

最佳算法是按照  $v_i/l_i$  的递减顺序安排作业。我们通过交换论证来证明这种算法的最优性。

因此，我们可以考虑任何其他的时间表。按照交换论证的标准，我们注意到该计划必须包含一个反转：一对工作  $i, j$  在交替方案中  $i$  排在前面，而在贪婪方案中  $j$  排在前面。在贪婪方案中排在  $i$  之前。此外，事实上必须有一对相邻的  $i, j$ 。注意，对于这对工作，我们有  $v_i/l_i < v_j/l_j$ 。如果我们能证明交换这对  $i, j$  不会增加完成时间的加权和，那么我们就可以迭代进行，直到不再有反转为止，从而得到贪心计划，而不会增加函数的最小值。这样一来，贪心算法就是最优的。

因此，请考虑将  $i$  和  $j$ 。所有其他工作的完成时间保持不变。假设在  $i$  和  $j$  之前的工作的完成时间为  $N$ ，那么在调换之前， $i$  和  $j$  对总和的贡献为  $v_i(N+1/l_i) + v_j(N+1/l_j)$ ，而在调换之后，其贡献为  $v_j(N+1/l_j) + v_i(N+1/l_i)$ 。调换后的值与调换前的值的差值是（去掉两个表达式中的共同项） $v_j/l_j - v_i/l_i$ 。由于  $v_i/l_i < v_j/l_j$ ，这个差值的上界为 0，因此完成时间的加权总和不会因为交换而增加。