
  - X is in NP
  - For every Y in NP, Y \leq_p X

- X is a “hardest” problem in NP

- If X is NP-Complete, Z is in NP and X \leq_p Z
  - Then Z is NP-Complete

Cook's Theorem

- There is an NP Complete problem
  - The Circuit Satisfiability Problem

Populating the NP-Completeness Universe

- Circuit Sat \leq_p 3-SAT
- 3-SAT \leq_p Independent Set
- 3-SAT \leq_p Vertex Cover
- Independent Set \leq_p Clique
- 3-SAT \leq_p Hamiltonian Circuit
- Hamiltonian Circuit \leq_p Traveling Salesman
- 3-SAT \leq_p Integer Linear Programming
- 3-SAT \leq_p Graph Coloring
- 3-SAT \leq_p Subset Sum
- Subset Sum \leq_p Scheduling with Release times and deadlines

Graph Coloring

- NP-Complete
  - Graph 3-coloring
- Polynomial
  - Graph 2-Coloring

Graph 4-Coloring

- Given a graph G, can G be colored with 4 colors?
- Prove 4-Coloring is NP Complete
- Proof: 3-Coloring \leq_p 4-Coloring
- Show that you can 3-Color a graph if you have an algorithm to 4-Color a graph

3-Coloring \leq_p 4-Coloring
P vs. NP Question

Garey and Johnson