

Total tardiness minimalization for multi-CPU jobs – preliminary results

This preliminary results show the **total tardiness** and **time required** to generate schedule when using the following heuristics:

- **MM** – Minimum Makespan dispatching rule – selects resource with currently lowest expected makespan and then add the new job to the end of this resource's schedule.
- **MTEDD** – Minimal Tardiness Earliest Due Date (*see the article for description*)
- **MTERD** – Minimal Tardiness Earliest Release Date (*see the article for description*)
- **MM+TS** – Minimal Makespan + Tabu Search after each job
- **MTEDD+TS** – Minimal Tardiness Earliest Due Date + Tabu Search after each job (*see the article for description*)
- **MTERD+TS** – Minimal Tardiness Earliest Release Date + Tabu Search after each job (*see the article for description*)

Each machine consisted of 3 CPUs and therefore each job may require 1, 2 or 3 CPUs for its run (according to its parameter). Each machine's incoming queue is handled by FCFS algorithm, i.e., if job A arrive earlier than job B on the machine then the job A will start its execution earlier or at the same time as B (enough CPUs to run both jobs) but not later.

