This problem can be regarded as a SLS problem but with fixed length of each "line" (m). For the sub problem in the minutes $m_1, m_2, \cdots m_j$, if we denote wait time as t, the optimal solution will be:

$$\begin{aligned} OPT(j) &= \min_{1 \le i \le j}(t_{i,j} + OPT(i-1)) \\ &= \min_{1 \le i \le j}(mc_i + (m+1)c_{i+1} + \cdots + (m+j-i)c_j + OPT(j-1)) \end{aligned} \tag{1}$$

The algorithm is stated as follow:

Algorithm 1 Algorithm for problem 3

Set array $T[0, 1, \cdots n]$
Set $T[0] = 0$
For all pairs $i \le j$
  Compute the time cost
Endfor
For $j = 1, 2, \cdots, n$
  Use the recurrence (Equation (1)) to compute $T[j]$
Endfor
Return $T[n]$

Time complexity analysis:
In this algorithm, if we have pre-computed the time cost, the time complexity would be $O(n^2)$ since pairs (i, j) are within i to j.

An idea that may decrease the complexity to $O(mn)$: (just some thoughts)
Since we know the processing time m is fixed. We can just let i be no less than $j - m$ (i.e., let $j - m \le i \le j$ in Equation (1) (if $j - m < 0$ then 0)). Then we let j skip 0 if it meets 0. So that the scope of i will be decreased from n to m, which gives $O(mn)$.