Journal of Computer and System Sciences ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

On the optimality of exact and approximation algorithms for scheduling problems ☆,☆☆

Lin Chen [a,*], Klaus Jansen [b], Guochuan Zhang [a]

[a] College of Computer Science, Zhejiang University, Hangzhou, 310027, China
[b] Department of Computer Science, Kiel University, 24098 Kiel, Germany

ARTICLE INFO

Article history:
Received 24 April 2015
Received in revised form 13 August 2016
Accepted 26 March 2018
Available online xxxx

Keywords:
Approximation schemes
Scheduling
Lower bounds
Exponential time hypothesis

ABSTRACT

We consider the classical scheduling problem on parallel identical machines to minimize the makespan. There is a long history of study on this problem, focusing on exact and approximation algorithms. It is thus natural to consider whether these algorithms can be further improved in terms of running times. We provide strong lower bounds on the running times of exact and approximation schemes for the classical scheduling problem based on Exponential Time Hypothesis, showing that the best known approximation and exact algorithms are almost the best possible in terms of the running time.

© 2018 Published by Elsevier Inc.

1. Introduction

Computational complexity theory allows us to rule out polynomial time algorithms for many fundamental optimization problems under the assumption $P \neq NP$. On the other hand, however, this does not give us (non-polynomial) lower bounds on the running time for such algorithms. For example, under the assumption $P \neq NP$, there could still be an algorithm with running time $n^{O(\log n)}$ for 3SAT or BIN PACKING. A stronger assumption, the Exponential Time Hypothesis (ETH), was introduced by Impagliazzo, Paturi, and Zane [12]:

Exponential Time Hypothesis (ETH): There is a positive real δ such that 3SAT with n variables and m clauses cannot be solved in time $2^{\delta n}(n+m)^{O(1)}$.

Using the Sparsification Lemma by Impagliazzo et al. [3], ETH implies that there is no algorithm for 3SAT with n variables and m clauses that runs in time $2^{\delta m}(n+m)^{O(1)}$ for some $\delta > 0$ as well. Under ETH, lower bounds on the running time for several graph problems have been obtained. For example, there is no $2^{\delta n}$ time algorithm for 3-COLORING, INDEPENDENT SET, VERTEX COVER, and HAMILTONIAN PATH unless ETH fails, where n is the number of vertices in the given problem instance. An essential property of the underlying strong reductions to show these lower bounds is that the main parameter, the number of vertices, is increased only linearly. These lower bounds together with the fact that there exist algorithms of running time $2^{O(n)}$ [9] give us some evidence that ETH is true, i.e., a subexponential time algorithm for 3SAT is unlikely to exist. For a nice survey about lower bounds via ETH we refer to Lokshtanov et al. [22]. Interestingly, using ETH one can also prove lower

☆ Part of the work was done when the first author was visiting Kiel University. Supported in part by Chinese Scholarship Council, NSFC (11271325) and the DFG, project, Laufzeitschranken für Scheduling- und Packungsprobleme unter Annahme der Exponentialzeithypothese, JA 612/16-1.
☆☆ An extended abstract appeared in SODA 2014.
* Corresponding author.
E-mail addresses: chenlin198662@gmail.com (L. Chen), kj@informatik.uni-kiel.de (K. Jansen), zgc@zju.edu.cn (G. Zhang).

https://doi.org/10.1016/j.jcss.2018.03.005
0022-0000/© 2018 Published by Elsevier Inc.

bounds on the running time of approximation schemes. For example, Marx [23] proved that there is no PTAS of running time $2^{O((1/\epsilon)^{1-\delta})}n^{O(1)}$ for MAXIMUM INDEPENDENT SET on planar graphs.

In this paper, we consider the classical scheduling problem of jobs on identical machines with the objective of minimizing the makespan, i.e., the largest job completion time. Formally, an instance I is given by a set $\mathcal{M}$ of m identical machines and a set $\mathcal{J}$ of n jobs with processing times p_j. The objective is to compute a non-preemptive schedule or an assignment $a : \mathcal{J} \to \mathcal{M}$ such that each job is executed by exactly one machine and the maximum load $\max_{i=1,\ldots,m} \sum_{j:a(j)=i} p_j$ among all machines is minimized. In scheduling theory, this problem is denoted by $Pm||C_{max}$ if m is a constant or $P||C_{max}$ if m is an arbitrary input.

This problem is NP-hard even if $m = 2$, and is strongly NP-hard if m is an input. On the positive side, it is known that for any $\epsilon > 0$, a $(1+\epsilon)$-approximation algorithm exists for $Pm||C_{max}$ [11] as well as $P||C_{max}$ [10]. As a consequence, there is a long history of improvements on the running time of such algorithms. We give a brief introduction as follows.

In 1976, Horowitz and Sahni [11] presented an FPTAS (fully polynomial-time approximation scheme) for a more generalized scheduling model $Rm||C_{max}$, where each job could have different running times on different machines. The running time of this algorithm is $O(nm(nm/\epsilon)^{m-1})$ and it was improved by Shmoys and Tardos [20] to $(n+1)^{m/\epsilon}poly(|I|)$, where $poly(|I|)$ denotes some polynomial of the input length $|I|$. In 2001, Jansen and Porkolab [17] designed an FPTAS of running time $O(n(m/\epsilon)^{O(m)})$. For sufficiently small ϵ (e.g., $1/\epsilon > m$), the running time could be further improved to $O(n)+(1/\epsilon)^{O(m)}$ [16]. All the above mentioned algorithms keep a high dimensional vector through dynamic programming and have a running time exponentially dependent on $O(m)$. The running time thus increases drastically as m increases, which is also known as the curse of dimensionality. It is not yet known that whether an FPTAS of running time $(1/\epsilon)^{O(m^{1-\delta})}poly(|I|)$ exists even for the much simpler problem $Pm||C_{max}$, where δ is some positive real number.

For $P||C_{max}$, Hochbaum and Shmoys [10] gave a PTAS (polynomial-time approximation scheme) of running time $O((n/\epsilon)^{1/\epsilon^2})$. It was later improved by Leung to $O((n/\epsilon)^{1/\epsilon\log(1/\epsilon)})$. In 1998, Alon et al. [1] presented an EPTAS (efficient polynomial-time approximation scheme) of running time $f(1/\epsilon)+O(n)$ where $f(1/\epsilon)$ is doubly exponential in $1/\epsilon$. In 2010, Jansen [13] gave an EPTAS of running time $2^{O(1/\epsilon^2\log^3(1/\epsilon))}+O(n\log n)$. Very recently, Jansen, Klein and Verschae [15] provided an EPTAS of running time $2^{O(1/\epsilon\log^{O(1)}(1/\epsilon))}+O(n\log n)$.

Exact algorithms for the scheduling problem are also under extensive research. Recently Lente et al. [21] provided algorithms of running time $2^{n/2}$ and $3^{n/2}$ for $P2||C_{max}$ and $P3||C_{max}$, respectively. O'Neil [25] gave a sub-exponential time algorithm of running time $2^{O(m\sqrt{|I|})}$ for the bin packing problem with m bins where $|I|$ is the length of the input. This algorithm, designed for the bin packing problem, works also for $Pm||C_{max}$. FPT algorithms for $P||C_{max}$ or $R||C_{max}$, parameterized by different parameters, were studied very recently by Mnich and Wiese [24], Knop and Koutecký [18], Chen et al. [6].

On the other hand, there are only few lower bounds known for scheduling and packing problems. Chen et al. [4] showed that precedence constrained scheduling on m machines cannot be solved in time $f(m)|I|^{o(m)}$ (where $|I|$ is the length of the instance) for any recursive function f, under the assumption that the parameterized complexity classes FPT and $W[1]$ are not equal. Under the same assumption, Kulik and Shachnai [19] proved that there is no PTAS for the 2D KNAPSACK problem with running time $f(\epsilon)|I|^{O(1)}$ for any recursive function f. We remark that it is proved by Downey and Fellows [7] that ETH implies the assumption that FPT and $W[1]$ are not equal, hence all the above-mentioned lower bounds hold under ETH. Patrascu and Williams [26] proved using ETH a lower bound of $n^{o(k)}$ for the k-SUM problem, where the k-SUM problem asks whether a set of n numbers contains a subset of k numbers that sum to 0. Recently, Jansen et al. [14] showed a lower bound of $2^{o(n)}|I|^{O(1)}$ for the PARTITION problem and proved that there is no PTAS for the MULTIPLE KNAPSACK problem and 2D KNAPSACK problem with running time $2^{o(1/\epsilon)}|I|^{O(1)}$ and $n^{o(1/\epsilon)}|I|^{O(1)}$, respectively. Chen et al. [6] showed lower bounds for $R||C_{max}$ with respect to the parameter which is the rank of the matrix formed by job processing times.

The main contribution of this paper is to characterize lower bounds on the running times of exact and approximation algorithms for the classical scheduling problem.

Theorem 1. *For any $\delta > 0$, there is no $2^{O((1/\epsilon)^{1-\delta})}+n^{O(1)}$ time PTAS for $P||C_{max}$, unless ETH fails.*

Theorem 2. *For any $\delta > 0$, there is no $2^{O(n^{1-\delta})}$ time exact algorithm for $P||C_{max}$ with n jobs even if we restrict that the processing time of each job is bounded by $O(n)$, unless ETH fails.*

Theorem 3. *For any $\delta > 0$, there is no $(1/\epsilon)^{O(m^{1-\delta})}+n^{O(1)}$ time FPTAS for $Pm||C_{max}$, unless ETH fails.*

Theorem 4. *For any $\delta > 0$, there is no $2^{O(m^{1/2-\delta}\sqrt{|I|})}$ time exact algorithm for $Pm||C_{max}$, unless ETH fails, where $|I|$ is the length of the input.*

We also prove that the traditional dynamic programming algorithm for the scheduling problem runs in $2^{O(\sqrt{m|I|\log^2 m})}$ time, and is thus essentially the best possible exact algorithm in terms of running time. An overview about the known and new results is in Table 1.

Our results imply that the existing exact and approximation algorithms for the scheduling problem are essentially the best possible. Indeed, when the conference version of this paper was published, the best PTAS for $P||C_{max}$ was

Please cite this article in press as: L. Chen et al., On the optimality of exact and approximation algorithms for scheduling problems, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.03.005

Download English Version:

https://daneshyari.com/en/article/6874663

Download Persian Version:

https://daneshyari.com/article/6874663

Daneshyari.com