1 Problem 2

1.1 Correctness

We will first show that algorithms A and B are not guaranteed to produce optimal schedules using proof by contradiction.

Lemma 1. *Ordering jobs in decreasing order of weight is not guaranteed to produce the optimal schedule.*

Proof of Lemma 1. Assume greedy algorithm B produces an optimal schedule. Consider the case where there are $n = 2$ customers with corresponding jobs taking $t_1 = 1$ and $t_2 = 10$ units of time and $w_1 = 2$ and $w_2 = 5$.

1. Greedy algorithm A favors job 1. Hence, the weighted total of the unhappiness

$$\begin{aligned}\sum_{n=1}^{2} w_i C_i &= (2)(1) + (5)(11) \\ &= 57.\end{aligned}$$

2. Greedy algorithm B favors job 2. Hence, the weighted total of the unhappiness

$$\begin{aligned}\sum_{n=1}^{2} w_i C_i &= (5)(10) + (2)(11) \\ &= 72.\end{aligned}$$

3. Greedy algorithm C favors job 1. Hence, the weighted total of the unhappiness is the same as in greedy algorithm A in this case.

We see greedy algorithm B produces a less optimal schedule compared to A and C, which contradicts our original assumption, thus B does not produce an optimal schedule.

Lemma 2. *Ordering jobs in increasing order of the time their require is not guaranteed to produce the optimal schedule.*

Proof of Lemma 2. Assume greedy algorithm A produces an optimal schedule. Consider the case where there are $n = 2$ customers with corresponding jobs taking $t_1 = 1$ and $t_2 = 5$ units of time and $w_1 = 1$ and $w_2 = 10$.

1. Greedy algorithm A favors job 1. Hence, the weighted total of the unhappiness

$$\begin{aligned}\sum_{n=1}^{2} w_i C_i &= (1)(1) + (10)(6)\\ &= 61.\end{aligned}$$

2. Greedy algorithm C favors job 2. Hence, the weighted total of the unhappiness

$$\begin{aligned}\sum_{n=1}^{2} w_i C_i &= (10)(5) + (1)(6)\\ &= 56.\end{aligned}$$

3. Greedy algorithm B favors job 2. Hence, the weighted total of the unhappiness is the same as in greedy algorithm C in this case.

We see greedy algorithm A produces a less optimal schedule compared to B and C, which contradicts our original assumption, thus A does not produce an optimal schedule.

We will now prove that greedy algorithm C is guaranteed to produce optimal schedules using an exchange argument.

Lemma 3. *Ordering jobs in decreasing order of the w_i/t_i ratios is guaranteed to produce an optimal schedule.*

We describe schedule O as: assume that each job r is scheduled with the time interval C_r with unhappiness $w_r C_r$. Let U denote the weighted total of the unhappiness of schedule O.

Proof of Lemma 3. First, let us assume the lemma and prove that the greedy ordering produces an optimal schedule. Among all optimal schedules, choose schedule O which contains a pair of adjacent jobs (i, j) in which job i is scheduled before j, and j is scheduled before i in the greedy solution. For this pair, $w_j/t_j \geq w_i/t_i$ by the greedy algorithm. We will decrease the number of inversions in O by swapping the requests i and j in the schedule O. We will show the new swapped schedule has total weighted unhappiness no larger than that of O.

Note, that the pair (i, j) formed an inversion in O, this inversion is eliminated by the swap, and no new inversions are created. Hence, the contribution of all jobs besides (i, j) to weighted total of the unhappiness remains unchanged. Let finishing time before job i be C. Let u_{ij} denote the contribution of (i, j) to unhappiness before the swap and u_{ji} denote the pair's contribution after the swap.

$$\begin{aligned}u_{ij} &= w_i(C + t_i) + w_j(C + t_i + t_j)\\ &= w_i(C + t_i) + w_j(C + t_j) + w_j t_i\\ u_{ji} &= w_j(C + t_j) + w_i(C + t_j + t_i)\\ &= w_i(C + t_i) + w_j(C + t_j) + w_i t_j.\end{aligned}$$

By the greedy rule

$$\begin{aligned} w_j/t_j &> w_i/t_i \\ w_j t_i &> w_i t_j, \quad \text{assuming} \quad w_i, w_j \geq 0. \end{aligned}$$

Hence, $u_{ij} > u_{ji}$, which shows that the swap does not increase the weighted total of the unhappiness of the schedule.

Thus, we can iteratively swap out of order pairs of an alternative optimal schedule starting at the initial difference without increasing the the weighted total of the unhappiness of the schedule. After all pairs are swapped we will reach the greed schedule. Hence, the schedule obtained by the greedy algorithm is optimal.