Mathematical Modeling of the Two-Part Type Machine

Young Jae Jang
Massachusetts Institute of Technology
May 12, 2004

(Courtesy of Young Jang. Used with permission.)
Contents

- Introduction
- Previous work
- Issues on multiple-part-type line
- Two machine line
- Decomposition for type one and type two
- Heuristics
- Algorithm
- Comparisons with simulation
- Future work and summary
Introduction

- Multiple-Part-Type Processing Line
- Example of Two-Part-Type Processing Line
  - Supply Machines: \( M_{s1}, M_{s2} \)
  - Demand Machines: \( M_{d1}, M_{d2} \)
  - Processing Machine: \( M_i \)
  - Homogeneous Buffer: \( B_{i,j} \)
    - Size: \( N_{ij} \)
    - Number of Parts at \( B_{ij} \) at time \( t = n_{ij}(t) \)

- Priority Rule: Type one has a priority over type two
- Why supply and demand machines are needed?
Introduction - Parameters

- Discrete time Model
- Identical processing time
- One time unit = processing time for one part

\[ r_i = Pr [\alpha_i(t + 1) = 1 | \alpha_i(t) = 0] \]
\[ p_i = Pr [\alpha_i(t + 1) = 0 | \{\alpha_{i,1}(t) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t) < N_{i,1}\} \cup \{\alpha_{i,1}(t) = 1 \cap (n_{i-1,1}(t) = 0 \cup n_{i,1}(t) = N_{i,1}) \cap n_{i-1,2}(t) > 0 \cap n_{i,2}(t) < N_{i,2}\}] \]

for \( i = 1, \ldots, k \)
Introduction – Importance

- Most of processing machines these days are able to process several different part types
- Lines are usually processing more than one type of products
- In LCD or Semiconductor FAB multi-loop processing is common
- Better scheduling for multiple part type line
- GM Needs
- It will be a part of analysis of CPP of multiple part type line
Previous Work

  - Single Failure Mode
  - Too complicated
  - Lines are not very well converged

- **Diego Syrowicz**: MIT MS Thesis 1999
  - Multiple failure mode
  - Did not complete decomposition equations

- **Tulio Tolio**, 2003
  - Two machine two buffer building block, discrete time, multi-failure model
  - Too complicated, some equations are not clear
Approach

- **Observers**
  - Think that machine is working only single part type
  - Type2 guy thinks that machine is down, when the machine is working on part type one
Issues on Multiple-Part-Type Line – Idleness Failure

- Idleness failure: Failure while machine is starved or blocked
- There is no idleness failure in a single part type case
- Example
  1. $M_3$ down
  2. $n_{2,1} = N_{2,1}$, $n_{2,2} <= N_{2,2}$, $n_{s,1} > 0$, $n_{s,2} > 0$
  3. $M_2$ blocked for type 1, starts working type two
  4. $M_2$ goes to down, but $n_{2,1} = N_{2,1}$, $n_{s,1} > 0$
- Observer for part type one sees that $M_1$ goes to down when it is blocked for part type one!

![Diagram](image)
Issues on Multiple-Part-Type Line  
– Failure Mode Change

- Failure mode shift might be detected by an observer
- There is no failure mode change in a single part type line
- Example
  - $M_4$ down
  - $n_{3,1} = N_{3,1}$, $n_{3,2} < N_{3,2}$
  - $M_3$ is blocked for type one and start working type two part
  - $M_3$ is down while working on type two
  - $M_4$ is up and $n_{3,1} < N_{3,1}$
  - $M_3$ is still down
Two Machine Line Parameters

- Two machine line
  - Single-up, multiple-down, failure mode changing Markov chain
  - $\alpha(t)$ : machine states at time $t$
  - $\Upsilon$: Machine is up, $\Delta_i$: Machine is down at mode $i$

\[
\begin{align*}
    r^u_j &= Pr[\{\alpha^u(t+1) = \Upsilon^u\} \mid \{\alpha^u(t) = \Delta^u_j\}] \\
    r^d_k &= Pr[\{\alpha^d(t+1) = \Upsilon^d\} \mid \{\alpha^d(t) = \Delta^d_k\}] \\
    p^u_j &= Pr[\{\alpha^u(t+1) = \Delta^u_j\} \mid \{\alpha^u(t) = \Upsilon^u\} \cap \{n(t) < N\}] \\
    p^d_k &= Pr[\{\alpha^d(t+1) = \Delta^d_k\} \mid \{\alpha^d(t) = \Upsilon^d\} \cap \{n(t) > 0\}] 
\end{align*}
\]
Two Machine Line – Markov Model

- Two machine line
  - Failure mode change and Idleness failure
  - \( z^*_{ij} \): Failure mode change parameters
  - \( q^*_{l,ij} \): Idleness failure parameters
    \((^* = u \text{ or } d)\)
  - **These parameters are zero**

\[
q^u_j = Pr[\{\alpha^u(t+1) = \Delta^u_j\} | \{\alpha^u(t) = Y^u\} \cap \{n(t) = N\}]
\]
\[
q^d_k = Pr[\{\alpha^d(t+1) = \Delta^d_k\} | \{\alpha^d(t) = Y^d\} \cap \{n(t) = 0\}]
\]
\[
z^u_{j,j'} = Pr[\{\alpha^u(t+1) = \Delta^u_{j'}\} | \{\alpha^u(t) = \Delta^u_j\}]
\]
\[
z^d_{k,k'} = Pr[\{\alpha^d(t+1) = \Delta^d_{k'}\} | \{\alpha^d(t) = \Delta^d_k\}]
\]
\[
Q^u = \sum_{j=1}^{J} q^u_j, \quad J = \text{Total number of failure mode of } M^u
\]
\[
Q^d = \sum_{l=1}^{L} q^u_l, \quad L = \text{Total number of failure mode of } M^u
\]
One Machine Markov Model

- Single-up, Multiple-down, Failure Mode Changing Markov Chain
- Example of 3 down modes Markov Chain
Two Machine Line – Efficiency of the line

Single part type case (without idleness failure), efficiency is

\[
E^u = \Pr[\{\alpha^u(t) = \Upsilon^u\} \cap \{n(t) < N\}]
\]

\[
E^d = \Pr[\{\alpha^d(t) = \Upsilon^d\} \cap \{n(t) > 0\}]
\]

\[
E^u = E^d
\]

However, with idleness failure,

\[
\Pr[\{\alpha^u(t) = \Upsilon^u\} \cap \{n(t) < N\}] \neq \Pr[\{\alpha^d(t) = \Upsilon^d\} \cap \{n(t) > 0\}]
\]

Efficiency needs to be derived from the definition,

\[
E^u = \Pr[\{\alpha^u(t + 1) = \Upsilon^u\} \cap \{n(t) < N\}]
\]

with the fact

\[
\tau_j^u(\Pr[\alpha^u = \Delta_j^u \cap n < N] + \Pr[\alpha^u = \Delta_j^u \cap n = N])
\]

\[
= p_j^u \Pr[\alpha^u = \Upsilon_j^u \cap n < N] + q_j^u \Pr[\alpha^u = \Upsilon_j^u \cap n = N]
\]
Two-Machine Line – Efficiency

\[ E^u = E^d \]

\[
E^u = \sum_{n=0}^{N-1} Pr(n, \gamma^u, \gamma^d) + \sum_{n=0}^{N-1} \sum_{l=1}^{L} Pr(n, \gamma^u, \Delta^d_j) + Q^u \sum_{l=1}^{L} Pr(N, \gamma^u, \Delta^d_l) - \sum_{j=1}^{J} r^u_j \sum_{l=1}^{L} Pr(N, \Delta^u_j, \Delta^d_l)
\]

\[
E^d = \sum_{n=1}^{N} Pr(n, \gamma^u, \gamma^d) + \sum_{n=1}^{N} \sum_{j=1}^{J} Pr(n, \Delta^u_j, \gamma^d) + Q^d \sum_{j=1}^{J} Pr(0, \Delta^u_j, \gamma^d) - \sum_{l=1}^{L} r^d_l \sum_{j=1}^{J} Pr(0, \Delta^u_j, \Delta^d_l)
\]
Heuristics

- Two pseudo-machines: Lp1 and Lp2
- Analyze two lines separately using single part type machine line decomposition
Heuristic Results

- Type one production rate % error: 2%
- Type two production rate % error:

<table>
<thead>
<tr>
<th>Case</th>
<th>Parameter</th>
<th>Mean</th>
<th>Expected Demand Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-100</td>
<td>$r$</td>
<td>0.1</td>
<td>$E_1 = 0.55$</td>
</tr>
<tr>
<td></td>
<td>$p_1$</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p_2$</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>101-200</td>
<td>$r$</td>
<td>0.1</td>
<td>$E_1 = 0.55$</td>
</tr>
<tr>
<td></td>
<td>$p_1$</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p_2$</td>
<td>0.15</td>
<td>$E_2 = 0.40$</td>
</tr>
<tr>
<td>201-300</td>
<td>$r$</td>
<td>0.1</td>
<td>$E_1 = 0.40$</td>
</tr>
<tr>
<td></td>
<td>$p_1$</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p_2$</td>
<td>0.08</td>
<td>$E_2 = 0.55$</td>
</tr>
</tbody>
</table>

Table 1: Randomly generated demand machine parameters
Decomposition Equations:

- I don’t want you to get bored…
Algorithm

- Jang DDX algorithm
- 6 unknowns, 13 equations => Over constrained equations
- Very robust -> always converges
Simulation Case

Random Number Generation
- Triangular distribution
- Two processing machines two supply machine and two demand line case

$e_4 = \text{efficiency of } M'_{d1}$
$e_5 = \text{efficiency of } M'_{d2}$
$E_1$ Error

- Abs($E_1$ error) = 0.80712 %
- Std($E_1$ error) = 0.8627

$$\% Error = 100 \times \frac{E_{decomp} - E_{sim}}{E_{sim}}$$

![Error Plot](image)
$E_2$ Error

- $\text{Abs}(E_2 \text{ error}) = 1.24\%$
- $\text{Std} = 0.8934$
Case 1

- Type one supply varies
- Reliable type two machines
Case 2

- Type two demand varies
- Reliable type two machines
Case 3

- Type 2 demand decreases
Future Work

- Expand Line to long processing line with five part type cases
- Applying re-entrance flow
- CPP with multiple-part type
- Continuous time line
Reference


- Diego A. Syrowicz, Decomposition Analysis of a Deterministic, Multiple-Part-Type, Multiple-Failure-Mode Production Line, Massachusetts Institute of Technology, SM Thesis 1998