Scheduling Unrelated Parallel Machines

Algorithms, Complexity, and Performance

Dissertation of Andreas Wotzlaw

Abstract

A bank of parallel machines is an important setting from both theoretical and practical points of view. It is a generalization of a single machine which occurs very often in the real-world. Furthermore, when dealing with parallel machines, the makespan becomes an objective of significant interest. In practice one often has to *balance* the load on parallel machines and by minimizing the makespan the scheduler ensures a good load balance.

In this work we consider the problem of *scheduling unrelated parallel machines* ($R||C_{\max}$). The goal of this $\mathcal{NP}$-hard problem is to schedule without preemption a set of n independent jobs on a set of m parallel machines such that the makespan is minimum. For each job we are given its processing time p_{ij} on every machine it can be assigned to. We consider only *deterministic* scheduling problems for which n, m, and p_{ij} are known in advance.

The goal of our research is the development of new efficient algorithms and the practical study on various algorithmic tools for $R||C_{\max}$. The main theoretical result is a new *combinatorial* algorithm, *Unsplittable-Truemper*, for computing a 2-approximative schedule to $R||C_{\max}$. We prove that our approach runs within the best so far running time. In particular, this is the first time that a combinatorial algorithm always beats the interior point approach for $R||C_{\max}$.

Additionally, we introduce two other algorithms. The first one is a randomized version of the classical *two-step* approach for $R||C_{\max}$. The second one is a *branch-and-price* approach for $R||C_{\max}$ with two heuristic improvements which significantly improve the computations of integer solutions.

Finally, we present a comprehensive evaluation of eighteen methods for $R||C_{\max}$ using various algorithmic approaches. Our efforts aim at the performance and the quality of solutions obtained with *Unsplittable-Truemper* in comparison with two-step approaches using linear programming and with different heuristics and exact methods.