Assignment 1: CS60023: Approximation and online algorithms

Spring 2024
You must answer all questions totalling 100 marks.
All proofs and arguments must be complete and clear.
Follow lucid writing style, using suitable notation and maintaining rigour.
Submission in class on January 22, 2024 at 5 pm in neatly stapled sheets by handwriting.

January 14, 2024

(1) Show that we can design a polynomial time f-factor algorithm for the *weighted set covering* problem if each element is covered in at most f sets. To show this you may use the primal rounding technique, that is, when the optimal solution has a value at least $\frac{1}{f}$ for an indicator variable, we choose the corresponding indicated set in the set cover. You must first clearly state and explain the formulations of the primal and dual LPs. Secondly, you must show that the collection of sets picked up is indeed a set cover. Finally, you have to argue why the set cover computed is an f-factor approximation. (6+4+5 marks)

(2) Consider the local search (offline) approximation algorithm for scheduling n jobs on m identical machines, where p_i is the time required for the job i, $1 \leq i \leq n$, we start with some schedule for all the n jobs and then undertake *local actions* in several steps to alter the schedule till the termination of the algorithm. The local action is that of transfering the latest ending job (say b) to the machine that has processed least. How many times can such a local step be repeated until termination? [Note that with each local step of the transfer of a job from one machine to another, the total processing time is improved. Argue whether the same job can be transfered repeatedly from machine to machine.]

When no more transfer of any job is possible, argue that all machines must have been busy at the starting time of the last ending job.

Develop lower bounds for the total optimal time OPT for processing the jobs.

Estimate an upper bound on the total time required to complete all jobs in terms of OPT. (3+3+4+5 marks)

(3) In the maximum coverage problem, we pick k sets from a collection of l given subsets $S_1, S_2, S_3, ..., S_l$, of a universal set U, where $|U| = n$, so that the k sets picked cover the maximum number of elements from U.

Consider the greedy method where every next set picked by the algorithm covers the maximum possible number of yet uncovered elements from U. If OPT is the maximum coverage possible by some k sets of the $l \geq k$ given subsets of U, then show that the greedy method covers at least $(1-(1-\frac{1}{k})^k) > (1-\frac{1}{e})OPT$ elements. [Exercise 2.15 from Vazirani's text: Problem 2.18] (6+9 marks)

(4) The (unweighted) vertex covering problem may be viewed as a set covering problem

where we need to cover all the edges incident at each of the vertices. So, for each vertex $v \in V$ of the given unweighted undirected graph $G(V, E)$, we need to form its set $E(v)$ of its incident edges from the set E. Show that using the greedy cardinality set cover heuristic we can compute a vertex cover for $G(V, E)$ which has at most $O(\log |V|) \times OPT$ vertices, where OPT is the size of the minimum cardinality vertex cover for $G(V, E)$. (10 marks)

(5) Exercise 2.3 (Problem 2.14) [1] (15 marks)

(6) Exercise 2.4 (Problem 2.15) [1] (10 marks)

(7) Exercise 2.5 (N. Vishnoi) [1]. (20 marks)

References

[1] V. Vazirani, Approximation algorithms, Springer, 2003.