Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 52-13 (2019) 241–246

Optimal Control of Production and Maintenance Operations in Smart Custom Manufacturing Systems with Multiple Machines

Elluri Vignan Gupta*, D. G. Mogale*, Manoj Kumar Tiwari*

**Deparment of Industrial and Systems Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, India, 721 302, (e-mail: vignangupta@gmail.com) (e-mail: dgmogle@gmail.com) (e-mail: mkt09@hotmail.com)*

Abstract: Enterprises equipped with IoT (Internet of Things) are the new generation of manufacturing industry. There is a need for new optimization models which incorporate the advantages of IoT. In this paper, a new mathematical model and heuristic algorithm are developed to minimize the total cost in a multiple machine environment which enables the industries to take economically better decisions and effectively use their resources. A heuristic algorithm is developed for identical machines which process with the same tool. A system in which jobs with stochastic workloads arrive randomly and upon arrival, their workload is facilitated by IoT. The proposed algorithm determines the assignment of workload to the machine and processing speed. The algorithm works for both online and offline frameworks.

Keywords: Online Algorithm, Machine Scheduling, Multi-machine, Internet of Things

1. INTRODUCTION

In the latest era of industrial transformation i.e., Industry 4.0, IoT is the crucial aspect and industries are endeavouring new technology which is compatible with IoT. Industries equipped with IoT will increase consistently (Sutherland et al. 2016). IoT enables universal manufacturing resource availability by integrating and connecting physical assets into an information network (Gao et al. 2015). All the physical entities work jointly to increase the productivity and efficiency of the system (Takenaka et al. 2016). Hence, there is a need for new optimization models and algorithms that include the advantages and additional information made available by IoT (Xu et al. 2019).

In this paper, a manufacturing system with N machines is considered. Every machine is a comprehensive model of a machine tool employed for manufacturing operations that can perform a wide range of subtractive and additive manufacturing operations. The machine is equipped with IoT, it can communicate its internal state (e.g., tool condition) and can gather status of various IoT-enabled jobs across the enterprise (Xu et al. 2019). This work can be extended for developing new algorithms for large-scale IoT based smart custom manufacturing systems.

The rest of the paper is organized as follows, Section 2 gives an overview of the related literature. Section 3 describes the problem statement. Mathematical formulation of the problem is given in Section 4. We develop a new heuristic algorithm to solve the model in Section 5. Section 6 discusses the numerical results and Section 7 provides concluding remarks and future work.

2. LITERATURE REVIEW

Literature related to manufacturing control has been discussed for decades. Every time there is a technology advancement, new literature is developed. The workload controlling problem has been addressed by Martinelli (2007), Srivatsan and Dallery (1998), Akella and Kumar (1986) and Sharifnia (1988) considering continuous workload for minimization of holding cost. The problem of discrete workload to minimize the processing cost was discussed, but the workload was revealed at the end of processing (Song 2006; Cheng et al. 2011; Pang 2015; Feng and Xiao 2002; Boukas et al. 1995; and Conrad and McClamroch 1987). Furthermore, Hall et al. (1997) and Chekuri et al. (2001) minimized the average completion time with the consideration of deterministic arrival of workload but does not include aspects like processing speed, a machine with the degrading tool and machine maintenance. In order to minimize the makespan, Gans and Van Ryzin (1997) controlled the processing speed without consideration of tool degradation. Sevastianov and Woeginger (1998) and Hall (1998) tackled the offline problem of machine scheduling to minimize makespan where the workload revealed deterministically at the beginning. However, job arrival is not discussed in these two articles. Xu et al. (2019) have taken into account the discrete part manufacturing, machine with degrading tools and degradation while optimizing the completion time. In literature, many authors have not considered the following features simultaneously in their studies: 1) Discrete part manufacturing, 2) Multiple machine system, 3) Each machine with degrading tool and degradation as a function processing speed, 4) Jobs with workload revealed

10.1016/j.ifacol.2019.11.148

upon arrival and 5) Objective of minimizing total cost. Here arrival need not be at the machine (Xu et al. 2019).

3. PROBLEM DESCRIPTION

In enterprises equipped with IoT, job arrives randomly and arbitrarily. Upon arrival, their process requirements i.e., workloads are revealed (known deterministically). Then a process plan is developed by interacting with the cloud database. This plan is a set of sub-tasks to be performed by machine and each subtask is characterised by amount of workload. Also each machine interacts with the tool and gets updates regarding the remaining tool life and decides whether it can perform one more sub-task or not.

A system with N smart manufacturing machines and a degrading tool is considered where tool replacement takes a significant amount of time. A list of sub-tasks $L = (w_1, w_2, \ldots, w_{n(L)})$ are to be performed using the machines. We need to determine on which machine w_i should be processed and its processing speed s_i. Also, the epoch when the tool should be replaced on each machine. The minimization of makespan considered in Xu et al. (2019) cannot be used here, as the objective function continuously decreases with an increase in the number of machines, which is practically not correct. Thus, here cost plays an important role in decision making. With the increase in the number of machines, the setup cost comes into play, and with makespan decreasing the holding cost decreases. The processing cost and tool cost have appeared in a single machine system. Hence, the new objective function is the minimization of total cost.

Z = Min [setup cost + processing cost + tool cost + holding cost]

If we consider a single machine system, the setup cost is constant and the holding cost is also constant as there is a fixed number of workload for the machine. Hence, it gets transformed into a makespan minimization problem described in Xu et al. (2019). Therefore, the current work is an extension of Xu et al. (2019) work.

4. MODEL FORMULATION

In the objective function, consider the setup cost which is dependent on the machine, so for a machine, it's fixed. The processing cost depends on the amount of time we process on a machine. If we are using m machines and each machine is assigned a list of workloads say L_i then (Xu et al. 2019),

$$\text{Processing time} = \sum_{i=1}^{m} \sum_{j=1}^{n(L_i)} \frac{w_{ij}}{s_{ij}} \tag{1}$$

After a job is processed the tool level ($h_{i,j}$) gets updated as follows (Xu et al. 2019),

$$h_{i,j+1} = h^{\star} \times x_{i,j+1} + \left(1 - x_{i,j+1}\right) \times \left[h_{i,j} - \frac{D(s_{ij})}{s_{ij}} \times w_{ij}\right] \tag{2}$$

Where $h^{\star}$ is tool level of a new tool, x_{ij} is a binary variable defined as follows,

$$x_{ij} = \begin{cases} 0, & no\ tool\ replacement\ before\ processing\ w_{ij} \\ 1, & tool\ is\ replaced\ before\ processing\ w_{ij} \end{cases}$$

$D(s)$ is the tool degrading rate and is defined as follows,

$$D(s) = \frac{s^v}{C} \times h^{\star} \tag{3}$$

Where s is the processing speed, C, v are constants from Taylor's tool life equation (Xu et al. 2019).

$$s^v T = C,\ v > 1 \tag{4}$$

So in the objective function, tool cost can be defined as follows,

$$\text{Tool cost} = \sum_{i=1}^{m} \sum_{j=1}^{n(L_i)} x_{ij} \times R_i\ , \tag{5}$$

We assume that a tool cannot be replaced while processing is going on,

$$h_{i,j} - \frac{D(s_{ij})}{s_{ij}} \times w_{ij} > 0 \tag{6}$$

Thus, the two conditions on tool level make sure that we have sufficient tool level to process current job and next job (Xu et al. 2019).
Holding cost depends on the inventory level of the system and as defined in Feng and Yan (2000),

$$\text{Holding cost} = E\left[\int_0^T I(t) \times c(t)dt\right] \tag{7}$$

$I(t)$ = inventory level at time t
$c(t)$ = holding cost per unit at time t

If a significant interval is considered for reviewing the inventory (say every hour), the integral can be replaced with a large operator (Σ). Let us assume that holding cost per unit is constant irrespective of the time, so the modified expression is as follows,

$$\text{Holding cost} = \sum_{t=0}^{T} I(t) \times C \tag{8}$$

Now for a workload w_i, it has arrived at $t = 0$, and it got processed during the interval $(t', t'+1)$, it will be present in the inventory from $t=0$ to $t = t'$. So holding cost can be expressed as the product of time taken for a workload to start getting processed after it has arrived and the holding cost per unit. Hence if we consider w_i,

$$\text{Holding cost of } w_i = \left(\sum_{j=1}^{i-1} \frac{w_j}{s_j}\right) \times C^* \tag{9}$$

If we modify the holding cost per unit into holding cost per unit time per unit work load C^* is replaced with $w_i \times H$. Here we assume that after a job is processed it will be shipped immediately.

Notations used:

Parameters:

m	number of machines under use
L	list of workloads to be processed
n(L)	number of workloads in list L

w_{ij} j[th] workload processed on machine i

F_i setup cost of machine i

P_i processing cost on machine i per unit time

R_i tool replacement cost of machine i

H holding cost per unit time per unit workload

$h_{i,j}$ remaining tool level on machine i before processing job j of machine i

$h^\star$ tool level of an new tool

$D(s)$ tool degrading rate for processing speed, determined using Taylor's tool life equation

Decision variables:

L_i list of workloads assigned to machine *i*

x_{ij} decision variable for tool replacement on machine i before processing job j of machine i

s_{ij} processing speed of j[th] job on machine i

Objective function:

Minimize,

$$\sum_{i=1}^{m}\sum_{j=1}^{n(L_i)}\left[F_i + \frac{w_{ij}}{s_{ij}} \times P_i + x_{ij} \times R_i + \left[\sum_{k=1}^{j-1}\frac{w_{ik}}{s_{ik}}\right] \times w_{ij} \times H\right] \quad (10)$$

Subject to:

$$h_{i,j+1} = h^\star \times x_{i,j+1} + \left(1 - x_{i,j+1}\right) \times \left[h_{i,j} - \frac{D(s_{ij})}{s_{ij}} \times w_{ij}\right] \quad \forall\, i,j \quad (11)$$

$$h_{i,j} - \frac{D(s_{ij})}{s_{ij}} \times w_{ij} > 0 \quad \forall\, i,j \quad (12)$$

$$\textstyle\sum_{i=1}^{m} n(L_i) = n(L) \quad (13)$$

$$x_{ij} = \{0,1\} \quad \forall\, i,j \quad (14)$$

$$s_{ij} > 0, \quad \forall\, i,j \quad (15)$$

Equation (11) updates the remaining tool life after processing the current task. Equation (12) makes sure that there is sufficient tool life to process current task and Equation (13) ensures that each task is assigned to a machine. Equation (14) states that x_{ij} is a binary variable and Equation (15) is a non-negative constraint for s_{ij}.

5. SOLUTION APPROACH

To solve the above formulation, first, fix the number of machines and then we should assign the workloads to machines and solve for each machine.

We assume that all the machines are identical and process with the same tool. So setup cost and per unit processing cost, tool cost, holding cost are assumed to be the same for each machine. This assumption makes the problem less complex so that the novelty of the problem can be understood easily.

In the objective function, there are four parts.

Z = Min [setup cost + processing cost + tool cost + holding cost]

Minimizing each part minimizes the whole cost, hence we can find an optimal solution.

Setup cost depends on a number of machines. To minimize the processing, we should minimize the time taken to process all the jobs. In case of tool cost, we should efficiently use each and every tool. For holding cost, as it depends on the time taken for a workload to get processed i.e., the time taken from the arrival of a job to till it gets processed. Thus, to minimize the holding cost we should minimize makespan.

For minimizing the processing and tool cost, we use **fixed buffer approach**. Fixed buffer approach is an online algorithm developed by Xu et al. (2019), which minimizes the makespan of one individual machine with an optimal number of tool replacements.

Consider the setup and holding cost, setup cost increases as the number of machines increase. Holding cost decreases as the number of machines increases since the makespan of the total workload decreases. Hence, there is a trade-off between setup cost and holding cost and we can find a middle ground where we can find our optimal solution with minimum cost.

In the formulation, we basically assign workload for each machine and then pack workload for each tool. But we can also do that the other way round, first we pack workload into minimum possible buffers and then assign buffers to machines.

$$L = (w_1, w_2, \ldots, w_{n(L)}) \rightarrow L = (b_1, b_2, \ldots.., b_k)$$

Where b_i is a buffer with maximum capacity of W. for converting the list into buffers we use the following algorithm.

Fixed buffer approach (Xu et al. 2019):

l = current buffer level

W = maximum buffer level or buffer capacity

For i = 1:n(L) do

If $l + w_i \le W$ *then put* w_i *in buffer and* $l \leftarrow l + w_i$;

Else if $l + w_i > W$ && $w_i \le W$ *then*

Put w_i *in a new buffer*

Else

Allot one complete buffer to w_i;

End if

If i = n(L) then consider the current level l as a buffer;

End if

To assign the buffers holding cost plays an important role as rest of the parts in the objective function are fixed. To minimize the total processing time we should minimize the number of buffers assigned to each machine, the optimal assignment for this will be assign equal number of buffers to each machine

5.1 Algorithm

For m = 1:N do

1. Convert the given list of workloads into a list of buffers with a maximum buffer capacity W.
2. Break the list (*L*) of buffers into 'm' approximately equal lists (L_1 , L_2, ... , L_m).
3. For each list L_i find the processing time, holding time of each buffer and calculate the total cost.
4. If the total cost of m is less than the total cost (m-1), continue to next m. Otherwise, (m-1) is the optimal number of machines to use to minimize the total cost.

Pseudo code of the heuristic algorithm is presented below.

For m = 1:N do

Split(L, m) = (L_1 , L_2, ... , L_m)

Processing time = $\sum_{i=1}^{m} \sum_{j=1}^{n(L_i)} L_i[j]$

Processing cost = processing time x P

For each L_i :

MS_i = Makespan = $\sum_{j=1}^{n(L_i)} [L_i[j] + \tau]$, τ *is tool replacement time*

Find holding time of $L_i[j]$ = $\sum_{k=i}^{j-1} L_i[k]\ \forall i,j$

Holding cost = holding time x H

Tool cost = n(L) x R

Setup cost = F x m

Makespan of total workload = Max{ MS_1, MS_2, ... ,MS_m}

Total cost = setup cost + tool cost + processing cost + holding cost

If total cost[m-1] > total cost[m]:

Continue;

Else:

Break;

6. NUMERICAL RESULTS AND DISCUSSION

In this section, we analyzed the numerical results obtained. We have simulated a system with 16 smart machines in which workloads are generated using uniform distribution with an upper bound, the buffer capacity is around $1.5w_b$, where w_b is the optimal tool level determined using Taylors' tool life constants (Xu et al. 2019).

$$w_b = \left((v-1)\tau C^{\frac{1}{v-1}}\right)^{\frac{v-1}{v}} \quad (16)$$

Where v, C are constants in Taylor's tool life equation and τ is the tool replacement time.
The trade-off between holding cost, setup cost and the variation of the total cost is shown in Figure 1.

So we can clearly see that we have a middle ground where we can find an optimal solution. Similarly, we have many parameters like a number of workloads (in this section 'n'), upper bound etc. Now we show how the total cost varies with different values of *n*, Figure 2.

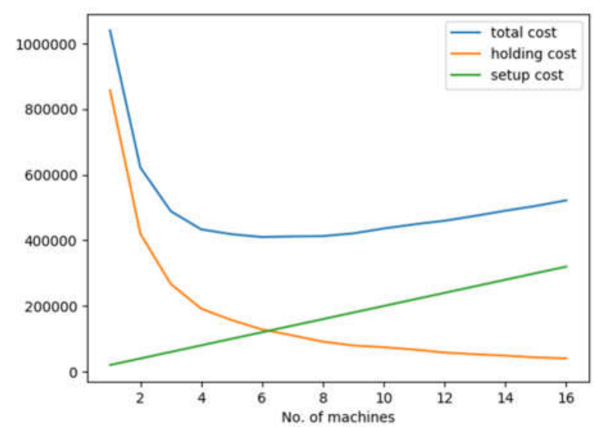

Figure 1: Plot showing the variation of different types of costs

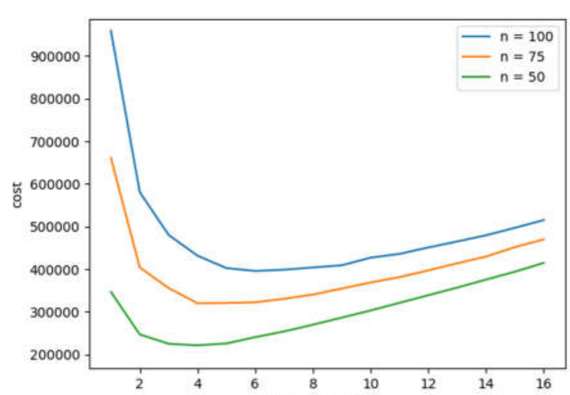

Figure 2: Plot showing variation of total cost with different 'n'

As we can see that as the n value is increasing the optimal solution i.e., the minimum point is moving towards the right, which means we should use more number of machines with more workload. In all the cases using one machine is very costly and there is a drastic decrease in the cost, this is due to fact that the makespan is almost reduced to half if we use 2 machines instead of 1. Similarly, the makespan reduces to one third if we use 3 machines instead of 1. If we go for higher values of n, the plots are as shown in Figure 3.

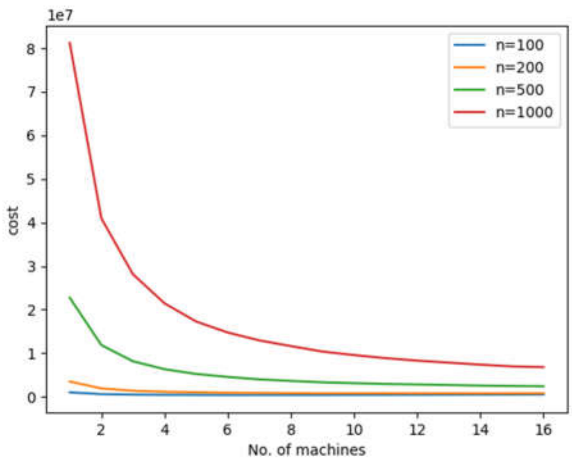

Figure 3: Plot showing variation of total cost with large 'n'

With an increase in the number of workloads, we should use as many machines as possible so that we could reduce our cost drastically. The variation of total cost with different upper bounds (U) is shown in Figure 4.

This plot can be understood as different buffer capacities for a given upper bound. So with an increase in buffer capacity, we can put a lot of workload into a single buffer which decreases the makespan as there is less number of tool changes resulting in lower tool cost and lower holding cost as tool replacement consumes a significant amount of time.

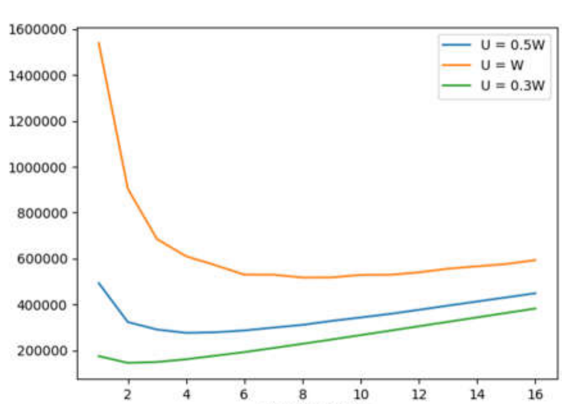

Figure 4: Plot showing variation of total cost with U values

7. CONCLUSION AND FUTURE WORK

In this paper, we consider a custom-manufacturing framework with IoT. The main advantage of IoT is that we can precisely determine the workloads given their physical characteristics and requirements. Also, we get the information regarding the distribution of workloads. The developed new model and heuristic algorithm minimizes total cost unlike a single type of cost as seen in previous literature. We have simulated a very simple system which can address our requirements and we got promising results. This algorithm can be extended for more complex scenarios with holding cost having a discount factor and fixed cost is a function of a number of machines. We have considered identical machines with the same type of degrading tool, so there are several extension for this research which will be useful in developing large-scale IoT-based smart manufacturing systems with customized production.

REFERENCES

Akella, R., & Kumar, P. (1986). Optimal control of production rate in a failure prone manufacturing system. *IEEE Transactions on Automatic control*, *31*(2), 116-126.

Boukas, E. K., Zhang, Q., & Yin, G. (1995). Robust production and maintenance planning in stochastic manufacturing systems. *IEEE Transactions on Automatic Control*, *40*(6), 1098-1102.

Chekuri, C., Motwani, R., Natarajan, B., & Stein, C. (2001). Approximation techniques for average completion time scheduling. *SIAM Journal on Computing*, *31*(1), 146-166.

Cheng, T. C. E., Gao, C., & Shen, H. (2011). Production and inventory rationing in a make-to-stock system with a failure-prone machine and lost sales. *IEEE transactions on automatic control*, *56*(5), 1176-1180.

Coffman Jr, E. G., So, K., Hofri, M., & Yao, A. C. (1980). A stochastic model of bin-packing. *Information and Control*, *44*(2), 105-115.

Conrad, C., & McClamroch, N. (1987). The drilling problem: a stochastic modeling and control example in manufacturing. *IEEE Transactions on automatic control*, *32*(11), 947-958.

Feng, Y., & Xiao, B. (2002). Optimal threshold control in discrete failure-prone manufacturing systems. *IEEE Transactions on Automatic Control*, *47*(7), 1167-1174.

Feng, Y., & Yan, H. (2000). Optimal production control in a discrete manufacturing system with unreliable machines and random demands. *IEEE Transactions on Automatic Control*, *45*(12), 2280-2296.

Gans, N., & Van Ryzin, G. (1997). Optimal control of a multiclass, flexible queueing system. *Operations Research*, *45*(5), 677-693.

Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., & Helu, M. (2015). Cloud-enabled prognosis for manufacturing. *CIRP annals*, *64*(2), 749-772.

Hall, L. A. (1998). Approximability of flow shop scheduling. *Mathematical Programming*, *82*(1-2), 175-190.

Hall, L. A., Schulz, A. S., Shmoys, D. B., & Wein, J. (1997). Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Mathematics of operations research*, *22*(3), 513-544.

Martinelli, F. (2007). Optimality of a two-threshold feedback control for a manufacturing system with a production dependent failure rate. *IEEE Transactions on Automatic Control*, *52*(10), 1937-1942.

Pang, Z. (2015). Optimal control of a single-product assemble-to-order system with multiple demand classes and backordering. *IEEE Transactions on Automatic Control*, *60*(2), 480-484.

Sevastianov, S. V., & Woeginger, G. J. (1998). Makespan minimization in open shops: A polynomial time approximation scheme. *Mathematical Programming*, *82*(1-2), 191-198.

Sharifnia, A. (1988). Production control of a manufacturing system with multiple machine states. *IEEE Transactions on Automatic Control*, *33*(7), 620-625.

Song, D. P. (2006). Optimal production and backordering policy in failure-prone manufacturing systems. *IEEE transactions on automatic control*, *51*(5), 906-911.

Srivatsan, N., & Dallery, Y. (1998). Partial characterization of optimal hedging point policies in unreliable two-part-type manufacturing systems. *Operations Research*, *46*(1), 36-45.

Sutherland, J. W., Richter, J. S., Hutchins, M. J., Dornfeld, D., Dzombak, R., Mangold, J., & Friemann, F. (2016). The role of manufacturing in affecting the social dimension of sustainability. *CIRP Annals*, *65*(2), 689-712.

Takenaka, T., Yamamoto, Y., Fukuda, K., Kimura, A., & Ueda, K. (2016). Enhancing products and services using smart appliance networks. *CIRP Annals*, *65*(1), 397-400.

Xu, J., Tran, H. M., Gautam, N., & Bukkapatnam, S. T. (2019). Joint production and maintenance operations in smart custom-manufacturing systems. *IISE Transactions*, 1-16.