Theoretical Computer Science 600 (2015) 70–85

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Improved approximation algorithms for scheduling parallel jobs on identical clusters

Marin Bougeret [a,*], Pierre-Francois Dutot [b], Denis Trystram [b], Klaus Jansen [c], Christina Robenek [c]

[a] LIRMM, Université Montpellier 2, 34095 Montpellier, France
[b] LIG, Laboratoire d'Informatique de Grenoble, 38334 Saint Ismier Cedex, France
[c] Department of Computer Science, Universität Kiel, Christian-Albrechts-Platz 4, 24118 Kiel, Germany

ARTICLE INFO

Article history:
Received 15 April 2015
Received in revised form 10 June 2015
Accepted 4 July 2015
Available online 9 July 2015
Communicated by D.-Z. Du

Keywords:
Scheduling
Parallel job
Strip packing
Approximation algorithm

ABSTRACT

The Multiple Cluster Scheduling Problem corresponds to minimizing the maximum completion time (makespan) of a set of n parallel rigid (and non-preemptive) jobs submitted to N identical clusters. It cannot be approximated with a ratio better than 2 (unless $\mathcal{P} = \mathcal{NP}$). We present in this paper the methodology that encompasses several existing results [1,2]. We detail first how to apply it for obtaining a $\frac{5}{2}$-approximation. Then, we use it to provide a new $\frac{7}{3}$-approximation running in $\mathcal{O}(\log(nh_{max})N(n+\log(n)))$, where h_{max} is the processing time of the longest job. Finally, we apply it to a restriction of the problem to jobs of limited size, leading to a 2-approximation which is the best possible ratio since the restriction remains 2-inapproximable.

1. Introduction

1.1. Problem statement

In the grid computing paradigm, several clusters share their computing resources in order to better distribute the workload. Each cluster is composed of a set of identical processors connected by a fast local interconnection network [3]. Jobs are submitted over time in successive packets called batches. The objective is to minimize the time when all the jobs of a batch are completed, thus, minimizing the date when the next batch of jobs can be processed. Many such computational grid systems are available all over the world, and the efficient management of the resources is known to be one of the most important challenges today. Let us start by stating the corresponding Multiple Cluster Scheduling Problem (MCSP) more formally.

Definition 1 *(MCSP).* We are given n parallel rigid non-preemptive jobs J_j, $1 \le j \le n$, and N clusters. A job J_j requires q_j processors during p_j units of time, and each cluster owns m identical processors. The objective is to schedule all the jobs in the clusters, minimizing the maximum completion time (makespan). The constraints are:

1. the q_j processors allocated to job J_j must belong to the same cluster

* Corresponding author.
E-mail address: marin.bougeret@lirmm.fr (M. Bougeret).

http://dx.doi.org/10.1016/j.tcs.2015.07.003
0304-3975/

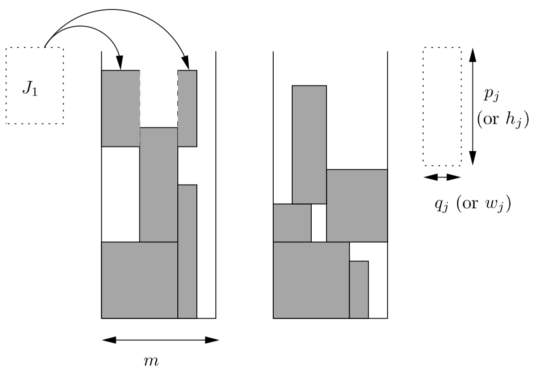

Fig. 1. Example (for $n = 9$ jobs and $N = 2$ clusters) of a solution that is feasible for MCSP and not feasible for MSPP. Notice there hat J_1 is packed in a "non-continuous" way (using non-consecutive indexes of processors).

Problem	Ratio	Remarks	Source
MCSP, MSPP	$2 + \epsilon$	Need solving $P\|\|C_{max}$ with a ratio $1 + \frac{\epsilon}{2}$	[7]
MCSP	5/2	Fast algorithm	[1]
MSPP	2	Costly algorithm (at least $\Omega(n^{256})$)	[6]
MCSP, MSPP	AFPTAS	Additive constant in $\mathcal{O}(\frac{1}{\epsilon^2})$, and in $\mathcal{O}(1)$ for large values of N	[6]
MCSP	3	Fast (and decentralized) algorithm that handles clusters of different sizes	[5]
MCSP	7/3	Fast algorithm	This paper
MCSP	2	Requires $\mathbf{max_j w_j \le \frac{1}{2}}$ Fast algorithm	This paper, from [2]

Fig. 2. Summary of existing results.

2. at any time, the total number of used processors in any cluster must be lower or equal to m.

This problem is closely related to the following Multiple Strip Packing problem (MSPP).

Definition 2 *(MSPP).* We are given n rectangles r_j, $1 \le j \le n$, and N strips. Rectangle r_j have height h_j and width w_j, and all the strips have width 1. The objective is to pack all the rectangles in the strips such that the maximum reached height is minimized under the following constraints.

1. a rectangle must be entirely packed into a strip (it cannot be split between two strips)
2. at any level of any strips, the total width of packed rectangles must be lower or equal to 1
3. a rectangle must be allocated "contiguously".

Thus, the only difference between MCSP and MSPP is constraint 3), which in term of job scheduling amounts to force the jobs to use consecutive indexes of processors (see Fig. 1). Of course, the results for MCSP generally do not apply to MSPP because of the additional contiguous constraints. The converse is also not clear, since the approximation ratios for MSPP may not be preserved when considering MCSP. However, as we can notice in Fig. 2, many results for MSPP directly apply to MCSP, as the proposed algorithms build contiguous schedules that are compared to non-contiguous optimal solutions. In this paper, the studied problem (MCSP) is seen as MSPP without constraint 3), and from now on we use the vocabulary and notations of packing.

1.2. Related work

As shown in [4] using a gap reduction from the Partition problem, MCSP (and MSPP) are 2-inapproximable in polynomial time unless $\mathcal{P} = \mathcal{NP}$, even for $N = 2$. The main positive results for MCSP are summarized in Fig. 2. For the sake of readability, we call "fast algorithm" algorithms with a running time in $\mathcal{O}(n^p)$, with $p \le 3$ (the exact complexity of these algorithms is not relevant here).

We must distinguish the 3-approximation in [5] and the $\frac{5}{2}$-approximation in [1] that have a low cost from the costly 2-approximation in [6] and $2 + \epsilon$-approximation in [7].

The 2-approximation does not apply to *MCSP*. Moreover, it is directly obtained from asymptotic approximation algorithms when the number of strips is larger than a constant N_0, but requires algorithms that are exponential in N_0 when the number of strips is lower than N_0. Thus, the value of this constant ($\approx 10^4$) makes this algorithm impossible to use for real size instances.