European Journal of Operational Research 000 (2016) 1–8

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Approximation algorithms for the workload partition problem and applications to scheduling with variable processing times

Daniel Oron[a,*], Dvir Shabtay[b], George Steiner[c]

[a] The University of Sydney Business School, NSW 2006, Australia
[b] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer Sheva, Israel
[c] Operations Management Area, McMaster University, Hamilton, Ontario, Canada

ARTICLE INFO

Article history:
Received 7 May 2015
Accepted 28 June 2016
Available online xxx

Keywords:
Workload partition problem
Scheduling
Identical parallel machines
Controllable processing times
Approximation algorithms

ABSTRACT

In the *Workload Partition Problem* ($\mathcal{WPP}$) we are given a set of n jobs to be scheduled on a set of m identical parallel machines. Each job has its own workload and the scheduling cost on each machine is a convex function of the total workload of the jobs assigned to it. The objective is to minimize the *total* cost on the set of m machines. Shabtay and Kaspi (2006) showed that the $\mathcal{WPP}$ is equivalent to a scheduling problem on m identical machines with controllable processing times and with the scheduling criterion of minimizing the makespan. They also proved that the $\mathcal{WPP}$ is $\mathcal{NP}$-hard when $m = 2$. However, they left as an open question whether the problem is ordinary or strongly $\mathcal{NP}$-hard. Moreover, they provided no practical tools to solve the problem. We bridge those gaps in the literature by showing that the $\mathcal{WWP}$ problem is strongly $\mathcal{NP}$-hard when m is part of the input. Furthermore, we present two different approximation algorithms for solving the $\mathcal{WWP}$ problem. The first one is a fully polynomial time approximation scheme ($\mathcal{FPTAS}$) for a fixed number of machines, while the second is a modification of the well-known longest processing time ($\mathcal{LPT}$) heuristic. We show that our modified $\mathcal{LPT}$ heuristic guarantees a solution with a constant approximation ratio, whose value depends on the instance parameters.

1. Introduction and motivation

Consider the following*Workload Partition Problem* ($\mathcal{WPP}$): A set of n jobs $J = \{J_1, J_2, \ldots, J_n\}$ has to be processed on a set of m identical machines $M = \{M_1, M_2, \ldots, M_m\}$ working in parallel and preemption is not allowed. For $j = 1, \ldots, n$, let w_j be a given parameter indicating the workload of job J_j, and let τ be a partition of set J into m subsets $J_{M_1}, J_{M_2}, \ldots, J_{M_m}$, where J_{M_i} is the set of jobs to be processed on machine M_i for $i = 1, \ldots, m$. Given partition τ, let $W_i = \Sigma_{J_j \in J_{M_i}} w_j$ be the workload on machine M_i for $i = 1, \ldots, m$. The scheduling cost on M_i is an increasing convex function of W_i and is given by $(W_i)^k$, where $k > 1$ is a given constant. Our objective is to find a partition which minimizes the *total* scheduling cost, given by

$$f(\tau) = \sum_{i=1}^{m} (W_i)^k = \sum_{i=1}^{m} \left(\sum_{J_j \in J_{M_i}} w_j \right)^k. \quad (1)$$

The rest of this section is divided into two subsections. In the first one we present two interesting applications of our $\mathcal{WPP}$ in scheduling with deteriorating machines and scheduling with controllable processing times, while the second one is devoted to surveying the existing results for the $\mathcal{WPP}$ and states our research objectives.

1.1. Applications of our $\mathcal{WPP}$

A direct application for the $\mathcal{WPP}$ is in scheduling systems with machine deterioration. In such systems the machine efficiency decreases while processing and thus the processing cost on each machine is an increasing convex function of the process duration. Shabtay and Kaspi (2006) provided another important application of $\mathcal{WPP}$ in the extensively studied field of scheduling with controllable processing times (see Chudzik, Janiak, Lichtenstein, and Janiak (2006), Shabtay and Steiner (2007b) and Janiak, Janiak, and Lichtenstein (2007) for surveys). They consider a scheduling problem on a set of m identical machines working in parallel, where the processing time, $p_j(u_j)$, of job J_j is a non-increasing function of the amount of a nonrenewable resource, u_j, that is allocated to the

* Corresponding author. Tel.: +61 293516583.
E-mail addresses: d.oron@econ.usyd.edu.au, daniel.oron@sydney.edu.au (D. Oron), dvirs@bgu.ac.il (D. Shabtay), george.steiner4@gmail.com (G. Steiner).

http://dx.doi.org/10.1016/j.ejor.2016.06.062

processing operation and is given by

$$p_j(u_j) = \left(\frac{\theta_j}{u_j}\right)^{k'}, \tag{2}$$

where θ_j for $j = 1, \ldots, n$ and k' are positive parameters. They define a solution in terms of the job partition to machines and resource allocation vector $\mathbf{u} = (u_1, u_2, \ldots, u_n)$. The objective is to find a solution that minimizes the *makespan scheduling criterion* which is the completion time of the entire set of n jobs, given by

$$C_{\max} = \max_{i=1,\ldots,m}\left\{\sum_{J_j\in J_{M_i}} p_j(u_j)\right\} = \max_{i=1,\ldots,m}\left\{\sum_{J_j\in J_{M_i}}\left(\frac{\theta_j}{u_j}\right)^{k'}\right\}. \tag{3}$$

subject to

$$\sum_{j=1}^{n} u_j \le \overline{U}, \tag{4}$$

where $\overline{U}$ is an upper bound on the total resource available.

By applying the equivalent load method (see Monma, Schrijver, Todd, & Wei, 1990) Shabtay and Kaspi showed that the optimal resource allocation for job $J_j \in J_{M_i}$ is given by

$$u_j^* = (\theta_j)^{\frac{k'}{k'+1}}\left(\sum_{J_l\in J_{M_i}}(\theta_l)^{\frac{k'}{k'+1}}\right)^{\frac{1}{k'}}\frac{\overline{U}}{w_G}, \tag{5}$$

where

$$w_G = \sum_{i=1}^{m}\left(\sum_{J_j\in J_{M_i}}(\theta_j)^{\frac{k'}{k'+1}}\right)^{\frac{k'+1}{k'}}, \tag{6}$$

and this resource allocation satisfies (4) as an equality. Moreover, they demonstrate that the minimal makespan value (as a function of τ) can be computed by

$$C_{\max}(\tau, \mathbf{u}^*(\tau)) = \left(\frac{w_G}{\overline{U}}\right)^{k'}, \tag{7}$$

where $\mathbf{u}^*(\tau)$ is the optimal resource allocation for a given τ. Since both k' and $\overline{U}$ are constant values, Shabtay and Kaspi conclude that the problem reduces to a purely combinatorial problem of finding a partition τ that minimizes w_G (given by (6)). By setting $k = (k'+1)/k'$ and $w_j = (\theta_j)^{1/k}$ the equivalent relation between this reduced problem and $\mathcal{WPP}$ is achieved.

Remark 1. Note that even if θ_j for $j = 1, \ldots, n$ and k are restricted to integer values, the value of $w_j = (\theta_j)^{1/k}$ may not be an integer value. Therefore, in order to capture the application in scheduling with controllable processing time, we do not restrict the instance of the $\mathcal{WPP}$ to integer values.

We note that Jansen and Mastrolilli (2004) provided a PTAS for the problem of minimizing the makespan on parallel machines when the individual processing times get compressed linearly with the amount of resource allocated. The processing time function in (2) models the rule of diminishing marginal returns from the allocation of additional resources. It has been used extensively in continuous resource allocation theory (e.g., Armstrong, Gu, & Lei, 1995, 1997; Monma et al., 1990; Scott & Jefferson, 1995; Shabtay, 2004; Shabtay & Kaspi, 2004 and Shabtay & Steiner, 2007a) and captures many real life applications. For example, Flynn, Hung, and Rudd (1999) showed that in computer microprocessor manufacturing the processing time is reduced by approximately the cube root of the allocated power, implying that $k' = 1/3$. This cube-root rule for Complementary Metal–Oxide Semiconductor (*CMOS*) based devices was later adopted by many other researchers (see the survey paper by Irani & Pruhs (2005)). Monma et al. (1990) pointed out that the time required to perform many actual government and industrial operations can be expressed by Eq. (2) with $k' = 1$, and that the time required to perform very large scale integration (VLSI) circuit design operations may also be represented by (2) with $k' = 0.5$. Yao, Demers, and Shenker (1995) also use Eq. (2) with $k' = 0.5$ for modeling CPU time via energy consumption.

1.2. Existing results, research objectives and paper organization

Shabtay and Kaspi (2006) proved that the $\mathcal{WPP}$ is $\mathcal{NP}$-hard when $m = 2$ by reducing the Partition problem to it. However, they left open the question whether this problem is ordinary or strongly $\mathcal{NP}$-hard. Moreover, they did not provide any practical tools to solve the $\mathcal{WPP}$. Our main aim in this paper is to bridge these gaps which exist in the literature. The rest of the paper is organized as follows. In Section 2 we show that the $\mathcal{WPP}$ is strongly $\mathcal{NP}$-hard. Approximation algorithms for solving the problem appear in Sections 4 and 5. A summary section concludes our paper.

2. The Strong $\mathcal{NP}$-hardness of the $\mathcal{WPP}$

Consider first the following auxiliary problem (AP).

$$(AP)\quad \min Z(x_1, \ldots, x_m) = \sum_{j=1}^{m}(x_j)^k$$

s.t.

$$\sum_{j=1}^{m} x_j = D$$

The following lemma can be easily obtained by using the Lagrangian method:

Lemma 1. *The* unique *optimal solution for problem AP is* $x_j^* = D/m$ *for* $j = 1, \ldots, n$ *and the minimal objective value is given by* $m(D/m)^k$.

In this section, we prove that the decision version of the $\mathcal{WPP}$, which we denote by $\mathcal{DVP}$, is $\mathcal{NP}$-complete in the strong sense. Unless $\mathcal{P} = \mathcal{NP}$, this result shows that no pseudo-polynomial time algorithms can exist to solve this problem.

Definition 1. $\mathcal{DVP}$: Given a set of n elements $W = \{w_1, w_2, \ldots, w_n\}$ and parameters k and Q, is there a partition τ of set W into m subsets $J_{M_1}, J_{M_2}, \ldots, J_{M_m}$ such that $f(\tau) = \sum_{i=1}^{m}(\sum_{J_j\in J_{M_i}} w_j)^k \le Q$?

The $\mathcal{NP}$-completeness of $\mathcal{DVP}$ will be proven by showing that the strongly $\mathcal{NP}$-complete *3-Partition* problem, defined below, can be polynomially reduced to it.

Definition 2. 3-Partition: Given a set of indices $A = \{1, 2, \ldots, 3h\}$ each of which is associated with a positive integer b_j for $j = 1, \ldots, 3h$, where $\sum_{j=1}^{3h} b_j = hB$ and $B/4 < b_j < B/2$ for $j = 1, \ldots, 3h$. Can the set A be partitioned into h disjoint subsets, $A_1, A_2, \ldots, A_h$ such that $\sum_{j\in A_i} b_j = B$ for $i = 1, \ldots, h$?

Theorem 1. *$\mathcal{DVP}$ is $\mathcal{NP}$-complete in the strong sense.*

Proof. Given an instance $\mathcal{I} = \{b_1, b_2, .., b_{3h}, B\}$ for the *3-Partition* problem, we construct an instance $\widetilde{\mathcal{I}} = \{n, m, W, Q\}$ for $\mathcal{DVP}$ with $m = h$, $n = 3h$, $W = (w_1, w_2, .., w_{3h}) = \{b_1, b_2, .., b_{3h}\}$ and $Q = mB^k$.

We first show that if there exists a solution S, which yields a *YES* answer for the *3-Partition* problem, then there exists a solution $\widetilde{S}$, for the corresponding instance of $\mathcal{DVP}$ with $\sum_{i=1}^{m}(\sum_{J_j\in J_{M_i}} w_j)^k \le Q = mB^k$. Since solution S yields a *YES* answer for the *3-Partition* problem we have that under solution S, set A is divided into h subsets, $A_1, A_2, \ldots, A_h$, such that $\sum_{j\in A_i} b_j = B$ for $i = 1, \ldots, h$.