Journal of Process Control 22 (2012) 1478–1489

Contents lists available at SciVerse ScienceDirect

Journal of Process Control

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

On defect propagation in multi-machine stochastically deteriorating systems with incomplete information

Rakshita Agrawal[a], Matthew J. Realff[a], Jay H. Lee[b,*]

[a] Department of Chemical and Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[b] Department of Chemical and Biomolecular Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea

ARTICLE INFO

Article history:
Received 24 June 2011
Received in revised form
13 December 2011
Accepted 24 January 2012
Available online 7 March 2012

Keywords:
Markov decision processes
Dynamic programming
Random deterioration
Maintenance
Inspection

ABSTRACT

In many manufacturing environments, costly job inspection provides information about the random deterioration of the machines. The resulting maintenance and inspection problem is extensively studied for a single machine system by using the framework of Partially Observable Markov Decision Processes (POMDPs). In this work, this concept is extended to multiple operations and multiple job types by considering two process flow topologies: (i) re-entrant flow, (ii) hybrid flow. The resulting (significantly large sized) POMDPs are solved using a point based method called PERSEUS, and the results are compared with those obtained by conventionally used periodic policies.

1. Introduction

Many manufacturing systems have machines/equipments that deteriorate randomly. Examples can be seen in auto-parts manufacturing, semi-conductor manufacturing, chemical process industry, etc. The effect of this deterioration is generally reflected as one or a combination of the followings: a lower yield, higher fraction of defective intermediates, higher operating or maintenance cost, or increased probability of complete failure of the equipment. The random deterioration in a single machine is often modeled as a Markov chain [1], where the equipment can be in one of N states at any time. Several researchers have leveraged the flexibility associated with Markov chains for modeling machine deterioration problems [3,11,14–17,20] since the 60 s. Machine state is typically designated as $i = 1,2, \ldots, N$, with '1' being the best state and the machine progressively degrading until it reaches an absorbing state 'N'. The state N may characterize a completely failed state or a state of worst possible machine performance leading to least economically favorable production scenario. The state that a machine occupies at a particular time is rarely known with certainty to the decision-maker and the machine may end up in state N (failed) without the decision-maker's notice, which is termed as 'silent failures' in [2].

To keep the machine from ending up in a failed state, an optimal maintenance policy is needed. Typical decisions include renewal/replacement (to bring the machine back to the state '1'), repair (bring it back to a relatively newer state), machine inspection (incur a cost to know the machine condition), or inspection of machine's output. The inspection is needed because the machine condition is not directly observable and may lead to perfect or imperfect knowledge of the condition depending on the quality of observations [4], presents a survey on replacement and repair policies for randomly deteriorating systems found in existing literature and industry. Typical policy structures are block replacement, age replacement, order replacement, failure limit policy, sequential preventive maintenance policy, and repair cost limit policy. A more rigorous approach, introduced by [13], is to use the Partially Observable Markov Decision Process (POMDP) framework to obtain a mathematically optimal repair and inspection policy. To this end, a survey of maintenance studies on single machine systems prone to stochastic degradation is given in [18,19]. Structural properties of the optimal value function and optimal policies are derived for many cases.

A case of particular interest considered in [3] is the one where the machine deterioration is reflected in increased production of defective jobs. The information about the machine condition may be known only by means of costly inspection of the machine output, which relates probabilistically with the machine condition. The problem of costly job inspection is considered by [3,15,16,20] and is referred to as the case with 'imperfect and incomplete observation'.

* Corresponding author. Tel.: +82 42 350 3926; fax: +82 42 350 3910.
E-mail address: jayhlee@kaist.ac.kr (J.H. Lee).

0959-1524/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2012.01.018

Due to high cost of inspection, it may not be economically favorable to test every processed job. Such characteristics are prevalent in jobs requiring specialized testing for quality variables like electrical properties, radioactivity, product composition, uniformity, etc.

In most real world situations, a job undergoes a series of operations on multiple machines. Therefore, the notion of incomplete job inspection motivates the analysis of defect accumulation and propagation in the systems with multiple operations and (or) multiple machines, which is the main subject of this paper. Defect propagation here means that untested defective intermediates would go through the system, until found defective in the final testing. Due to the possibility of accumulation of defective intermediates, job scheduling may also be affected by machine renewal and job inspection decisions. It should be noted that even if the inspection of all jobs was favorable, in the presence of inspection errors of type I [5], defective jobs would be reported non-defective and allowed to propagate through the system. Only error-free or perfect inspection is considered in this work but the analysis can be easily extended to type I errors. The above mentioned aspects of defect accumulation and propagation in systems with stochastically degrading machine(s) are addressed by considering two process flow topologies:

(i) a re-entrant flow system characterized by a job going through the same operation more than once
(ii) a hybrid flow system which is a combination of serial and re-entrant flow

Since the knowledge of deterioration level of the machine(s) and the un-tested defective jobs is not available in complete form, the problems are naturally formulated as partially observable Markov decision processes (POMDPs). However, the addition of scheduling decisions in the presence of partially observable states leads to fairly large size problems even for simple real world systems. Recent research [6–8] in the area of approximate solution methods for large POMDPs proves helpful in this regard. In this work, a point based solution method called PERSEUS [6] is used to solve the above-mentioned problems and the numerical results are reported. Comparison with commonly used periodic policies for maintenance and inspection are also presented to demonstrate the benefit from taking the more rigorous POMDP approach. It turns out that, due to the added complications in the illustrated cases, the optimal policies cannot be characterized in a convenient, simple form.

The article is organized as follows: to start with, an overview of the theory and solution methods for Partially Observable Markov Decision Processes (POMDPs) is provided in Section 2. The point-based methods used to solve medium sized POMDPs are discussed briefly. The nomenclature used in this article is summarized in Section 3. Section 4 presents details about the manufacturing systems considered in this work along with modeling assumptions. Section 5 contains three illustrations. Specific system models, parameter values, and solution procedures are reported. Results and important findings are discussed in Section 6 and conclusions are reported in Section 7.

2. POMDP overview

2.1. Definition and notation

POMDP describes a discrete-time stochastic control process when the states of the environment are partially observed. At any time, the system is in one of the states $s \in S$ where S is a set of all permissible states and is called 'state space'. By taking an action a, the system transitions to the next state $s' \in S$ according to a known probability of $p(s'|s,a)$ and accrues a reward $r(s,a)$. The next state s' is not completely observed but an observation o may be made, which is probabilistically related to the state s' and action a by $p(o|s',a)$ through stochastic system dynamics. Throughout this paper, symbol $p(.)$ is used to denote probability of a quantity.

More formally, it corresponds to a tuple (S, A, Θ, T, O, R) where S is a set of states, A is a set of actions, Θ is a set of observations, T: $S \times A \times S \rightarrow [0,1]$ is a set of transition probabilities that describe the dynamic behavior of the modeled environment, O: $S \times A \times \Theta \rightarrow [0,1]$ is a set of observation probabilities that describe the relationships among observations, states and actions, and R: $S \times A \times S \rightarrow R^1$ denotes a reward model that determines the reward when action a is taken in state s leading to next state s'. Mostly, the notational convention for POMDPs is adopted from [10]. $r(.)$ represents the reward received for a state-action pair. The dependence of reward function on s' is usually suppressed by taking a weighted average over all possible next states ($r(s,a) = \sum_{s'} p(s'|s,a)R(s,a,s')$). p_{ij} denoting transition from state $s=i$ to $s'=j$ is used to denote transition probabilities associated with the system dynamics. T_a and O_a are used to represent the probability transition matrix and observation matrix corresponding to action a. Symbols s, s', o and a are used to denote current state, next state, observation and action and belong to sets S, S, Θ and A, respectively.

The goal is to maximize the discounted sum of rewards over a time horizon, which can be either finite or infinite. When the states are completely observed, the resulting problem is simply a Markov Decision Process (MDP) and the goal can be achieved by solving the Bellman Equation for finite or infinite horizon problems. It is well-known [9] that for infinite horizon problems, a stationary optimal policy of the form in (1) exists, where $V^*(s)$ is the average discounted infinite horizon reward obtained when the optimal policy is followed starting from s until infinity [9]. $a^*(s)$ is the optimal action to be taken when the system is in state s, independent of time t. $V^*(s)$ is called the optimal value function and is obtained as the solution to Bellman Eq. (2) for all s. $0 \leq \gamma < 1$ is the discount rate that discounts the future rewards.

$$a^*(s) = \arg\max_{a \in A} \sum_{s' \in S} p(s'|s,a)\{R(s,a,s') + \gamma V^*(s')\} \quad \forall s \tag{1}$$

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} p(s'|s,a)\{R(s,a,s') + \gamma V^*(s')\} \quad \forall s \tag{2}$$

For the solution of MDP in this work, one of the popular solution methods called value iteration is chosen. Starting with an arbitrary value function $V_0(s)$, the value function is iteratively improved using (3) until ε-convergence is reached. The operator for one iteration can be denoted as H such that $V_{n+1} = HV_n$, as below:

$$V_{n+1}(s) = \max_{a \in A} \sum_{s' \in S} \{R(s,a,s') + \gamma p(s'|s,a)V_n(s')\} \quad \forall s \quad \text{until} \quad |V_{n+1} - V_n|_\infty \leq \varepsilon \tag{3}$$

When the system state is not perfectly observed, a history of all actions and observations since $t=0$ need to be maintained. Due to Markov property, this information is contained in the probability distribution over all states at any time. The probability distribution is referred to as belief state $b(s)$ for $s \in S$. The belief states are continuous since they contain the probability values, which are continuous numbers between 0 and 1. The partial observability thus converts the original problem into a fully observable MDP (FOMDP) with continuous states. Since all the elements of a belief state add up to 1, the state dimension of the surrogate FOMDP is one less than the size of the original state space.

Similar to MDP, an infinite horizon POMDP has an optimal stationary policy $\pi^*(b)$, which maps the *belief* states to optimal actions. A policy π can be characterized by a value function V^π which is

defined as the expected future discounted reward. $V^{\pi}(b)$ is accrued when system is initially in state b and policy π is followed, where $0 \leq \gamma < 1$ is the discount rate that discounts the future rewards.

$$V^{\pi}(b) = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(b_t, \pi(b_t))|b_0 = b\right] \quad (4)$$

The value function corresponding to the optimal policy maximizes $V(b)$ over all feasible π and satisfies the Bellman Eq. (5) for all b.

$$V^*(b) = \max_{a \in A}\left[\sum_{s \in S} r(s,a)b(s) + \gamma \sum_{o \in O} p(o|b,a)V^*(b_a^o)\right] \quad \forall b \in B \quad (5)$$

Here b_a^o is the belief state for the subsequent time obtained when action a is taken in state b and observation o is made. The expression for each element of b_a^o is as below:

$$b_a^o(s') = \frac{p(o|s',a)\sum_{s \in S} p(s'|s,a)b(s)}{p(o|b,a)} \quad (6)$$

Here $b_a^o(s')$ represents the element of the belief vector b_a^o corresponding to the state s'. Similar to the value iteration for MDPs, the value update step for a belief point b is shown in (7).

$$V_{n+1}(b) = \max_{a \in A}\left[\sum_{s \in S} r(s,a)b(s) + \gamma \sum_{o \in O} p(o|b,a)V_n(b_a^o)\right] \quad (7)$$

However, due to the continuous nature of the belief state space and consequently an infinite number of belief points, it may not be feasible to perform the exact value iteration. To alleviate this problem, researchers have looked into ways to exploit the fact that the optimal value function corresponding to POMDP has a parametric form. For finite horizon problems, the value function is piece-wise linear and convex (PWLC) function of the belief states [3] and for discounted infinite horizon POMDP, it can be approximated arbitrarily well with a PWLC function [11]. Over the years, many methods have been developed that make use of this property to solve the POMDP. Since, the exact solution methods are limited to problems of very small sizes, approximate point based solution methods like PERSEUS [6], HSVI [7], BPVI [8], etc. have been studied recently, which expand the scope of POMDP to problems of much larger sizes. In particular, PERSEUS uses the concept of asynchronous dynamic programming and randomly updates only a subset of belief states in one value iteration step. In this work, value updates in spirit similar to PERSEUS are used.

2.2. PERSEUS – an approximate solution method

Here we summarize the basics of the algorithm. For details, the readers are referred to [6].

Given the PWLC structure of the value function, the value function at the nth iteration (V_n) is parameterized by a finite set of gradient vectors α_n^i, $i = 1, 2, \ldots, |V_n|$, as shown in (8). The gradient vector that maximizes the value at a belief state b (also referred to as a belief point or just point) in the infinite belief space is represented by $\alpha_n^{(b)}$ as in (9). Superscript i indicates the ith gradient vector in the set and superscript (b) indicates the vector that maximizes $V_n(b)$ for a particular b. During an exact value iteration step then, the value ($V_{n+1}(b)$) and the gradient ($\alpha_{n+1}^{(b)}$) corresponding to any point can be updated using the Bellman backup operator as shown in (10), which can be derived based on (5) and (6):

$$V_n(b) = \max_{\alpha_n^i} \langle b, \alpha_n^i \rangle \quad (8)$$

$$\alpha_n^{(b)} = arg\max_{\alpha_n^i} \langle b, \alpha_n^i \rangle \quad (9)$$

$$backup(b) = \alpha_{n+1}^{(b)} = \underset{\{g_a^b\}_{a \in A}}{arg\max} \langle b, g_a^b \rangle \quad (10)$$

where

$$g_a^b(s) = r(s,a) + \gamma \sum_{o \in \Theta} \underset{\{g_{a,o}^i\}_i}{arg\max} \langle b, g_{a,o}^i \rangle$$

$$g_{a,o}^i(s) = \sum_{s' \in S} p(o|s',a)p(s'|s,a)\alpha_n^i(s')$$

$$V_n(b) \leq V_{n+1}(b) \leq HV_n(b) \quad \forall b \in B \quad (11)$$

As before, the notation $g_a^b(s)$ represents the scalar element of the vector g_a^b corresponding to the state s. The same is true for $g_{a,o}^i(s)$ and $\alpha_n^i(s')$.

In PERSEUS, a subset B of belief points is obtained by taking random actions. This belief set is fixed and chosen as the new belief space for value function updates. Due to parameterization of the value function (8), an updated gradient vector for a belief point may improve the value of many other points in the belief set. This leads to the concept of approximate PERSEUS backups as shown in the algorithm below. In each value backup stage, the value of all points in the belief set can be improved by only updating the value and gradient of only a subset of points. The resulting value function estimate will follow the condition shown in (11) where HV_n is the estimate of the value function at nth iteration if the *entire* belief space were updated. For more details on PERSEUS, please refer to [6].

Perseus backup stage: $V_{n+1} = H_{perseus} V_n$
1. Set $V_{n+1} = \emptyset$. Initialize $\tilde{B}$ to B
2. Sample a belief point b uniformly at random from $\tilde{B}$ and compute $\alpha = \text{backup}(b)$
3. If $b.\alpha \geq V_n(b)$ then add α to V_{n+1}, otherwise add $\alpha' = arg\max_{\{\alpha_n^i\}_i} b.\alpha_n^i$ to V_{n+1}.
4. Compute $\tilde{B} = \{b \in \tilde{B}: V_{n+1}(b) < V_n(b)\}$
5. If $\tilde{B} = \emptyset$ then stop, else go to 2.

PERSEUS is an elegant and fast method for solution of POMDPs with proven convergence properties. For convergence, it is required that the initial value function is under-estimated everywhere. However, there are no performance guarantees with respect to the optimal value function. This is because the method considers a randomly selected belief set on which value iteration updates are carried out. This is done under the assumption that the parameterization using the gradient vectors would generalize well to the entire belief space. However, there is no indication of how good that generalization will be, even after the convergence criterion is met. Therefore, re-sampling techniques are used to ensure that the value function generalizes well to different parts of the belief space. A detailed discussion on the algorithms we used is deferred until Section 4. In the following section, the general properties of the systems considered in this work are presented.

3. Formal nomenclature

This section is dedicated to presenting the nomenclature for modeling and solution variables at one place.

Recall from the POMDP definition in Section 2:

POMDP model

$S = \{s_i \text{ for } i = 1,2,\ldots|S|\}$ – underlying state space
$A = \{a_i \text{ for } i = 1,2,\ldots|A|\}$ – action space
$\Theta = \{o_i \text{ for } i = 1,2,\ldots|\Theta|\}$ – observation space
T: $p(s'|s,a)$ for all $s \in S$, $s' \in S$, $a \in A$ – transition probability matrix
O: $p(o|s',a)$ for all $s' \in S$, $a \in A$ – observation probability matrix
R: $r(s,a)$ for all $s \in S$, $a \in A$ – reward matrix
γ-discounting factor
$B = \{b_i \text{ for } i = 1,2,\ldots|B|\}$ – belief space

Optimal value function, action and policy

V^*—optimal value function associated with state $s \in S$ or belief state $b \in B$ for infinite horizon problem
π^*—optimal policy associated with state $s \in S$ or belief state $b \in B$ for infinite horizon problem
a^*—optimal action associated with state $s \in S$ or belief state $b \in B$ for infinite horizon problem

Other variables

V_n—approximation of value function at nth iteration
$\alpha_n^i(s)$ — ith gradient vector for $i = 1,2, \ldots, |V_n|$ and s belonging to S used for characterizing piece-wise linear value function V_n at nth iteration
ε—parameter for criterion for stopping value iteration
β_s—probability of defect in state s
L—total number of layers in the final product of re-entrant flow operation
η—length/size of queue for re-entrant flow operation
C_x—cost/reward for various operations/events x, e.g. production, inspection, successful completion of non-defective product, etc.
n_l—number of total jobs in the queue which have undergone re-entrant flow operations l times
d_l—number of *defective* jobs in the queue which have undergone re-entrant flow operations l times
$\tilde{A}$, $\tilde{B}$, $\tilde{C}$—designation for different types of *machines* (sign $\sim$ used to distinguish between machine related and state/action space related variables)
$\tilde{a_l}$, $\tilde{b_l}$, $\tilde{c_l}$—designation for different types of *jobs* that are processed at machines $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, respectively and have $l = 1,2, \ldots, |L|$ layers deposited on them

4. System description

In this work, discrete manufacturing systems with single or multiple machines are considered. The general characteristics of the system and modeling assumptions are as follows:

4.1. Modeling machine deterioration

All machines considered in subsequent problems are modeled to be deteriorating according to an underlying Markov chain, as discussed in Section 1. A good state is differentiated from a bad state by the associated probability of defect generation β_s, such that $\beta_s < \beta_{s+1}$ for $s = 1,2, \ldots, N-1$. Actions of machine renewal, job inspection and job scheduling are considered. The processed job is observed to be either defective or non-defective with complete accuracy whenever job inspection is performed. In case of multiple machine systems, the state transition probabilities and defect probabilities corresponding to machine states are independent from one machine to another unless otherwise mentioned.

4.2. Defect accumulation and propagation

It is assumed that a defective job can be scrapped or reworked (depending on the problem specification), only when a job inspection is carried out at that instant. If the job is not inspected due to economic reasons, the defective jobs tend to accumulate and propagate through the system. Fig. 1(a) and (b) shows a serial manufacturing system and a parallel assembly system, respectively [12]. The jobs (denoted by $\tilde{a_l}$ for job completing the lth operation) that are found defective may be reworked/repaired in the serial manufacturing system, while defective jobs would typically be scrapped in the assembly system when found defective. The

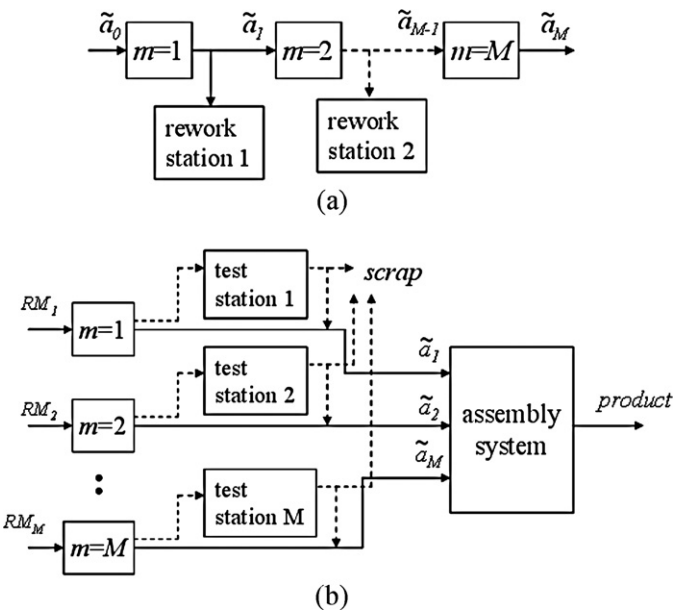

Fig. 1. (a) Serial production system with rework. (b) Assembly system with scrap.

defective jobs that are not inspected would go on to the next operation or final assembly. It is assumed that when a job is inspected, defects caused by all prior operations are revealed as opposed to just the last operation.

For the purpose of modeling, the defective jobs in the system need to be kept track of at all time. Therefore, at any time, the system state can be fully characterized by two pieces of information: (i) the state of all the machines; (ii) the total number of intermediates and the fraction of defective items in them. To differentiate the general system state from machine state, the latter is referred by machine *condition* or *deterioration level* in the subsequent analysis.

4.3. Objective

Most studies on optimal maintenance policies for randomly deteriorating systems minimize the finite or infinite horizon cost [1–3]. This is because the degradation of the machine is reflected in increased operating cost and/or increased maintenance cost. For example, in [2], the cost of repair increases with the extent of repair, which in turn depends on severity of the deterioration. In this work, it is assumed that the cost of renewal is the same for all machine regimes. Since inspection is carried out on jobs only, and not on machines, inspection cost is not a function of machine regime. Consequently, the deterioration is reflected only in the fraction of defective jobs, an increase in which leads to a lower revenue. Therefore, the infinite horizon profit is maximized for all illustrations.

A good heuristic used in industrial applications is to employ an age replacement/renewal policy and periodic inspection policy. In similar spirit, heuristics of the following nature are used to establish a lower bound on the POMDP solution.

- maintain every θ_m time units
- test every θ_t time units

The best periodic policy also helps to obtain a sample set of belief points representing the relevant region of the belief space for carrying out the PERSEUS iterations. The Fully Observable MPD (FOMDP) solution, which makes an unrealistic assumption of the full knowledge of the Markov states at all times, provides a loose but unachievable theoretical upper bound.

4.4. The single machine system

For a single machine prone to random degradation, the following cases have been extensively studied:

- Fully observable – the machine state is perfectly observable
- Unobservable – no information about the machine condition is available at any time
- Imperfect observation – imperfect observations, e.g. information about processed job is readily available at all times
- Costly inspections – machine inspection or job inspection may be carried out at a cost.

For all of the above cases [16,17,20] have proven the existence of optimal control limit policies under certain assumptions on the system dynamics and reward function. A particular instance is the unobservable case where the optimal policy is to replace every m runs, where m can be infinity [17]. The conditions for such a policy to be optimal are:

(i) $r(s,a)$ is nonincreasing in s $\forall a$
(ii) Foran ordering $a' > a$ $r(s, a') - r(s, a)$ ismonotonein s
(iii) $\sum_{j=k}^{N} p_{ij}$ is nondecreasing in i for $\forall a$, $k \in \{1, 2, \ldots, N\}$

5. Illustrations, specific models and solution

In order to understand the concept of partial and incomplete observation and lay the foundation for future illustrations, an instance of the 'general repair and inspection model' presented in [19] is shown as Illustration I. For ease of exposition, the transition probability matrix is considered to be an upper triangular matrix for all actions except that of machine renewal. This requires that $p_{ij} = 0$ for $i < j$ and $p_{NN} = 1$, making $s = N$ an absorbing state. The transition probabilities corresponding to the renewal action have $p_{i1} = 1$, $p_{ij} = 0, j \neq 1$.

Illustration I: A hypothetical machine produces one job per unit time and is prone to deterioration according to the model described earlier in this section. Pertinent decisions include machine renewal and job inspection, both of which are assumed to be instantaneous and have associated costs of C_M and C_I, respectively. The machine may transition to a different deterioration level at each time. Degradation time-scales are therefore controlled by the probability transition matrix corresponding to the action(s) of non-renewal. A reward of C_P is received only if the processed job is non-defective (which is determined for all jobs during final testing before product sale).

$S = \{1,2, \ldots, N\}$
$A = \{a_1,a_2,a_3\}$: a_1—do nothing; a_2—inspect; a_3—renew
$O = \{o_1,o_2,o_3\}$: o_1—no defect; o_2—defect; o_3—no observation
$R(s,a,s') = (1 - \beta_{s'})C_P - I_M(a)C_M - I_I(a)C_I$
$O_{a2}(o_1|s') = 1 - \beta_{s'}$; $O_{a2}(o_2|s') = \beta_{s'}$
$O_a(o_3|s') = 1$ for $a \neq a_2$

where $I_M(a)$ and $I_I(a)$ are binary numbers equal to 1 when the machine is renewed and job inspection is performed, respectively, and 0 otherwise. The POMDP for $N = 3$ (three levels of deterioration) with $C_P = 1000$, $C_M = 10{,}000$ and three different values of C_I (parameter sets 1, 2 and 3 shown in Table 1), is solved using PERSEUS (as discussed in Section 2) and the optimal policy structure is shown in Fig. 2 (for $C_I = 150$). It is seen that the optimal policy has a *control limit* structure due to the following system properties.

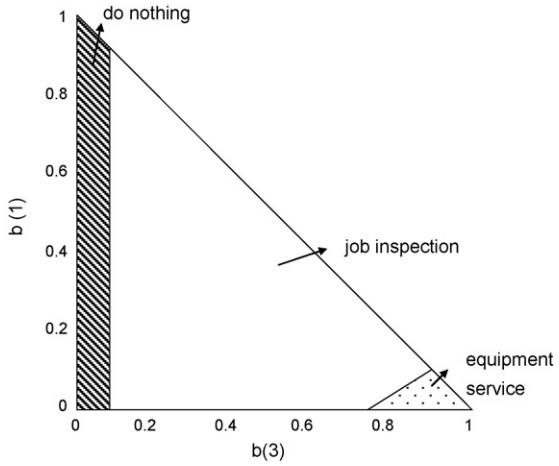

Fig. 2. Optimal policy for three-state single machine problem.

The policy can therefore be compactly represented as (12) for the above parameter values. It can be shown that a

(i) $r(s, a) - r(s1, a) \geq 0$ for $s < s1$
(ii) $p(s'|s, a) - p(s'|N, a) \geq 0$ for $s' \neq N$
(iii) $p(o_1|s, a) - p(o_1|s1, a) \geq 0$ for $s < s1$
(iv) $p(o_1|s = 1, a) \geq p(o_2|s = 1, a)$ $\forall a$

general system with N-state deterioration and above properties satisfies the monotonicity properties shown by [20]. Those noted above represent sufficient conditions for the monotonicity results to hold. The proofs are omitted due to space restrictions; however, they can be found on the authors' webpage

$$\text{policy} = \begin{cases} \text{do nothing} & \text{if} \quad b(3) \leq 0.09 \\ \text{renew} & \text{if} \quad \frac{4}{3}b(3) - 2b(1) \geq 1 \\ \text{inspect job} & \text{otherwise} \end{cases} \tag{12}$$

In the illustrations to follow, the concept of imperfect and incomplete observation is extended to multiple type of jobs operated on a single machine (Illustration 2) and finally to multiple type of jobs operated on multiple machines (Illustration 3). In both examples, the issue of propagation of defective jobs is central to the problem contributing the most to the problem size and computational complexity.

Illustration 2: The machine in Illustration 2 again operates on one job per time unit and undergoes degradation at each time, according to the Markov chain similar to the one in Illustration 1. However, the job cycles back to the machine until it undergoes the same operation L times, after which it leaves the system as product. (This re-entrant characteristic is observed in semi-conductor fabrication, for example, where multiple layers are deposited on silicon wafers. Therefore, jobs at various stages of production compete for the same resources.) The process is shown in Fig. 3, where $\tilde{a_l}$ refers

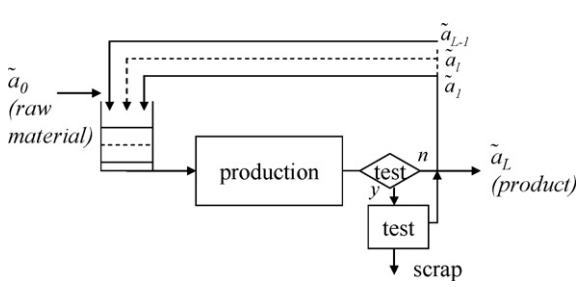

Fig. 3. Re-entrant flow problem.

Table 1
Parameter sets.

Parameter set number	C_P	C_M	C_0	C_l	C_l			C_R	Machine regime ransition	Beta
					$\tilde{a_0}$	$\tilde{a_1}$	$\tilde{a_2}$			
1	1000	10,000	–	150	–	–	–	–	$\begin{bmatrix} 0.99 & 0.0075 & 0.0025 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{bmatrix}$	[0.05 0.1 0.5]
2	1000	10,000	–	50	–	–	–	–	$\begin{bmatrix} 0.99 & 0.0075 & 0.0025 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{bmatrix}$	[0.05 0.1 0.5]
3	1000	10,000	–	500	–	–	–	–	$\begin{bmatrix} 0.99 & 0.0075 & 0.0025 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{bmatrix}$	[0.05 0.1 0.5]
4	2000	10,000	100	150	20	100	200	–	$\begin{bmatrix} 0.99 & 0.0075 & 0.0025 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{bmatrix}$	[0.05 0.1 0.5]
5	2000	10,000	100	150	100	200	500	–	$\begin{bmatrix} 0.99 & 0.0075 & 0.0025 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{bmatrix}$	[0.05 0.1 0.5]
6	2000	10,000	100	150	100	200	500	–	$\begin{bmatrix} 0.99 & 0.0075 & 0.0025 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{bmatrix}$	[0.05 0.25 0.75]

Parameter set	C_P	C_M			C_l		C_R		Machine regime transition ($\tilde{A}$, $\tilde{B}$, $\tilde{C}$)	β_s
		$\tilde{A}$	$\tilde{B}$	$\tilde{C}$	$\tilde{a}$	$\tilde{b}$	$\tilde{a}$	$\tilde{b}$		
7	1000	2000	2000	3000	25	25	175	175	$\begin{bmatrix} 0.99 & 0.01 \\ 0 & 1 \end{bmatrix}$	[0.10 0.60] ($\tilde{A}$) [0.25 0.75] ($\tilde{B}$) [0 0.5 0] ($\tilde{C}$)
8	1000	1000	1500	3000	25	25	175	175	$\begin{bmatrix} 0.99 & 0.01 \\ 0 & 1 \end{bmatrix}$	[0.25 0.75] ($\tilde{A}$) [0.25 0.75] ($\tilde{B}$) [0 0.5 0] ($\tilde{C}$)
9	1000	2000	2000	3000	25	25	175	275	$\begin{bmatrix} 0.99 & 0.01 \\ 0 & 1 \end{bmatrix}$	[0.10 0.60] ($\tilde{A}$) [0.25 0.75] ($\tilde{B}$) [0 0.50] ($\tilde{C}$)

to the job and subscript $l = 0,1, \ldots, L$ refers to the number of operations that the job has gone through. For simplicity they are called *l layers*. There is a queue before the operation where intermediate jobs wait for processing. Therefore, an added decision in this case is job scheduling, i.e. which of the intermediates $\tilde{a_l}$, $l = 0,1, \ldots, L-1$ to admit for processing. Each intermediate can be inspected if the decision-maker so chooses. Defect in all existing layers can be detected at the time of inspection. If found defective, the intermediate job is immediately scrapped/removed from the system. But if the inspection is not carried out at each time, then defective items would propagate through the system. The product brings revenue C_P only if all the L layers are non-defective. The costs for machine renewal, job inspection, processing the lth layer and raw material $\tilde{a_0}$ are C_M, C_I, C_l and C_0, respectively. The overall objective motivated by quality management is to devise an optimal machine renewal, job inspection and job scheduling policy that maximizes the infinite horizon profit from product sales. It is assumed that supply of $\tilde{a_0}$ is unlimited and final product $\tilde{a_L}$ is always tested. (The symbol $\tilde{a}$ is used to differentiate the job from action a. The nomenclature with *tilda* ($\sim$) is maintained to denote jobs and machines in the subsequent analysis, in order to avoid confusion with variables related to the state and action spaces.)

The above problem is interesting in the following ways:

i. It allows for analysis of the propagation and accumulation of defective jobs by means of a compact system representation.
ii. For very small and very large queue sizes, the system would behave as a serial production (Fig. 1(a)) and assembly system (Fig. 1(b)), respectively. For example, when no jobs are allowed to wait in the queue, one job remains in the system until completion. This is similar to the job going through a sequence of L operations in series. On the other hand, if a large number of intermediates are waiting in the queue for processing, then it acts more like an assembly system.
iii. Job inspection now serves two purposes, i.e. it not only provides information about the machine degradation, but also gives information about defective intermediates so that they can be picked out of the system to save the cost of additional operation on them.
iv. The job gathers value with each deposited layer. With better inspection and job scheduling, it is possible to reduce the number of good layers lost on bad products. This is because, a product is considered defective if at least one layer on it is defective.

Similar to the previous illustration, the system is modeled as a POMDP. The modeling details are included in the formulation and examples are presented for a three layer product, i.e. for $L = 3$. The problem is referred to as the *re-entrant flow* problem. It is worth noticing that the state dimension and consequently, the size of the state space are significantly larger in this case. This is because now the total number of intermediates in the system and the fraction of defective intermediates need to be accounted for. The action space has an added dimension of job scheduling and the state transition probability matrix also takes into account the probability of defect propagation.

5.1. Formulation as POMDP

State: The system at any time is fully characterized by the total number of jobs, the fraction of defective jobs and the deterioration level of the machine. Therefore,

$$s = [n_1, n_2, \ldots, n_{L-1}\ d_1, d_2, \ldots, d_{L-1}\ \Gamma]$$

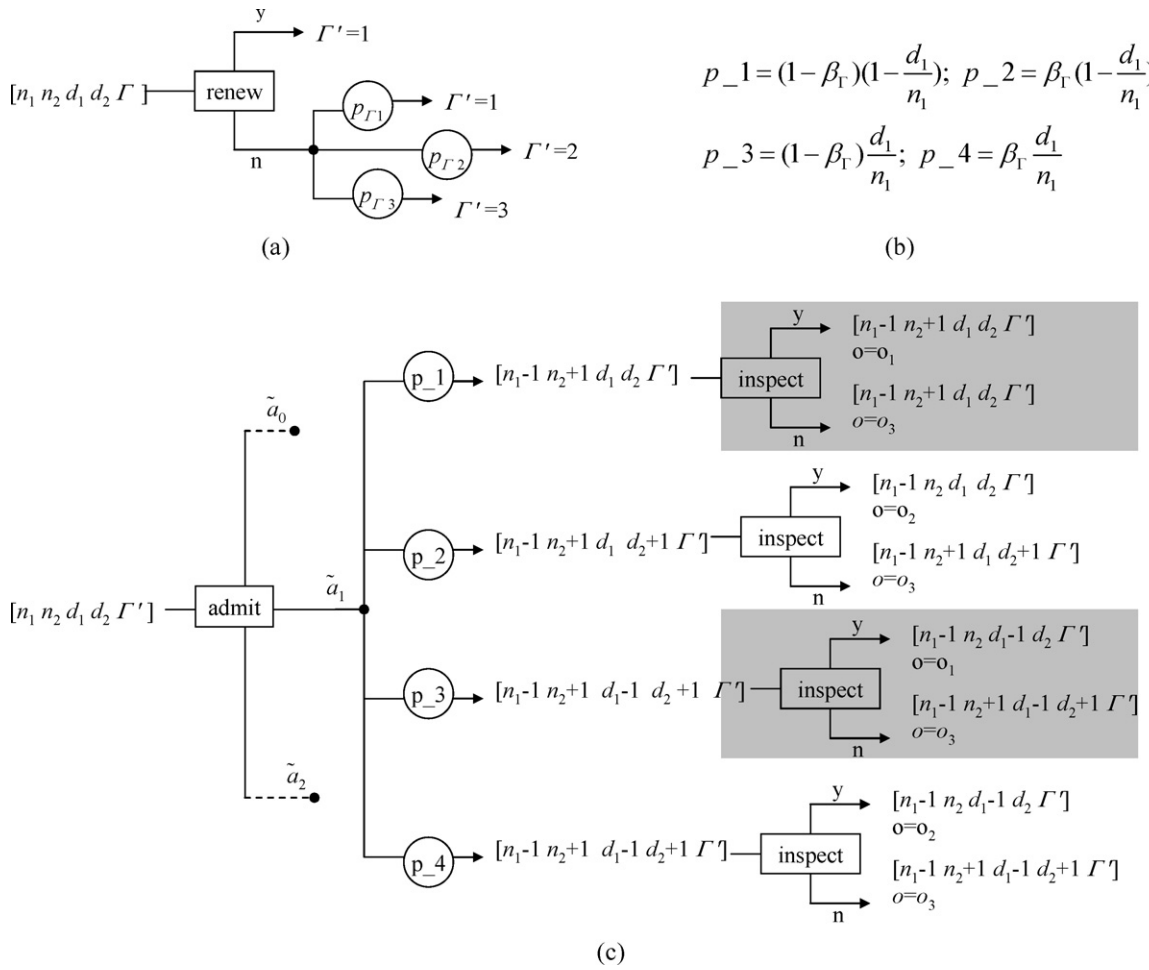

Fig. 4. State transition for the re-entrant flow problem for 3 levels of machine deterioration and $L = 3$. (a) Possible values of the machine condition at the next time step (Γ') given the current machine condition (Γ) and the renewal decision. (b) Probabilities associated with defect generation (β_Γ) and propagation (d_1/n_1). (c) Possible next states and observations depending upon job scheduling and job inspection decisions and realization of uncertainty.

$n_l \in \{0,1,2,\ldots,\eta\}$—total number of $\tilde{a_l}$ in the queue for $l = 1,2,\ldots,L-1$
$d_l \in \{0,1,2,\ldots,n_l\}$—number of defective $\tilde{a_l}$ in the queue for $l = 1,2,\ldots,L-1$
$\Gamma \in \{1,2,\ldots,N\}$—discrete integer representing the deterioration level of the machine.

The state space consists of all possible combinations of the above parameters. For instance, if $L = 3$ and if the maximum allowable number of jobs in the queue (η) is limited to 3 (i.e. $n_1 + n_2 \leq 3$), then there are following (n_1, n_2) combinations: (3,0) (2,1) (1,2) (0,3) (2,0) (1,1) (0,2) (1,0) (0,1) (0,0). For a particular value of n_1, say 3, d_1 can hold 4 possible values from 0, 1, 2, 3. With such calculations, the total number of possible combinations for $[n_1\ n_2\ d_1\ d_2]$ is 35. With 3 deterioration levels for the machine, the size of the state space is 105 (35×3). Similarly, the size of the state space for maximum queue sizes of 4 and 5 are 210 and 378, respectively.

Action/decision

$$a = [a_1\ a_2\ a_3]$$

where $a_1 \in \{0,1,2,\ldots,L-1\}$ pertains to the job scheduling decision (admit $\tilde{a_0}$, $\tilde{a_1}$, $\ldots$ $\tilde{a_{L-1}}$); $a_2 \in \{0,1\}$ pertains to job inspection decision (test (1) the processed job or not (0)); $a_3 \in \{0,1\}$ pertains to renewal decision (renew the machine (1) or not (0)). Assuming all final products are tested, the size of action space for $L = 3$ is $((3 \times 2) - 1) \times 2 = 10$.

Observation

$$o \in \{o_1, o_2, o_3\}$$

where o_1 represents 'no defect', o_2 'defect', and o_3 'no observation' as before.

Transition and observation probability matrices

For a queue length of 3, T is a $105 \times 105 \times 10$ matrix incorporating the 3 sources of uncertainty mentioned below:

(1) Machine regime switching—as shown in illustration 1, the machine can switch between regimes with certain probabilities in a non-deterministic manner.
(2) Defect generation—defect generation is probabilistic and the defect probability (β_s) is set by the regime in which the machine is operating.
(3) Error propagation—since not all intermediates are tested, the queue can contain defective intermediates, designated as d_1 and d_2 in the state description. Probability that a defective intermediate is picked and operated upon is given by q:

$$q = \frac{d_l}{n_l} \quad \text{For } \tilde{a}_l \text{ being operated}$$

For a queue length of 3, O is a $105 \times 10 \times 3$ matrix. It must be noted that the total number of the intermediates in the system $(n_1\ n_2\ \ldots\ n_{L-1})$ are always observable. The specific form of state

i. Generate a sample belief B with 10,000 belief points by simulating the system under the policies below:
   (i) FOMDP optimal policy assuming full observability
   (ii) Various periodic maintenance and inspection policies
   (iii) Random actions
   Initialize $V^{init} = \min(R(s,a))$

ii. Using B and V^{init}, run PERSEUS iterations as shown in section 2 for $\varepsilon = 0.01$. The converged value function is denoted by $V^{*}_{PERSEUS}$

iii. Use $V^{*}_{PERSEUS}$ to sample another belief set *Bbar* with 10,000 states.

iv. Set $V^{0} = V^{*}_{PERSEUS}$. Make one PERSEUS iteration to obtain V^{1}

v. If $\| V^{1} - V^{0} \|_{\infty} \leq 0.01$, stop, else set *B=Bbar*, $V^{init} = V^{1}$, go to step ii.

Fig. 5. Algorithm I for solving POMDP.

transitions for three levels of machine deterioration, $L = 3$ and $a_1 = 1$ is shown in Fig. 4. If $n_l = 0$, admitting $\tilde{a_l}$ for processing is not a permissible action. To avoid this situation while implementing the POMDP policy, $\tilde{a_{l-1}}$ is admitted.

Objective

The infinite horizon discounted profit/reward is given by (13):

$$V^{\pi *} = \max_{a_1, a_2 \ldots a_\infty} \sum_{t=1}^{\infty} \gamma^{t-1} (C_P I_P(a_t) - C_M I_M(a_t) - C_I I_I(a_t) - C_l I_l(a_t) - C_0 I_0(a_t)) \quad (13)$$

where γ is the discounting factor, a_t is the action at time t and all I's (I_P, I_M, I_I, I_l, I_0) are binary and are equal to 1 when a non-defective product is produced, when a maintenance job is run, when an intermediate job is tested when $\tilde{a_l}$ is run, and when raw material $\tilde{a_0}$ is admitted at time t, respectively.

The maximum allowable size of the queue largely governs the size of the state space which eventually controls the size of the problem. The above problem is solved for maximum queue lengths of 3, 4 and 5. Three different parameter sets (4–6) shown in Table 1(a) are considered. The parameter values are reasonably chosen to represent the trade-offs among different cost heads in a typical manufacturing environment. Since the queue length is constrained, holding cost/work in progress (WIP) cost is not considered. All problem instances are solved using the algorithm shown in Fig. 5. An initial belief sample set is obtained by using three different policies (i) optimal policy for the FOMDP problem, (ii) best periodic or block replacement and inspection policy, and (iii) policy to select a random action at each time. The algorithm uses PERSEUS (Section 2) iterations on the fixed initial belief set until ε convergence is achieved. A new sample is then obtained using the current value function for POMDP and the above is repeated (this is called one *sample iteration*). These sample iterations are carried out until the performance of two subsequent sample iterations is found to be δ-close for a randomly chosen test belief set.

The results are reported in Table 2. Table 2 includes the size of the problem for the queue sizes considered. $|V|$ is the size of the optimal policy, i.e. the number of gradient vectors α_n^i, $i = 1, 2, \ldots, |V_n|$. The reported profit figure for the POMDP is the average profit obtained by starting in $s = 1$ ([0 0 0 0 1] – no jobs in the system and best machine condition) and following the optimal policy. Average is taken over 100 experiments in all cases. The profit figure reported for the FOMDP case starting in state $s = 1$, acts as a loose upper bound, which cannot be achieved. The difference in the

Table 2
Problem sizes and results for all three illustrations (discounting factor $\gamma = 0.99$ for all cases).

	$\|S\|$	$\|A\|$	$\|O\|$	Parameter set	$\|V\|$	FOMDP profit, $s=1$ ($\times 10^4$)	POMDP* profit, $s=1$ ($\times 10^4$)	Periodic heu, $s=1$ ($\times 10^4$)	QMDP approx, $s=1$ ($\times 10^4$)	profit with linear V, $s=1$ ($\times 10^4$)	Size of decision tree
Single machine	3	4	3	1	73	8.90 ± 0.39	8.49 ± 0.49	8.38 ± 0.89	8.04 ± 1.12	8.44 ± 0.58	4
				2	81	8.90 ± 0.39	8.61 ± 0.38	8.38 ± 0.89	8.09 ± 1.17	8.54 ± 0.44	
				3	84	8.90 ± 0.39	8.41 ± 0.20	8.38 ± 0.89	8.09 ± 1.17	8.21 ± 0.26	
Re-entrant flow (queue = 3)	105	10	30	4	423	3.19 ± 0.64	2.98 ± 0.45	2.45 ± 0.16	2.86 ± 0.57	2.96 ± 0.61	15
				5	442	1.64 ± 0.48	1.35 ± 0.23	0.78 ± 0.54	0.627 ± 0.8	1.37 ± 0.46	
				6	360	1.30 ± 0.68	0.99 ± 0.08	0.57 ± 0.74	0.92 ± 0.70	0.97 ± 0.28	
Re-entrant flow (queue = 4)	210	10	45	4	390	3.20 ± 0.61	2.91 ± 0.54	2.53 ± 0.21	2.75 ± 0.75		
				5	387	1.66 ± 0.61	1.35 ± 0.23	0.91 ± 0.45	1.09 ± 0.83		
				6	365	1.33 ± 0.75	1.00 ± 0.34	0.78 ± 0.63	0.68 ± 0.85		
Re-entrant flow (queue = 5)	378	10	63	4	300	3.20 ± 0.51	2.95 ± 0.71	2.80 ± 0.19	2.84 ± 0.85		
				5	292	1.66 ± 0.61	1.35 ± 0.23	0.81 ± 0.44	0.90 ± 0.73		
				6	383	1.36 ± 0.75	1.00 ± 0.34	0.62 ± 0.64	0.72 ± 0.65		
Hybrid-flow system	352	32	27	7	83	1.11 ± 0.65	1.00 ± 0.46	0.40 ± 0.12	0.34 ± 0.12	0.50 ± 0.16	19
				8	83	1.09 ± 0.34	0.91 ± 0.15	0.55 ± 0.18	0.45 ± 0.18	0.71 ± 0.22	
				9	74	1.05 ± 0.23	0.88 ± 0.10	0.73 ± 0.18	0.71 ± 0.31	0.45 ± 0.49	

two values provides the extent to which the partial observability affects the performance. For parameter set 5, the processing costs for layers 1, 2 and 3 are higher as compared to those for parameter set 4. This causes reduction in the overall profit as compared with parameter set 4 but evidently no difference in the optimal policy for the fully observable problems. However, in the partially observable case, the (near) optimal policy instructs an increase in the average queue size (number of intermediates in the queue at any time) with an increase in the processing cost. This is because the system is more cautious about running an expensive intermediate when the machine deterioration level is high. In parameter set 6, the defect probabilities associated with machine deterioration levels 2 and 3 are increased which leads to further reduction in the overall profit. The general characteristics of the FOMDP policies are discussed in further details.

5.2. Characterization of FOMDP policy

In order to understand the system behavior, the optimal policy corresponding to the FOMDP problem is analyzed. It is seen that the machine renewal, job inspection and job scheduling decisions are mutually correlated and therefore a compact representation of the policy is not possible, unlike in the single operation case previously considered. The general characteristics of the optimal policy are discussed further:

1. The optimal policy is a strong function of the probability of defect *generation* and that of defect *propagation*. The former is the determined by β_s, the defect probability associated with machine deterioration level s and the latter is the probability that the incoming job is already defective. The latter is given by d_l/n_l. Also the term expensive intermediate is used to denote a_l with relatively large l.
2. The machine is renewed when defect generation probability is high (3 and 2) and defect propagation probability is low.
3. Job inspection is carried out when both of the above probabilities are high and when an expensive intermediate is admitted for processing. Note that according to problem specification, $\tilde{a_L}$ is always tested.
4. An expensive intermediate $\tilde{a_l}$ is admitted for processing whenever $n_l \neq 0$ and defect generation and propagation probabilities are low. Otherwise, a_{l-1} is picked for processing.

For the cost values considered, the systems tends to keep a small number of intermediates in the system as guided by the optimal policy. This is the reason why the optimal policy and the performance of the reentrant flow problems with varying limits on queue sizes (3, 4 and 5) are the same (please see Table 2).

As for the structure of the POMDP policy, trends similar to the FOMDP policy are observed. However, job inspection also serves the purpose of determining the machine condition, which is not known with certainty along with the fraction of defective intermediates. The policy space is very large in the case of POMDP problem to be represented in a meaningful way. Since the policy is characterized by the value function, some conjectures on the structure of the value function for re-entrant flow problem are presented in Section 5.

The case with multiple machines in the *hybrid-flow* example is presented below. It combines the re-entrant flow feature with serial flow topology as shown further.

Illustration 3: There are three machines ($\tilde{A}$, $\tilde{B}$, $\tilde{C}$) similar to the one in Illustration 1 that undergo degradation according to separate independent Markov chains and defect probabilities. The machines are in series and the jobs have a pre-defined order of operation as shown below:

1. Three layers at machine $\tilde{A}$
2. Two layers at machine $\tilde{B}$
3. One layer at machine $\tilde{C}$

Here *layer* again refers to one machine operation for simplicity. The jobs being operated at machines $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ are designated as $\tilde{a_l}$, $\tilde{b_l}$, and $\tilde{c_l}$ respectively, where the subscript l refers to the number of layers already deposited. The process is schematically shown in Fig. 6. For simplicity, it is assumed that all machines can transition to a lower deterioration level at each time unit but the result is only reflected on the next job to be processed and not on the current job. There is an inspection station after machines $\tilde{A}$ and $\tilde{B}$. If an intermediate is tested and found defective, it is sent to a repair station where only the topmost layer can be repaired. It costs C_R for each repair and the repaired job is returned to the system for further processing. It takes 2 time units for an operation at $\tilde{A}$, 3 units at $\tilde{B}$ and 6 units at $\tilde{C}$. Unlike in Illustration 2, there is no possibility of queuing the jobs in this system. Jobs are fed sequentially, there is one job at each machine at any time and product is obtained every 6 time units. It is assumed that machine maintenance, job inspection and rework take negligible time.

One additional feature we introduce is that, if a defect is detected in the last layer of $\tilde{a}$ or $\tilde{b}$, then repair is not required and it can be repaired by the subsequent operation ($\tilde{B}$ or $\tilde{C}$ resp.) without any additional cost. Reward is received only when all the layers in the final product $\tilde{c_1}$ are non-defective. The objective is to maximize the average infinite horizon discounted profit while obtaining an optimal renewal policy and job inspection policy for all three operations. It is assumed that the final product is always tested. The feature of the above problem that a defect in the last layer of $\tilde{a}$ and $\tilde{b}$ can be corrected by subsequent operations is seen in automotive assembly where downstream correction of errors in physical dimensions of jobs is possible. Due to this feature, the maintenance decisions downstream affect the upstream processing. The system is balanced since a job spends exactly 6 time units at each machine. A time counter $t(t = 1, 2, \ldots, 6)$ is used to designate the time elapsed since the job first entered the machine. It is assumed that the machine can be serviced and job can be inspected only at the end of a run. Therefore, maintenance and inspection at machine $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ can be done when t is a multiple of 2, 3 and 6, respectively. The formulation of the problem as a POMDP is presented below.

5.3. Formulation as POMDP

State

The system is fully characterized by the following:

$s = [\Gamma_{\tilde{A}}\ \Gamma_{\tilde{B}}\ \Gamma_{\tilde{C}}\ t\ \text{defect}_{\tilde{a}}\ \text{defect}_{\tilde{b}}\ \text{defect}_{\tilde{c}}]$

where

$\Gamma_{\tilde{A}}, \Gamma_{\tilde{B}}, \Gamma_{\tilde{C}} \in \{1,2,\ldots,N\}$ represent the regime of machines $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$

$t \in \{1,2,\ldots,6\}$ is the time counter which goes from 1 to 6 and then resets to 1.

$defect \in \{0,1\}$ shows whether the jobs $\tilde{a}$, $\tilde{b}$, $\tilde{c}$ that are being processed have one or more defective layer(s) (1) or not (0)

Action/decision

$a = [\text{renew}_{\tilde{A}}\ \text{renew}_{\tilde{B}}\ \text{renew}_{\tilde{C}}\ \text{test}_{\tilde{a}}\ \text{test}_{\tilde{b}}]$

$renew_i \in \{0,1\}$ for $i = \tilde{A}, \tilde{B}, \tilde{C}$, pertains to whether to renew the machine i (1) or not (0)

$test_i \in \{0,1\}$ for $i = \tilde{a}, \tilde{b}$ pertains to whether to test the processed job i(1) or not (0)

Observation

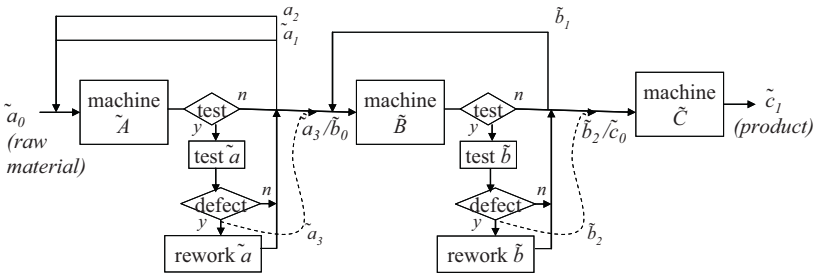

Fig. 6. Hybrid-flow system.

$o = [o_{\tilde{a}}\ o_{\tilde{b}}\ o_{\tilde{c}}]$
where $o_i \in \{o_1, o_2, o_3\}$ for $i = \tilde{a}, \tilde{b}$
$o_i \in \{o_1, o_2\}$ for $i = \tilde{c}$

This is because the final product is always tested
Transition probability matrix
T incorporates the following sources of uncertainty

a. Machine regime switching—as shown in Illustration 1 (Fig. 4(a)), the machines can switch between regimes with certain probabilities in a non-deterministic manner.
b. Defect generation—defect generation is probabilistic and the defect probability (β_s) is set by the regime in which the machines are operating.

Objective
Maximization of the infinite horizon discounted profit/reward given by (14):

$$V^{\pi*} = \max_{a_1, a_2, \ldots a_\infty} \sum_{t=1}^{\infty} \gamma^{t-1} \{C_P I_P(a_t) - \sum_{i=\tilde{A}}^{\tilde{C}} C_M^i I_M^i(a_t) - \sum_{j=\tilde{a}}^{\tilde{b}} C_I^j I_I^j(a_t) - C_0 I_0(a_t)\} \quad (14)$$

γ is the discounting factor, a_t is the action at time t and all I's (I_P, I_M, I_I, I_0) are binary and are equal to 1 when a non-defective product is produced, when a maintenance job is run, when an intermediate job is tested, and when raw material $\tilde{a}_0$ is admitted at time t, respectively.

The above is solved for two possible regimes ($N = 2$) for each machine and the parameter sets 7, 8 and 9 shown in Table 1. The POMDP is solved using the algorithm shown in Fig. 5 and the results are reported in Table 2. Similar to the results for Illustration 2, the FOMDP profit is also reported to highlight the loss due to the partial observability in each case. Similar to the re-entrant flow case, the policy space is complicated leading to difficulties with compact policy representations even for the fully observable problem. For the parameter sets 7, 8 and 9 shown in Table 1, the characteristics of the optimal FOMDP policy are as follows:

1. When the time counter $t = 2$, only machine $\tilde{A}$ can be renewed and $\tilde{a}_1$ inspected. The optimal policy pertaining to all parameter sets (7, 8 and 9) is to never renew the machine and always inspect the job $\tilde{a}_1$.
2. When $t = 3$, the optimal policy is never to renew machine $\tilde{B}$ and always inspect job $\tilde{b}_1$
3. When $t = 4$, the optimal policy is to not renew machine A but to inspect job $\tilde{a}_2$ only when $\tilde{a}_1$ is non-defective. This is expected since only the top layer can be repaired upon inspection. However since all $\tilde{a}_1$s are tested at $t = 2$, this situation never arises in the fully observable case.
4. When $t = 6$, only machine C is renewed when machine is in deterioration level 2 and the incoming job is non-defective, for parameter sets 7 and 9. For parameter set 8, machine $\tilde{A}$ is also renewed when in deterioration level 2. This difference can be attributed to the lower renewal cost in the case of parameter set 8.

The high dimensionality associated with the (near) optimal policy for the partially observed hybrid flow problem prevents a compact representation. Some conjectures on the POMDP policy together with those on the partially observed reentrant flow problem are presented in the following section. Alternative policies are also discussed in order to establish the goodness of the POMDP solution.

6. Discussion on results and policy structure

6.1. Comparison

In order to understand the advantages of a rigorous approach to solving this class of problems, the following is used as a basis of comparison:

(i) FOMDP solution—the performance of the MDP, assuming that the system state is fully observed, establishes a non-achievable upper bound to the POMDP solution and the gap between the performances show the extent to which the partial observability affects the performance of the system. It also helps understand the policy structure and the relevant region of the state space in certain cases. The optimal discounted infinite horizon reward for starting in $s = 1$ for all illustrations and parameter sets is reported in Table 2. For the single machine problem, the changing inspection cost has no effect on the solution since the state is fully observed and inspection is never carried out. (State $s = 1$ in all illustartions, represents the starting state with the best machine regime(s) and no jobs in the system.)
(ii) Q^{MDP} approximation—a lower bound on the close-to-optimal solution of the POMDP is established by using a simple function approximation scheme [10]. The optimal Q-function associated with the fully observable MDP is shown in (15), where V^{MDP} is the optimal value function. The Q-function associated with the partially observable problem for each belief state and action is then approximated as shown in (16). The approximate Q function can then serve as a basis for the decision-making. The resulting performance is contained in Table 2. Note that the optimal policy corresponding to the FOMDP does not contain the inspection decision for gauging machine regime when states are fully observed. That is the reason why, at times, the performance of this approximation is worse than that of periodic policy as discussed next.

$$Q^{MDP}(s, a) = r(s, a) + \sum_{s'} p(s'|s, a) V^{MDP}(s) \quad (15)$$

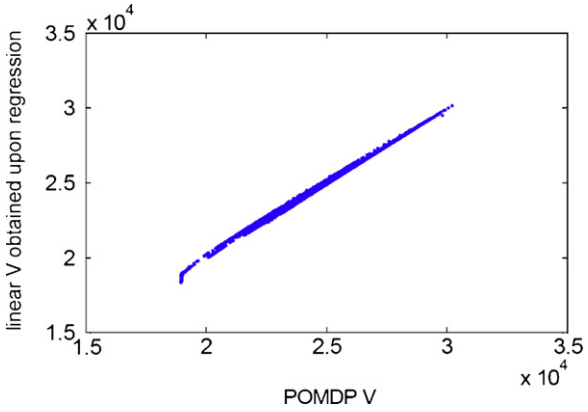

Fig. 7. Value function v/s linear approximation plot.

$$\widehat{Q}(b,a)=\sum_{s=1}^{N} Q^{MDP}(s,a)b(s) \tag{16}$$

(iii) Periodic maintenance and (or) inspection policies—as mentioned in Section 2, the periodic policies are easy to implement and form the industrial standard for maintenance and job inspection decisions. For single machine problem, a periodic maintenance policy is optimal when inspection costs are prohibitively high (shown as the unobservable case in Section 2). As seen in Table 2, the performance of the POMDP for the single machine problem drops with increasing cost of inspection. For parameter set 3, the periodic policy gives a performance similar to that of the POMDP.

6.2. Empirical findings and conjectures

In addition to the rigorous results, the following empirical observations are reported for the illustrations that were studied:

(i) In the relevant belief space, the close-to-optimal value function for the partially observed re-entrant flow problem and for the single machine could be well represented as a linear function of the belief states.
(ii) In the relevant belief space, the close-to-optimal decision rule for single machine, reentrant flow and hybrid flow could be well represented as a decision tree of size substantially smaller than the dimension of the belief space.

The value function in this case can be claimed as only close-to-optimal because there are no guarantees for optimality of solutions yielded by PERSEUS in solving large size POMDPs. Relevant belief space refers to the set of belief points that are visited by following the close-to-optimal policy. Fig. 7 is a plot between the actual value function v/s that obtained by a linear regression for the re-entrant flow case and parameter set 4. The value function is plotted for the belief states that are visited when POMDP close-to-optimal policy is followed. The band around 45° line represents a good fit. It is seen that the points mostly lie within that band.

6.3. Value function approximation

It is well-known that for a general infinite horizon POMDP, the optimal value function can be closely approximated as a piecewise linear and convex function [11] as shown in (8). From the finding in (i) above, it turns out that in the relevant region of the belief space, the value function can be approximated as a single linear function shown in (17) where w_s are the weights for each system state.

i. Arbitrarily initialize W^0, where W is the vector of w_i, for i=1,2,... $|S|$. Set $N = |S|$ and $B = I_N$, where I_N is the identity matrix of size N
ii. Given $V(b) = bW^0 \ \forall b$, run value iteration (equation 4) for belief set B, until $|V^{i+1} - V^i|_\infty \le 0.01$. Denote resulting V as V_B^*
iii. Determine $W^1 = (B^T B)^{-1} B^T V_B^*$
iv. Use W^1 to randomly sample another belief set *Bbar* with 10,000 belief points such that rank(*Bbar*) $\ge N$.
v. Run one value iteration step on *Bbar* to obtain V_{Bbar}.
vi. If $|V_{Bbar} - V_B^*|_\infty \le 0.01$, then stop, else set *B*=*Bbar*, W^0=W^1 and go to step ii

Fig. 8. Algorithm II to solve POMDP with linear value function approximation.

$$\text{s.t.} \quad \widetilde{V}(b)=\sum_{s=1}^{|S|} w_s b(s) \tag{17}$$

Therefore, the close to optimal value function can be represented as a set of weights $\{w_1, w_2, \ldots, |S|\}$. In order to determine these weights, value iterations can be carried out on $|S|$ different belief points where $|S|$ is the dimension of the belief state. Fig. 8 shows a simplified version of algorithm I (Fig. 5) which utilizes the linear value function. The results from algorithm II are also reported in Table 2. It is seen that the performance for the single machine and re-entrant flow cases are comparable with that of algorithm I.

6.4. Decision-tree analysis

A decision tree serves as a good tool to represent a policy. The size of the decision-tree is determined by the number of levels at which decisions are determined by conditions on the state variables. The size of the decision-tree is governed by two factors:

(i) The number of actual states s visited while following the close-to-optimal policy. Let us say $S_v \in S$ is the set of actual states visited and $S_{v'} = S/S_v$. The size of the decision-tree depends on the size of S_v. When the close-to-optimal policy is implemented, the belief dimensions corresponding to $S_{v'}$ contribute less and less to decision-making.
(ii) The level of similarity between actual states s—due to the similarity, a cluster of states would correspond to the same (near)optimal action. In the region of belief states, the states belonging to these clusters would form hyper-planes and lead to a decision-tree of much lower dimension. The sizes of the decision-trees are also reported in Table 2 and are substantially smaller than the size of S_v. This indicates the formation of clusters of states that behave in a similar manner.

Consequently, the decision tree allows for compressing the information corresponding to a potentially large policy and thus helps representing it in a compact manner.

7. Conclusions

Judging by the research efforts in the area of partially observable degradation of manufacturing equipments and costly inspection, the extension of the concepts to multiple operations is important. In this work this problem is addressed for different a re-entrant and a hybrid flow topology. The significance of rigorous treatment of this class of problems is demonstrated for the illustrated cases by

comparing the results with those of heuristic methods. The POMDP formulations result in significant improvements (up to ∼40% as seen in Table 2) in performance over those of best heuristics. However, the POMDP problem grows fast with increasing problem sizes. Therefore, characterization of the (near) optimal policies is an important direction for future work in this area. Decision tree analysis serves as a promising direction in this regard.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Grant No. 20110006839). This work was also partially supported by the Advanced Biomass R&D Center (ABC) of Global Frontier Project funded by the Ministry of Education, Science and Technology (ABC-2011-0031354).

References

[1] S. Osaki, in: S. Osaki (Ed.), Stochastic Models in Reliability and Maintenance, vol. 1, Springer-Verlag, 2001.
[2] J.S. Ivy, S.M. Pollock, Marginally monotonic maintenance policies for a multi-state deteriorating machine with probabilistic monitoring, and silent failures, IEEE Transactions on Reliability 54 (3) (2005) 489–497.
[3] R.D. Smallwood, E.J. Sondik, The optimal control of partially observable Markov processes over a finite horizon, Operations Research 21 (5) (1973) 1071–1088.
[4] C. Xiong, Y. Rong, R.P. Koganti, M.J. Zaluzec, N. Wang, Geometric variation prediction in automotive assembling, Assembly Automation 22 (3) (2002) 260–269.
[5] J. Lee, S. Unnikrishnan, Planning quality inspection operations in multistage manufacturing systems with inspection errors, International Journal of Production Research 38 (1) (1998).
[6] T.J. Spaan Matthijs, N. Vlassis, Perseus: randomized point-based value iteration for POMDPs, Journal of Artificial Intelligence Research 24 (2005) 195–220.
[7] T. Smith, R. Simmons, Heuristic search value iteration for POMDPs, in: Proceeding of Uncertainty in Artificial Intelligence, 2004.
[8] J. Pineau, G. Gordon, S. Thrun, Point-based value iteration: an anytime algorithm for POMDPs, in: Proc. Int. Joint Conf. on Artificial Intelligence, Acapulco, Mexico, 2003.
[9] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley and Sons, New York, 1994.
[10] M. Hauskretch, Value-function approximations for partially observable Markov decision processes, Journal of Artificial Intelligence Research 13 (2000) 33–94.
[11] E.J. Sondik, The optimal control of partially observable Markov processes over the infinite horizon: discounted Costs, Operations Research 26 (2) (1978) 282–304.
[12] S.S. Mandroli, A.K. Shrivastava, Y. Ding, A survey of inspection strategy and sensor distribution studies in discrete-part manufacturing processes, IIE Transactions 38 (4) (2006) 309–328.
[13] M. Girshick, A.N.D.H. Rubin, A Bayes' approach to a quality control model, Annals of Mathematics and Statistics 23 (1952) 114–125.
[14] S. Ross, Quality control under Markovian deterioration, Management Science 17 (1971) 587–596.
[15] S. Ehrenfeld, On a sequential Markovian decision procedure with incomplete information, Computers and Operations Research 3 (1976) 39–48.
[16] C. White, Optimal control-limit strategies for a partially observed replacement problem, International Journal of Systems Science 10 (1979) 321–331.
[17] C. Derman, Optimal replacement rules when changes of state are Markovian, in: R. Bellman (Ed.), Mathematical Optimization Techniques, Univ. of California Press, Berkeley, CA, 1963, pp. 1–3.
[18] W. Pierskalla, J. Voleker, A survey of maintenance models: the control and surveillance of deteriorating systems, Naval Research Logistics Quarterly 23 (1976) 353–388.
[19] G.E. Monahan, A survey of partially observable Markov decision processes: theory, models and algorithms, Management Science 28 (1) (1982) 1–16.
[20] G.E. Monahan, Optimal stopping in a partially observable markov process with costly information, Operations Research 28 (1980) 1319–1334.