Computers & Operations Research 73 (2016) 84–91

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A fast and effective subset sum based improvement procedure for workload balancing on identical parallel machines

Stefan Schwerdfeger [a], Rico Walter [b,*]

[a] Friedrich Schiller University Jena, Chair for Management Science, Carl-Zeiß-Straße 3, D-07743 Jena, Germany
[b] Fraunhofer Institute for Industrial Mathematics ITWM, Department of Optimization, Fraunhofer-Platz 1, D-67663 Kaiserslautern, Germany

ARTICLE INFO

Article history:
Received 14 September 2015
Received in revised form
30 November 2015
Accepted 16 March 2016
Available online 12 April 2016

Keywords:
Scheduling
Identical parallel machines
Workload balancing
NSSWD-criterion
Heuristics

ABSTRACT

The paper on hand addresses the workload balancing problem that asks for an assignment of n independent jobs to m identical parallel machines so that the normalized sum of squared workload deviations (*NSSWD*-criterion) is minimized. For the special case of $m=3$ machines we propose an exact algorithm that requires solving a sequence of subset sum problems. This algorithm also builds the core of our local search procedure for solving the general case of $m \geq 3$ machines. The main innovation of our approach compared to existing methods, therefore, consists in using triples of machines as a neighborhood instead of pairs of machines. Results of a comprehensive computational study on the benchmark library established by Ho et al. [10] and Cossari et al. [5] attest to the effectiveness of our approach. In addition, we demonstrate its capability to determine high-quality solutions for further balancing criteria as discussed in Cossari et al. [6].

1. Introduction

1.1. Problem definition

In this paper we investigate the following workload balancing problem. Given a set $\mathcal{I}$ of $m \geq 2$ identical parallel machines and a set $\mathcal{J}$ of $n > m$ independent jobs with integer processing times $p_j \in \mathbb{N}$ ($j = 1, \ldots, n$), the objective is to minimize the normalized sum of squared workload deviations (*NSSWD*). Using the standard three-field notation, this balancing problem is abbreviated as $P \| NSSWD$. The *NSSWD*-criterion has been introduced by Ho et al. [10] and is defined as

$$NSSWD = \frac{1}{\mu}\left[\sum_{i=1}^{m}(C_i - \mu)^2\right]^{1/2} \tag{1}$$

where C_i denotes the completion time of machine i ($i = 1, \ldots, m$) and $\mu = \sum_{j=1}^{n} p_j/m$ represents the average machine completion time. Introducing binary variables x_{ij} which take the value 1 if job j is assigned to machine i and 0 otherwise, a straightforward formulation of $P \| NSSWD$ as an integer linear program consisting of objective function (2) subject to (3)–(5) is provided below:

$$\text{Minimize} \quad z = \frac{1}{\mu}\left[\sum_{i=1}^{m}(C_i - \mu)^2\right]^{1/2} \tag{2}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} p_j \cdot x_{ij} = C_i \quad i = 1, \ldots, m \tag{3}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n \tag{4}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \ldots, m; \; j = 1, \ldots, n \tag{5}$$

Objective function (2) minimizes the *NSSWD*-value. The machine completion times are determined by (3). Constraints (4) ensure that each job is assigned to exactly one machine. Finally, the domains of the binary variables are set by (5). By a straightforward reduction from Partition it is readily verified that problem $P \| NSSWD$ is $\mathcal{NP}$-hard. Throughout the paper we assume without loss of generality the jobs to be labeled so that $p_1 \geq p_2 \geq \cdots \geq p_n > 0$.

Balancing the workload among a given set of machines or workers encompasses several practical issues. If we think of human workers, distributing tasks among them as equally as possible does not only comprise aspects of fairness and equitable payment, it also helps to prevent the employees from being dissatisfied and unmotivated [6]. Aside from that, an even distribution of workload leads to an efficient utilization of resources [10] and reduces idle times as well as work in process [16]. In a wider sense, further

* Corresponding author.
E-mail addresses: stefan.schwerdfeger@uni-jena.de (S. Schwerdfeger), rico.walter@gmail.com, rico.walter@itwm.fraunhofer.de (R. Walter).

http://dx.doi.org/10.1016/j.cor.2016.03.008

applications arise in the context of packing problems where one aims at an equal loading of bins (cf. [10]).

1.2. Literature review

Due to its practical relevance and its various facets, workload balancing has attracted many researchers' interest. Consequently, in the scheduling literature we can find a great variety of contributions to balancing in general or to problems that incorporate balancing aspects – at least implicitly. So, in a wider sense, also the classical makespan minimization problem $P \parallel C_{max}$ or its dual, i.e. the machine covering problem $P \parallel C_{min}$ (see e.g. [21,9,18,20]) can be understood as very basic balancing problems. In this context, we can also mention problem versions where the difference between the maximum and the minimum machine completion time, i.e. $C_{max} - C_{min}$ (see e.g. [11,16]), as well as their ratio, i.e. C_{max}/C_{min} (see e.g. [4,12,22]), is to be minimized.

Clearly, balancing issues are taken only implicitly into account by the aforementioned problems as they solely consider the "borders", i.e. the maximum and/or the minimum, of the machine completion times while neglecting at the same time the distribution of work among the remaining machines. However, to adequately account for balancing issues it is essential to incorporate the completion times of all machines into an objective function as done for instance by the *NSSWD*-criterion or, more generally, by the $\mathcal{L}_p$-norm $(\sum_{i=1}^m C_i^p)^{1/p}$ $(1 < p < \infty)$ whose minimization is equivalent to minimizing the sum of the p-th powers of the machine completion times $(\sum_{i=1}^m C_i^p)$. In particular, $P \parallel NSSWD$ is equivalent to $P \parallel \sum_{i=1}^m C_i^2$ (cf. [16]) because

$$\begin{aligned}\sum_{i=1}^{m}(C_i-\mu)^2 &= \sum_{i=1}^{m}(C_i^2 - 2\cdot C_i\mu + \mu^2)\\ &= \sum_{i=1}^{m} C_i^2 - 2\cdot\mu\sum_{j=1}^{n}p_j + m\cdot\mu^2\\ &= \sum_{i=1}^{m} C_i^2 - m\cdot\mu^2. \end{aligned} \tag{6}$$

So, in general, it makes no difference whether we minimize *NSSWD* or $\sum_{i=1}^m C_i^2$. However, we decided for the *NSSWD*-criterion as it represents a normalized objective function which is independent from the order of magnitude of the data and, thus, allows for a more meaningful comparison of solutions (cf. also [5]).

For the problem of minimizing the sum of squared machine completion times, Chandra and Wong [3] prove that the worst-case performance of the LPT-algorithm is bounded by $\frac{25}{24}$. This result has been slightly tightened by Leung and Wei [15]. In the special case of two machines, Koulamas and Kyparisis [13] investigate the worst-case performance of a delayed-start LPT-algorithm which sequences the five longest jobs optimally and assigns the remaining jobs according to the common LPT-rule. They prove that the worst-case ratio is $\frac{50}{49}$. Regarding ideal sets, i.e. there exists an optimal solution in which all $m \ge 2$ machines run equally long, Goldberg and Shapiro [7] improve the worst-case ratio of the LPT-algorithm to $\frac{37}{36}$. Furthermore, Goldberg and Shapiro [8] prove that the worst-case performance of a class of algorithms relaxing the LPT-rule is $\frac{4}{3}$ on ideal sets while minimizing the general $\mathcal{L}_p$-norm. Besides, Alon et al. [1,2] tackle the problem of minimizing the sum of the p-th powers of the machine completion times and construct polynomial time approximation schemes.

Kumar and Shanker [14] compare various balancing objectives and present a comprehensive look at imbalance measures. Rajakumar et al. [17] propose the measure *RPI* (Relative Percentage of Imbalance) whose minimization turns out to be equivalent to minimizing C_{max} in case of identical parallel machines.

To the best of the authors' knowledge the specific literature on $P \parallel NSSWD$ is scarce. The *NSSWD*-criterion has been introduced by Ho et al. [10]. They discuss properties of *NSSWD* and provide an algorithm, called Workload Balancing (WB), for solving the problem. Their work has been extended by Cossari et al. [5] who developed an algorithm, called Partial Solutions and Interchange Algorithm (PSIA), which improves on the results reported by Ho et al. [10] in most cases. Moreover, Cossari et al. [6] propose three other workload balancing measures next to *NSSWD* and suggest the Workload Balancing Algorithm (WBA). With respect to these measures, most often WBA is superior to the procedures studied in Ho et al. [10].

1.3. Contribution and paper structure

The above-mentioned workload balancing algorithms WB, PSIA, and WBA have in common that they construct a feasible solution at first and afterwards they successively consider pairs of machines and swap jobs between them. Using pairs of machines as a neighborhood involves a certain drawback: Since all of the objectives mentioned within Section 1.2 are equivalent in case of two identical parallel machines, an improvement of the *NSSWD*-value is, for instance, associated with an improvement of the makespan. Having in mind that (i) a *NSSWD*-optimal schedule is not necessarily a C_{max}-optimal schedule (which has recently been proved by Walter and Lawrinenko [19]) and (ii) it is generally impossible to transfer a makespan-optimal solution into a *NSSWD*-optimal one via local interchanges of jobs between pairs of machines (cf., again, [19]), there seems to be some potential to improve on the existing workload balancing algorithms. Therefore, we design an exact algorithm for the case of three machines and use this algorithm as the core of our local search procedure for multiple machines. So, the major algorithmic innovation proposed in this paper consists in using triples of machines as a neighborhood instead of pairs of machines. Results of a comprehensive computational study on a large set of benchmark instances demonstrate the benefits of this novel approach for solving $P \parallel NSSWD$.

The remainder of the paper is organized as follows. A straightforward reduction procedure and a lower bound are introduced in Section 2. Our exact algorithm for solving problem $P3 \parallel NSSWD$ is presented in Section 3. Then, in Section 4 we propose suited construction heuristics and an effective local search procedure. The results of our extensive computational study on the benchmark library established by Ho et al. [10] and Cossari et al. [5] are reported and evaluated in Section 5. Finally, Section 6 concludes the paper and suggests ideas for future research.

2. Preprocessing and a simple lower bound

Based on the observation that in case $p_1 \ge \mu$ there exists at least one optimal solution in which a machine solely processes the longest job, we can derive the following straightforward reduction procedure (see Fig. 1). Hence, in the remainder of the paper we can assume $p_j < \mu \ \forall j \in \mathcal{J}$.

```
0. Set i = 1, P = Σ_{j∈𝒥} p_j
1. while p_i ≥ μ
2.     Remove machine i and job i, i.e.:
       Set n = n − 1, m = m − 1, 𝒥 = 𝒥 \ {i}, ℐ = ℐ \ {i}
3.     Set P = P − p_i, μ = P/m, i = i + 1
   end
```

Fig. 1. Reduction procedure.