Let φ^* be an optimal allocation, and let ℓ_j^* be the load of factory j in this allocation, i.e.

$$\ell_j^* = \sum_{i:\varphi^*(i)=j} t_i$$

Let T^* be the maximum load, i.e. $T^* = \max\{\ell_j^*\}$. Consider the first iteration i in which a factory j gets a load of more than $2 \cdot T^*$. First, it is clear $t_i \leq T^*$, as the factory to which φ^* assigns product i has load $t_i \leq \ell_j^* \leq T^*$. Let ℓ_j be the load of factory j just before iteration i. Then $t_i + \ell_j \geq 2 \cdot T^*$, and thus $\ell_j \geq T^*$. Since the algorithm assigns product i to the least-loaded factory, it follows that for every $1 \leq k \leq M$, $\ell_k \geq T^*$. But then we have

$$\sum_{s=1}^{n} t_n > M \cdot T^*$$

That is, the sum of all the production times is (strictly) larger than $M \cdot T^*$. As φ^* must allocate every product to a factory, and there are only M factories, there is at least one factory that gets assigned a load greater than T^*, contradicting the definition of T^*.

Thus there is no iteration i in which a factory gets a load of more than $2 \cdot T^*$, i.e. the algorithm is a 2-approximation.