A Resource Minimizing problem with Fuzzy Restricted Processing Time

Sangheon Han

Abstract

This paper investigates a resource minimization scheduling problem with restricted processing times on an identical parallel environment. Up to now, some multi-machine scheduling problems, including identical parallel environments etc., have concentrated on minimizing make-span under a fixed amount of resources (facilities or machines). However, in real situations, the make-span may be decided by accumulated experiences of experts or desired due–date of customers, etc. So, we treat the make-span with a fixed and prospected constant, and focus on scheduling to minimize resources with a fixed make-span. For solving the problem, a linear time algorithm based on the bin packing problem is proposed and its validity is discussed.

Keywords: Bin Packing Problem, Restricted Processing Time, Minimize Number of Machines

1. Introduction

Up to now many classic machine-scheduling problems have concentrated on minimizing make-span (maximum completion time). Some multi-machine scheduling algorithms on identical parallel environments etc., have proposed minimizing make-span under a fixed number of machines (facilities or resources) in general. However, in real situations, the completion time may be prospected by accumulated experiences of experts or desired due-dates of customers, etc. So, in this paper we treat the make-span not with a decision variable but a fixed and prospected constant.

Furthermore, the processing times of tasks may be ambiguous such as in tasks in the Theory of Constraints Scheduling. Theory of Constraints Scheduling points out the reason for ambiguous processing time due to prolongation of starting point of tasks, overestimation of processing time, closed door of completion of tasks etc. In [5], all values having more digits are rounded to that maximal number of digits and the rounding procedure error is not relevant. Therefore, we introduce a concept of fuzzy processing times for reflecting a more exact restricted processing time. The concept can transform fuzzy and ambiguous processing time to an actual one and play the role of relaxing computational complexity. In [6], is proposed a hybrid improvement procedure for the bin packing problem such as a Tabu-search. This heuristic has several features: the use of lower bounding strategies; the generation of initial solutions by reference to the dual min-max problem; the use of load redistribution based on dominance.

In this paper, we consider a non-preemptive task scheduling problem to minimize the number of machines under fixed desired completion time on identical parallel machines as an example model and suppose fuzzy restricted processing time. From a cost viewpoint of minimizing resources, it is important to minimize the number of machines. Our problem is equivalent to the bin packing problem by treating the number of machines, desired completion time to numbers of bins, capacity of bins, respectively. For solving the above problem, we propose a linear time algorithm based on a bin packing algorithm and discuss its validity.

2. Bin Packing Problems

Johnson D.S. first studied the bin packing problem in 1973 [2] as follows:

Bin Packing Problem (***BPP***)

(1) $U=\{u_1, \cdots, u_n\}$ is a set of elements

(2) $g(u_j) \in (0, 1)$ is a rational weight of uj ($j=1, \cdots, n$)

(3) Find minimal number k of partitions $\{U_1, \cdots, U_k\}$ of U

where, $\sum_{n \in Ui} g(u) \leq 1$ ($i=1, 2, \cdots, k$)

Afterwards, some papers, e.g. [3], [4], proposed the generalized bin packing problem and analyzed it.

Generalized Bin Packing Problem (***GBPP***)

(1) $U=\{u_1, \cdots, u_n\}$

(2) $g(u_j) \in [0, 1]^s$ ($j=1, \cdots, n$)

(3) Find minimal number k of partitions $\{U_1, \cdots, U_k\}$ of U

where, $\sum_{n \in Ui} g(u) \leq (1, \cdots, 1)$ ($i=1, \cdots, k$).

The ***GBPP*** differs from ***BPP*** in that weights are vectors of a fixed dimension (consisting of s elements) over the interval [0, 1]. This problem is known to as the *s-dimensional bin packing problem* and is proved NP-Complete by [4].

In 1983, Blazewicz, J. and Ecker, K. proposed the following bin packing problem [5], where the set of possible weights of elements is restricted by the natural number p. They pointed out that the reason for restricted weights was that all values having more digits are rounded to that maximal number of digits and the rounding procedure error is not relevant.

Fuzzy Restricted Bin Packing Problem (***FRBPP***)

(1) $U=\{u_1, \cdots, u_n\}$

(2) Fixed a natural number $p \in N$,

$g(u_j) \in \{0, \frac{1}{p}, \frac{2}{p}, \cdots, \frac{p-1}{p}, \}^s - \{0, \cdots 0\}$, ($j=1, \cdots, n$)

(3) Find minimal number k of partitions $\{U_1, \cdots, U_k\}$ of U

where, $\sum_{n \in Ui} g(u) \leq (1, \cdots, 1)$ ($i=1, \cdots, k$).

The above problem is called an *s-dimensional p-restricted bin packing problem*. A linear time algorithm (hereafter we call it *RBPA*) by special integer programming is proposed for solving *RBPP* [5].

Here, fuzzy processing time is characterized by a membership function of fuzzy theory. The type of membership function is various according to the decision maker. If we fix a satisfaction level, the ambiguous processing time is defined as in the above assumption (2). Basically, we solve all restricted processing time problems with respect to a satisfaction level of [0, 1]. Selecting the method of best solution depends on the decision maker in usual cases.

3. Scheduling with Fuzzy Restricted Processing Time on Identical Machines

3. 1 Classic problem to minimize make-span

The classic problem of scheduling non-preemptive tasks can be formulated as follows;

(1) $T = \{T_1, T_2, \cdots, T_n\}$ denotes set tasks
(2) $P = \{p_1, p_2, \cdots, p_n\}$ is set of processing times of task T
(3) $M = \{M_1, M_2, \cdots, M_n\}$ is set of identical parallel machines
(4) find non-preemptive schedule to minimize maximum completion time.

3. 2 Problem to minimize numbers of machine

Our problem can be formulated as follows:

(1) $T = \{T_1, T_2, \cdots, T_n\}$ denotes set of non-preemptive tasks
(2) $P = \{p_1, p_2, \cdots, p_n \leq p\}$ is set of fuzzy processiong times of task T
for asatisfaction level
p is sifficiently large natural number
(3) $M = \{M_1, M_2, \cdots\}$ is set of identical parallel machines
(4) c denotes desired completion time
(5) find non-preemptives schedule to minimize number of machines k and to complete all tasks until c, simultaneously.

Our scheduling problem with p-restricted processing times is equivalent to the *1-dimensional P-restricted bin packing problem* by treating the number of machines as k, the desired completion time as c to number of bins, capacity of bin, respectively (Fig.1).

For solving our problem, it is sufficient to utilize the algorithm RBPA. We illustrate a constructive example as follows for understanding the algorithm *RBPA;*

Problem: There are 4 tasks $\{T_1, T_2, T_3, T_4\}$ and their processing times are restricted by the natural number $p = 6$ and given by $\{p_1 = 2, p_2 = 4, p_3 = 6, p_4 = 4\}$. When we fixed the desired completion time to $c = 6$, we employ a schedule to minimize the necessary number of machines finishing all tasks up to 6.

First of all, we can obtain weights of tasks

$$\{g(T_1) = 1/3, g(T_2) = 2/3, g(T_1) = 1, g(T_4) = 2/3\}$$

by dividing each processing time by the desired completion time c=6. All weights are possessed into possible 6-restricted processing times $\{0, 1/3, 2/3, 1\} - \{0\}$. Therefore, the above problem is equivalent to a *1-dimensional 6-restricted bin packing problem*.

Intuitively, we can consider a minimal number of partitions $k = 3$ such that

$$U = \{U_1 = \{T_1, T_2\}, U_2 = \{T_3\}, U_3 = \{T_4\}\}$$

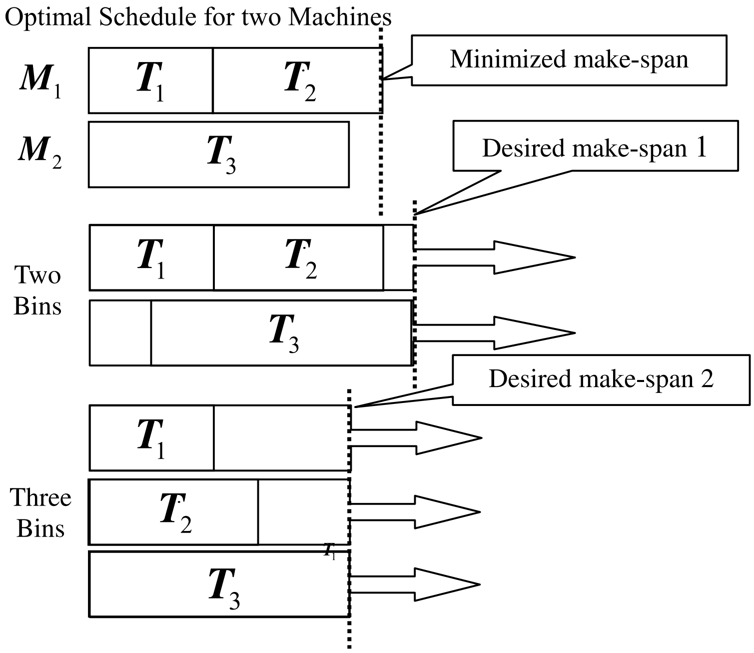

Fig.1. Desired make-span and minimized number of machines

as a necessary number of machines.

For explaining the algorithm, easily, see Table 1.

Where, each b_1, b_2, b_3, denotes the number of each possible element $1/3$, $2/3$, 1, respectively. For

Table 1. When weights of $\{g(T_1)=1/3, g(T_2)=2/3, g(T_1)=1, g(T_4)=2/3\}$, ie., $g(T_i)\in\{0,1/3,2/3,1\}^1-\{0,\cdots,0\}^1$, situation of bin

no. of elements in one bin	index i	pattern in bin (≤ 1)	$\mathbf{b}_i=(b_1, b_2, b_3)$	$\mathbf{v}=(\nu_1, \nu_2, \nu_3)$
1	1	$\frac{1}{3}$	$\mathbf{b}_1=(1, 0, 0)$	$\mathbf{v}=(1, 2, 1)$
	2	$\frac{2}{3}$	$\mathbf{b}_2=(0, 1, 0)$	
	3	1	$\mathbf{b}_3=(0, 0, 1)$	
2	4	$\frac{1}{3}+\frac{1}{3}$	$\mathbf{b}_4=(0, 0, 0)$	
	5	$\frac{1}{3}+\frac{2}{3}$	$\mathbf{b}_5=(1, 1, 0)$	
3	$K=6$	$\frac{1}{3}+\frac{1}{3}+\frac{1}{3}$	$\mathbf{b}_{K=6}=(3, 0, 0)$	

example, $\mathbf{b}_5=(1, 1, 0)$ denotes that the 5-th bin pattern consists of two tasks; one has a processing time $1/3$ and the other has $2/3$. Each ν_1, ν_2, ν_3, denotes the number of each given element $1/3$, $2/3$, 1, respectively. For example, $\mathbf{v}=(1, 2, 1)$ denotes one $1/3$, two $2/3$ and one 1 when weights of

$$\{g(T_1)=1/3,\ g(T_2)=2/3,\ g(T_1)=1,\ g(T_4)=2/3\}$$

are given.

K denotes the number of all possible patterns in a bin. There are 6 patterns in the example from a bin consisting of one task to 3 tasks.

So the central point of the algorithm is how an input can be partitioned into a minimal number of patterns as in table 1. For solving the problem, it is necessary to find $x_1, x_2, \cdots, x_{K=6}$ for fixed $K=6$, in following the integer programming problem;

$$\begin{aligned} \text{Minimize} \quad & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ \text{Subject to} \quad & x_1 + 2x_4 + x_5 + 3x_6 = 1 \\ & x_2 + x_5 = 2 \\ & x_3 = 1 \\ & x_i \geq 0,\ x_i (i = 1, 2, \cdots, 6) \text{ is integer.} \end{aligned}$$

The General case of the algorithm is formulated as the following integer problem:

***Fuzzy Restricted Bin Packing Algorithm* (*FRBPA*)**

For a satisfaction level fixed K, find $x_1, x_2, \cdots, x_K$ such that the objective function

$$\text{Minimize } \sum_{i=1}^{K} x_i$$

$$\text{Subject to } \sum_{i=1}^{K} x_i \mathbf{b_i} = \mathbf{v}$$

$x_i \geq 0$, x_i is an integer.

Where K denotes the number of possible patterns satisfying bin capacity (less than or equal to 1) by all restricted weights and K is obtained by fixing s, p.

Theorem 1 For each fixed natural numbers (s, p) and a satisfaction level, there is an algorithm ***FRBPA*** of linear time complexity which finds an optimal solution for the problem of ***FRBPP***.

Proof: clear by Blazewicz, J. and Ecker, K.. Refer to [5].

4. Conclusion and Future Extensions

We treated a restricted scheduling problem to minimize the number of machines in this paper. A linear time Algorithm based on Blazewicz J. is proposed and its validity and computational complexity discussed.

As further research, it is necessary to extend the algorithm to a restricted resource constrained scheduling problem and other restricted scheduling factors except processing time. Furthermore, it is thought that the real test of an application program is also required.

References

[1] Lenstra, Jr.,K.L.(1981). Integer programming with a fixed number of variables. Mathematisch Instituut, Amsterdam, *Report* 81-01

[2] Johnson, D.S.(1973). Near-optimal bin packing algorithms. *Doctoral Thesis,* Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA.

[3] Krause, K.L, Shen, V.Y, Schwetman, H.D.(1975). Analysis of several task-scheduling algorithms for a model of multi-programming computer systems. *J. ACM,* **22**, 522-550.

[4] Garey, M.R, Johnson, D.S. (1975). Complexity results for multiprocessor scheduling under resource constraints. *SIAM J. Comput,* **4**, 397–411.

[5] Blazewicz, J., Ecker, K. (1983). A linear time algorithm for restricted bin packing and scheduling problems. *Operations research letters,* **2**, 80–83.

[6] A. C. F. Alvim, D. J. Aloise, F. Glover, and C. C. Ribeiro. (2001) A hybrid Improvement heuristic for the bin packing problem. In Extended Abstracts of the IV Metaheuristics International Conference, pages 63–68.