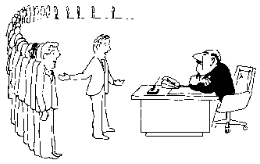


I can't find an efficient algorithm, but neither can all these famous people.

CSE 421 Algorithms

Richard Anderson

Lecture 28

Coping with NP-Completeness

Announcements

- Final exam,
  - Monday, December 12, 2:30-4:20 pm
  - Comprehensive (2/3 post midterm, 1/3 pre midterm)
- Review session
  - Lowe 101
  - Friday, December 9, 2:30-4:20
  - Ben and Max

NP Complete Problems

1. Circuit Satisfiability
2. Formula Satisfiability
   a. 3-SAT
3. Graph Problems
   a. Independent Set
   b. Vertex Cover
   c. Clique
4. Path Problems
   a. Hamiltonian cycle
   b. Hamiltonian path
   c. Traveling Salesman
5. Partition Problems
   a. Three dimensional matching
   b. Exact cover
6. Graph Coloring
7. Number problems
   a. Subset sum
8. Integer linear programming
9. Scheduling with release times and deadlines

Exact Cover (sets of size 3) XC3

Given a collection of sets of size 3 of a domain of size 3N, is there a sub-collection of N sets that cover the sets

{(A, B, C), (D, E, F), (A, B, G), (A, C, I), (B, E, G), (A, G, I), (B, D, F), (C, E, I), (C, D, H), (D, G, I), (D, F, H), (E, H, I), (F, G, H), (F, H, I)}

A B C D E F G H I

Number Problems

- Subset sum problem
  - Given natural numbers $w_1, \ldots, w_n$ and a target number W, is there a subset that adds up to exactly W?
- Subset sum problem is NP-Complete
- Subset Sum problem can be solved in O(nW) time

XC3 $<_P$ SUBSET SUM

Idea: Represent each set as a bit vector, then interpret the bit vectors as integers. Add them up to get the all one's vector.

$\{x_3, x_5, x_9\}$ => 001010001000

Does there exist a subset that sums to exactly 111111111111?

Annoying detail: What about the carries?

Integer Linear Programming

- Linear Programming – maximize a linear function subject to linear constraints
- Integer Linear Programming – require an integer solution
- NP Completeness reduction from 3-SAT

Use 0-1 variables for x_i's

Constraint for clause $x_1 \vee \overline{x_2} \vee \overline{x_3}$

$x_1 + (1 - x_2) + (1 - x_3) > 0$

Scheduling with release times and deadlines

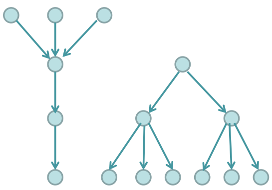
- Tasks $T_1,\ldots,T_n$ with release time r_i, deadline d_i, and work w_i
- Reduce from Subset Sum
  - Given natural numbers $w_1,\ldots, w_n$ and a target number K, is there a subset that adds up to exactly K?
  - Suppose the sum $w_1+\ldots+w_n = W$
- Task T_i has release time 0 and deadline W+1
- Add an additional task with release time K, deadline K+1 and work 1

Coping with NP-Completeness

- Approximation Algorithms
- Exact solution via Branch and Bound
- Local Search

Multiprocessor Scheduling

- Unit execution tasks
- Precedence graph
- K-Processors



- Polynomial time for k=2
- Open for k = constant
- NP-complete is k is part of the problem

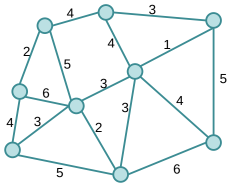
Highest level first is 2-Optimal

Choose k items on the highest level

Claim: number of rounds is at least twice the optimal.

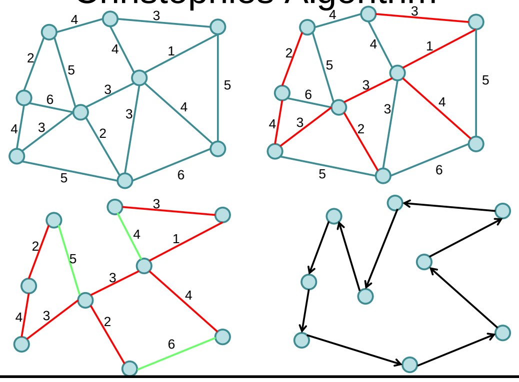
Christofides TSP Algorithm

- Undirected graph satisfying triangle inequality



1. Find MST
2. Add additional edges so that all vertices have even degree
3. Build Eulerian Tour

Christophies Algorithm



Bin Packing

- Given N items with weight w_i, pack the items into as few unit capacity bins as possible
- Example: .3, .3, .3, .3, .4, .4

First Fit Packing

- First Fit
  - Theorem: FF(I) is at most 17/10 Opt(I) + 2
- First Fit Decreasing
  - Theorem: FFD(I) is at most 11/9 Opt (I) + 4

Branch and Bound

- Brute force search – tree of all possible solutions
- Branch and bound – compute a lower bound on all possible extensions
  - Prune sub-trees that cannot be better than optimal

Branch and Bound for TSP

- Enumerate all possible paths
- Lower bound, Current path cost plus MST of remaining points
- Euclidean TSP
  - Points on the plane with Euclidean Distance
  - Sample data set: State Capitals

Local Optimization

- Improve an optimization problem by local improvement
  - Neighborhood structure on solutions
  - Travelling Salesman 2-Opt (or K-Opt)
  - Independent Set Local Replacement