# IE418: Integer Programming

Jeff Linderoth

Department of Industrial and Systems Engineering
Lehigh University

23rd February 2005





## Computational Complexity

- The ingredients that we need to build a theory of computational complexity for problem classification are the following
  - A class $\mathcal{C}$ of problems to which the theory applies
  - A (nonempty) subclass $\mathcal{C}_{\mathcal{E}} \subseteq \mathcal{C}$ of “easy” problems
  - A (nonempty) subclass $\mathcal{C}_{\mathcal{H}} \subseteq \mathcal{C}$ of “hard” problems
  - A relation $\lhd$ “not more difficult than” between pairs of problems
- Our goal is just to put some definitions around this machinery
  - Thm: $Q \in \mathcal{C}_{\mathcal{E}}, P \lhd Q \Rightarrow P \in \mathcal{C}_{\mathcal{E}}$
  - Thm: $P \in \mathcal{C}_{\mathcal{H}}, P \lhd Q \Rightarrow Q \in \mathcal{C}_{\mathcal{H}}$

# The “Easy” problems—Class $\mathcal{P}$

- $\mathcal{P}$ is the class of problems for which there exists a polynomial algorithm.
- $\mathcal{P} \in \mathcal{NP} \cap$ co-$\mathcal{NP}$: Why?
- Some problems in $\mathcal{P}$

## Matching

- Given: Graph $G = (V, E), k \in \mathbb{Z}$
- Question: Does $\exists$ a matching $M$ in $G$ with $|M| \geq k$. A matching is a subset of edges such that no two edges share a common endpoint). More mathy: $(i, j) \in M \Rightarrow (i, k) \notin M \forall k \neq j$.





# More Problems in $\mathcal{P}$

## LP

- Given: $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$.
- Question: Does $\exists x \in \mathbb{R}^n_+$ such that $Ax \leq b, c^T x \geq K$?

## Assignment Problem

- Given: set $N = \{1, 2, \dots, n\}$, costs $c_{ij} \in \mathbb{Z}_+ \forall (i, j) \in (N \times N)$
- Question: Does $\exists$ a permutation $\Pi$ of $N$ such that $\sum_{i \in N} c_{i\pi(i)} \geq Q$

## More Problems in $\mathcal{P}$

**Longest Path in a DAG**

- Given: A directed acyclic graph $G = (N, A)$, lengths $\ell_a \in \mathbb{Z}\ \forall a \in A$
- Question: Does $\exists$ a path $P$ in $G$ such that $\sum_{a \in P} \ell_a \geq K$?





## The "Hard Problems"—Class $\mathcal{NPC}$

- We want to ask the question—What are the hardest problems in $\mathcal{NP}$?
    - We'll call this class of problems $\mathcal{NPC}$, "$\mathcal{NP}$-Complete".
- Using the definitions we have made, we would like to say that if $P \in \mathcal{NPC}$, then $Q \in \mathcal{NP} \Rightarrow Q \lhd P$
    - If $P \in NP$ and we can convert in polynomial time *every* other problem $Q \in NP$ to $P$, then $P$ is in this sense the "hardest" problem in $\mathcal{NP}$. $P \in \mathcal{NPC}$
- Is it obvious that such problems exist?
    - **No!** – We'll come to this later...
- Thm: $Q \in \mathcal{P}, P \lhd Q \Rightarrow P \in \mathcal{P}$
- Thm: $P \in \mathcal{NPC}, P \lhd Q \Rightarrow Q \in \mathcal{NPC}$

# Polynomial Reduction

- If problems $P, Q \in \mathcal{NP}$, and if an instance of $P$ can be converted *in polynomial time* to an instance of $Q$, then $P$ is *polynomially reducible* to $Q$.
  - This is the "not (substantially) more difficult than" relation that we want to use.
  - We will write this as $P \lhd Q$
- Depending on time, I will show a couple reductions here.





# Knapsack $\lhd_?$ Longest Path in DAG

**Knapsack Problem**

- Given: set $N$, profits $p_j \in \mathbb{Z}_+ \ \forall j \in N$, sizes: $s_j \in \mathbb{Z}_+ \ \forall j \in N$, $B \in \mathbb{Z}$, $K \in \mathbb{Z}$
- Question: Does $\exists$ a subset $N' \subseteq N$ such that $\sum_{j \in N'} s_j \leq B$ and $\sum_{j \in N'} p_j \geq K$?

construct something here...

**Prove:** $\exists$ a path of length $Q$ in the DAG above $\Leftrightarrow$ $\exists$ a subset $N' \subseteq N$ such that $\sum_{j \in N'} s_j \leq B$ and $\sum_{j \in N'} p_j \geq K$.

# SHOW ME THE MONEY!

- The following statements are true:
  1. Longest Path in DAG is in $\mathcal{P}$
  2. Knapsack is in $\mathcal{NPC}$
- Thm: If $P \cap \mathcal{NPC} \neq \emptyset \Rightarrow \mathcal{P} = \mathcal{NP}$
- Did I just win $1M?
  - Convert Knapsack to Longest Path on DAG, and solve!





# The Satisfiability Problem

- This is the first problem to be shown to be $NP$-complete.
- The problem is described by
  - a finite set $N = \{1, \ldots, n\}$ (the *literals*), and
  - $m$ pairs of subsets of $N$, $C_i = (C_i^+, C_i^-)$ (the *clauses*).
- An instance is feasible if the set

$$\left\{ x \in \mathbb{B}^n \mid \sum_{j \in C_i^+} x_j + \sum_{j \in C_i^-} (1 - x_j) \geq 1 \ \forall i = 1, \ldots, m \right\}$$

  is nonempty.
- This problem is in $\mathcal{NP}$. **Why?**
- In 1971, Cook defined the class $\mathcal{NP}$ and showed that satisfiability was NP-complete.
- We will not attempt to understand the proof

# The Line Between $\mathcal{P}$ and $\mathcal{NPC}$

The line between these two classes is very thin!

---

- Shortest Path (with non-negative edge weights) is in $\mathcal{P}$.
- Longest Path (with non-negative edge weights) is in $\mathcal{NPC}$

**Chinese Postman**

- Given: Undirected graph $G = (V, E), w_e \in \mathbb{Z}_+ \forall e \in E, B \in \mathbb{Z}$
- Question: Does $\exists$ a cycle in $G$ traversing each edge at least once whose total weight is $\leq B$?

Chinese Postman $\in \mathcal{P}$



Review
Relations Between Problems
$NP$-complete proofs
In Conclusion...
Reductions
$\mathcal{NPC}$

# The Line Between $\mathcal{P}$ and $\mathcal{NPC}$

**Directed Chinese Postman**

- Given: Directed Graph $G = (N, A), w_a \in \mathbb{Z}_+ \forall a \in A, B \in \mathbb{Z}$
- Question: Does $\exists$ a cycle in $G$ traversing each arc at least once whose total weight is $\leq B$?

Directed Chinese Postman $\in \mathcal{P}$

**Mixed Chinese Postman**

- Given: Mixed Graph $G = (V, A \cup E), w_e \in \mathbb{Z}_+ \forall e \in (A \cup E), B \in \mathbb{Z}$
- Question: Does $\exists$ a cycle in $G$ traversing each edge and each arc at least once whose total weight is $\leq B$?

Mixed Chinese Postman $\in \mathcal{NPC}$

Jeff Linderoth IE418 Integer Programming

## That Thin, Thin Line

- Consider a 0-1 matrix $A$ an integer $k$ defining the decision problem

$$\exists \{x \in \mathbf{B}^n \, | Ax \leq e, e^T x \geq k\}?$$

- If we limit the number of nonzero entries in each column to 2, then this problem is known to be in $\mathcal{P}$.
  - What is this problem?
- If we allow the number of nonzero entries in each column to be 3, then this problem is $\mathcal{NP}$-complete!
  - What is this problem?
- If we allow at most one '1' per row, the problem is in $\mathcal{P}$
  - Prove it!
- If we allow two '1's per row, it is in $\mathcal{NPC}$





## Proving $\mathcal{NP}$-completeness

- In fact, that is what we will prove now...
- Once we know that satisfiability is $\mathcal{NP}$-complete, we can use this to prove other problems are $\mathcal{NP}$-complete using the "reduction theorem":
  - $P \in \mathcal{NPC}, P \lhd Q \Rightarrow Q \in \mathcal{NPC}$
- Let's prove that **Node Packing** is NP-Complete.

**Node Packing (or Independent Set)**

- Given: Graph $G = (V, E), k \in \mathbb{Z}$
- Question: Does $\exists\, U \subseteq V$ such that $|U| \geq k$ and $U$ is a *node packing*. $(u \in U \Rightarrow v \notin U \ \forall v \in \delta(u))$

# Reduction

Your writing here...





# Other Problems we just proved $\mathcal{NP}$-Complete

- Given graph $G = (V, E)$, its complement is $\bar{G} = (V, \{(i, j) \mid i \in V, j \in V, (i, j) \notin E\})$
  - For the math-illiterate: Same set of vertices, all edges *not* in the original graph.

**Theorem**

*Given $G = (V, E)$. The following statements are equivalent.*

1. *$I$ is an independent set for $G$*
2. *$V \setminus I$ is a vertex cover for $G$*
3. *$I$ is a clique in $\bar{G}$*

# Other NP-Complete Problems

## Maximum Clique

- Given: Graph $G = (V, E), k \in \mathbb{Z}$
- Question: Does $\exists\ U \subseteq V$ such that $|U| \geq k$ and $U$ is a *clique*. $(u \in U \Rightarrow v \in U\ \forall v \in \delta(u))$

**Proof:** Max Independent Set $\lhd$ Maximum Clique.
Given MIS instance: $G_1 = (V, E), q$. Construct MC instance consisting of $G = \bar{G}_1$, and $k = q$. $U$ is a clique of size $q$ in $G$ if and only if $U$ is an independent set of size $k$ in $G_1$.





# Other NP-Complete Problems

## Minimum Vertex Cover

- Given: Graph $G = (V, E), k \in \mathbb{Z}$
- Question: Does $\exists\ U \subseteq V$ such that $|U| \leq k$ and $U$ is a *vertex cover*. $(\forall (u, v) \in E$ either $u \in U$ or $v \in U)$.

**Proof:** Maximum Independent Set $\lhd$ Minimum Vertex Cover.
Given MIS instance: $G_1 = (V, E), q$. Construct MVC instance consisting of $G = G_1$, and $k = |V| - q$. $U$ is a vertex cover of size $k$ in $G$ if and only if $V \setminus U$ is an independent set of size $q$ in $G_1$.

# Other methods of proving $\mathcal{NP}$-Completeness

This is from the Garey and Johnson Handout...

1. Restriction
   - If you can show that a special case of the problem is an $\mathcal{NP}$-complete problem, then the problem must be $\mathcal{NP}$-complete
2. Local Replacement
   - This is like what we did to prove node packing is NP-complete.
3. Component Design
   - These are hard.
   - I welcome you to read and look through some of the proofs for yourself!





# Theory versus Practice

- In practice, it is true that most problem known to be in $\mathcal{P}$ are "easy" to solve.
- This is because most known polynomial time algorithms are of relatively low order.
- It seems very unlikely that $\mathcal{P} = \mathcal{NP}$
- Although all NP-complete problems are "equivalent" in theory, they are not in practice.
- TSP vs. QAP
  - TSP—Solved instances of size $\approx 17000$
  - QAP—Solved instances of size $\approx 30$

## That's It!

- Next Time: Review for midterm
- Midterm on 3/2!