Generic Single Channel Multi Phase Waiting Line Model applied to Production Line for Productivity Improvement

Mr. Girish R. Naik¹, Dr. V.A.Raikar², Dr. Poornima.G. Naik³,
¹(Associate Professor, Production Department, KIT’s College of Engineering, Kolhapur, India.)
²(Principal, Government Engineering College, Karwar, India.)
³(Assistant Professor, Department of Computer Studies/CSIBER, Kolhapur, India.)

Abstract: A service mechanism in a mass production type of manufacturing consists of service facilities which in turn contain parallel service channels. The prime objective of analyzing production line is to understand and analyze line performance which enables manager to identify bottlenecks affecting performance, constrained resources and predict system serviced cost and system waiting cost. This enables the manager to compute the total system cost at various levels of service which ensures optimal allocation of existing resources. This paper focuses on Single Channel Multi-Phase production line. The authors have developed a mathematical model for single channel multi-phase production line which is implemented in Visual Basic (VB). The production line configuration information such as machine name, cycle time, machine hour rate, shift busy period, production rate is centralized and stored in an XML file which is parsed using DOM (Document Object Model) parser for VB. A tool is developed for designing of a production line. WIP accumulation between different machines and individual machine idles times are determined. Finally, optimum level of service is determined from system service cost and system waiting cost. The model is generic and can be applied to any single channel, multi-phase production line.

Keywords: Extensible Markup Language, Manufacturing, Production Line, Single Channel Multi Phase Queues, Work In Process.

I. Introduction

A production line is a line in which units move continuously through a sequence of stations. The main objective is to optimize line performance which can be attained by best allocation of task to an ordered sequence of stations subject to precedence constraints.

The main issues are associated with queuing, idle time among stations during the task achievement, WIP accumulation between machines which become obstacles to an efficient production line. Bottlenecks and congestion in queuing system has serious consequence. It has serious negative implication on both the system service cost and system waiting cost. The immediate consequence of the increase in idle time and WIP is corresponding increase in the operational cost. Further the failure to comply with the customer orders results in customer dissatisfaction and eventually loss of market share.

1.1 Problem Definition

In flow production type of manufacturing system in the production line, whenever existing demand exceeds capacity of the service facility, the items arriving cannot receive immediate service due to busy servers, which results in accumulation of WIP (Work In Process) between machines. Due to varying levels of production, the issues such as idle time of machines and WIP accumulation arise which hamper line performance. The main objective of generic model development is minimization of system service cost with respect to accumulated WIP and idle time of machines.

1.2 Objectives of the Work

- To reduce manufacturing lead time.
- To analyze idle time, WIP accumulation between machines and identify bottleneck resource.
- To analyze delays in production system with variable service times.
- To determine the level of service which minimizes total expected cost of service and the expected cost of waiting for that service.
- To analyze WIP in line for productivity improvement.
- To study system utilization and cost incurred, strike economic balance with minimum waste in system.

1.3 Assumptions made in Model Development

1. Mass production type of manufacturing system is assumed.
2. All jobs visit machines or work centers in the same sequence complying standard routing.
3. The products are processed in a sequential order of machines in the production line.
4. Machine working conditions are reflected in cycle time of products.
5. All the parameters such as system service level, system service cost, and system waiting cost are computed for one working shift.
6. Cycle time is modified to take account of machine age using linear extrapolation.
7. Demand forecasting is the basis for the production level requirement.
8. Arrival rate of products at the first machine is assumed to be deterministic and finite.
9. First come first serve queue discipline is assumed.

II. Literature Survey and Prior Art

A lot of research work has been carried out to describe and solve more realistic generalized assembly line balancing problems. An exhaustive survey was carried out by Becker and Scholl on problems and methods in generalized assembly line balancing [1]. There is a substantial study on single channel waiting line system using various statistical approaches such as poisson arrivals and exponential service time. However, they are based on stochastic mathematical models [2]. A lot of literature exists on PLBP (Production Line Balancing Problem) where a technical constraint was carried out to minimize queuing problem and regulate the workers by applying hybrid models with an emphasis to reduce queuing and idle time. A new approach on traditional assembly line balancing problem is presented using 0-1 integer programming model to establish balance of assembly with minimum number of station and resources [3]. Andreas C. Nearchou has devised a novel method based on particle swarm optimization for simple assembly line balancing problems. Two criteria are simultaneously considered for optimization to minimize cycle time and maximize workload smoothing. [4]. In spite of lot of literature available on basic assembly line balancing, there remains a gap between requirements of real configuration problems and status of research. Most of the work focuses on just single or only few practical extensions at a time. Real world assembly system requires a lot of these extensions to be considered simultaneously. Nils Boysen et.al. have proposed a criteria for model selection according to characteristic practical settings and have highlighted relevant model extensions to reflect real world problems [5].

Rajendran and Alicke [6] carried out a study on the problem of dispatching in flow shops with bottleneck machines. Here, the objective is to develop dispatching rules for scheduling by taking into account the presence of bottleneck machines on the basis of TOC methodology. The measures of performance are the minimization of total flow time of jobs, the minimization of the sum of earliness and tardiness of jobs and the minimization of total tardiness of jobs are considered separately. The results of the experimental investigation show that the proposed dispatching rules emerge to be superior to the conventional dispatching rules. Murty [7] presented an Indian case study where in application of TOC-DBR leads to significant improvement in throughput. The implementation is carried out only for connecting rod line and flywheel housing line of engine division of company. This has resulted in an increase of about 50% in throughput of about 50% in the throughput without any additional investment. Besides this, there is considerable improvement in work in process inventory as well as machine utilization.

Frazier and Reyes [8] applied the TOC-DBR concept to obtain substantial improvement in its production performance in one of the high-tech manufacturing industry. The primary measures are used to assess the production performance are percentage of work orders complicated in the month, month end raw material inventory implementation period it is observed that WIP inventory is decreased to one third of its previous level, raw material inventory value is decreased by 30% and percentage of on time completion of jobs are increased by more than 20%. Chakravorty [9] explained the result obtained by implementing the Drum-Buffer-Rope (DBR) in a V-plant operation. The annual sales and performance measure before, during and after implementation of DBR are compared. It was found that the annual sales are increased by 4% inventory of finished window unit decreased by 67% and percentage of the orders reduced by 63%.

III. Mathematical Model for Generic Production Line

Let $M_1, M_2, \ldots, M_n$ be $n$ machines in a multi-phase assembly. Let $CT_{M_1}, CT_{M_2}, \ldots, CT_{M_n}$ be cycle times of a product on machines $M_1, M_2, \ldots, M_n$, respectively. The following table depicts time duration at which different items enter and leave different machines.
Generic Single Channel Multi Phase Waiting Line Model applied to Production Line for…

<table>
<thead>
<tr>
<th>Item No.</th>
<th>M1</th>
<th>OUT</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td></td>
<td></td>
<td>IN</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>CT_{M1}</td>
<td>CT_{M1}+ CT_{M2}</td>
</tr>
<tr>
<td>2</td>
<td>CT_{M1}</td>
<td>2CT_{M1}</td>
<td>Max [2CT_{M1} , \Sigma_{i=1}^{2} CT_{M1} ]</td>
</tr>
</tbody>
</table>

Generalizing the above process to the n\textsuperscript{th} machine, the time at which the product enters n\textsuperscript{th} machine is given by,

\[ \text{Max} \left\{ \begin{array}{c}
\text{Max} \left[ \begin{array}{c}
\text{Max} \left[ \begin{array}{c}
2CT_{M1}, \Sigma_{i=1}^{2} CT_{M1}
\end{array}\right] , \Sigma_{i=1}^{3} CT_{M1} \\
\end{array}\right] , \Sigma_{i=1}^{4} CT_{M1} , \ldots , \Sigma_{i=1}^{n} CT_{Mi}
\end{array}\right\} \]

Let it be denoted by M.

and the time at which the product leaves n\textsuperscript{th} machine is given by

\[ M + CT_{Mn} \]

where CT_{Mn} is the cycle time of the product on the n\textsuperscript{th} machine.

The first product enters the first machine M\textsubscript{1} at time t=0 and leaves M\textsubscript{1} at time CT_{M1} where CT_{M1} is the cycle time of the product on machine M\textsubscript{1}. At this point of time M\textsubscript{2} is free. Hence the product is immediately processed on machine M\textsubscript{2}, where it enters M\textsubscript{2} at time CT_{M1} and leaves M\textsubscript{2} at a time [CT_{M1} + CT_{M2}]. Continuing in this manner, the first product enters n\textsuperscript{th} machine at a time

\[ \sum_{i=1}^{n-1} CT_{Mi} \]

and leaves n\textsuperscript{th} machine at a time

\[ \sum_{i=1}^{n-1} CT_{Mi} + CT_{Mn} = \sum_{i=1}^{n} CT_{Mi} \]

Hence the total time spent by the first product in an assembly line is

\[ \sum_{i=1}^{n} CT_{Mi} \]

However, due to the differing cycle times of a product on different machines the situation is different for second and subsequent products. The second item enters the first machine at a time CT_{M1} and leaves it at a time 2CT_{M1} at which it is ready to be processed by machine M\textsubscript{2}. However, M\textsubscript{2} becomes free at a time CT_{M1} + CT_{M2} after processing first item. Hence the second item can enter M\textsubscript{2} at the time given by,
Max \[2CT_{M1}, \sum_{i=1}^{a} CT_{Mi}\] 

(1)

The second item leaves \(M_2\) at a time given by

Max \[2CT_{M1}, \sum_{i=1}^{a} CT_{Mi}\] + CT_{M1}

With the same argument the in and out times of second product on machine \(M_3\) are given by

Max \(\{\text{Max} \left[2CT_{M1}, \sum_{i=1}^{a} CT_{Mi}\right], \sum_{i=1}^{b} CT_{Mi}\}\}

Generalizing the above procedure for the \(k^{th}\) item on the \(i^{th}\) machine, the waiting time is given by

Max \(\{\text{Max} \left[\text{Max} \left[\text{Max} \left[\text{Max} \left[2CT_{M1}, \sum_{i=1}^{a} CT_{Mi}\right],\sum_{i=1}^{b} CT_{Mi}\right],\sum_{i=1}^{c} CT_{Mi}\right],\ldots,\sum_{i=1}^{k} CT_{Mi}\right]\right]\]\}

We consider the following special cases.

Case 1 : CT_{Mi} < CT_{M2} < CT_{M3} < \ldots \ldots < CT_{Mi}

In this case each machine is continuously busy processing the products. Hence the idle time of any machine is zero. However, it results in WIP accumulation between machines. In this case, in equ (1), second term will be maximum as shown below and hence selected in the expression.

CT_{M1} < CT_{M2}

Adding to CT_{M1} both the sides, we get

2CT_{M1} < CT_{M1} + CT_{M2}

2CT_{M1} < \sum_{i=1}^{a} CT_{Mi}

Hence, Max \(\left[2CT_{M1}, \sum_{i=1}^{a} CT_{Mi}\right] = \sum_{i=1}^{a} CT_{Mi}\)

Applying the same logic to next station, we get

Max \(\{\text{Max} \left[2CT_{M1}, \sum_{i=1}^{a} CT_{Mi}\right], \sum_{i=1}^{b} CT_{Mi}\}\} = \sum_{i=1}^{b} CT_{Mi}

Calculation of accumulated system WIP per unit time. Waiting time of items between machines \(M_1\) and \(M_2\) is given by

\(\sum_{i=1}^{a} CT_{Mi} - 2CT_{M1}\)

and between machines \(M_2\) and \(M_3\) is given by

\(\sum_{i=1}^{a} CT_{Mi} - \{\sum_{i=1}^{a} CT_{Mi} - CT_{M2}\}\)

Continuing in this manner the system WIP is given by

\(\{\sum_{i=1}^{a} CT_{Mi} - 2CT_{M1}\} + \{\sum_{i=1}^{b} CT_{Mi} - (\sum_{i=1}^{a} CT_{Mi} + CT_{M2})\}\)

+ \{\sum_{i=1}^{a} CT_{Mi} - (\sum_{i=1}^{b} CT_{Mi} + CT_{M3})\} + \ldots \ldots \ldots \)
Generic Single Channel Multi Phase Waiting Line Model applied to Production Line for...

\[ \text{Total WIP per unit time is equal to } \text{Total}_{wip} = \text{CT}_{Mk} - \text{CT}_{M1} \]

which depends only on cycle times of first and kth machines and is independent of cycle time of intermediate machines.

**Case 2 :** \( \text{CT}_{M1} > \text{CT}_{M2} > \text{CT}_{M3} > \ldots > \text{CT}_{Mk} \)

In this case each machine, except the first one is idle waiting for the product. Hence the WIP accumulation is zero. In this case in equ(1), first term will be maximum as shown below:

\[ \text{CT}_{M1} > \text{CT}_{M2} \]

Adding to \( \text{CT}_{M1} \) both the sides, we get

\[ 2\text{CT}_{M1} > \text{CT}_{M1} + \text{CT}_{M2} \]

\[ 2\text{CT}_{M1} > \sum_{i=1}^{2} \text{CT}_{Mi} \]

Hence, \( \text{Max} \left[ 2\text{CT}_{M1}, \sum_{i=1}^{2} \text{CT}_{Mi} \right] = 2\text{CT}_{M1} \)

Applying the same logic to next station, we get,

\[ \text{Max} \left\{ \text{Max} \left[ 2\text{CT}_{M1}, \sum_{i=1}^{2} \text{CT}_{Mi} \right], \sum_{i=1}^{3} \text{CT}_{Mi} \right\} \]

\[ = \text{Max} \left\{ 2\text{CT}_{M1}, \sum_{i=1}^{3} \text{CT}_{Mi} \right\} \]

\[ = 2\text{CT}_{M1} \]

**Calculation of System Idle Time**

The idle time of the machine \( M_i \) is given by,

\[ 2\text{CT}_{M1} - \sum_{i=1}^{2} \text{CT}_{Mi} \]

\[ = 2\text{CT}_{M1} - \text{CT}_{M1} - \text{CT}_{M2} \]

\[ = \text{CT}_{M1} - \text{CT}_{M2} \]

The idle of the machine \( M_2 \) is given by,

\[ = \sum_{i=1}^{2} \text{CT}_{Mi} + \text{CT}_{M2} - \sum_{i=1}^{3} \text{CT}_{Mi} \]

\[ = \text{CT}_{M1} + 2\text{CT}_{M2} - \text{CT}_{M1} - \text{CT}_{M2} - \text{CT}_{M3} \]

\[ = \text{CT}_{M2} - \text{CT}_{M3} \]

Hence the system idle time is given by,

\[ \text{Total System Idle Time} = \text{CT}_{M1} - \text{CT}_{Mk} \]

Which again depends on first and last machines only. For all other variations, the WIP accumulation and system idle time are dependent on all the intermediate stations

**IV. Results And discussion**

The model developed above is implemented in VB with MS-Access as backend. The structure of the database employed for this purpose is depicted in Fig 1.
The production line configuration information is stored in an XML file. A sample XML file is shown below. Figure 2 shows a GUI screen for analysis of production line. The production line can be designed by employing a simple drag and drop operation as shown in Fig 3(a). Fig 3(b) presents a GUI for the modification of machine configuration information stored in an XML file. The XML file is automatically generated based on the parameters entered during the design phase. The structure of the XML file storing the machine configuration information is shown in Figure 4. The corresponding Document Type Definition (DTD) for an XML document is shown in Figure 5.
Generic Single Channel Multi Phase Waiting Line Model applied to Production Line for...

Fig. 3 (a)-(b) GUI GUI for Design of Production Line

```
<AssemblyLine>
  <WorkStations>
    <WorkStation>
      <stage>1</stage>
      <MachineName>TurretLathe1</MachineName>
      <CycleTime>9</CycleTime>
      <MachineHourRate>300</MachineHourRate>
      <LabourHourRate>100</LabourHourRate>
      <ShiftBusyPeriod>6</ShiftBusyPeriod>
      <ProductionRate>6.69</ProductionRate>
    </WorkStation>
    <WorkStation>
      <stage>2</stage>
      <MachineName>TurretLathe2</MachineName>
      <CycleTime>9</CycleTime>
      <MachineHourRate>300</MachineHourRate>
      <LabourHourRate>100</LabourHourRate>
      <ShiftBusyPeriod>5.5</ShiftBusyPeriod>
      <ProductionRate>6.69</ProductionRate>
    </WorkStation>
    ...
  </WorkStations>
</AssemblyLine>
```

Fig. 4 XML File Storing Machine Configuration Information
Fig. 5 Document Type Definition for an XML File Storing Machine Configuration Information.

The XML file is parsed using Microsoft XML parser available as a dynamic link library, msxml6.dll. The VB code used for parsing is shown in Appendix A.

Fig. 6 and 7 show computation of WIP accumulation between various machines and machine idle times, respectively. Due to the varying cycle time of different machines, there is a WIP accumulation between different machines and also there is some idle time at some work stations. The following two scenarios arise:

**Scenario 1**: Idle Time Minimization.

**Scenario 2**: WIP Minimization.

The probability of a machine remaining idle is inversely proportional to its utilization. If the cycle time of (n+1)th machine is greater than the cycle time of nth machine, idle time of (n+1)th machine will increase. On the contrary, if the cycle time of (n+1)th machine is less than the cycle time of nth machine, WIP accumulation between nth and (n+1)th machines will increase. Hence it is observed that there is a tradeoff between minimization of WIP and minimization of idle time of machines in a production line. However, the total system cost which is the sum of the total waiting cost and the total service cost is found to be minimum for a certain level of service which is the economic level of WIP for the system under consideration.
Computation of system service cost and system idle cost is depicted in Fig. 8. For the present system under consideration, the total system cost is found to be minimum at level of service 30 as shown in Fig. 9. Fig. 10 depicts the two scenarios discussed above for a hypothetical production Line.

**Fig. 7** Computation of Machine Idle Times

**Fig. 8** Computation of System Service Cost and System Waiting Cost.

**Fig. 9** Computation of Total System Cost
V. Conclusion and Scope for Future Work

In the present work, the authors have developed and implemented a generic model for a single channel multi-phase production line. A GUI is presented for the design of an assembly line which stores the machine configuration information in an XML file. A DTD for the XML file is presented. WIP accumulation between various machines and individual machine idle times are determined. Optimum level of service is determined from system service cost and the system waiting cost. The model is generic and can be applied to any single channel multi-phase production line.

Based upon the cycle time of each work station, the service rate is determined and if the machine is kept busy on the basis of these derived times, the line is perfectly balanced to zero WIP level. In future, by setting the production line with various levels of WIP and analysing the utilization of resources, the cost impact can be assessed as a measure towards productivity improvement. To account for constrained resources which results in bottleneck, provision for time buffer or capacity buffer can be considered to attain synchronized manufacturing. Our future work focuses on fuzzifying input parameters to implement a more realistic model.

References

Appendix A

Parsing of XML file using Microsoft XML Parser

```vbscript
Dim objXML As MSXML2.DOMDocument
Dim objElem As MSXML2.IXMLDOMElement
Dim objSub As MSXML2.IXMLDOMElement

If con.State = adStateClosed Then
  con.Open
  "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\waitingline.mdb;Persist Security Info=False"
End If

Set objXML = New MSXML2.DOMDocument
If Not objXML.Load(fname) Then
  Err.Raise objXML.parseError.errorCode, , objXML.parseError.reason
End If

Set objElem = objXML.selectSingleNode("WorkStations")
  ' iterate its sub-nodes
cnt = 0
  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem = objSub.selectSingleNode("MachineName")
    mname(cnt) = objElem.Text
    cnt = cnt + 1

    Set objElem = objSub.selectSingleNode("CycleTime")
    ct(cnt) = objElem.Text
    Next
    cnt = 0

  Next
  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem = objSub.selectSingleNode("MachineHourRate")
    mhr(cnt) = objElem.Text
    cnt = cnt + 1

    Set objElem = objSub.selectSingleNode("LabourHourRate")
    lr(cnt) = objElem.Text
    Next
    cnt = 0

    Set objElem =
    objSub.selectSingleNode("ShiftBusyPeriod")
    avg_busy_period(cnt) = objElem.Text
    Next
    cnt = 0

  Next
  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem =
    objSub.selectSingleNode("ProductionRate")
    prod(cnt) = objElem.Text
    Next
    cnt = 0

  Next
  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem =
    objSub.selectSingleNode("ShiftBusyPeriod")
    avg_busy_period(cnt) = objElem.Text
    Next
    cnt = 0

  Next
  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem =
    objSub.selectSingleNode("ProductionRate")
    prod(cnt) = objElem.Text
    Next
    cnt = 0

  Next

For Each objSub In objXML.selectNodes("//WorkStation")
  Set objElem =
  objSub.selectSingleNode("ShiftBusyPeriod")
  avg_busy_period(cnt) = objElem.Text
  Next
  cnt = 0

  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem =
    objSub.selectSingleNode("ProductionRate")
    prod(cnt) = objElem.Text
    Next
    cnt = 0

  Next
  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem =
    objSub.selectSingleNode("ProductionRate")
    prod(cnt) = objElem.Text
    Next
    cnt = 0

  Next
  For Each objSub In objXML.selectNodes("//WorkStation")
    Set objElem =
    objSub.selectSingleNode("ProductionRate")
    prod(cnt) = objElem.Text
    Next
    cnt = 0

  Next

nm = cnt 'No of machines
If (prod(1) > 1 And prod(1) < 10) Then
  llimit = 10
  ulimit = 70
  con.Execute "Delete from servicecost"
  For i = 10 To 70 Step 10
    con.Execute "INSERT INTO servicecost(LevellofService) VALUES(" & i & ")"
    Next
End If

If (prod(1) > 10 And prod(1) < 100) Then
  llimit = 100
  ulimit = 700
  con.Execute "Delete from servicecost"
  For i = 100 To 700 Step 100
    con.Execute "INSERT INTO servicecost(LevellofService) VALUES(" & i & ")"
    Next
End If

If (prod(1) > 100 And prod(1) < 1000) Then
  llimit = 1000
  ulimit = 7000
  con.Execute "Delete from servicecost"
  For i = 1000 To 7000 Step 1000
    con.Execute "INSERT INTO servicecost(LevellofService) VALUES(" & i & ")"
    Next
End If
```