Probability in the Engineering and Informational Sciences, **27**, 2013, 209–235.
doi:10.1017/S0269964812000423

SINGLE-MACHINE MULTIPLE-RECIPE PREDICTIVE MAINTENANCE

YIWEI CAI
PayPal, Inc., 7700 W. Parmer Lane, Austin, TX 78729
E-mail: ewaycai@gmail.com

JOHN J. HASENBEIN, ERHAN KUTANOGLU, MELODY LIAO
Graduate Program in Operations Research and Industrial Engineering, Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712
E-mails: jhas@mail.utexas.edu; erhank@mail.utexas.edu; m.liao@mail.utexas.edu

This paper studies a multiple-recipe predictive maintenance problem with M/G/1 queueing effects. The server degrades according to a discrete-time Markov chain and we assume that the controller knows both the machine status and the current number of jobs in the system. The controller's objective is to minimize total discounted costs or long-run average costs which include preventative and corrective maintenance costs, holdings costs, and possibly production costs. An optimal policy determines both when to perform maintenance and which type of job to process. Since the policy takes into account the machine's degradation status, such control decisions are known as predictive maintenance policies. In the single-recipe case, we prove that the optimal policy is monotone in the machine status, but not in the number of jobs in the system. A similar monotonicity result holds in the two-recipe case. Finally, we provide computational results indicating that significant savings can be realized when implementing a predictive maintenance policies instead of a traditional job-based threshold policy for preventive maintenances.

1. INTRODUCTION

The optimal timing of maintenance decisions is an important issue in a variety of manufacturing systems. In a conventional preventive maintenance policy, it is typically assumed that the decision maker knows only when the last maintenance was performed, and the failure distribution for the machine or part of interest. Thus, conventional preventive maintenance policies are often one of two types: calendar-based or job-based. In a calendar-based policy, the preventive maintenance is performed if the machine has been running for a certain amount of time, for example, 10 hours, since the last preventative maintenance (PM) or corrective maintenance (CM). If the machine fails before the specified maintenance time, a CM is executed to repair the machine. In a job-based policy, a PM is executed when the machine finishes certain number of jobs since the last PM or CM. The goal then is to find the optimal calendar- or job-based threshold which determines when a PM is performed. These

© Cambridge University Press 2013 0269-9648/13 $25.00

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

policies are simple to implement and are popular in practice. However, in many manufacturing sectors, it is now possible for the controller to obtain much more knowledge about the internal state of a machine, meaning that the machine's condition can be determined more precisely. This information can also be used to make a better prediction about the next failure time. PM policies that incorporate the current machine state in this way are often referred to as "predictive maintenance" policies.

Note that traditional PM policies suffer from several potential problems. First of all, a machine may wear differently by processing different products or recipes, which is very common in semiconductor manufacturing. In semiconductors, even first wafer effects and cleaning may change the machine state [2]. Thus, it is difficult to specify the PM interval in a complex multiple-product environment. Secondly, even if a PM interval is specified, it may cause unnecessary PM costs and still miss the chance to do a PM. For example, a PM may be performed even when the machine is still in a very good state, and thus a PM cost is incurred and machine availability is unnecessarily decreased. Similarly, when the machine is very close to failure, a PM may not be executed simply because it is not the "scheduled" time to do a PM task. Clearly, such problems can be reduced by more closely monitoring the state of the machine to determine how close it is to failure. This is the idea behind automotive "maintenance minder" systems that monitor the oil quality to indicate when an oil change should be performed. Note that the traditional oil change policy is a combination of calendar-based (next oil change in 3 months) and "job"-based (next oil change in 3000 miles) policy, since it recommends a PM whenever the first of two thresholds is reached.

Considering the potential improvements, many manufacturing companies are seeking to develop condition-based predictive maintenance policies. Traditionally, predictive maintenance is costly and it is very difficult to monitor machine status continuously. Thanks to rapid technological progress, it is now possible to monitor the system continuously with much lower costs [9] in many sectors. Furthermore, as demonstrated in [11], condition-based predictive maintenance can greatly improve system safety.

Another potential problem with traditional PM policies is that they do not consider the current workload at the machine. If it is costly to have a large amount of work in process (WIP) in a system, then a good policy should consider both the current machine state and the WIP in the system. In this paper, we present a model of a single machine that is subject to degradation over time and which must undergo periodic maintenance. Our cost structure implies that the timing of maintenance is important both to reduce holding costs, and to avoid unscheduled maintenance. In particular, we study an M/G/1 single-machine predictive maintenance problem within a semi-Markov decision process (SMDP) framework.

Section 2 gives a brief literature review. Section 3 presents the one-recipe predictive maintenance model under the total discounted cost objective and provides structural results regarding the optimal policy. The two-recipe version of the problem is explored in Section 4. In Section 5 numerical examples are given which illustrate characteristics of the optimal policy in the discounted case. An average cost comparison of standard PM policies versus predictive maintenance policies is provided in Section 6. Finally, conclusions are given in Section 7.

2. LITERATURE REVIEW

The literature on maintenance scheduling is vast. Apart from the two review papers mentioned below, we confine our review to the research most closely related to our model. McCall [10] surveys scheduling policies for a stochastically failing machine. This survey identifies some common structure and clarifies the relations among various maintenance

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

policies. Dekker [5] provides an overview of applications of maintenance optimization models, and delivers a general evaluation of the value of maintenance optimization models as tools for management.

It should be noted that the majority of existing papers do not consider the interaction between the PM policy and production scheduling. We next summarize some studies which do incorporate both of these elements. In [15], Van der Duyn Schouten and Vanneste assume a constant demand rate and that incoming jobs are always available. Their PM policy considers both the age of the machine and the finished goods inventory level. The objective is to find a policy that minimizes the average "lost demand" per unit time. Some structural properties of the optimal policy are proved, and a numerical study is performed. Iravani and Duenyas [7] consider a single-machine single-product make-to-stock production system. The integrated decision of maintenance and production is formulated as a Markov decision process (MDP). In their problem setting, the production rate is affected by the machine degradation process. The objective is to find a joint production and PM policy to minimize the total average cost per unit time. A numerical study shows that the optimal policies have a very complicated structure and the optimal policies for some examples are not of threshold form. Exact and approximate methods are presented to evaluate the performance of the policy.

Sloan and Shanthikumar [14] consider multiple-product problems in semiconductor manufacturing. In their problem setting, different products have the same degradation effect on the machine, but the machine status may affect the yield of various products differently. They propose an MDP model and explore its structural properties. They note that the deterioration of the machine in the etching process in semiconductor manufacturing is not influenced by the product types. The reason is that the processing of one product type does not generate more particles, which cause the machine to degrade, than any other product type. However, this is not generally true for other processes in semiconductor manufacturing. As has been observed by many companies, different products or recipes may affect machine degradation differently, which is one of the problems studied in this paper.

Cassady and Kutanoglu [4] study an integrated model for job and PM operations scheduling. A complete enumeration approach is proposed and tested against small problems to obtain some insights. The main insight in this work is that the integrated model provides benefits relative to optimal age-based PM policies combined with the optimal schedule even for scheduling-only objectives such as weighted tardiness and even for a single machine. Ruiz, García-Díaz, and Maroto [12] propose tools to implicitly consider different preventive maintenance policies for a flowshop problem. Six existing heuristic and metaheuristic methods are revised and evaluated. Their work demonstrates the significance of taking into consideration preventive maintenance in job sequencing.

Zhou et al. [18] consider incorporating manufacturing system reconfiguration into preventive maintenance decisions for improved system performance in terms of reduced total cost, which is composed of maintenance cost, reconfiguration cost, and a penalty for unfulfilled production goals. An integrated reconfiguration and age-based maintenance policy is applied to a parallel–serial manufacturing system. A simulation-based heuristic optimization procedure is used to estimate and minimize the expected total cost. Their work shows that the manufacturing system could have a higher probability of fulfilling production requirements at a lower cost under the integrated PM policy. Yang, Djurdjanovic, and Ni [16] consider the problem of joint scheduling of maintenance and throughput adjustment operations in a manufacturing system. They assume that the machine is either producing products or it is being maintained, and that the machine is not idling at any point, that is, incoming jobs are always available. The objective of joint scheduling of PM and

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

throughput changing operations is to slow down more degraded machines or accelerate freshly maintained machines by adjusting the throughput rates of different machines. In this way, the production targets are met and maintenance operations can be adjusted so as to minimize the total cost. A simulation-based method and a genetic algorithm are proposed to search for the best schedule.

None of the papers cited above consider queueing effects and most of them assume that incoming jobs are always available. A few of the papers below consider both PM policy and production with queueing effects. Hsu [6] studies the maintenance problem for an M/G/1 production system. Analytical results are presented for the M/M/1 queue with an exponential breakdown rate. However, again in this problem setting, the PM policy used is a job-based policy.

The model in this paper is partially motivated by Chapter 3 of Yao's Ph.D. dissertation [17] (which apparently has not been published in another format). He considers an M/G/1 queueing system that produces a single product and develops an MDP model to derive optimal policies. The state space in this model has two dimensions: (1) the current WIP, and (2) the number of jobs that have been processed since the last PM or CM. At the completion time of a job, the machine has a certain probability of failing. If the machine does not fail, the number of jobs that have been processed since the last PM or CM is increased by one. Some structural properties of the optimal policy are analyzed and the existence of an optimal threshold policy is proved, but no numerical studies related to this part of his dissertation are performed. To some extent, the number of jobs that have been processed since the last PM or CM may be regarded as a surrogate for the machine status. However, this quantity does not use the information of the machine status explicitly, and it can not account for the case where the machine may deteriorate over multiple steps. Moreover, his model does not consider the multiple-product case, where different products may affect the machine differently.

Finally, another closely related paper is Kaufman and Lewis [8]. The state space in their paper differs from Yao's model above, in that both the current WIP and the machine status are known to the controller. They consider a model in which repairs (of random duration) may be used to restore the machine, and another model in which instantaneous machine replacements are allowed. An MDP framework is then used to obtain structural results for the optimal policy. Furthermore, they provide examples that demonstrate the non-monotonicity of optimal policies (we observe similar phenomena in our model). Our model is different in that we consider non-exponential service times, systems with multiple products, and a different degradation mechanism.

We believe that our paper is the first to examine a predictive maintenance problem with queueing effects, and multiple types of jobs. Our numerical results demonstrate that the optimal control for such a model may be counterintuitive in a number of ways. Using a simulation model, we are also able to explicitly quantify the savings possible when using predictive maintenance instead of standard PM and scheduling policies.

3. THE SINGLE-RECIPE PREDICTIVE MAINTENANCE MODEL

3.1. Discounted Cost SMDP Model

An M/G/1 queueing system with an unreliable machine that produces a single product, or recipe, is considered in this section. In traditional queueing parlance, a "single-recipe" problem is the same as a queueing model with a single class of jobs, that is, all jobs are stochastically equivalent in terms of their processing requirements and their effect on the machine. Jobs are assumed to arrive to the system according to a Poisson process with

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

rate λ. Service times are i.i.d. random variables drawn from a general distribution. This describes a typical M/G/1 queue. However, in our model the machine also has a state associated with it which we call the machine status. The machine status s takes values in a set $\{0, 1, 2, \ldots, M\}$. State 0 indicates that the machine has just been maintained and is in the "new" state. State M indicates that the machine has failed and it then requires a CM. The controller has the option to repair the machine before it reaches state M. In this case, the action is called a PM. The CM times are assumed to be i.i.d. random variables, drawn from a known continuous distribution. Similarly, the PM times are i.i.d. random variables, which in general have a distribution (potentially) different from the CM distribution. The service and maintenance distributions are assumed to have finite first moments.

The machine status is constant between job completion times. After a job completion the machine status may be the same as in the previous service time or it may change (degrade) to a worse status, if no PM or CM is performed. We assume therefore that the status, as sampled after a job completes processing, evolves according to a discrete time Markov chain with known transition probabilities. If a PM or CM is performed after a job is processed, then the status of the machine after the maintenance is "new", that is, $s = 0$. The system controller is allowed to take action at decision points that occur just after the completion of processing. The controller may either continue processing, if there are jobs in the queue, or perform a PM, if the machine has not failed. If the machine has just failed the controller is forced to perform a CM. Of course, if there are no jobs in the system, the controller must wait for an arrival to continue processing. When the queue is empty and the machine is idle, that is, a PM is not performed, the next decision epoch is at the arrival of the next job. If a PM is performed when the queue is empty, and a PM is performed, the next decision epoch is at the completion time of the PM.

Finally, there are costs associated with various actions and system events. We assume that there are costs for performing PMs and CMs and that there is a holding cost keeping jobs in the system. The overall objective for the controller is to minimize total discounted costs over an infinite horizon. Clearly, this problem can be formulated as discounted cost SMDP. Below we introduce the notation and details for such a formulation.

In the SMDP we sample the state and make decisions when a job completes processing, or when there is an arrival to an empty system. The system state is denoted by (w, s), and the state space is $Z_+ \times S$, where Z_+ denotes the set of non-negative integers. If the machine fails, that is, machine state is M, then a CM (denoted by C) must be performed. If the machine status is not M, then the controller may choose from one of three possible actions:

P : perform a PM,
A : continue processing the next job if the queue is not empty, or
W : wait for the next job if the queue is empty.

Thus the state-dependent action space is,

$$U_{w,s} = \begin{cases} \{P, A\} & \text{if } w > 0 \text{ and } s < M \\ \{P, W\} & \text{if } w = 0 \text{ and } s < M \\ \{C\} & \text{if } s = M. \end{cases}$$

Recall that the machine state may change from level r to any level s ($s \geq r$) with conditional probability denoted q_{rs} after the completion of a job.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

Before introducing the cost functions in the SMDP, we summarize the notation and parameters to be used in the model:

w_t : number of jobs in the systems at time t
s_t : machine state at time t
S : set of machine states, that is, $\{0, 1, 2 \ldots, M\}$, where M represents failure state
u_t : action chosen at decision epoch t (u may be used instead of u_t if it is not necessary to write t explicitly)
$u_{w,s}$: action chosen in state (w, s) in a stationary policy
$U_{w,s}$: set of the available actions in state (w, s)
q_{rs} : probability of transition from machine state r to s $(s \geq r)$
τ_c : time needed for a CM, a continuous r.v. with c.d.f. $G_c(\cdot)$
τ_p : time for a PM, a continuous r.v. with c.d.f. $G_p(\cdot)$
τ_a : job processing time, a continuous r.v. with c.d.f. $G_a(\cdot)$
T_u : expected time required to perform action u
λ : arrival rate of incoming jobs
h : holding cost per job per time unit in the system
TC_u : expected holding costs for jobs that enter the system during action u
c_p : cost for a PM
c_c : cost for a CM
β : discount factor
$I(A)$: indicator function of an event A.

Let Π be the set of non-anticipating policies. For a fixed policy $\pi \in \Pi$ we append a superscript to indicate the value of a quantity when the system is operated under π. For example, the system size at time t under π is denoted by w_t^π. In our SMDP model, the decision epochs are formed by the times at which jobs complete processing and arrival times to an empty system. We denote these decisions epochs as $t_1, t_2, \ldots$.

Under a policy π and for initial state (w, s), we consider the infinite horizon discounted cost of the form

$$J_\beta^\pi(w, s) = \lim_{N \to \infty} E_{(w,s)} \left\{ \int_0^{t_N} e^{-\beta t} g^\pi(w_t, s_t, u_t)\, dt \right\}, \tag{1}$$

where g^π is given by

$$g^\pi(w_t, s_t, u_t) = h w_t^\pi + c_p \delta(u_t^\pi = P) + c_c \delta(u_t^\pi = C),$$

and $\delta(\cdot)$ is the Dirac delta function. Later on, we also consider the infinite horizon average cost:

$$J^\pi(\boldsymbol{w}, s) = \limsup_{N \to \infty} \frac{1}{N} E_{(\boldsymbol{w},s)} \left\{ \int_0^{t_N} g^\pi(\boldsymbol{w}_t, s_t, u_t)\, dt \right\}.$$

Let J_β be the optimal cost function, defined by

$$J_\beta(w, s) = \inf_{\pi \in \Pi} J_\beta^\pi(w, s),$$

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

with the optimal average cost function defined analogously. The discounted cost function satisfies the following Bellman equations:

$$J_\beta(w,s) = \min\{Q^u_\beta(w,s) \,|\, u \in U_{w,s}\} \tag{2}$$

for all (w,s) in the state space. $Q^u_\beta(w,s)$ is the so-called "cost-to-go" corresponding to action $u \in U_{w,s}$. The derivation of these functions are similar to [17], and thus are omitted here. Furthermore, in the remainder of this section, we drop the β subscript for simplicity, reviving it when the average cost case is examined. The formulas for $Q^u(w,s)$ are given next:

1. For $w > 0, s \neq M$,

$$Q^A(w,s) = hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A + \sum_{k=0}^{\infty} \left\{ \int_0^\infty e^{-\beta t} p_k(\lambda t) dG_a(t) \sum_{j=s}^{M} q_{sj} J(w-1+k, j) \right\} \tag{3}$$

$$Q^P(w,s) = c_p + hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_p(t) + TC_P + \sum_{k=0}^{\infty} \int_0^\infty e^{-\beta t} p_k(\lambda t) dG_p(t) J(w+k, 0), \tag{4}$$

where $p_k(\lambda t) = e^{-\lambda t}(\lambda t)^k/k!$. The first term on the right-hand side (RHS) of (3) is the one-stage expected holding cost of the w jobs already in the system, and TC_A, which is given below, is the expected holding cost of new jobs arriving until the next decision epoch. The third term is the discounted future costs, which depends on the state in the next epoch. Similarly, the first term on the RHS of (4) is the PM cost, the second and third terms are the expected total holding costs, and the last term accounts for future costs. The calculations of TC_C and TC_P are the same as in Section 3.3 of [17]:

$$TC_A = \int_0^\infty \frac{\lambda h}{\beta^2}[1-(1+\beta t)e^{-\beta t}] dG_a(t) \tag{5}$$

$$TC_P = \int_0^\infty \frac{\lambda h}{\beta^2}[1-(1+\beta t)e^{-\beta t}] dG_p(t). \tag{6}$$

2. For $w \geq 0, s = M$

$$Q^C(w,M) = c_c + hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_c(t) + TC_C + \sum_{k=0}^{\infty} \int_0^\infty e^{-\beta t} p_k(\lambda t) dG_c(t) J(w+k, 0), \tag{7}$$

and the corresponding TC_C is:

$$TC_C = \int_0^\infty \frac{\lambda h}{\beta^2}[1-(1+\beta t)e^{-\beta t}] dG_c(t). \tag{8}$$

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

3. For $w = 0, s \neq M$

$$Q^W(0,s) = \int_0^\infty e^{-\beta t}\lambda e^{-\lambda t}dtJ(1,s) = \frac{\lambda}{\lambda+\beta}J(1,s) \tag{9}$$

$$Q^P(0,s) = c_p + TC_P + \sum_{k=0}^{\infty}\int_0^\infty e^{-\beta t}p_k(\lambda t)dG_p(t)\cdot J(k,0). \tag{10}$$

Recall that when the queue is empty, the next decision epoch is either the arrival of the next job if W is chosen at $(0,s)$ or the completion of a PM if P is chosen at $(0,s)$.

3.2. Structural Results

The main result in this section is that there exists an optimal threshold policy, also know as a control limit policy, in the one-recipe problem. The threshold is relative to a fixed number of jobs in the system. In other words, for each w, there exists a threshold status s_w^* such that for this machine status, or worse, a PM should be performed. Otherwise, the machine should continue to process jobs, or idle if necessary. Clearly, such a result is useful in narrowing the search for optimal stationary policies.

An outline of the proof is as follows. First, $J(w,s)$ is proved to be non-decreasing with respect to w. Then, under some additional assumptions, $J(w,s)$ is proved to be non-decreasing with respect to s. Finally, Theorem 3.3 implies that an optimal threshold policy exists for this problem. Below, $J_m(w,s)$ is defined to be the mth-stage value function in value iteration, and $Q_m^\mu(w,s)$ is defined to be the mth-stage cost-to-go function corresponding to $\mu \in \{C,P,A,W\}$ in the Bellman equation (2). Also, let $p_l(k,\lambda) \equiv \int_0^\infty e^{-\beta t}p_k(\lambda t)dG_l(t)$, where $l \in \{a,p,c\}$.

LEMMA 3.1: *The optimal cost function $J(w,s)$ is non-decreasing with respect to w, that is,*

$$J(w,s) \leq J(w+1,s), \tag{11}$$

for all s.

PROOF: The proof uses induction, along with value iteration. First, we show that $J_m(1,s) \geq J_m(0,s)$ for all $m \geq 1$ and $s < M$. Note that $J_{m+1}(1,s) \geq J_m(1,s)$ due to the fact that the one-stage cost is nonnegative and that the dynamic programming operator is monotone (see [3], Lemma 1.1.1 on page 7, Vol. II). Thus,

$$Q_{m+1}^A(1,s) \geq J_{m+1}(1,s) \geq J_m(1,s) = \frac{\lambda+\beta}{\lambda}Q_{m+1}^W(0,s) > Q_{m+1}^W(0,s).$$

The first inequality is due to the fact that

$$J_{m+1}(1,s) = \min\{Q_{m+1}^A(1,s), Q_{m+1}^P(1,s)\}.$$

Also, it is obvious that $Q_{m+1}^P(1,s) > Q_{m+1}^P(0,s)$, thus, $J_{m+1}(1,s) > J_{m+1}(0,s)$ for all $m \geq 0$.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

Assuming $J_m(w,s) \le J_m(w+1,s)$ for all $m \ge 1$, it can be shown that (11) holds for $w > 0$, $s < M$. Next, we have that

$$\begin{aligned} Q^A_{m+1}(w+1,s) &= h(w+1)\int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A \\ &\quad + \sum_{k=0}^{\infty}\left\{ p_a(k,\lambda) \sum_{j=s}^{M} q_{sj} \cdot J_m(w+k,j) \right\} \\ &\ge hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A \\ &\quad + \sum_{k=0}^{\infty}\left\{ p_a(k,\lambda) \sum_{j=s}^{M} q_{sj} \cdot J_m(w-1+k,j) \right\} \\ &= Q^A_{m+1}(w,s). \end{aligned}$$

An analogous argument shows that $Q^P_{m+1}(w,s)$ is also non-decreasing with respect to w. Thus (2) immediately implies that $J_m(w,s)$ is non-decreasing with respect to w when $s < M$. We omit the nearly identical argument showing that $J_m(w,M)$ is non-decreasing with respect to w.

Since the single-stage cost is non-negative, and the control constraint set is finite for every state, standard results (see, e.g., Proposition 3.1.5 of Section 3.1 in [3], Volume II) yield

$$J(w,s) = \lim_{m\to\infty} J_m(w,s).$$

Therefore, the monotonicity and induction applies to the function $J(w,s)$ also, establishing the result. ■

The following assumptions are used in the next structural result:

A-1 For any fixed $j \in M$, $q_{sj} < q_{s'j}$ for $\forall\, s < s' \le j$, that is, the probability of going to a given bad state j from a better state $s \le j$ increases as s gets worse.

A-2 $c_c > c_p$, that is, a CM costs more than a PM.

A-3 $\tau_c \overset{D}{=} \tau_p$.

Assumption A-1 is a common assumption in maintenance models (see Section 1.2 in Vol. II of [3]). It is essentially equivalent to the condition of an increasing failure rate for the machine. A-2 is also quite reasonable for any practical model. A-3, equality in distribution of the maintenance times, is perhaps the most restrictive and it is possible that it can be weakened. Similar assumptions are used in [17].

LEMMA 3.2: *Under assumptions A-1, A-2 and A-3, for every w and $u \in U_{w,s}$, $Q^u(w,s)$ and $J(w,s)$ are non-decreasing with respect to s.*

PROOF: Again, the result is proved by induction and value iteration.

1. As usual, assume that $J_0(w,s) = 0\ \forall\, w,\ s$. For $w > 0$, $s < s' < M$,

$$Q^A_1(w,s) = hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A = Q^A_1(w,s').$$

Similarly, $Q^P_1(w,s) = Q^P_1(w,s') < Q^C_1(w,M)$, where the inequality follows from assumptions A-2 and A-3. Also, it is clear that $J_1(0,s) = J_1(0,s')$ for $s < s' < M$. Therefore,

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

$J_1(w,s) = J_1(w,s') < J_1(w,M)$, for $s < s' < M$. Next we have

$$\begin{aligned} Q_2^A(w,s') &= hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A + \sum_{k=0}^{\infty} \left\{ p_a(k,\lambda) \sum_{j=s'}^{M} q_{s'j} \cdot J_1(w+k,j) \right\} \\ &\geq hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A + \sum_{k=0}^{\infty} \left\{ p_a(k,\lambda) \sum_{j=s}^{M} q_{s,j} \cdot J_1(w+k,j) \right\} \\ &= Q_2^A(w,s). \end{aligned}$$

To verify the inequality, we need only argue that the following expression is non-negative:

$$\begin{aligned} &\sum_{j=s'}^{M} q_{s'j} \cdot J_1(w+k,j) - \sum_{j=s}^{M} q_{s,j} \cdot J_1(w+k,j) \\ &\quad = \sum_{j=s'}^{M} (q_{s'j} - q_{s,j}) \cdot J_1(w+k,j) - \sum_{j=s}^{s'-1} q_{s,j} \cdot J_1(w+k,j). \end{aligned}$$

On the RHS, note that the coefficients $(q_{s'j} - q_{s,j})$ are positive by assumption A-1. Second, we must have

$$\sum_{j=s'}^{M} (q_{s'j} - q_{s,j}) = \sum_{j=s}^{s'-1} q_{s,j}$$

since the $q_{s,j}$ must sum to one for each s. Third, each term $J_1(w+k,j)$ in the first sum on the RHS is no smaller than every $J_1(w+k,j)$ term in the second sum, as shown above. These three observations together imply the desired non-negativity.

Next, we obviously have $Q_2^P(w,s) = Q_2^P(w,s')$, for $s < s' < M$. Thus,

$$\begin{aligned} J_2(w,s) &= \min\{Q_2^A(w,s), Q_2^P(w,s)\} \\ &\leq \min\{Q_2^A(w,s'), Q_2^P(w,s')\} \\ &= J_2(w,s'), \end{aligned}$$

where $s < s' < M$. Also, for $w > 0$ and $s < M$,

$$J_2(w,M) = Q_2^C(w,M) > Q_2^P(w,s) \geq J_2(w,s),$$

where again the inequality is due to assumptions A-2 and A-3. Thus, $J_2(w,M) > J_2(w,s)$ for $s < M$. Also, it is easy to see that $J_2(0,s) = J_2(0,s')$ for $s < s' < M$. Altogether, we have proved that for $m = 1$ and 2, $J_m(w,s)$ is non-decreasing in s, which accomplishes the first step of the induction.

2. Next, assume that for a fixed m, $Q_m^u(w,s)$ $(u \in U_{w,s})$ and $J_m(w,s)$ are non-decreasing with respect to s, then

$$\begin{aligned} &Q_{m+1}^A(w,s+1) - Q_{m+1}^A(w,s) \\ &\quad = \sum_{k=0}^{\infty} \left\{ \sum_{j=s+1}^{M} [q_{s+1,j} J_m(w+k-1,j)] - \sum_{j=s}^{M} [q_{s,j} J_m(w+k-1,j)] \right\} p_a(k,\lambda). \end{aligned}$$

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

The terms in the curly bracket can be rewritten as

$$\begin{aligned}
&\sum_{j=s+1}^{M} [q_{s+1,j} J_m(w+k-1,j)] - \sum_{j=s}^{M} [q_{s,j} J_m(w+k-1,j)] \\
&= \sum_{j=s+1}^{M} [(q_{s+1,j} - q_{s,j}) J_m(w+k-1,j)] - q_{s,s} J_m(w+k-1,s) \\
&\geq \sum_{j=s+1}^{M} [(q_{s+1,j} - q_{s,j}) J_m(w+k-1,s)] - q_{s,s} J_m(w+k-1,s) \\
&= \left(\sum_{j=s+1}^{M} q_{s+1,j} \right) J_m(w+k-1,s) - \left(\sum_{j=s}^{M} q_{s,j} \right) J_m(w+k-1,s) \\
&= 1 \cdot J_m(w+k-1,s) - 1 \cdot J_m(w+k-1,s) \\
&= 0.
\end{aligned}$$

The inequality is due to $(q_{s+1,j} - q_{s,j}) \geq 0$, and the fact that $J_m(w,s)$ is non-decreasing with respect to s for any fixed w. The second to last equation is due to $\sum_{j=s}^{M} q_{s,j} = \sum_{j=s+1}^{M} q_{s+1,j} = 1$, that is, for any machine state v, the next state must be in $\{v, v+1, \ldots, M\}$ if the action is A. Thus, $Q^A_{m+1}(w,s+1) - Q^A_{m+1}(w,s) \geq 0$, and $Q^A_{m+1}(w,s)$ is non-decreasing with respect to s. Notice also that in evaluating the original double summations we implicitly use the fact that $Q^A_{m+1}(w,s)$ is finite for all w and s. As above we have that $Q^P_{m+1}(w,s)$ is constant with respect to s.

Next, since

$$Q^W_m(0,s) = \frac{\lambda}{\lambda+\beta} \cdot J_m(1,s),$$

and $J_m(1,s)$ is non-decreasing with respect to s, $Q^W_m(0,s)$ is also non-decreasing with respect to s. Due to assumptions A-2 and A-3, $Q^P_m(w,s) \leq Q^C_m(w,M) = J_m(w,M)$ for all s. Thus, for any s,

$$\begin{aligned}
J_{m+1}(w,s) &= \min(Q^A_{m+1}(w,s), Q^P_{m+1}(w,s)) \\
&\leq Q^P_{m+1}(w,s) \\
&\leq J_{m+1}(w,M).
\end{aligned}$$

3. As argued in the proof of Lemma 3.1, we have that $Q^u(w,s) = \lim_{m\to\infty} Q^u_m(w,s)$ and $J(w,s) = \lim_{m\to\infty} J_m(w,s)$. Therefore, $Q^u(w,s)$ is non-decreasing with respect to s, which implies that $J(w,s)$ is also. ■

We are finally able to present our main result for the one-recipe case, which is that there exists an optimal threshold policy for this model.

THEOREM 3.3: *Under assumptions A-1, A-2, and A-3, there exists an optimal policy with the following form:* $\forall w$, $\exists\ s^*_w$, *s.t. for each system state* (w,s), *if* $s \geq s^*_w$, *then* P *is the optimal action; otherwise the optimal action is to continue processing* (A) *if the queue is not empty or wait* (W) *if the queue is empty.*

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

PROOF: Since $Q^P(w,s)$ is constant with respect to s, and $Q^u(w,s)$ is a non-decreasing function of s for $u \in \{A, W\}$, one can conclude that

$$Q^u(w,s) - Q^P(w,s) \leq Q^u(w,s+1) - Q^P(w,s+1), \forall s < M.$$

Thus, there exists an s_w^* s.t. $Q^u(w,s_w^*) - Q^P(w,s_w^*) \geq 0$. Otherwise, $s_w^* = M+1$ implying that A or W is always the optimal action. Also, due to the non-decreasing property of $Q^u(w,s)$, $\forall s' \geq s_w^*$, $Q^u(w,s') \geq Q^P(w,s')$. Thus when the WIP level is w, P is the optimal action if the machine state is greater than or equal to s_w^*. ■

4. TWO-RECIPE PREDICTIVE MAINTENANCE

In semiconductor manufacturing, different recipes may affect the machine degradation differently. Thus when different products, or recipes, are processed on the same machine, the corresponding decision model should take these effects into account. As such, we now consider an extension to the one-recipe model considered above. Specifically, we study a single-machine two-recipe predictive maintenance problem. The two recipes are called A and B, and for simplicity, there is no difference between these two recipes with respect to production parameters, for example, the arrival rate, processing distribution, and holding costs are all the same for both recipes. What differentiates the two recipes is their effect on the machine degradation. Otherwise, the model is similar to the one-recipe model considered earlier. It should be noted then that the controller's decision now includes two factors: when to perform a PM and which job (if any) to process next when the machine becomes free. Therefore, the model formulated below is a joint scheduling and PM control problem.

We now describe the model in more detail. Jobs from recipes A and B arrive according to independent Poisson processes, each with rate λ. Thus, the total arrival process is also Poisson with rate 2λ. Also, since a PM or a CM is related to the machine rather than to the recipe, the distributions of PM and CM times are assumed to be the same for both recipes. Let q_{rs}^a (q_{rs}^b) represent the conditional probability for a machine to degrade from state r to state s after processing a job from recipe A (B). Jobs from the two recipes cause different degrees of damage to the machine. To be specific, $q_{rs}^a \neq q_{rs}^b$ at least for some r, $s \in S$. Other settings of the SMDP model are quite similar to those in Section 3.1, and thus are omitted here.

Throughout this section, it is assumed without loss of generality that it costs more to process A than to process B. To be specific, it costs a non-negative amount \$$\Delta$ to process one job from recipe A and there is no cost to process jobs from recipe B. General structural results are provided first, and then some special cases and structural properties are presented for the case where recipe B is "superior" to recipe A, that is, type B jobs cause less damage to the machine and cost less to process than type A jobs do.

4.1. General Structural Properties

This section uses notation similar to that in Section 3. Let $\boldsymbol{w} = (w_a, w_b)$ and $w = w_a + w_b$, where w_a and w_b represent the non-negative WIP from recipe A and B, respectively. If $w > 0$ and $s < M$, the set of the available actions are $U_{\boldsymbol{w},s} = \{A, B, P\}$, where A (B) means to process jobs from recipe A (B). If $w = 0$ and $s < M$, then $U_{\boldsymbol{w},s} = \{W, P\}$. If $s = M$, then $U_{\boldsymbol{w},s} = \{C\}$.

Let J be the optimal cost function, which satisfies the following Bellman equations:

$$J(\boldsymbol{w}, s) = \min\{Q^u(\boldsymbol{w}, s) \,|\, u \in U_{\boldsymbol{w},s}\}. \tag{12}$$

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

In the remainder of this section, $p_l(k,\lambda) \equiv \int_0^\infty e^{-\beta t} p_k(\lambda t) dG_l(t)$, where $l \in \{c,p,a,b\}$. As before, we must calculate the cost-to-go functions for various state/action pairs. These calculations are detailed below.

1. For $w > 0$, $s < M$,

$$Q^P(\boldsymbol{w},s) = c_p + hw \cdot \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_p(t) + TC_P + \sum_{k=0}^{\infty}\sum_{k'=0}^{\infty} p_p(k,\lambda)p_p(k',\lambda)J(w_a+k, w_b+k', 0); \quad \textbf{(13)}$$

if $w_a > 0$,

$$Q^A(\boldsymbol{w},s) = \Delta + hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A + \sum_{k=0}^{\infty}\sum_{k'=0}^{\infty} p_a(k,\lambda)p_a(k',\lambda) \left\{ \sum_{j=s}^{M} q_{sj}^a J(w_a-1+k, w_b+k', j) \right\}; \quad \textbf{(14)}$$

if $w_b > 0$,

$$Q^B(\boldsymbol{w},s) = hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_B + \sum_{k=0}^{\infty}\sum_{k'=0}^{\infty} p_b(k,\lambda)p_b(k',\lambda) \left\{ \sum_{j=s}^{M} q_{sj}^b J(w_a+k, w_b-1+k', j) \right\}, \quad \textbf{(15)}$$

where

$$TC_A = TC_B = \int_0^\infty \frac{2\lambda h}{\beta^2}[1-(1+\beta t)e^{-\beta t}] dG_a(t) \quad \textbf{(16)}$$

$$TC_P = \int_0^\infty \frac{2\lambda h}{\beta^2}[1-(1+\beta t)e^{-\beta t}] dG_p(t). \quad \textbf{(17)}$$

2. For $w_a \geq 0, w_b \geq 0, s = M$

$$J(\boldsymbol{w},M) = c_c + hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_c(t) + TC_C + \sum_{k=0}^{\infty}\sum_{k'=0}^{\infty} p_c(k,\lambda)p_c(k',\lambda)J(w_a+k, w_b+k', 0). \quad \textbf{(18)}$$

The corresponding TC_C is:

$$TC_C = \int_0^\infty \frac{2\lambda h}{\beta^2}[1-(1+\beta t)e^{-\beta t}] dG_c(t). \quad \textbf{(19)}$$

3. For $w_a = w_b = 0, s < M$

$$Q^W(0,0,s) = \int_0^\infty e^{-\beta t} 2\lambda e^{-2\lambda t} dt \frac{1}{2}(J(1,0,s) + J(0,1,s)) = \frac{\lambda}{2\lambda+\beta} \cdot (J(1,0,s) + J(0,1,s)) \quad \textbf{(20)}$$

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

$$Q^P(0,0,s) = c_p + TC_P + \sum_{k=0}^{\infty}\sum_{k'=0}^{\infty} p_p(k,\lambda)\cdot p_p(k',\lambda)J(k,k',0). \quad \textbf{(21)}$$

Following a similar procedure as for Lemma 3.2, it is straightforward to prove the following lemma and thus the proof is omitted.

LEMMA 4.1: *Under assumptions A-1, A-2, and A-3, in Section 3, for any fixed* $\boldsymbol{w} = (w_a, w_b)$, $Q^u(\boldsymbol{w}, s)$ *(*$u \in U_{\boldsymbol{w},s}$*), and* $J(\boldsymbol{w}, s)$ *are non-decreasing with respect to* s.

Similarly, the next lemma can be proved using arguments similar to those in the proof of Lemma 3.1. Note also that both the lemmas above can also be proved using the sample path technique used to demonstrate Proposition 3.3 in [8]. Lemma 4.2 is only used later, when we investigate the average cost case.

LEMMA 4.2: *Under assumptions A-1, A-2, and A-3, in Section 3, for any fixed* s *and* w_a, $J(\boldsymbol{w}, s)$ *is non-decreasing in* w_b. *Similarly, for any fixed* s *and* w_b , $J(\boldsymbol{w}, s)$ *is non-decreasing in* w_a.

Finally, the following theorem can be proved by using an argument analogous to the proof of Theorem 3.3, by proving the recipe A and B cases, respectively.

THEOREM 4.3: *For any* $\boldsymbol{w}$, $\exists\, s^*_{\boldsymbol{w}}$, *s.t.* $\forall\, s^*_{\boldsymbol{w}} \le s < M$,

$$Q^P(\boldsymbol{w}, s) = \min\{Q^u(\boldsymbol{w}, s) \mid u \in U_{\boldsymbol{w},s}\},$$

which means P *is optimal when the machine status is at or above the threshold* $s^*_{\boldsymbol{w}}$.

This result is the main structural result for the two-recipe case, and it implies that a type of threshold policy is optimal. However, it should be noted that the threshold only indicates (via the machine state) when a PM should be performed. Unfortunately, it does not give any indication of the scheduling policy, that is, when to process recipe A versus recipe B.

4.2. One Superior Recipe

In this section, we provide an example in which recipe B appears to be "better" for the system than recipe A. In this case, a myopic optimizer would always choose to process recipe B jobs first. We show, however, that this greedy decision is not always optimal, that is, in some states it is better in the long-run to process recipe A first. So, we define recipe B to be "superior" to recipe A if the following two conditions are satisfied:

1. It costs more to process jobs of recipe A than jobs of recipe B, that is, $\Delta > 0$.
2. For each $s : q^a_{sj} > q^b_{sj}$ for all $j > s$ and $q^a_{ss} < q^b_{ss}$.

The second condition implies that recipe B causes less damage to the machine than recipe A does. Intuitively, it should be better to process recipe B than A whenever jobs from recipe B are available. However, the following counterexample shows that this is not always the case.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

Example: In this example, it costs $\Delta = \$0.5$ to process a job from recipe A, and nothing to process jobs from recipe B. In machine state s, the degradation probabilities are:

$$q^a_{sj} = \frac{1}{M-s+1}, \quad \forall j \geq s$$

$$q^b_{sj} = \begin{cases} \dfrac{0.49}{M-s}, & \forall j > s \\ 0.51, & j = s. \end{cases}$$

The holding cost is \$10 per job per unit time and is the same for both recipes. The PM cost is \$12, and the CM cost is \$20. The discount factor (β) is set to 0.05. The arrival rate (λ) for each type of job is 0.08. The deterministic processing time is 5 time units. A PM takes 8 time units exactly, and a CM takes 20 time units exactly. In order to solve the problem using linear programming, the state space has to be truncated. The maximum number of jobs allowed in the system is 20, that is, $w_a + w_b \leq 20$. The machine has six states ($s \in \{0, \ldots, 5\}$), and a CM is enforced if $s = 5$.

It takes 18.4 seconds to solve the problem in Xpress [1] using linear programming. The result shows that in most states, the optimal policy chooses recipe B over recipe A. However, if the machine state is 4, that is, one level before the failure state ($s = 5$), and if there are only one or two jobs from recipe B, then the optimal policy chooses to process jobs from recipe A. The structure of the optimal policy is as follows.

1. Wait for the next incoming job under the following conditions:
   (a) $s \leq 3$, $w_a = w_b = 0$.
2. Perform a PM under the following conditions:
   (a) $s = 4$, $w_a + w_b = 20$, and $w_a \leq 16$
   (b) $s = 4$ and $w_a + w_b = 0$.
3. Process a job from recipe A under the conditions:
   (a) $s \leq 4$ and $w_b = 0$
   (b) $s = 4$ and $w_b = 1$
   (c) $s = 4$, $w_b = 2$, and $w_a = 18$.
4. Other states: process a job from recipe B.

As can be seen from cases 3b and 3c, even when jobs from recipe B are available, the optimal policy still chooses to process jobs from recipe A. In order to check if such a counterintuitive result is due to the truncation of the state space, the allowable system size was varied from 20 to 40, and the pattern of the optimal policy does not change at all in this truncation range. Therefore, it is plausible that such a counterexample is not just an artifact of the truncation.

We now state a value function property in Theorem 4.4, which may be useful in proving structural policy results for the superior recipe case.

THEOREM 4.4: *In a two-recipe problem, suppose recipe B is superior over recipe A. Given s and w, if w_a, w_b, w'_a, and w'_b satisfy the following two conditions:*

(i) $w_a + w_b = w'_a + w'_b = w$,
(ii) $w_a \geq w'_a$ *and* $w_b \leq w'_b$,

then $J(w_a, w_b, s) \geq J(w'_a, w'_b, s)$.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

PROOF: The proof uses induction, along with value iteration. Suppose $J_m(\boldsymbol{w}, s)$ and $Q_m(\boldsymbol{w}, s)$ are the mth-stage functions in value iteration for state $(\boldsymbol{w}, s)$. Without loss of generality, it is assumed that it takes a non-negative amount \$$\Delta$ to process one job from recipe A, and nothing to process one job from recipe B.

1. First we set $J_0(\boldsymbol{w}, s) = 0$. The next step is to prove $J_1(w_a, w_b, s) \geq J_1(w'_a, w'_b, s)$ for any fixed s. For all w_i and w'_i $(i = a, b)$, we have that

$$\begin{aligned} Q_1^P(w_a, w_b, s) &= c_p + hw \int_0^\infty \frac{1 - e^{-\beta t}}{\beta} dG_p(t) + TC_P \\ &= Q_1^P(w'_a, w'_b, s). \end{aligned}$$

For $w_a > 0$, and $w_b > 0$,

$$Q_1^A(w_a, w_b, s) = \Delta + hw \int_0^\infty \frac{1 - e^{-\beta t}}{\beta} dG_a(t) + TC_A = Q_1^A(w'_a, w'_b, s),$$
$$Q_1^B(w_a, w_b, s) = hw \int_0^\infty \frac{1 - e^{-\beta t}}{\beta} dG_b(t) + TC_B = Q_1^B(w'_a, w'_b, s).$$

Thus, $J_1(w_a, w_b, s) = J_1(w'_a, w'_b, s)$ for $w_i \neq 0$, and $w'_i \neq 0$, where $i \in \{a, b\}$. It is easy to see that $TC_A = TC_B$ due to fact that the two recipes have the same processing distribution $(G_a(t) = G_b(t))$. Thus, TC_A and TC_B are interchangeable in the following:

$$\begin{aligned} Q_1^A(1, 0, s) &= \Delta + h \int_0^\infty \frac{1 - e^{-\beta t}}{\beta} dG_a(t) + TC_A \\ &> h \int_0^\infty \frac{1 - e^{-\beta t}}{\beta} dG_b(t) + TC_B \\ &= Q_1^B(0, 1, s), \end{aligned}$$

and $Q_1^P(0, 1, s) = Q_1^P(1, 0, s)$ for all s. Thus for any fixed s,

$$\begin{aligned} J_1(1, 0, s) &= \min\{Q_1^A(1, 0, s), Q_1^P(1, 0, s)\} \\ &\geq \min\{Q_1^B(0, 1, s), Q_1^P(0, 1, s)\} \\ &= J_1(0, 1, s). \end{aligned}$$

Therefore, $J_1(w_a, w_b, s) \geq J_1(w'_a, w'_b, s)$.

2. Suppose now that $J_m(w_a, w_b, s) \geq J_m(w'_a, w'_b, s)$ holds for a fixed m. The following shows that the inequality then also holds for $m + 1$. If $w_i > 0$ and $w'_i > 0$ (for $i \in \{a, b\}$), then

$$\begin{aligned} &Q_{m+1}^P(w_a, w_b, s) \\ &\quad = c_p + hw \int_0^\infty \frac{1 - e^{-\beta t}}{\beta} dG_p(t) + TC_P \\ &\qquad + \sum_{k=0}^{\infty} \sum_{k'=0}^{\infty} p_p(k, \lambda) p_p(k', \lambda) J_m(w_a + k, w_b + k', 0) \end{aligned}$$

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

$$\begin{aligned}
&\geq c_p + hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_p(t) + TC_P \\
&\quad + \sum_{k=0}^{\infty} \sum_{k'=0}^{\infty} p_p(k,\lambda) p_p(k',\lambda) J_m(w'_a + k, w'_b + k', 0) \\
&= Q^P_{m+1}(w'_a, w'_b, s),
\end{aligned}$$

and

$$\begin{aligned}
&Q^A_{m+1}(w_a, w_b, s) \\
&= hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A \\
&\quad + \sum_{k=0}^{\infty} \sum_{k'=0}^{\infty} p_a(k,\lambda) p_a(k',\lambda) \left\{ \sum_{j=s}^{M} q^a_{sj} J_m(w_a - 1 + k, w_b + k', j) \right\} \\
&\geq hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A \\
&\quad + \sum_{k=0}^{\infty} \sum_{k'=0}^{\infty} p_a(k,\lambda) p_a(k',\lambda) \left\{ \sum_{j=s}^{M} q^a_{sj} J_m(w'_a - 1 + k, w'_b + k', j) \right\} \\
&= Q^A_{m+1}(w'_a, w'_b, s).
\end{aligned}$$

These inequalities hold by invoking the induction assumption and the following observations:

$$\begin{aligned}
(w_a - 1 + k) &\geq (w'_a - 1 + k), \\
(w_b - 1 + k) &\leq (w'_b - 1 + k), \\
(w_a - 1 + k) + (w_b - 1 + k) &= (w'_a - 1 + k) + (w'_b - 1 + k).
\end{aligned}$$

Similarly, it can be shown that $Q^B_{m+1}(w_a, w_b, s) \geq Q^B_{m+1}(w'_a, w'_b, s)$. Therefore, for the case where $w_i > 0$ and $w'_i > 0$ $(i \in \{a, b\})$,

$$\begin{aligned}
J_{m+1}(w_a, w_b, s) &= \min\{Q^A_{m+1}(w_a, w_b, s), Q^B_{m+1}(w_a, w_b, s), Q^P_{m+1}(w_a, w_b, s)\} \\
&\geq \min\{Q^A_{m+1}(w'_a, w'_b, s), Q^B_{m+1}(w'_a, w'_b, s), Q^P_{m+1}(w'_a, w'_b, s)\} \\
&= J_{m+1}(w'_a, w'_b, s).
\end{aligned}$$

If $w_a = 0$ or $w_b = 0$ but $w = (w_a + w_b) > 0$, it can be shown that $Q^A_{m+1}(w, 0, s) \geq Q^B_{m+1}(0, w, s)$ as below:

$$\begin{aligned}
Q^A_{m+1}(w, 0, s) = \Delta + hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A \\
+ \sum_{k=0}^{\infty} \sum_{k'=0}^{\infty} p_a(k,\lambda) p_b(k',\lambda) \left\{ \sum_{j=s}^{M} q^a_{sj} J_m(w - 1 + k, k', j) \right\}
\end{aligned}$$

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

$$
\begin{aligned}
&\geq \Delta + hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_A \\
&\quad + \sum_{k=0}^{\infty} \sum_{k'=0}^{\infty} p_a(k,\lambda) p_b(k',\lambda) \left\{ \sum_{j=s}^{M} q_{sj}^b J_m(w-1+k,k',j) \right\} \\
&\geq hw \int_0^\infty \frac{1-e^{-\beta t}}{\beta} dG_a(t) + TC_B \\
&\quad + \sum_{k=0}^{\infty} \sum_{k'=0}^{\infty} p_a(k,\lambda) p_b(k',\lambda) \left\{ \sum_{j=s}^{M} q_{sj}^b J_m(k,w-1+k',j) \right\} \\
&= Q_{m+1}^B(0,w,s).
\end{aligned}
$$

The first inequality is due to the fact that $J_m(\boldsymbol{w}, s)$ is non-decreasing in s, and the assumption that $q_{sj}^a \geq q_{sj}^b$ for $\forall\, j > s$, and $q_{ss}^a \leq q_{ss}^b$. The second inequality is true because $J_m(w+k,k',j) \geq J_m(k,w+k',j)$ and $\Delta \geq 0$. Obviously, $Q_{m+1}^P(w,0,s) \geq Q_{m+1}^P(0,w,s)$ due to the induction assumption. Therefore,

$$
\begin{aligned}
J_{m+1}(w,0,s) &= \min\{Q_{m+1}^A(w,0,s), Q_{m+1}^P(w,0,s)\} \\
&\geq \min\{Q_{m+1}^B(0,w,s), Q_{m+1}^P(0,w,s)\} \\
&= J_{m+1}(0,w,s).
\end{aligned}
$$

Putting all of these observations together yields $J_{m+1}(w_a,w_b,s) \geq J_{m+1}(w_a',w_b',s)$.

Based on the same arguments used in the proof of Lemma 3.1, we have that $J(w_a,w_b,s) = \lim_{m\to\infty} J_{m+1}(w_a,w_b,s)$, and hence $J(w_a,w_b,s) \geq J(w_a',w_b',s)$. ■

Structural results for the average cost case, similar to the results obtained in the the last two sections, follow from fairly standard arguments. Those arguments are summarized in the Appendix.

5. OPTIMAL POLICY EXAMPLES: DISCOUNTED COSTS

In this section, we use the linear programming (LP) methodology to compute optimal policies for the one-recipe problem in the case with discounted costs. In subsequent sections, we present a more detailed computational study for the average cost case. The structural results proved in Section 3 are confirmed by the experiments in this section.

As before, in order to undertake numerical investigations it is assumed that the queue has a limited buffer, that is, we truncate the state space. The maximum number of jobs allowed in the queue is n and the state space of the jobs in the queue is denoted by $N = \{0,1,\ldots,n\}$. Note that in the two-recipe case studied in the next section, the limit n is applied to the total number of jobs in the system (i.e., the truncation is not applied separately to the two job types). Thus, the overall state space for the MDP is now $N \times S$.

One issue needs special attention in this model. When there is an unlimited buffer, a PM action (P) is available for almost all states except when the machine fails ($s = M$). It is easy to see that the corresponding countable state space Markov chain has a single recurrent class for any stationary policy. However, in the finite state space case, if the action P is available when the queue is full, that is, $(w,s) = (n,0)$, then under a policy that chooses P

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

at $(n,0)$ the process remains in state $(n,0)$ forever. Thus, state $(n,0)$ is an absorbing state in the corresponding Markov chain, which makes the problem very complicated. On the other hand, there is no benefit from doing a PM at $(n,0)$ because the machine is already "new" and there is a positive cost for doing a PM. Therefore, in the finite state space MDP used to generate our numerical results, P is not allowed in state $(n,0)$.

In many sectors with automated production, processing times are often nearly constant, and thus in the computational experiments the processing times are assumed to be deterministic. To simplify the numerical investigation, the PM and CM times are also assumed to be deterministic. However, note that the method also applies for general distributions.

For the one-recipe problem, the base setting is as follows:

- $n = 30$; $M = 10$; $\lambda = 0.1$ jobs per time unit;
- $T_A = 8$ time units; $T_P = 9$ time units; $T_C = 30$ time units;
- $h = \$0.05$ per job per time unit; PM cost $C_p = \$0$; CM cost $C_c = \$20$;
- Discount rate $\beta = 0.01$;
- Degradation structure: $\forall s, r \in S$ and $r \geq s$,

$$q^a_{s,r} = \begin{cases} \dfrac{1 - p_{\text{deg}}}{M - s}, & r > s, \\ p_{\text{deg}}, & r = s, \\ 0, & \text{o.w.}, \end{cases}$$

  where p_{deg} is the probability that machine state stays in its current level after processing a job. In the base setting, $p_{\text{deg}} = 0.9$.

The optimal PM policy for the base setting can be seen in the upper left-hand graph with the title of "$h = 0.05$" in Figure 1.

Figure 1 shows the effect of different holding costs from 0.05 to 0.14 in increments of 0.03. It can be seen in Figure 1 that the threshold value s^*_w increases as the holding cost increases. When holding cost is low, for example, $h = 0.05$, a PM is executed in state $s = 8$ or $s = 9$ regardless of the WIP level, and in state $s = 7$ for some WIP levels. When the holding cost is high, a PM is only performed when the machine state is $s = M - 1$. When there is no WIP, the optimal policy chooses to wait for the next job when the machine state s is less than or equal to 3, it performs a CM at $s = M$, and performs a PM otherwise. The behavior of the optimal policy at $w = n$ is due to the effects of truncating the state space. One interesting feature of the optimal policy occurs in the case where $h = 0.11$. When the machine is in state $s = 8$, notice that the policy is non-monotonic in the WIP level. This unusual feature also appeared in the model studied by Kaufman and Lewis [8].

Figure 2 shows the effect of different degradation structures, that is, p_{deg} is varied from 0.3 to 0.9 with an increment of 0.2. A large p_{deg} means that the machine tends to stay in its current state at the completion of a job, and a small p_{deg} means that the machine tends to degrade to a state worse than its current state. It is reasonable that the threshold value s^*_w increases as p_{deg} increases.

If p_{deg} is close to 1, the machine tends to stay in its current state, and it fails infrequently, even when PMs are not performed. Thus, the cost difference between the optimal policy and the no-PM policy decreases as p_{deg} increases.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

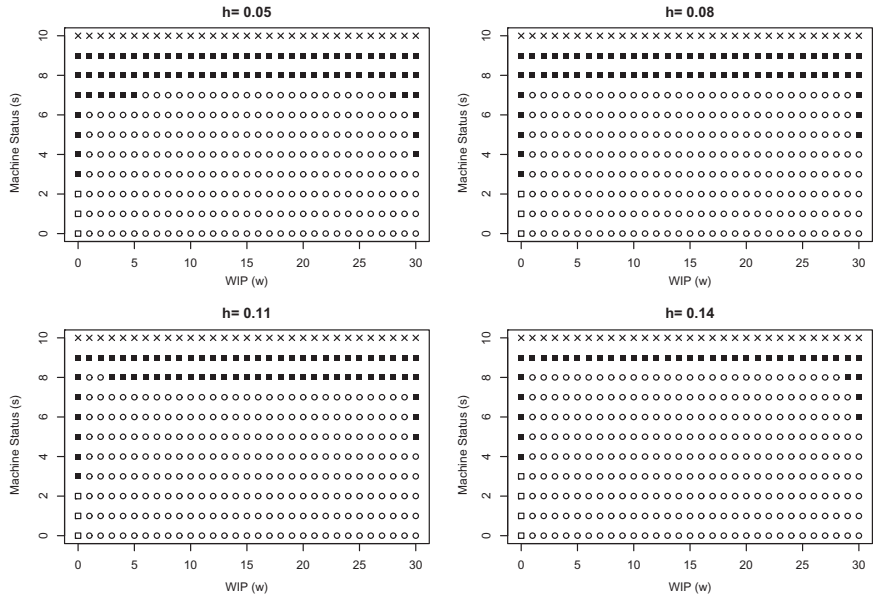


FIGURE 1. Effect of holding cost in the single-recipe problem. ○ process; □ wait; ■ PM; × CM

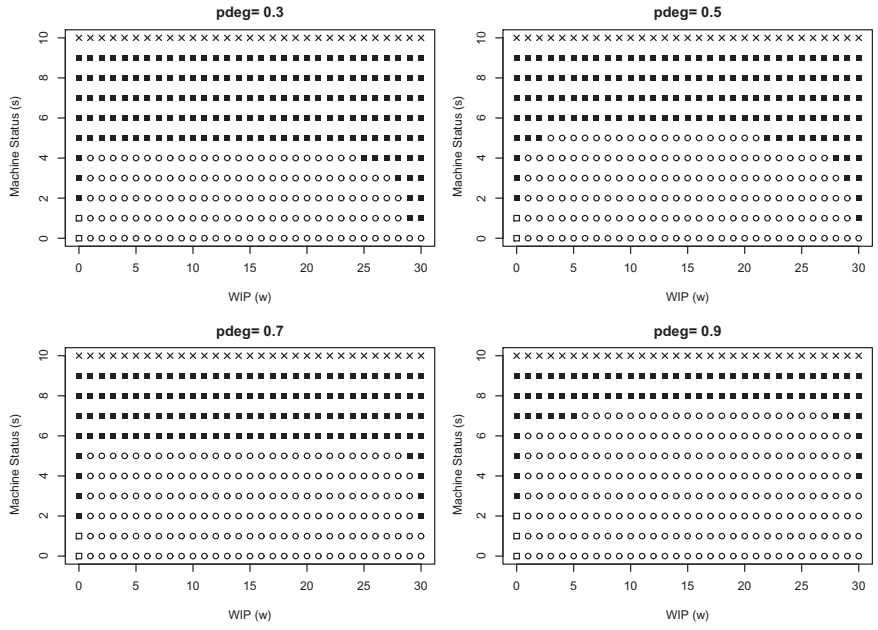


FIGURE 2. Effect of p_{deg} in the single-recipe problem. ○ process; □ wait; ■ PM; × CM

6. COMPUTATIONAL RESULTS: AVERAGE COST CASE

Apart from examining the nature of the optimal policy, it is also of interest to compare the optimal predictive maintenance policy to standard PM policies. As mentioned before, the typical PM policies sets a job- or calendar-based threshold and performs a PM when the threshold is exceeded. We performed a number of experiments to determine how much

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

improvement is possible when moving from a job-based policy to a predictive maintenance policy. One impediment to performing such a comparison is that evaluating the long-run average cost under a job-based policy is not numerically tractable in our model. A second issue is that one would really like to use the "optimal" job-based threshold to compare to the predictive maintenance policy. With these issues in mind, we developed a discrete-event simulation model in order to evaluate costs for standard PM policies.

The simulation model was built in Arena and exactly replicated the dynamics of the one- and two-recipe models, with the following exception. In order to implement the LP model for the average cost case (as in the discounted cost case) we needed to truncate the state space at 30 jobs. In the Arena simulation, we did not impose such a limit. However, the buffer limit was set high enough such that the WIP level rarely exceeded this level in the simulation runs. Therefore, any buffer limit effects are very small (likely much smaller than the sampling error from simulating). We compared policies for a number of parameter cases and for each fixed set of model parameters we ran 40 independent simulation runs of 200,000 time units each. Due to the small system size and lengthy simulation horizon, initialization bias was not an issue and therefore no warm-up period was used.

In order to determine the best job-based policy threshold simulations were performed over a large number of threshold values (in most cases, more than 20 values). For comparison, we then chose the policy with the lowest point estimate of average cost over obtained via the 40 simulation runs. The confidence intervals generated by these runs were generally small (these are reported in the tables), but in some cases they were not small enough to make strong statistical statements regarding the optimality of the particular threshold used. However, in the cases with overlapping confidence intervals for adjacent policies, the objective function also appeared to be quite "flat," that is, small changes in the threshold value had a very small effect on the total cost. Therefore, we feel that the simulation results provide a reasonable representation of "near-optimal" costs for job-based PM policies.

For each parameter case studied, we report the long-run average cost under the optimal policy, as given by the solution to the LP and the point estimate of the cost under the best job-based PM policy. We also provide confidence intervals for the cost of the job-based policy and the improvement from the optimal policy, along with the optimal threshold.

6.1. One-recipe Model

We first give results for the one-recipe model. The base settings for this case are as follows:

- $n = 30$; $M = 10$; $\lambda = 0.1$ jobs per time unit;
- $T_A = 6$ time units; $T_P = 7$ time units; $T_C = 30$ time units;
- $h = \$0.05$ per job per time unit; PM cost $C_p = \$0$; CM cost $C_c = \$20$;
- Degradation structure: $\forall s, r \in S$ and $r \geq s$,

$$q^a_{s,r} = \begin{cases} \frac{1 - p_{\text{deg}}}{M - s}, & r > 0, r > s \\ \frac{1}{N}, & s = 0, 0 \leq r \leq N - 1 \\ p_{\text{deg}}, & s > 0, r = s \\ 0, & \text{o.w.}, \end{cases}$$

where p_{deg} is the probability that machine state stays in its current level after processing a job. In the base setting, $p_{\text{deg}} = 0.9$.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

Table 1. Effect of Holding Cost on System Cost

Holding Cost	Job-Based Policy (95% CI)	Optimal Policy	Relative Savings (95% CI)
0.05	0.1742 (0.1718, 0.1766)	0.1103	36.70% (35.82%, 37.55%)
0.10	0.2821 (0.2784, 0.2859)	0.1933	31.47% (30.55%, 32.39%)
0.10	0.3884 (0.3833, 0.3936)	0.2762	28.89% (27.93%, 29.83%)
0.20	0.4946 (0.4879, 0.5014)	0.3576	27.70% (26.70%, 28.68%)

Table 2. Effect of Machine Reliability on System Cost

p_{deg}	Job-Based Policy (95% CI)	Optimal Policy	Relative Savings (95% CI)
0.7	0.8129 (0.7761, 0.8493)	0.2727	66.87% (64.87%, 67.89%)
0.8	0.3447 (0.3370, 0.3523)	0.1789	48.08% (46.90%, 49.21%)
0.9	0.1742 (0.1718, 0.1766)	0.1103	36.70% (35.82%, 37.55%)

Table 3. Effect of PM Cost on System Cost

PM Cost	Job-Based Policy (95% CI)	Optimal Policy	Relative Savings (95% CI)
0	0.1742 (0.1718, 0.1766)	0.1103	36.70% (35.82%, 37.55%)
1	0.1810 (0.1787, 0.1833)	0.1224	32.37% (31.50%, 33.24%)
2	0.1862 (0.1838, 0.1885)	0.1316	29.31% (23.41%, 30.17%)
3	0.1896 (0.1874, 0.1918)	0.1395	26.44% (25.56%, 27.28%)

We first examined the effect of the holding cost, varying this cost from 0.05 to 0.20 in increments of 0.05. The corresponding job-based thresholds were 9, 12, 14, and 14. As the holding cost increases, the controller has a greater incentive to keeping the WIP level low. Therefore processing jobs take a greater priority over PM's. With a higher holding cost, the optimal control allows more failures at the expense of the frequency of PM's. Table 1 gives the results comparing the best job-based threshold policy and the optimal policy.

Next, we varied the degradation rate of the machine. Accordingly, p_{deg} was set to 0.7, 0.8, and 0.9. The corresponding job-based thresholds were 6, 6, and 9. Note that an increase in p_{deg} corresponds to a more reliable machine in the sense that average time to failure increases with p_{deg}. Thus, it is natural to expect the best job-based threshold to increase along with p_{deg}. Table 2 summarizes the cost savings that can be realized by operating under a predicative maintenance policy, for the three machine reliability levels. Notice that the optimal cost and the relative potential savings both increase as the machine reliability decreases.

In the next set of experiments, we varied the PM cost, setting it at 0, 1, 2, and 3. The corresponding job-based thresholds were 9, 12, 17, and 19. Obviously, as PM's become more costly, a job-based policy tends to defer maintenance longer. Table 3 summarizes the results for this set of experiments. Note that although the job-based policy costs and the optimal costs vary little with the PM cost, the cost savings between the two policy classes are still significant. Interestingly, the results indicate that greater savings can be realized when the PM costs are *lower*. This same effect occurs when the holding costs are varied (see Table 1).

Finally, we varied the time needed for a PM, setting the time to 5, 7, 9, and 11 with corresponding job-based thresholds of 6, 9, 12, and 15. Note again that maintenance is

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

TABLE 4. Effect of PM Time on System Cost

PM Time	Job-Based Policy (95% CI)	Optimal Policy	Relative Savings (95% CI)
5	0.1643 (0.1619, 0.1665)	0.1017	38.09% (37.21%, 38.94%)
7	0.1742 (0.1718, 0.1766)	0.1103	36.70% (35.82%, 37.55%)
9	0.1800 (0.1776, 0.1824)	0.1185	34.17% (34.17%, 33.27%)
11	0.1844 (0.1820, 0.1868)	0.1257	31.83% (30.92%, 32.70%)

deferred longer at the PM time grows. This is due now to the indirect cost of doing a PM. When PM's take longer, there are generally greater holding costs for the jobs delayed due to the PM. Of course, this effect is one of the main motivations for using predictive maintenance policies. Table 4 gives the results for the PM time experiments.

6.2. Two-recipe Model

One set of experiments was performed in the two-recipe case, to gauge the effect of different holding costs on the cost savings, and the optimal policy. In this, there is an additional complication when comparing job-based policies with the optimal policy. Note that the optimal policy actually encompasses two types of policy decisions: when to perform maintenance and what scheduling policy to use when the machine is working. Even without maintenance considerations, the optimal scheduling policy for an M/G/1 queue with an unreliable machine is not generally tractable. In the simulation model, we used FIFO scheduling for jobs, along with the job-based threshold policy for PMs. Since the FIFO policy is most likely sub-optimal, this will increase the apparent cost savings when using the optimal policy. Although this is in some sense "unfair" to the simulation model, it does appropriately reflect the savings that are realized when a jointly optimal scheduling/PM policy is used instead of commonly used scheduling and PM policies.

For simplicity the arrival rate for recipe A and recipe B jobs were assumed to be equal. The base settings for the other model parameters are as follows:

- $n = 20$; $M = 10$; $\lambda_a = \lambda_b = 0.05$ jobs per time unit;
- $T = 6$ time units; $T_P = 7$ time units; $T_C = 40$ time units;
- $h = \$0.05$ per job per time unit; PM cost $C_p = \$0$; CM cost $C_c = \$20$;
- Process A costs \$0.5, while it costs nothing to process B;
- Degradation structure: $\forall s, r \in S$ and $r \geq s$,

$$q^a_{s,r} = \begin{cases} \dfrac{1 - p^a_{\text{deg}}}{M - s}, & r > 0, r > s \\ \frac{1}{N}, & s = 0, 0 \leq r \leq N - 1 \\ p^a_{\text{deg}}, & s > 0, r = s \\ 0, & \text{o.w.}, \end{cases}$$

where $a \in \{A, B\}$, and $p^A_{\text{deg}} = 0.7$, $p^B_{\text{deg}} = 0.9$.

In the two-recipe model experiments the holding costs were varied from 0.05 to 0.20, in increments of 0.05 and the respective thresholds were 6, 9, 9, and 9. From Table 5, it can be seen that quite significant savings are possible, with relative savings much larger than

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

TABLE 5. Effect of Holding Cost on System Cost: Two-recipe Case

Holding Cost	Job-Based Policy (95% CI)	Optimal Policy	Relative Savings (95% CI)
0.05	0.6944 (0.6679, 0.7209)	0.2128	69.35% (68.14%, 70.48%)
0.10	1.1878 (1.1331, 1.2424)	0.3353	71.77% (70.41%, 73.01%)
0.15	1.6687 (1.5871, 1.7502)	0.4569	72.62% (71.21%, 73.89%)
0.20	2.1497 (2.012, 2.2380)	0.5785	73.09% (71.66%, 74.38%)

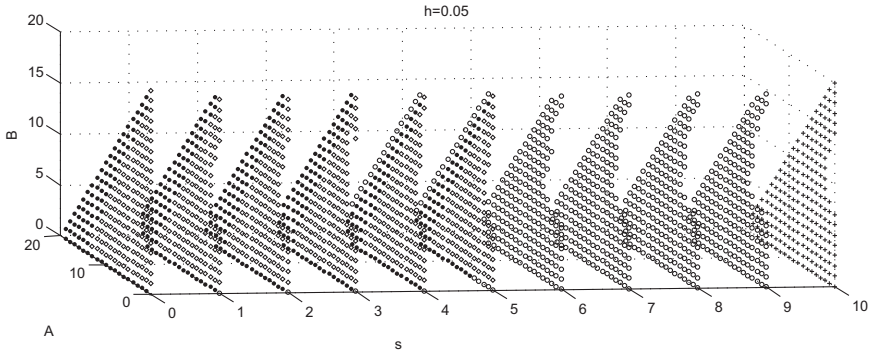


FIGURE 3. Optimal Policy: Two Recipes and $h = 0.05$ + CM; ∘ PM; • Process A; ◇ Process B

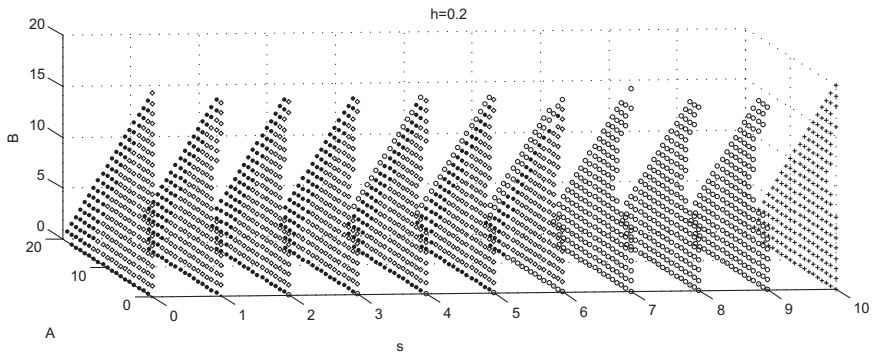


FIGURE 4. Optimal Policy: Two Recipes and $h = 0.20$ + CM; ∘ PM; • Process A; ◇ Process B

those seen in the one recipe examples. Of course, part of this is due to the suboptimality of the scheduling policy used in the job-based PM model.

Finally, we present depictions of the optimal policy in the two-recipe case. The optimal policy for the base setting is depicted in Figure 3 and the policy for the same parameters, except with holding cost $h = 0.2$, is shown in Figure 4. For these parameters note that recipe B is "superior" to recipe A, that is, it seems myopically optimal to process B instead of A whenever B is available. However, Figures 3 and 4 show results that are similar to the discounted cost counterexample given in Section 4.2: A is chosen over B in some states even when B is available. Furthermore, the figures also show that whether to perform a PM or not depends not only on the total WIP but also on number of jobs from each recipe present. With the same total WIP (i.e., $w = w_a + w_b$), different combinations of w_a and w_b lead to

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

different PM decisions. Thus, as noted earlier, the optimal policy cannot be described with a single total WIP threshold for a given machine state.

7. CONCLUSION

This paper focuses on single-machine predictive maintenance problems. The structural results provided confirm one's intuition in that if a PM is recommended for a given machine status and WIP level, it is also recommended when the machine status is worse. However, the numerical results also show that such monotonicity in the WIP level does not hold. Therefore, a system controller does indeed require a sophisticated model to determine when to take the opportunity to perform a PM. Similarly, the two-recipe examples with "one superior recipe" also indicates that simple intuition may fail.

Our numerical results on system costs for standard PM policies versus predictive maintenance indicate that significant savings can be realized by monitoring the machine status. Given the increasingly sophisticated system models available to estimate the state of machines in many sectors, true predictive maintenance is quite feasible.

We used an LP model to obtain the optimal policy for our computational experiments. For the two-recipe model considered here, the LP model is quite tractable. The computational complexity, however, will increase substantially as the number of recipes is increased. Thus, it may be necessary to consider special large-scale linear programming methods or to use more efficient SMDP methods (policy or value iteration) to solve larger multiple-recipe problems. Also, to extend the model to a more realistic setting, it would be desirable to consider a multiple-machine, multiple-recipe predictive maintenance problem, that is, to model tool groups. In that case, the integrated optimal PM/production policy would consider machine state and degradation structure to allocate jobs to various machines within the tool group. Obviously, solving such a model computationally would also require significant numerical innovation.

References

1. *Xpress-mosel user guide* (2004). Englewood Cliffs, NY: Dash Optimization Ltd.
2. Baek, K.H., Jung, Y., Min, G.J., Kang, C., Cho, H. K., & Moon, J.T. (2005). Chamber maintenance and fault detection technique for a gate etch process via self-excited electron resonance spectroscopy. *Journal of Vacuum Science and Technology B* 23(1): 125–129.
3. Bertsekas, D.P. (2007). *Dynamic programming and optimal control*. Vol. I & II. Belmont, MA: Athena Scientific.
4. Cassady, C.R. & Kutanoglu, E. (2003). Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. *IIE Transactions* 35: 503–513.
5. Dekker, R. (1996). Applications of maintenance optimization models: a review and analysis. *Reliability Engineering and System Safety* 51: 229–240.
6. Hsu, L.F. (1992). Optimal preventive maintenance policies in an M/G/1 queue-like production system. *European Journal of Operational Research* 58(1): 112–122.
7. Iravani, S.M.R. & Duenyas, I. (2002). Integrated maintenance and production control of a deteriorating production system. *IIE Transactions* 34: 423–435.
8. Kaufman, D. & Lewis, M. (2007). Machine maintenance with workload considerations. *Naval Research Logistics* 54: 750–766.
9. Lee, J. (1998). Teleservice engineering in manufacturing: challenges and opportunities. *International Journal of Machine Tools and Manufacture* 38: 901–910.
10. McCall, J.J. (1965). Maintenance policies for stochastically failing equipment: a survey. *Management Science* 11(5): 493–524.
11. Mobley, R.K. (2002). *An introduction to predictive maintenance*. Oxford: Butterworth-Heinemann.
12. Ruiz R., García-Díaz J.C., & Maroto, C. (2007). Considering scheduling and preventive maintenance in the flowshop sequencing problem. *Computers and Operations Research* 34(11): 3314–3330.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

13. Sennott, L. (1989). Average cost semi-Markov decision processes and the control of queueing systems. *Probability in the Engineering and Informational Sciences* 3: 247–272.
14. Sloan, T.W. & Shanthikumar, J.G. (2000). Combined production and maintenance scheduling for a multiple-product, single-machine production system. *Production and Operations Management* 9(4): 379–399.
15. Van der Duyn Schouten, F.A. & Vanneste, S.G. (1995). Maintenance optimization of a production system with buffer capacity. *European Journal of Operational Research* 82(2): 323–338.
16. Yang, Z., Djurdjanovic, D., & Ni, J. (2007). Maintenance scheduling for a manufacturing system of machines with adjustable throughput. *IIE Transactions* 39: 1111–1125.
17. Yao, X. (2003) *Optimal preventive maintenance policies for unreliable queueing and production systems*. PhD thesis, University of Maryland.
18. Zhou, J., Djurdjanovic, D., Ivy, J., & Ni, J. (2007). Integrated reconfiguration and age-based preventive maintenance decision making. *IIE Transactions*, 39: 1085–1102.

APPENDIX

We extend the results on the optimality of threshold policies to the average cost case, with a focus on the two-recipe model. Obviously, the single recipe results follow as special cases. The extension of discounted cost results to average cost results is fairly standard and we invoke the usual results of Sennott [13]. Recall that under a policy π and initial state $(\boldsymbol{w}, s)$, we define the average cost rate to be:

$$J^{\pi}(\boldsymbol{w}, s) = \limsup_{N\to\infty} \frac{1}{N} E_{(\boldsymbol{w},s)} \left\{ \int_0^{t_N} g^{\pi}(\boldsymbol{w}_t, s_t, u_t)\, dt \right\},$$

where $g^{\pi}(\boldsymbol{w}_t, s_t, u_t)$ is as defined in Section 3. Let J be the optimal cost function, defined by

$$J(\boldsymbol{w}, s) = \inf_{\pi\in\Pi} J^{\pi}(\boldsymbol{w}, s).$$

We also define the average cost optimality inequalities (ACOI) as follows:

$$J(\boldsymbol{w}, s) \geq \min \left\{ Q^{u}(\boldsymbol{w}, s) \,|\, u \in U_{w,s} \right\}, \tag{1}$$

for all $(\boldsymbol{w}, s)$ in the state space. Again, $Q^{u}(\boldsymbol{w}, s)$ is the cost corresponding to action $u \in U_{\boldsymbol{w},s}$ under the average cost criterion. The expressions for these costs are slight modifications of the discounted cost expressions and are not detailed here. We follow closely the arguments used in [8] to move from the discounted cost case to the average cost case. In particular, we need to verify the so-called SENSM [13] conditions for SMDPs, as given below:

- SENSM1: There exist a $\delta > 0$ and $\epsilon > 0$ such that for every state and action the probability that the transition time is greater than δ is at least ϵ.
- SENSM2: There exists an R such that for every state $(\boldsymbol{w}, s)$ and action u, $\tau[(\boldsymbol{w}, s), u] \leq R$, where $\tau[(\boldsymbol{w}, s), u]$ is the mean transition time out of state $(\boldsymbol{w}, s)$ under action u.
- SENSM3: $J_{\beta}(\boldsymbol{w}, s) < \infty$ for every state $(\boldsymbol{w}, s)$ and $\beta > 0$.
- SENSM4: There exist a $\beta_0 > 0$ and non-negative numbers $K_{(\boldsymbol{w},s)}$ such that for every state $(\boldsymbol{w}, s)$ and β satisfying $0 < \beta < \beta_0$, $h_{\beta}(\boldsymbol{w}, s) \leq K_{(\boldsymbol{w},s)}$ where $h_{\beta}(\boldsymbol{w}, s) = J_{\beta}(\boldsymbol{w}, s) - J_{\beta}(0)$ for a distinguished state 0. Furthermore, for every state $(\boldsymbol{w}, s)$ there exists an action $u(\boldsymbol{w}, s)$ such that

  $$\sum_{(\boldsymbol{w}',s')} p_{(\boldsymbol{w},s),(\boldsymbol{w}',s')}(u(\boldsymbol{w}, s)) K_{(\boldsymbol{w}',s')} < \infty,$$

  where $p_{(\boldsymbol{w},s),(\boldsymbol{w}',s')}(u)$ is the transition probability under action u.
- SENSM5: There exist a $\beta_0 > 0$ and non-negative number L such that $-L \leq h_{\beta}(\boldsymbol{w}, s)$ for all $(\boldsymbol{w}, s)$ and $0 < \beta < \beta_0$.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press

The SENSM conditions insure that the SMDP is regular, has bounded mean transition times, and that the standard procedure of taking the limit of the discount factor is valid. The following lemma is the key to extending the discounted cost results.

LEMMA .1: *Suppose there exists a stationary policy such that the induced discrete-time Markov chain (DTMC) is ergodic and corresponding expected cost, under the stationary measure, is finite. Then SENSM1 through SENSM5 are satisfied for the SMDP. Furthermore, for every state* $(\boldsymbol{w}, s)$ *there exists a limit*

$$Q(\boldsymbol{w}, s) = \lim_{k \to \infty} [Q_{\beta_k}(\boldsymbol{w}, s) - Q_{\beta_k}(0, 0)],$$

where $\beta_k \to 0$. *There also exists a constant* J, *such that the* $Q(\boldsymbol{w}, s)$ *and* J *satisfy the ACOI.*

PROOF: In the SMDP model, there is a transition either when an arrival occurs, a job completes service, or after a maintenance is performed. Since the total arrival rate is finite and the job processing times have positive means, as do the maintenance times, SENSM1 holds. SENSM2 holds because $\lambda > 0$, and the mean processing and maintenance times are obviously bounded above. Thus, the mean transition time out of any state, under any action, is also uniformly bounded above. For SENSM3 through SENSM5, it will be useful to designate the distinguished state. To this end, we let the distinguished state be (0,0). Then, for this distinguished state, Lemmas 4.1 and 4.2 imply that this state has the lowest cost among all states. This means that $h_\beta(\boldsymbol{w}, s) \geq 0$ for all $(\boldsymbol{w}, s)$ and β. Therefore we can set $L = 0$ in SENSM5. The ergodicity and finite expected cost assumptions imply SENSM3 and SENSM4 via Proposition 2 in [13]. Finally, the result then follows directly from Theorem 2 in [13]. ■

The lemma above immediately allows us to translate the monotonicity results from the discounted case to the average cost case. In particular Lemma 4.1 and Theorem 4.3 also hold in the average cost case and so a threshold type policy is optimal in the average cost problem. As before, the threshold is only based on the machine state, not the WIP level. Below, we state the main structural result for the average cost case:

THEOREM .2: *For any* $\boldsymbol{w}$, $\exists\, s^*_{\boldsymbol{w}}$, *s.t.* $\forall\, s^*_{\boldsymbol{w}} \leq s < M$,

$$Q^P(\boldsymbol{w}, s) = \min\left\{Q^u(\boldsymbol{w}, s) \,|\, u \in U_{\boldsymbol{w},s}\right\},$$

which means P *is optimal when the machine status is at or above the threshold* $s^*_{\boldsymbol{w}}$.

https://doi.org/10.1017/S0269964812000423 Published online by Cambridge University Press