problem sheet 3

Sunday, 17 August 2025 2:40 PM

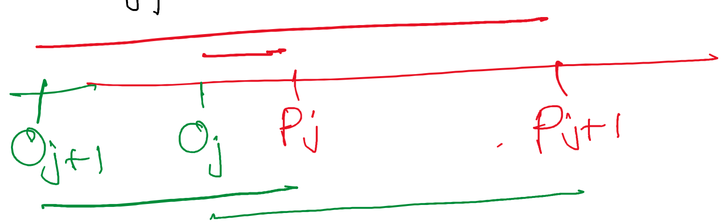
Q1) Renumber the skiers & ski so that $p_1 \le p_2 \le p_3 \cdots \le p_n$ & $s_1 \le s_2 \le s_3 \le \cdots s_n$

a) Let $p_1 = 1, s_1 = 3, p_2 = 4, s_2 = 6$. Picking pair which has minimum difference in heights yields the solution $(s_1, p_2), (s_2, p_1)$ which has value $1 + 5 = 6$. Optimum soln is $(p_1, s_1), (p_2, s_2)$ which has value 4.

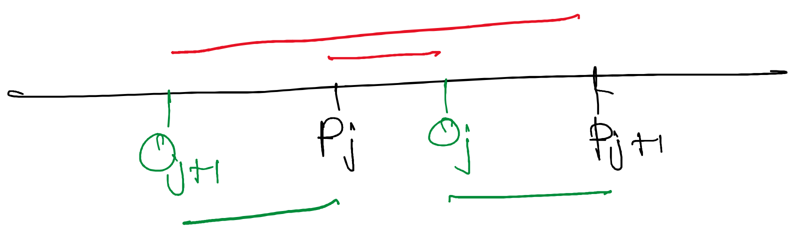
b) Let g_i (resp o_i) be the ski assigned to the i^{th} skier in the greedy (resp optimum) solution. By our numbering it follows that $g_i = i$. Let j be such that $o_j > o_{j+1}$.
Skiers $j, j+1$ contribute $|p_j - s_{o_j}| + |p_{j+1} - s_{o_{j+1}}|$ to the optimum value. Switching the assignment of these skiers contributes

$$|p_j - s_{o_{j+1}}| + |p_{j+1} - s_{o_j}|$$

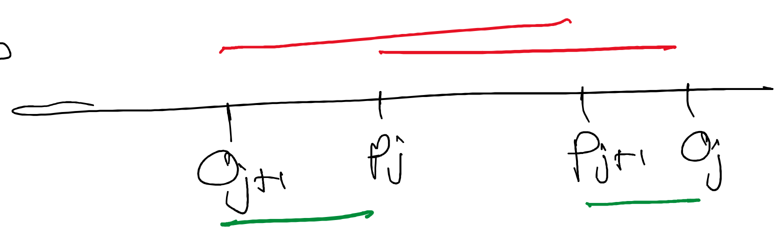
Case 1 identical



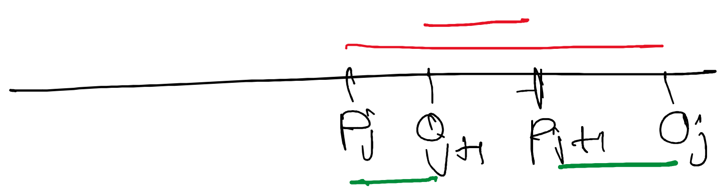
Case 2: switching reduces value of OPT



Case 3: switching reduces value of OPT

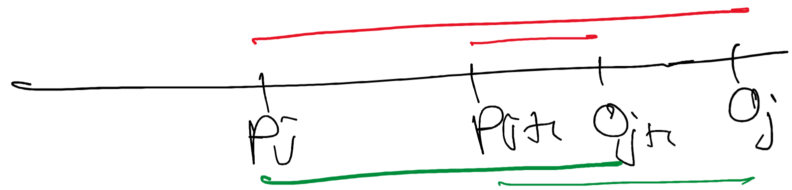


Case 4: switching reduces value of OPT



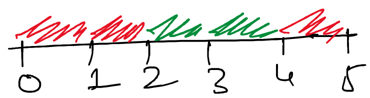
Case 4: identical

Case 4: identical



in all cases, the value of the optimum solution either reduce or remains unchanged. Hence we can make that switch & get an optimum solution which is lexicographically smaller (if we started with the assumption that the optimum solution we consider is the lexicographically minimum we obtain a contradiction)

Q2 a) SJF is not optimum. Let (r_i, x_i) be $(0,3), (2,2)$, the schedule is



& $C_1 = 5$, $C_2 = 4$

OPT schedule is

$C_1 = 3$, $C_2 = 5$

b) SRPT is optimum.
Consider an optimum schedule & suppose t is the first time at which it does not follow the SRPT rule. Let OPT schedule job j_1 at time t & suppose j_2 was the job with the smallest remaining processing time at time t. In the OPT schedule in the slots after t occupied by j_1, j_2, we first schedule j_2 & then j_1. Doing this reduces the sum of completion time of j_1, j_2 while those of other jobs are unchanged. This yields a contradiction

Q3

A greedy algorithm is to consider jobs in order of increasing size and to schedule them on the m machines in a round robin manner. Number jobs in order of increasing size.
Claim: The optimum solution schedules jobs on each machine in order of increasing size.
Let i_1<i_2<\cdots <i_p and j_1<j_2<\ldots <j_q be the jobs scheduled by the optimum solution on machines i,j and let i_1 < j_1.
Claim: i_k<j_k<i_{k+1}, 1\le k\le p-1
Proof: Let k be the smallest index such that j_k<i_k. If p \ge q then exchanging i_k, j_k reduces total completion time. If p<q then moving j_k to machine i before i_k reduces total completion time. A similar argument can be repeated if j_k>i_{k+1} thus proving the claim.
The above claim implies that on any two machines, the optimum schedules the jobs on these machines in a round-robin manner. Since this is true for all pairs of machines, the optimum must be following a

round-robin procedure on all m machines.

Q4: Consider the solutions obtained by greedy, G, and the optimum, O, and let i be the first row they differ in. Let j,k be such that G[I,j]=1, O[I,j]=0 and G[I,k]=0, O[I,k]=1. Since, at the ith step, column j needs more 1's than column k, there must be a row I' >I such that O[i',j]=1 and O[i',k]=0. We modify the optimum solution by switching the 0,1 in rows I,i' at columns j,k. This exchange makes the optimum solution "closer" to greedy and we can continue doing this till the solutions are identical.

Q5(a) In each unit-interval the machine is on, schedule the job with the earliest deadline. Suppose the jobs are numbered by increasing deadline and let $o_1,o_2,\ldots, o_n$ be the order in which the optimum schedules jobs. Let i be the least index such that $o_i>o_{i+1}$ and suppose these jobs are scheduled in the unit intervals beginning at times t_i, t_{i+1}. Since these jobs have their deadlines after t_{i+1} swapping these jobs gives a feasible schedule which is "closer" to the greedy schedule.

(b) A greedy algorithm for this problem would delay turning on the machine as much as possible while ensuring feasibility. The latest time that the machine can be turned on is $s_1=\min_j (d_j-j)$. Thus the machine remains on in the interval $[s_1,s_1+L]$ and we schedule jobs which have been released in the order of increasing deadlines.

Let $s_1 < s_2 < \cdots < s_k$ be the times at which the machine is switched on in our solution and $o_1 < o_2 <\cdots < o_p$ be the times at which the machine is switched on in an optimal lexicographic maximum solution. We argue these sequences are identical and let i be the first index at which they differ.

If $o_i > s_i$ then the optimal solution would not be feasible since the greedy algorithm picked the furthest start time to ensure feasibility.
If $o_i < s_i$ then we can get a lexicographic larger optimum solution by replacing o_i by s_i.

In both cases we have a contradiction which implies p=k.

Q6

A) At any time t, at any node v, among the packets waiting at this node forward the one whose destination is the furthest.

B) All packets have node 1 as their source and node 2 as their destination. At each time we forward that packet on the edge between nodes 1 and 2 that was released the earliest.

C) At any time t, at any node v, among the packets waiting at this node forward the one whose destination is the nearest.

Q7
If good G is assigned to the husband then we replace it with a +1 in the husband's list and with a -1 in the wife's list. The partition is fair if every prefix sum of the husband's and wife's list is non-negative.
We go through both lists simultaneously. Suppose we are at the good i on the husband's list and the good j on the wife's list. If these goods are identical then assign it to the person for whom this good is critical (not assigning it to them will create a prefix with sum -1); if the good is critical for both then there is no solution and if it is not critical for either then assign it to

either partner. If the goods under consideration are different then assign the goods to the husband/wife. From the description it follows that of the algorithms succeeds then no prefix has a negative sum.

We consider the case when the good, g, was critical for both and suppose the husband's list has a prefix of length 2a+1 that sums to 0 and the wife's list has a prefix of length 2b+1 that sums to zero. The good g is the only unassigned good in these prefixes. All numbers in the 2 lists appearing after g are -1 and since the sum of the 2 lists is 0, there are no +1 or -1 outside the 2 prefixes. The husband's list has a, +1's and a, -1's. This implies the wife's list too has a, -1's and a, +1's which implies a=b. Thus the set of goods in these prefixes are identical. If each of these 2a+1 goods had a value 1 for each partner and the remaining goods had value 0, then each partner would like to receive at least a+1 of these goods which implies no solution is possible.