Computers & Industrial Engineering 94 (2016) 1–5

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Minimizing make-span in 2-stage disassembly flow-shop scheduling problem

Mehran Hojati *

Edwards School of Business, University of Saskatchewan, 25 Campus Drive, Saskatoon, Saskatchewan S7N 5A7, Canada

ARTICLE INFO

Article history:
Received 19 January 2012
Received in revised form 4 January 2016
Accepted 22 January 2016
Available online 28 January 2016

Keywords:
Disassembly
Flow-shop
Scheduling
Make-span

ABSTRACT

In this paper we address the 2-stage m-machine (in Stage 2) disassembly flow-shop (2SmMDF) scheduling problem. It consists of n jobs, each requiring $m + 1$ tasks. The first task is the disassembly of a job, and it leads to m processing tasks each of which is performed on a different machine. The objective is to minimize the make-span. First we show that 2SmMDF problem for the make-span criterion is NP-complete. Then we propose the following three heuristic methods for its solution: H1: determine the maximum of Stage 2 task times for each job and use that and the time required for the first task to schedule the jobs according to the Johnson's rule; H2: determine the total time required at each Stage 2 machine over all the jobs and use the time of the machine with the largest total time and the time required for the first task to schedule the jobs according to the Johnson's rule; H3: determine the average of Stage 2 times for each job and use that and the time required for the first task to schedule the jobs according to the Johnson's rule. Finally, we present the worst-case performance analysis for each method, and show that method H3 has a slightly better worst-case performance bound.

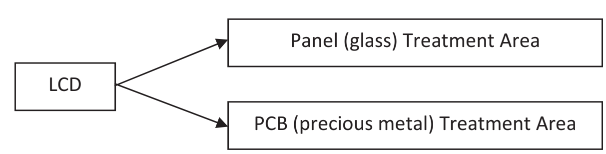
1. Introduction

The 2 stage m-Machine (in Stage 2) Disassembly Flow-shop (2SmMDF) Scheduling problem consists of n jobs, each requiring $m + 1$ tasks. The first task is the disassembly of a job, and it leads to e m processing tasks each of which is performed on a different machine.

The time required for each task is known but task times are likely different. For each job, the second-stage tasks cannot start until the first task is completed, but then they can be done in parallel. The objective is to minimize the make-span criterion.

We assume that (i) no machine can handle more than one job at a time, (ii) no job can be pre-empted, and (iii) there is no space limitation for the jobs waiting to be processed on a machine.

The following example of the special case 2S2MDF (i.e., 2 machines in Stage 2) shows a common recycling process for LCDs where the second-stage tasks involve recovery of glass from the panel and precious metal from the printed circuit board.



Minimizing the make-span results in more efficient use of the resources.

The remainder of this paper is as follows: first we review the literature on 2SmMDF. Next, we define some notation and give an example of the special case 2S2MDF. Then, we show that minimizing the make-span of the 2SmMDF problem is NP-complete. Finally, we present three heuristic methods and their worst case performance analysis.

2. Literature review

We searched the business database ABI/INFORM Complete and engineering database Inspec, using search terms such as "flow shop scheduling" and "disassembly", but could not find any papers directly related to the 2SmMDF scheduling problem. However, the following papers are indirectly related to it.

* Tel.: +1 306 966 8428; fax: +1 306 966 2515.
E-mail address: Hojati@Edwards.usask.ca

http://dx.doi.org/10.1016/j.cie.2016.01.015
0360-8352/

[Johnson (1954)](#) provided a polynomial-time algorithm for minimizing the make-span of the 2-machine (series) flow-shop scheduling problem. Given n jobs each requiring two tasks, one task following the other task, requiring times (P_{i1}, P_{i2}), $i = 1, \ldots, n$, the minimum make-span sequence can be determined as follows: Find the smallest task time; if it is a machine 1 time, put that job first in the sequence; else put it last in the sequence; delete the job and its times, and continue. Alternatively, one can first sort the jobs into two types: Type A jobs have $P_{i1} < P_{i2}$, and Type B jobs have $P_{i1} \geqslant P_{i2}$; then, sort the Type A jobs in ascending order and Type B jobs in descending order; and finally form the sequence as the ordered A jobs followed by the ordered B jobs. The above method is known as Johnson's rule.

Several researchers have tried to extend Johnson's rule to more general scheduling problems, but have discovered that the more general problems are NP-complete (i.e., computationally hard to solve). For example, it is known that minimizing the make-span of the general 3-machine (series) flow-shop problem is NP-complete.

Lee (1993) showed that the 3-machine assembly flow-shop (i.e., when two component tasks feed into a common final assembly task) scheduling problem, with the objective of minimizing the make-span, is NP-complete, and provided some heuristics and a branch-and-bound solution method for finding its optimal solution. Note that this problem is the reverse of our special case 2S2MDF (i.e., 2 machines in Stage 2) problem, with the time reversed.

Lin and Hwang (2011) considered the 2-stage "differentiation" flow-shop scheduling problem which is similar to 2SmMDF but there is no disassembly– jobs of m types start with a common Stage 1 task and then are processed on one of m different machines in Stage 2. The objective is to minimize total completion time.

Veerakamolmal and Gupta (1998) showed how, for a specific job, the operations can be sequenced to recover the desired material in a minimum make-span.

Stuart and Christina (2003) tested various scheduling rules applied to a disassembly and bulk recycling process.

Even though the special case 2S2MDF problem is the reverse of the 3-machine assembly flow-shop problem of Lee (1993), the proof of NP-completeness and the worst case examples for the three heuristic methods are not identical. Also, we have extended our results to m, $m \geqslant 2$, machines in Stage 2. Therefore, our paper makes a new contribution to the literature.

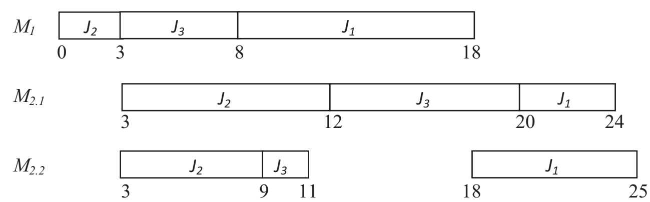
3. Notation and example

Let

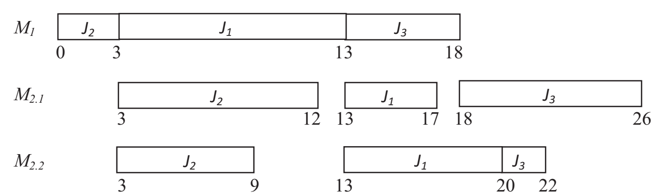
$J = \{J_1, J_2, \ldots, J_n\}$ = set of jobs.
M_1 = Stage 1 (disassembly) machine.
M_{2j} = Stage 2 machine j, $j = 1, \ldots, m$.
P_{i1} = processing time of J_i on M_1.
$P_{i2.j}$ = processing time of J_i on M_{2j}, $j = 1, \ldots, m$.
$T_1 = \sum_{i=1}^{n} P_{i1}$.
$T_{2.j} = \sum_{i=1}^{n} P_{i2.j}$.
$T_{2.max} = max\{T_{2.1}, T_{2.2}, \ldots, T_{2.m}\}$.
$S = J_i J_j \ldots J_k$ = a sequence (a permutation of J).
$C(S)$ = completion time (make-span) of S.
S^* = the S with smallest $C(S)$.

For convenience, we use the name of a job to also represent its processing times: $J_i = (P_{i1}, P_{i2.1}, \ldots, P_{i2.m})$.

Example. Consider the following 3 job 2S2MDF problem, where $J_1 = (10, 4, 7)$, $J_2 = (3, 9, 6)$ and $J_3 = (5, 8, 2)$. It can be shown that $S^* = J_2 J_3 J_1$ and $C(S^*) = 25$:



To illustrate the nature of the 2S2MDF problem, let's consider the schedule of another permutation $S' = J_2 J_1 J_3$:



Note that $M_{2.1}$ finished last in S', whereas $M_{2.2}$ finished last in S^*. This illustrates the difficulty with finding the optimal solution to the 2SmMDF: it is not possible to determine which Stage 2 machine will finish last without examining every sequence.

4. NP-completeness result

We will show that the special case 2S2MDF (with only 2 machines in Stage 2) is NP-Complete, therefore implying that the general 2SmMDF problem is NP-complete.

Lemma 1. *Minimizing the make-span of the 2S2MDF problem is equivalent to the 3-partition problem, a well-known NP-complete problem.*

The 3-partition problem

Given a set A of positive integers x_i, $i = 1, 2, \ldots, 3m$, m any positive integer greater than 1, and a positive integer B such that $mB = \sum_{i=1}^{3m} x_i$ and $B/4 < x_i < B/2$, $i = 1, 2, \ldots, 3m$, can A be partitioned into m disjoint sets $A_1, A_2, \ldots, A_m$ such that $B = \sum_{x_i \in A_i} x_i$ and $|A_i| = 3$, $i = 1, 2, \ldots, m$?

Proof. For each x_i in A, $i = 1, 2, \ldots, 3m$, we construct a job $J_i = (x_i, B_{xi}, 0)$, and we also construct a "spacer" job $J_x = (0, B, B + B^2)$ and $m - 1$ other "spacer" jobs $J_s = (B^2, B, B + B^2)$. We will show that the optimal solution to the 2S2MDF associated with these jobs will have a make-span of $m(B + B^2)$ if and only if the 3-partition problem has a feasible solution.

If part

If the 3-partition problem has a feasible solution $A_1, A_2, \ldots, A_m$, then the sequence $J_x, \{J_i, J_j, J_k\} \in A_1$, J_s, $\{J_i, J_j, J_k\} \in A_2, \ldots, J_s$, $\{J_i, J_j, J_k\} \in A_m$ will have a minimum make-span of $m(B + B^2)$ (with no idle time on either of the second stage machines) as the following Gantt chart illustrates: