Mid-semester Examination: CS60023: Approximation and online algorithms

LTP 3-0-0: Credits 3: Time 2 hours: Autumn 2017
You must answer questions totalling at least 100 marks.
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.

September 22, 2017, 2-4 pm

(1) Show that the minimum cost of a perfect matching on the set V' of odd degree vertices in an MST of $G(V, E)$ is at most the half of OPT where OPT is the minimum cost of a metric TSP tour of the complete graph $G(V, E)$. Here the positive edge weights in $G(V, E)$ obey the triangle inequality.

(15 marks)

(2) Online scheduling on m identical machines for a sequence of n jobs can be done by assigning the next arriving job to the least (so far) loaded machine. When the last job is being assigned, show that the makespan is at most the sum of (i) the average of the lengths of all the n jobs, and (ii) $1 - \frac{1}{m}$ times the size of the longest job. Using suitable lower bounds now show that the above online algorithm is $(2 - \frac{1}{m})$-competitive.

(8+7 marks)

[Note that we are dealing with an online algorithm. The parts (i) and (ii) are about upper bounds. Lower bounds carry 7 marks. The entire rectangle with area $m \times LIST(I)$ includes the total time to process all the n jobs as well as the idle times on the $m - 1$ machines that do not process the last job of time requirement p_k. So, we can understand part (i); for part (ii), note that the idle times on $m - 1$ machines cannot be more than the idle time on the machine which processes the last job because the last job was assigned to the least loaded machine. So, the idle times add up to at most $(m-1) \times p_k \leq (m-1) \times p_{max}$.

The lower bounds are two separate items, the average processing time for n jobs and then time span of the longest job. From the above, the desired competitive ratio is immediate.]

(3) In the method *weighted majority algorithm* for *predicting from expert advice* the algorithm can commit mistakes in predictions and the weights of the erring experts are halved. If the algorithm makes M mistakes then show that the total weight W is at most $n(\frac{3}{4})^M$. If m is the number of mistakes committed by the best expert then show that M is at most $O(m + \log n)$, where n is the number of experts.

(8+7 marks)

[The total weight of all the experts is n to begin with. The penalty of weight halving is awarded to each expert making a wrong prediction. Since the sum of weights of erring experts exceeds the sum of weights of correct experts, the total weight reduction due to penalties is at least half of the half pf the total weight at that instance. So, the total weight goes down by at least a fraction of one quarter. The final weight is thus at most a $\frac{3}{4}$ fraction of the current weight. This compounded M times gives the upper bound on the weight.

The lower bound is obvious because the total weight must be at least the weight of the best expert, whose weight even after m mistakes is not too small!]

(4) (i) How does the algorithm discussed in class, for computing a spanning tree of low maximum vertex degree (for an undirected connected graph $G(V, E)$), try to reduce the maximum vertex degree of the current spanning tree using local search? State the details of this local action clearly stating the termination condition for the algorithm.

In the amortized analysis of time complexity for this algorithm, state the potential function used and its maximum and minimum values for the n-vertex graph $G(V, E)$. In the iterative step if the algorithm successfully identifies a vertex u in the current spanning tree T for reduction in its vertex degree, and is able to add an edge (v, w) to T to get the next spanning tree T', then show that $\Phi(T') \leq (1 - \frac{2}{27n^3})\Phi(T)$, where Φ is the potential function.

(7+8 marks)

(5) State the K-centre problem precisely and the greedy approximation algorithm discussed in class for the K-centre problem. Show that this algorithm achieves the factor two approximation bound.

(7+8 marks)

[The two cases must be argued fully; in one case each center computed by the approximation algorithm lies in a distinct cluster of the optimal solution. The other case is the complementary case.]

(6) In the multiway cut problem, state the main lower bounding arguments linking an optimal solution with the properties of the computed solution. Show that for multiway cut, an approximation ratio of two is possible.

(8+7 marks)

[This is an easier problem where k vertices to be separated are spelt out in the input.]

(7) 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 may use either the primal rounding technique or the dual rounding technique. You must clearly state and explain the formulations of the primal and dual LPs.

(7+8 marks)

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

(ii) Using amortized analysis (where the potential function is the sum of weights in the range $[1, k]$ for the set $S = S_{LRU} \setminus S_{OPT}$), show that when OPT has a page fault then the potential increases by at most k. 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.

(7+8 marks)

[For part (i), note that using $k + 1$ pages, the k-page memory of the online paging algorithm can be made to be faulted on each of the k page requests. Also, OPT cleverly reacts to any fault, using its resources of lookahead, determining a page to be evicted, which is not requested in the future $k - 1$ requests.

For part (ii), if OPT is faulted for a page p and has to evict a page r, where r is also in the fast memory of LRU then r appears in the set $S = S_{LRU} \setminus S_{OPT}$.]

(9) 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 $\sigma(t)$. When x is moved to the fromt of the list of MFT the cost $C_{MFT}(t)$, at time t, plus the change $\Phi(t) - \Phi(t-1)$ in the *potential* function Φ must be suitably upper bounded. We must also establish a lower bound for $C_{OPT}(t)$. Show how these upper and lower bounds are established using the notion of inversions with a suitable potential function to show that $C_{MFT}(t) + \Phi(t) - \Phi(t-1) \leq 2.C_{OPT}(t) - 1$.

(3+3+5+4 marks)