Assignment 8: Approximation Algorithms

Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

January 6, 2015

Notice:

1. - **Due** Jan. 16, 2015 for graduate students at UCAS;
   - **Due** Feb. 3, 2015 for Ph.D. Candidates at ICT;
2. Please submit your answers in hard copy.
3. Please choose at least two problems from Problem 1-4, and choose at least one problem from Problem 5-6.
4. When you are asked to give an approximation algorithm, you should do at least the following things:
   - Describe the basic idea of your algorithm in natural language **AND** pseudo-code;
   - Prove the correctness of your algorithm;
   - Analyze the complexity and the guarantee of your algorithm.
5. Contact TA: Chao Wang (janiccordan@gmail.com).

1 Bin Packing

Given a bin S of size V and a list of n items with sizes $a_1, \ldots, a_n$ to pack, find an integer number of bins B and a B-partition $S_1 \cup \ldots \cup S_B$ of set $\{1, \ldots, n\}$ such that

$$\sum_{i \in S_k} a_i \leqslant V$$

for all $k = 1, \ldots, B$. A solution is optimal if it has minimal B. Give a 2-approximation algorithm for this problem. And prove the following claim: For any $\epsilon > 0$, there is no approximation algorithm having a guarantee of $3/2 - \epsilon$ for the bin packing problem, assuming $P \neq NP$.

2 Vertex Cover

Consider the following algorithm for unweighted : In each connected component of the input graph execute a depth first search (DFS). Output the nodes that are not the leaves of the DFS tree. Show that the output is indeed a vertex cover, and that it approximates the minimum vertex cover within a factor of 2.

3 3D Matching

Consider the following maximization version of the 3-Dimensional Matching Problem. Given disjoint sets X, Y, Z, and given a set $T \subseteq X \times Y \times Z$ of ordered triples, a subset $M \subseteq T$ is a 3-dimensional matching if each element of $X \cup Y \cup Z$ is contained in at most one of these triples. Assume that $|X| = |Y| = |Z|$. The Maximum 3-Dimensional Matching Problem is to find a 3-dimensional matching M of maximum size. Give a polynomial-time algorithm that finds a 3-dimensional matching of size at least 1/3 times the maximum possible size.

4 Hitting Set

Consider an optimization version of the Hitting Set Problem defined as follows. We are given a set $A = a_1, a_2, \ldots, a_n$ and a collection $B_1, B_2, \ldots, B_m$ which are subsets of A. Also, each element $a_i \in A$ has a weight $w_i \geqslant 0$. The problem is to find a hitting set $H \subseteq A$ such that the total weight of the elements in H, that is,

$$\sum_{a_i \in H} w_i$$

is as small as possible. Notice that H is a hitting set if $H \cap B_i$ is not empty for each i. Let $b = max_i|B_i|$ denote the maximum size of any of the sets $B_1, B_2, \ldots, B_m$. Give a polynomial-time approximation algorithm for this problem that finds a hitting set whose total weight is at most b times the minimum possible.

5 Programming 1: LP+Rounding

The MAX-3SAT is a problem in the computational complexity subfield of computer science. It generalizes the Boolean satisfiability problem which is a decision problem considered in complexity theory. It is defined as: Given a 3 conjunctive normal formula ϕ (i.e. with at most 3 variables per clause), find an assignment that satisfies the largest number of clauses.

Formulate this problem under ILP and then use the LP-based rounding technique to derive an approximation algorithm for it. Achieve it in any programming language.

6 Programming 2: PTAS

Recall the PTAS algorithm of KNAPSACK problem of Lecture 11. Achieve it in any programming language.