CSE 347 Analysis of Algorithms April 15, 2019

Lecture 22 Summary

Guest lecturer: Zihao Deng

1. It is easy to reduce Vertex Cover to *integer* linear programming
   (a) In integer linear programming, the feasible solutions are vectors of integers. The constraints and objective are the same as for standard linear programming.
   (b) If we let our integer variables denote whether or not a vertex is selected for the cover, then we just need the sum over these variables to be at least 1 for each edge, and our objective is to minimize the total sum of these variables. Thus, we get an integer linear program that corresponds exactly to a Vertex Cover instance that we can generate in polynomial time.
   (c) Unfortunately, this means that solving an integer linear program is NP-hard.
2. We *relax* the integer linear program to a linear program, that we can solve efficiently. We then *round* the (fractional) solution to the linear program to obtain an approximately-optimal vertex cover.
   (a) "Relaxing" the program just means that we drop the constraint that the solutions must be integers. Since linear programming and integer linear programming use the same constraints otherwise, this gives us a linear program that we can solve in polynomial time.
   (b) Unfortunately, the fractional solution may "cheat" by assigning fractional values to the vertices that merely sum to 1 for each edge. The objective value for the linear program could then be far smaller than is possible for any actual vertex cover.
   (c) But, since each edge has only two endpoints, at least one of the endpoints of each edge must have a fractional value of at least 1/2. So if we "round" values of at least 1/2 up to 1 (and round values less than 1/2 down to 0) then the rounded up vertices give a vertex cover.
   (d) Since we are only rounding up values that were at least 1/2, the objective value is at most twice what it was previously. Since the optimum for the linear program is at most the optimum vertex cover's size (the integer optimum is a feasible solution), we therefore obtain a vertex cover of at most twice the optimal size.
3. If we want to get $(1 \pm \epsilon)$-approximate solutions for all ϵ, this is called a *Polynomial Time Approximation Scheme* (PTAS). If the problem is intractable to solve exactly, we can't hope to get a PTAS that runs in polynomial time in terms of $1/\epsilon$ (in addition to the problem size). Thus, we only seek to guarantee that the PTAS runs in polynomial time for each fixed ϵ.