Table of Contents

Question 1:

(a)
(b)
(c)
(d)
(e)

Question 2:

(a)
(b)
(c)
(d)

Question 1.

(a)
The minimum number of trucks required by the optimal solution is $\lceil S \rceil$. Any less than $\lceil S \rceil$ and we don't have enough space for the boxes.

(b)
There must be at most 1 truck less than half-full.

Proof by contradiction:
Asusume to the contrary that there are more than 1 truck that is less than half-full.
Then because the algorithm says that we take each box and place it into the first truck that accommodates it, we would have taken the boxes from one of the half-full trucks and placed them in the first truck that can accommodate them, therefore the truck would be empty because since there are both less than half-full, we can fit all of the 2nd truck items into the first truck. So any two half-full trucks would be combined into one more than half-full truck, so you can only have one truck that is less than half-full.

(c)
There is at most n trucks used if we put every box into the first truck that that can accommodate it. This is because in the worst case scenario, where every box is more than half of a truck size, we are only able to fit one box into every truck, which uses up n trucks.

(d)
Proof of $\alpha = 2$:
Let n = approximate solution, m = optimal solution
By part a), $\text{m} \geq \lceil S \rceil \geq \sum_{i=1}^{n} s_i$
By part b), at least $n-1$ trucks are more than half-full, so $\sum_{i=1}^{n} s_i \geq \frac{n-1}{2}$
$\therefore \text{m} \geq \frac{n-1}{2} \implies 2m \geq n-1$
Since the greedy solution is bounded by $2\times$ the optimal solution, then $\alpha = 2$

(e)

Algorithm

Assign_Boxes (`boxes`: array of doubles $\in [0, 1]$)

```
1: trucks = []
2:
3: for item in boxes do
4:     loaded_truck = false
5:     for truck in trucks do
6:         if item + truck ≤ 1 then
7:             truck += item
8:             loaded_truck = true
9:             break
10:    if not loaded_truck then
11:        trucks.append(item)
12:
13: return trucks.length()
```

This algorithm is $O(n)$, because there is only one for loop which loops over all the items in the input array of boxes. Every other line is $O(1)$

Question 2.

(a)

Let $\texttt{sum} = \sum_{i=1}^{n} t_i$

Given $\{t_1, t_2, \cdots, t_n\}$ Suppose to the contrary that $\frac{1}{2}T_1 \geq T_2$

This means that $T_1 \geq 2T_2$

$$\begin{aligned} \texttt{sum} &= T_1 + T_2 \\ &\geq 2T_2 + T_2 \\ &= 3T_2 \end{aligned}$$

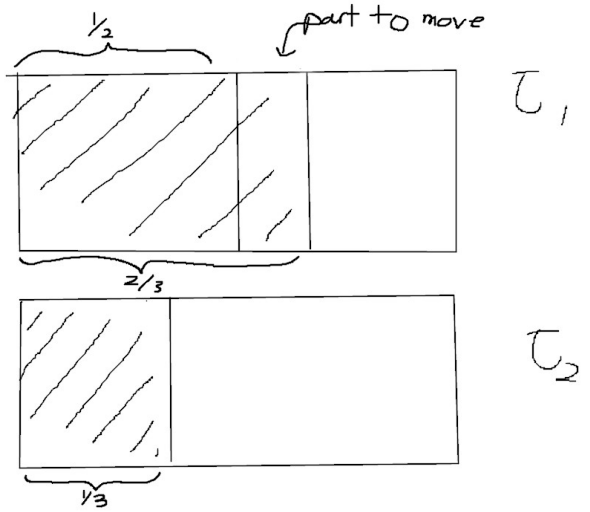
So $T_2 \leq \frac{1}{3}\texttt{sum}$ Which means $T_1 \geq \frac{2}{3}\texttt{sum}$

$|T_1 - T_2| \geq \frac{1}{3}\texttt{sum}$

By the given, which is that no job dominates, T_1 can have a job that is at most $\frac{1}{2}\texttt{sum}$. Therefore there must be jobs summing up to $\frac{2}{3}\texttt{sum} - \frac{1}{2}\texttt{sum} = \frac{1}{6}\texttt{sum}$ or more.

The time of jobs moved must be $\in \left[\frac{1}{6}\texttt{sum},\ \frac{1}{3}\texttt{sum}\right)$, and this will reduce $|T_1 - T_2|$ by $2 \times \left[\frac{1}{6}\texttt{sum},\ \frac{1}{3}\texttt{sum}\right) = \left[\frac{1}{3}\texttt{sum},\ \frac{2}{3}\texttt{sum}\right)$.

This contradicts our assumption that this was a local minimum.



For the condition that $T_2 \geq 2T_1$, we can apply the same contradiction proof, which results in the same conclusion.

(b)
Without loss of generality, suppose $T_1 \geq T_2$

$\exists t_i$ in T_1 such that $\left|(T_1 - t_i) - (T_2 + t_i)\right| <= |T_1 - T_2|$ for some $i \in \{1, \cdots, n\}$

Let t_j be the largest number for the above condition.
Then any change after t_j should reduce the absolute difference by less than t_j.
If we move t_j again, that implies that we have increased T_2 so that $T_2 \geq t_j + T_1$, but this is impossible because any element moved from T_1 to T_2 after t_j must be less than t_j. And for the element to have been moved, $T_1 \geq T_2$

(c)
Let the job times be $\{5,\ 6,\ 7,\ 8\}$
We can get an arbitrary allocation as follows:
$T_1 = \{5,\ 6\}$
$T_2 = \{7,\ 8\}$

$$\sum_{j \in T_1} t_j = 11$$
$$\sum_{j \in T_2} t_j = 15$$
For this heuristic, moving any job here would increase the absolute difference, not reduce. So the algorithm would halt here after the arbitrary allocation with no movement of jobs.

But the optimal solution is:
$T_1 = \{5,\ 8\}$
$T_2 = \{6,\ 7\}$

$$\sum_{j \in T_1} t_j = 13$$
$$\sum_{j \in T_2} t_j = 13$$

(d)

Let $N = 2 \times \sum_{i=1}^{n} t_i$

Objective function:

$\min Z$

Constraints:

$$\begin{aligned} z_i \in \{0,\ 1\} \quad \forall i \in \{1,\ 2,\ \dots\ ,\ n\} \\ \sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) + N \times Y \geq Z \\ -\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) + N \times (1 - Y) \geq Z \\ X \leq Z \\ -X \leq Z \\ Y \in \{0,\ 1\} \end{aligned}$$

For each job t_i, we have z_i to indicate if it's in the first machine or the second machine. The objective function should be $\left|\sum_{i=1}^{n}(t_i \times z_i) - \big(t_i \times (1 - z_i)\big)\right|$. However, because this is not a linear function, it is instead replaced by the variable Z. Z is set to $\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big)$ or $-\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big)$ based on Y.

If $Y = 0$, we get

$$\begin{aligned} \sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) \geq Z \\ -\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) + N \geq Z \\ \sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) \leq Z \\ -\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) \leq Z \end{aligned}$$

And since $N = 2 \times \sum_{i=1}^{n} t_i$, we get $-\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) + N \geq \sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big)$

So by constraints 1 and 3, $Z = \sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big)$

If $Y = 1$, we get

$$\begin{aligned}
\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) + N &\geq Z \\
-\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) &\geq Z \\
\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) &\leq Z \\
-\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) &\leq Z
\end{aligned}$$

And since $N = 2 \times \sum_{i=1}^{n} t_i$, we get $\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big) + N \geq \sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big)$

So by constraints 2 and 4, $Z = -\sum_{i=1}^{n} \big(t_i z_i - t_i(1 - z_i)\big)$