Article

U-shaped Production Line Design with Walking Times and Exchangeable of Task Locations

|  |
| --- |
| **Citation:** Kuo, Y.; Yang, T.; Huang, T.L. U-shaped Production Line Design with Walking Time and Exchangeable of Task Locations. *Appl. Sci.* **2021**, *11*, x. https://doi.org/10.3390/xxxxx  Academic Editor: Firstname Lastname  Received: date  Accepted: date  Published: date  **Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.    **Copyright:** © 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). |

Yiyo Kuo 1,\*, Taho Yang 2 and Tzu-Lin Huang 3

1 Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City 24301, Taiwan; yiyo@mail.mcut.edu.tw

2 Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan City 701, Taiwan; tyang@mail.ncku.edu.tw

3 Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City 24301, Taiwan; m08218013@o365.mcut.edu.tw

\* Correspondence: yiyo@mail.mcut.edu.tw; Tel. +886-2-2908-9899#3118

**Abstract:** U-shaped production lines have some advantages that improve labor productivity. Unlike traditional straight production line, in U-shaped production lines, production tasks located in both legs of the line can be assigned to a single operator. Operators can process production tasks in a crossover workstation. Because the two legs are close, the walking time of operators can be minimized and then the cycle times can be reduced. When designing a U-shaped production line, the locations of all tasks have to follow the precedence relationship of production and the walking times between tasks must be considered. This research proposed an integer programming to optimize the locations of all production tasks and assign production tasks to operators. When given a number of operators, the proposed integer programming can minimize the cycle time that taking walking time into account. The results of an experiment show that the U-shaped production lines outperform transitional straight-lines and that U-shaped production lines designed taking walking time into consideration provide an opportunity to reduce cycle times.

**Keywords:** U-shaped production line; integer programming; precedence relationship; walking time; line balancing problems

1. Introduction

Production lines can be classified into two general groups depending on their configuration: traditional straight production lines and U-shaped production lines [1]. A traditional straight production line organizes the tasks sequentially in one direction to form stations [2]. A U-shaped production line, however, is divided into two sub-lines: the entrance sub-line and exit sub-line, and thus an operator may perform tasks on either one of the two sub-lines or on both sub-lines simultaneously [3]. In U-shaped production lines, if tasks assigned to a workstation are located in both sub-lines then the corresponding operator has to move between the two sub-lines to perform combinations of tasks. In that case the operator has to perform one workpiece by some production tasks in one sub-line sequentially and pass the workpiece to the next adjacent operator, then the operator moves to another sub-line and produces another workpiece. A workstation that handles two workpieces in the same cycle is called a crossover station [4] For example, in Figure 1, there are 8 production tasks. Figure 1(a) is a traditional straight production line. The workpieces flow from the left hand side to the right hand side, tasks A, B and C are assigned to workstation 1, tasks D, E and F are assigned to workstation 2 and tasks G and H are assigned to workstation 3. Figure 1(b) is a U-shaped production line. Workpieces flow from the left hand side to the right hand side on the entrance sub-line and flow from the right hand side to the left hand side on the exit sub-line. The U- shaped production line provides more options for task assignment. For example, in Figure 1(b), tasks A, B and H are assigned to workstation 1, tasks F and G are assigned to workstation 2 and tasks C, D and E are assigned to workstation 3.



**Figure 1.** An example of traditional straight production line and U- shaped production line

U-shaped layouts have been widely employed in many industries [5]. The benefits associated with U-shaped production lines include better visibility of the whole production process, more communications among operators, better teamwork, lower material handling cost, fewer operator (or workstation) requirements, higher productivity, lower work-in-process inventory, less space requirement, shorter lead time, faster response to change in market demand and motivation for a higher quality level [6-10].



**Figure 2.** An example of precedence diagram

Many studies have confirmed that the U-shaped production line balancing problem is a very significant problem for modern production systems [11]. It aims to assign tasks to the workstations without violating any restriction [12], so that all workstations have equal amounts of work assigned to them [13]. For example, in Figure 2, there are 8 production tasks. The arrows indicate the precedence relationship between a pair of tasks. For example, the direction of the arrow between tasks B and E is from B to E. That means task E can be started only after task B is completed. The assignment of tasks to locations in the U-shaped production line has to follow the task sequence. Based on the precedence diagram, there are many feasible assignments of tasks to locations. Figure 3 provides a feasible assignment of tasks to locations based on the precedence diagram in Figure 2. In some production lines the assignment tasks to the line locations is fixed due to the sequence of machines on the line is often fixed [14]. However, the current situation of operations is characterized by an increasingly varied demand [15], multifunction and easy changeover is common in advanced machine tool. It makes machines on the line to exchange their tasks possible. Therefore, this research assumes that the assignment of tasks to location can be changed.



**Figure 3.** A feasible assignment of tasks to locations

In addition to assigning tasks to locations, assignment of tasks to workstations is also very important. It will affect the amounts of work in each workstation. In this research, it is assumed that one operator is assigned to carry out all tasks assigned to one workstation. Therefore, the number of operators is equal to number of workstations. When assigning tasks to workstations, the locations of tasks should be considered at the same time, or the operators will get in each other’s way. For example, in Figure 3, if tasks A and H are assigned to workstation 1, tasks B, E and D assigned to workstation 2, task C, F and G assigned to workstation 3. The result is shown in Figure 4. It can be seen that the walking paths of the operators in workstation 2 and 3 cross each other, an outcome that can be avoided if the task in location 7 is assigned to workstation 2. Therefore, if a location in the entrance sub-line (or exit sub-line) is close to the entrance (or exit), then the corresponding tasks cannot to be assigned to the workstation that is further from the entrance (or exit). This is called “location sequence” in the presented study. The task sequence and location sequence are illustrated in Figure 5. Figure 5 (a) illustrates the situation when the number of tasks is even and Figure 5 (b) illustrates the situation when the number of tasks is odd.



**Figure 4.** An example of interruption between operators



**Figure 5.** An illustration of task and location sequences for U-shaped production line

Moreover, the distances the operators need to walk and the required travel times in stations is also an important criterion [16]. Therefore, walking times should be taken into account to derive the exact cycle time [17]. For example, in Figure 6, although all tasks assigned to locations follow the task sequence, and all locations assigned to workstations follow the location sequence, tasks in location 1 and 5 assigned to workstation 2. Locations 1 and 5 are the furthest apart of all pairs of locations and the walking time between tasks usually depends on the locations of the tasks. Therefore, the design illustrated in Figure 6 would increase the walking time of the operator in workstation 2 and reduce the performance of the entire U-shaped production line.



**Figure 6.** An illustration of for long walking distance

Scholl and Becker [18] summarize four types of production line balancing problems (LBP). Type-1 aims to minimize the number of stations for a given cycle time. Type-2 aims to minimize the cycle time for a given number of workstations. Type-3 aims to maximize the efficiency without a specified cycle time or number of workstations. And Type-4 aims to find a feasible design for a given cycle time and number of workstations. This research deals with Type-2 LBP which is encountered frequently when the production rhythm must be adjusted in the manufacturing process [3]. If only one operator is assigned to a workstation in a U-shaped production line, a Type-2 LBP involves balancing the design of a product type for a given number of operators. That means for every product, the cycle times can be determined when given a number of operators. According to a survey of literature related to operator assignment, the results of all configurations are the main input of the decision model. Examples are given by [19-28]. The presented study extends these operator assignment related studies to manufacturing cell systems with U-shaped production lines.

This research deals with a U-shaped production line balancing problem (UPLBP) in which the task sequence has to satisfy all precedence relationships and location sequence has to satisfy the relationship between workstations. Moreover, the walking time between all locations in the U-shaped production line are also considered. A mathematical model which can be optimized using software LINGO 11 is developed for the proposed UPLBP. Some cases that have been studied in the literature are modified for testing, and the results for traditional straight production lines and U-shaped production lines are compared, with and without consideration of walking time. The remainder of this paper is organized as follows. Section 2 reviews the relevant literature. Section 3 presents the model formulation of the proposed problem. Details of the empirical illustration are discussed in Section 4. Conclusions and future research opportunities are addressed in Section 5.

2. 2. Literature review

The UPLBP is similar to the U-shaped assembly line balancing problem (UALBP). Both UPLBP and UALBP related research are reviewed. Miltenburg and Wijngaard [29] was the first study that proposed an exact methodology for UALBP [30]. They developed a mathematical model of UALBP, and then proposed a dynamic programming procedure for balancing small-size problems. For larger problems, a greedy heuristic based on ranked positional weight was proposed. The UALAP then became a classic problem [31] and has been widely studied in the literature [32]. Huang and Katayama [33] dealt with workload balancing problems in mixed-model U-shaped line and designed an amelioration structure with a genetic algorithm (ASGA) to improve it. Rabbani et al. [32] develop a genetic algorithm (GA) to balance a mixed model U-shaped production system. In their proposed mathematical model, the objective to minimize the number of crossover stations and maximize line efficiency at the same time. Avikal et al. [2] proposed a Critical Path Method (CPM) approach for assigning tasks to workstations in the U-shaped assembly line. They evaluated labor productivity in U-shaped line systems and straight line systems, and show that the proposed CPM can reduce the minimum number of workstations for assignment of the tasks in a U-shaped line layout below the number needed in a straight line layout. Nourmohammadi et al. [34] proposed an imperialist competitive algorithm (ICA) to minimize the number of workstations and the variation of workload in UALBP, simultaneously. The computational results show that the proposed algorithm outperforms GA, especially in large-sized test problems. Jayaswal and Agarwal [4] proposed a Simulated Annealing (SA) algorithm for resource dependent UALBP, in which the duration of task is dependent on equipment types and any assistance available. The results show that the proposed SA algorithm is able to solve most small-to-moderate size problem instances to optimality or close to optimality very efficiently. Hazir and Dolgui [5] deal with the robust UALBP, in which operation times were modeled using intervals and a robust optimization model was developed. An iterative approximate algorithm was proposed and tested with some computational experiments. They conclude that the model and solution algorithm developed could form a good base for further research on robust design and balancing of assembly lines. Oksuz et al. [30] assumed that all workers carry out all tasks with unequal performance levels and formulated a mathematical model for worker assignment in UALBP. Then an Artificial Bee Colony Algorithm and a Genetic Algorithm were used to solve it. Li et al. [3] deal with a UALBP and formulated an integer programming model. They proposed a novel heuristic approach based on multiple rules to minimize the cycle time with a given number of stations. Sahin and Kellegoz [35] developed an efficient solution method for UALBP. They examined six different solution methods for comparison. The results show that the proposed grouping genetic algorithm (GGA) with problem-specific crossover and mutation operators produces better solutions than the mathematical formulations and other solution methods available in the literature to solve different types of assembly line balancing problems. Aydogan et al. [1] proposed a novel particle swarm optimization algorithm to solve the UALBP with stochastic task times. The results of the computational study show that the proposed approach performs quite effectively. Zhang et al. [36] deal with task allocation and worker assignment in UALBP in which the processing time of each task depends on the workers. Then an enhanced migrating birds optimization algorithm (EMBO) was used to minimize the cycle time. The numerical results demonstrate that the proposed algorithm outperforms other algorithms. Zhang et al. [31] formulate a UALBP with worker assignment to simultaneously minimize cycle times and ergonomic risks. A Restarted Iterated Pareto Greedy algorithm is designed to optimize both objectives. The proposed multi-objective algorithm outperforms existing methods on a large number of benchmark instances. In all the above literature, walking time is not considered.

Studies that deal with walking time in UALBP or UPLBP are rare. Ohno and Nakade [17] deal with a U-shaped production line, and consider processing, operation and walking times. They formulated the optimal worker allocation problem that minimizes the overall cycle time and showed the advantages of the U-shaped layout over the linear layout. Nakade and Ohno [37] consider the optimal worker allocation problem for the U-shaped production line. They proposed an algorithm for finding an optimal allocation of workers to machines that minimizes the cycle time for the minimum number of workers. Shewchuk [14] addressed the worker allocation problem for lean U-shaped production lines. A mathematical model that takes into account walking time, with workers following circular paths and walking around other workers was developed. A heuristic algorithm was developed for optimizing the allocation of workers. The above literature related to U-shaped production line deals with walking time, but they all focus on the worker allocation problem in which locations of machines (or tasks) are fixed. Therefore, they ignore the alternatives locations of tasks that following precedence relationships.

The present research deals with UPLBP in which the precedence relationship between tasks and walking times between locations of tasks are taken into consideration simultaneously. The problem that the present study addresses is not found in the literature. This is the main contribution of this research.

3. Integer Programming

The main purpose of the proposed mathematical model is to minimize the maximum cycle time of all workstations for a given number of operators. One operator is assigned to each workstation, so the number of workstations is equal to the number of operators. The walking times between tasks are included in the cycle time of all operators. The locations of all tasks must follow the precedence relationship of production. Operators can carry out production tasks in a crossover workstation, but interruption between operators is not allowed. For the development of the mathematical model, the notation is defined in Table 1.

The mathematical model aims to minimize the cycle time as given in Equation (1). Equation (2) ensures that each assembly task is only assigned to one location and Equation (3) ensures that only one task is assigned to each location. If assembly task *r* has to be done before assembly task *s* based one assembly sequence, Equation (4) avoids assembly task *r* being assigned to the location after assembly task *s* based on task sequence as illustrated in Figure 1 and Figure 5. Equation (5) ensures that each location is assigned to only one workstation. In Equation (6), if location *p* is closer to the entrance (or exit) than location *q* in the U-shaped assembly line, location *q* will not be assigned to the workstation closer to the entrance (or exit) than location *p*. If task *t* is assigned to location *l* and location *l* is assigned to workstation *w*, task *t* is assigned to workstation *w*. Equation (7) is used for calculating the assembly times for all workstations.

**Table 1.** Definition of notations

|  |  |  |
| --- | --- | --- |
| Indices | | |
| *t, r, s* | task index | |
| *l, p, q* | location index | |
| *w* | workstation index | |
| Parameters | | |
| *T* | number of tasks | |
| *L* | number of locations (equal to number of tasks) | |
| *Wmax* | number of workstations (equal to number of operators available) | |
|  | a precedence relationship between tasks; production task *r* must be completed before task *s* starts | |
|  | a precedence relationship between locations; location *p* must be assigned to a workstation no further from the entrance (or exit) than that to which location *q* is assigned | |
| *CT* | cycle time | |
| *Ot* | production time of task *t* | |
| *ATw* | total production time of tasks assigned to workstation *w* | |
| *dpq* | walking time for the path between location 𝑝 and 𝑞 | |
| *WTw* | total walking time in workstation *w* | |
| Decision variables | | |
| *Xtl* |  | 1, task *t* assigned to location *l*  0, otherwise |
| *Ylw* |  | 1, location *l* assigned to workstation *w*  0, otherwise |
| *Zpqw* |  | 1, operator walk follows the path between location 𝑝 and 𝑞 in workstation *w*  0, otherwise |

This research assumes that all operators walk in a clockwise direction in their workstation. Equation (8) - (13) determine the walking paths of all operators in their corresponding workstations. Equation (8) adjusts the walking path between adjacent locations *p* and *q*, and *q*-*p*=1. Equation (9) adjusts the walking path between nonadjacent locations *p* and *q*, and *q*-*p*≥2. When an operator completes all the tasks assigned to their workstation, they have to walk back to the first location of the workstation. Equation (10)-(13) adjust the walking back path between locations *p* and *q*, and *p*-*q*≥1. Equation (10) adjusts the path when *p*≠*L* and *q*≠1. Equation (11) adjusts the path when *p*=*L* and *q*≠1. Equation (12) adjusts the path when *p*≠*L* and *q*=1. Equation (13) adjusts the path when *p*=*L* and *q*=1. Eq. For example, in Figure 2, there are three workstations, of which workstation 1 and 3 are crossover workstations. Equation (8) can determine *Z*121, *Z*672, *Z*343 and *Z*453, Equation (9) can determine *Z*281, Equation (10) can determine *Z*762 and *Z*533, Equation (13) can determine *Z*811. Equation (11) and (12) are not relevant for walking paths in Figure 2. Equation (14) calculates the walking times of operators in all workstations. Total production time and walking time in a workstation is the cycle time of the workstation and cannot exceed the cycle time given in Equation (15).

|  |  |
| --- | --- |
| Minimize *CT* | (1) |
|  | (2) |
|  | (3) |
|  | (4) |
|  | (5) |
|  | (6) |
|  | (7) |
|  | (8) |
|  | (9) |
|  | (10) |
|  | (11) |
|  | (12) |
|  | (13) |
|  | (14) |
|  | (15) |

4. Experimental results

The proposed integer programming model was validated on 13 problems in the literature. The corresponding information about the problems is summarized in Table 2. Because the walking times and layouts were ignored in previous studies, the present research adopted the ratios of average processing time as the walking time of one unit. All U-shaped assembly lines are laid out in a grid system, as illustrated in Figure 7(a). The walking time between any adjacent locations is one unit, and the minimal walking time between both sides of the cell is 2 units. In this research, walking time of one unit will be the ratio of average processing time, ranging between 0% and 20%. For example, in problem 1, proposed by Bowman [38], there are 8 tasks and the average processing time is 9.38. If one unit of walking time is 5% of the average processing time, the walking time between any adjacent locations is 0.469 (9.38×5%). To ensure that the walking time between adjacent locations is 1 unit as in traditional straight assembly, when the number of tasks is even the right hand side of the cell is laid out as in Figure 7(b). As all tasks are laid out in a grid system, walking times between all pairs of locations can be easily calculated using Pythagoras’ theorem.

For the 13 problems, 2, 3 and 4 workstations are tested using software LINGO with Intel Core i7 2.40 GHz notebook computer. The resulting cycle times for different walking times are shown in the last five columns of Table 2. For comparison, this research also adopts the proposed integer programming for designing traditional straight assembly lines. The location sequence and task sequence are illustrated in Figures 8. The results are shown in column 3 to 7 in Table 3.

**Table 2.** Summary of experimental problems

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Source** | | **Number of tasks** | **Average processing time** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | Figure 1  Figure 1  Figure 1  Figure 2  Figure 3  Figure 4  Figure 3  Figure 1  Figure 1  Figure 3  Figure 4  Figure 1  Figure 2 | 8  12  9  10  7  11  7  10  9  8  7  8  8 | 9.38  5.00  4.11  4.10  3.29  4.18  5.29  3.60  6.11  0.38  0.58  0.28  0.48 |



**Figure 7.** Layout of the U-shaped assembly cells



**Figures 8.** An illustration of task and location sequences for traditional straight assembly lines

**Table 3.** Cycle times of traditional straight and U-shaped assembly lines

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Problems** | **Traditional straight assembly line** | | | | | **U-shaped assembly line** | | | | |
| 0% | 5% | 10% | 15% | 20% | 0% | 5% | 10% | 15% | 20% |
| 2 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 38.00  30.00  19.00  21.00  12.00  23.00  19.00  18.00  28.00  1.61  2.04  1.41  1.93 | 41.75  32.50  20.23  22.64  12.66  24.88  20.59  19.44  30.44  1.66  2.21  1.18  2.12 | 45.50  35.00  21.47  24.28  13.31  26.76  22.17  20.88  32.89  1.78  2.39  1.28  2.31 | 49.25  37.50  22.93  25.92  13.97  28.65  23.76  22.32  35.33  1.89  1.36  1.36  2.50 | 53.00  40.00  24.58  27.56  14.94  30.53  25.34  23.76  37.78  2.00  2.74  1.44  2.68 | 38.0030.00  19.00  21.00  12.00  23.00  19.00  18.00  28.00  1.60  2.04  1.16  1.91 | 40.73  32.00  20.08  22.11  12.66  24.51  19.90  19.30  30.11  1.62  2.19  1.18  2.03 | 43.46  34.00  21.47  23.56  13.31  26.03  21.17  20.60  32.38  1.73  2.34  1.27  2.18 | 46.20  36.00  22.70  24.84  13.97  27.76  22.76  21.70  34.57  1.48  2.48  1.35  2.32 | 48.93  38.00  23.93  24.84  14.63  29.02  24.34  22.76  36.33  1.95  2.64  1.44  2.46 |
| 3 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 28.00  22.00  13.00  14.00  8.00  16.00  13.00  13.00  19.00  1.03  1.46  0.80  1.34 | 28.94  23.50  13.82  15.23  8.66  17.25  13.53  13.72  20.83  1.11  1.55  0.86  1.44 | 29.88  25.00  14.64  16.46  9.31  18.51  14.06  14.44  22.44  1.18  1.66  0.91  1.53 | 30.81  26.50  15.47  17.46  9.97  19.76  14.59  15.24  23.67  1.26  1.78  0.97  1.63 | 32.50  28.00  16.29  18.28  10.31  21.02  15.23  16.32  24.89  1.33  1.90  1.02  1.72 | 26.00  22.00  13.00  14.00  8.00  16.00  13.00  12.00  19.00  1.03  1.46  0.78  1.34 | 26.94  22.43  13.70  15.11  8.33  16.71  13.53  13.06  20.60  1.11  1.55  0.85  1.43 | 28.91  23.41  14.64  16.05  8.99  18.03  14.06  14.13  22.09  1.18  1.66  0.90  1.52 | 30.8125.12  15.47  17.08  9.97  19.28  14.59  15.02  23.67  1.25  1.78  0.96  1.61 | 32.50  26.83  16.29  18.06  10.31  20.38  15.23  15.77  24.89  1.32  1.90  0.96  1.69 |
| 4 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 22.00  16.00  11.00  11.00  6.00  12.00  11.00  10.00  15.00  0.88  1.13  0.61  1.10 | 22.94  17.00  11.41  11.82  6.66  12.84  11.53  10.72  16.22  0.92  1.19  0.64  1.15 | 24.75  18.00  11.82  12.64  7.31  13.67  12.06  11.44  17.44  0.96  1.25  0.67  1.20 | 26.63  19.00  12.23  13.23  7.97  14.51  12.95  12.08  18.33  1.01  1.30  0.69  1.26 | 28.50  20.00  12.64  13.64  8.31  15.35  13.11  12.88  19.89  1.06  1.36  0.72  1.31 | 20.00  15.00  10.00  11.00  6.00  12.00  10.00  10.00  14.00  0.88  1.09  0.58  1.04 | 22.24  16.19  10.82  11.41  6.56  12.35  10.53  9.61  15.60  0.92  1.19  0.61  1.14 | 23.88  17.41  11.64  12.61  7.12  13.58  11.99  11.39  16.44  0.96  1.25  0.66  1.20 | 24.81  18.08  12.23  13.23  7.68  13.84  12.59  11.83  17.67  1.00  1.30  0.69  1.26 | 25.75  19.37  12.64  13.82  8.24  14.84  13.11  12.44  18.89  1.06  1.36  0.72  1.31 |
| #: number of workstation | | | | | | | | | | | |

The percentage improvements of U-shaped production lines are shown in Table 4. From the total of 195 (13×5×3) cases, the design of U-shaped production lines results shorter cycle times in 131 cases. For different numbers of workstations and walking times, the average improvements range between 10.2% and 4.53%. Therefore, the comparison results confirm some of the arguments for the advantage of U-shaped production lines found in the literature. Moreover, the results of 0% for both traditional straight and U-shaped production lines are based on walking times that are 0% of the average processing time. That means these results correspond to calculations ignoring walking time. This research calculated the cycle times based on these results with different working times. The results are shown in Table 5 and the corresponding comparison with and without considering walking time are shown in Table 6.

**Table 4.** Improvement of U-shaped production lines (percentage)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Problems | 0% | 5% | 10% | 15% | 20% |
| 2 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | -  -  -  -  -  -  -  -  -  0.62  -  17.73  1.04 | 2.44  1.54  0.78  2.36  -  1.48  3.32  0.71  1.09  2.34  1.00  -  4.13 | 4.47  2.86  -  2.98  -  2.76  4.51  1.32  1.56  2.81  2.17  0.20  5.84 | 6.20  4.00  1.02  4.19  -  3.08  4.21  2.78  2.17  2.64  3.29  0.28  7.30 | 7.68  5.00  2.62  9.89  2.10  4.94  3.95  4.21  3.82  2.49  3.65  0.35  8.21 |
|  | average | 1.49 | 1.63 | 2.42 | 3.17 | 4.53 |
| 3 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 7.14  -  -  -  -  -  -  7.69  -  -  -  2.50  - | 6.91  4.54  0.87  0.82  3.80  3.13  -  4.78  1.12  -  -  0.43  0.44 | 3.23  6.34  -  2.49  3.45  2.61  -  2.17  1.59  0.01  0.00  0.81  0.83 | -  5.20  -  2.21  -  2.43  -  1.47  -  0.42  -  0.29  1.17 | -  4.18  -  1.20  -  3.04  -  3.37  -  0.77  -  5.93  1.48 |
|  | average | 1.33 | 2.07 | 1.81 | 1.02 | 1.54 |
| 4 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 9.09  6.25  9.09  -  -  -  9.09  -  6.67  -  3.54  4.92  5.45 | 3.02  4.76  5.16  3.47  1.45  3.77  8.67  10.40  3.84  -  -  4.70  1.09 | 3.54  3.27  1.50  0.22  2.63  0.67  0.56  0.47  5.73  -  -  1.35  - | 6.81  4.85  -  -  3.62  4.62  -  2.09  3.64  0.97  -  -  - | 9.65  3.15  -  2.64  0.85  3.29  -  3.42  5.03  -  -  -  - |
|  | average | 4.16 | 3.87 | 1.53 | 2.05 | 2.16 |
| #: number of workstation | | | | | | |

**Table 5.** Real cycle times based on results without considering walking times

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Problems | Traditional straight assembly line | | | | | U-shaped assembly line | | | | |
| 0% | 5% | 10% | 15% | 20% | 0% | 5% | 10% | 15% | 20% |
| 2 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 38.00  30.00  19.00  21.00  12.00  23.00  19.00  18.00  28.00  1.61  2.04  1.41  1.93 | 45.75  32.50  20.23  22.64  12.99  24.88  20.59  19.44  30.44  1.76  2.21  1.52  2.12 | 49.50  35.00  21.47  24.28  13.97  26.76  22.17  20.88  32.89  1.91  2.39  1.63  2.31 | 53.25  37.50  22.93  25.92  14.96  28.65  23.76  22.32  35.33  2.06  2.56  1.74  2.50 | 57.00  40.00  24.58  27.56  15.94  30.53  25.34  23.76  37.78  2.22  2.74  1.85  2.69 | 38.00  30.00  19.00  21.00  12.00  23.00  19.00  18.00  28.00  1.60  2.04  1.16  1.91 | 40.73  32.00  20.77  22.64  12.86  24.91  19.90  19.30  30.19  1.71  2.19  1.25  2.03 | 43.46  34.00  22.54  24.28  13.72  26.82  21.61  20.60  32.38  1.82  2.34  1.35  2.18 | 46.20  36.00  24.31  25.92  14.58  28.72  23.41  21.91  34.57  1.94  2.50  1.44  2.32 | 48.93  38.00  26.08  27.56  15.44  30.63  25.22  23.21  36.75  2.05  2.65  1.54  2.46 |
| 3 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 28.00  22.00  13.00  14.00  8.00  16.00  13.00  13.00  19.00  1.03  1.46  0.80  1.34 | 28.94  24.00  14.23  15.23  8.66  17.25  14.06  13.72  20.83  1.11  1.55  0.86  2.05 | 29.88  26.00  15.47  16.46  9.31  18.51  15.11  14.44  22.67  1.18  1.66  0.91  2.20 | 30.81  28.00  16.70  17.69  9.97  19.76  16.17  15.24  24.50  1.26  1.78  0.97  2.34 | 32.50  30.00  17.93  18.92  10.63  21.02  17.23  16.32  26.33  1.33  1.90  1.02  2.48 | 26.00  22.00  13.00  14.00  8.00  16.00  13.00  12.00  19.00  1.03  1.46  0.78  1.34 | 28.45  24.16  13.82  15.11  8.66  17.69  14.38  13.09  20.60  1.11  1.55  0.87  1.46 | 30.91  26.32  14.64  16.21  9.31  19.37  14.38  14.18  22.20  1.18  1.66  0.97  1.59 | 33.36  28.47  15.47  17.32  9.97  21.06  17.15  15.27  23.80  1.26  1.78  1.06  1.71 | 35.82  30.63  16.29  18.42  10.63  22.74  18.54  16.37  25.40  1.33  1.90  1.16  1.84 |
| 4 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 22.00  16.00  11.00  11.00  6.00  12.00  11.00  10.00  15.00  0.88  1.13  0.61  1.10 | 22.94  17.50  11.41  11.82  6.66  12.84  11.53  10.72  16.22  1.06  1.19  0.64  1.15 | 24.75  19.00  11.82  12.64  7.31  13.67  12.06  11.44  17.44  1.13  1.25  0.67  1.20 | 26.63  20.50  12.23  13.46  7.97  14.51  12.59  12.16  18.67  1.21  1.30  0.69  1.30 | 28.50  22.00  12.64  14.28  8.63  15.35  13.11  12.88  19.89  1.28  1.36  0.72  1.39 | 20.00  15.00  10.00  11.00  6.00  12.00  10.00  10.00  14.00  0.88  1.09  0.58  1.04 | 22.24  17.31  10.82  11.81  6.73  13.25  11.06  10.72  15.60  0.95  1.27  0.63  1.14 | 25.40  18.62  11.64  12.61  7.47  14.51  12.11  11.44  17.20  1.02  1.44  0.68  1.23 | 28.60  19.93  12.47  13.42  8.20  15.76  13.17  12.16  18.80  1.10  1.62  0.75  1.33 | 31.80  21.24  13.29  14.22  8.94  17.02  17.02  12.88  20.40  1.20  1.80  0.81  1.42 |
| #: number of workstation | | | | | | | | | | | |

In Table 6, there are 156 (13×4×3) cases that take walking time into consideration for both of traditional straight and U-shaped production lines. For traditional straight lines, there are 55 cases where performance improves when walking time is considered. For U-shaped production lines, 121 cases result shorter cycle time when walking time is considered. This means that the improvement of taking walking time into account is more obvious for U-shaped production lines than for traditional straight production lines. Moreover, longer walking times give more improvement on average for both traditional straight and U-shaped production lines.

**Table 6.** Improvement considering walking times (percentage)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Problems | Traditional straight assembly line | | | | U-shaped assembly line | | | |
| 5% | 10% | 15% | 20% | 5% | 10% | 15% | 20% |
| 2 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | 8.74  -  -  -  2.53  -  -  -  -  5.56  -  22.20  - | 8.08  -  -  -  4.70  -  -  -  -  7.10  -  21.78  - | 7.51  -  -  -  6.59  -  -  -  -  8.41  -  21.99  - | 7.02  -  -  -  6.27  -  -  -  -  9.54  -  22.17  0.37 | -  -  3.33  2.36  1.58  1.58  -  -  0.25  5.09  -  5.70  - | -  -  4.75  2.98  2.96  2.94  2.03  -  -  5.29  0.32  5.59  - | -  -  6.61  4.19  4.18  3.34  2.81  0.95  -  4.90  0.71  6.10  - | -  -  8.22  9.89  5.26  5.26  3.47  1.91  1.14  4.55  0.42  6.56  - |
| average | 3.00 | 3.20 | 3.42 | 3.49 | 1.53 | 2.07 | 2.60 | 3.59 |
| 3 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | -  2.08  2.89  -  -  -  3.76  -  -  -  -  -  30.00 | -  3.85  5.32  -  -  -  6.99  -  0.98  -  -  -  30.45 | -  5.36  7.39  1.30  -  -  9.81  -  3.40  -  -  -  30.56 | -  6.67  9.17  3.38  2.96  -  11.61  -  5.49  -  -  -  30.65 | 5.33  7.14  0.87  -  3.80  5.50  5.59  0.21  -  -  -  2.60  2.15 | 6.47  11.02  -  1.00  3.45  6.95  2.27  0.39  0.51  0.01  -  6.77  4.57 | 7.65  11.77  -  1.40  -  8.43  14.96  1.70  0.56  0.42  -  9.41  6.09 | 9.26  12.41  -  1.97  2.96  10.40  17.84  3.64  2.01  0.77  -  17.00  7.91 |
| average | 2.98 | 3.66 | 4.45 | 5.38 | 2.58 | 3.34 | 4.80 | 6.63 |
| 4 | Bowman [38]  Chen and Plebani [39]  Hwang and Katayama [33] 1  Hwang and Katayama [33] 2  Hwang and Katayama [33] 3  Hwang and Katayama [33] 4  Avikal et al. [2]  Li et al. [3]  Oksuz et al. [30]  Kuo and Liu [27] 1  Kuo and Liu [27] 2  Kuo et al. [28] 1  Kuo et al. [28] 2 | -  2.86  -  -  -  -  -  -  -  13.06  -  -  - | -  5.26  -  -  -  -  -  -  -  15.53  -  -  - | -  7.32  -  1.71  -  -  -  0.66  1.79  16.57  -  -  2.51 | -  9.09  -  4.48  3.64  -  -  -  -  17.26  -  0.07  5.76 | -  6.46  -  3.35  2.58  6.81  4.78  10.40  -  2.95  6.17  3.22  - | 6.01  6.48  -  -  4.65  6.40  1.03  0.47  4.39  6.44  13.61  3.81  2.44 | 13.25  9.27  1.87  1.40  6.36  12.21  4.45  2.73  6.03  9.11  19.43  7.42  4.72 | 19.03  8.79  4.85  6.64  7.78  12.80  22.94  3.42  7.41  11.29  24.10  11.55  7.75 |
| average | 1.22 | 1.60 | 2.35 | 3.10 | 3.59 | 4.29 | 7.56 | 11.41 |

The average computation times for different problems and number of workstations are illustrated in Figure 9. The computation times are highly dependent on the number of workstations. In every problem, more workstations led to longer computation times. As shown in Table 2, the number of tasks in problems is greatest in the cases reported by Chen and Plebani [39], Hwang and Katayama [33]2, Hwang and Katayama [33]4 and Li et al. [3]. The computation times for these 4 problems are longer than other problems. In particular, when there are 4 workstations, computation times exceed 1000 seconds.

**Figure 9.** Computation time

5. Discussion

The benefit of using U-shaped production lines has been demonstrated both in industry and in research. This research deals with the problem of U-shaped production line design and aims to minimize cycle times when given the number of operators. An integer programming model was developed to optimize the assignment of tasks to locations and locations to workstations. Unlike other related studies, the proposed integer programming takes both precedence relationships of tasks and walking times between locations into consideration. The proposed integer programming was then validated on 13 problems that have been reported in the literature. As the information about walking times was not provided in these problems, it was assumed that walking times were a proportion of average production times. The experimental results confirm U-shaped production lines can reduce cycle times when compared with traditional straight production lines. The results also show that when walking times are taken into consideration for designing production lines, the advantages of U-shaped production lines are more obvious, especially when walking times are longer.

Computation time increases when the number of workstations or tasks are increases. In order to solve the problems with more tasks or more workstations, an efficient heuristic, such as Genetic Algorithms, Tabu Search or Simulated Annealing, might be proposed, and this may be an opportunity for future research.

**Author Contributions:** Conceptualization, Y.K. and T.Y.; methodology, Y.K.; software, Y.K.; validation, Y.K., T.Y. and T.-L.H.; formal analysis, Y.K. and T.-L.H.; investigation, Y.K.; resources, Y.K.; data curation, Y.K. and T.-L.H.; writing—original draft preparation, Y.K. and T.Y; writing—review and editing, Y.K. and T.Y.; visualization, Y.K.; supervision, Y.K. and T.Y.; project administration, Y.K.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments**: This work was supported in part by the Ministry of Science and Technology of Taiwan, Republic of China, under grant MOST-109-2221-E-006 -152 -MY3.

**Conflicts of Interest:** The authors declare no conflict of interest.

References

1. Aydogan, E.K.; Delice, Y.; Ozean, U.; Gencer, C.; Ozkan, B. Balancing stochastic U-lines using particle swarm optimization, *Journal of Intelligent Manufacturing* 2019, 30(1), 97-111.
2. Avikal, S.; Jain, R.; Mishra, P.K.; Yadav, H.C., A heuristic approach for U-shaped assembly line balancing to improve labor productivity. *Computers and Industrial Engineering* **2013**, 64(4), 895-901.
3. Li, M.; Tang, Q.; Zheng, Q.; Xia, X.; Floudas, C.A. Rules-based heuristic approach for the U-shaped assembly line balancing problem, *Applied Mathematical Modelling* **2017**, 48, 423-439.
4. Jayaswal, S.; Agarwal, P. Balancing U-shaped assembly lines with resource dependent task times: A Simulated Annealing approach. *Journal of Manufacturing Systems* **2014**, 33(4), 522-534.
5. Hazir, O.; Dolgui, A. A decomposition based solution algorithm for U-type assembly line balancing with interval data. *Computers and Operations Research* **2015**, 59, 123-131.
6. Scholl, A., and Klein, R. ULINO: Optimally balancing u-shaped JIT assembly lines. *International Journal of Production Research* **1999**, 37(4), 721-736.
7. Miltenburg J. The effect of breakdowns on U-shaped production lines. *International Journal of Production Research* **2000**, 38(2), 353-364.
8. Miltenburg J. U-shaped production lines: a review of theory and practice. *International Journal of Production Economics* **2001**, 70(3), 201-214.
9. Cheng, C.; Miltenburg, J.; Motwani, J. The effect of straight- and U-shaped lines on quality. *IEEE Transactions on Engineering Management* **2000**, 47(3), 321-334.
10. Aase, G.R., Olson, J.R. and Schniederjans, M.J. U-shaped assembly line layouts and their impact on labor productivity: an experimental study. *European Journal of Operational Research* **2004**, 156(3), 698–711.
11. Simaria, A.S.; Zanella de Sá, M.; Vilarinho, P.M. Meeting demand variation using flexible U-shaped assembly lines. *International Journal of Production Research* **2009**, 47(14), 3937-3955.
12. Pilati, F.; Ferrari, E.; Gamberi, M.; Margelli, S. Multi-manned assembly line balancing: workforce synchronization for big data sets through simulated annealing. *Applied Sciences* **2021**, 11, 2523.
13. Mukund Nilakantan, J.; Ponnambalam, S.G. Robotic U-shaped assembly line balancing using particle swarm optimization. *Engineering Optimization* **2016**, 48(2), 231-252.
14. Shewchuk, J.P. Worker allocation in lean U-shaped production lines. *International Journal of Production Research* **2008**, 46(13), 3485-3502.
15. Miqueo, A.; Torralba, M.; Yague-Fabra, J.A. Lean manual assembly 4.0: a systematic review, *Applied Science* **2020**, 10, 8555.
16. Zulch, M.; Zulch, G. Production logistics and ergonomic evaluation of U-shaped assembly systems. *International Journal of Production Economics* **2017**, 190, 37-44.
17. Ohno, k.; Nakade, K. Analysis and optimization of a U-shaped production line. *Journal of the Operations Research Society of Japan* **1997**, 40(1), 90-104.
18. Scholl, A.; Becker, C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research* **2006**, 168(3), 666-693.
19. Suer G.A. Optimal operator assignment and cell loading in labor-intensive manufacturing cells. *Computers and Industrial Engineering* **1996**, 31(1/2), 155-158.
20. Suer, G.A. Designing parallel assembly lines. *Computers and Industrial Engineering* **1998**, 48(3-4), 467-470.
21. Suer G.A.; Bera, I.S. Optimal operator assignment and cell loading when lot-splitting is allowed. *Computers and Industrial Engineering* **1998**, 48(3-4), 431-434.
22. Suer, G.A. and Dagli, C. Intra-cell manpower transfers and cell loading in labor-intensive manufacturing cells. *Computers and Industrial Engineering* **2005**, 48(3), 643-655.
23. Kuo, Y.; Yang, T. A case study on the operator allocation decision for TFT-LCD inspection and packaging process. *International Journal of Advanced Manufacturing Technology* **2006**, 17(3), 363-375.
24. Suer, G.A.; Tummaluri, R.R. Multi-period operator assignment considering skills, learning and forgetting in labour-intensive cells. *International Journal of Production Research* **2008**, 46(2), 469-493.
25. Suer G.A.; Cosner, J.; Patten, A. Models for cell loading and product sequencing in labor-intensive cells. *Computers and Industrial Engineering* **2009**, 56(1), 79-105.
26. Suer, G.A.; Kamat, K.; Mese, E.; Huang, J. Minimizing total tardiness subject to manpower restriction in labor-intensive manufacturing cells. *Mathematical and Computer Modelling* **2013**, 57(3-4), 741-753.
27. Kuo, Y.; Liu, C.C. Operator assignment in a labor-intensive manufacturing cell considering inter-cell manpower transfer. *Computers and Industrial Engineering* **2017**, 110, 83-91.
28. Kuo, Y.; Chen, Y.P.; Wang, Y.C. Operator assignment with cell loading and product sequencing in labour-intensive assembly cells – a case study of a bicycle assembly company. *International Journal of Production Research* **2018**, 56(16), 5495-5510.
29. Miltenburg, G.J.; Wijngaard, J. The U-Line line balancing problem. *Management Science* **1994**, 40(10), 1378-1388.
30. Oksuz, M.K.; Buyukozkan, K.; Satoglu S.I. U-shaped assembly line worker assignment and balancing problem: A mathematical model and two meta-heuristics. *Computers and Industrial Engineering* **2017**, 112, 246-263.
31. Zhang, Z.; Tang, Q.; Ruiz, R.; Zhang, L. Ergonomic risk and cycle time minimization for the U-shaped worker assignment assembly line balancing problem: A multi-objective approach. *Computers and Operations Research* **2020**, 118, 104905.
32. Rabbani, M.; Kazemi, S.M.; Manavizadeh, N. Mixed model U-line balancing type-1 problem: A new approach. *Journal of Manufacturing Systems* **2012**, 31(2), 131-138.
33. Hwang R.; Katayama, H. A multi-decision genetic approach for workload balancing of mixed-model U-shaped assembly line systems. *International Journal of Production Research* **2009**, 47(14), 3797-3822.
34. Nourmohammadi, A.; Zandieh, M.; Tavakkoli-Moghaddam, R. An imperialist competitive algorithm for multi-objective U-type assembly line design. *Journal of Computational Science* 2013, 4(5), 393-400.
35. Sahin, M.; Kellegoz, T. An efficient grouping genetic algorithm for U-shaped assembly line balancing problems with maximizing production rate. *Memetic Computing* **2017**, 9(3), 213-229.
36. Zhang, Z.; Tang, Q.; Han, D.; Li, Z. Enhanced migrating birds optimization algorithm for U-shaped assembly line balancing problems with workers assignment. *Neural Computing and Applications* **2019**, 31(11),7501-7515.
37. Nakade, K.; Ohno, K. An optimal walker allocation problem for a U-shaped production line. *International Journal Production Economics* **1999**, 60-61, 353-358.
38. Bowman, E. H. Assembly line balancing by linear programming. *Operations Research* **1960**, 8(3), 385–389.
39. Chen, S.; Plebani, L. Heuristic for balancing U-shaped assembly lines with parallel stations. *Journal of the Operations Research Society of Japan* **2008**, 51(1), 1-14.