CMPSCI 611 Advanced Algorithms **Fall 2001**

Lecture 25: 12/10

Lecturer: Micah Adler *Scribe: Tom Perkins*

Topics for Today: Approximation Algorithms, especially showing that some problems cannot be efficiently approximated.

25.1 Graph Coloring

Definition 25.1 *G, an undirected graph, is k-colorable if there exists some function $f : v \rightarrow Z$, that maps the vertices of G to the integers from 1 to k, such that if there is an edge between any pair of vertices, those vertices are mapped to different numbers by f.*

Definition 25.2 *The chromatic number of a graph G is the minimum number of colors needed to color the vertices of G such that no adjacent vertices are the same color.*

Claim 25.3 *If $P \neq NP$ then For all $r < \frac{4}{3}$ there is no r-approximation to the chromatic number of a graph.*

Proof: Suppose there was such an r-approximation which uses algorithm A. Then 3-colorability could be solved in polynomial time. Run A on some graph G. If G is 3-colorable, then $A(G) \leq r * 3 < 4$. If G is not 3-colorable, then $A(G) \geq 4$. (There are other techniques which show that the result applies at higher numbers) ■

Claim 25.4 *A Probabilistically checkable proof can show that if $P \neq NP$ then the chromatic number of a graph cannot be approximated within $|V|^{\frac{1}{7}-\epsilon}$ for $\epsilon > 0$.*

Definition 25.5 *ZPP is the class of Las Vegas randomized algorithms with expected polynomial runtimes.*

Claim 25.6 *If $NP \neq ZPP$ then the Chromatic Number of G cannot be approximated within $|V|^{1-\epsilon}$.*

Claim 25.7 *The Chromatic Number of a Planar graph can be decided to within 1 in polynomial time. This is because all Planar graphs are 4-colorable, and 2-colorability can be checked in polynomial time.*

25.2 General Traveling Salesman problem

Claim 25.8 *There is no k-approximation for any constant k unless $P = NP$.*

Proof: We will reduce from the Hamiltonian Cycle problem. The value of k is considered to be known for the purposes of this proof. Assume we have a k-approximation algorithm. Building the graph for the GTSP: Directly transfer V; for each $i, j : i \neq j$: if (i, j) is an edge in the original graph then its cost in the TSP graph is 1, otherwise its cost is $(k+1) * |V|$. If G has a Hamiltonian Cycle, then in the reduced problem, the optimal cost is $|V|$, so the cost found by the algorithm is $\leq k * |V|$. If G does not have a Hamiltonian Cycle, then in the reduced problem, the optimal cost is at least $(k+1) * |V|$, so the cost found by the approximation algorithm would have to be at least that. This would allow us to solve the Hamiltonian Cycle problem in Polynomial time, as the approximation algorithm takes polynomial time. ■

25.3 Types of Approximation

- n^{ϵ} : chromatic number, clique, Independent Set.
- log n : Set Cover

  Definition 25.9 *Set-Cover takes a set S and C a collection of subsets of S. Output: minimum cover C' a subset of C, such that every s in S is in some set in C'.*

- C : vertex cover, metric TSP, max-cut.
- PTAS : Euclidean TSP
- FPTAS : Knapsack problem.

Reminder FPTAS means there is a $1 + \epsilon$ approximation that runs in time polynomial in $\frac{1}{\epsilon}$

Theorem 25.10 *Problem π. If π has the following three properties, then if there is a FPTAS for π, then $P = NP$. Property 1: π is Strongly NP-Complete. Property 2: all values in the input and output are integers. (not strictly necessary for the input, but it makes the proof easier). Property 3: For all possible inputs I, the cost of the optimal solution is less than $q(The\, Input\, Size, Num(I))$, where Num(I) is the largest number in I, and q is a polynomial function.*

Proof: Assume a minimization problem (similar proof can be used in other cases). Let ϵ be $\frac{1}{q(|I|, Num(I))}$. Run the approximation scheme, this will give some solution. Optimal Cost $\leq$ Cost found by algorithm $\leq (1+\epsilon) * Optimal\, Cost$. Equivalently (Cost found by algorithm) - Optimal Cost $\leq \epsilon * Optimal\, Cost < 1$. Since all values are integers, and the difference between the cost the algorithm found and the optimal cost is less than 1, the approximation algorithm found an optimal solution. ■

References

VG2001 V. Gaudin, Scribe notes, Lecture 24, Advanced Algorithms, Fall 2001