# Proof of $T_p \leq T_1/p + T_\infty \leq 2\text{OPT}$ based on a thread counting argument

Alexei Finski

$T_p \leq T_1/p + T_\infty$

# complete steps $\leq T_1/p$

Suppose # complete steps $> T_1/p$. The size of a complete step is $p$. The work performed is $> T_1$. Contradiction.

# incomplete steps $\leq T_\infty$

Wlog, let the execution time for each thread be unit time. Every path in G starts from a single source thread and has a length that is shorter than or equal to $T_\infty$. Let $s_i$ be the set of threads that consists of $t_i$, a thread in a longest path of G, and the threads executed in parallel to $t_i$ given infinitely many processors.

After a greedy scheduler executes $s_i$, every thread in $s_{i+1}$ is executable or executed. By induction, at any time before program completion there exists $s_i^*$ that consists of executed and executable threads with at least one thread that is executable.

An incomplete step of a greedy scheduler must execute the last executable thread of $s_i^*$. Otherwise the step is complete. Thus # incomplete steps $\leq T_\infty$

.

$T_1/p + T_\infty \leq 2\text{OPT}$

$T_1/p \leq \text{OPT}$ and $T_\infty \leq \text{OPT}$.