Approximation Algorithms

Department of Mathematics and Computer Science
Drexel University

Today’s Lecture:

- Wrapping it all up, Approximation classes:
  - Absolute Approximation, and negative results.
  - Relative Approximation:
    - \( r \)-approximation, \( \varepsilon \)-approximation
  - Limits to approximation
    - Gap-technique.
Remember:

- Given optimization problem $P=(I,SOL,m,\text{goal})$, an algorithm $A$ is an approximation algorithm for $P$ if, for any given instance $x$ in $I$, it returns an approximation solution, that is a feasible solution $A(x)$ in Sol($x$).
- We accept a solution as an approximation that is feasible whose solution is “not far” from the optimum.
- Our objective is to find how far a solution is from optimal.

Absolute approximation:

- Given an optimization problem $P$, for any instance $x$ and for any feasible solution $y$ of $x$, the absolute error of $y$ with respect to $x$ is defined as

$$D(x, y) = |m^*(x) - m(x, y)|$$

- Where $m^*(x)$ denotes the measure of an optimal solution of instance $x$ and $m(x, y)$ denotes the measure of solution $y$. 
Absolute approximate algorithm:

- Given an optimization problem $P$, and an approximation algorithm $A$ for $P$, we say that $A$ is an absolute approximation algorithm if there exists a constant $k$ such that, for any instance $x$ of $P$ in $I$, $D(x, A(x)) \leq k$.

- Most of problem such as Minimum Traveling Salesman and Maximum Independent Set, so not allow for polynomial-time absolute approximation algorithms.

Example:

- Positive Example:
  - 6-coloring of a planar graphs.

- Negative results:
  - Unless $P=NP$, no polynomial time absolute approximation algorithm exists for Maximum Knapsack.
Relative approximation:

- Given an approximation problem \( P \), for any instance \( x \) and for any feasible solution \( y \) of \( x \), the relative error of \( y \) with respect to \( x \) is defined as
  \[
  E(x, y) = \frac{|m^*(x) - m(x, y)|}{\max(m^*(x), m(x, y))}
  \]

- Both in the case of maximization and minimization the relative error is 0 when the solution is optimal.

Bonded performance:

- Given an approximation problem \( P \), for any instance \( x \) of \( P \) and for any feasible solution \( y \) of \( x \), the performance of \( y \) with respect to \( x \) is defined as
  \[
  R(x, y) = \max\left(\frac{m(x, y)}{m^*(x)}, \frac{m^*(x)}{m(x, y)}\right)
  \]

- Both in the case of minimization and maximization problems, the value of performance ratio for an optimal solution is 1.
r-approximation algorithm:

- Given an approximation problem $P$, and an approximation algorithm $A$ for $P$, we say $A$ is an $r$-approximate for $P$ if given any instance $x$ of $P$, the performance ratio of approximation solution $A(x)$ is bounded by $r$, that is
  \[ R(x, A(x)) \leq r \]

- Notice that if a given approximation algorithm $A$ for problem $P$, we have that, for all instances $x$ of $P$, $m(x, A(x)) \leq r m^*(x) + k$, then $A$ is $(r+k)$-approximation

r-approximate Max-Sat

- Remember:
  - Set $C$ of disjunctive clauses of a set of variables $V$.
  - Truth assignment $f$ from $V$ to \{True, False\}.
  - Goal: Maximum number of clauses satisfied.

- Greedy Algorithm (program 3.1):
  - Identify the literal with maximum frequency.
  - Set the value appropriately and clean up the function.

- This is 2-approximation algorithm.
Class APX

- NPO problems P that admit a polynomial-time $r$-approximation algorithm, for given constant $r \geq 1$ then P is said to be $r$-approximable

- Examples: MINIMUM BIN PACKING, MAXIMUM SAT, MAXIMUM CUT, MINIMUM VERTEX COVER

TSP

- Is an important example of an NPO that cannot be $r$-approximated, no matter how large the is performance ratio $r$.

- If Minimum TSP belong to APX, then P=NP.
- If P is not equal to NP then APX is a subset of NPO.
Practicality of APX

- In practice knowing that a problem belongs to APX is partially satisfactory.
- For some problems we can find arbitrary close approximate solutions.
- The idea is that, we have two inputs to our algorithm, the instance $x$ and the error $r>1$, and the algorithm can produce an $r$-approximate solution for any given value of $r$.

Limits of approximation and Gap Theorem

- Sometimes the approximation technique can lead to very tight approximation solutions, but then a threshold $t$ exists such that $r$-approximability, with $r < t$, becomes computationally intractable.
- Let $P'$ be an NP-complete decision problem and let $P$ be an NPO minimization approximation problem. Let us assume that there are two polynomial time functions $f$ from instance of $P'$ to instance of $P$ and $c$ from instances of $P'$ to $N$, and a constant $\text{gap}>0$, such that for any instance $x$ of $P'$,

$$m^*(f(x)) = \begin{cases} 
  c(x) & \text{if } x \text{ is a positive instance} \\
  c(x)(1+\text{gap}) & \text{O.W.} 
\end{cases}$$

- Then no polynomial time $r$-approx. algorithm can exist with $r < 1 + \text{gap}$, unless $P=NP$. 

Application:
- **Consider**: minimum graph coloring
  - We will use gap-method as reduction from coloring for planar graphs.
  - Remember planar graphs are colorable with at most 4 colors.
  - The problem of deciding whether a planar graph is colorable with at most 3 colors is NP-complete.

Hardness of graph coloring
- $f(G) = G$ 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: gap = 1/3

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

- **Consider**: bisection-problem
  - We would like to decide whether a set of integers $I$ can be partitioned into two equal sets.
  - The problem is known to be NP-hard.
- Construct an instance of Bin-packing:
  - $f(I) = (I, B)$ where $B$ is the set of bins each 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: $g = 1/2$

Bin-packing

- **Theorem**: MINIMUM BIN PACKING has no $r$-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 $p$ of $V$

- **MEASURE:** Cost of the tour, i.e.,
  $$\sum_{1 \leq k \leq |V|-1} w(v_{p[k]}, v_{p[k+1]}) + w(v_{p[|V|]}, v_{p[1]})$$

Inapproximability of TSB.

- Let us choose the Hamiltonian circuit as NP-Complete Problem.

- Remember:
  - It is 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 $w(u,v)=1+|V|g$
Reduction:

- 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 $g > 0$

- MINIMUM TSP has no $r$-approximation algorithm with $r > 1$ (unless P=NP).