Approximation algorithms

How to deal with intractable problems

- Small input size – exponential algorithm can be effective.
- Special subclasses of hard problems can have polynomial-time algorithms
  - You can find an optimal vertex cover for a *tree* in linear time.
  - DNF-SAT, 2-CNF-SAT $\in$ P.
- Find a polynomial-time algorithm to find near-optimal solutions : ***approximation algorithms***

Approximation ratio

- Consider optimization problems whose solutions have a positive cost.
- An algorithm has an ***approximation ratio*** $\rho(n)$ if, for every input of size n, the cost C of the produced solution satisfies

  $$\max \{ C/C^* , C^*/C \} \leq \rho(n) ,$$

  where C^* is the cost of the optimal solution.
- For minimization problems $C^* \leq C$.
- For maximization problems $C \leq C^*$.
- An algorithm with approximation ratio $\rho(n)$ is a ***ρ(n)-approximation algorithm.***

Approximation algorithm for vertex cover

APPROX-VERTEX-COVER(G)

$C = \emptyset$

$E' = E[G]$

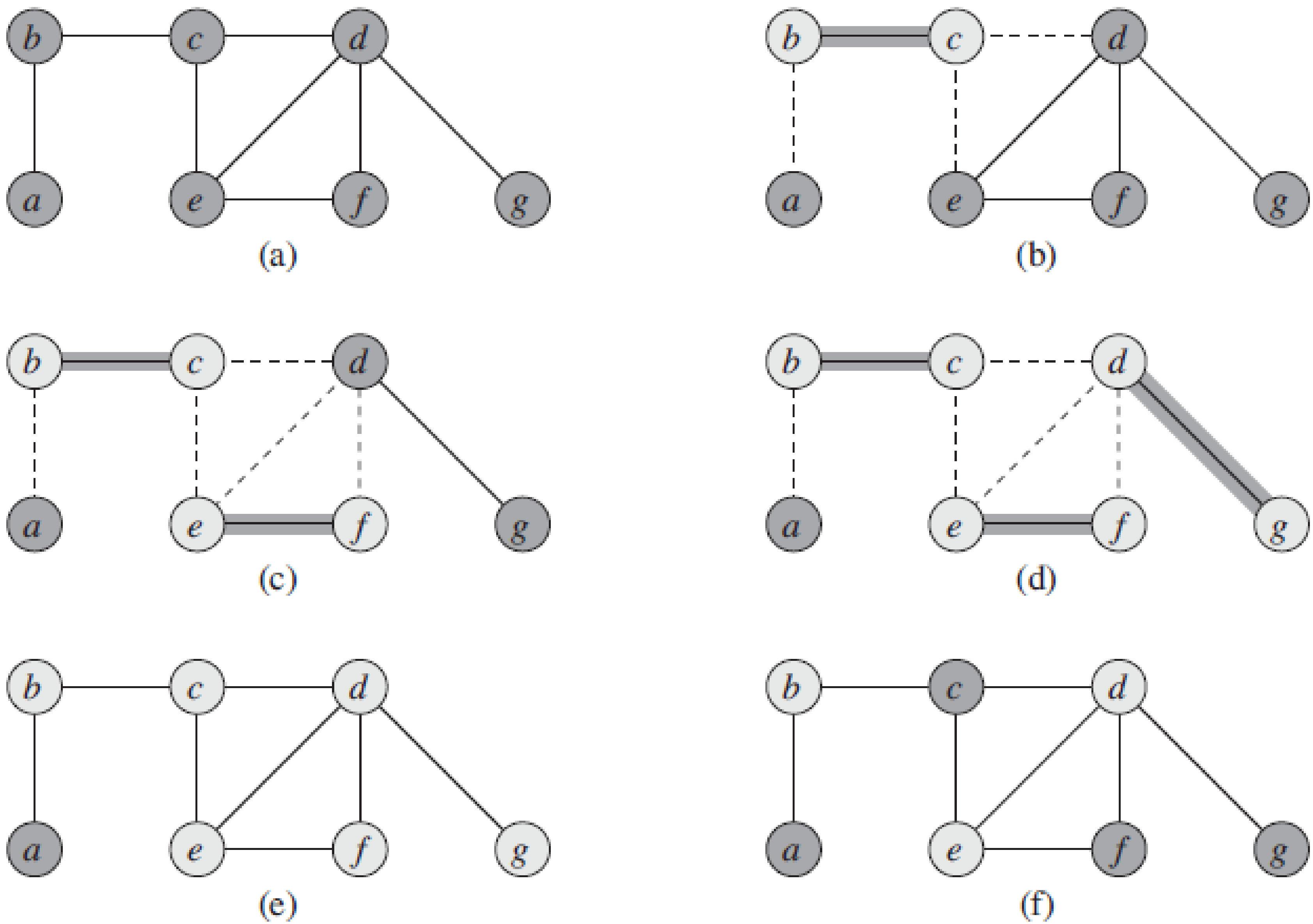
while $E' \neq \emptyset$

do let (u,v) be an arbitrary edge of E'

$C = C \cup \{u, v\}$

remove from E' every edge incident on either u or v

return C


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

Theorem : APPROX-VERTEX-COVER is a polynomial-time 2-approximation algorithm.

Pf) APPROX-VERTEX-COVER runs in $O(V+E)$ time.

C is a vertex cover since the algorithm loops until every edge $E[G]$ has been covered by some vertex in C.

Let A be the set of edges that were picked.

In order to cover the edges in A, any vertex cover (including optimal cover C^*) must include at least one endpoint of each edge in A.

No two edges in A share an endpoint, since once an edge is picked, all other edges incident on its endpoints are deleted.

Thus, no two edges in A are covered by the same vertex from C^*, $|C^*| \geq |A|$. $|C| = 2|A| \leq 2|C^*|$.

Traveling-salesman problem with triangle inequality

- Given a complete undirected graph $G = (V, E)$ that has a nonnegative integer cost $c(u, v)$ for each edge $(u, v) \in E$, find a hamiltonian cycle (a tour) of G with minimum cost.
- Cost function c satisfies the ***triangle inequality*** if, for all vertices $u, v, w \in V$, $c(u, w) \leq c(u, v) + c(v, w)$.
- Many practical applications satisfy the triangle inequality. (e.g., Euclidean distance)
- TSP is NP-complete even if we require that the cost function satisfies the triangle inequality.(Prove it!)
- For TSP with triangle inequality, we have a polynomial-time 2-approximation algorithm.

Approximation algorithm for TSP with triangle inequality

Approx-TSP-Tour(G, c)

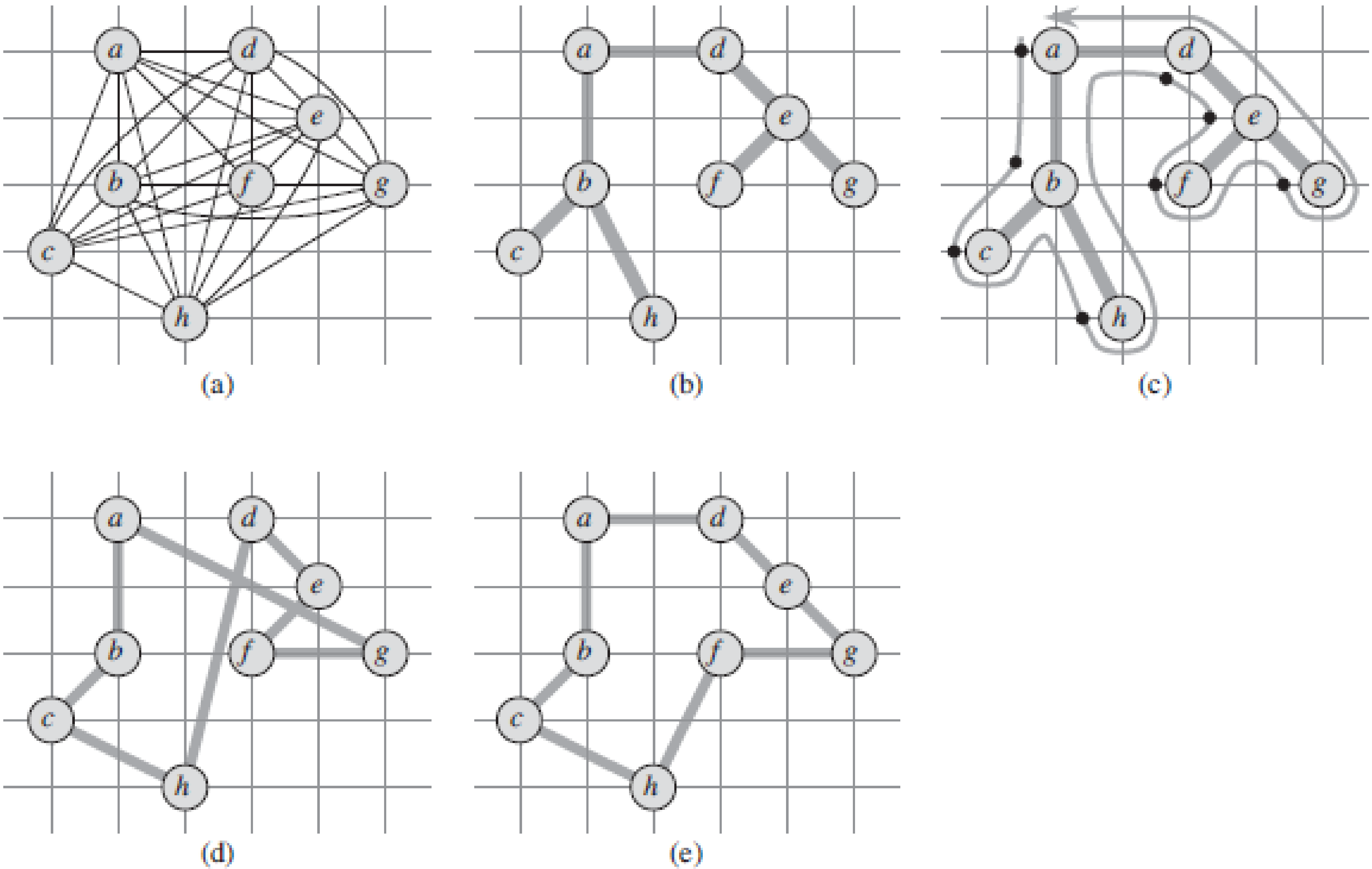
1 select a vertex $r \in V[G]$ to be a "root" vertex

2 compute a MST T for G from root r

using MST-Prim(G, c, r)

3 let H be the list of vertices, ordered according to when they are first visited in a preorder tree walk of T

4 return the hamiltonian cycle H


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

Theorem Approx-TSP-Tour is *a polynomial-time 2-approximation* algorithm with the triangle inequality.

Proof)

1. Show that Approx-TSP-Tour runs in *polynomial time*
2. Prove the approximation ratio 2.

Theorem Approx-TSP-Tour is *a polynomial-time 2-approximation* algorithm with the triangle inequality.

Proof)

1. *polynomial time*

   i) MST-Prim -> $\Theta(V^2)$

   ii) a preorder tree walk -> $\Theta(V)$

   i) + ii) = $\boldsymbol{\Theta(V^2)}$**!!**

Theorem Approx-TSP-Tour is *a polynomial-time 2-approximation* algorithm with the triangle inequality.

Proof)

2. *2-approximation algorithm*

H^* : an optimal tour

T : the MST for given graph

$c(T) \leq c(H^*)$

W : a full walk of T

$c(W) = 2c(T)$

$\therefore\ c(W) \leq 2c(H^*)$

H : the cycle corresponding to the preorder walk

$c(H) \leq c(W)$

$\therefore\ c(H) \leq 2c(H^*)$

$c(H)/c(H^*) \leq 2$

- **Theorem** If P $\neq$ NP, then for any constant $\rho \geq 1$, there is no polynomial-time approximation algorithm with approximation ratio ρ for the general traveling-salesman problem.

Proof Idea) By contradiction.

Suppose that there exists a polynomial-time ρ-approximation algorithm **A** for a constant ρ.

Show how to use **A** to solve hamiltonian-cycle problem (HAM) in polynomial time.

Since HAM is NP-complete, if we can solve it in polynomial time, P=NP.

- **Theorem** If P $\neq$ NP, then for any constant $\rho \geq 1$, there is no polynomial-time approximation algorithm with approximation ratio ρ for the general traveling-salesman problem.

Proof)

Suppose that for some number $\rho \geq 1$, there is a polynomial-time approximation algorithm **A** with approximation ratio ρ.

Let $G=(V, E)$ be an instance of the hamiltonian-cycle problem.

Let $G'=(V, E')$ be the complete graph on V.

$$E'=\{(u, v) : u, v \in V \text{ and } u \neq v\}$$

$$c(u, v)= \begin{cases} 1 & \text{if } (u, v) \in E \\ \rho|V| + 1 & \text{otherwise} \end{cases}$$

Consider the TSP (G', c).

If the graph G has a hamiltonian cycle H, then (G', c) contains a tour of cost $|V|$.

If G does not contain a hamiltonian cycle, then a tour has a cost of at least

$$(\rho|V| + 1) + (|V| - 1) = \rho|V| + |V| > \rho|V|$$

So, we can use **A** to solve the hamiltonian-cycle problem in polynomial time.