Available online at www.sciencedirect.com
SCIENCE DIRECT®
Operations Research Letters 33 (2005) 127–133
Operations Research Letters
www.elsevier.com/locate/dsw

An optimal rounding gives a better approximation for scheduling unrelated machines

Evgeny V. Shchepin[a,1], Nodari Vakhania[b,c,*,2]

[a] *Steklov Mathematical Institute 117966, Gubkina 8, Moscow, Russia*
[b] *Science Faculty, State University of Morelos, Av. Universidad 1001, Cuernavaca 62210, Morelos, Mexico*
[c] *Institute of Computational Mathematics, Akuri 8, Tbilisi 93, GA, USA*

Received 19 March 2001; accepted 20 May 2004

Abstract

A polynomial-time algorithm is suggested for non-preemptive scheduling of n-independent jobs on m-unrelated machines to minimize the makespan. The algorithm has a worst-case performance ratio of $2 - 1/m$. This is better than the earlier known best performance bound 2. Our approach gives the best possible approximation ratio that can be achieved using the rounding approach.

Keywords: Scheduling; Unrelated machines; Approximation algorithm; Rounding

1. Introduction

The problem of constructing a non-preemptive schedule of the minimal makespan (maximal job or machine completion time) on m-identical machines, $P//C_{\max}$ is NP-hard even when $m = 2$ and the jobs are independent (i.e have no precedence relations). Therefore, no polynomial algorithm can give an optimal schedule for $m \geqslant 2$ machines unless $P = NP$. We try to approximate the optimum in polynomial time. For a given schedule, its *performance ratio* is the ratio of the makespan of this schedule to the optimal makespan. An algorithm with a worst-case performance ratio κ is called a κ-*approximation algorithm.*

Due to Graham's well-known result [4], a linear on-line list-scheduling algorithm for $P/prec/C_{\max}$ (the version of $P//C_{\max}$ with precedence relations) gives a worst-case performance ratio $2 - 1/m$. An $O(n \log n)$ MULTIFIT algorithm for $P//C_{\max}$ gives performance ratio $\frac{13}{11}$ for identical machines and $\frac{7}{5}$ for uniform machines (see [2,3,11]). Hochbaum and Shmoys [5] suggest polynomial-approximation schemes for uniform machines (a family of polynomial algorithms with performance ratio arbitrarily close to 1).

The situation is much more complicated for unrelated machines. The first approximation scheme proposed by Horowitz and Sahni [6] is polynomial in n but non-polynomial in m. This algorithm with performance ratio $1 + \varepsilon$ has time complexity $O(n^{2m}/\varepsilon)$ and its space complexity is non-polynomial. The first

* Corresponding author. Science Faculty, State University of Morelos, Av. Universidad 1001, Cuernavaca 62210, Morelos, Mexico.
E-mail addresses: scepin@mian.ras.su (E.V. Shchepin), nodari@servm.fc.uaem.mx (N. Vakhania).
[1] Partially supported by the program "Algebraical and combinatorial methods of mathematical cybernetics" of the Russian Academy of Sciences?
[2] Partially supported by CONACyT Grant 28937A.

0167-6377/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2004.05.004

polynomial-time approximation algorithm for $R//C_{\max}$ was proposed by Ibarra and Kim [7] with performance ratio m, and a better polynomial-time algorithm with performance ratio within $2\sqrt{m}$ was suggested by Davis and Jaffe [1]. Potts [9] gave an approximation algorithm with performance ratio 2. He used a relaxed linear program in which boolean assignment variables are replaced by real variables from the interval [0,1]. A solution of this linear program is a preemptive distribution: unlike a schedule, it involves no start times and only assigns jobs to machines. Such distributions may split a job into parts and assign these parts to different machines (as a result of the real assignment variables). These distributions have no more than $m-1$ preempted jobs. Then a rounding of real variables to boolean variables is applied. All possible assignments of the preempted $m-1$ (or less) jobs to m machines are tested by a complete enumeration. This imposes an exponential dependence on m. Since the initial distribution is optimal and the preempted $m-1$ (or less) jobs are also scheduled optimally, the makespan of the resulting non-preemptive schedule must be no larger than twice the makespan of an optimal one.

Lenstra et al. [8] developed Potts' rounding approach [9] avoiding complete enumeration of preempted jobs and hence exponential dependence on m. Their algorithm (polynomial in both n and m) has the same performance ratio as the algorithm in [9]. The linear program used in [8] is different from that of in [9], it imposes additional artificial constraints on the problem. Let a *B-balanced distribution* be a distribution of makespan at most B, such that every job assigned to a machine has a processing time of at most B on that machine. B-balanced distributions allow a polynomial 2-approximation rounding in the following way. It can be checked in polynomial time whether there exists a feasible B-balanced distribution or not, for any B (one part of this task reduces to a complete matching problem in a bipartite graph with binary search, and another to linear programming). Let D_{opt} be the optimal non-preemptive makespan. We can easily find lower and upper bounds on D_{opt} and apply binary search in this interval. In other words, we can find the minimal $B = B_{\text{opt}}$, such that there exists a feasible B-balanced distribution. By this construction, D_{opt} cannot be less than B_{opt}, since no job in an optimal non-preemptive schedule can be assigned to a machine on which it takes more than D_{opt} time, and B_{opt} is the minimal B with this kind of restriction. The resulting B_{opt}-balanced distribution δ can have at most m preempted jobs. Then the rounding of these jobs is applied. The makespan of the resulting non-preemptive schedule will be at most $2B_{\text{opt}} \leqslant 2D_{\text{opt}}$.

In [8] it is also shown that no polynomial algorithm with performance ratio 1.5 or less can exist unless $P = NP$, leaving open the question whether there can exist a polynomial-approximation algorithm with a worst-case ratio between 1.5 and 2 (for $m = 2$, an 1.5-approximation linear-time algorithm is given in [9]). In the present paper, we make the first step forward towards this direction suggesting an $(2 - 1/m)$ polynomial-approximation algorithm. This result is the best-possible for any rounding-based algorithm, as no such an algorithm can give a better worst-case ratio (Section 4).

We apply an approach, similar to that of Lenstra et al. [8] building a B_{opt}-balanced distribution. Our B_{opt}-balanced distribution δ has at most $m-1$ preemptions and we eliminate them by rounding not more than $m-1$ preempted jobs. A job cannot be rounded on a machine unless a fraction of at least $1/m$ of that job was originally assigned to it in δ. We give an $O(m^2)$ algorithm constructing this type of rounding with no machine receiving more than $(m-1)/m$ new job parts. This ensures that the resulting makespan will exceed the makespan of the initial B_{opt}-balanced distribution by at most $(m-1)/m\, p^{\delta}_{\max}$, $p^{\delta}_{\max}$ being the maximal job-processing time in δ. Since $p^{\delta}_{\max} \leqslant B_{\text{opt}}$, a $(2 - 1/m)$-approximation to the optimum is guaranteed. In the preliminary form, our result has appeared in the proceedings of IFIP Intern. Conference in Computer Science [10].

2. Basic notions and terminology

2.1. Schedules and distributions

The sets of *jobs* and *machines* are denoted by $\mathscr{J} = \{J^1, \ldots, J^n\}$ and $\mathscr{M} = \{M_1, \ldots, M_m\}$, respectively. $M(J)$ is the *processing time* (length) of job J on machine M (this uncommon notation avoids the use of an additional symbol). A (non-preemptive) *schedule* σ is a mapping which assigns each job to a machine and a starting time on that machine; every