Analysis of LPT algorithm:

Theorem: LPT achieves approximation ratio $\frac{4}{3} - \frac{1}{3m}$ to the minimum makespan problem.

Recall that the jobs are sorted such that $p_1 \leq p_2 \leq \ldots \leq p_n$

Claim 1 : If OPT assigns at most two jobs on every machine then LPT is optimal.

Proof: If n ≤m then a possible optimal solution puts a single job on some n machines. If m<n≤2m then an optimal schedule places job k alone for k<2m-n+1, and pair jobs k and (2m-k+1) for k>= 2m-n+1. Every schedule that does not fulfil this property can be exchanged to fulfill it without hurting the makespan. Observe that this is exactly what is done by LPT.

Proof of Thm: For an instance I, denoted the value of an optimal solution by C*(I), and the value of the makespan produced by LPT by $C_A(I)$. Assume by contradiction that I is an instance for which the statement is false. Let k be the job determining the makespan $C_A(I)$. W.l.o.g., k is the shortest job in I, as otherwise, we can remove all the shorter jobs and get a smaller instance I' such that $C_A(I') = C_A(I)$ and $C^*(I') \leq C^*(I)$, therefore, also for I' we have

$$\frac{C_A(I')}{C^*(I')} \geq \frac{C_A(I)}{C^*(I)} \geq \frac{4}{3} - \frac{1}{3m}.$$

In the analysis of List-scheduling, which is valid also here, we saw that

$$C_A(I) \leq \frac{\sum_j p_j}{m} + \frac{p_k(m-1)}{m}$$

Therefore, $\frac{4}{3} - \frac{1}{3m} < \frac{\sum_j p_j}{mC^*(I)} + \frac{p_k(m-1)}{mC^*(I)}$.

It holds that $C^*(I) \geq \frac{\sum_j p_j}{m}$, therefore, $\frac{4}{3} - \frac{1}{3m} < 1 + \frac{p_k(m-1)}{mC^*(I)}$.

Multiple by $C^*(I)$ to get: $\frac{4}{3}C^*(I) - \frac{C^*(I)}{3m} < C^*(I) + \frac{m-1}{m}p_k$.

Rearranging yields $\frac{C^*(I)}{3}\left(1 - \frac{1}{m}\right) < \left(1 - \frac{1}{m}\right)p_k$. Thus $C^*(I) < 3p_k$.

Therefore, in an optimal schedule there are at most two jobs on each machine. By Claim 1 above, LPT is optimal for such an instance, contradicting our assumption that the instance I does not fulfil the statement.