Samuel Sicklick

Algorithm

The algorithm behind this implementation of *WaitNoMore* sorts the jobs by their ratio of duration (t_i) to weight (w_i). The intuition behind this is that lower ratio values will allow corresponding W_i values to be lowered in relation to w_i, thereby minimizing the summation $\sum_{i=i}^{n} w_i \times W_i$.

Proof

1. Let O be an optimal schedule and G be the schedule produced by the greedy algorithm.
   a. Counter Example to Sorting by highest w_i alone (t_i, w_i):
      i. $\{\{1, 1\}, \{3, 2\}\} = (2 \times 0) + (1 \times 3) = 3$, while $(1 \times 0) + (2 \times 1) = 1$
   b. Counter Example to Sorting by lowest t_i alone (t_i, w_i):
      i. $\{\{1, 2\}, \{2, 5\}\} = (2 \times 0) + (5 \times 1) = 5$, while $(5 \times 0) + (2 \times 2) = 4$
2. Theorem: $G = O$.
   a. Lemma: Let j_1 be the first job picked by G. There exists an O which also starts with j_1.
      i. Let O be some arbitrary optimum solution. If it starts with j_1 then this is confirmed.
      ii. If O does not start with j_1 then j_1 must not be the job with the lowest weight to duration ratio.
         1. Axiom: The solution must minimize the summation $\sum_{i=i}^{n} w_i \times W_i$.
         2. W_i increases as i increases, by definition of waiting time.
         3. Lower ratio jobs must be multiplied with lower values of W_i, otherwise the summation would not be minimized due to the definition of multiplication.
         4. By contradiction, j_1 must not be the job with the lowest weight to duration ratio.
      iii. By contradiction, j_1 could not have been used to start G, by definition of the greedy algorithm as explained in *Algorithm*.
   b. By the lemma, any O can be turned into O' by swapping j_i and j_k, the i^{th} and k^{th} jobs in O, with the corresponding the i^{th} and k^{th} jobs in G. (In other words: $O = \{j_1, j_3, j_2\}$, $G = \{j_1, j_2, j_3\}$ then $O' = \{j_1, j_2, j_3\}$.)
      i. Base Case: n, the number of jobs, is 1, in which case both O and G must select it.
      ii. Inductive Step: Assuming O and G have parallel schedules for the first i jobs, then the $(i + 1)^{th}$ job will be treated as the base case anew such that it is subject to the lemma.
      iii. Implication: Every job i in O, can be shifted to become O' which is equal to G.
   c. QED