Solving a Multi Objective Mixed Model Assembly Line Balancing Problem in Garment Industry

Sandeep Choudhary\textsuperscript{a}, Sunil Agrawal\textsuperscript{b}

\textsuperscript{a,b}PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, 482005, E-mail: \textsuperscript{a}mech10165@gmail.com, \textsuperscript{b}sa@iiitdmj.ac.in

Abstract - The motivation of our work is from the paper by Chen et al. (2012) in which the assembly line balancing (ALB) problem in the garment manufacturing process for a single model is solved by using genetic algorithm. Out of four steps of manufacturing the product, we focus on the third step in their paper in which there is a sewing line which is to be operated by different employees having different skill levels. Further, textile industry manufactures many varieties of cloths with different designs and it is not possible to keep new lines for all different designs of cloth. The best way is to launch the models on the same line that can efficiently handle all the different operations needed by the models. This will also reduce the manufacturing cost of the plant. Therefore, in this paper we have considered the stitching of more than one model during the sewing operation on the same line. The problem is to assign proper task to workstations so that machines on workstations can perform the assigned task with balanced workload.

Keywords – Assembly line; Mixed model; Multiobjective; Branch and bound

1. Introduction

Garment industry or apparel industry is one the leading industry in the secondary sectors that employs thousands of workers working under the same shed to make the multiple varieties of a product. The raw material may be of different types, i.e., natural or synthetic. The process to make cloth from the raw material to finish product is too long and require enough capital investment. We are more concerned with the work carried out on an assembly line. Few tasks in manufacturing of cloth are bleaching, dyeing, screen printing, curing of screen printed cloth, knitting fusing stream pressing etc. These tasks are distributed on a series of long interconnecting workstations, where each individual task is assigned at the only single workstation. Growing demand for new designed cloth is one of the challenges for many companies in term of producing it in large quantities. For producing more than one variety at a small investment is an interest of every company. Making two separate assembly lines for two different varieties is not the best option. The best viable option for this is to launch the two varieties on the same line by adopting the concept of mixed model assembly line.

The concept of mixed model assembly line says that, the tasks are to be assigned on any workstation in such a way that the precedence of tasks does not get violated. Of course, to make any product we need to go through the sequence of task without skipping or jumping directly to the any other task. There are certain assumptions that need to follow before designing any assembly plant layout. On mixed model assembly line, there is more than one precedence diagram for different models. Combined precedence is to be formed by using the individual item’s precedence diagrams.

Thomopoulos (1967) was the first to use the combined precedence diagram to solve mixed model assembly line balancing problem (MALBP). Fokkert and Kok (1997) summarized the advantages and disadvantages of the combined precedence diagram method. According to their study, an advantage of this method is that every repetition of a task is performed by the same workstation, resulting in minimum learning costs. On the other hand, disadvantage of this method is related to the balancing on shift basis (Guden, 2006). The assembly line is the optimization problem where tasks are needed to be arranged over workstation in an optimal way. For this purpose optimization model is to be adopted. There are many optimization models found in the literature. Branch & bound, genetic algorithm, particle swarm algorithm, ant colony optimization, tabu search etc are few of those used.
2. Mathematical formulation

In the garment industry, workers have different skill levels to do the work. In order to run an assembly line in an effective way, tasks are to be assigned to the workers who are capable of handling the respective tasks. For example, stitching a button on a shirt, sewing pocket and printing tattoo are the three tasks to be performed by any worker. These three tasks have their respective task times. The tasks are to be assigned to workers according to the skill they hold. The problem arises when same skilled is hold by two different workers, whom to assign the task? Such type of problem is tackled in this paper by introducing four new constraints. These four constraints are added to already existing mathematical formulation given by Bukchine and Rabinowitch (2006) with an objective of minimizing the total cost (the sum of the station and task costs).

![Diagram showing skill level of machine and worker](image)

**Figure 1.** Skill level of machine and worker

Indices

- \( i \) : indices of the task (\( i, h=1,2,...n \))
- \( j \) : indices of the model (\( j=1,2,...q \))
- \( m \) : indices of the machine (\( m=1,2,...M \))
- \( k \) : indices of the station (\( k=1,2,...K \))
- \( w \) : indices of the worker (\( w=1,2,...W \))

Parameters

- \( t_{ij} \) : process time of task \( i \) when performed on model \( j \).
- \( IP_{ij} \) : set of immediate predecessors of task \( i \) in model \( j \).
- \( c_j \) : required cycle time for model \( j \).
- \( SC \) : station cost—fixed cost associated with each station.
- \( TC_i \) : task cost—fixed cost associated with each station to which task \( i \) is assigned
- \( D_{im} \) : matrix contains data of different skill level of machine performing different task.
- \( E_{mw} \) : matrix contains data of different skill level of worker performing different machine.

Decision variables

- \( z \) number of stations to be used in the assembly line.
- \( A_{ijm} = \begin{cases} 1 & \text{if machine } m \text{ is assigned to task } i \text{ of model } j, \\ 0 & \text{otherwise.} \end{cases} \)
- \( B_{mw} = \begin{cases} 1 & \text{if worker } w \text{ is assigned to machine } m, \\ 0 & \text{otherwise.} \end{cases} \)
- \( x_{ijk} = \begin{cases} 1 & \text{if task } i \text{ of model } j \text{ is assigned to station } k, \\ 0 & \text{otherwise.} \end{cases} \)
\[ \tau_{ik} = \begin{cases} 1 & \text{if task } i \text{ of any model is assigned to station } k, \\ 0 & \text{otherwise}. \end{cases} \]

Objective

\[
\text{Min=} \quad SC \cdot z + \sum_{i=1}^{n} TC_i \sum_{k=1}^{n} \tau_{ik} \quad (1)
\]

s.t.

\[
\sum_{k=1}^{n} x_{ijk} = 1 \quad \forall i..n, j..q \quad (2)
\]

\[
\sum_{m=1}^{n} A_{ijm} \cdot D_{im} = 1 \quad \forall i..n, j..q \quad (3)
\]

\[
\sum_{i=1}^{n} A_{ijm} \cdot D_{im} = 1 \quad \forall m..M, j..q \quad (4)
\]

\[
\sum_{w=1}^{n} B_{mw} \cdot E_{mw} = 1 \quad \forall m..M \quad (5)
\]

\[
\sum_{m=1}^{n} B_{mw} \cdot E_{mw} = 1 \quad \forall w..W \quad (6)
\]

\[
\sum_{k=1}^{n} k \cdot x_{ijk} \leq \sum_{l=1}^{n} k \cdot x_{hjl} \quad \forall j, g, h \quad \text{s.t. } g \in IP_{hj} \quad (7)
\]

\[
\sum_{i=1}^{n} x_{ijk} \cdot \tau_{ij} \leq c_j \quad \forall i..n, k..K \quad (8)
\]

\[
\sum_{k=1}^{n} k \cdot x_{ijk} \leq z \quad \forall i..n, j..q \quad (9)
\]

\[
\frac{1}{m} \sum_{j=1}^{m} x_{ijk} \leq \tau_{ik} \quad \forall i..n, k..K \quad (10)
\]

\[x_{ijk}, \tau_{ik} \in [0,1] \quad (11)\]

The objective function (1) is to minimize the total cost (sum of the station and task costs). Equity constraint (2) ensures that each task has to be assigned to exactly one station. Constraint (3) and constraint (4) ensures that for each task, there must be one assigned machine. For each machine, there must be one assigned worker is depicted by constraint (5) and constraint (6). Inequity (7) shows the precedence relationships among the tasks, which ensures that a task will be assigned to a certain station \( k \) only if its immediate predecessors have all been assigned to that station or to an upstream station. Constraint (8) restricts the total process time of each model at each station not to exceed the model cycle time. Constraint set (9) restricts the total number of stations to be used. Constraint set (10) examines whether task \( i \) of any model is assigned to station \( k \). [Bukchine and Rabinowitch (2006)] (constraint (3), (4), (5) and (6) are the newly added)
3. Numerical illustration

The problem considers two different models to be assembled on an assembly line. There are 15 machines and 15 workers required to do the complete job. The precedence diagrams of two models are shown in figure 2 and figure 3 respectively. Figure 4 shows the combined precedence diagram of both the models. Further, the time required by a task to perform the operation by each models and task cost are given in table 1. Station cost required to open new station is 10 and cycle time for model 1 and model 2 is 12 and 10. Table 2 show the capable machine and capable worker to perform task (task 1 can be performed by any of the capable machines 1, 3, 5 correspondingly to operate the machine any worker can be chosen among the worker 1, 2, 5, 6. Similarly if machine 3 is selected, the selection of worker should be among 2, 3, 4, 5 respectively and so). The problem is to distribute the tasks on workstations with corresponding capable machines with an objective of minimization of the total cost (the sum of the station and task costs.

![Figure 2. Precedence diagram of model 1](image1)

![Figure 3. Precedence diagram of model 2](image2)

![Figure 4. Precedence diagram of combine models](image3)
Table 1. Process times and task costs

<table>
<thead>
<tr>
<th>Task number</th>
<th>Task times of Model 1</th>
<th>Task times of Models 2</th>
<th>TC_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2. Matrix for task, machine and worker

<table>
<thead>
<tr>
<th>Task</th>
<th>Capable</th>
<th>Machine</th>
<th>Worker capable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,3,5</td>
<td></td>
<td>1,2,5,6</td>
</tr>
<tr>
<td>2</td>
<td>4,2</td>
<td></td>
<td>5,6,7,8</td>
</tr>
<tr>
<td>3</td>
<td>5,3</td>
<td></td>
<td>2,3,4</td>
</tr>
<tr>
<td>4</td>
<td>1,2,5,10</td>
<td></td>
<td>1,2,5,6</td>
</tr>
<tr>
<td>5</td>
<td>1,8</td>
<td></td>
<td>1,2,5,6</td>
</tr>
<tr>
<td>6</td>
<td>9,7</td>
<td></td>
<td>2,13,14</td>
</tr>
<tr>
<td>7</td>
<td>6,10,11</td>
<td></td>
<td>5,6,7,2,11,14,15</td>
</tr>
<tr>
<td>8</td>
<td>6,9</td>
<td></td>
<td>5,6,7,2,13,14</td>
</tr>
<tr>
<td>9</td>
<td>4,7</td>
<td></td>
<td>5,6,7,8,11,12</td>
</tr>
<tr>
<td>10</td>
<td>8,9,12</td>
<td></td>
<td>11,14,15,2,13,14,3,4,11,12</td>
</tr>
<tr>
<td>11</td>
<td>11,12</td>
<td></td>
<td>1,2,3,15,3,4,11,12</td>
</tr>
<tr>
<td>12</td>
<td>14,15</td>
<td></td>
<td>3,6,7,9,1,5,7,8</td>
</tr>
<tr>
<td>13</td>
<td>13,14</td>
<td></td>
<td>5,7,14,15,3,6,7,9</td>
</tr>
<tr>
<td>14</td>
<td>10,13</td>
<td></td>
<td>2,11,14,15,5,7,14,15</td>
</tr>
<tr>
<td>15</td>
<td>15,6</td>
<td></td>
<td>1,5,7,8,5,6,7</td>
</tr>
</tbody>
</table>

4. Results and Discussion

The following result is obtained after solving the mathematical model (eqs. 1 to 11) by branch & bound algorithm using lingo 10. Table 3 shows the optimized result after computation. Each worker is assigned to a machine according to his/her skill level. Similarly every task of two different models is assigned to a machine corresponding to their ability to perform it.

Table 3. Task, machine and worker assignment

<table>
<thead>
<tr>
<th>Station</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task model-1</td>
<td>5,6</td>
<td>2,12</td>
<td>1</td>
<td>3</td>
<td>4,11,9</td>
<td>7</td>
<td>8,10,15</td>
</tr>
<tr>
<td>Task model-1</td>
<td>5,6</td>
<td>2,12</td>
<td>1</td>
<td>3</td>
<td>4,9</td>
<td>7,14</td>
<td>8,13,15</td>
</tr>
<tr>
<td>Machine</td>
<td>8,7</td>
<td>2,15</td>
<td>1</td>
<td>3</td>
<td>5,4,11</td>
<td>10,13</td>
<td>9,12,14,6</td>
</tr>
<tr>
<td>Worker</td>
<td>14,12</td>
<td>10,4</td>
<td>1</td>
<td>5</td>
<td>4,6,3</td>
<td>2,15</td>
<td>13,11,9,7</td>
</tr>
</tbody>
</table>

Table 4 shows the performance of assembly line when two different models are launched on the same assembly line with different cycle times. Workstations and total expenditure to open station is minimized.

Table 4. Performance of assembly line

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Description</th>
<th>Computation result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cycle time (model 1, model 2)</td>
<td>12, 10</td>
</tr>
<tr>
<td>2</td>
<td>Total operations</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>Total workers</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>Total Machines</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Total Workstations</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>Total cost expenditure to open stations</td>
<td>70</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, every worker has multiple skill levels to do the tasks. Similarly the machines have different capability levels to perform the tasks. Therefore, due to the large number of combinations, the assignment problem becomes...
complex. However, this is overcome by introducing the four new constraints in the paper. Keeping in view of the objective function of minimization of workstation and manufacturing cost, the problem is solved by branch and bound algorithm using lingo 10. As a result, the number of workstations and the cost associated to open any new workstations is minimized.

REFERENCES


