



Decision Support

Production-driven opportunistic maintenance for batch production based on MAM–APB scheduling

Tangbin Xia^{a,b,*}, Xiaoning Jin^b, Lifeng Xi^a, Jun Ni^b^a State Key Laboratory of Mechanical System and Vibration, Department of Industrial Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China^b Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109-2136, USA

ARTICLE INFO

Article history:

Received 10 January 2014

Accepted 2 August 2014

Available online 11 August 2014

Keywords:

Maintenance

Batch production

Advance-postpone balancing

Set-up opportunity

Cost saving

ABSTRACT

Nowadays, modern production patterns, such as batch production, have brought new challenges for multi-unit maintenance decision-making. The maintenance scheduling should not only consider individual machine deterioration, but also apply to batch production with variable lot size. An interactive bi-level maintenance strategy is thus proposed in a multi-unit batch production system with degrading machines. In the machine-level scheduling, a multi-attribute model (MAM) is used to obtain maintenance intervals according to individual machine degradation. In the system-level scheduling, a novel production-driven opportunistic maintenance strategy is developed by considering both machine degradation and characteristics of batch production. In this strategy, advance-postpone balancing (APB) utilizes set-up times as opportunities to make real-time schedules for system-level maintenance. The numerical example shows that the proposed MAM–APB methodology can efficiently eliminate unnecessary production breaks, achieve significant cost reduction and overcome complexity of system scheduling.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

In today's competitive industrial market, maintenance scheduling has played a more and more important role to meet the innovation requirements of modern production patterns. However, huge cost wastes are caused by improper scheduling in realistic environments (Moblely, 2002; Ruiz-Castro & Li, 2011). To satisfy the increasing customer demands, multi-unit manufacturing systems need appropriate maintenance policy to achieve high system performance with minimum cost. On the one hand, individual machine degradation should be considered in the machine-level scheduling. Monitoring techniques of condition-based maintenance (CBM) and prognostic & health management (PHM) make it more feasible to obtain machine degradation data (Lee et al., 2014). On the other hand, modern production characteristics should be studied in the system-level scheduling. Therefore, the enabling methods for real-time production and maintenance scheduling are needed to operate in a cost effective manner for

practical applications (Li, Liu, & Chen, 2012; Topal & Ramazan, 2010).

As one of modern production patterns, batch production is widely applied to response quickly to customer demands. This production pattern has changed the manufacturing process from the “push” pattern to the “pull” pattern. In batch production, batch orders are processed through multi-unit systems according to market demands (Anzanello, Albin, & Chaovalitwongse, 2012; Cardenas-Barron, 2009; Li, 1997). Different from traditional flow-line production, batch production has the following modern production characteristics: (1) batch orders are independent with variable lot size; (2) these sequential batches are ordered only a short time beforehand; (3) a set-up work happens when one batch switches to another; and (4) each batch cycle prefers no interruptions to ensure product quality. Therefore, other than classical opportunistic maintenance, a novel system-level maintenance strategy is required to eliminate unnecessary production breaks, achieve significant cost reduction and overcome complexity of system scheduling by considering these production characteristics.

For developing a bi-level maintenance strategy (interactive machine-level and system-level scheduling), it should be noticed that batch production systems are usually equipped with various machines, which have different reliability parameters and degrading processes. Each machine undergoes increasing wear and

* Corresponding author at: State Key Laboratory of Mechanical System and Vibration, Department of Industrial Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. Tel./fax: +86 21 3420 6539.

E-mail addresses: xtbxtb@sjtu.edu.cn, tangbinxia1121@gmail.com (T. Xia).

deterioration with aging, which will finally lead to failures and interrupt the normal production. In the recent decades, numerous valuable studies have been devoted to single-machine maintenance scheduling by combining maintenance plan and production schedule (Chen, 2006, 2009; Ji, He, & Cheng, 2007; Liao & Chen, 2003). In the notable studies, Wang and Liu (2013) developed a branch and bound (B&B) algorithm for production scheduling and preventive maintenance in a single machine with its time to failure subject to a Weibull probability distribution. Cassady and Kutanoglu (2005) proposed an integrated model that coordinated preventive maintenance planning decisions with single-machine scheduling decisions so that the total expected weighted completion time of jobs was minimized. Chen (2008) considered the single machine scheduling problem and assumed that machine maintenance must be undertaken following a given period, where multiple periodic maintenance intervals existed, with the maintenance interval being flexible through the planning horizon. Compared with these groundbreaking studies, the machine-level scheduling in this study focuses on taking advantage of the multiple attribute value theory and imperfect maintenance to dynamically obtain maintenance intervals according to machine degradation. Firstly, some other scheduling objectives, besides cost, can be considered by decision makers (Costa, Carnero, & Oliveira, 2012; Jiang & Ji, 2002). The multiple attribute value theory is utilized as a tool to formally handle a multi-attribute decision problem. Secondly, it has been recognized that PM cannot recover a machine to an “as good as new” state. Pham and Wang (1996), and more recently, Doyen and Gaudoin (2004) have discussed the maintenance scheduling with imperfect maintenance. Thirdly, the machine-level maintenance intervals should be dynamically integrated into the following system-level scheduling process. The machine-level scheduling mode is required to support the system-level scheduling of preventive maintenance and batch production in a real-time and sequential mode. Therefore, a multi-attribute model (MAM) developed by Xia, Xi, Zhou, and Du (2012) is used in this study.

Compared with the machine-level scheduling, the system-level scheduling is much more complex, since the interactions between machines should also be considered in decision making (Dekker, Wildeman, & van der Duyn Schouten, 1997; Okogbaa, Otieno, Peng, & Jain, 2008). With the extensive application of highly complex systems, there has been a growing interest in multi-unit maintenance scheduling. Though the existing complexity of system scheduling, some valuable system-level maintenance strategies have been proposed in the conception of opportunistic maintenance (Bedford, Dewan, Meilijson, & Zitrou, 2011; Rachaniotis & Pappis, 2008). Opportunistic maintenance basically refers to the scheme in which preventive maintenance is carried out at opportunities. When one machine in the system fails or is preventive maintained, PM opportunities arise for other machines. The advantage of opportunistic maintenance is that one PM combination with several PM actions can be used to save much group maintenance costs. Zhou, Xi, and Lee (2009) proposed an opportunistic PM scheduling algorithm for series systems by maximizing cumulative cost savings. Xia, Xi, Zhou, and Lee (2012) developed the maintenance time window (MTW) to make a cost-effective schedule by dynamically utilizing opportunities in series-parallel systems. However, this classical type of opportunistic maintenance for flow-line production focuses on opportunities caused by maintenance arrangements and does not consider production plans in a holistic manner, thus it is only called maintenance-driven opportunistic maintenance. To meet the innovation requirements of batch production pattern, there is a great need to develop a novel type of opportunistic maintenance, which not only considers the degradation of each machine, but also satisfies the characteristics of batch production.

In order to propose a production-driven opportunistic maintenance strategy suitable for batch production, it is necessary to analyze the impacts of maintenance schedules and batch orders. The challenges from production characteristics should be emphasized in the system-level scheduling. Due to the complexity of jointly scheduling for preventive maintenance and production plan, there are very few studies on methodologies for multi-unit degrading systems (Lee, Wu, Wen, & Chung, 2008; Liao, Shyur, & Lin, 2005; Mellouli, Sadfi, Chu, & Kacem, 2009; Sun & Li, 2010). Allaouia, Lamourib, Artibab, and Aghezaf (2008) studied the problem of jointly scheduling n immediately available jobs and the preventive maintenance in a two-machine flow shop to minimize the makespan. Tan, Chen, and Zhang (2011) considered a parallel-machine scheduling problem with machine maintenance to minimize the total completion time. In all, these works play a great role in the joint scheduling for preventive maintenance and batch production. However, according to batch production characteristics, there still remain three issues to be addressed. First, most of these strategies focus on two-machine scheduling and suffer from intractability when the number of machines grows. The traditional description of the system condition (i.e., Markov process) makes the analysis extremely complicated in multi-unit system modeling (Wildeman, Dekker, & Smit, 1997). Second, the classical maintenance-driven opportunistic maintenance only schedules to advance PM actions. In production practice, it can be extensionally analyzed whether to advance or to postpone each PM activity. Third, considering the increasing uncertainty of customer demands in batch production, real-time scheduling ability of multi-unit systems in a cost effective manner is needed in industrial companies that makes various products in large and discrete batches, when demand and the production process are stochastic. Therefore, five improvable directions have been considered in the proposed bi-level maintenance strategy for batch production: (1) dynamic PMs with flexible intervals are more practical than periodic maintenance; (2) the machine-level and the system-level scheduling should be interactive in a dynamic way; (3) the set-up time between two batches can be the new opportunities for group maintenance; (4) according to the “pull” manufacturing pattern, real-time system-level scheduling is conducted cycle by cycle; and (5) production-driven opportunistic maintenance based on cost analysis in each cycle can resolve the complexity of system-level scheduling.

In this paper, an interactive bi-level maintenance policy is proposed by considering both individual machine degradation and batch production characteristics. This strategy helps industrial companies to eliminate unnecessary production breaks, achieve significant cost reduction and overcome complexity of system scheduling. For each single machine, a multi-attribute model (MAM) is used to obtain maintenance intervals based on availability maximization and cost minimization. Based on real-time machine-level intervals, an advance-postpone balancing (APB) strategy is presented to operate production-driven opportunistic maintenance in the system-level scheduling. Each set-up time between two batches could be used to perform PM on non-failure machines, so that unnecessary breakdown in batch production could be avoided. The proposed APB programming dynamically analyzes the cost savings of PM advancement and PM postponement, and then makes real-time schedules to satisfy no-disruption requirement and reduce system maintenance cost. The rest of this paper is organized as follows: Section 2 presents the formal description of production-driven opportunistic maintenance. Section 3 illustrates the proposed MAM-APB scheduling method for batch production with a multi-unit system. In Section 4, case studies are investigated with the proposed methodology and results are discussed. In Section 5, conclusions are drawn and future work directions are given.

2. Research design and methodology

For achieving the system cost reduction for batch production, the proposed MAM–APB methodology is applied as the decision-making method, which extends classical opportunistic maintenance policies. As an effective method of group maintenance, a maintenance-driven opportunistic maintenance strategy utilizes a PM action of one machine as opportunities for other machines (Xia, Xi, Zhou, & Lee, 2012; Zhou et al., 2009). However, breakdown within each batch needs to be avoided to ensure product quality. Thus, each set-up work between batches could be considered as new opportunities to maintain several machines simultaneously. The scheme of production-driven opportunistic maintenance based on MAM–APB scheduling is shown in Fig. 1.

This novel production-driven opportunistic maintenance strategy consists of four layers: (1) *Physical layer*: this layer describes and characterizes the complex multi-unit system undertaking batch production of variable lot size. (2) *Data processing layer*: production data flow (order demand, lot size of batch, and production plan) and maintenance data flow (reliability parameter, machine hazard rate, and maintenance effects) are interactively communicated to support the decision-making process. (3) *Decision making layer*: this is the core layer where MAM–APB strategy is applied. According to sequential batches, the real-time PM intervals are derived from the machine-level scheduling process. Then in the system-level scheduling, APB programming could dynamically utilize the set-up works and analyze the cost savings to make real-time PM adjustments. MAM and APB programs iteratively proceed from batch to batch in real time. (4) *Application layer*: schedules from MAM–APB decision-making process are applied in industrial practice.

Compared with maintenance-driven opportunistic maintenance based on maintenance time window (MTW) for flow-line production (Xia, Xi, Zhou, & Lee, 2012), production-driven opportunistic maintenance based on APB have following advantages suitable for batch production. First, APB considers no-disruption requirement, other than traditional constant production assumption. Second, it takes advantage of planned production downtimes as opportunities. Third, APB is scheduled at each set-up time point between two consecutive batches. Fourth, this strategy dynamically analyzes the cost savings of PM advancement and PM postponement to make decision. Last but not the least, a set-up work before a new batch means a breakdown of all the machines, thus the complexity of system scheduling can be significantly reduced for determining whether to advance or postpone PM activities in APB programming. The detailed scheduling process of MAM–APB method will be presented in the following section.

3. MAM–APB strategy for batch production

In the proposed methodology, two kinds of maintenance activities are considered to reduce unplanned downtime in each batch cycle. For each machine, PM is scheduled during set-up times between two batches, while minimal repair is used if it fails during a batch production. As shown in the decision making layer of Fig. 1, MAM provides PM intervals according to machine degradation cycle by cycle. Given PM intervals and sequential batch orders, APB dynamically makes maintenance schedules by utilizing set-up opportunities. Then real-time APB decisions are fed back to MAM calculation for the next batch cycle. By considering individual machine degradation and batch production characteristics, this bi-level strategy aims to reduce the total system maintenance cost.

Assumptions considered in this paper are given in the following:

- (1) The machines in a batch production system are independent with different deterioration process. All enter service at time $t = 0$. The hazard rate at t reflects the health condition.
- (2) PM is imperfect maintenance, which does not make the machine be as good as new, but younger. Minimal repair only recovers the machine to the failure rate it had when it failed.
- (3) Sequential batch orders are independent and can be finished in corresponding production times. A set-up work arises when one batch switches to another.
- (4) The processing time of a single product is negligible in comparison with the production time of a batch order; hence all machines can be seen to finish one batch production at the same time.

Notations

j	index of machine $M_j, j \in \{1, 2, \dots, J\}$
i	index of PM cycles in machine-level scheduling, $i \in \{1, 2, \dots, I\}$
u	index of batch $B_u, u \in \{1, 2, \dots, U\}$
TB_u	time duration of batch B_u
t_{ij}	the time point of the i th PM at machine-level
tb_u	the set-up time point after B_u at system-level
$\Theta(j, tb_u)$	maintenance decision for M_j at tb_u
G_u	PM combination set after B_u
$SCA_{j(u+1)}$	saved cost of PM advancement for M_j in B_{u+1}
$SCP_{j(u+1)}$	saved cost of PM postponement for M_j in B_{u+1}
$APB_{j(u+1)}$	advance-postpone balancing for M_j in B_{u+1}
c_{sj}	set-up cost rate

(continued on next page)

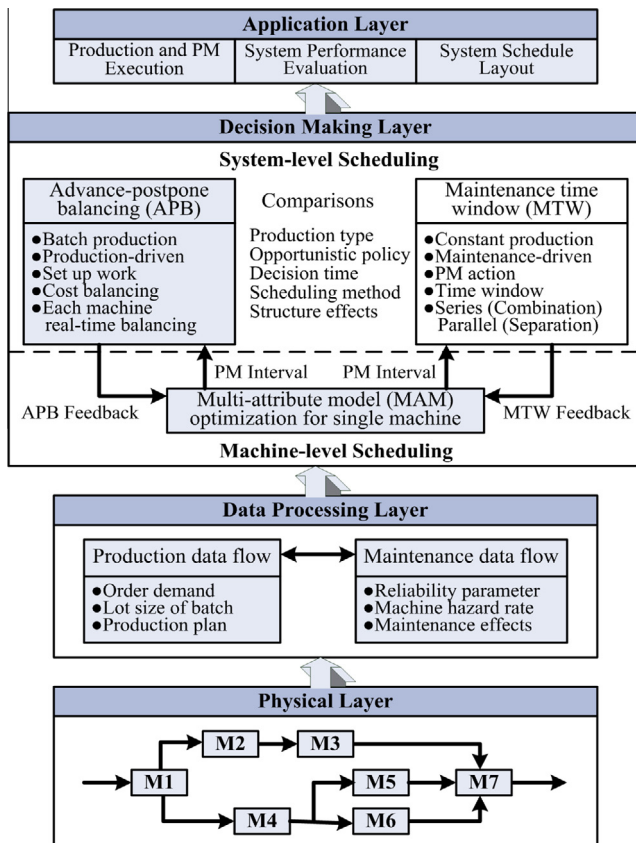


Fig. 1. Scheme of MAM–APB methodology.

c_{dj}	downtime cost rate
$\lambda_{ij}(t)$	hazard rate function prior to the i th PM
C_{pij}	cost of PM action
T_{pij}	time duration of PM action
T_{pumax}	maximum duration for PM actions combined in G_u
C_{rij}	cost of minimal repair
T_{rij}	time duration of minimal repair
A_{ij}	availability of the i th PM cycle
c_{rij}	cost rate of the i th PM cycle
T_{aij}	PM interval of availability model
T_{cij}	PM interval of cost model
T_{oij}	machine-level PM interval

3.1. Multi-attribute model (MAM) in machine-level scheduling

In the machine-level scheduling, MAM method takes advantages of multiple attribute value theory and imperfect maintenance. In our previous paper (Xia, Xi, Zhou, & Du, 2012), MAM has been presented to determine the real-time PM intervals considering both machine availability and maintenance cost in an overall objective function.

In the availability model, a PM cycle consists of two intervals: mean uptime denoted by T_{aij} , and mean downtime. Suppose $\int_0^{T_{aij}} \lambda_{ij}(t)dt$ is the expected frequency of the failures between two successive PM activities. The availability of the i th PM cycle for M_j is given by:

$$A_{ij} = \frac{T_{aij}}{T_{aij} + (T_{pij} + T_{rij} \int_0^{T_{aij}} \lambda_{ij}(t)dt)} \tag{1}$$

The numerator equals to the useful interval and the denominator equals to the cycle duration. Output PM interval T_{aij}^* corresponding to the maximum A_{ij}^* through $dA_{ij}/dT_{aij}|_T = 0$.

In the cost model, the cost of PM action and the possible cost of minimal repair are considered. Suppose T_{cij} is the PM interval in the cost model. The maintenance cost rate of the i th PM cycle is defined as:

$$c_{rij} = \frac{C_{pij} + C_{rij} \int_0^{T_{cij}} \lambda_{ij}(t)dt}{T_{cij} + (T_{pij} + T_{rij} \int_0^{T_{cij}} \lambda_{ij}(t)dt)} \tag{2}$$

The numerator equals to the maintenance cost and the denominator equals to the cycle duration. Output PM interval T_{cij}^* corresponding to the minimum c_{rij}^* through $dc_{rij}/dT_{cij}|_T = 0$.

Then, MAM method is developed to give an overall objective, denoted by V_{ij} , composed of machine availability and maintenance cost. Suppose w_{1ij} and w_{2ij} are the weights of these two terms ($w_{1ij} \geq 0, w_{2ij} \geq 0, w_{1ij} + w_{2ij} = 1$). By combining two single-objective models together, the overall objective function is thus defined as:

$$V_{ij} = -w_{1ij} \frac{A_{ij}}{A_{ij}^*} + w_{2ij} \frac{c_{rij}}{c_{rij}^*} \tag{3}$$

In this function, PM interval denoted by T_{oij} , takes place of T_{aij} and T_{cij} . For the i th cycle for M_j , PM intervals T_{oij}^* can be obtained by $dV_{ij}/dT_{oij}|_T = 0$. By pulling machine-level PM intervals of all machines cycle by cycle, the proposed APB programming dynamically analyzes the cost savings for the system-level scheduling.

In practice, a machine after PM is not as good as brand new one, that is, the hazard rate value is decreased while always greater than zero. Simultaneously, the machine tends to have more frequent maintenance since the hazard rate increases more quickly than it did in the previous PM interval. To sum up, PM not only

decreases the hazard rate to a certain value, but also changes the slope of the hazard rate function. Therefore, for the next PM cycle, given the actual interval T_{ij} from APB feedback, the relationship between hazard rates of consecutive cycles is defined as:

$$\lambda_{(i+1)j}(t) = b_{ij} \lambda_{ij}(t + a_{ij} T_{ij}), \quad t \in (0, T_{(i+1)j}) \tag{4}$$

The age reduction factor a_{ij} , $a_{ij} \in (0, 1)$ shows that imperfect PM makes the machine's initial failure rate reduce to $\lambda_{ij}(a_{ij} T_{ij})$. Meanwhile, the hazard rate increase factor $b_{ij} > 1$ indicates that PM increases the failure rate $b_{ij} \lambda_{ij}(t)$. It should be noticed that when $t = 0$, the hazard rate increase factor has no effect on the failure rate, therefore we could have $\lambda_{(i+1)j}(t) = \lambda_{ij}(a_{ij} T_{ij}), t = 0$. These factors can be deduced from the historical maintenance data and online operational state of the machine (Doyen & Gaudoin, 2004; Pham & Wang, 1996).

When each batch production was finished, current machine-level PM intervals would be pulled to APB programming. In the following process, system-level opportunistic maintenance would be scheduled by taking real-time batch orders into consideration.

3.2. Advance-postpone balancing (APB) in system-level scheduling

Consider a multi-unit manufacturing system with J machines, which are of different types and suffer increasing wear at various rates. For the core system-level scheduling, APB programming dynamically analyzes the cost savings of PM advancement and PM postponement. This strategy focuses on the key observation that each PM advancement or postponement incurs three potential cost savings: downtime cost saving, PM costs saving and minimal repair cost saving. It aims to make a real-time decision for achieving more cost savings. Thus, unplanned downtime in each batch could be eliminated and the total system maintenance cost could be reduced. The schematic illustration of APB programming is shown in Fig. 2.

The APB program works as follows. When the last batch B_u has been finished, and the next batch B_{u+1} has not started, this moment is defined as the current decision time to schedule APB. At this set-up time point tb_u , PM opportunities for machines arise and the time duration of next batch production $T_{B_{u+1}}$ can be pulled. In this batch cycle, PM time points t_{ij} for machines M_1, M_j and M_J are originally in batch B_{u+1} . To avoid production breakdown, these PM actions have to be advanced to the current tb_u (the current PM combination set G_u), or postponed to the next tb_{u+1} (the next PM combination set G_{u+1}). However, there is no need to operate

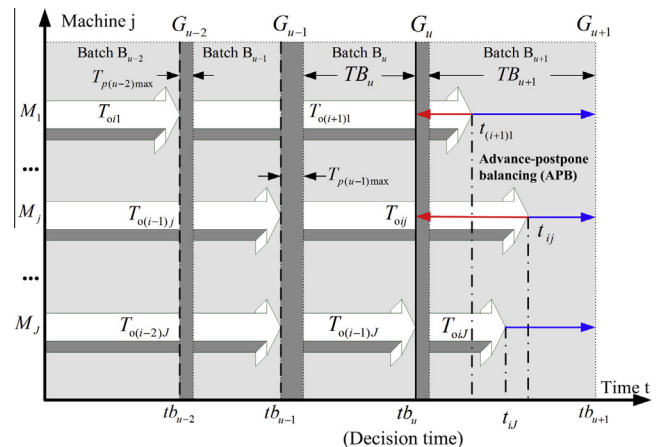


Fig. 2. Schematic illustration of APB programming.

PM twice in the same set-up time; thus the PM of machine M_j should be postponed.

In each batch cycle, APB programming makes real-time decision of PM advancement or PM postponement by comparing corresponding cost savings. On the one hand, if machine M_j is prevented maintained at time tb_u , the saved cost by advancing PM in batch B_{u+1} can be evaluated as:

$$SCA_{j(u+1)} = SCA_{j(u+1)}^d + SCA_{j(u+1)}^f - SCA_{j(u+1)}^p \quad (5)$$

where $SCA_{j(u+1)}^d$ is the downtime cost saving, $SCA_{j(u+1)}^f$ is the minimal repair cost saving, $SCA_{j(u+1)}^p$ is the PM cost saving of PM advancement.

Firstly, according to the downtime cost rate c_{dj} and the set-up cost rate c_{sj} , the downtime cost saving can be defined as:

$$SCA_{j(u+1)}^d = T_{p_{ij}}(c_{dj} - c_{sj}) \quad (6)$$

Secondly, if the PM action is advanced, the PM interval decreases to $T_{oij}^* - (t_{ij} - tb_u)$. Thus the cumulative failure risk is reduced, and the minimal repair cost saving can be represented as:

$$SCA_{j(u+1)}^f = \left[\int_0^{T_{oij}^*} \lambda_{ij}(t) dt - \int_0^{T_{oij}^* - (t_{ij} - tb_u)} \lambda_{ij}(t) dt \right] C_{fij} \quad (7)$$

Thirdly, PM advancement could increase PM cost, since shorter intervals mean that more PM actions would be needed in the same scheduling horizon. According to the ratio of PM interval change and the actual PM interval, the PM cost saving to be minus can be shown as:

$$SCA_{j(u+1)}^p = \frac{t_{ij} - tb_u}{T_{oij}^* - (t_{ij} - tb_u)} C_{p_{ij}} \quad (8)$$

On the other hand, if PM of machine M_j is postponed to the next set-up time point tb_{u+1} , the minimal repair cost saving will be minus and the PM cost saving will be plus. Therefore, the saved cost by postponing PM in batch B_{u+1} can be evaluated as:

$$SCP_{j(u+1)} = SCP_{j(u+1)}^d - SCP_{j(u+1)}^f + SCP_{j(u+1)}^p \quad (9)$$

where $SCP_{j(u+1)}^d$ is the downtime cost saving, $SCP_{j(u+1)}^f$ is the minimal repair cost saving, $SCP_{j(u+1)}^p$ is the PM cost saving of PM postponement. Correspondingly, these cost savings can be obtained by the following equations:

$$SCP_{j(u+1)}^d = T_{p_{ij}}(c_{dj} - c_{sj}) \quad (10)$$

$$SCP_{j(u+1)}^f = \left[\int_0^{T_{oij}^* + (tb_{u+1} - t_{ij})} \lambda_{ij}(t) dt - \int_0^{T_{oij}^*} \lambda_{ij}(t) dt \right] C_{fij} \quad (11)$$

$$SCP_{j(u+1)}^p = \frac{tb_{u+1} - t_{ij}}{T_{oij}^* + (tb_{u+1} - t_{ij})} C_{p_{ij}} \quad (12)$$

Based on above cost saving analysis, $APB_{j(u+1)}$ could be defined as the criterion to decide weather to advance or postpone this PM action. According to the values of SCA and SCP, the APB function can be shown as:

$$APB_{j(u+1)} = SCA_{j(u+1)} - SCP_{j(u+1)} \quad (13)$$

If $APB_{j(u+1)} > 0$, it means that the saved cost of PM advancement is greater, thus PM on machine M_j is advanced to time tb_u ($j \in G_u$). If $APB_{j(u+1)} \leq 0$, the PM action will be postponed to time tb_{u+1} ($j \in G_{u+1}$). This is the APB decision for one machine in a system-level batch cycle. The procedure of production-driven opportunistic maintenance for sequential batches based on MAM-APB scheduling method will be presented in the following section.

3.3. Production-driven opportunistic maintenance based on MAM-APB strategy

To achieve an effective system maintenance schedule for batch production, the proposed bi-level strategy not only considers individual machine degradation, but also integrates batch production characteristics. In essence, APB dynamically determines production-driven opportunistic maintenances by utilizing set-up works and analyzing the cost savings, while MAM supplies real-time machine-level PM intervals. The following procedure obtains system-level maintenance schedules in sequential batch cycles based on MAM-APB strategy.

Step 1: Start from the first cycle $i = 1$. Pull the real-time PM intervals from the machine-level MAM scheduling and evaluate expected PM time points:

$$t_{ij} = T_{oij}^* \quad (j = 1, 2, \dots, J) \quad (14)$$

Step 2: Obtain the first batch order (production duration T_{B_1}). Start from the first cycle $u = 0$, $tb_0 = 0$. Check whether M_j ($j = 1, 2, \dots, J$) is expected to be maintained in B_1 :

$$\Theta(j, tb_0) = \begin{cases} 0 & t_{ij} \notin (tb_0, tb_0 + TB_1] \\ 1 & t_{ij} \in (tb_0, tb_0 + TB_1] \end{cases} \quad (15)$$

Step 3: No PM is needed at $tb_0 = 0$. Thus all PM actions are postponed to the end of B_1 . $\forall \Theta(j, tb_0) = 1$, define $APB_{j_1} < 0$ and $j \in G_{u+1}$. For $u = u + 1 = 1$, the second decision time is given by:

$$tb_u = tb_0 + TB_u = TB_1 \quad (u = 1) \quad (16)$$

Besides, the expected PM time points of M_j ($j = 1, 2, \dots, J$) have been adjusted as:

$$t_{ij} = \begin{cases} t_{ij} + \delta(G_u)T_{pu\max} & \Theta(j, tb_0) = 0 \\ tb_u + \delta(G_u)T_{pu\max} + T_{oij}^*(i = i + 1) & \Theta(j, tb_0) = 1 \end{cases} \quad (17)$$

$$\delta(G_u) = \begin{cases} 0 & |G_u| = 0 \\ 1 & |G_u| > 0 \end{cases} \quad (18)$$

where $|G_u| = 0$ means there is no PM action in the PM combination G_u . Otherwise, $|G_u| > 0$.

Step 4: Time check: When batch B_u ($u = 1, 2, \dots$) has been finished and a new batch order with $T_{B_{u+1}}$ is obtained, identify whether the expected PM actions will be performed during the new batch B_{u+1} :

$$\Theta(j, tb_u) = \begin{cases} 0 & t_{ij} \notin (tb_u, tb_u + TB_{u+1}] \\ 1 & t_{ij} \in (tb_u, tb_u + TB_{u+1}] \end{cases} \quad (19)$$

Step 5: APB check: For $\forall \Theta(j, tb_u) = 1$, analyze the cost savings of PM adjustments in the system-level scheduling, and use the APB criterion to advance or postpone PM actions (if we already have $j \in G_u$, define $j \in G_{u+1}$):

$$j \in \begin{cases} G_u & APB_{j(u+1)} = SCA_{j(u+1)} - SCP_{j(u+1)} > 0 \\ G_{u+1} & APB_{j(u+1)} = SCA_{j(u+1)} - SCP_{j(u+1)} < 0 \end{cases} \quad (20)$$

Step 6: Update time and feedback: For the next batch cycle $u = u + 1$, update the set-up time point after B_u in the system-level scheduling and the expected PM time points of M_j ($j = 1, 2, \dots, J$) based on the machine-level scheduling (MAM determines new PM intervals according to the actual previous intervals from APB decision through Eq. (4)):

$$tb_u = tb_{u-1} + \delta(G_{u-1})T_{p(u-1)\max} + TB_u \quad (21)$$

$$t_{ij} = \begin{cases} t_{ij} + \delta(G_u)T_{pu\max} & \Theta(j, tb_{u-1}) = 0 \\ tb_{u-1} + \delta(G_{u-1})T_{p(u-1)\max} + T_{oij}^*(i = i + 1) & \Theta(j, tb_{u-1}) = 1 \quad j \in G_{u-1} \\ tb_u + \delta(G_u)T_{pu\max} + T_{oij}^*(i = i + 1) & \Theta(j, tb_{u-1}) = 1 \quad j \in G_u \end{cases} \quad (22)$$

Step 7: If there is following batch orders coming sequentially, turn to time check for $\Theta(j, tb_u)$ in Step 4; perform APB programming $APB_{j(u+1)} = SCA_{j(u+1)} - SCP_{j(u+1)}$ to make the system-level schedules in Step 5; and update time and feedback for the interactive bi-level scheduling in Step 6. This cyclic system-level maintenance scheduling based on APB is shown in Fig. 3.

By applying MAM–APB strategy dynamically, plant managers can obtain the real-time opportunistic maintenance schedule for batch production cycle by cycle. To summarize, this interactive bi-level methodology decides the system-level PM arrangements based on both production data flow and maintenance data flow with three steps:

- (1) *MAM scheduling in the machine-level scheduling:* flexible PM intervals considering maintenance effects are scheduled based on machine degradation. These intervals will support the system-level scheduling, and be updated with APB decision feedbacks for the next cycle.
- (2) *APB programming in the system-level scheduling:* based on PM intervals and batch orders, PM actions are scheduled to be advanced or postponed according to real-time cost saving analysis. The outputs of the system-level scheduling include PM combinations G_u , set-up time points tb_u and breakdown durations for PM $T_{pu\max}$.
- (3) *System-level performance evaluation:* the total system maintenance cost is evaluated by system-level outputs. Let c_{sj} be the set-up cost rate, C_{pij} be the cost of PM action, and C_{fij} be the cost of minimal repair. The total system maintenance cost with U batches can be evaluated by:

$$ETC = \sum_{u=0}^{U-1} \delta(G_u) T_{pu\max} \left(\sum_{j=1}^J c_{sj} \right) + \sum_{i=1}^I \sum_{j \in G_u} C_{pij} + \sum_{i=1}^I \sum_{j=1}^J \left(C_{fij} \int_0^{T_{oj}^{Updated}} \lambda_{ij}(t) dt \right) \tag{23}$$

In Eq. (23), it can be found that the total system maintenance cost consists of three cost parts: the first part is the total set-up cost for sequential batches; the second part is the total PM cost for all machines; the third part is the total minimal repair cost in sequential maintenance cycles. It should be noticed that $T_{oj}^{Updated}$ means the actual PM intervals according to APB programming: PM advancement makes $T_{oj}^{Updated} = T_{oj}^* - (t_{ij} - tb_u)$, while PM postponement makes $T_{oj}^{Updated} = T_{oj}^* + (tb_{u+1} - t_{ij})$.

Moreover, traditional maintenance-driven opportunistic maintenance policies calculate the cost-savings of all possible PM combinations at every decision time. With the number of machines grows, possible PM combinations will increase exponentially, which means the system-level scheduling will suffer from intractable complexity. As a dynamic method, APB can provide decision outputs at each set-up time point according to real-time cost saving analysis. Therefore, production-driven opportunistic maintenance based on MAM–APB methodology can effectively handle a manufacturing system with growing machine number and changeable batch orders.

4. Numerical results and discussion

To validate MAM–APB methodology, a complex manufacturing system consists of seven various machines is considered here, as shown in Fig. 1. For dynamically scheduling production-driven opportunistic maintenance, both maintenance data flow and production data flow are synthetically collected in a hydraulic steering factory.

In the maintenance data flow, the reliability of each machine is formulated by a Weibull failure probability function $\lambda_{ij}(t) = (m_j/\eta_j)(t/\eta_j)^{m_j-1}$, which has been widely used to fit repairable equipment in electronic and mechanical engineering. Relative parameters are estimated by maintenance engineers and presented in Table 1.

In the production data flow, batch orders are processed through this 7-unit system based on fast-paced market demands. According to “pull” production pattern, new batch order would be placed

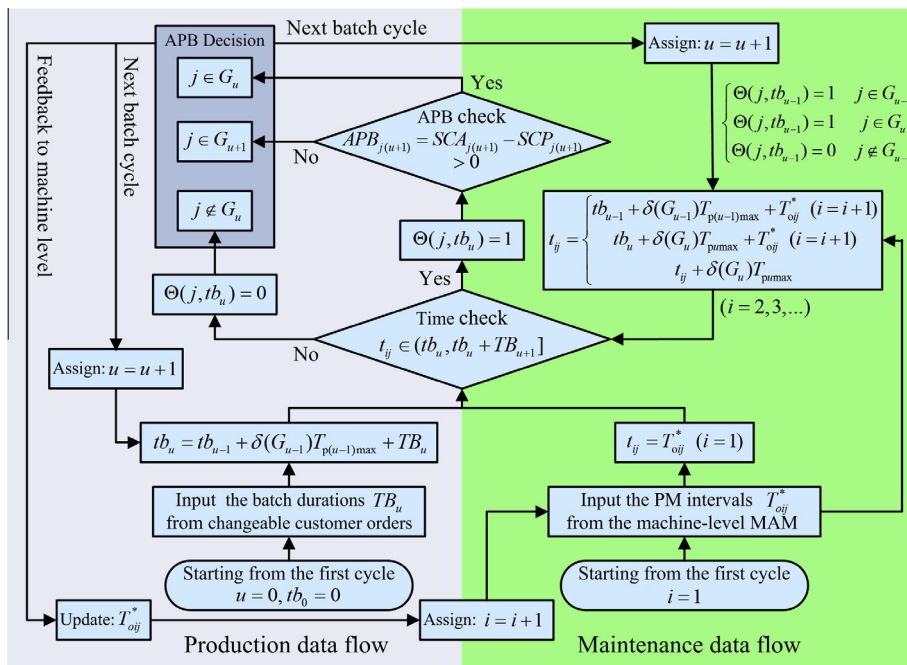


Fig. 3. Flowchart of system-level scheduling based on APB.

only at each decision time. The lot sizes of sequential batches are given in Fig. 4.

4.1. MAM results in machine level

In the machine-level scheduling, MAM takes advantages of multiple attribute value theory and imperfect maintenance to find maintenance requirements according to individual machine degradation. This study considers the situation that maintenance resources are available in the machine-level scheduling, while the system-level maintenance opportunities are caused by batch production characteristics. The machine-level outputs of PM intervals would be provided for the system-level scheduling at each set-up time point.

As concluded by Xia, Xi, Zhou, and Du (2012), these machine-level results reveal following conclusions: (1) the PM interval decreases while PM cycle increases, since the underlying hazard rate evolution becomes faster with the degradation process; (2) machine availability will be lower and maintenance cost will be higher as a machine ages due to the consideration of maintenance effects; and (3) ignoring the effects of a maintenance activity leads to less availability and extra cost, and MAM contributes to more practicality of PM intervals.

4.2. APB scheduling in system level

Our focus is on the effectiveness of the system-level scheduling for batch production. PM intervals from the machine-level scheduling usually lead to unnecessary production breakdown. Thus, at each set-up time point, APB strategy dynamically analyzes the cost savings of PM advancement and PM postponement, and then makes real-time schedules to satisfy no-disruption requirement. APB programming at the third set-up time point is taken as an example. Table 2 shows the scheduling results of expected PM actions in batch 4.

It can be seen that when batch 3 has been finished ($tb_3 = 8700$), PM opportunities appear and the time duration $TB_4 = 5000$ is ordered. At this decision time, the expected PM time points t_{ij} for M1, M2, M3, M5, M6 and M7 are originally scheduled to be operated in B_4 .

For example, according to the machine-level scheduling, PM of M1 was arranged to be performed at time 12,129. However, this PM action (original interval $T_{021}^* = 5829$) should be advanced ($T_{021}^A = 2400$, uptime of TB_3) or postponed ($T_{021}^P = 7400$, uptime of TB_3 and TB_4) in the system-level scheduling. Since $APB_{14} = -3269$ is negative, PM of M1 should be postponed to G_4 . Other calculated APB values at this set-up time point are given in Table 3.

It can be seen that M1's downtime cost can be reduced as $SCA_{14}^d = SCP_{14}^d = 20,000$, since PM is moved out of batch 4. If PM is advanced, the minimal repair cost $SCA_{14}^f = 3862$ can be saved, while the PM cost $SCA_{14}^p = 9287$ will be increased. In sum, the saved cost of PM advancement is $SCA_{14} = 14,575$, while PM postponement leads to $SCP_{14} = 17,844$. Therefore, $APB_{14} = -3269$ means PM postponement can save more maintenance cost. From Tables 2

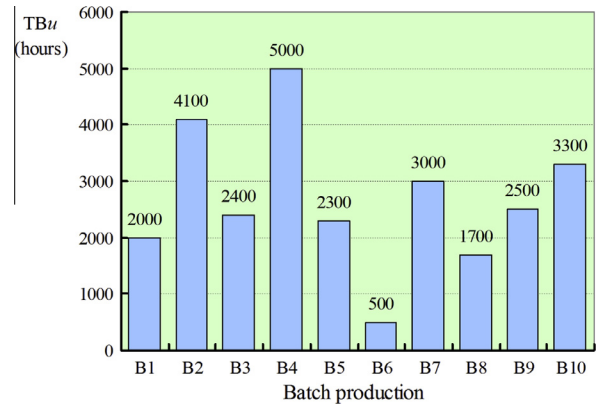


Fig. 4. Production data of batches.

and 3, the system-level PM arrangements at the third set-up time point could be obtained, and the same APB programming would be performed at other sequential set-up time points.

4.3. Results of production-driven opportunistic maintenance

Faced with sequential batch orders, APB strategy dynamically utilizes set-up works and analyzes the cost savings to reduce the total system maintenance cost. PM intervals from the machine level and various batch orders are pulled to make opportunistic maintenances cycle by cycle. And the APB decision feedback will be returned to update PM intervals for the next cycle. The results of production-driven opportunistic maintenance are presented in Tables 4 and 5.

In Table 4, a positive value indicates that the saved cost of PM advancement is greater, the machine should be maintained in advance, while a negative value makes PM to be postponed. It can be found that a larger C_{fij} promotes PM advancements to reduce cumulative failure risks, while a larger C_{pij} leads to PM postponements to avoid extra maintenance activities. In summary, APB chooses every PM adjustment with a large cost saving for every machine at each set-up time point. Correspondingly, Table 5 shows the complete layout of these opportunistic maintenances.

4.4. Effectiveness of MAM–APB methodology

To validate the proposed bi-level methodology of production-driven opportunistic maintenance for batch production, we investigate the cumulative cost-saving achieved by APB programming in each batch. Furthermore, we compare the expected total system maintenance cost (ETC) of MAM–APB scheduling with other opportunistic maintenance policies to show the significant cost reduction. The cumulative cost-savings achieved in sequential batches are presented in Fig. 5.

Furthermore, three maintenance-driven opportunistic maintenance policies discussed by Xia, Xi, Zhou, and Lee (2012) are used to make a comparison with APB strategy in the horizon of 280,000 h: (1) individual maintenance mode (IMM): PM is conducted on a machine only when it reaches its original PM intervals; (2) simultaneous maintenance mode (SMM): when one of the machines reaches its intervals, PM actions are carried out on all machines; and (3) maintenance time window (MTW): one machine's PM arises PM opportunities of non-failed machines within that batch.

Besides, two traditional production-driven opportunistic maintenance policies are introduced in the comparison: (1) advanced maintenance mode (AMM): PM actions originally planned to be performed in the next batch are all shifted to the current set-up

Table 1 Maintenance data of machines.

M_j	(m_j, η_j)	(a_{ij}, b_{ij})	T_{pij}	T_{fij}	C_{pij}	C_{fij}	c_{dj}	c_{sj}
M1	(2.5, 10,000)	(0.04, 1.05)	200	600	6500	15,000	120	20
M2	(1.5, 8000)	(0.02, 1.035)	80	400	3000	7500	70	20
M3	(3, 12,000)	(0.03, 1.02)	150	700	4000	8000	60	20
M4	(2.8, 11,000)	(0.025, 1.06)	240	450	8000	10,000	100	20
M5	(1.8, 7000)	(0.04, 1.04)	100	300	5000	12,000	65	20
M6	(2.4, 13,000)	(0.035, 1.03)	200	360	2000	6000	75	20
M7	(3.2, 15,000)	(0.05, 1.025)	300	800	8500	18,000	140	20

Table 2
APB programming at the third set-up time point.

M_j	tb_3	TB_4	t_{ij}	$\Theta(j, tb_3)$	T_{oij}^A	T_{oij}^+	T_{oij}^P	APB_{j4}	$j \in G_3$	$j \in G_4$
M1	8700	5000	12,129	1	2400	5829	7400	-3269		Y
M2			12,994	1	2400	6694	7400	33	Y	
M3			13,664	1	2400	7364	7400	-6332		Y
M4			16,638	0	-	-	-	-	-	-
M5			10,985	1	2400	4685	7400	5762	Y	
M6			13,221	1	2400	6921	7400	-2332		Y
M7			9474	1	8500	9274	13,500	6490	Y	

Table 3
Cost saving analysis at the third set-up time point.

j	SCA			SCA_{j4}	SCP			SCP_{j4}	APB_{j4}
	SCA_{j4}^d	SCA_{j4}^f	SCA_{j4}^p		SCP_{j4}^d	SCP_{j4}^f	SCP_{j4}^p		
1	20,000	3862	9287	14,575	20,000	3516	1380	17,844	-3269
2	4000	4712	5367	3345	4000	974	286	3312	33
3	6000	1931	8273	-342	6000	29	19	5990	-6332
4	-	-	-	-	-	-	-	-	-
5	4500	4376	4761	4115	4500	7982	1835	-1647	5762
6	11,000	1316	3768	8548	11,000	249	129	10,880	-2332
7	36,000	940	774	36,166	36,000	8984	2660	29,676	6490

Table 4
APB results in sequential batch cycles.

APB (cost)	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
M1		-9204		-3269			-2725			4022
M2			2262	33	1865			1746		2790
M3			392	-6332				105		-902
M4			-110				4033			4555
M5		-78		5762	5934		3687	1758	-1640	
M6			526	-2332				336		151
M7				6490			1951			2250

Table 5
Results of production-driven opportunistic maintenance.

Schedule (time point)	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
	tb1 2000	tb2 6100	tb3 8700	tb4 14,000	tb5 16,500	tb6 17,000	tb7 20,300	tb8 22,200	tb9 24,700	tb10 28,300
M1		Y		Y			Y		Y	-
M2		Y	Y	Y			Y		Y	-
M3		Y		Y			Y			Y
M4			Y			Y			Y	-
M5		Y	Y	Y		Y	Y		Y	-
M6		Y		Y			Y		Y	-
M7			Y			Y			Y	-

time point; (2) postponed maintenance mode (PMM): all the PM activities are delayed to the end of that batch. The total system maintenance costs of APB programming and above five policies are shown in Fig. 6.

Based on the results in Fig. 6, it can be found that *ETC* of MTW policy is 1,273,728. This is the lowest in maintenance-driven opportunistic maintenance policies, and MTW has been proven as a cost-effective strategy (*ETC*-saving rate 46.06% with IMM and 10.82% with SMM). However, huge downtime cost caused by production interruptions increases *ETC* value. Thus, production-driven opportunistic maintenance policies are more suitable for batch production. Here we can see that *ETC* of APB policy is 487,542, which is the most cost-effective system-level strategy (*ETC*-saving rate 16.71% with PMM and 10.86% with AMM).

In the general sense, MAM-APB methodology can efficiently eliminate unnecessary production breaks, achieve significant cost

reduction and overcome complexity of system scheduling with various maintenance data flow and production data flow. Different manufacturing systems with various machine reliabilities and changeable batch orders would lead to different *ETC*-saving rates. However, the mechanism of APB programming can ensure the dynamic scheduling performance. On the one hand, with sequential PM advancement or postponement, this production-driven opportunistic maintenance strategy eliminates unnecessary production breaks by utilizing set-up opportunities between successive batches. Huge downtime cost saving ensures that *ETC* of APB policy is much lower than those of maintenance-driven opportunistic maintenance policies (e.g. IMM, SMM and MTW). On the other hand, APB programming dynamically compares cost savings and chooses the PM adjustment with $\text{Max}\{SCA_{ju}, SCP_{ju}\}$ at each set-up opportunity. This sequential decision-making process ensures that the maximization of *ETC*-saving rate can be achieved, which

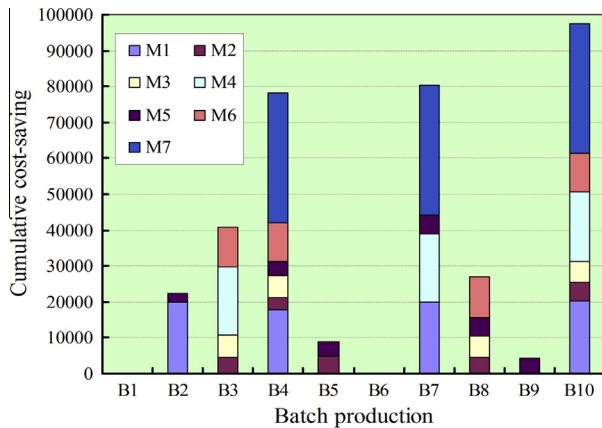


Fig. 5. Cumulative cost-saving in sequential batches.

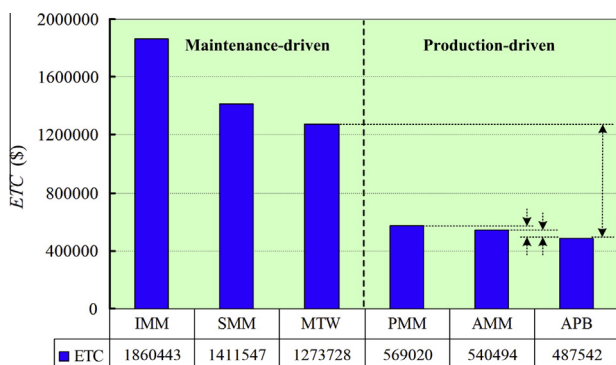


Fig. 6. Results comparison of opportunistic maintenance policies.

makes APB more effective than traditional production-driven opportunistic maintenance policies (e.g. AMM and PMM). In sum, APB method achieves significant system cost reduction because it not only considers batch production characteristics but also makes the PM adjustment according to the larger cost saving for every machine at each set-up time point.

5. Conclusions

In this paper, a production-driven opportunistic maintenance policy for batch production is proposed based on MAM-APB scheduling method. This bi-level maintenance strategy systematically considers not only individual machine degradation, but also batch production characteristics. Other than classical maintenance-driven opportunistic maintenance, the new methodology utilizes set-up opportunities to eliminate unnecessary production breaks, achieve significant cost reduction and overcome complexity of system scheduling. The developed MAM method is used to obtain maintenance intervals based on availability maximization and cost minimization. Furthermore, APB strategy dynamically analyzes the cost savings of PM advancement and PM postponement, and then makes real-time schedules to satisfy no-disruption requirement and reduce system maintenance cost. The cost savings achieved by applying this bi-level maintenance strategy have been demonstrated through a case study of seven-unit system. Results indicate that the total system maintenance cost saving achieved by APB strategy is much higher than traditional opportunistic maintenance policies. It can be concluded that proposed MAM-APB methodology is a viable and effective policy to reduce system maintenance cost for batch production.

Further research is required for improving the industrial implementation and demonstration of the newly proposed APB policy. Although this production-driven opportunistic maintenance can significantly reduce the system maintenance cost, how to rapidly react to practical changes and updates of batch orders needs to be studied, especially in the global competitive market. Furthermore, how to introduce the limitation of maintenance resources into the MAM-APB strategy will be investigated in future studies.

Acknowledgements

The authors would like to thank the editor and three anonymous referees for their remarkable comments. The research is funded partially by Programme of Introducing Talents of Discipline to Universities (B06012), Foundation for Innovative Research Groups of the National Natural Science Foundation of China (50821003), National Natural Science Foundation of China (51075277, 51275558), Program for New Century Excellent Talents in University (NCET-11-0328) and National Science Foundation Award # 0825789 at the NSF Sponsored I/UCRC Center for Intelligent Maintenance Systems at the University of Michigan.

References

- Allaouia, H., Lamourib, S., Artibab, A., & Aghezzaf, E. (2008). Simultaneously scheduling n jobs and the preventive maintenance on the two-machine flow shop to minimize the makespan. *International Journal of Production Economics*, 112(1), 161–167.
- Anzanello, M. J., Albin, S. L., & Chaovalitwongse, W. A. (2012). Multicriteria variable selection for classification of production batches. *European Journal of Operational Research*, 218(1), 97–105.
- Bedford, T., Dewan, I., Meilijson, I., & Zitrou, A. (2011). The signal model: A model for competing risks of opportunistic maintenance. *European Journal of Operational Research*, 214(3), 665–673.
- Cardenas-Barron, L. E. (2009). On optimal batch sizing in a multi-stage production system with rework consideration. *European Journal of Operational Research*, 196(3), 1238–1244.
- Cassady, C. R., & Kutanoglu, E. (2005). Integrating preventive maintenance planning and production scheduling for a single machine. *IEEE Transactions on Reliability*, 54(2), 304–309.
- Chen, W. J. (2006). Minimizing total flow time in the single-machine scheduling problem with periodic maintenance. *Journal of Operational Research Society*, 57, 410–415.
- Chen, J. S. (2008). Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of Operational Research*, 190(1), 90–102.
- Chen, W. J. (2009). Minimizing number of tardy jobs on a single machine subject to periodic maintenance. *Omega*, 37(3), 591–599.
- Costa, C. A. B., Carnero, M. C., & Oliveira, M. D. (2012). A multi-criteria model for auditing a predictive maintenance programme. *European Journal of Operational Research*, 217(2), 381–393.
- Dekker, R., Wildeman, R. E., & van der Duyn Schouten, F. A. (1997). A review of multi-component maintenance models with economic dependence. *Mathematical Methods of Operational Research*, 45(3), 411–435.
- Doyen, L., & Gaudoin, O. (2004). Classes of imperfect repair models based on reduction of failure intensity or virtual age. *Reliability Engineering and System Safety*, 84(1), 45–56.
- Jiang, R., & Ji, P. (2002). Age replacement policy: A multi-attribute value model. *Reliability Engineering and System Safety*, 76(3), 311–318.
- Ji, M., He, Y., & Cheng, T. C. E. (2007). Single-machine scheduling with periodic maintenance to minimize makespan. *Computers & Operations Research*, 34(6), 1764–1770.
- Lee, W., Wu, C., Wen, C., & Chung, Y. (2008). A two-machine flow-shop make-span scheduling problem with deteriorating jobs. *Computers & Industrial Engineering*, 54(4), 737–749.
- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42, 314–334.
- Li, S. L. (1997). A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. *European Journal of Operational Research*, 102(1), 142–156.
- Liao, C. J., & Chen, W. J. (2003). Single-machine scheduling with periodic maintenance and nonresumable jobs. *Computers & Operations Research*, 30(9), 1335–1347.
- Liao, C. J., Shyr, D. L., & Lin, C. H. (2005). Makespan minimization for two parallel machines with an availability constraint. *European Journal of Operational Research*, 160(2), 445–456.

- Li, D. C., Liu, C. W., & Chen, T. L. (2012). Recursive operation time maximization model for the maintenance of power generation equipment. *Computers & Operations Research*, 39(5), 1117–1121.
- Mellouli, R., Sadfi, C., Chu, C. B., & Kacem, I. (2009). Identical parallel-machine scheduling under availability constraints to minimize the sum of completion times. *European Journal of Operational Research*, 197(3), 1150–1165.
- Mobley, R. K. (2002). *An introduction to predictive maintenance*. Butterworth-Heinemann, New York: Elsevier Science.
- Okogbaa, O. G., Otieno, W., Peng, X., & Jain, S. (2008). Transient analysis of maintenance intervention of continuous multi-unit systems. *IIE Transactions*, 40(10), 971–983.
- Pham, H., & Wang, H. (1996). Imperfect maintenance. *European Journal of Operational Research*, 94(3), 425–438.
- Rachaniotis, N. P., & Pappis, C. P. (2008). Preventive maintenance and upgrade system: Optimizing the whole performance system by components' replacement or rearrangement. *International Journal of Production Economics*, 112(1), 236–244.
- Ruiz-Castro, J. E., & Li, Q. L. (2011). Algorithm for a general discrete k -out-of- n : G system subject to several types of failure with an indefinite number of repairpersons. *European Journal of Operational Research*, 211(1), 97–111.
- Sun, K., & Li, H. (2010). Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines. *International Journal of Production Economics*, 124(1), 151–158.
- Tan, Z. Y., Chen, Y., & Zhang, A. (2011). Parallel machines scheduling with machine maintenance for minsum criteria. *European Journal of Operational Research*, 212(2), 287–292.
- Topal, E., & Ramazan, S. (2010). A new MIP model for mine equipment scheduling by minimizing maintenance cost. *European Journal of Operational Research*, 207(2), 1065–1071.
- Wang, S., & Liu, M. (2013). A branch and bound algorithm for single-machine production scheduling integrated with preventive maintenance planning. *International Journal of Production Research*, 51(3), 847–868.
- Wildeman, R. E., Dekker, R., & Smit, A. C. J. M. (1997). A dynamic policy for grouping maintenance activities. *European Journal of Operational Research*, 99(3), 530–551.
- Xia, T., Xi, L., Zhou, X., & Du, S. (2012). Modeling and optimizing maintenance schedule for energy systems subject to degradation. *Computers & Industrial Engineering*, 63(3), 607–614.
- Xia, T., Xi, L., Zhou, X., & Lee, J. (2012). Dynamic maintenance decision-making for series-parallel manufacturing system based on MAM-MTW methodology. *European Journal of Operational Research*, 221(1), 231–240.
- Zhou, X., Xi, L., & Lee, J. (2009). Opportunistic preventive maintenance scheduling for a multi-unit series system based on dynamic programming. *International Journal of Production Economics*, 118(2), 361–366.