CMSC 858F: Algorithmic Lower Bounds
Fall 2014
Approximation Hardness

Instructor: Mohammad T. Hajiaghayi
Scribe: Saeed Seddighin

September 30, 2014

1 Overview

Most of the problems that we have in real-world are NP-hard and cannot be solved in polynomial time by the methods that we know so far. However, it is important to find a way to overcome the hardness even though we cannot solve them efficiently. Here are three algorithmic goals that we have regarding real-world problems:

1. Finding exact solutions. (i.e. optimal solution)
2. Solving problems efficiently. (i.e. polynomial time)
3. Solving hard problems. (i.e. NP-hard problems)

Assuming P$\neq$NP we cannot satisfy all three goals together. Therefore we need to sacrifice at least one of the goals.

If we give up on solving hard problems we can only wish for solving a very small fraction of algorithmic problems and the solutions are usually well-known methods that we learned in undergraduate courses.

If we don't want to sacrifice the correctness and we still want to solve hard problems, we should design an algorithm that does not run in polynomial time but finds the exact solution of the problem. Again, we can not completely ignore the running time, therefore we introduce the Fixed Parameter Tractable algorithms in order to find the solution in almost polynomial time. ($\mathtt{f}(\mathtt{x}) \times \mathtt{poly}(\mathtt{n})$).

If we sacrifice the first goal, then we have an NP-hard problem and we want to solve it in polynomial time. Obviously, we cannot completely ignore the correctness of the algorithms, but we can do something in between. Thet's where the notion of *approximation algorithms* shows up. Of course, in the ideal case we are looking for an approximation algorithm that finds a $1+\epsilon$ approximate solution for the problem in polynomial time for all $\epsilon > 0$ but as we see throughout this lecture, such algorithms do not exist for all problem.

Definition 1 *A* **PTAS** *is an algorithm which takes an instance of an optimization problem and a parameter $\epsilon > 0$ and, in polynomial time, produces a solution that is within a factor $1+\epsilon$ of being optimal (or $1-\epsilon$ for maximization problems). For example, for the Euclidean traveling salesman problem, a PTAS would produce a tour with length at most $(1+\epsilon)L$, with L being the length of the shortest tour. The running time of a PTAS is required to be polynomial in n for every fixed ϵ but can be different for different ϵ.*

2 Approximability & Inapproximability

In this lecture we will see that although minimization and maximization problems are the same (you can reduce one to another by just negating all the numbers), we have different levels of inapproximability for problems with different objective functions.

2.1 Approximation Algorithms in General

Approximation algorithms are the most popular ways to combat NP-hardness. Consider any problem $X \in$NP and the goal maximization or minimization. We say an algorithm is c-approximation if (assuming that the running time of the algorithm in polynomial) the goal is minimizing the objective function and

$$\frac{\mathsf{Cost}(\mathsf{Alg}(X))}{\mathsf{Cost}(\mathsf{Opt}(X))} \leq C$$

or the goal is maximizing the objective function and

$$\frac{\mathsf{Cost}(\mathsf{Opt}(X))}{\mathsf{Cost}(\mathsf{Alg}(X))} \leq C.$$

2.2 Approximability of Minimization problems

In this section we talk about different levels of approximability for minimization problems.

2.2.1 $\theta(1)$ approximation

This class includes all the problems that can be minimized up to a constant factor approximation of the exact solution. The following problems are in this class:

- **Steiner tree [1]**

Definition 2 *Steiner tree: we are given an edge-weighted graph* $G = (V, E, w)$ *and a subset* $S \subset V$ *of required vertices. A Steiner tree is a tree in* G *that spans all vertices of* S*. In the cost minimization version of the problem, we're looking for a Steiner tree with minimum total length of edges.*

- **Steiner forest [12]**

Definition 3 *Steiner forest: The same as Steiner tree, except that not all the vertices in* S *need to be connected. We're given a set of pairs* (s_i, t_i) *that need to be connected via edges in the target set.*

- **TSP [2]**

Definition 4 *TSP: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?*

- **Vertex cover [3]**

Definition 5 *Vertex cover: A vertex cover (sometimes node cover) of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex of the set. In this problem we are looking for finding a minimum vertex cover of the graph.*

- **Facility location [4]**

Definition 6 *Facility location: We are given an undirected graph* G*. We want to open some facilities in some cities such that the total sum of the costs for opening facilities plus the transportation cost of all cities is minimized. Opening each facility costs us* k *dollars and transportation cost of each city is equal to the minimum distance of a vertex containing a facility to that vertex.*

2.2.2 $\theta(\log^*(n))$ approximation

Iterated logarithm of n, written $\log^* n$ (usually read "log star"), is the number of times the logarithm function must be iteratively applied before the result is less than or equal to 1. The simplest formal definition is the result of this recursive function:

$$\log^* n := \begin{cases} 0 & \text{if } n \leq 1; \\ 1 + \log^*(\log n) & \text{if } n > 1 \end{cases} \tag{1}$$

This class contains all the problems that can be approximated within a $\log^*$ factor of the optimal solution. The asymmetric K-center problem belongs to this class:

- **Asymmetric K-center [5]**

  Definition 7 *Asymmetric K-center: Given a complete directed graph $G = (V, E)$ with distances $d(v_i, v_j) \in R$ satisfying the triangle inequality, find a subset $S \subset V$ with $|S| = k$ while minimizing*

$$\max_{v \in V} \min_{s \in S} d(v, s).$$

2.2.3 $\theta(\log(n))$ approximation

In this class we have problems that can be approximated with a $\theta(\log(n))$ factor approximation algorithm. The following problems are in this class.

- **Set cover (optimizing set) [6]**

  Definition 8 *Set cover (optimizing set): Given a collection of subsets of a universe, find the minimum number of subsets that cover all the all the universe.*

- **Node weighted Steiner tree [13]**

  Definition 9 *Node weighted Steiner tree: This problem is the same as edge weighted Steiner tree except we have weights on the nodes.*

2.2.4 $\theta(\log^2(n))$ approximation

This class contains problems that can be approximated within a $\theta(\log^2(n))$ approximation factor. Some examples are listed below.

- **Approximate group Steiner tree [16]**

  Definition 10 *Approximate group Steiner tree: the same as Steiner tree except we are given several groups of terminals (possible intersecting) and we want to connect one terminal from each group. (The problem is hard unless $NP \subset ZTIME(n^{poly})$.*

2.2.5 $O(n^\epsilon)$ for any constant ϵ, $\Omega(\log^2 n)$

Example:

- **Directed Steiner tree [17]**

  Definition 11 *Directed Steiner tree: The same as Steiner tree except the graph is directed.*

2.2.6 $O(n^{\frac{1}{3}})$, $\Omega(n^{1-\epsilon})$

Example:

- **Coloring [7]**

  Definition 12 *Given an undirected graph G, find the minimum number of colors for vertices such that every two adjacent vertices have different colors.*

2.3 Approximability of maximization problems

In this section we talk about the approximability of maximization problems. You can see that for some problems the levels of approximability for the maximization and minimization versions are different.

2.3.1 $\Theta(1)$-approximation

Example:

- **Max coverage [8]**

  Definition 13 *Max coverage Given several sets of elements and a number k, choose at most k sets such that the maximum number of elements are covered; i.e. the union of the selected sets has maximum size.*

- **Max cut [9]**

  Definition 14 **Max cut:** *Partition the set of nodes of the graph into two sets A and B such that the number of edges between A, B is maximized*

2.3.2 $\Theta(\log(n))$-approximation

Examples:

- **Unique coverage [10]**

  Definition 15 *Unique coverage: Given an input the same as Set Cover problem, find a sub-collection to maximize the number of elements uniquely covered; i.e. the elements that appear in exactly one chosen set. [Novel hardness result; only $O(\log^{\epsilon} - \text{hardness})$].*

- **Domatic number [11]**

  Definition 16 *Domatic number: Find maximum k such that the graph $G = (V, E)$ has a partition of V into disjoint sets $V_1, V_2, \ldots, V_k$ such that each V_i is a dominating set for G.*

2.3.3 $O(n^{\frac{1}{3}}), \Omega(2^{\log^{1-\epsilon}(n)})$-approximation

Example:

- **Label cover (Max-rep) [14]**

  Definition 17 *Label cover (Max-rep): Given a bipartite graph* $G = (A, B, E)$*, where* $|A| = |B| = N$ *and equitable partitions* A, B *into* k *sets of size* $n = \frac{N}{k}$*, namely* $A_1, A_2, \ldots, A_k$ *and* $B_1, B_2, \ldots, B_k$*.*

  Goal: choose a subset $A' \subseteq A$ *and a subset* $B' \subseteq B$ *such that* $|A' \cap A_i| = 1$ *and* $|B' \cap B_i| = 1$ *and the graph induced on* $A' \cup B'$ *has maximum number of edges.*

2.3.4 $\tilde{O}(n), \Omega(n^{1-\epsilon})$-approximation

Example:

- **Independent set [15]**:

  Definition 18 *Independent set: Find a maximum size subset of vertices with no edges in between.*

2.4 Planar graphs

Note that the above hardness often happens for general graphs. For special graphs such as planar graphs (which can be drawn on the plane with no crossing) we do not have so much hardness; e.g, for coloring we have $\frac{4}{3}$-hardness and for prize-collecting Steiner forest is *APX-hard*. Lots of other problems have *PTAS* if the graph is planar. The class of APX-hard problems is defined as follows.

Definition 19 *A problem is said to be* **APX-hard** *if there is a PTAS reduction from every problem in* APX *to that problem. To say a problem is APX-hard is generally bad news, because if* $P \neq NP$*, it denies the existence of a PTAS, which is the most useful sort of approximation algorithm.*

3 Two Important Problems for Hardness of Approximation Algorithms

In this section we talk about two important problems that can be used to show hardness results for approximability of problems.

Definition 20 ***Max-rep*** *Given a bipartite graph* $G = (A, B, E)$*, where* $|A| = |B| = N$ *and equitable partitions* A, B *into* k *sets of size* $n = \frac{N}{k}$*, namely* $A_1, A_2, \ldots, A_k$ *and* $B_1, B_2, \ldots, B_k$*.*

Goal: choose a subset $A' \subseteq A$ *and a subset* $B' \subseteq B$ *such that* $|A' \cap A_i| = 1$ *and* $|B' \cap B_i| = 1$ *and the graph induced on* $A' \cup B'$ *has maximum number of edges.*

Definition 21 **Min-Rep** *Input is the same as before.*

Goal: Choose a subset A' *of* A *and a subset* B' *of* B *(maybe more than one element from each group) such that all superedges are covered while minimizing* $|A'| + |B'|$.

We say there is a super-edge (i, j) *if there is an edge in* $G(A_i, B_j)$. *We say a super-edge is covered if* $a \in A' \cap A_i, b \in B' \cap B_j$ *and* $(a, b) \in E(G)$.

Sometimes it is convenient(and possible) to restrict the Min-Rep(or Max-Rep) so that for every super-edge (A_i, B_j); i.e. $G(A_i, B_j)$, each vertex in B_j is adjacent to at most one vertex in A_i(Star property since $G(A_i, B_j)$ is a collection of vertex-disjoint stars). This has applications e.g. in proof of hardness for survivable network design.

Definition 22 *A* **Unique Games Instance** *is an instance in which* $G(A_i \cup B_j)$ *is further a matching.*

References

[1] Steiner tree problem. (2014, September 14). In Wikipedia, The Free Encyclopedia. Retrieved 18:53, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Steiner_tree_problem&oldid=625581749

[2] Travelling salesman problem. (2014, October 29). In Wikipedia, The Free Encyclopedia. Retrieved 19:02, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Travelling_salesman_problem&oldid=631582075

[3] Vertex cover. (2014, November 4). In Wikipedia, The Free Encyclopedia. Retrieved 19:03, November 10, 2014, from

[4] Facility location problem. (2014, October 3). In Wikipedia, The Free Encyclopedia. Retrieved 19:04, November 10, 2014, from

[5] Metric k-center. (2014, May 28). In Wikipedia, The Free Encyclopedia. Retrieved 19:05, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Metric_k-center&oldid=610504326

[6] Set cover problem. (2014, September 23). In Wikipedia, The Free Encyclopedia. Retrieved 19:06, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Set_cover_problem&oldid=626730768

[7] Graph coloring. (2014, November 6). In Wikipedia, The Free Encyclopedia. Retrieved 19:07, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Graph_coloring&oldid=632710837

[8] Maximum coverage problem. (2014, February 12). In Wikipedia, The Free Encyclopedia. Retrieved 19:08, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Maximum_coverage_problem&oldid=595175424

[9] Maximum cut. (2014, May 23). In Wikipedia, The Free Encyclopedia. Retrieved 19:09, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Maximum_cut&oldid=609806921

[10] Demaine, Erik D., et al. "Combination can be hard: Approximability of the unique coverage problem." SIAM Journal on Computing 38.4 (2008): 1464-1483.

[11] Domatic number. (2013, March 14). In Wikipedia, The Free Encyclopedia. Retrieved 19:11, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Domatic_number&oldid=543992212

[12] Feldman, Moran, Guy Kortsarz, and Zeev Nutov. "Improved approximating algorithms for directed Steiner forest." Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2009.

[13] Klein, Philip, and R. Ravi. "A nearly best-possible approximation algorithm for node-weighted Steiner trees." Journal of Algorithms 19.1 (1995): 104-115.

[14] Dinur, Irit, and Shmuel Safra. "On the hardness of approximating label-cover." Information Processing Letters 89.5 (2004): 247-254.

[15] Independent set (graph theory). (2014, November 5). In Wikipedia, The Free Encyclopedia. Retrieved 19:15, November 10, 2014, from http://en.wikipedia.org/w/index.php?title=Independent_set_(graph_theory)&oldid=632609529

[16] Garg, Naveen, Goran Konjevod, and R. Ravi. "A polylogarithmic approximation algorithm for the group Steiner tree problem." Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 1998.

[17] Charikar, Moses, et al. "Approximation algorithms for directed Steiner problems." Journal of Algorithms 33.1 (1999): 73-91.