Near-Linear Approximation Algorithms for Scheduling Problems with Setup Times

Max Deppert[1] and Klaus Jansen[1]

University of Kiel, Germany
made,kj@informatik.uni-kiel.de

Keywords: Scheduling Theory, Approximation Algorithms, Setup Times.

1 Problem Definition

Scheduling problems with setup times have been intensively studied for over 30 years now; in fact, they allow very natural formulations of scheduling problems.

In the general scheduling problem with setup times, there are m identical and parallel machines, a set J of $n \in \mathbb{N}$ jobs $j \in J$, $c \in \mathbb{N}$ different classes, a partition $\dot{\bigcup}_{i=1}^{c} C_i = J$ of c nonempty and disjoint subsets $C_i \subseteq J$, a *processing time* of $t_j \in \mathbb{N}$ time units for each job $j \in J$ and a *setup* (or *setup time*) of $s_i \in \mathbb{N}$ time units for each class $i \in [c]$. The objective is to find a schedule which minimizes the makespan while holding the following. All jobs (or its complete sets of job pieces) are scheduled. Whenever a machine switches processing from one job to another, a setup may be necessary. There are various types of setups discussed; here we focus on *sequence-independent* batch setups, i.e. a setup only gets necessary when switching from one class of jobs to another *different* class on a machine and it does not depend on the previous job/class. All machines are *single-threaded* (jobs (or job pieces) and setups do not intersect in time on each machine) and no setup is preempted. There are three variants of scheduling problems with setup times which have been gaining the most attention in the past. There is the *non-preemptive* case where no job may be preempted, formally known as problem $\mathrm{P}|\mathrm{setup}{=}\mathrm{s_i}|\mathrm{C_{max}}$. Another variant is the *preemptive* context, namely $\mathrm{P}|\mathrm{pmtn}, \mathrm{setup}{=}\mathrm{s_i}|\mathrm{C_{max}}$, where a job may be preempted at any time but be processed on at most one machine at a time, so a job may not be parallelized. In the generous case of *splittable* scheduling, known as $\mathrm{P}|\mathrm{split}, \mathrm{setup}{=}\mathrm{s_i}|\mathrm{C_{max}}$, a job is allowed to be split into any number of job pieces which may be processed on any machine at any time.

2 Related results

Monma and Potts (1989) began their investigation of these problems considering the preemptive case. They found first dynamic programming approaches for various single machine problems polynomial in n but exponential in c. Furthermore, they showed NP-hardness for $\mathrm{P}|\mathrm{pmtn}, \mathrm{setup}{=}\mathrm{s_i}|\mathrm{C_{max}}$ even if $m = 2$. In a later work Monma and Potts (1993) found a heuristic which resembles McNaughton's preemptive wrap-around rule; see also (McNaughton 1959). It requires $\mathcal{O}(n)$ time for being $(2 - (\lfloor \frac{m}{2} + 1 \rfloor)^{-1})$-approximate. Notice that this ratio is truly greater than $\frac{3}{2}$ if $m \geq 4$ and the asymptotic bound is 2 for $m \to \infty$. Monma and Potts also discussed the problem class of *small batches* where for any batch i the sum of one setup time and the total processing time of all jobs in i is smaller than the optimal makespan, i.e. $s_i + \sum_{j \in C_i} t_j \leq OPT$. Most suitable for this kind of problems, they found a heuristic that first uses list scheduling for complete batches followed by an attempt of splitting some batches so that they are scheduled on two different machines. This second approach needs a running time of $\mathcal{O}(n + (m + c) \log(m + c))$ and considering only small batches it is $(\frac{3}{2} - \frac{1}{4m-4})$-approximate if $m \leq 4$ whereas it is

$(\frac{5}{3}-\frac{1}{m})$-approximate for small batches if m is a multiple of 3 and $m \geq 6$. Then Chen (1993) modified the second approach of Monma and Potts. For small batches Chen improved the heuristic to a worst case guarantee of $\max\{\frac{3m}{2m+1}, \frac{3m-4}{2m-2}\}$ if $m \geq 5$ while the same time of $\mathcal{O}(n+(m+c)\log(m+c))$ is required.

Schuurman and Woeginger (1999) studied the preemptive problem for *single-job-batches*, i.e. $|C_i| = 1$. They found a PTAS for the uniform setups problem $s_i = s$. Furthermore, they presented a $(\frac{4}{3}+\varepsilon)$-approximation in case of arbitrary setup times. Both algorithms have a running time linear in n but exponential in $1/\varepsilon$. Then Xing and Zhang (2000) turned to the splittable case. Without other restrictions they presented an FPTAS if m is fixed and a $\frac{5}{3}$-approximation in polynomial time if m is variable. They give some simple arguments that the problem is weakly NP-hard if m is fixed and NP-hard in the strong sense otherwise. More recently Mäcker *et. al.* (2015) made progress to the case of non-preemptive scheduling. They used the restrictions that all setup times are equal ($s_i = s$) and the total processing time of each class is bounded by γOPT for some constant γ, i.e. $\sum_{j\in C_i} t_j \leq \gamma OPT$. Mäcker et al. found a simple 2-approximation, an FPTAS for fixed m, and a $(1+\varepsilon)\min\{\frac{3}{2}OPT, OPT + t_{\max} - 1\}$-approximation (where $t_{\max} = \max_{j\in J} t_j$) in polynomial time if m is variable. Jansen and Land (2016) found three different algorithms for the non-preemptive context without restrictions. They presented an approximation ratio 3 using a next-fit strategy running in time $\mathcal{O}(n)$, a 2-dual approximation running in time $\mathcal{O}(n)$ which leads to a $(2+\varepsilon)$-approximation running in time $\mathcal{O}(n\log(\frac{1}{\varepsilon}))$, as well as a PTAS. Recently Jansen *et. al.* (2019) found an EPTAS for all three problem variants. For the preemptive case they assume $|C_i| = 1$. They make use of n-fold integer programs, which can be solved using the algorithm by Hemmecke, Onn, and Romanchuk. However, even after some runtime improvement the runtime for the splittable model is $2^{\mathcal{O}(1/\varepsilon^2\log^3(1/\varepsilon))}n^2\log^3(nm)$, for example. These algorithms are interesting answers to the question of complexity but they are useless for solving actual problems in practice. Therefore the design of *fast* (and especially polynomial) approximation algorithms with small approximation ratio remains interesting.

3 New Results

For all three problem variants we give a 2-approximate algorithm running in time $\mathcal{O}(n)$ as well as a $(\frac{3}{2}+\varepsilon)$-approximation with running time $\mathcal{O}(n\log(\frac{1}{\varepsilon}))$. With some runtime improvements we present some very efficient near-linear approximation algorithms with a constant approximation ratio equal to $\frac{3}{2}$. In detail, we find a $\frac{3}{2}$-approximation for the splittable case with running time $\mathcal{O}(n+c\log(c+m)) \leq \mathcal{O}(n\log(c+m))$. Also we will see a $\frac{3}{2}$-approximate algorithm for the non-preemptive case that runs in time $\mathcal{O}(n\log(T_{\min}))$ where $T_{\min} = \max\{\frac{1}{m}N, \max_{i\in[c]}(s_i + t_{\max}^{(i)})\}$, $t_{\max}^{(i)} = \max_{j\in C_i} t_j$ and $N = \sum_{i=1}^{c} s_i + \sum_{j\in J} t_j$. For the most complicated case of these three problem contexts, the preemptive case, we study a $\frac{3}{2}$-approximation running in time $\mathcal{O}(n\log(c+m)) \leq \mathcal{O}(n\log n)$. For the long version we refer to (Deppert and Jansen 2018). Especially the last result is interesting; we make progress to the general case where classes may consist of an *arbitrary* number of jobs. The best approximation ratio was the one by Monma and Potts (1993) mentioned above. All other previously known results for preemptive scheduling used restrictions like *small batches* or even *single-job-batches*, i.e. $|C_i| = 1$. As a byproduct we give some new *dual* lower bounds.

References

B. Chen. A better heuristic for preemptive parallel machine scheduling with batch setup times. *SIAM Journal on Computing*, 22(6):1303–1318, 1993.

M. A. Deppert and K. Jansen. Near-linear approximation algorithms for scheduling problems with batch setup times. *CoRR*, abs/1810.01223, 2018.

K. Jansen, K. Klein, M. Maack, and M. Rau. Empowering the configuration-ip - new PTAS results for scheduling with setups times. In A. Blum, editor, *Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *LIPIcs*, pages 44:1–44:19, 2019.

K. Jansen and F. Land. Non-preemptive scheduling with setup times: A PTAS. In P. Dutot and D. Trystram, editors, *European Conference on Parallel and Distributed Computing (Euro-Par 2016)*, volume 9833 of *LNCS*, pages 159–170. Springer, 2016.

A. Mäcker, M. Malatyali, F. Meyer auf der Heide, and S. Riechers. Non-preemptive scheduling on machines with setup times. In F. Dehne, J. Sack, and U. Stege, editors, *Symposium on Algorithms and Data Structures (WADS 2015)*, volume 9214 of *LNCS*, pages 542–553. Springer, 2015.

R. McNaughton. Scheduling with deadlines and loss functions. *Management Science*, 6(1):1–12, Oct. 1959.

C. L. Monma and C. N. Potts. On the complexity of scheduling with batch setup times. *Operations Research*, 37(5):798–804, 1989.

C. L. Monma and C. N. Potts. Analysis of heuristics for preemptive parallel machine scheduling with batch setup times. *Operations Research*, 41(5):981–993, 1993.

P. Schuurman and G. J. Woeginger. Preemptive scheduling with job-dependent setup times. In R. E. Tarjan and T. J. Warnow, editors, *Symposium on Discrete Algorithms (SODA 1999)*, pages 759–767.

W. Xing and J. Zhang. Parallel machine scheduling with splitting jobs. *Discrete Applied Mathematics*, 103(1-3):259–269, 2000.