Design and Analysis of Algorithm

Lecture-27:

Approximation Algorithm

Contents

Introduction

Many problems of practical significance can only be solved in exponential time.

We have at least three ways to get around exponential problems.

- If the actual inputs are small, an algorithm with exponential running time may be perfectly satisfactory.
- We may be able to isolate important special cases that we can solve in polynomial time.
- We might come up with approaches to find near-optimal solutions in polynomial time

Performance ratios for approximation algorithms

Approximation ratio of an algorithm is defined as

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \le \rho(n)$$

Where,

C is the solution obtained by Algorithm
C^* is the optimal solution

If an algorithm achieves an approximation ratio of $\rho(n)$, we call it a $\boldsymbol{\rho(n)}$ **−approximation algorithm.**

Approximation Scheme

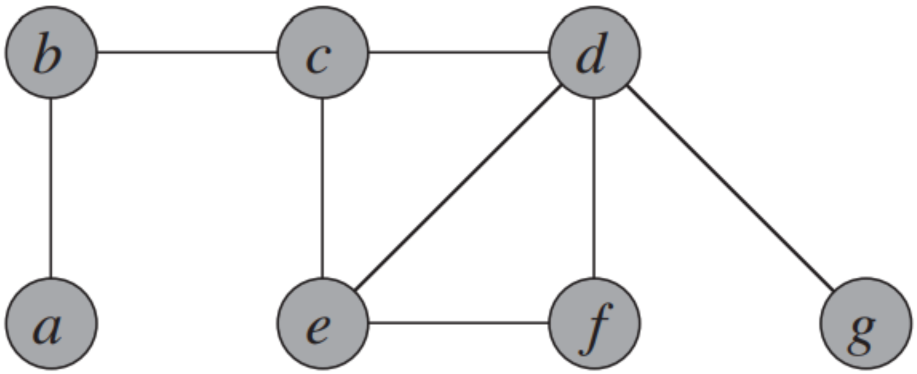
An approximation scheme for an optimization problem is an approximation algorithm that takes as input not only an instance of the problem, but also a value $\in > 0$ such that for any fixed $\in$ the scheme is a $(1 + \in)$ $-$approximation algorithm.

- We say that an approximation scheme is a polynomial-time approximation scheme if for any fixed $\in > 0$, the scheme runs in time polynomial in the size n of its input instance

The vertex-cover problem

A Vertex Cover of a graph G is a set of vertices such that each edge in G is incident to at least one of these vertices.

Let G=(V, E). The subset S of V that meets every edge of E is called the vertex cover.



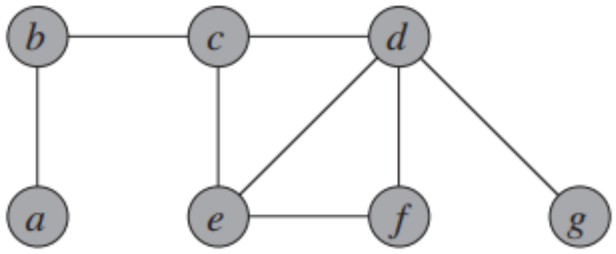
The Vertex Cover problem is to find a vertex cover of the minimum size.

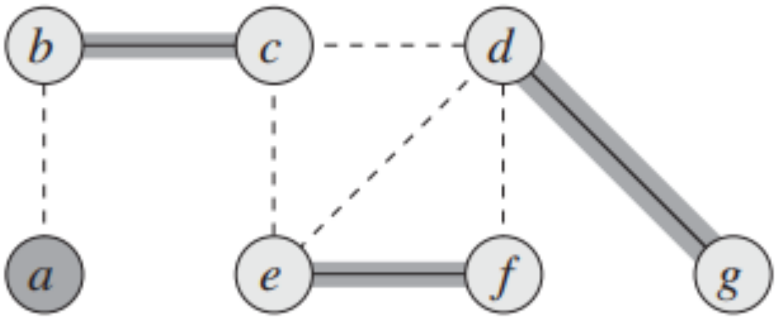
Algorithm

APPROX-VERTEX-COVER(G)

$C = \emptyset$
$E' = G.E$
while $E' \neq \emptyset$
    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

Example



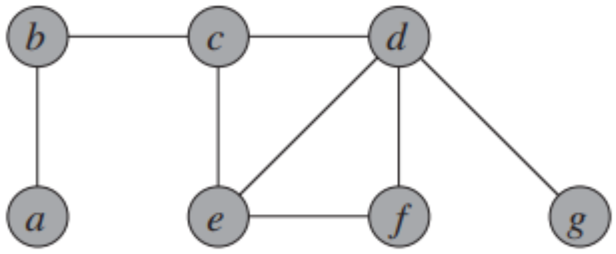


Select Edge → bc

Select Edge → ef

Select Edge → dg

Example



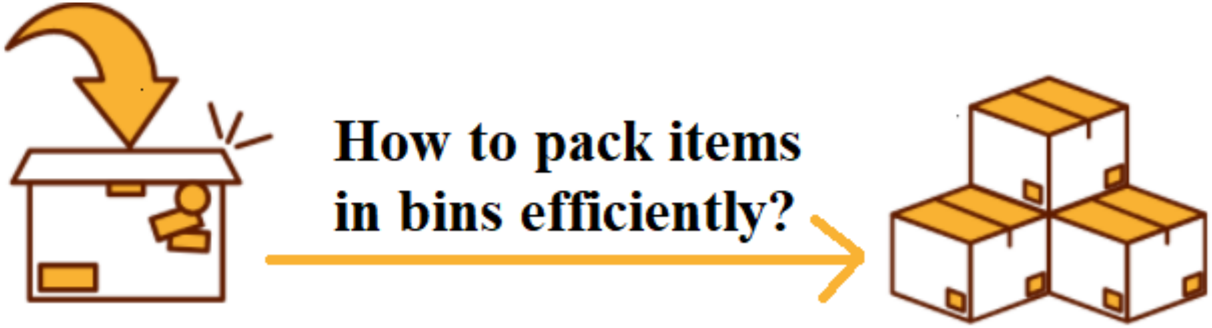
Vertex cover produced by the approximation algorithm is b, c, d, e, f, g

Optimal solution is $b\ , d, e$

Bin Packing Problem

Bin Packing problem involves assigning n items of different weights and bins each of capacity c to a bin such that number of total used bins is minimized.

It may be assumed that all items have weights smaller than bin capacity.



Mathematical Formulation of Bin Packing

Given n items and n knapsacks (or bins), with

$$\boxed{\begin{aligned} W_j &= \text{weight of item } j, \\ c &= \text{capacity of each bin} \end{aligned}}$$

$$\text{minimize} \quad z = \sum_{i=1}^{n} y_i$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_{ij} \leq c y_i, \quad i \in N = \{1, \ldots, n\},$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j \in N,$$

$$y_i = 0 \text{ or } 1, \quad i \in N,$$

$$x_{ij} = 0 \text{ or } 1, \quad i \in N, j \in N,$$

where

$$y_i = \begin{cases} 1 & \text{if bin } i \text{ is used;} \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is assigned to bin } i; \\ 0 & \text{otherwise.} \end{cases}$$

Approximation Algorithms

Input Order dependent Algorithms

- Next Fit algorithm
- First Fit algorithm
- Best Fit algorithm
- Worst Fit algorithm

Input Order Independent Algorithms

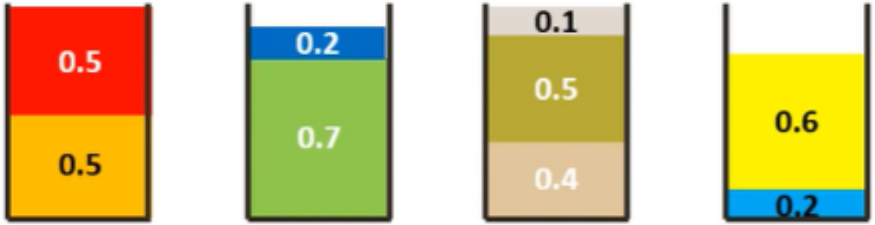
- Next Fit algorithm
- First Fit algorithm
- Best Fit algorithm
- Worst Fit algorithm

Example

Assume

- the sizes of the items be {0.5, 0.7, 0.5, 0.2, 0.4, 0.2, 0.5, 0.1, 0.6}.
- the capacity of each bin is 1

<u>Optimal Output</u>

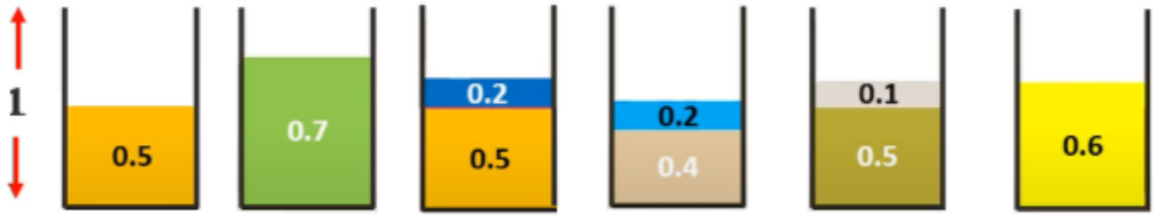


$M_{opt} = 4$

Next Fit Algorithm

Assume

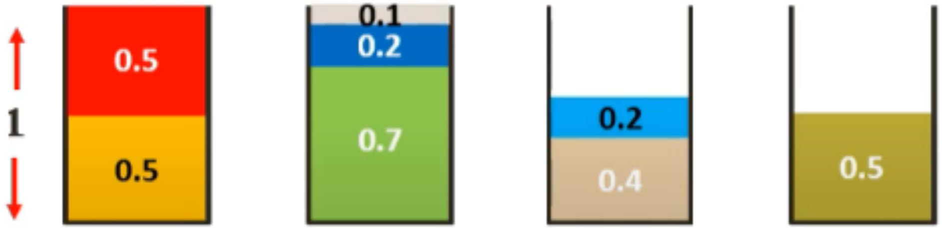
- the sizes of the items be {0.5, 0.7, 0.5, 0.2, 0.4, 0.2, 0.5, 0.1, 0.6}.
- the capacity of each bin is 1



First Fit Algorithm

Assume

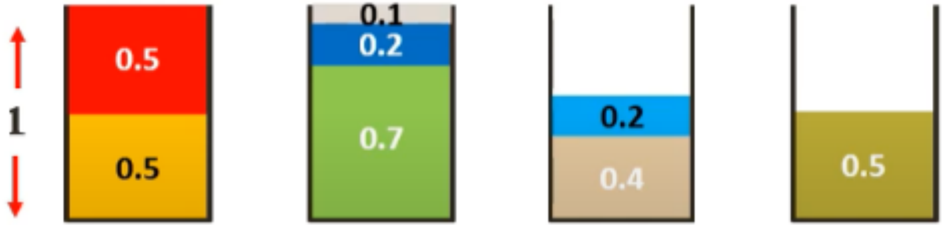
- the sizes of the items be {0.5, 0.7, 0.5, 0.2, 0.4, 0.2, 0.5, 0.1, 0.6}.
- the capacity of each bin is 1



Best Fit Algorithm

Assume

- the sizes of the items be {0.5, 0.7, 0.5, 0.2, 0.4, 0.2, 0.5, 0.1, 0.6}.
- the capacity of each bin is 1



First Fit Algorithm (Independent of Order)

Assume

- the sizes of the items be {0.5, 0.7, 0.5, 0.2, 0.4, 0.2, 0.5, 0.1, 0.6}.
- the capacity of each bin is 1
- the sorted Order {0.7, 0.6, 0.5, 0.5, 0.5, 0.4, 0.2, 0.2, 0.1}