Mid-semester Examination: CS60023: Approximation and online algorithms

LTP 3-0-0: Credits 3: Time 2 hours: Autumn 2018
Maximum marks one can score is 100.
All proofs and arguments must be complete and clear.
Follow lucid writing style, using suitable notation and maintaining rigour.
Questions 1, 3, 4 and 6 are compulsory. Out of the remaining questions attempt any three.

September 17, 2018, 2-4 pm

(1) In the *weighted majority algorithm* for *predicting from expert advice*, predictions are made based on the sums of weights of experts deciding different predictions, and then weights of the erring experts are halved. Why are probabilities used instead, in the "randomized" method? Unlike the previous method of weighted majority, how are probabilities used in the "randomized" method, to derive an upper bound on the "expected" number of falures? How is the penalty parameter $\beta < 1$ relevant? Outline the derivation of the upper bound on the ratio $\frac{M}{m}$, where M is the expected number of mistakes committed by the algorithm and m is the number of mistakes committed by the best expert.

{Check the research paper for details as uploaded in moodle site.}

(3+5+2 marks)

(2) State the *k-centre problem* precisely and the greedy approximation algorithm for the k-centre problem. Assume that the triangle inequality applies throughout, on the pairwise distances between vertices in the complete positive weighted underlying graph of pairwise distance costs. Show that the k-centre greedy approximation algorithm achieves an approximation ratio bound of two.

{There are two cases to consider when we fix our attention on an optimal solution; we perceive the solution computed by the greedy algorithm with respect to the optimal solution and apply the triangle inequality to demonstrate the 2-approximation bound.}

(4+4 marks)

Given an integer k and an undirected graph $G(V, E)$, the problem of deciding whether G has a *dominating set* of size at most k is known to be NP-complete. A dominating set $S \subseteq V$ is such that every vertex $v \in V$ is either in S or is adjacent to a vertex in S. Show that it is possible to construct an instance of the k-centre problem for any given instance $< G(V, E), k >$ of the dominating set problem, so that G has a dominating set of size at most k if and only if the created k-centre instance has minimum radius of unit length for its optimal solution. Further show that it is possible to tune the construction of the k-centre instance from an arbitrary dominating set instance in such a way that the existence of a polynomial time approximation algorithm with approximation ratio strictly less than two for the k-center problem would imply $P = NP$.

{See the solution for this hardness result in the text by Willaimson et al. in Theorem 2.4}

(2+5 marks)

(3) 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; you must use the primal-dual method. You must clearly state and explain the formulations of the primal and dual LPs, the relaxed complementary slackness conditions, and argue the correctness of the method for achieveing the approximation ratio.

{See the handouts, slides and texts.}

(7+8 marks)

(4) (i) Suppose a deterministic online paging algorithm is c-competitive. Show that $c \geq k$, where k is the size of the fast memory.

(7 marks)

{The deterministic online algorithm can be forced to have a page fault on every page request whereas the offline algorithm can keep track of the next $k-1$ requests on a page fault, and evict a page that is not requested in the next $k-1$ requests.}

(ii) In the amortized analysis (where the potential function is the sum of "weights" in the range $[1, k]$ for elements the set $S = S_{LRU} \setminus S_{OPT}$), we know that the potential function value increases by at most k when OPT has a page fault. Here OPT is an optimal deterministic paging algorithm. S_{LRU} and S_{OPT} are respectively the fast memory pages for the LRU online algorithm and the optimal offine algorithm. Here k is the number of pages in the fast memory of either algorithm.

Show how the potential function value changes on a page request where OPT does not have a page fault whereas LRU has a page fault.

{See the handout.}

(8 marks)

(5) In the deterministic online Move-To-Front method (MFT), an element x is accessed and moved to the front of the list at time t for processing item $\sigma(t)$, from a stream of items σ. We compare the performance of MFT with respect to the optimal deterministic method OPT for each item $\sigma(t)$. Suppose k is the number of items preceeding x in both the lists of MFT as well as of OPT, and l is the number of items preceeding x in MFT's list but following x in OPT's list.

(i) Show that exactly l inversions are destroyed while serving $\sigma(t)$.

(ii) Show that at most k inversions are created while serving $\sigma(t)$.
(iii) Show that the actual cost of MFT is $k + l + 1$ for serving $\sigma(t)$.
(iv) Show that the actual cost of OPT is at least $k + 1$.

{See the original research paper uploaded in moodle.}

(3+3+5+4 marks)

(6) Let $G(V, E)$ be a complete edge-weighted undirected graph where each edge weight is positive. Furthermore, the triangle inequality is obeyed by the weights. Let $V = R \cup S$ where $R \cap S = \phi$. Let T be a tree that certainly has all the vertices of R, but may or may not have one or more vertices from S. The weight of T is the sum of weights of its edges. In polynomial time we wish to compute such as tree T of as small weight as possible. Design such a constant factor approximation algorithm where the weight of the computed tree T is at most c times the cost of such a minimum weighted tree. Here $c > 1$ is a positive constant.

{See Theorem 3.3 from Vazirani's text.}

(15 marks)

(7) Define the conditions satisfied by loads assigned to machines by jobs on *related* machines. Show that even if such conditions do not hold and the machines are *unrelated* then the competitive ratio for scheduling jobs is no worse than m, where m is the number of unrelated machines.

{See the original research paper uploaded in moodle.}

(5+10 marks)

(8) Given a weighted undirected simple graph $G(V, E)$, with positive vertex weights $w(v)$ for vertex variables v in the vertex set V, and edge variables e in the edge set E, show that $|C| \leq 2 \times OPT$. Here OPT is the size of the optimal weighted vertex cover for G and C is the output of the following algorithm.

1. Step 1:
$C = \phi$
For each $v \in V$, $t(v) = w(v)$
For each $e \in E$, $c(e) = 0$
(Here $c(e)$ and $c(u, v)$, both indicate a non-negative weight for edge $e = \{u, v\}$.)

Step 2. While C is not a vertex cover do:
{Pick an uncovered edge, say (u, v). Let $m = min(t(u), t(v))$
$t(u) = t(u) - m$; $t(v) = t(v) - m$; $c(u, v) = m$}
Include in C all vertices having $t(v) = 0$
Step 3. Output C

{See the handout and the interpretation of the method in the primal-dual approach.}

(15 marks)

(9) (i) Given a collection $\mathcal{S}$ of m subsets of U, where U has n elements, we need to pick as few sets from $\mathcal{S}$ as possible in polynomial time so that all elements of U are covered. It is known that all m sets from $\mathcal{S}$ together cover all elements of U. Let $\mathcal{C}^*$ be one collection of sets from $\mathcal{S}$ covering all elements of U, where $|\mathcal{C}^*|$ is minimised over the sizes $|\mathcal{C}|$ of all possible such covers $\mathcal{C}$ of U. Suppose we compute $\mathcal{C}$ using the greedy method of always selecting the next set as the one which covers the maximum number of so far uncovered elements. Then show that $|\mathcal{C}| \leq O(\log n) \times |\mathcal{C}^*|$.

{Use the same arguments as we use for the weighted version but use unit weights.}

Use the above covering algorithm for computing a small *hitting* set in a hypergraph $G(V, E)$ where E has subsets of V as hyperedges. What approximation ratio do you achieve?

{Direct reduction may be used to the set cover problem.}

(10+5 marks)

(10) For the complement G' of a connected chordal graph G, show that $\chi_{\mathcal{FF}}(\mathcal{G}')$ is bounded by a constant factor times $\omega(G')$, where $\omega(G')$ is the size of the maximum clique of G'.

{See the handout.}

(15 marks)