A Scheduling Problem of Manufacturing Two Types of Components at a Two-machine Pre-assembly Stage

Chang Sup Sung and Sang Hum Yoon

Department of Industrial Engineering,
Korea Advanced Institute of Science and Technology

Abstract

This paper analyses a deterministic scheduling problem concerned with manufacturing two types of components at a pre-assembly stage which consists of two independent feeding machines each producing its own type of component. Each type represents a unique component which may have variations in its size or quality. Therefore, the completion time of each component depends on both its type and quality (size) variations. Such manufactured components are subsequently assembled into various component-dependent products. The problem has the objective measure of minimizing the total weighted completion time of a finite number of jobs(products) where the completion time of each job is measured by the latest completion time of its two components at the pre-assembly stage. The problem is shown to be NP-complete in the strong sense. A WSPT rule coupled with a machine-aggregation idea is developed for good heuristics which show the error bound of 2.

1. Introduction

This paper considers a simple manufacturing system where two types of components are produced first at its pre-assembly stage and then assembled into final products(jobs) at an assembly stage. The pre-assembly stage consists of two independent machines each producing its own type of components. Each assembly operation cannot start its processing until each type pair of the associated components are prepared from the pre-assembly stage.

In the proposed manufacturing system, this paper is interested in a deterministic scheduling problem concerned with manufacturing two types of components at the pre-assembly stage where the completion time of a product is measured by the latest completion time of its type pair of components at the stage. Minimizing the total weighted completion time of a finite number of products is considered as the scheduling measure.

To order to respond to a variety of customer needs for products, considerable effort has been made in designing products and manufacturing processes in the direction of assuring agility, flexibility and reconfigurability. For example, the concept of modular component design allows products to be produced for a variety of customer demands by using various combinations of standard components and to have high maintainability with ease of replacing any whole defective module directly by a good one. More importantly, as pointed out by Stoll[6], Boothroyd[1], He and Kusiak[3] and Whitney[7], the modular design concept simplifies the final assembly operation because of its contribution to reducing components to assemble. Moreover, each module can be fully checked prior to installation. These characteristics of the modular design give us a motivation to investigate a scheduling problem for producing components at a pre-assembly stage in a situation where the pre-assembly stage is a bottleneck in an assembly line of two stages.

Lee et al.[4] and Potts et al.[5] have discussed problems similar to the proposed problem of this paper. Specifically, Lee et al. have considered a scheduling problem in the aforementioned assembly system with two feeding machines and one assembly machine under makespan measure. Potts et al. have extended the problem of Lee et al. to the case of multiple feeding machines.

2. Problem Statement

There are n jobs, denoted by $J_1, J_2, \cdots, J_n$, to be processed on two machines M_1 and M_2. All jobs consist of two different types of tasks(components). The tasks of each job J_i are assigned to their associated machines such that the task assigned to M_1 must be processed for uninterrupted a_i time units and the other one on M_2 for uninterrupted b_i time units. The machines are not identical but operated in parallel, so both tasks of a job can be handled independently at their respective machines. Each job is processed only when its associated tasks are all available. This implies that the completion time of J_i under a schedule S is measured as

$$C_i(S) = \max\{c_{i1}(S), c_{i2}(S)\},$$

where $c_{ik}(S)$, $k = 1,2$, denotes the completion time of J_i on M_k under S.

For each job J_i, a weight factor w_i is given, so that the total weighted completion time of the schedule S is expressed as

$$WC(S) = \sum_{i=1}^{n} w_i \cdot \max\{c_{i1}(S), c_{i2}(S)\}.$$

This leads to the solution objective of this paper to find an optimal schedule S^* such that $WC(S^*) \le WC(S)$ for any other possible schedule S.

3. Analysis of the Problem

The following property is derived to show that it is sufficient to consider only $n!$ permutations to solve the proposed problem. In the property verification made below, one of job perturbation operations, namely job-insertion operation, will be used. For example, given a sequence on M_k, an operation denoted by $I(i,j,M_k)$ will be implemented to generate a new sequence by inserting J_i into the position of J_j and then shifting each of J_j and all other jobs located between the positions of J_i and J_j (possibly, can be none) to one smaller position(respectively, one greater position) when J_i precedes J_j (resp., J_j precedes J_i) in the original sequence.

Property 1. For the given problem, only permutation schedules need to be considered.

Proof. Consider a non-permutation schedule S. Somewhere in S, there must be a pair of jobs J_i and J_j in different orders on both machines. Without loss of generality, let M_k be the machine on which the last completion time of both J_i and J_j is less than that on the other machine.

If J_i precedes J_j (otherwise, J_j precedes J_i) on M_k, then the operation $I(i,j,M_k)$ (resp., $I(j,i,M_k)$) does not increase the completion time of any job. By repeating the job-insertion operation similarly for every other differently-positioned job, a permutation not worse than the schedule S will be generated. This completes the proof.

It will now be shown that the given problem is NP-complete in the strong sense, and hence, even a pseudo-polynomial time algorithm is unlikely to exist for the given problem. The NP-completeness will be proved by showing a reduction of the 3-partition problem to an instance of the given problem(referring Garey and Johnson[8]).

3-partition problem;
Given a positive integer B and a set of integers X of $3m$ elements $X = \{x_1, x_2, \cdots, x_{3m}\}$ such that $\sum_{x_i \in X} x_i = mB$ and $B/4 < x_i < B/2$ for $i = 1,2,\cdots,3m$, does there exist a partition of X into m disjoint sets $X_1, X_2, \cdots, X_m$ such that $\sum_{x_i \in X_k} x_i = B$ for $k = 1,2,\cdots,m$?

Theorem 1. The given problem is NP-complete in the strong sense.

Proof. Given the 3-partition instance, we construct the following instance of our scheduling problem;

- number of jobs: $n = 4m$,
- job set: $N_1 \cup N_2$, where $N_1 = \{J_1, J_2, \cdots, J_{3m}\}$ and $N_2 = \{J_{3m+1}, J_{3m+2}, \cdots, J_{4m}\}$,
- job processing times: $(a_i, b_i) = (x_i, 0)$ for $J_i \in N_1$ and $(a_i, b_i) = (1, B+1)$ for $J_i \in N_2$,
- weight factors: $w_i = x_i$ for $J_i \in N_1$, and $w_i = m^2 B$ for $J_i \in N_2$.

Let's define the following;

$$T = \sum_{i=1}^{3m}\sum_{j=i}^{3m} x_i x_j + B(B+1)m^3(m+1)/2 + Bm(m-1)/2.$$

We will now show that the above instance has a schedule S such that $WC(S) \le T$ if and only if X has a 3-partition solution.

(a) If X has a partition $X_1, X_2, \cdots, X_m$, then it is easy to check that the total weighted completion time of a schedule S with the order $\{J_i | x_i \in X_1\}, J_{3m+1}, \{J_i | x_i \in X_2\}, J_{3m+2}, \cdots, \{J_i | x_i \in X_m\}, J_{4m}$ is equal to T.

(b) Consider a schedule S such that $WC(S) \le T$. In the schedule, the jobs in N_2 should be finished at the times $B+1, 2(B+1), \cdots,$ and $m(B+1)$; otherwise, $WC(S) > T$ because the additional weight per unit time is m^2B. This further implies that on M_1 at least α jobs in N_2 must finish by the time $\alpha(B+1)$ for $\alpha = 1,2,\cdots,m$. It is easy to check that inserting as many as possible jobs in N_1 before each job z on M_1 will be beneficial for the total weighted completion time, but the sum of any four x_i is greater than B because $x_i > B/4$. Thus, if possible, the best schedule is to get jobs in N_2 finished exactly at the times $B+1, 2(B+1), \cdots,$ and $m(B+1)$ on M_1, which will give the sum of the processing times of the jobs in N_1 before each z exactly equal to B. Moreover, the resulting schedule should constitute a 3-partition solution. This completes the proof.

4. Two Simple Heuristics

The first heuristic $H1$ is briefly described as follows: first, compute $p_i = \max\{a_i, b_i\}$ for each job J_i, and then, generate the WSPT schedule for the single machine instance $\{p_i, w_i | i = 1,2,\cdots,n\}$. Let S_{H1} and S^* denote the schedule generated by $H1$ and an optimal schedule, respectively.

Theorem 2. $WC(S_{H1})/WC(S^*) \le 2$.

Proof. Without loss of generality, let jobs be ordered in S_{H1} such that $p_1/w_1 \le p_2/w_2 \le \cdots \le p_n/w_n$, where $p_i = \max\{a_i, b_i\}$. Let $C_i(S_{H1})$ denote the completion time of J_i under S_{H1}. It is then easy to check that

$$C_i(S_{H1}) \le \sum_{j=1}^{i} p_j \quad \text{for } i = 1,2,\dots,n.$$

Thus, the total weighted completion time of S_{H1} is derived as

$$WC(S_{H1}) = \sum_{i=1}^{n} w_i C_i(S_{H1}) \le \sum_{i=1}^{n} w_i \sum_{j=1}^{i} p_j. \quad (1)$$

Now, consider the feature of S^*. Let $[i]$ denote the subscript to indicate job position under S^*. The completion time of $J_{[i]}$ under S^* is

$$C_{[i]}(S^*) = \max\left(\sum_{j=1}^{i} a_{[j]}, \sum_{j=1}^{i} b_{[j]}\right) \ge \sum_{j=1}^{i}\left(a_{[j]} + b_{[j]}\right)/2. \quad (2)$$

It follows that

$$\begin{aligned} WC(S^*) &\ge \sum_{i=1}^{n} w_{[i]} \sum_{j=1}^{i}\left(a_{[j]} + b_{[j]}\right)/2 \\ &\ge \sum_{i=1}^{n} w_{[i]} \sum_{j=1}^{i} \max\left(a_{[j]}, b_{[j]}\right)/2 \quad (3) \\ &\ge \sum_{i=1}^{n} w_i \sum_{j=1}^{i} p_j/2. \end{aligned}$$

The last inequality of (7) is based on the fact that the WSPT rule guarantees an optimal schedule for the single machine case. From (1), the result is confirmed. This completes the proof.

The second heuristic $H2$ is now briefly described as follows: first, compute $p_i = (a_i + b_i)$ for each job J_i, and then, generate the WSPT schedule for the single machine instance $\{p_i, w_i | i = 1,2,\cdots,n\}$. Let S_{H2} be the heuristic schedule generated by $H2$.

Theorem 3. $WC(S_{H2})/WC(S^*) \le 2$.

Proof. Without loss of generality, let jobs be ordered in S_{H2} such that $p_1/w_1 \le p_2/w_2 \le \cdots \le p_n/w_n$, where $p_i = (a_i + b_i)$. Then, the total weighted completion time of S_{H2} is derived as

$$WC(S_{H2}) = \sum_{i=1}^{n} w_i C_i(S_{H2}) \le \sum_{i=1}^{n} w_i \sum_{j=1}^{i} p_j.$$

Thus, from (2) of Theorem 2, the proof is completed.

5. Conclusion

The proposed problem is important in its own right, but it is also significant in a sense that the results of this paper will provide the underlined scheme and insight for any extended problem.

References

[1] Boothroyd, G. (1992) Assembly Automation and Product Design. Marcell Dekker. New York.

[2] Garey, M.R. and Johnson, D.S. (1979) Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman. San Francisco.

[3] He, D.W. and Kusiak, A. (1996) Performance Analysis of Modular Products. International Journal of Production Research, 34, 253-272.

[4] Lee, C.Y., Cheng, T.C.E. and Lin, B.M.T. (1993) Minimizing the Makespan in the 3-Machine Assembly-type Flowshop Scheduling Problem. Management Science, 39, 616-625.

[5] Potts, C.N., Sevast'janov, S.V., Strusevich, V.A., Wassenhove, L.N. and Zwaneveld, C.M. (1995) The Two-stage Assembly Scheduling Problem: Complexity and Approximation. Operations Research, 43, 346-355.

[6] Stoll, H.W. (1986) Design for Manufacture: An Overview. Applied Mechanics Review, 39, 1356-1364.

[7] Whitney, D.E. (1993) Nippondenso Co. Ltd: A Case Study of Strategic Product Design. Research in Engineering Design, 5, 1-20.