Design Techniques for Approximation Algorithms and Approximation Classes

Summary

- Sequential algorithms
- Greedy technique
- Local search technique

Performance ratio

- Given an optimization problem P, an instance x and a feasible solution y, the performance ratio of y with respect to x is

$$R(x,y) = \max(m(x, y)/m^*(x), m^*(x)/m(x, y))$$

- An algorithm is said to be an r-approximation algorithm if, for any instance x, returns a solution whose performance ratio is at most r

Sequential algorithms

- Sequential algorithms are suitable for partition problems.
- In general, a sequential algorithm consists in:
  1. Finding an order in which items are processed
  2. Processing the items sequentially to build the solution

MINIMUM BIN PACKING

- INSTANCE: Finite set I of rational numbers $\{a_1, \ldots, a_n\}$ with $a_i \in (0,1]$

- SOLUTION: Partition $\{B_1, \ldots, B_k\}$ of I into k bins such that the sum of the numbers in each bin is at most 1

- MEASURE: Cardinality of the partition, i.e., k

Sequential algorithms

- Polynomial-time 2-approximation algorithm for MINIMUM BIN PACKING
  - *Next Fit* algorithm

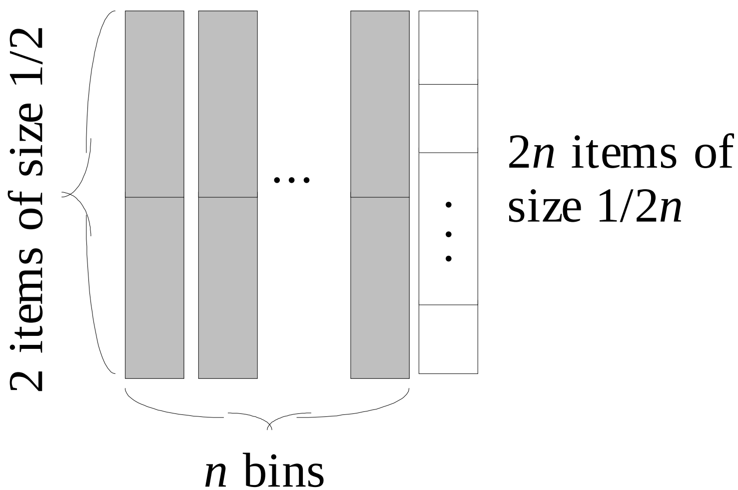
begin
  for each number *a*
    if *a* fits into the last open bin **then** assign *a* to this bin
    else open new bin and assign *a* to this bin
  return *f*
end.

Sequential algorithms

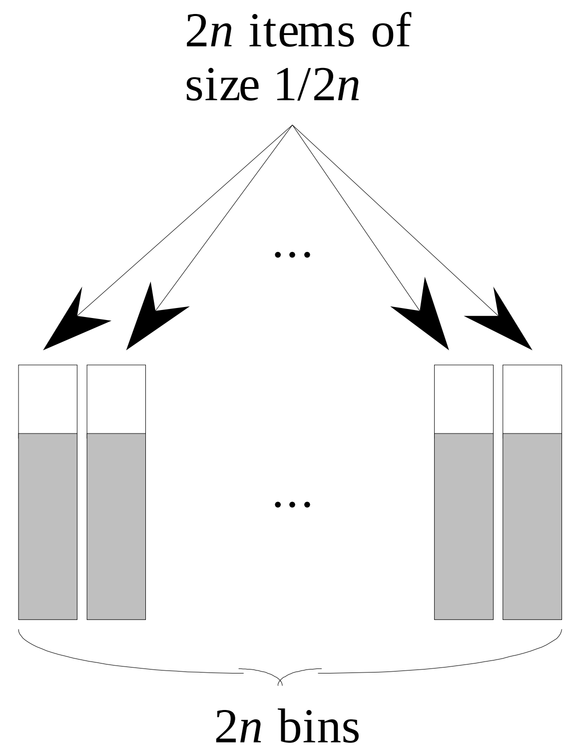
- Proof
  - Number of bins used by the algorithm is less than $2\lceil A \rceil$, where A is the sum of all numbers
    - For each pair of consecutive bins, the sum of the number included in these two bins is greater than 1
  - Each feasible solution uses at least $\lceil A \rceil$ bins
    - Best case each bin is full (i.e., the sum of its numbers is 1)
  - Performance ratio is at most 2

Sequential algorithms

- Tightness: R(x,y)<2
  - Let $I=\{1/2,1/2n,1/2,1/2n,\ldots,1/2,1/2n\}$ contain $4n$ items



optimal packing



Next Fit packing

Sequential algorithms

- Other algorithms for MIN BIN PACKING
  - *First Fit technique:* item a_i is assigned to the first used bin that has enough avaiable space to include it; if no bin can contain it, a new bin is open.
    It can be shown that $m_{FF}(x,y) \leq 1.7m^*(x)+2$
  - *First Fit Decreasing technique:* sort items in non-increasing order and the process by First Fit algorithm.
    It can be shown that $m_{FFD}(x,y) \leq 11/9m^*(x)+7/9$
  - *Best Fit Decreasing technique:* sort items in non-increasing order; pack a_i in the bin with min empty space.
    It can be shown that $m_{BFD}(x,y) \leq 11/9m^*(x)+7/9$

Sequential algorithms

- Gavril's algorithm for vertex cover

```
begin
        U=ø;
        for any edge (u,v) do
                if (u is not in U) and (v is not in U) then
                        insert u and v in U;
        return U
end.
```

- **Theorem**: Gavril's algorithm is a polynomial-time 2-approximation algorithm
  - $m^*(x) \geq m(x,y)/2$ otherwise an edge has not be considered!

MINIMUM GRAPH COLORING

- INSTANCE: Graph $G=(V,E)$

- SOLUTION: A coloring of V, that is, function f such that, for any edge (u,v), $f(u) \neq f(v)$

- MEASURE: Number of colors, i.e., cardinality of the range of f

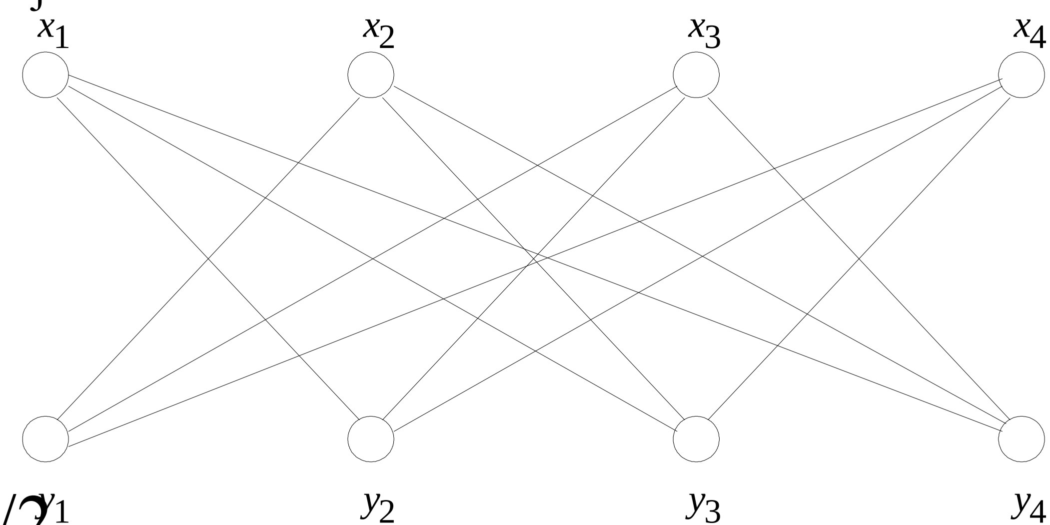
Sequential algorithms

- A bad sequential algorithm for MINIMUM GRAPH COLORING

```
begin
        sort V in decreasing order with respect to the degree;
        for each node v do
                if there exists color not used by neighbors of v then assign this color to v
                else create new color and assign it to v
end.
```

Sequential algorithms

- Example of bad sequential algorithm
  - $G=(\{x_1,\ldots x_n, y_1,\ldots y_n\}, \{\{x_i,y_j\} \mid i\neq j\})$
  - $d(x_i)=d(y_j)=n-1$
  - The order $(x_1,y_1\ldots x_n,y_n)$ requires n colours
  - The optimal value is 2
  - The performance ratio is n/2
  - Generalizing, the performance ratio is $\Delta+1$ where Δ is the highest degree of nodes in G



Greedy technique

- Greedy technique is suitable for problems where instances are set of items and the goal is to find a subset of these items that satisfies the constraints and min/max the measure function.
- In general, a greedy algorithm consists in:
  1. Sorting the items according to some criterion;
  2. Building the solution from a empty set;
  3. Processing sequentially items to put in the solution: the decision is only based on the items that have been already selected.

3. Typically, a greedy algorithm has O(n log n) time complexity

Greedy technique

- A polynomial-time 2-approximation algorithm for MAXIMUM SAT

```
begin
        for any variable v
                begin
                        p := number of clauses, which contain v;
                        q := number of clauses, which contain not v;
                        if p ≥ q then f(v) := TRUE else f(v) := FALSE;
                        simplify the formula;
                end;
        return f
end.
```

Greedy technique

- Proof
  - By induction on the number n of variables, we prove that the algorithm satisfies at least half of the m clauses
    - n=1: trivial
    - Suppose that for n = i, *f()* satisfies at least $m/2$ clauses
    - Inductive step. Let v be the i+1 variable to which a value has been assigned. Assume $p \geq q$ (that is, $f(v)$=TRUE). By induction hypothesis at least

      $$p+(m-p-q)/2 \geq m/2$$

      clauses are satisfied

MAXIMUM INDEPENDENT SET

- INSTANCE: Graph $G=(V,E)$
- SOLUTION: A subset V' of V such that, for any edge (u,v), either u is not in V' or v is not in V'
- MEASURE: Cardinality of V'

Greedy technique

- A bad greedy algorithm for MAXIMUM INDEPENDENT SET

```
begin
        V':=ø; U:=V;
        while U is not empty do
        begin
                x := vertex of minimum degree in graph induced by U;
                insert x in V';
                eliminate x and all its neighbors from U
        end;
        return V'
end.
```

Greedy technique

Example for the greedy algorithm for MAX INDIPENDENT SET

- Optimal solution is I_4
- Found solution is given by external node and one node of K_4
- The performance ratio is 2
- Generalizing, the performance ratio is n/4.