Polynomial time approximation algorithms for machine scheduling: Ten open problems

Petra Schuurman * Gerhard J. Woeginger †

Abstract

We discuss what we consider to be the ten most vexing open questions in the area of polynomial time approximation algorithms for NP-hard deterministic machine scheduling problems. We summarize what is known on these problems, we discuss related results, and we provide pointers to the literature.

Prologue

In the early days of scheduling, the work of the theoretical research branch mainly consisted of classifying scheduling problems into easy (i.e., polynomially solvable) and hard (i.e., NP-hard) ones. Nowadays, researchers have become interested in a better understanding and a much finer classification of the hard problems; one very active branch of research classifies hard scheduling problems according to their *approximability*. In the sequel, we summarize and discuss what in our opinion are the most outstanding open questions in the approximation of scheduling problems. Throughout the paper, we use the standard three-field scheduling notation (see e.g. Graham, Lawler, Lenstra & Rinnooy Kan [26] and Lawler, Lenstra, Rinnooy Kan & Shmoys [41]). All scheduling problems discussed are *minimization* problems, where the goal is to find a feasible schedule with minimum possible cost. We consider only scheduling problems that are NP-hard (cf. Garey & Johnson [19]). We distinguish between *positive results*, which establish the existence of some approximation algorithm, and *negative results*, which disprove the existence of certain approximation results under the assumption P $\neq$ NP.

Positive (approximability) results. A standard way of dealing with NP-hard problems is not to search for an optimal solution, but to go for *near-optimal* solutions. An algorithm that returns near-optimal solutions is called an *approximation algorithm*; if it does this in polynomial time, then it is called a *polynomial time* approximation algorithm. An approximation algorithm that always returns a near-optimal solution with cost at most a factor ρ above the optimal cost (where $\rho > 1$ is some fixed real number) is called a *ρ-approximation algorithm*, and the value ρ

*`petra@win.tue.nl`. Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands.

†`gwoegi@opt.math.tu-graz.ac.at`. Institut für Mathematik B, TU Graz, Steyrergasse 30, A-8010 Graz, Austria. Supported by the START program Y43-MAT of the Austrian Ministry of Science.

is called the *worst-case performance guarantee*. A family of $(1+\varepsilon)$-approximation algorithms over all $\varepsilon > 0$ with polynomial running times is called a *polynomial time approximation scheme* or PTAS, for short. If the time complexity of a PTAS is also polynomially bounded in $1/\varepsilon$, then it is called a *fully* polynomial time approximation scheme or FPTAS, for short. With respect to relative performance guarantees, an FPTAS is essentially the strongest possible polynomial time approximation result that we can derive for an NP-hard problem.

Positive results in the area of approximation concern the design and analysis of such polynomial time approximation algorithms and schemes. For every hard problem, we would like to know whether it possesses a polynomial time approximation algorithm with constant worst-case performance guarantee, or a PTAS, or even an FPTAS. For surveys on polynomial time approximation algorithms for scheduling we refer the reader to Hall [27] and to Lenstra & Shmoys [45].

Negative (inapproximability) results. Negative results in the area of approximation disprove the existence of good approximation algorithms under the assumption P $\neq$ NP. E.g. if P $\neq$ NP, then *strongly* NP-hard problems cannot have an FPTAS (cf. Garey & Johnson [18, 19]); strongly NP-hard problems are problems that remain NP-hard even if the numbers in their input are unary encoded. One way of disproving the existence of a PTAS for a scheduling problem under P $\neq$ NP is via a *gap-reduction*: Such a reduction transforms the YES-instances of some NP-hard problem into scheduling instances with objective value at most c^*, and it transforms the NO-instances into scheduling instances with objective value at least $\rho \cdot c^*$, where $\rho > 1$ is some fixed real number. Then a polynomial time approximation algorithm for the scheduling problem with performance guarantee strictly better than ρ would be able to separate the YES-instances from the NO-instances, thus yielding a polynomial time solution algorithm for an NP-hard problem. Another way of disproving the existence of a PTAS for a scheduling problem under P $\neq$ NP is to establish its MAX SNP-hardness. This essentially means that there is an approximation preserving reduction from a problem that is known to be hard to approximate under P $\neq$ NP. For an exact explanation we refer the reader to Papadimitriou & Yannakakis [53] and Arora & Lund [2]. For a compendium of publications on inapproximability results we refer to Crescenzi & Kann [13].

The open problems

We have ordered the open problems according to their objective function. Our first seven open problems deal with the makespan criterion, problems 1 through 4 in parallel machine environments and problems 5 through 7 in shop environments. The objective in problems 8 and 9 is to minimize total job completion time. Finally, in problem 10 the flow time criterion is handled. In the sequel we use the expressions $c - \delta$ and $c + \delta$ to denote real numbers that are strictly smaller than c and strictly larger than c, respectively; i.e., δ always denotes some small positive real number that does not depend on the input.

Makespan minimization on identical machines under precedence constraints. Already in the 1960s, Graham [25] showed that a simple greedy algorithm called *List Scheduling* has a worst-case performance guarantee of $2 - 1/m$ on m identical machines. Till now, there is no polynomial time approximation algorithm known that beats this performance guarantee, even for unit processing times. Lenstra & Rinnooy Kan [44] proved via a gap reduction that, unless P = NP, one cannot reach a worst-case performance guarantee better than 4/3 for $P \,|\, prec, p_j{=}1 \,|\, C_{\max}$ in polynomial time.

Open problem 1 *Design a polynomial time approximation algorithm for $P \,|\, prec \,|\, C_{\max}$ or for $P \,|\, prec, p_j{=}1 \,|\, C_{\max}$ with worst-case performance $2 - \delta$ (in fact, even an algorithm whose time complexity is exponential in m might be interesting). Provide a $4/3 + \delta$ inapproximability result for $P \,|\, prec \,|\, C_{\max}$.*

Design a PTAS for problem $P2 \,|\, prec \,|\, C_{\max}$. Design a PTAS (or even an FPTAS) for problem $P3 \,|\, prec, p_j{=}1 \,|\, C_{\max}$.

We remark that Du, Leung & Young [14] showed $P2 \,|\, prec \,|\, C_{\max}$ to be strongly NP-hard (even for chain-type precedence constraints); hence, it cannot have an FPTAS. Problem $P2 \,|\, prec, p_j{=}1 \,|\, C_{\max}$ is polynomially solvable by techniques from matching theory (Fujii, Kasami & Ninomiya [17]). It is unknown whether $P3 \,|\, prec, p_j{=}1 \,|\, C_{\max}$ is NP-hard; determining its computational complexity is one of the four still unresolved open problems in the list of twelve open problems in the book of Garey & Johnson [19]. The best polynomial time approximation algorithm known for $P3 \,|\, prec, p_j{=}1 \,|\, C_{\max}$ has a performance guarantee of 4/3 and is due to Lam & Sethi [39]. In fact, these authors provide polynomial time approximation algorithms with performance guarantee $2 - 2/m$ for the more general problem $Pm \,|\, prec \,|\, C_{\max}$ and for its preemptive variant $Pm \,|\, pmtn, prec \,|\, C_{\max}$.

Makespan minimization on uniform machines under precedence constraints. This problem is considerably harder than the corresponding problem on identical machines: No polynomial time approximation algorithm with constant performance guarantee is known. In 1980, Jaffe [36] designed a polynomial time approximation algorithm for $Q \,|\, prec \,|\, C_{\max}$ with worst-case performance $O(\sqrt{m})$ for m machines. In 1997, Chudak & Shmoys [12] developed a better algorithm with worst-case performance $O(\log m)$; another polynomial time approximation algorithm with the same order of worst-case performance but a simpler worst-case analysis was given by Chekuri & Bender [6].

Open problem 2 *Design a polynomial time approximation algorithm for $Q \,|\, prec \,|\, C_{\max}$ with constant performance guarantee (i.e., independent of the number m of machines), or prove a non-constant lower bound under* P $\neq$ NP *(i.e., a bound that tends to infinity as m goes to infinity).*

The preemptive problem $Q \,|\, pmtn, prec \,|\, C_{\max}$ is NP-hard (Ullman [66] shows that its special case $P \,|\, pmtn, prec \,|\, C_{\max}$ is NP-hard), and the approach of Chudak & Shmoys [12] yields a polynomial time $O(\log m)$-approximation algorithm for it. Nothing better is known, and we think that the preemptive version might be easier to attack than the non-preemptive version.

Makespan minimization under precedence constraints with communication delays. In this setting (cf. Papadimitriou & Yannakakis [52] and Veltman, Lageweg & Lenstra [67]) data transmission times between machines have to be taken into account: The presence of an arc $j \to k$ (that corresponds with a precedence relation between j and k) implies that if job j and job k are processed on different machines, then the processing of k cannot start earlier than c_{jk} time units after the completion time of j; on the other hand, if j and k are processed on the same machine, then k cannot start before j has been completed. This situation is denoted by an entry c_{jk} in the second field of the scheduling notation, respectively by c if all communication delays are equal (so-called *uniform* communication delays). We consider two basic variants of scheduling with communication delays: $P \,|\, prec, c_{jk} \,|\, C_{\max}$ where the number of machines is restrictively given as part of the input, and $P\infty \,|\, prec, c_{jk} \,|\, C_{\max}$ where the number of machines to be used may be chosen by the scheduler.

Munier & Hanen [49] provide a polynomial time $(7/3 - 4/(3m))$-approximation algorithm for $P \,|\, prec, p_j{=}1, c{=}1 \,|\, C_{\max}$. Hoogeveen, Lenstra & Veltman [32] show by a gap-reduction that $P \,|\, prec, p_j{=}1, c{=}1 \,|\, C_{\max}$ does not possess a polynomial time approximation algorithm with performance guarantee better than 5/4 (unless P = NP). Munier & König [50] present a polynomial time 4/3-approximation algorithm for problem $P\infty \,|\, prec, p_j{=}1, c{=}1 \,|\, C_{\max}$ which is based on a linear relaxation of an integer linear programming formulation. Hoogeveen, Lenstra & Veltman [32] prove a lower bound of 7/6 on the performance guarantee of any polynomial time approximation algorithm for this problem.

Open problem 3 *For $P \,|\, prec, p_j{=}1, c{=}1 \,|\, C_{\max}$, improve the performance guarantee to $7/3 - \delta$ or improve the inapproximability bound to $5/4 + \delta$. For $P\infty \,|\, prec, p_j{=}1, c{=}1 \,|\, C_{\max}$, improve the performance guarantee to $4/3 - \delta$ or improve the inapproximability bound to $7/6 + \delta$.*

Decide whether there exists a polynomial time approximation algorithm with constant performance guarantee for $P\infty \,|\, prec, p_j{=}1, c \,|\, C_{\max}$. Decide whether there exists a polynomial time approximation algorithm with constant performance guarantee for $P\infty \,|\, prec, c_{jk} \,|\, C_{\max}$.

Papadimitriou & Yannakakis [52, 54] claim (without providing a proof) a lower bound of 2 on the performance guarantee of any polynomial time approximation algorithm for problem $P\infty \,|\, prec, p_j{=}1, c \,|\, C_{\max}$. It would be nice to have a proof for this claim.

Makespan minimization on unrelated machines. On unrelated machines, the approximation of makespan is even interesting without any precedence constraints or communication delays. Lenstra, Shmoys & Tardos [46] give a polynomial time 2-approximation algorithm for $R \,|\,|\, C_{\max}$. Moreover, they prove via a gap-reduction that (unless P = NP) one cannot reach a worst-case performance guarantee better than 3/2 in polynomial time.

Open problem 4 *Design a polynomial time approximation algorithm for $R \,|\,|\, C_{\max}$ with worst-case performance $2 - \delta$ or provide a $3/2 + \delta$ inapproximability result for $R \,|\,|\, C_{\max}$.*

It would even be interesting to improve on the results of Lenstra, Shmoys & Tardos [46] in the so-called *restricted assignment* variant of $R\,|\,|\,C_{\max}$: In this variant, the processing time p_{ij} of job j on machine i fulfills $p_{ij} \in \{p_j, \infty\}$, i.e., the processing time of job j essentially equals p_j, but the job can only be processed on a subset of the machines. The gap-reduction (and hence the 3/2 inapproximability result) of Lenstra, Shmoys & Tardos [46] also applies to this restricted assignment variant. Furthermore, there is no better polynomial time approximation algorithm known for this problem than the one for the general $R\,|\,|\,C_{\max}$ problem.

We remark that the simpler problem $Rm\,|\,|\,C_{\max}$ on a fixed number of unrelated machines possesses an FPTAS (Horowitz & Sahni [35]). The simpler problem $Q\,|\,|\,C_{\max}$ on an arbitrary number of uniform machines possesses a PTAS (Hochbaum & Shmoys [31]).

———————

Makespan minimization in open shops. The open shop on a fixed number of machines is fairly well understood. Sevastianov & Woeginger [61] give a PTAS for $Om\,|\,|\,C_{\max}$. However, it is unknown whether $Om\,|\,|\,C_{\max}$ is strongly NP-hard or whether $Om\,|\,|\,C_{\max}$ can have an FPTAS. On the other hand, the open shop $O\,|\,|\,C_{\max}$ with an arbitrary number of machines is known to be strongly NP-hard. Williamson, Hall, Hoogeveen, Hurkens, Lenstra, Sevastianov & Shmoys [68] prove that unless P = NP, one cannot reach a worst-case performance guarantee better than 5/4 for $O\,|\,|\,C_{\max}$. On the positive side, Racsmány (private communication cited by Bárány & Fiala [3]) observed that so-called *dense* schedules yield a simple polynomial time 2-approximation algorithm. A feasible schedule for the open shop problem is called *dense* when any machine is idle only if there is no job that currently could be processed on that machine. It is conjectured that dense schedules even yield a $(2 - 1/m)$-approximation of the optimal makespan. Chen & Strusevich [9] prove this conjecture for $m \leq 3$, and Chen & Yu [10] prove it for $m = 4$.

Open problem 5 *Design a polynomial time approximation algorithm for $O\,|\,|\,C_{\max}$ with worst-case performance $2 - \delta$. Provide a $5/4 + \delta$ inapproximability result for $O\,|\,|\,C_{\max}$.*

The preemptive variant $O\,|\,pmtn\,|\,C_{\max}$ is polynomially solvable (Gonzalez & Sahni [23]). It has been conjectured that the optimum non-preemptive makespan is always at most a factor of 3/2 above the optimum preemptive makespan; if true, this should lead to a polynomial time 3/2-approximation algorithm for $O\,|\,|\,C_{\max}$.

———————

Makespan minimization in flow shops. As for the open shop problem above, the approximability status of the flow shop strongly depends on whether or not the number of machines is part of the input. On the one hand, problem $Fm\,|\,|\,C_{\max}$ (minimizing the makespan in a flow shop with a fixed number of machines) has a PTAS (Hall [28]). Since $F3\,|\,|\,C_{\max}$ is strongly NP-hard (Garey, Johnson & Sethi [20]), problem $Fm\,|\,|\,C_{\max}$ does not have an FPTAS unless P = NP. On the other hand, Williamson & al. [68] prove that unless P = NP, there is no polynomial time approximation algorithm with a worst-case performance guarantee better than 5/4 for the problem $F\,||\,C_{\max}$ with an arbitrary number of machines. By extending the ideas of Shmoys, Stein & Wein [62] and Goldberg, Paterson, Srinivasan & Sweedyk [22], Feige & Scheideler [16] construct polynomial time approximation algorithms with a performance guarantee of $O(\log m \log\log m)$ for $F\,||\,C_{\max}$ on m machines.

Open problem 6 *Design a polynomial time approximation algorithm for $F\,||\,C_{\max}$ with constant (i.e., independent of the number m of machines) worst-case performance. Provide a $5/4+\delta$ inapproximability result for $F\,||\,C_{\max}$.*

A *permutation schedule* for a flow shop instance is a schedule in which each machine processes the jobs in the same order. It is strongly NP-hard to compute the best permutation schedule for $Fm\,||\,C_{\max}$ even for $m = 3$ machines (cf. Graham, Lawler, Lenstra & Rinnooy Kan [26]). The PTAS of Hall [28] can be modified to approximate the best permutation schedule for $Fm\,||\,C_{\max}$; the corresponding approximation problem for an arbitrary number of machines is open.

In general, the best permutation schedule can be far away from the optimal schedule. Röck & Schmidt [57] prove that the makespan of the best permutation schedule is at most a factor of $\lceil m/2 \rceil$ above the optimal makespan. Potts, Shmoys & Williamson [56] construct a family of instances for which the makespan of the best permutation schedule is a factor of $\lceil 1/2 + \sqrt{m} \rceil$ above the optimal makespan. It would be nice to have tighter upper and lower bounds on the ratio between these two quantities.

Makespan minimization in job shops. In the general job shop problem, the same job may return many times to the same machine. Job j is a chain $(O_{1j}, \ldots, O_{m_j j})$ of operations. The processing time of operation O_{ij} equals p_{ij} and O_{ij} has to be processed by one of the machines. We denote by μ the maximum number of operations over all jobs. Only very recently Jansen, Solis-Oba & Sviridenko [37] designed a PTAS for $Jm\,||\,C_{\max}$ on a fixed number m of machines where the parameter μ is also bounded by a constant. Since the job shop problem contains the flow shop problem as a special case, the 5/4 lower bound result of Williamson & al. [68] for the flow shop problem $F\,||\,C_{\max}$ also carries over to the job shop problem $J\,||\,C_{\max}$. Feige & Scheideler [16] give polynomial time approximation algorithms with performance guarantee $O(\log(m\mu)\log\log(m\mu))$ for $J\,||\,C_{\max}$.

Open problem 7 *Decide whether there exists a polynomial time approximation algorithm for $J\,||\,C_{\max}$ whose worst-case performance is independent of the number m of machines and/or independent of the maximum number μ of operations. Provide a $5/4 + \delta$ inapproximability result for $J\,||\,C_{\max}$. Provide an inapproximability result for $J\,||\,C_{\max}$ whose value grows with the number m of machines to infinity.*

Design a PTAS for $J2\,||\,C_{\max}$ for the case where μ is part of the input, or disprove the existence of such a PTAS under $\mathrm{P} \neq \mathrm{NP}$.

The work of Leighton, Maggs & Rao [42] and Leighton, Maggs & Richa [43] yields polynomial time approximation algorithms with constant performance guarantee for a special case of $J \,|\, p_{ij}{=}1 \,|\, C_{\max}$ where all operations have unit processing times and where every job is processed exactly once on every machine. It is unknown whether this variant (or the more general variant of $J \,|\, p_{ij}{=}1 \,|\, C_{\max}$ where the jobs may return to machines) allows a PTAS. It is also unknown whether the results in Leighton, Maggs & Rao [42] and Leighton, Maggs & Richa [43] can be carried over to the job shop problem with arbitrary processing times. Feige & Rayzman [15] demonstrate that with respect to certain properties, the combinatorics of the job shop with unit processing times is considerably simpler than the combinatorics of the job shop with arbitrary processing times.

In the somewhat related *assembly line* problem $Am \,|\,|\, C_{\max}$, every job must be processed exactly once on every machine. The last operation of a job is an assembly operation performed on the last machine; it can only be started when the first $m-1$ operations all have been completed. The first $m-1$ operations, however, may be run in parallel and may overlap in time. A simple modification of the technique in Hall [28] yields a PTAS for $Am \,|\,|\, C_{\max}$ with a fixed number of machines. Since the problem $A3 \,|\,|\, C_{\max}$ is strongly NP-hard (Potts, Sevastianov, Strusevich, Van Wassenhove & Zwaneveld [55]), one cannot hope for an FPTAS for $Am \,|\,|\, C_{\max}$. For $A \,|\,|\, C_{\max}$ on an arbitrary number m of machines they give a polynomial time $(2-1/m)$-approximation algorithm. Apart from this result, the approximability behavior of $A \,|\,|\, C_{\max}$ is absolutely unclear; it might even have a PTAS.

Total job completion time without precedence constraints. Recently, there has been quite some progress in this area. For various parallel machine problems different PTASes have been constructed. Skutella & Woeginger [65] gave a PTAS for $P \,|\,|\, \sum w_j C_j$. Shortly after this result, there followed PTASes for $P \,|\, r_j \,|\, \sum w_j C_j$, $P \,|\, r_j, pmtn \,|\, \sum w_j C_j$, $Rm \,|\, r_j \,|\, \sum w_j C_j$, and $Rm \,|\, r_j, pmtn \,|\, \sum w_j C_j$; this is work in progress by various subsets of the author set {Chekuri, Karger, Khanna, Queyranne, Skutella, Stein and Sviridenko} [7]. Currently (i.e., at the moment of writing these lines), the approximability status of $Q \,|\, r_j \,|\, \sum w_j C_j$ is still unknown. However, we feel that modifications of the methods of [7] should eventually lead to a PTAS.

Chudak [11] and Schulz & Skutella [58] give a polynomial time approximation algorithm for the general problem $R \,|\,|\, \sum w_j C_j$ with performance guarantee $3/2+\varepsilon$ where $\varepsilon > 0$ can be made arbitrarily close to 0. Independently of each other, Skutella [63] and Sethuraman & Squillante [60] derive polynomial time 3/2-approximation algorithms for this problem and thus get rid of the ε in the performance guarantee. For $R \,|\, r_j \,|\, \sum w_j C_j$, Schulz & Skutella [59] provide a polynomial time $(2+\varepsilon)$-approximation algorithm with arbitrarily small $\varepsilon > 0$. As for $R \,|\,|\, \sum w_j C_j$, Skutella [64] gets rid of the ε to obtain a polynomial time 2-approximation algorithm for $R \,|\, r_j \,|\, \sum w_j C_j$. On the negative side, Hoogeveen, Schuurman & Woeginger [33] prove that $R \,|\,|\, \sum w_j C_j$ and $R \,|\, r_j \,|\, \sum C_j$ are MAX SNP-hard and hence do not have a PTAS.

Open problem 8 *Design a polynomial time approximation algorithm with performance guarantee $3/2-\delta$ for $R \,|\,|\, \sum w_j C_j$. Design a polynomial time approximation algorithm with performance guarantee $2-\delta$ for $R \,|\, r_j \,|\, \sum w_j C_j$ (or for $R \,|\, r_j \,|\, \sum C_j$).*

Derive better inapproximability results for $R \,|\,|\, \sum w_j C_j$ and $R \,|\, r_j \,|\, \sum w_j C_j$.

We remark that $Q\,|\,pmtn\,|\sum C_j$ is polynomially solvable (Lawler & Labetoulle [40]) and that $Qm\,|\,|\sum w_jC_j$ has an FPTAS (Woeginger [69]). $R\,|\,|\sum C_j$ is polynomially solvable (Horn [34]). $R2\,|\,|\sum w_jC_j$ is known to be NP-hard in the ordinary sense (Bruno, Coffman & Sethi [5]); deciding whether it is strongly NP-hard is an open problem. Whereas $R\,|\,pmtn\,|\sum w_jC_j$ has been proven NP-hard (in fact even its special case $P2\,|\,pmtn\,|\sum w_jC_j$ is NP-hard [26]), the complexity of $R\,|\,pmtn\,|\sum C_j$ is unknown. It would be interesting to get for it at least a polynomial time approximation algorithm with performance guarantee close to 1. Skutella [64] gives a polynomial time 2-approximation algorithm for $R\,|\,pmtn\,|\sum w_jC_j$ and a polynomial time 3-approximation algorithm for $R\,|\,r_j, pmtn\,|\sum w_jC_j$.

Very little is known about the approximability of shop problems with minsum criteria. Gonzalez & Sahni [24] provide a polynomial time approximation algorithm for the m machine flow shop problem $F\,|\,|\sum C_j$ with worst-case guarantee m. Hoogeveen, Schuurman & Woeginger [33] show that both $F\,|\,|\sum C_j$ and $O\,|\,|\sum C_j$ are MAX SNP-hard and thus do not have a PTAS. It would be nice to have polynomial time approximation algorithms for the above shop problems with small constant worst-case performance guarantee. A first step might be to understand the approximability of $O2\,|\,|\sum C_j$ and $F2\,|\,|\sum C_j$. Both problems are strongly NP-hard; cf. Achugbue & Chin [1] for the complexity of the open shop problem, and Garey, Johnson & Sethi [20] for the complexity of the flow shop problem. For both problems, nothing is known but the trivial polynomial time 2-approximation algorithm that sequences the jobs in order of non-decreasing total length (Gonzalez & Sahni [24]).

Total job completion time under precedence constraints. Adding precedence constraints to the total job completion time criterion seems to make the problem a lot harder, even on a single machine: Hall, Schulz, Shmoys & Wein [29] give polynomial time 2-approximation algorithms for the problems $1\,|\,prec\,|\sum w_jC_j$ and $1\,|\,prec, r_j, pmtn\,|\sum w_jC_j$, and they give a polynomial time 3-approximation algorithm for $1\,|\,prec, r_j\,|\sum w_jC_j$. Schulz & Skutella [58] give a polynomial time approximation algorithm for $1\,|\,prec, r_j\,|\sum w_jC_j$ whose performance guarantee can be made arbitrarily close to the Euler constant e. Independently of each other, Chekuri & Motwani [8] and Margot, Queyranne & Wang [48] provide (identical) extremely simple polynomial time 2-approximation algorithms for $1\,|\,prec\,|\sum w_jC_j$. Goemans & Williamson [21] provide a nice geometric proof for the performance guarantee of this algorithm. Hall, Schulz, Shmoys & Wein [29], Schulz & Skutella [58], and Chekuri & Motwani [8] argue that approaches based on certain classes of relaxations are not able to improve on this factor of 2.

Open problem 9 *Prove that $1\,|\,prec\,|\sum C_j$ and $1\,|\,prec\,|\sum w_jC_j$ do not have polynomial time approximation algorithms with performance guarantee $2-\delta$, unless* P = NP.

In fact, it would even be interesting to derive one of the following weaker results: (i) Establish MAX SNP-hardness of $1\,|\,prec\,|\sum C_j$ and $1\,|\,prec\,|\sum w_jC_j$. (ii) Prove that $1\,|\,prec\,|\sum w_jC_j$ is not more difficult to approximate than $1\,|\,prec\,|\sum C_j$. In other words, prove that the existence of a polynomial time ρ-approximation algorithm for $1\,|\,prec\,|\sum C_j$ implies the existence of a polynomial time ρ-approximation algorithm for $1\,|\,prec\,|\sum w_jC_j$. (iii) Prove that the existence of a polynomial time ρ-approximation algorithm for $1 \mid prec \mid \sum C_j$ implies the existence of a polynomial time ρ-approximation algorithm for the *vertex cover* problem. The vertex cover problem takes an undirected graph as an input and asks for a minimum cardinality subset of the vertices that touches every edge. It has been proven that (unless P = NP) vertex cover does not have a polynomial time $(7/6-\delta)$-approximation algorithm (Håstad [30]), and it is strongly conjectured that it does not have a polynomial time $(2-\delta)$-approximation algorithm.

Munier, Queyranne & Schulz [51] give a polynomial time 4-approximation algorithm for $P \mid prec, r_j \mid \sum w_j C_j$. Building on the ideas of Lenstra & Rinnooy Kan [44], Hoogeveen, Schuurman & Woeginger [33] prove that unless P = NP, one cannot reach a worst-case performance guarantee better than 8/7 for $P \mid prec, p_j{=}1 \mid \sum C_j$ or a worst-case performance guarantee better than 4/3 for $P \mid prec \mid \sum C_j$. It would be interesting to improve on all these bounds.

Flow time criteria. Finally, we briefly look at objective functions that depend on the flow time $F_j = C_j - r_j$ of job j. Kellerer, Tautenhahn & Woeginger [38] give a polynomial time $O(\sqrt{n})$-approximation algorithm for $1 \mid r_j \mid \sum F_j$ where n denotes the number of jobs. Moreover, they prove via a gap-reduction that for any value of $\delta > 0$ the existence of a polynomial time $O(n^{1/2-\delta})$-approximation algorithm would imply P = NP. Leonardi & Raz [47] prove similar results for $P \mid r_j \mid \sum F_j$: they give a polynomial time $O(\sqrt{n}\log n)$-approximation algorithm and an $\Omega(n^{1/3-\delta})$ inapproximability result. Hence, the non-preemptive variants of total flow time are hopelessly difficult. For the preemptive problem $P \mid pmtn, r_j \mid \sum F_j$, Leonardi & Raz [47] prove that the shortest remaining processing time rule (SRPT-rule) has a performance guarantee of $\Theta(\log n)$.

Open problem 10 *Design polynomial time approximation algorithms with constant performance guarantees for* $1 \mid pmtn, r_j \mid \sum w_j F_j$ *and for* $P \mid pmtn, r_j \mid \sum F_j$. *A good starting point might be to understand the slightly easier problem* $P2 \mid pmtn, r_j \mid \sum F_j$.

Another interesting problem is minimizing the maximum flow time. Bender, Chakrabarti & Muthukrishnan [4] show that a polynomial time greedy algorithm for $P \mid r_j \mid F_{\max}$ has performance guarantee $3 - 2/m$. It would be nice to get a PTAS for this problem.

Epilogue

Some of the open problems formulated above have been open for more than twenty years by now. Some of them may well remain open for another twenty years. Progress on any of them would be very important. We hope that this problem list will inspire and stimulate research in this area and that it will trigger several breakthroughs in the near future.

Acknowledgement. We would like to thank Chandra Chekuri, Bo Chen, Han Hoogeveen, Jan Karel Lenstra, Andreas Schulz, Jiří Sgall, David Shmoys, Martin Skutella, and Steef van de Velde for many comments and suggestions on preliminary versions of this problem list.

References

[1] J.O. Achugbue and F.Y. Chin [1982]. Scheduling the open shop to minimize mean flow time. *SIAM Journal on Computing 11*, 709–720.

[2] S. Arora and C. Lund [1997]. Hardness of approximation. In: D.S. Hochbaum (ed.) *Approximation algorithms for* NP*-hard problems*, PWS Publishing Company, Boston, 1–45.

[3] I. Bárány and T. Fiala [1982]. Nearly optimum solution of multimachine scheduling problems. *Szigma Mathematika Közgazdasági Folyóirat 15*, 177–191 (in Hungarian).

[4] M.A. Bender, S. Chakrabarti, and S. Muthukrishnan [1998]. Flow and stretch metrics for scheduling continuous job streams. *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, 270–279.

[5] J.L. Bruno, E.G. Coffman, Jr., and R. Sethi [1974]. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM 17*, 382–387.

[6] C. Chekuri and M.A. Bender [1998]. An efficient approximation algorithm for minimizing makespan on uniformly related machines. *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO'98)*, Springer LNCS 1412, 383–393.

[7] C. Chekuri, D. Karger, S. Khanna, M. Queyranne, M. Skutella, C. Stein, and M.I. Sviridenko [1998/99]. Personal communications, December 1998 – March 1999.

[8] C. Chekuri and R. Motwani [1999]. Minimizing weighted completion time on a single machine. *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, S873–S874.

[9] B. Chen and V.A. Strusevich [1993]. Approximation algorithms for three-machine open shop scheduling. *ORSA Journal on Computing 5*, 321–326.

[10] B. Chen and W. Yu [1998]. How good is a dense shop schedule? Research Paper No. 299, Warwick Business School, University of Warwick, Coventry, U.K.

[11] F.A. Chudak [1999]. A min-sum 3/2-approximation algorithm for scheduling unrelated parallel machines. *Journal of Scheduling 2*, 73–77.

[12] F.A. Chudak and D.B. Shmoys [1997]. Approximation algorithms for precedence-constrained scheduling problems on parallel machines that run at different speeds. *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, 581–590. Journal version in *Journal of Algorithms 30*, 1999, 323–343.

[13] P. Crescenzi and V. Kann [1997]. A compendium of NP-optimization problems. http://www.nada.kth.se/nada/theory/problemlist.html.

[14] J. Du, J.Y.-T. Leung, and G.H. Young [1991]. Scheduling chain-structured tasks to minimize makespan and mean flow time. *Information and Computation 92*, 219–236.

[15] U. Feige and G. Rayzman [1997]. On the drift of short schedules. *Proceedings of the 3rd Italian Conference on Algorithms and Complexity (CIAC'97)*, Springer LNCS 1203, 74–85.

[16] U. Feige and C. Scheideler [1998]. Improved bounds for acyclic job shop scheduling. *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC'98)*, 624–633.

[17] M. Fujii, T. Kasami, and K. Ninomiya [1969]. Optimal sequencing of two equivalent processors. *SIAM Journal on Applied Mathematics 17*, 784–789. Erratum: *SIAM Journal on Applied Mathematics 20*, 1971, 141.

[18] M.R. Garey and D.S. Johnson [1978]. 'Strong' NP-completeness results: Motivation, examples, and implications. *Journal of the ACM 25*, 499–508.

[19] M.R. Garey and D.S. Johnson [1979]. *Computers and Intractability: A Guide to the Theory of* NP*-Completeness*. Freeman, San Francisco.

[20] M.R. Garey, D.S. Johnson, and R. Sethi [1976]. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research 1*, 117–129.

[21] M.X. Goemans and D.P. Williamson [1999]. Two-dimensional Gantt charts and a scheduling algorithm of Lawler. *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, 366–375.

[22] L.A. Goldberg, M. Paterson, A. Srinivasan, and E. Sweedyk [1997]. Better approximation guarantees for job-shop scheduling. *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, 599–608.

[23] T. Gonzalez and S. Sahni [1976]. Open shop scheduling to minimize finish time. *Journal of the ACM 23*, 665–679.

[24] T. Gonzalez and S. Sahni [1978]. Flow shop and job shop schedules: Complexity and Approximation. *Operations Research 26*, 36–52.

[25] R.L. Graham [1966]. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal 45*, 1563–1581.

[26] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan [1979]. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics 5*, 287–326.

[27] L.A. Hall [1997]. Approximation algorithms for scheduling. In: D.S. Hochbaum (ed.) *Approximation algorithms for* NP*-hard problems*, PWS Publishing Company, Boston, 1–45.

[28] L.A. Hall [1998]. Approximability of flow shop scheduling. *Mathematical Programming 82*, 175–190.

[29] L.A. Hall, A.S. Schulz, D.B. Shmoys, and J. Wein [1997]. Scheduling to minimize average completion time: Offline and online approximation algorithms. *Mathematics of Operations Research 22*, 513–544.

[30] J. Håstad [1997]. Some optimal inapproximability results. *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC'97)*, 1–10.

[31] D.S. Hochbaum and D.B. Shmoys [1988]. A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SIAM Journal on Computing 17*, 539–551.

[32] J.A. Hoogeveen, J.K. Lenstra, and B. Veltman [1994]. Three, four, five, six, or the complexity of scheduling with communication delays. *Operations Research Letters 16*, 129–137.

[33] J.A. Hoogeveen, P. Schuurman, and G.J. Woeginger [1998]. Non-approximability results for scheduling problems with minsum criteria. *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO'98)*, Springer LNCS 1412, 353–366.

[34] W.A. Horn [1973]. Minimizing average flow time with parallel machines. *Operations Research 21*, 846–847.

[35] E. Horowitz and S. Sahni [1976]. Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM 23*, 317–327.

[36] J. Jaffe [1980]. Efficient scheduling of tasks without full use of processor resources. *Theoretical Computer Science 12*, 1–17.

[37] K. Jansen, R. Solis-Oba, and M.I. Sviridenko [1999]. Makespan minimization in job shops: A polynomial time approximation scheme. To appear in the *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC'99).*

[38] H. Kellerer, T. Tautenhahn, and G.J. Woeginger [1996]. Approximability and nonapproximability results for minimizing total flow time on a single machine. *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC'96)*, 418–426.

[39] S. Lam and R. Sethi [1977]. Worst case analysis of two scheduling algorithms. *SIAM Journal on Computing 6*, 518–536.

[40] E.L. Lawler and J. Labetoulle [1978]. On preemptive scheduling of unrelated parallel processors by linear programming. *Journal of the ACM 25*, 612–619.

[41] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys [1993]. Sequencing and scheduling: Algorithms and complexity. In: S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin (eds.) *Logistics of Production and Inventory*, Handbooks in Operations Research and Management Science 4, North-Holland, Amsterdam, 445–522.

[42] F.T. Leighton, B.M. Maggs, and S.B. Rao [1994]. Packet routing and job shop scheduling in O(Congestion+Dilation) steps. *Combinatorica 14*, 167–186.

[43] F.T. Leighton, B.M. Maggs, and A. Richa [1996]. Fast algorithms for finding O(Congestion+Dilation) packet routing schedules. Technical Report CMU-CS-96-152, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA.

[44] J.K. Lenstra and A.H.G. Rinnooy Kan [1978]. Complexity of scheduling under precedence constraints. *Operations Research 26*, 22–35.

[45] J.K. LENSTRA AND D.B. SHMOYS [1995]. Computing near-optimal schedules. In: P. Chrétienne, E.G. Coffman, Jr., J.K. Lenstra, and Z. Liu (eds.) *Scheduling Theory and its Applications*, Wiley, Chichester, 1–14.

[46] J.K. LENSTRA, D.B. SHMOYS, AND É. TARDOS [1990]. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming 46*, 259–271.

[47] S. LEONARDI AND D. RAZ [1997]. Approximating total flow time on parallel machines. *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC'97)*, 110–119.

[48] F. MARGOT, M. QUEYRANNE, AND Y. WANG [1997]. Unpublished manuscript.

[49] A. MUNIER AND C. HANEN [1995]. An approximation algorithm for scheduling unitary tasks on m processors with communication delays. Internal Report LITP 12, Université P. et M. Curie, Paris, France.

[50] A. MUNIER AND J.-C. KÖNIG [1997]. A heuristic for a scheduling problem with communication delays. *Operations Research 45*, 145–147.

[51] A. MUNIER, M. QUEYRANNE, AND A.S. SCHULZ [1998]. Approximation bounds for a general class of precedence constrained parallel machines scheduling problems. *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO'98)*, Springer LNCS 1412, 367–382.

[52] C.H. PAPADIMITRIOU AND M. YANNAKAKIS [1990]. Towards an architecture-independent analysis of parallel algorithms. *SIAM Journal on Computing 19*, 322–328.

[53] C.H. PAPADIMITRIOU AND M. YANNAKAKIS [1991]. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences 43*, 425–440.

[54] C.H. PAPADIMITRIOU AND M. YANNAKAKIS [1997]. Personal communication, May 1997.

[55] C.N. POTTS, S.V. SEVASTIANOV, V.A. STRUSEVICH, L.N. VAN WASSENHOVE, AND C.M. ZWANEVELD [1995]. The two-stage assembly scheduling problem: Complexity and approximation. *Operations Research 43*, 346–355.

[56] C.N. POTTS, D.B. SHMOYS, AND D.P. WILLIAMSON [1991]. Permutation vs. non-permutation flow shop schedules. *Operations Research Letters 10*, 281–284.

[57] H. RÖCK AND G. SCHMIDT [1983]. Machine aggregation heuristics in shop-scheduling. *Methods of Operations Research 45*, 303–314.

[58] A.S. SCHULZ AND M. SKUTELLA [1997]. Random-based scheduling: New approximations and LP lower bounds. *Proceedings of the Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'97)*, Springer LNCS 1269, 119–133.

[59] A.S. SCHULZ AND M. SKUTELLA [1997]. Scheduling-LPs bear probabilities: Randomized approximations for min-sum criteria. *Proceedings of the 5th European Symposium on Algorithms (ESA'97)*, Springer LNCS 1284, 416–429.

[60] J. Sethuraman and M.S. Squillante [1999]. Optimal scheduling of multiclass parallel machines. *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, S963–S964.

[61] S.V. Sevastianov and G.J. Woeginger [1998]. Makespan minimization in open shops: A polynomial time approximation scheme. *Mathematical Programming 82*, 191–198.

[62] D.B. Shmoys, C. Stein, and J. Wein [1994]. Improved approximation algorithms for shop scheduling problems. *SIAM Journal on Computing 23*, 617–632.

[63] M. Skutella [1998]. Semidefinite relaxations for parallel machine scheduling. *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, 472–481.

[64] M. Skutella [1999]. Convex quadratic programming relaxations for network scheduling problems. To appear in the *Proceedings of the 7th European Symposium on Algorithms (ESA'99).*

[65] M. Skutella and G.J. Woeginger [1999]. A PTAS for minimizing the weighted sum of job completion times on parallel machines. To appear in the *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC'99).*

[66] J.D. Ullman [1976]. Complexity of sequencing problems. In: E.G. Coffman, Jr. (ed.) *Computer and Job-Shop Scheduling Theory*, Wiley, New York, 139–164.

[67] B. Veltman, B.J. Lageweg, and J.K. Lenstra [1990]. Multiprocessor scheduling with communication delays, *Parallel Computing 16*, 173–182.

[68] D.P. Williamson, L.A. Hall, J.A. Hoogeveen, C.A.J. Hurkens, J.K. Lenstra, S.V. Sevastianov, and D.B. Shmoys [1997]. Short shop schedules. *Operations Research 45*, 288–294.

[69] G.J. Woeginger [1999]. When does a dynamic programming formulation guarantee the existence of an FPTAS? *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, 820–829.