Information Processing Letters 112 (2012) 762–766

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Batch scheduling of identical jobs on parallel identical machines

Baruch Mor [a,b], Gur Mosheiov [b,*]

[a] The Department of Economics and Business Administration, Ariel University Center of Samaria, Israel
[b] School of Business Administration, The Hebrew University, Jerusalem, Israel

ARTICLE INFO

Article history:
Received 4 July 2011
Received in revised form 22 June 2012
Accepted 24 June 2012
Available online 4 July 2012
Communicated by Jinhui Xu

Keywords:
Scheduling
Batch scheduling
Parallel identical machines
Flowtime
Setup times

ABSTRACT

We study the classical batch scheduling problem with identical job processing times and identical setups on parallel identical machines. We show that, similar to the single machine case, the solution is given by a closed form, consisting of identical decreasing arithmetic sequences of batch sizes on the different machines. A very close-to-optimal integer solution is obtained in $O(m+\sqrt{n})$ time, where m is the number of machines, and n is the number of jobs.

1. Introduction

The classical batch scheduling problem, where n identical jobs need to be processed on a single machine, and the objective function is minimum flowtime, was solved twenty five years ago (Santos and Magazine [1]). The closed form solution consists of an expression for the number of batches, and of a decreasing arithmetic sequence of the batch sizes. Recently, Mor and Mosheiov [2] extended these results to the setting of a two uniform machines. Their algorithm runs in $O(n)$ time, where n is the number of jobs. In this note we focus on the setting of m *parallel identical* machines. We show that similar to the single machine case, an optimal solution (the optimal number of batches on each machine and their sizes) is given in a closed form.

These results refer to the "relaxed" version of the problem, i.e. when non-integer batch sizes are allowed. Shallcross [3] and Mosheiov et al. [4] introduced solutions for the integer version of the single machine problem. We introduce here an efficient $O(m+\sqrt{n})$ approximation procedure for the case of parallel identical machines. [It should be noted that this procedure is not polynomial in the input size, which consists of four values: number of jobs, number of machines, jobs' identical processing time, and the constant setup time; see below.] Our proposed procedure is shown numerically to produce extremely close-to-optimal schedules (with average optimality gaps of no more than 0.02%).

Formally, n identical jobs (with p denoting their common processing time), need to be processed on m parallel identical machines. These jobs can be processed in batches, and for a given job allocation to batches, we denote by K_i the number of batches assigned to machine i, $i=1,\dots,m$. Let $n_{i,j}$, $i=1,\dots,m$, $j=1,\dots,K_i$, denote the number of jobs assigned to batch j on machine i. Let $n_i=\sum_{j=1}^{K_i} n_{i,j}$ denote the total load assigned to machine i, $i=1,\dots,m$. (Clearly $n=\sum_{i=1}^{m} n_i=\sum_{i=1}^{m}\sum_{j=1}^{K_i} n_{i,j}$.)

Following Santos and Magazine [1], we assume that when starting a new batch, a constant setup time, denoted by s is incurred. Note that the setup time is both machine- and batch-independent. We also assume (as in Santos and Magazine) *batch availability*, i.e. the completion time of a job is defined as the completion time of the entire batch to which it is assigned.

* Corresponding author.
E-mail address: msomer@mscc.huji.ac.il (G. Mosheiov).

0020-0190/$ – see front matter
http://dx.doi.org/10.1016/j.ipl.2012.06.020

Let $C_{i,j}$ denote the completion time of batch j on machine i, $i = 1, \dots, m$, $j = 1, \dots, K_i$. The objective function is the minimum sum of job completion times. Due to the property of batch availability, the contribution of batch j to the objective function is $n_{i,j}C_{i,j}$. The sum of job completion times is given by $\sum C \equiv \sum_{i=1}^{m}\sum_{j=1}^{K_i} n_{i,j}C_{i,j}$. Thus, the problem solved in this note is $Pm/batch, s, p_j = p/\sum C$.

2. A closed form solution for the non-integer problem

We focus first on the "relaxed" version of the problem, where, as mentioned, batch sizes are not forced to be integers. For convenience we assume unit processing times, i.e. $p = 1$. In this case, the sum of the job completion times is given by

$$f = \sum_{i=1}^{m}\Bigg[(s + n_{i,1})n_{i,1} + (2s + n_{i,1} + n_{i,2})n_{i,2} + \cdots + \Bigg((K_i - 1)s + \sum_{j=1}^{K_i-1} n_{i,j}\Bigg)n_{i,K_{i-1}} + \Bigg(K_i s + \sum_{j=1}^{K_i} n_{i,j}\Bigg)n_{i,K_i}\Bigg]$$
$$= \sum_{i=1}^{m}\Bigg[\sum_{j=1}^{K_i}\Bigg(\sum_{l=1}^{j} n_{i,l}\Bigg)n_{i,j} + s\sum_{j=1}^{K_i} j n_{i,j}\Bigg].$$

It is easy to verify the following equality:

$$\sum_{j=1}^{K}\Bigg(\sum_{i=1}^{j} n_i\Bigg)n_j = \frac{1}{2}\sum_{j=1}^{K} n_j^2 + \frac{1}{2}\Bigg(\sum_{j=1}^{K} n_j\Bigg)^2.$$

The objective function can be written as

$$f = \sum_{i=1}^{m}\Bigg[\frac{1}{2}\sum_{j=1}^{K_i} n_{i,j}^2 + \frac{1}{2}\Bigg(\sum_{j=1}^{K_i} n_{i,j}\Bigg)^2 + s\sum_{j=1}^{K_i} j n_{i,j}\Bigg]. \quad (1)$$

Thus, we have to solve the following problem:

$$\text{Min } f$$
$$\text{s.t.} \quad \sum_{i=1}^{m}\sum_{j=1}^{K_i} n_{i,j} = n,$$
$$n_{i,j} \geqslant 0, \quad i = 1, \dots, m, \ j = 1, \dots, K_i.$$

Note that f is a quadratic convex function of the batch sizes. The global minimum can be found by applying the Karush–Kuhn–Tucker (KKT) conditions. The Lagrangian, with a single Lagrange multiplier λ, is

$$L = f - \lambda\Bigg(\sum_{i=1}^{m}\sum_{j=1}^{K_i} n_{i,j} - n\Bigg).$$

The KKT conditions are

$$\frac{\partial L}{\partial n_{i,j}} = n_{i,j} + \sum_{j=1}^{K_i} n_{i,j} + js - \lambda = 0,$$
$$j = 1, \dots, K_i, \ i = 1, \dots, m, \quad (2)$$

$$\lambda\Bigg(\sum_{i=1}^{m}\sum_{j=1}^{K_i} n_{i,j} - n\Bigg) = 0, \quad (3)$$

$$n_{i,j} \geqslant 0, \quad i = 1, \dots, m, \ j = 1, \dots, K_i. \quad (4)$$

From (2) we note that on each machine the batch sizes follow a decreasing arithmetic sequence, with s being the constant difference:

$$n_{i,j} = n_{i,1} - (j - 1)s, \quad i = 1, \dots, m, \ j = 2, \dots, K_i. \quad (5)$$

The sum of each arithmetic sequence is given by

$$\sum_{j=1}^{K_i} n_{i,j} = \frac{K_i}{2}\big(2n_{i,1} - (K_i - 1)s\big)$$
$$= K_i n_{i,1} - \frac{1}{2}sK_i(K_i - 1), \quad i = 1, \dots, m. \quad (6)$$

It follows from (2) that $\lambda > 0$. Thus, from (3) we must have $\sum_{i=1}^{m}\sum_{j=1}^{K_i} n_{i,j} = n$.

We consider a symmetric solution, i.e. identical load allocation among the machines:

$$\sum_{j=1}^{K_1} n_{1,j} = \sum_{j=1}^{K_2} n_{2,j} = \cdots = \sum_{j=1}^{K_m} n_{m,j}.$$

It follows that

$$n/m = \sum_{j=1}^{K_i} n_{i,j}, \quad i = 1, \dots, m. \quad (7)$$

[Below we show that this symmetric solution satisfies all KKT conditions, implying that indeed it is globally optimal. However, symmetry, i.e. equal load allocation, also follows from the convexity of the objective function (1).]

In order to obtain the size of the first batch on each machine, we equate (6) and (7):

$$n_{i,1} = \frac{n}{mK_i} + \frac{1}{2}s(K_i - 1), \quad i = 1, \dots, m. \quad (8)$$

Clearly, the last term in the sequence must be strictly positive. Therefore

$$n_{i,K_i} = n_{i,1} - (K_i - 1)s > 0, \quad i = 1, \dots, m. \quad (9)$$

Substituting $n_{i,1}$ (from (8)), we get

$$msK_i^2 - msK_i - 2n < 0, \quad i = 1, \dots, m. \quad (10)$$

When solving this quadratic inequality, we obtain that the largest positive value of K_i ($i = 1, \dots, m$) that satisfies (10) is

$$K_i = \Bigg\lfloor\sqrt{\frac{1}{4} + \frac{2n}{ms}} + \frac{1}{2}\Bigg\rfloor, \quad i = 1, \dots, m.$$

The latter is the (identical) number of batches assigned to each machine.