ABSTRACT
In recent years, many manufacturers have adopted a Just-in-Time (JIT) approach to manufacturing. One of the important changes resulting from JIT implementation is the replacement of the traditional straight lines with U-shaped production lines. The important characteristic of these new configurations is that multiskilled workers perform various tasks of different stations along the production line. This feature allows assigning tasks both from the beginning and the end of the precedence diagram to the same stations. On the other hand, the problem of assigning tasks on a U-line configuration is more complex than the straight line due to the much larger search space. This paper addresses the U-shaped assembly line balancing problem type I (UALBP-1; the number of workstations is minimized for a given cycle time) using heuristic methods. To measure performance of the heuristics, they are applied to solve a large number of benchmark problems. The results are compared with the optimal solutions obtained from the previous published research. The computational results indicate that the maximum task time rule is able to identify better solutions and performs quite effectively. Finally, conclusion and future research are also presented in this paper.

KEY WORDS
Just-In-Time (JIT), U-Shaped Assembly Line Balancing Problem (UALBP), Optimization Problem, Heuristic Methods

1. Introduction
A production line is often used to take advantage of mass production. As a consequence of the implementation of just-in-time (JIT) principles into manufacturing, many companies are organizing their production processes into U-shaped production lines (Figure 1) rather than traditional straight production lines [3]. When compared to straight lines, they typically have better balancing, improved visibility and communications, fewer work stations, more flexibility for adjustment, minimization of operation travel, and easier material handling [2].

The traditional line or straight line assembly line balancing problem considers a production line in which stations are arranged consecutively in a line. A balance is determined by grouping tasks into stations while moving forward through a precedence diagram. However, the U-line assembly line balancing problem is more complex than the straight line because tasks can be assigned by moving forward, backward, or simultaneously in both directions through the precedence diagram.

Figure 1: U-shaped production line, [10]

In terms of the solution of the line balancing problem, this implies that the solution of the U-shaped line configuration dominates the solution of the traditional straight line configuration due to the number of stations on a U-line is less than or equal to the number of stations required on a straight line [2].

2. Literature Review
Single model and mixed model straight line assembly line balancing have been thoroughly researched since the first published work in 1955. However, the first published work on U-shaped lines was not until 1994. In comparison to the well studied straight assembly line balancing problem, there are many areas in U-line assembly line balancing which require further research [2].

The first UALBP study in the literature was by Miltenburg and Wijngaard [9], who developed a DP formulation for the single-model U-line to minimize the number of stations. The authors presented a Ranked Positional Weight Technique (RPWT)-based heuristic for larger size problems (111-tasks problems). Later, Miltenburge and Sparling [8] developed three exact algorithms to solve the UALBP. The first was based on a reaching DP formulation, whereas the other two were breadth- and depth-first branch-and-bound (B&B) algorithms.
Later, Urban [13] developed an integer linear programming formulation to solve small- to medium-sized of UALBP with up to 45 tasks. Scholl and Klein [10] developed a branch-and-bound procedure to solve, either optimally or suboptimally, problem with up to 297 tasks. Mixed-model U-lines were studied by Sparling and Miltenburg [12]. They developed a heuristic procedure for the U-line by which different products were assembled simultaneously. Their approximate solution algorithm that merges each model’s precedence diagram into a single precedence diagram solved problems with up to 25 tasks. Miltenburg [7] proposed a DP formulation for a U-line facility that consisted of numerous U-lines connected by multilines stations. Sparling [11] developed heuristic solution procedures for a U-line facility consisting of individual U-lines operating at the same cycle time and connected with multilines stations. Ajembil and Wainwright [1] developed a genetic algorithm. Erel et al. [4] proposed simulated annealing as solution methodologies for larger U-line and Hadi Gokcen et al. [6] presented a shortest route formulation of simple U-type assembly line balancing problem and illustrated on a numerical example.

This paper addresses the U-shaped assembly line balancing problem type I (UALBP-1; the number of workstations is minimized for a given cycle time) using heuristic methods. To measure performance of the heuristic rules, they are applied to solve a large number of benchmark problems and compared with the optimal solutions obtained from the previous published research.

3. A Definition of UALBP

The U-line assembly line balancing problem (UALBP) is an extension of simple assembly line balancing problem (SALBP) which is based on a U-shaped assembly line instead of a serial line. As in the case with SALBP, it can define three problem versions of UALBP (cf. Miltenburg and Wijngaard [9]) as well as Scholl and Klein [10].

- UALBP-1 : Given the cycle time (c), minimize the number of station (m)
- UALBP-2 : Given the number of stations (m), minimize the cycle time (c)
- UALBP-E : Maximize the line efficiency (E) for c and m being variable.

Since models for UALBP differ from those for SALBP only with respect to the precedence constraints. In SALBP all (direct and indirect) predecessors of a task j performed at a station k must be assigned to one of the stations 1,...,k.

In UALBP, each task in principle can share a station with any of its predecessors or successors. However, all predecessors or (and) all successors of a task j performed at a station k must be assigned to one of the station I,...,k. In many cases, a higher efficiency is possible with UALBP. Note that increasing the line efficiency has the further positive effect of smoothing the levels of station utilization, i.e., the stations get more equally loaded.

The simple U-line assembly line balancing problem defined by Miltenburg and Wijngaard [9] is given as follows: Miltenburg and Wijngaard’s definition follows from that given by Gutjahr and Nemhauser [5] for the traditional line balancing problem.

Given set of tasks $F = \{i \mid i = 1,2,\ldots, n\}$, a set of precedence constraints $P = \{(x,y)\}$ task x must be completed before task y, a set of task times $T = \{t_i \mid i = 1,2,\ldots, n\}$, cycle time c and a number of workstation m, find a collection of subsets of $F, (S1, S2,\ldots,Sn)$ where $Si = \{i\}$ task i is done at a workstation k, that satisfy the following conditions:

$$ \sum_{k=1}^{n} S_k = F $$

$$ S_j \bigcap_{k=1}^{n} S_k = \emptyset $$

$$ \sum_{i \in S_k} t_i \leq c, \quad k = 1,2,\ldots,n $$

For each task y, if $(x,y) \in P, x \in S_k, y \in S_i$, then $k \leq j$, for all x; or if $(y,z) \in P, y \in S_k, z \in S_i$, then $i \leq j$, for all z.

Condition 1 ensures that all tasks are assigned to a workstation. As a result of condition 2, each task is assigned only once. Condition 3 ensures that the work content of any workstation does not exceed the cycle time. Condition 4 ensures that the precedence constraints are not violated on the U-line. As a result of the objective function, the number of workstations will be minimized [9].

4. Heuristics

4.1 Heuristics rules

Two heuristic rules (maximum task time and minimum task time) are used in this research to find solutions to the UALBP-1. These heuristic rules were previously used to solve the SALBP. However, to allow them to work for UALBP some modifications were made because each task in principle can share a station with any of its predecessors or successors. The proposed heuristics are described below.
Let \( U_i^p \) be the set of tasks which must precede task \( i \),
Let \( U_i^s \) be the set of tasks which must succeed task \( i \),
Then at any instant the set of assignable tasks, \( V = \{ i \mid \text{all } x \in U_i^p \text{ or all } y \in U_i^s \text{ have already been assigned} \} \).

1.) Maximum task time
   The priority function \( p_{\text{max}}(i) \), called the U-line maximum task time
   \[
   p_{\text{max}}(i) = t(i) 
   \]

2.) Minimum task time
   The priority function \( p_{\text{min}}(i) \), called the U-line minimum task time
   \[
   p_{\text{min}}(i) = t(i) 
   \]

4.2 Numerical illustration
   An example problem with 7 tasks and a cycle time of 10 units is considered. The precedence diagram and straight line configuration results are shown in Figure 2 (a), (b) respectively. It can be seen that all tasks are performed at 5 stations in traditional line balancing. The illustration example for assigning tasks to work stations in U-line by using maximum task time rule is shown in Table 1.

<table>
<thead>
<tr>
<th>station</th>
<th>Candidate list</th>
<th>assigned</th>
<th>Task time</th>
<th>Idle time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,7,2,3,7</td>
<td>1</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2,3,4,5,6</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2,4,5,6</td>
<td>5</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2,6</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1 shows the illustration for assigning tasks to the stations in U-line by using maximum task time rule. The assignable tasks for the first station are \( V = \{1,7\} \). Consider max. task time, \( \max(t(1), t(7)) = \max(7,3) = 7 \), we assign task 1, then \( V = \{2,3,7\} \) as task 7 has the highest priority and sufficient cycle time remaining, it is also assigned to workstation 1, the remaining assignment process is described on Table 1. From the results, the U-line configuration is depicted in Figure 3.

Figure 3 illustrates the solution of UALBP-1 with four stations : \( S_1 = \{1,7\} \), \( S_2 = \{3\} \), \( S_3 = \{5,4\} \), \( S_4 = \{2,6\} \) and with station times of 10, 8, 9 and 10, respectively. In U-line, stations can include tasks located on different parts of the production line. For example the first station consists of tasks 1 and 7, where task 1 is located at the beginning of the line while task 7 is located at the end of the line. In this case, the number of stations required on a U-line is less than one stations on a straight line.

5. Computational Results

Several well-known test problems were considered by Miltenburg and Wijngaard [9]. We consider only the larger problems (21 or more tasks) in U-line. Table 2 presents the results of the proposed heuristics and compares the results with [9] by using a maximum ranked positional weight heuristic and the optimal solutions of the U-shaped line balancing problem [13].

As seen by the results, problems of up to 45 tasks can be solved with two priority rules of heuristics (maximum task time and minimum task time) for assigning tasks to stations. The problems were solved on a personal computer using FoxPro 6.0 with a Pentium 4, 3.0 GHz, 512 MB RAM and the operating system on windows XP. The computational results indicated that the maximum task time rule was able to identify better solutions than [9] on 2 out of 25 problems and take less than ten seconds in each instance problems (this take about 5.72 sec. on average). The minimum task time rule gave very poor results when compared to the others. It only obtained 4
optimal solutions out of 25 instances. Concerning the average relative deviations from the best known solution, the ranking is a max. task time (rank 1), max. RPW (rank 2) and a minimum task time (rank 3) respectively.

6. Conclusion

Recently, U-shaped line has been utilized in many production lines in place of the traditional straight line configuration due to the use of just-in-time principle. The shape of U-lines improves visibility and allows the construction of stations containing tasks on both sides of the line. This arrangement, combined with cross-trained operators, provides greater flexibility in station construction than is available on a comparable straight production line.

This paper studies the U-shaped assembly line balancing problem and two priority rules of heuristics for assigning tasks to stations are presented. The performance of the heuristics are applied to solve a large number of benchmark problems. Its results are compared with the optimal solutions and a maximum ranked positional weight heuristic which obtained from the previous published research. The computational results indicated that one of the heuristics rules (a maximum task time heuristic) can produce good solutions and the computational requirements are not high. This study has taken a step in the direction of finding good heuristic rules to solve the UALBP. It is possible that different heuristic rules with different problems may produce different results. Because there are a large variety of UALBP.

For further research, it would be interesting to use other heuristics, metaheuristics (e.g. tabu search, genetic algorithms, ant system etc.) and find more flexible solution approaches in the larger U-shaped assembly line balancing problem.

### Table 2: Solutions of the Test Problems for U-Shaped Line Balancing

<table>
<thead>
<tr>
<th>Problems</th>
<th>Cycle time</th>
<th>IP solution</th>
<th>Max. 2</th>
<th>Max. task time</th>
<th>Min. task time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>m</td>
<td>m %</td>
<td>m %</td>
<td>Cal. Time (sec.)</td>
<td>m % Cal. Time (sec.)</td>
</tr>
<tr>
<td>Mitchell 21</td>
<td>14</td>
<td>8</td>
<td>0</td>
<td>3.36</td>
<td>10 25</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>8</td>
<td>0</td>
<td>3.09</td>
<td>8 0</td>
</tr>
<tr>
<td>Heskiaoff 28</td>
<td>21</td>
<td>5</td>
<td>6</td>
<td>3.80</td>
<td>6 20</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>5</td>
<td>6</td>
<td>3.80</td>
<td>6 20</td>
</tr>
<tr>
<td>Sawyer 30</td>
<td>25</td>
<td>14</td>
<td>15</td>
<td>7.14</td>
<td>14 0</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>13</td>
<td>14</td>
<td>7.69</td>
<td>13 0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>11</td>
<td>12</td>
<td>9.09</td>
<td>12 0</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>10</td>
<td>11</td>
<td>9.09</td>
<td>12 0</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>10</td>
<td>10</td>
<td>5.32</td>
<td>12 20</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>8</td>
<td>9</td>
<td>5.32</td>
<td>9 12.50</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>6</td>
<td>7</td>
<td>5.92</td>
<td>7 0</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>5</td>
<td>5</td>
<td>5.92</td>
<td>5 0</td>
</tr>
<tr>
<td>Kilbridge &amp; Wester 45</td>
<td>57</td>
<td>10</td>
<td>10</td>
<td>8.42</td>
<td>12 20</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>7</td>
<td>8</td>
<td>14.28</td>
<td>8 14.28</td>
</tr>
<tr>
<td></td>
<td>92</td>
<td>6</td>
<td>7</td>
<td>16.67</td>
<td>7 16.67</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>6</td>
<td>6</td>
<td>9.14</td>
<td>6 0</td>
</tr>
<tr>
<td></td>
<td>138</td>
<td>4</td>
<td>5</td>
<td>8.93</td>
<td>5 25</td>
</tr>
<tr>
<td></td>
<td>184</td>
<td>3</td>
<td>4</td>
<td>9.16</td>
<td>4 33</td>
</tr>
<tr>
<td>Total optimal solutions (or lower bound) found from 25 instances problem</td>
<td>11</td>
<td>8.63</td>
<td>/inst.</td>
<td>13</td>
<td>8.35</td>
</tr>
</tbody>
</table>

1 Integer Programming solution, results from Urban[13] (in table 1, p.740)
2 Maximum Ranked Positional Weight heuristic researched by Miltenburg and Wijngaard [9].
3 m is the number of stations.
4 % is the average relative deviation from the best known solution
Acknowledgements

The authors would like to thank the faculty of Engineering, Ubon Rajathanee University for the financial support.

References