CS F364

Design & Analysis of Algorithms

# COMPLEXITY – OPTIMIZATION PROBLEMS

**Approximation Algorithms**

**- Relative Approximation**

**- Example: Bin-Packing**

# Relative Approximation

- Given an optimization problem $\pi$, for any input instance x and for any feasible solution y, the performance ratio of y is defined as:
  - $R(x,y) = \max( m(x,y)/m^*(x),\ m^*(x)/m(x,y))$
- Given an optimization problem $\pi$, an algorithm A is said to be an r-approximate algorithm if there exists a constant r such that
  - i.e. for any x in $I_\pi$ $R(x,A(x)) \le r$
- Example:
  - Greedy_Vertex_Cover is a 2-approximation algorithm

# RELATIVE APPROXIMATION – EXAMPLE – BIN PACKING

- Problem Definition: BIN PACKING
  - Given a set of N items each with values S1, S2, … Sn, distribute them into equal-sized bins such that the number of bins required is minimum.
- Application:
  - Memory allocation problem (e.g.):
    - Memory is available as fixed-size blocks (i.e. bins)
    - Memory requests come in different sizes
- Note:
  - As the bins are equal-sized, we can assume that they are of unit size and scale the values S1, S2, … Sn accordingly.

# Relative Approximation – Example – Bin Packing [2]

- Formal Problem Definition: BIN PACKING
  - I = { S | S is a finite multi-set of n rational numbers in (0,1] }
  - F(S) = a partition { $B_1$ , $B_2$ , … $B_k$ } of S
    s.t. $\Sigma_{a \text{ in } Bj}$ a <= 1 for each j
  - m(S, {$B_1$ , $B_2$ , … $B_k$ }) = k
  - goal = min

- Definition: Partition
  - { $B_1$ , $B_2$ , … $B_k$ } is said to be a partition of S if
  - $\cup B_j = S$ and $\cap B_j = \{\}$

# Relative Approximation – Example – Bin Packing [3]

- Lower Bound (on the optimal solution):
  - Given input instance S, let $A = \sum_{a \in S} a$
  - Claim:
    - $m^*(x) >= ceil(A)$
  - Proof:
    - Trivial (Perfect packing)

- Algorithm Next_Fit (S)
  1. i=0;
  2. for each item a in S:

     if $((a + \sum_{b \in B_i} b) <= 1)$ { assign a to $B_i$ }

     else { assign a to $B_{i+1}$ ; i = i+1; }

# RELATIVE APPROXIMATION – EXAMPLE – BIN PACKING [5]

- Comments:
  - The technique used by NEXT_FIT is referred to as *Sequencing.*
  - NEXT_FIT is an online algorithm.
- Theorem:
  - NEXT_FIT is a polynomial time 2-approximate algorithm for BIN PACKING.
  - Proof:
    - For each pair of consecutive bins
      - the sum of values assigned to these two bins is > 1
    - i.e. # bins used / 2 <= A where $A = \sum_{a \in S} a$
    - $m_{NEXT\_FIT}(S) <= 2 * ceil(A) <= 2 * m^*(S)$

## RELATIVE APPROXIMATION – EXAMPLE – BIN PACKING [4]

- Claim:
  - The approximation ratio of 2 is asymptotically tight for NEXT_FIT
  - Proof:
    - Given any integer n, consider the instance with 4n items

      S = { 1/2, 1/2n, 1/2, 1/2n, … , 1/2, 1/2n }
    - Optimal solution would require: n+1 bins
    - NEXT_FIT would require: 2*n bins
- Question:
  - What is the weakness of NEXT_FIT ?