Suppose we have n jobs $J_1, \ldots, J_n$ with run times $t_1 \geq t_2 \geq \cdots \geq t_n$ and $k > 1$ machines $M_1, \ldots, M_k$. Define:

$$A = \frac{1}{k} \sum_{i=1}^{n} t_i$$

If $t_1 \geq A$, then an optimal solution to the job makespan minimization problem can be obtained by assigning J_1 to M_1 and solving the makespan minimization subproblem for the $n-1$ jobs $J_2, \ldots, J_n$ with the $k-1$ machines $M_2, \ldots, M_k$.

Proof: Suppose for contradiction that this algorithm did not give the optimal solution to the makespan minimization problem. Then, since it was given that the subproblem was optimal, it must be the case that in the optimal solution, M_1 must run another job J_j. Since $t_1 \geq A$, and M_1 runs both jobs J_1 and J_j, M_1 has a run time strictly greater than A. But, since A is the average runtime for the machines, it then follows from this that some other machine M_l must have a run time strictly less than A. If we assign J_j instead to M_l, then the makespan of M_1 and M_l strictly decreases. This gives at least an equally valid solution (if not strictly better). In this way, we can reassign any other job assigned to M_1 (other than J_1) and still have an optimal solution, which is a contradiction.