Computers & Industrial Engineering 103 (2017) 242–249

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Minimizing total absolute deviation of job completion times on a single machine with cleaning activities

Ling-Huey Su [a,*], Hui-Mei Wang [b]

[a] Department of Industrial and Systems Engineering, Chung-Yuan Christian University, Taiwan, ROC
[b] Department of Hotel Management, Vanung University, Tao Yuan, Taiwan, ROC

ARTICLE INFO

Article history:
Received 1 June 2016
Received in revised form 8 November 2016
Accepted 10 November 2016
Available online 16 November 2016

Keywords:
Scheduling
Wafer manufacturing
TADC
BIP
Heuristic algorithm

ABSTRACT

This paper considers a new single machine scheduling problem with several unavailability periods, where the machine has to be stopped in order to remove the dirt. The problem appears in the process in wafer manufacturing company. During the processing of a wafer on a specific machine, the dirt, including particle, organic materials and metal-salts, etc., on the surface of the wafer while processing, has to be taken away by a cleaning agent. Once the accumulation of dirt is over a threshold value, it is necessary to interrupt the machine processing and replace the cleaning agent in order to avoid damaging the wafer. The objective is to minimize the total absolute deviation of job completion times (*TADC*). The problem integrates production scheduling and cleaning activities and is strongly NP-hard. Many properties for solving the problem efficiently are explored. A mixed binary integer programming model is developed to find the optimal solution. Based on the properties explored, symmetry of the *V-shaped* rule, and the dynamic programming on scheduling the cleaning activities, an effective heuristic is developed. Computational results indicate that the performance of the heuristic is robust and significantly outperforms the modified *TADC* solution in the literature. Furthermore, the efficiency of the mixed binary integer programming model and the impact of the dirt accumulation as well as cleaning time are explored in detail.

1. Introduction

This paper considers a single machine scheduling problem with several unavailability periods, where the machine has to be stopped in order to remove the dirt. During the processing of a wafer on a specific machine, the dirt, including particle, organic materials and metal-salts, etc., on the surface of the wafer, has to be taken away by a cleaning agent. Once the accumulation of dirt reaches a threshold value, it is necessary to interrupt the machine processing and replace the cleaning agent in order to avoid damaging the wafer. The majority of the studies in machine scheduling literature having unavailability periods often appeared in considering machine breakdown (stochastic) or preventive maintenance (deterministic). Preventive maintenances fall into two categories: (i) maintenances are required in fixed intervals or during some window times and (ii) maintenances are not carried out in fixed intervals and the starting time of each interval is a decision variable. Our study belongs to the second category since the starting time of each unavailability period is determined by the dirt accumulation amount. However, this situation commonly encountered in a wafer manufacturing company cannot be solved by the methods of the second category in the literature. The problem we dealt with considers both scheduling jobs and cleaning activities for removing the dirt. The objective is to minimize the total job completion times or cycle times. According to the notation of Pinedo (1995), the problem is denoted as $1|cleaning|TADC$, where the first field denotes a single machine, the second field *cleaning* denotes the cleaning activities.

A lot of literatures are related to the first category that maintenances are required in fixed intervals or during some window times. Among them, the first study related to our research was by Yang, Hung, Hsu, and Chern (2002). They considered the problem that the machines should be stopped for a constant maintenance or resetting during the scheduling period. They demonstrated that the problem is *NP-hard* and provided a heuristic algorithm with complexity $O(n \log n)$ to solve the problem. Other related articles see Liao and Chen (2003), Chen (2006a, 2006b), Suliman and Jawad (2012), Yin, Ye, and Zhang (2014), and Yin, Xu, Cheng, and Wang (2016).

For the second category that periodic maintenances are flexible, Kubzin and Strusevich (2006) was the first to introduce the concept of variable maintenance in the two-machine flow shop and

* Corresponding author.
E-mail addresses: linghuey@cycu.edu.tw (L.-H. Su), amywang@mail.vnu.edu.tw (H.-M. Wang).

http://dx.doi.org/10.1016/j.cie.2016.11.009
0360-8352/

two-machine open shop. Ni, Gu, and Jin (2015) investigated a dynamic time window for performing the preventive maintenance during production time. They estimated the active maintenance opportunity window to guide how machine can be strategically shut down for preventive maintenance. Compare, Martini, and Zio (2015) evaluated different techniques for maintenance optimization based on Genetic algorithms. The GA-based methods are also developed to handle the uncertainty and fitness of the model. Ying, Lu, and Chen (2016) dealt with four single machine scheduling problems with variable machine maintenance. They proposed an exact algorithm with the computational complexity $O(n^2)$ for each of the four objectives including mean lateness, maximum tardiness, total flow time and mean tardiness. Yin, Xu, Wu, Cheng, and Wu (2014) investigated a single-machine scheduling problem with due-date assignment, generalized position-dependent deteriorating jobs and deteriorating multi-maintenance activities simultaneously. The objective is to minimize an objective function including the cost of due-date assignment, the cost of discarding jobs and the earliness of the scheduled jobs. Yin, Cheng, and Wang (2016) introduced a rescheduling model in which both the original scheduling objective, the total completion time, and the deviation cost associated with a disruption of the original schedule in the presence of machine breakdowns are taken into account. Many variants are considered and solved.

Kanet (1981) was the first to introduce the measure of *TADC* in single machine scheduling problem, denoted as $1\|TADC$. He presented a *V-shaped* arrangement for minimizing this measure. A sequence is said to be *V-shaped* if all jobs to be processed before the job with the smallest processing time are scheduled in decreasing order of their processing times and all jobs to be processed after the job with the smallest processing time are processed in increasing order of their processing times. The *TADC* scheduling problem was extended by considering variations in processing time. Wang and Xia (2007) studied the *TADC* problem with controllable processing time. They solved the problem by formulating it as an assignment problem. Li, Li, Sun, and Xu (2009) considered Kanet's (1981) problem with deteriorating jobs and provided some properties and two heuristics to solve the problem. The same problem with linear function of a job-dependent growth rate and job's starting time was considered by Oron (2008), where the property of *V-shaped* with respect to job growth rates was proposed. Huang and Wang (2011) extended the single machine problem of Li et al. (2009) to the parallel machines problem. Mosheiov (2008) extended the *TADC* of Kanet's problem to multiple machines and position-dependent processing times. Mor and Mosheiov (2011) further extended the work of Mosheiov (2008) to bicriteria objective consisting of a linear combination of total job completion times and *TADC*. Mor and Mosheiov (2012) again studied the problem of scheduling a maintenance activity and due-window assignment based on common flow allowance. As to the *TADC* problem provides approximately the same degree of service to each customer, Li, Zhang, Zhong, and Zhai (2014) modeled the maintenance effect of servicing, analyzed the deterioration characteristics of system under scheduled servicing, and then established the deterioration model from the failure mechanism by compound Poisson process. Öhman et al. (2015) presented a statistical process control based measure that utilizes data typically available in preventive maintenance to improve service productivity.

The main contribution of this paper is the extension of the problem on the conventional machine unavailability by taking into the consideration of dirt constraint, a common practice in IC manufacturing industry. To the best of our knowledge, such a problem has not been studied in the scheduling literature. The complexity of our heuristic is very low, but the algorithm can quickly obtain a near or optimal schedule to satisfy the quick response requirement of the dynamic scheduling environment.

2. Notation and problem setting

We consider a single-machine problem where jobs are nonresumable, i.e., once a job is started it cannot be interrupted till its completion. The setup time of job is sequence independent and included in the processing time. All n jobs are ready for processing at time zero. Each job has a processing time p_i and an amount of dirt t_i, $i = 1, 2, \ldots, n$, left on the machine. A cleaning activity with time w is carried out before the accumulation of dirt reaches a threshold value T, where $t_i \leqslant T$.

The additional notation will be used throughout this paper:

J_i job number i $(i = 1, 2, \ldots, n)$.
C_j the completion time of the job at the j_{th} position in a given sequence $(j = 1, 2, \ldots, n)$.
k_j 1, if cleaning activity is taken immediately following the j_{th} position job; 0, otherwise.
x_{ij} 1, if J_i is scheduled at position j; 0, otherwise.

In addition, $J_{[j]}$ denotes the job placed in the j_{th} position, $p_{[j]}$ denotes its processing time, and $t_{[j]}$ denotes the dirt accumulation between the completion of the last cleaning activity and the completion of the j_{th} job.

The objective is to find a schedule that minimizes the total absolute deviation of job completion times (*TADC*), represented as

$$TADC = \sum_{i=1}^{n}\sum_{j=i}^{n}|C_j - C_i|$$

Denote a schedule π containing a sequence of jobs and several cleaning activities inserted in job sequence. In a schedule, those jobs processed between two adjacent cleaning activities form a batch, denoted as B_i, $i = 1, \ldots, L$. Thus a schedule π can be denoted as $\pi = (B_1, w, B_2, w, \ldots, B_L)$. Note L is a decision variable in our problem. Fig. 1 illustrates the representation of a schedule involving jobs and cleaning activities.

Theorem 1. *The problem* $1|cleaning|TADC$ *is strongly NP-hard.*

Proof. Consider a bin packing problem which considers a set of bins of limited capacity and a set of items of known weight, and the task is to assign items to bins in such a way that the sum of weights of items in each bin does not exceed the bin capacity and as few bins as possible are used. It is known that bin packing is strongly *NP-hard* (Coffman, Garey, & Johnson, 1996). The $1|cleaning|TADC$ problem is reduced to a bin packing problem if the processing time p_i of each job is ignored, and the dirt t_i left is treated as the weight of each item and the upper limit of dirt T corresponds to bin capacity. □

Obviously, there are at least $(L' - 1)$ times of cleaning activities in an optimal schedule, where L' is the minimum bin in the bin packing problem. It is worth mentioning when the processing time and the dirt left are agreeable, i.e., $p_i \leqslant p_j$ implies $t_i \leqslant t_j$, the problem $1|cleaning, agreeable|TADC$ is also *NP-hard*.

For the problem $1|cleaning|TADC$, the optimal schedule may have a larger number of batches than the schedule which contains the minimum number of batches. Therefore, the optimal schedule may need more times of cleaning activities to achieve less *TADC*. The following is a simple example.

Example 1. Let $n = 4$, $p_1 = 7$, $p_2 = 6$, $p_3 = 1$, $p_4 = 5$, $t_1 = 2$, $t_2 = 1$, $t_3 = 2$, $t_4 = 3$, $w = 1$, $T = 4$. The schedule π_1 with the minimum number of batches is $\pi_1 = (J_1 J_3, w, J_4, J_2)$ with $L = 2$ and $TADC = 45$.