Optimization problems such as

MAXSAT, MIN NODE COVER, MAX INDEPENDENT SET, MAX CLIQUE, MIN SET COVER, TSP, KNAPSACK, BINPACKING

do not have a polynomial time algorithm (unless P=NP). But we have to solve these problems anyway – what do we do?

What to do…

- Worst case exponential time algorithms behaving well in practice.
  - Branch and bound
  - Branch and cut
  - Branch and …..

- Focus on special cases (e.g., special structure, pseudopolynomial algorithms, parameterized complexity)

- Approximation algorithms (5th question on the exam) and approximation heuristics (essentially 6th question on the exam).

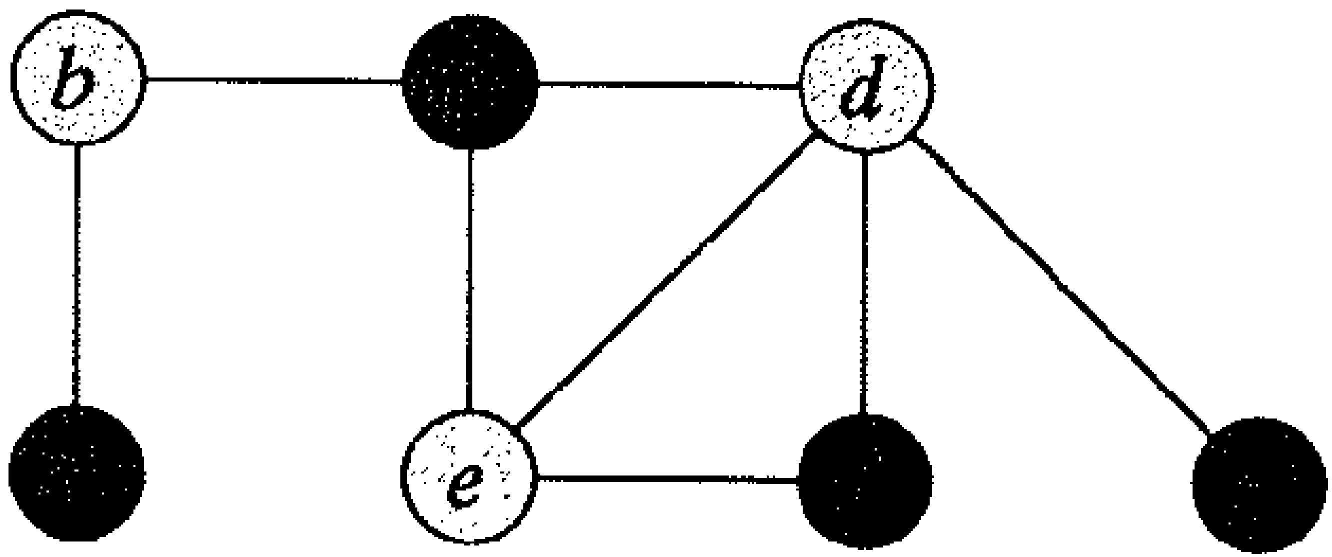
Approximation algorithms

- Given minimization problem (e.g. min vertex cover, TSP,…) and an efficient algorithm that **always returns some feasible solution.**

- The algorithm is said to have **approximation ratio** ρ if for all instances, cost(sol. found)/cost(optimal sol.) $\leq \rho$

Min vertex cover (node cover)

- Given an undirected graph $G=(V,E)$, find the smallest subset $C \subseteq V$ that covers E.



APPROX-VERTEX-COVER(G)

```
C ← ∅
E′ ← E[G]
while E′ ≠ ∅
    do let (u, v) be an arbitrary edge of E′
        C ← C ∪ {u, v}
        remove from E′ every edge incident on either u or v
return C
```

Approximation Ratio

- Approx-Vertex-Cover has approximation ratio 2.

Proof

- Let M be the set of edges chosen.
- M is a matching.
- $|M| \leq$ size of optimal cover
- $|C| = 2|M|$
- $|C| \leq 2 \cdot$ size of optimal cover.

General design/analysis trick

- We do not directly compare our solution to the optimal one – this would be difficult as we typically have no clue about the optimal solution.

- Instead, we compare our solution to some ***lower bound*** (for minimization problems) for the optimal solution.

- Our approximation algorithm often works by constructing some ***relaxation*** providing such a lower bound and turning the relaxed solution into a feasible solution without increasing the cost too much.

Traveling Salesman Problem (TSP)

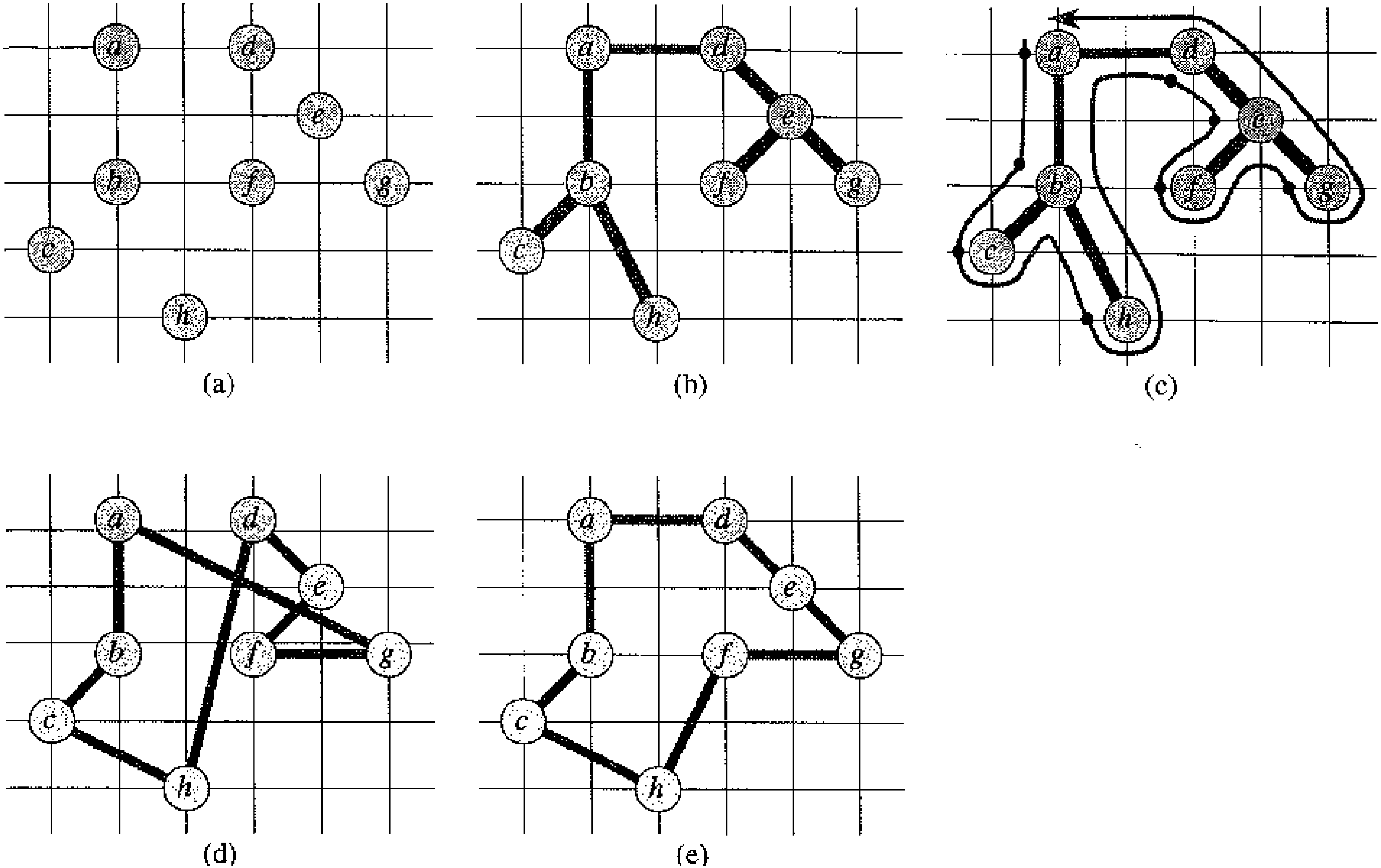
- Given $n \times n$ positive distance matrix (d_{ij}) find permutation π on $\{0,1,2,..,n\text{-}1\}$ minimizing $\sum_{i=0}^{n-1} d_{\pi(i),\, \pi(i+1 \bmod n)}$

- The special case of d_{ij} being actual distances on a map is called the ***Euclidean*** TSP.

- The special case of d_{ij} satistying the triangle inequality is called ***Metric*** TSP. We shall construct an approximation algorithm for the metric case.

Lower bound/relaxation

- What are suitable relaxations of traveling salesman tours?
- Ideas from branch and bound: **Cycle covers** and **minimum spanning trees**.
- We can turn a minimum spanning tree into a traveling salesman tour without increasing the cost too much.


a
b
c
d
e
f
g
h
(a)
(b)
(c)
(d)
(e)

APPROX-TSP-TOUR(G, c)

1 select a vertex $r \in V[G]$ to be a "root" vertex
2 compute a minimum spanning tree T for G from root r
using MST-PRIM(G, c, r)
3 let L be the list of vertices visited in a preorder tree walk of T
4 **return** the hamiltonian cycle H that visits the vertices in the order L

Approximation Ratio

- Approx-TSP-Tour has approximation ratio 2.

Proof

- Let w be the weight of the spanning tree.
- $w \leq$ length of optimal tour
- Length of tour found $\leq 2w$
- Length of tour found $\leq 2 \cdot$ length of optimal tour

Improvements

- Best known algorithm for metric TSP (Christofides algorithm) has approximation factor 3/2.

- Also uses minimum spanning tree, but gets a better tour from it than the preorder walk.

- Why do we only consider metric case?

Appoximating general TSP is NP-hard

- If there is an efficient approximation algorithm for TSP with *any* approximation factor ρ then **P**=**NP**.

- **Proof:** We use a modification of the reduction of hamiltonian cycle to TSP.

Reduction

- Given instance (V,E) of hamiltonian cycle, construct TSP instance (V,d) as follows:

$$
\begin{array}{ll}
d(u,v) = 1 & \text{if } (u,v) \in E \\
d(u,v) = \rho\,|V| + 1 & \text{otherwise.}
\end{array}
$$

- Suppose we have an efficient approximation algorithm for TSP with approximation ratio ρ. Run it on instance (V,d).

- (V,E) has a hamiltonian cycle if and only if the returned solution has length at most $\rho\,|V|$.

Remarks

- The reduction we constructed is called a ***gap creating*** reduction.
- Gap creating reductions have been constructed for many NP-hard optimization problems, proving a limit to their approximability by efficient algorithms.
- For instance, it is known that min vertex cover does not have an algorithm with approximation ratio better than 1.36.
- This reduction and most gap creating reductions are ***much*** harder to construct that the TSP reduction. Constructing them has been a major theme for complexity theory in the 1990’s.

Approximation algorithms

- Given maximization problem (e.g. MAXSAT, MAXCUT) and an efficient algorithm that always returns some feasible solution.

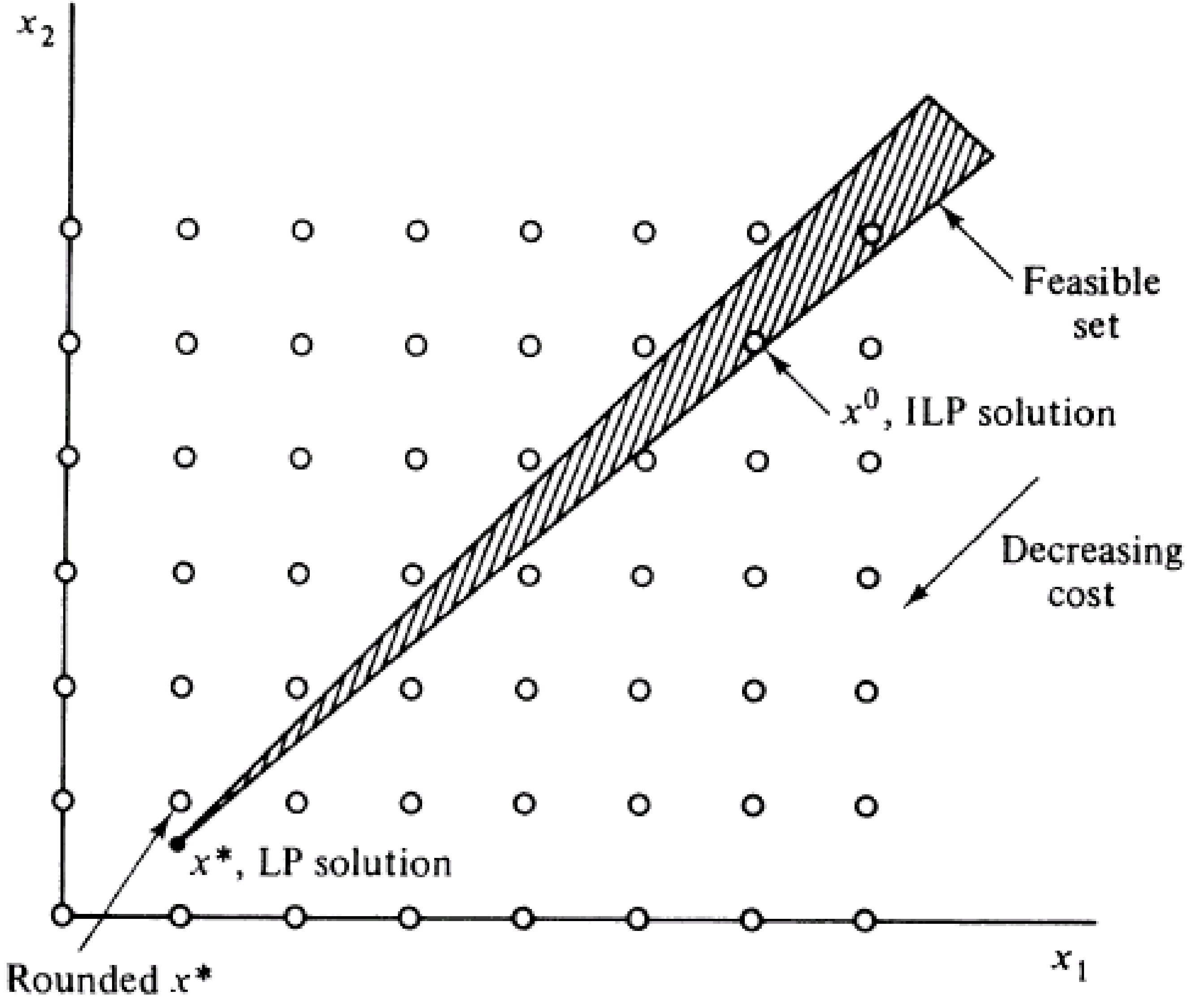
- The algorithm is said to have **approximation ratio** ρ if for all instances, cost(optimal sol.)/cost(sol. found) $\leq \rho$

General design/analysis trick

- Our approximation algorithm often works by constructing some relaxation providing a lower bound and turning the relaxed solution into a feasible solution without increasing the cost too much.

- The LP relaxation of the ILP formulation of the problem is a natural choice. We may then round the optimal LP solution.

Not obvious that it will work….



Min weight vertex cover

- Given an undirected graph $G=(V,E)$ with non-negative weights $w(v)$, find the ***minimum weight*** subset $C \subseteq V$ that covers E.

- Min vertex cover is the case of $w(v)=1$ for all v.

ILP formulation

Find $(x_v)_{v \in V}$ minimizing $\sum w_v x_v$ so that

- $x_v \in \mathbf{Z}$
- $0 \leq x_v \leq 1$
- For all $(u,v) \in E$, $\quad x_u + x_v \geq 1$.

LP relaxation

Find $(x_v)_{v \in V}$ minimizing $\sum w_v x_v$ so that

- $x_v \in \mathbf{R}$
- $0 \leq x_v \leq 1$
- For all $(u,v) \in E$, $\quad \mathrm{x}_{\mathrm{u}} + x_v \geq 1$.

Relaxation and Rounding

- Solve LP relaxation.
- Round the optimal solution x^* to an integer solution x: $x_v = 1$ iff $x^*_v \geq ½$.
- The rounded solution **is** a cover: If $(u,v) \in E$, then $x^*_u + x^*_v \geq 1$ and hence at least one of x_u and x_v is set to 1.

Quality of solution found

- Let $z^* = \sum w_v x_v^*$ be cost of optimal LP solution.

- $\sum w_v x_v \leq 2 \sum w_v x_v^*$, as we only round up if x_v^* is bigger than ½.

- Since $z^* \leq$ cost of optimal ILP solution, our algorithm has approximation ratio 2.

Relaxation and Rounding

- Relaxation and rounding is a very powerful scheme for getting approximate solutions to many NP-hard optimization problems.

- In addition to often giving non-trivial approximation ratios, it is known to be a very good heuristic, especially the **randomized rounding** version.

- Randomized rounding of $x \in [0,1]$: Round to 1 with probability x and 0 with probability $1-x$.

MAX-3-CNF

- Given Boolean formula in CNF form with ***exactly three distinct literals per clause*** find an assignment satisfying as many clauses as possible.

Randomized algorithm

- Flip a fair coin for each variable. Assign the truth value of the variable according to the coin toss.

- **Claim:** The ***expected*** number of clauses satisfied is at least 7/8 m where m is the total number of clauses.

- We say that the algorithm has an ***expected*** approximation ratio of 8/7.

Analysis

- Let Y_i be a random variable which is 1 if the i'th clause gets satisfied and 0 if not. Let Y be the total number of clauses satisfied.

- $\Pr[Y_i = 1] = 1$ if the i'th clause contains some variable and its negation.

- $\Pr[Y_i = 1] = 1 - (1/2)^3 = 7/8$ if the i'th clause does not include a variable and its negation.

- $E[Y_i] = \Pr[Y_i = 1] \geq 7/8$.

- $E[Y] = E[\sum Y_i] = \sum E[Y_i] \geq (7/8)\, m$

Remarks

- It is possible to ***derandomize*** the algorithm, achieving a deterministic approximation algorithm with approximation ratio 8/7.

- Approximation ratio 8/7 - ε is not possible for any constant $\varepsilon > 0$ unless **P**=**NP**. **Very hard to show** (shown in 1997).