The Bin Scheduling Problem

Elisabeth Günther * Felix G. König* Nicole Megow †

1 Introduction

We consider a natural generalization of multiprocessor scheduling, the *bin scheduling problem*. Here, we model the situation in which working resources are available during separate time windows, or shifts, only. Restricting to scheduling without job preemption, every task must be processed completely during one shift. Our aim is to find a feasible schedule in which the number of required shifts is minimized.

This generalization corresponds to an analogous relationship between strip packing and two-dimensional bin packing (see [4]) as demonstrated in Figure 1. This work is based on results from the comprehensive study of bin scheduling in [2].

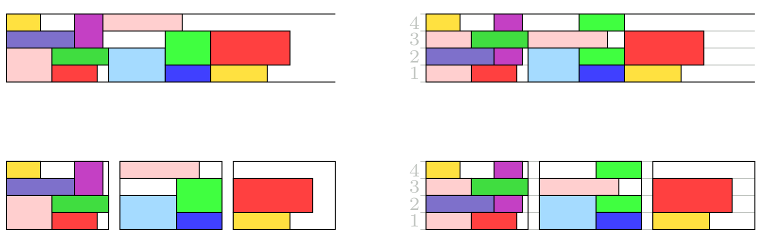

Figure 1: Corresponding instances of packing and scheduling: Strip packing and 2D bin packing solutions on the left, and multiprocessor schedules without and with bins on the right.

More precisely, we are given a set of n tasks J which have to be executed without preemption on m identical parallel processors subject to precedence constraints given by a partial order $(J, \prec)$. The tasks are *malleable*, i.e., the processing time of each task depends on the number of processors alloted to it. To reflect realistic conditions it is common to assume every function of processing times p_j to be nonincreasing in the number of alloted processors α_j, and to have the property that the resulting work function defined as $w_j(\alpha_j) = \alpha_j \cdot p_j(\alpha_j)$ is nondecreasing in α_j for each task $j \in J$. Such an *allotment* $(\alpha_j)_{j \in J}$ together with an assignment of starting times σ_j to each task is a *feasible schedule* if and only if the precedence constraints are respected and the number of used processors never exceeds the number of available processors m.

We refer to such an instance with the objective to minimize the makespan, as *makespan problem*. The input of the *bin scheduling problem* contains an additional constant D which represents the duration of each shift, or *bin*. A *feasible bin schedule* is given by a partition $J_1, \ldots, J_b$

*{eguenth,fkoenig}@math.tu-berlin.de. Technische Universität Berlin, Institut für Mathematik, Germany.
†nmegow@mpi-inf.mpg.de. Max-Planck-Institut für Informatik, Saarbrücken, Germany.

of J together with feasible schedules for every subset J_i with individual makespan at most D. Furthermore, the partition $J_1, \ldots, J_b$ must respect $(J, \prec)$, i.e., the index i_p of the subset J_{i_p} containing a task p must not be greater than the index i_s of the subset J_{i_s} containing task s for every $p \prec s$. Our objective is to find a feasible bin schedule requiring the minimum number of bins.

Since the makespan of every sub-schedule must not exceed D, every task must be alloted enough processors to have processing time no greater than D. Hence we have to extend the input of the makespan problem by constants $1 \leq \ell_j \leq u_j \leq m$ for each task $j \in J$ as lower and upper bound for the number of processors alloted to j and require $p_j(\ell_j) \leq D$. This makes parallel tasks with fixed processing times a special case.

It is worth noting that the makespan as well as the bin scheduling problem and also many special cases are $\mathcal{NP}$-hard in the strong sense.

Related work. The bin scheduling problem was motivated in the context of a real-world application, namely shutdown and turnaround scheduling, proposed in [6], where the considered resources are only available during given shifts.

The makespan problem without bounds on a feasible allotment has been considered in [3] and [5] where constant factor approximations have been derived. To the best of our knowledge no approximation algorithms respecting such bounds exist.

The only approximation algorithm for tasks with a fixed allotment we are aware of, arises from the fact, that the makespan scheduling problem is somewhat related to the strip packing problem. In fact tasks scheduled on contiguous processors can be seen as rectangles in packing. Yet in general scheduling, multiprocessors are not required to be contiguous, again see figure 1. In [1] precedence relations in strip packing are considered and a factor $\Theta(\log n)$ approximation is presented. Due to the fact that the lower bound used also holds for the makespan in nonmalleable scheduling approximation carries over.

2 Approximation Algorithms

Our most general result is an approximation algorithm for the bin scheduling problem with an approximation factor $\Theta(\log n)$. The approach is based on the observation that every ϱ-approximation algorithm for the makespan variant can be turned into a $2\lceil \varrho \rceil$-approximation algorithm for bin scheduling. Thus, the main difficulty within our framework lies in good approximations for the makespan scheduling problem.

Although the algorithms in [3] and [5] are not directly applicable in our setting, because they cannot handle arbitrary lower bounds on the number of processors alloted to each task, some of their ideas can be adopted to find an allotment yielding good lower bounds for our problem. Applying the techniques from [1] for the precedence constrained strip packing problem with this fixed allotment we finally achieve a $\Theta(\log n)$-approximation algorithm for our makespan problem and hence for bin scheduling. We can show that the analysis is tight and, following leads in [1], that it is not possible to prove constant performance guarantees without devising entirely new ideas for lower bounds. In this sense, our results are the best possible.

In case the structure of this core problem is simpler, that is, if we find constant ratio approximation algorithms for the makespan problem, we achieve constant performance results for our general algorithm. We consider the problem class with partial orders of width bounded by a constant κ. We prove that both problems, makespan scheduling and two-dimensional packing, remain $\mathcal{NP}$-hard for every $\kappa \geq 3$. We design an FPTAS for the makespan problem which also gives the first constant factor approximation for a strip packing problem with precedence constraints and polynomially bounded strip width. The FPTAS immediately results in a 4-approximation for the corresponding bin scheduling problem. With a little more effort, this can be improved to a 3-approximation.

These ideas also lead to pseudopolynomial exact algorithms for the makespan and the bin scheduling problem as well as precedence constrained strip packing and precedence constrained 2D bin packing respectively for arbitrary κ. For the special case $\kappa = m = 2$, we obtain an efficient algorithm solving bin scheduling to optimality.

References

[1] J. Augustine, S. Banerjee, and S. Irani. Strip packing with precedence constraints and strip packing with release times. In *Proceedings of the 18th Annual ACM symposium on Parallelism in algorithms and architectures (SPAA'06)*, pages 180–189, 2006.

[2] E. Günther. *Diploma thesis (in German)*, Technische Universitt Berlin, 2008.

[3] K. Jansen and H. Zhang. An approximation algorithm for scheduling malleable tasks under general precedence constraints. *ACM Transactions on Algorithms*, 2(3):416–434, 2006.

[4] A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252, 2002.

[5] R. Lepère, D. Tystram, and G. J. Woeginger. Approximation Algorithms for Scheduling Malleable Tasks Under Precedence Constraints. *International Journal of Foundations of Computer Science*, 13(4):613–627, 2002.

[6] N. Megow, R. H. Möhring, and J. Schulz. Turnaround scheduling in chemical manufacturing. In *Proceedings of the 8th Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP 2007)*, Istanbul, Turkey, 2007.