Approximation Classes and Non-Approximability Results (1st part)

Summary

- Randomized algorithms
- Approximation classes
  - The NPO world
- Non-Approximability Results
  - Gap technique
    - Examples: MINIMUM GRAPH COLORING, MINIMUM TSP, MINIMUM BIN PACKING

Randomized algorithms

- *A randomized algorithm is an algorithm that during some of its steps perform random choices*
  - Executing the algorithm several times with the same input it is possible to find different solutions.
- For many application, a randomized algorithm is either simpler or faster than a deterministic one.
  - For combinatorial problem, a randomized algorithm can be seen as a randomized approximation algorithm and the solution found as a random variable
    - The expected value of this random variable is an estimation of the behaviour of the algorithm

Randomized algorithms

- A very simple randomized algorithm for MAX SAT

```
Input: A CNF boolean expression φ on a set V of variables
begin
        for each v∈ V do
                f(v) = TRUE with probability ½;
        end for
        return f;
end
```

Randomized algorithms

- A very simple randomized algorithm for MAX SAT
  - $m_{RS}(x)$ = value of the solution found on input x (random variable).
  - Suppose all c clauses have at least k literals.
  - Theorem: $E[m_{RS}(x)] \geq (1-(1/2)^{k})$ c
    - The probability that any clause with k literals is not satisfied by the truth assignment found is 2^{-k}
    - The expected contribution of a clause with at least k literals to $m_{RS}(x)$ is at least $1-2^{-k}$
    - By summing over all clauses, we obtain the inequality

Randomized algorithms

- A very simple randomized algorithm for MAX SAT
  - The optimal value $m^*(x) \le c$
  - The expected performance ratio is

$$m^*(x)/\mathrm{E}[m_{RS}(x)] \le 2$$

Approaches to the approximation

- There are several approaches
  - Performance guarantee ⬅
    - For all instances the performance ratio is bounded (possibly by a constant)
    - We are interested to determine the algorithm with the minimum performance ratio
    - Be careful to take the worst case analysis as sole reference
  - Randomized algorithm
  - Probabilistic analysis
    - The behaviour of an algorithm is analysed with respect to the "average input" of the problem
  - Heuristics

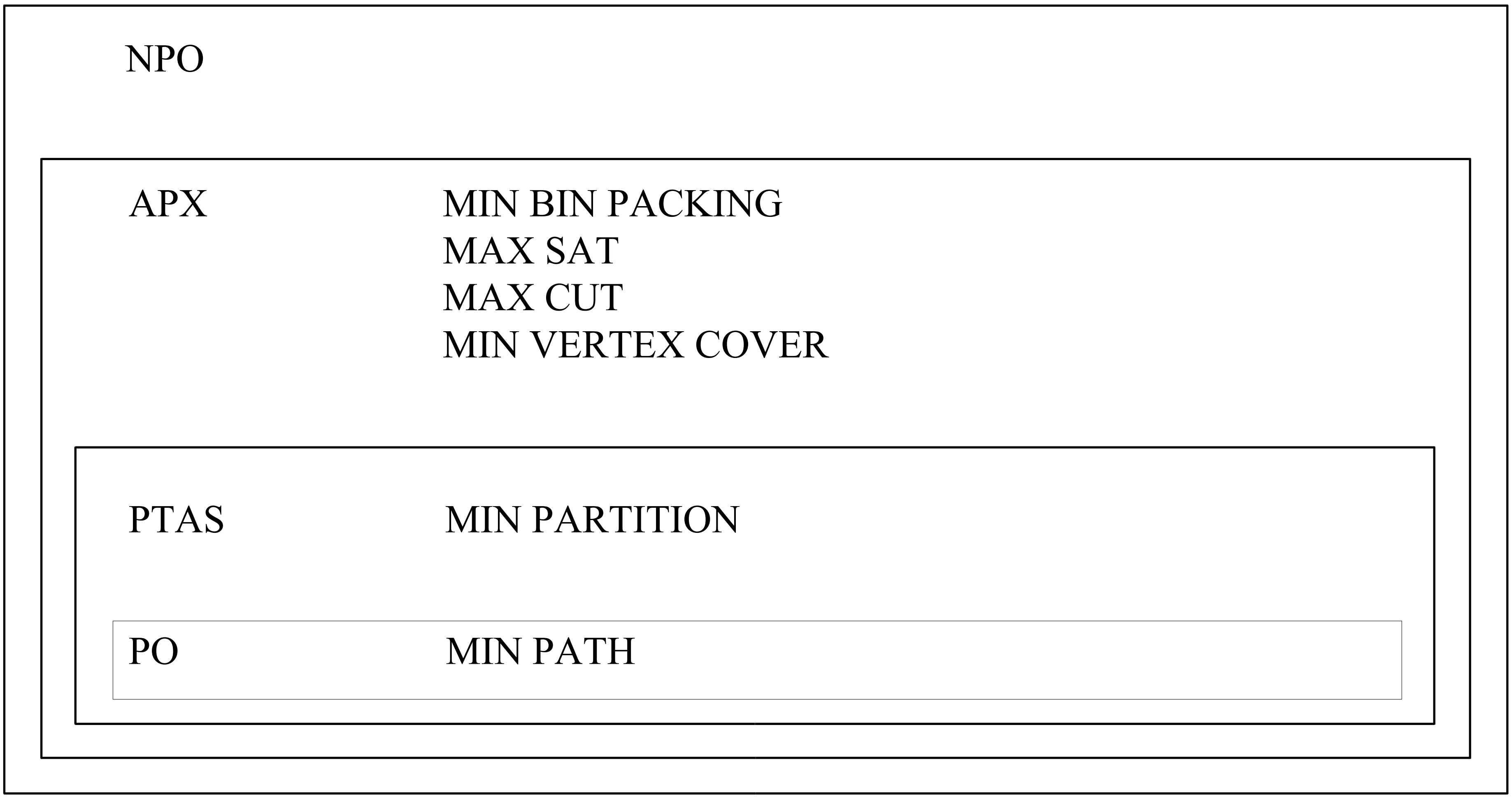
Approximation classes

- Class APX
  - NPO problems that admit a polynomial-time *r*-approximation algorithm, for given constant $r \geq 1$
  - $P \in$ APX is said to be *r*-approximable
  - Examples:
    - MIN BIN PACKING, (sequential 2-approximation a.)
    - MAX SAT, (greedy 2-approximation a.)
    - MAX CUT, (local search 2-approximation a.)
    - MINI VERTEX COVER, (sequential 2-approximation a.)

Approximation classes

- Class PTAS
  - NPO problems that admit a polynomial-time r-approximation algorithm, for any $r > 1$
  - Time must be polynomial in the length of the instance but not necessarily in $1/(r-1)$
    - Time complexity $O(n^{1/(r-1)})$ or $O(2^{1/(r-1)}n^3)$
  - $P \in$ PTAS is said to admit a polynomial-time approximation scheme
  - Example:
    - MINIMUM PARTITION (dynamic r-approximation algorithm)

The NPO world



The Gap Technique

- P_1: NPO minimization problem (same for maximization)
- P_2: NP-hard decision problem (NP-complete problem)
- Function f that maps instances x of P_2 into instances $f(x)$ of P_1
- Function c that maps instances x of P_2 into N^+
- And a constant *gap* g>0 such that:
  - If x is a YES-instance, then $m^*(f(x))=c(x)$
  - If x is a NO-instance, then $m^*(f(x)) \geq c(x)(1+g)$

The Gap Technique

Theorem: No polynomial-time r-approximation algorithm for P_1 with $r<(1+g)$ can exist, unless P=NP.

Proof:

A: r-approximation algorithm with $r<(1+g)$

If x is a YES-instance, then $m^*(f(x))=c(x)$.

Hence, $m(f(x),A(f(x))) \leq rm^*(f(x))=rc(x)<c(x)(1+g)$

If x is a NO-instance, then $m^*(f(x)) \geq c(x)(1+g)$. Hence,

$m(f(x),A(f(x))) \geq c(x)(1+g)$

A allows to decide P_2 in polynomial time

Non-Approximability Results

Inapproximability of graph coloring

- 3-COLORING is NP-complete
  - Any planar graph is 4-colorable
- $f(G)=G$ where G is a planar graph
  - If G is 3-colorable, then $m^*(f(G))=3$
  - If G is not 3-colorable, then $m^*(f(G))=4=3(1+1/3)$
  - Gap: g=1/3

Theorem: MINIMUM GRAPH COLORING has no polinomial time r-approximation algorithm with $r<4/3$ (unless P=NP)

Non-Approximability Results

Inapproximability of bin packing

- NP-hard to decide whether a set of integers I can be partitioned into two equal sets
- $f(I)=(I,B)$ where B is equal to half the total sum
  - If I is a YES-instance, then $m^*(f(I))=2$
  - If G is a NO-instance, then $m^*(f(G)) \geq 3=2(1+1/2)$
  - Gap: $g=1/2$

Theorem: MINIMUM BIN PACKING has no polynomial-time r-approximation algorithm with $r<3/2$ (unless P=NP)

MINIMUM TSP

- INSTANCE: Complete graph $G=(V,E)$, weight function on E

- SOLUTION: A tour of all vertices, that is, a permutation π of V

- MEASURE: Cost of the tour, i.e.,

$$\Sigma_{1\leq k\leq |V|-1} w(v_{\pi[k]}, v_{\pi[k+1]})+w(v_{\pi[|V|]}, v_{\pi[1]})$$

Non-Approximability Results

Inapproximability of TSP

- NP-hard to decide whether a graph contains an Hamiltonian circuit
- For any $g>0$, $f(G=(V,E))=(G'=(V,V^2),w)$ where $w(u,v)=1$ if (u,v) is in E, otherwise $w(u,v)=1+|V|g$
  - If G has an Hamiltonian circuit, then $m^*(f(G))=|V|$
  - If G has no Hamiltonian circuit, then

    $$m^*(f(G)) \geq |V|-1+1+|V|g=|V|(1+g)$$
  - Gap: any $g>0$

Theorem: MINIMUM TSP has no polynomial-time r-approximation algorithm with r>1 (unless P=NP)

The NPO world (unless P=NP)

NPO

MIN TSP

MIN GRAPH COLORING (⇩ ?) (Certainly not in PTAS)

APX

MIN BIN PACKING

MAX SAT (⇩ ?)

MIN VERTEX COVER(⇩ ?)

MAX CUT(⇩ ?)

PTAS

MIN PARTITION

PO

MIN PATH