OPTIMAL POLICIES FOR CONTROLLING THE IN-PROCESS INVENTORY IN A SERIAL MANUFACTURING SYSTEM WITH INSPECTION AND REWORK

Solon Venâncio de Carvalho
Rita de Cássia Meneses Rodrigues
Nandamudi Lankalapalli Vijaykumar
Paulo Renato de Morais

Laboratório Associado de Computação e Matemática Aplicada (LAC)
Instituto Nacional de Pesquisas Espaciais (INPE)
Av. dos Astronautas, 1758 - Caixa Postal 515
12201-970 - São José dos Campos - SP, Brazil
e-mail: {solon, rita, vijay, prenato}@lac.inpe.br

ABSTRACT

A manufacturing system consisting of two machines operating in series and an in-process inventory is considered. Products must be inspected after being processed by the machines. After inspection, products are either discarded, reworked, or are allowed to proceed to the next stage. The machines are subject to failure during use. The time-to-failure, the processing time, and the repair time of the machines, and the rework time of each product are considered to be exponentially distributed. The in-process inventory is controlled by a policy that decides dynamically whether to activate the first machine or not. The cost structure includes a processing cost, a repair cost, a storage cost, a restart cost, a starving cost, a cost if a product is lost, and a reward for delivering the final product. A Markov Decision Model is used to maximize the long-run average revenue per unit time. The optimal policy is compared with the optimal (s, S) policy and with the uncontrolled model. Numerical results are presented.

RESUMO

Considera-se um sistema de manufatura com duas máquinas M_1 e M_2 operando em série e um estoque intermediário. Os produtos processados pelas máquinas passam por uma inspeção, onde podem ser aceitos, descartados ou enviados para reprocessamento. As máquinas estão sujeitas a falhas durante o uso. O tempo até a quebra, o tempo de reparo, o tempo de processamento das máquinas e o tempo para reprocessar cada produto são considerados exponencialmente distribuídos. O estoque intermediário é controlado por uma política que decide dinamicamente sobre o bloqueio ou não da máquina M_1. A estrutura de custos inclui um custo de estocagem, um custo de reativação das máquinas, um custo de ociosidade da máquina M_2, um custo de processamento, um custo de reparo, um custo por perder um produto e um ganho por produto fabricado. Utiliza-se um Modelo Markoviano de Decisão para obter uma política que maximiza a receita média do sistema a longo prazo por unidade de tempo. A política ótima é comparada com a política (s, S) ótima e com a situação em que não se efetua nenhum controle sobre o modelo. Resultados numéricos são apresentados.

Key words: Inventory Models - Markov Chains - Serial Manufacturing Systems.

1. Introduction

Consider a manufacturing system with two machines M_1 and M_2 continuously producing a single product. Each product is processed first by machine M_1 and then by machine M_2. An in-process inventory is used to reduce machine M_2 idleness. The storage capacity N of the in-process inventory is finite. After being processed by the machines, each product must be inspected. After inspection, for each machine M_i , i = 1, 2, a product either (a) is discarded if it presents major defects (with probability p_{di}), (b) is sent back to the machine for rework if it presents minor defects (with probability p_{ri}), or (c) it proceeds to the next stage in case no defects are found (with probability p_{gi}). If the machine involved is M_1, the next stage would be storing the product in the in-process inventory or loading the product directly into the second machine M_2 (depending on the availability of M_2). In case the machine involved is M_2, the next stage would be sending the finished product to its final destination (see Figure 1).

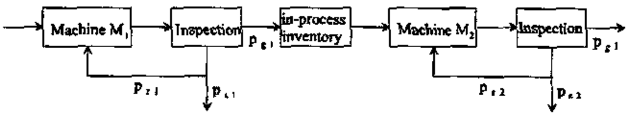

Figure I - Overview of the system

As the machines may fail during use, the main objective is to derive an optimal control policy of the in-process inventory. This control is achieved by blocking or unblocking machine M_1. The cost structure includes a processing cost, a repair cost, a storage cost, a machine restart cost, a starving cost, a cost if a product is lost, and a reward for delivering the final product. A Continuous Time Markov Decision Model is used to maximize the long-run average revenue per unit time. The control policies considered to block or unblock machine M_1 take into account not only the size of the in-process inventory but also the observed state for each machine (blocked, starved, processing, broken, reworking a product, or broken during rework). This paper extends some ideas of [Hwang and Koh, 1992] and [Gopalan and Kannan, 1994], using the mathematical approach presented in [Carvalho et alii, 1993].

[Hwang and Koh, 1992] formulated a Markovian model to obtain an (s, S) policy that minimizes the average cost of in-process inventory between two machines without considering rework or inspection. [Carvalho et alii, 1993] extended these results, considering control policies that take into account not only the size of the in-process inventory but also the state of each machine (blocked, starved, processing, or broken). This idea yielded a minimum average cost that is less than the one obtained in [Hwang and Koh, 1992]. [Gopalan and Kannan, 1994] discussed a similar model including inspection and rework. [Carvalho et alii, 1995] introduced an in-process inventory in Gopalan and Kannan's model and used a Markov Decision Model to obtain the optimal control policy.

The present paper compares the optimal policy obtained by [Carvalho et alii, 1995] with the optimal (s, S) policy which only considers the size of the in-process inventory and not the machines' states in the decision process, and with the same model with no control over it. The objective is to find the gain in using the optimal policy instead of simpler policies.

2. Model description

The set of possible states for each machine M_1 and M_2 is $E_M = \{W, P, R, B, BR\}$, where P denotes "processing a product", R denotes "reworking a product", B denotes "broken", and BR denotes "broken during rework". For machine M_1 the state W denotes "waiting to be unblocked" and for machine M_2 the state W denotes "waiting while the in-process inventory is empty".

The decision to block or unblock machine M_1 takes into account the state of the system. The state space of the system is defined as $E = \{(m_1, n, m_2) / m_1 \in E_{M1}, n \in \{0, 1, \ldots, N\}, m_2 \in E_{M2}\}$, where m_i is the state of machine M_i, $i = 1, 2$, and n is the size of the in-process inventory (including the product that eventually is being processed on machine M_2).

The dynamic behavior of the system is described by the change of its states. Each time the system changes its state, the new state configuratioo must be observed in order to decide what action is to be taken, i.e. whether the machine M_1 is to be blocked or not. M_1 must be unblocked whenever M_2 is in the W state ($n = 0$), and must be blocked whenever the in-process inventory is at its maximum capacity ($n = N$). Therefore, for each state $i = (m_1, n, m_2) \in E$, the space of possible actions is:

$$A(i) = \begin{cases} \{D\} & \text{if } n=0 \\ \{D,B\} & \text{if } 0<n<N \\ \{B\} & \text{if } n=N \end{cases}$$

where D and B denote respectively unblock and block M_1.

For each machine M_i, $i = 1, 2$, the processing time of each product, the time-to-failure, the repair time, the rework time, the time-to-failure during rework, and the repair time for failures during rework are independent, exponentially distributed random variables with rates β_i, λ_i, μ_i, β_{ri}, λ_{ri}, and μ_{ri}, respectively.

Products must be inspected after being processed by the machines. After inspection, products are either discarded with probability p_{si}, reworked with probability p_{ri}, or are allowed to proceed to the next stage with probability p_{gi}, for $i = 1, 2$.

In order to obtain a control policy that maximizes the long-run average revenue per unit time, this system is modeled as a Continuous Time Markov Decision Process. Given that at a decision epoch the system is in state $i \in E$ and action $a \in A(i)$ is chosen, $\tau(i,a)$ is the expected time until the next decision epoch, $p(i,j,a)$ the probability that in the next decision epoch the state will be $j \in E$, and $R(i,a)$ the expected revenue obtained until the next decision epoch.

$\Lambda_{i,j}(a)$ is defined as the transition rate from state e_i to state e_j (e_i, $e_j \in E$) when the last chosen action was $a \in A(i)$. The algorithm used to obtain transition rates $\Lambda_{i,j}(a)$ that describes the system behavior can be found in [Carvalho et alii, 1995]. Using the transition rates $\Lambda_{ij}(a)$, it is easy to obtain the total rate of output from each state given by $\Lambda_i(a) = \sum_{j \neq i} \Lambda_{ij}(a)$, and so the transition probabilities are given by $p(i,j,a) = \Lambda_{ij}(a)/\Lambda_i(a)$, and the expected time between transitions is given by $\tau(i,a) = 1/\Lambda_i(a)$. The expected revenue is given by:

$$R(i,a) = G(i,a) - C_h(i,a) - C_{p1}(i,a) - C_{p2}(i,a) - C_{b1}(i,a) - C_{b2}(i,a) - C_{r1}(i,a) - C_{r2}(i,a) - C_{l1}(i,a) - C_{l2}(i,a) - C_s(i,a),$$

where $G(i,a)$, $C_h(i,a)$, $C_{pi}(i,a)$, $C_{bi}(i,a)$, $C_{ri}(i,a)$, $C_{li}(i,a)$, $C_s(i,a)$ represent respectively the expected reward for delivering the final product, the expected storage cost, the expected processing cost for machine M_i, the expected repair cost for machine M_i, the expected restart cost for machine M_i, the expected cost of losing a product in machine M_i, $i = 1, 2$, and the expected starving cost for machine M_2, incurred until the next decision epoch, given that at the decision epoch the system is in state $i \in E$ and action $a \in A(i)$ is chosen. The expression for these costs can be found in [Carvalho et alii, 1995].

With the values of $\tau(i,a)$, $p(i,j,a)$, and $R(i,a)$, the Value-Iteration Algorithm [Tijms, 1986] was used to obtain the policy that maximizes the long-run average revenue per unit time.

3. Numerical Results

As an illustration, consider the input data shown in Table 1. Besides these values, the following data have been considered:

Storage cost: $c_h = 5$

Sale price of a product: $g = 50$

	Machine M_1	Machine M_2
Processing rate	8	5
Breakdown rate	0.5	1
Repair rate	2	1
Product rework rate	12	10
Breakdown-during-rework rate	0.5	1
Repair-of-breakdown-during-rework rate	2	1
Restart cost	10	10
Processing cost	50	50
Repair cost	50	50
Starving cost	-	20
Inspection: Cost of losing a product	50	70
Inspection: Probability of losing a product	0.05	0.05
Inspection: Rework probability	0.05	0.05
Inspection: Probability of no defect	0.9	0.9

Table I - Input data

Figure 2 shows the variation of the minimum average revenue in terms of the maximum size of the inventory (N). As can be seen, the size of the inventory up to a certain maximum limit is very important to improve the average revenue. In this example, it is not worth using an inventory size larger than 6.

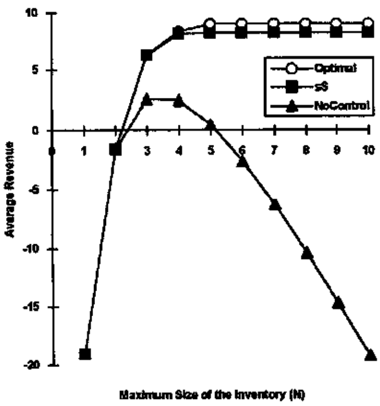

Figure 2 - Average revenue in terms of maximum size of the inventory

Based on the input values shown in Table 1 and an inventory size of 6, the following comparisons can be made:

average revenue using the optimal policy:	9.033
average revenue using the optimal (s, S) pnlicy :	8.247 (8.7 % less than the optimal policy)
average revenue using no control:	-2.621 (129% less than the optimal policy)

4. Conclusions

The models discussed in [Hwang and Koh, 1992] and [Gopalan and Kannan, 1994] have been extended using a Markov Decision approach. The present paper considered inspection, rework and in-process inventory. This paper shows that controlling the in-process inventory is essential whenever failures of manufacturing machines, in a transfer line system, must be considered.

The model discussed has been implemented in C++. A first prototype of a class structure constructed to deal with Markov and Semi-Markov Decision Processes has been used.

5. Bibliography

CARVALHO, S.V.; RODRIGUES, R.C.M.; MORAIS, P.R. Optimization of inventory between machines in a production line. *Proceedings of the 25th Brazilian Symposium on Operations Research*, Campinas, SP, Brazil, November 16-19, 1993, pp. 623-627. Published in Portuguese.

CARVALHO, S.V.; RODRIGUES, R.C.M.; VIJAYKUMAR, N.L.; MORAIS, P.R. Cnntrol of in-process inventory in a serial manufacturing system with inspection and rework. *Proceedings of the International Conference on Industrial Logistics*, Ouro Preto, MG, Brazil, December 11-14, 1995.

GOPALAN, M.N.; KANNAN, S. Expected duration analysis of a two-stage transfer-line production system subject to inspection and rework. *Journal of the Operational Research Society*, Vol. 45, No. 7, pp. 797-805, 1994.

HWANG, H.; KOH, S.-G. Optimal control of work-in-process with dual limit switches in two-stage transfer line. *European Journal of Operational Research*, Vol. 63, pp. 66-75, 1992.

TIJMS, H.C. *Stochastic modeling and analysis: A computational approach*. Wiley, New York, 1986.