Chapter 79
Balancing Serial U-Lines in Lean Production

Jing Zha, Hao-ping Li, and Xiang-feng Zeng

Abstract In lean production, multiple U-lines are always combined to eliminate waste, which is called large room effect. The paper discusses three types of serial U-lines balancing including tail-to-head type, head-to-head type and hybrid type. A goal programming model which considered the minimization of workstations and the first or last workstation's operation time is established in each situation. Example applications indicate that serial U-lines with tail-to-head type are more efficient to reduce manpower and serial U-lines with head-to-head type are more convenient of material flow. The hybrid way owns the above two type's advantages and demonstrates its effectiveness.

Keywords Serial U-lines • U-line balancing • Lean production

Introduction

In order to meet the varying demand flexibly and eliminate waste, lean production requires proper design of machinery and workstation layout, such as serial U-lines.

In a U-line, the input of material and the output of products are close to each other, forming a "U", and operators walk to perform combinations of tasks in two sides. U-line balancing problem (ULBP), which belongs to physical design of the U-line, is the problem of assigning tasks to a minimum number of workstations under the restriction of precedence relationship and cycle time (Boysen et al. 2007). Because there are more possibilities for grouping tasks into stations in a U-line, the number of stations required in a U-line is never more than that required in a straight line. More important, when the demand is changing, it is easy to wide or narrow the

J. Zha (✉) • H.-p. Li • X.-f. Zeng
College of Mechanical and Material Engineering, China Three Gorges University, Hubei Yichang, China
e-mail: maria_zhajing@hotmail.com

R. Dou (ed.), *Proceedings of 2012 3rd International Asia Conference on Industrial Engineering and Management Innovation(IEMI2012)*,
DOI 10.1007/978-3-642-33012-4_79,

group of tasks assigned to each operator in a U-line to increase or decrease the number of operators. This principle is called Shojinka in Japanese (Monden 1983).

Furthermore, the design of the single line like an islet is avoided in lean production. In contrast, it prefers to concentrate idle time in a single workstation when balancing a single U-line and build multiline workstation among the neighboring serial lines. The number of operators or workstations can be decreased in this way. According to the location of multiline workstation, serial U-lines can be divided three types: tail-to-head type, head-to-head type and hybrid type.

The paper discusses three types of serial ULBP and the rest of the paper is organized as follows. The next sections are about three versions of serial ULBP respectively. A goal programming model is established and example calculation is utilized in each section. Discussion and conclusions are given in the last section.

Literature Review

Miltenburg and Wijngaard (1994), that is the first study on U-line balancing in the literature, formulated dynamic programming model to solve ULBP. Urban (1998) proposed the first integer programming formulation for ULBP by establishing a "phantom" network attached to the original precedence graph. Scholl and Klein (1999) defined three version problems with respect to the precedence constraints: (79.1) ULBP-1. Given the cycle time, minimize the number of workstation; (79.2) ULBP-2. Given the number of workstation, minimize the cycle time; (79.3) ULBP-E. Maximize the line efficiency for cycle time and workstation being variable. U-LINO, a kind of depth-first branch and brand procedure are developed to solve the three problems. Deshpande et al. (2002) developed an integer programming formulation and developed a solution procedure based on Lagrangian relaxation to address ULBP. Gokcen et al. (2005) put forward a shortest route formulation of ULBP. Based on the integer programming formulation developed by Urban (1998), Gokcen and Agpak (2006) presented a goal programming model which considers several conflicting goals simultaneously and provides a considerable amount of flexibility to the decision maker. In real world, it is not easy to determine goal values precisely but generally defined these values such as "somewhat larger than", "substantially lesser than" or "around" the vague goal. Kara et al. (2009) proposed a binary fuzzy goal programming model to solve these problem. In order to solve large-scale ULBP, Baykasoglu (2007), (Sabuncuoglu and Ozbakir 2007) and Hwang et al. (2008) applied genetic algorithm. Sabuncuoglu et al. (2009) and Baykasoglu and Dereli (2009) modified ant colony optimization. All above literatures solve single U-line balancing problem.

Miltenburg (1998) presented a reaching dynamic programming algorithm to balance U-lines in a multiple U-line facility and only considered serial ULBP with tail-to-head type. Chiang et al. (2007) formulized three version multiple ULBP which were similar to Scholl (1999). Multiple U-lines in that paper were parallel lines and executed the same tasks to produce the same products. Therefore, it still needs to exploit serial ULBP systematically.

Serial U-Lines with Tail-to-Head Type

In this type, the last workstation of the former U-line and the first workstation of the next U-line are close to each other and combined as multiline workstation which is operated by one operator. Suppose that serial U-lines have the same cycle time. At first, balancing the first U-line to minimize the number of workstation and concentrate the idle time in the last workstation. Next, balancing the second U-line with the same objectives under the condition the first workstation has been assigned tasks of the last workstation in the first U-line. And then balancing the next U-line with the same method until all U-lines are balanced. In the end, the number of workstations in all serial U-lines is minimized.

Based on above analysis, goal programming model I of U-line l balancing is developed as follows.

$$Min \quad P_1({d_{ls}}^+), P_2({d_{lt}}^+) \tag{79.1}$$

$$s.t. \quad \sum_{k=1}^{m_{\max}} S_k - {d_{ls}}^+ = m_{\min}, {d_{ls}}^+ \geq 0 \text{ and integer} \tag{79.2}$$

$$\sum_{i=1}^{n} t_i \left(x_{i(m_{\min}+{d_{ls}}^+)} + y_{i(m_{\min}+{d_{ls}}^+)} \right) - {d_{lt}}^+ = 0,\ {d_{lt}}^+ > 0 \tag{79.3}$$

$$\sum_{i=1}^{n} t_i (x_{i1} + y_{i1}) \leq C - {d_{(l-1)t}}^+ \tag{79.4}$$

$$\sum_{i=1}^{n} t_i (x_{ik} + y_{ik}) \leq C \cdot S_k,\ k = 2, \ldots, m_{\max} \tag{79.5}$$

$$\sum_{k=FE_i}^{m_{\max}} x_{ik} + \sum_{k=BE_i}^{m_{\max}} y_{ik} = 1, \quad i = 1, 2, \ldots, n \tag{79.6}$$

$$S_{k+1} \leq S_k,\ k = m_{\min}, \ldots, m_{\max} - 1 \tag{79.7}$$

$$\sum_{k=FE_i}^{m_{\max}} (m_{\max} - k + 1) x_{ik} \leq \sum_{k=FE_j}^{m_{\max}} (m_{\max} - k + 1) x_{jk},\ \forall (i,j) \in G \tag{79.8}$$

$$\sum_{k=BE_i}^{m_{\max}} (m_{\max} - k + 1) y_{ik} \geq \sum_{k=BE_j}^{m_{\max}} (m_{\max} - k + 1) y_{jk},\ \forall (i,j) \in G \tag{79.9}$$

$$x_{ik}, y_{ik}, S_k \in \{0, 1\},\ \forall i, j, k \tag{79.10}$$

Notation: C – cycle time; t_i – operating time of task i; FE_i – the earliest workstation which task i can be assigned in the front side; BE_i – the earliest workstation which task i can be assigned in the back side; $\forall(i,j) \in G$ – in precedence graph, task j can be started only after task i is finished; $d_{ls}{}^{+}$–position deviation of the actual U-line l workstation number from lower bound of workstation number; $d_{lt}{}^{+}$ – operating time of the last workstation in U-line l; m – workstation number. $m_{\min}$ – Lower bound of workstation number. $m_{\max}$ – Upper bound of workstation number. $m \in [m_{\min}, m_{\max}]$;

$$x_{ik} = \begin{cases} 1 & \text{if task } i \text{ is assigned to the front side of workstation } k \\ 0 & \text{otherwise} \end{cases};$$

$$y_{ik} = \begin{cases} 1 & \text{if task } i \text{ is assigned to the back side of workstation } k \\ 0 & \text{otherwise} \end{cases};$$

$$S_k = \begin{cases} 1 & \text{if workstation } k \text{ is utilized} \\ 0 & \text{otherwise} \end{cases};$$

Objective P1 is to minimize the workstation number of U-line l. Objective P2 is to minimize the last workstation's operating time. P1 is superior to P2. Constraint (79.2) and (79.3) are soft constraints in goal programming. Constraint (79.4) restricts the first workstation's operating time no more than cycle time subtracting the last workstation's operating time of U-line (l-1). Constraint (79.5) restricts other workstations' operation time no more than cycle time. Constraint (79.6) means that every task should been assigned to one and only one workstation. Constraint (79.7) requires to assign the workstation with little index k first. Constraint (79.8) and (79.9) enforce the precedence relationships of tasks. If the task is assigned to the front side, constraint (79.8) should be ensured. If the task is assigned to the back side, constraint (79.9) should be ensured.

Next an example is solved to illustrate the serial U-lines with tail-to-head type discussed here. The precedence graph of U-line A and B are given in Figs. 79.1 and 79.2. The goal programming model is solved using MATLAB 7.3. Cycle time = 20. If separately balanced, U-line A needs 3 workstations and U-line B needs 3 workstations. If U-line A and B are located tail-to-head, task assignments are shown in Table 79.1 and line layout is shown as Fig. 79.3. After combined, U-line A and B needs 5 workstations or operators and one operator are reduced.

Serial U-Lines with Head-to-Head Type

However, from Fig. 79.3, we can see that there is complicated path of material flow because U-line A's output is far way from U-line B's input and it leads to wastes of material handling. Head-to-head type is an alternative layout. Multiple U-lines are located head-to-head and all input and output are close to each other. The output

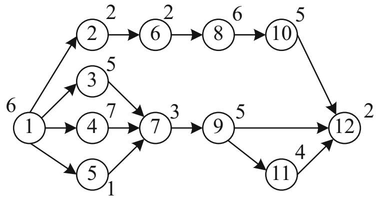


Fig. 79.1 Precedence graph of U-line A

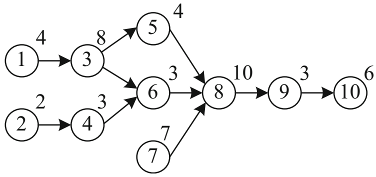


Fig. 79.2 Precedence graph of U-line B

Table 79.1 Task assignments

No of workstation	U-line	Front side	Back side	Time	Workstation time
1	A	1,2,5	12,11,10	20	20
2		3,4	9,7	20	20
3		6	8	8	20
	B	1,2	10	12	
4		–	9,8,7	20	20
5		2,3,4,5	6	20	20

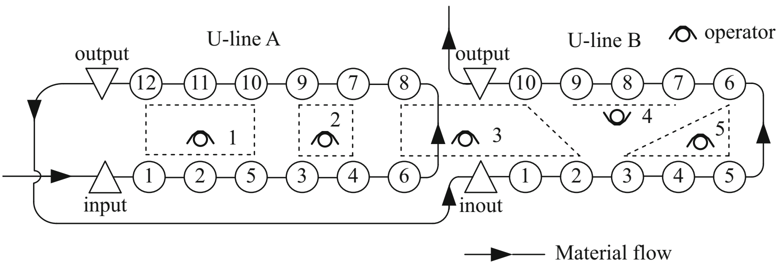


Fig. 79.3 Serial U-Lines layout with tail-to-head type

semi-finished product of the U-line can be transferred to the next U-line directly. Useless material handling is reduced and production rate is enhanced. Moreover, one operator can handle tasks in the input and output of multiple U-lines. It helps to achieve pull production. The operator is easy to perceive the unbalance between multiple U-lines and then trigger the improving actions.

The characteristic of this type is that the first workstations of multiple U-lines are combined as a multiline workstation. At first, balancing single U-line to minimize the number of workstation and concentrate the idle time in the first workstation. Next, assigning tasks in the first workstations of multiple U-lines with the

Table 79.2 Task assignments

	U-line A			U-line B		
No of workstation	Front	Back	Time	Front	Back	Time
1	1	12	8	1	10	10
2	2,3,5	11,9,7	20	–	9,8,7	20
3	4,6	10,8	20	2,3,4,5	6	20

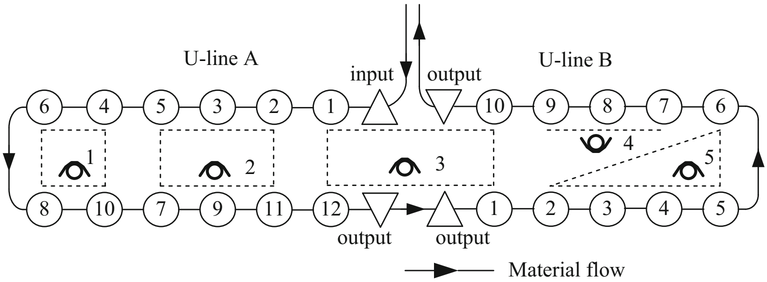


Fig. 79.4 Serial U-lines layout with head-to-head type

restriction of cycle time and determining multiline workstation's number. And then the number of workstations in all serial U-lines is minimized.

Goal programming model II of U-line l balancing is developed as follows.

$$Min \quad P_1(d_{ls}{}^+),\ P_2(d_{lf}{}^+) \tag{79.11}$$

$$s.t. \quad \sum_{i=1}^{n} t_i(x_{i1} + y_{i1}) - d_{lf}{}^+ = 0,\ d_{lf}{}^+ > 0 \tag{79.12}$$

$$\sum_{i=1}^{n} t_i(x_{ik} + y_{ik}) \le C \cdot S_k \quad k = 1, \ldots, m_{\max} \tag{79.13}$$

The rest restrictions are the same as (79.2), (79.6), (79.7), (79.8), (79.9), and (79.10).

Notation: $d_{lf}{}^+$ – operating time of the first workstation in U-line l;

If U-line A (see Fig. 79.1) and U-line B (see Fig. 79.2) are located head-to-head, task assignments are shown in Table 79.2 and line layout is shown in Fig. 79.4. Cycle time = 20. After combined, line A and B also needs 5 workstations or operators and the material flow is more simple and fluent.

Serial U-Lines with Hybrid Type

With the restriction of production environment such as factory building, it is not possible to use tail-to-head type or head-to-head type simply, but use the hybrid of two types flexibly. That is called large room effect (Boysen et al. 2007). The

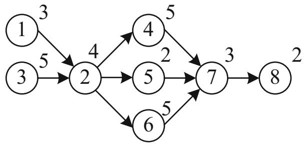


Fig. 79.5 Precedence graph of U-line C

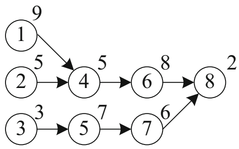


Fig. 79.6 Precedence graph of U-line D

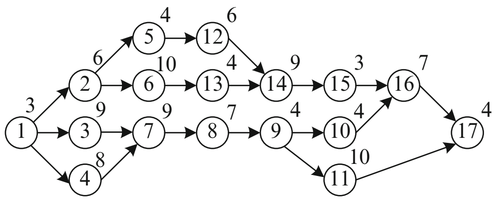


Fig. 79.7 Precedence graph of U-line E

characteristic of this type is that the large building is divided as multiple sectors by aisles. Each U-line is located at both sides of aisles. Some U-lines are tail-to-head and some U-lines are head-to-head in order to ensure fluent material flow and high production efficiency.

In a factory, there are U-line C, D and E besides A and B. Figs. 79.5, 79.6 and 79.7 are the precedence graphs. Cycle time = 20.

If separately balanced, U-line A needs 3 workstations; U-line B needs 3 workstations; U-line C needs 2 workstations; U-line D needs 3 workstations and U-line E needs 6 workstations. Totally needs 17 workstations. If U-lines are combined in head-to-head type, the first workstation's operating time of U-line A is 8; the first workstation's operating time of U-line B is 10; the first workstation's operating time of U-line C is 9; the first workstation's operating time of U-line D is 5 and the first workstation's operating time of U-line E is 7. With the restriction of cycle time, 3 multiline workstations are required. Totally needs 15 workstations or operators and 2 operators are reduced.

Supposed the locations of U-lines in the building are given. U-line A and B are at the left side of aisle 1; U-line C and D are between aisle 1 and 2 and U-line E is at the right side of aisle 2. Firstly, U-line A and B are balanced using goal programming model II and the first workstations' operating time is 8 (t_A) and 10 (t_B). ${d_{(l-1)t}}^{+} = 8 + 10 = 18$. U-line C are balanced using goal programming I and the last workstation's operating time is 7 (t_C). And then U-line D and E are balanced using goal programming II and the first workstations' operating time is 5 (t_D) and 7 (t_E). A multiline workstation can cover t_C, t_D and t_E. Hybrid U-lines layout is shown in Fig. 79.8. Only 2 multiline workstations are required. Totally needs 14 workstations and 3 operators are reduced.

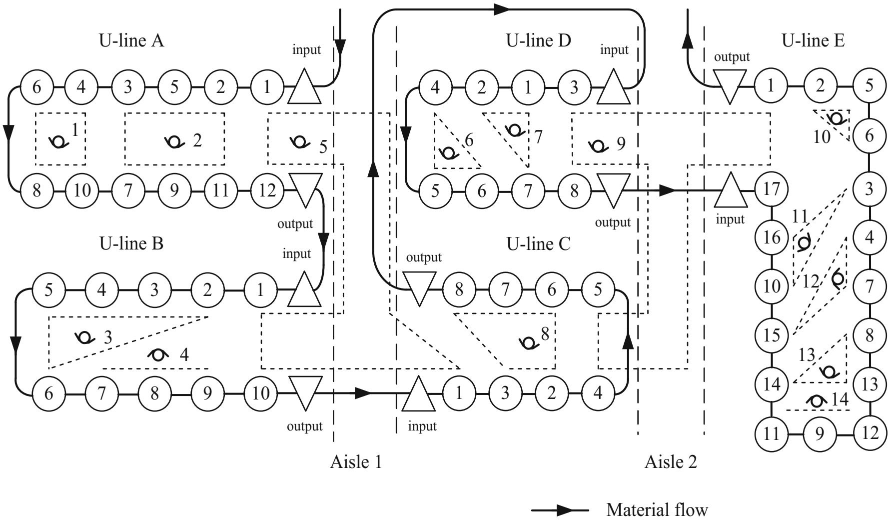


Fig. 79.8 Serail U-lines layout with hybrid type

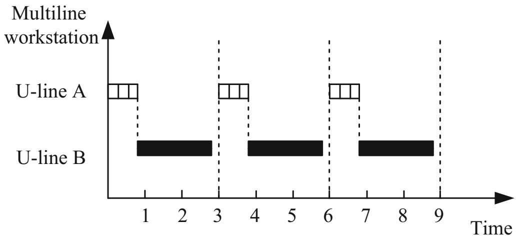
Material flow is more fluent than one with tail-to-head type and the number of workstations is two less than one with head-to-head type. The hybrid way owns the above two type's advantages.

Discussion and Conclusion

One hypothesis is that the cycle time of all U-lines is same. If all U-lines are part of the same product's manufacturing process, the hypothesis is feasible. There are two other cases in manufacturing. The arrangement of multiline workstations in the two cases is discussed as follows.

1. Semi-finished products manufactured by U-lines are parts of the same end product. The cycle times of all U-lines are proportional. For example, one pieces A and two piece B are assembled to C. $C_A = 2C_B$. Suppose $C_B = x \times C_A$, x > 1 and integer. In multiline workstation, the operating time of tasks from U-line A is t_A and the operating time of tasks from U-line B is t_B. If $\frac{t_A}{C_A} + \frac{t_B}{C_B} \leq 1$, one multiline workstation is feasible and the cycle time of multiline workstation is C_B. During a cycle, the time which an operator works in U-line A is $x \times t_A$; the time which works in U-line B is t_B and the idle time is $C_B - x \times t_A - t_B$. Given $C_A = 1$, x $= 3$, $t_A = 0.3$, $t_B = 2$. The working time sequence is showed as Fig. 79.9.

Fig. 79.9 Working time sequence of multiline workstation



2. The cycle times of all U-lines are different and not proportional. $C_B = x \times C_A$, $x > 1$ and not integer. If $\frac{t_A}{C_A} + \frac{t_B}{C_B} \leq 1$, one multiline workstation is feasible. If the cycle time of multiline workstation is C_A, an operator must return to U-line B every x cycle. Because x is not an integer, the operator can not return to U-line B in time. In that situation, a buffer should be set in U-line B to aviod line down. The operator needs to monitor the buffer and decide to work on which line based on production situation.

Therefore, it recommends to combine multiple U-lines with the same or proportional cycle time in serial. Among the three combination type, serial U-lines with tail-to-head type are more efficient to reduce manpower and serial U-lines with head-to-head type are more convenient of material flow. With the restriction of production environment such as factory building, it is not possible to use tail-to-head type or head-to-head type simply, but use the hybird of two types flexibly. The hybrid way owns the above two type's advantages and is recommended.

Acknowledgment This work is partially supported by National Science Foundation of China Grand (No. 71071059) and the scientific research fund of personnel from China Three Gorges University. (No. KJ2011B035).

References

Baykasoglu A, Dereli T (2009) Simple and U-type assembly line balancing by using an ant colony based algorithm [J]. Math Comput Appl 14(1):1–12

Boysen N, Fliedner M, Scholl A (2007) A classification of assembly line balancing problems [J]. Eur J Operat Res 183:674–693

Chiang WC, Kouvelis P, Urban TL (2007) Line balancing in a just-in-time production environment: balancing multiple U-lines [J]. IIE Trans 39:347–359

Deshpande V, Rajaram K, Guignard M (2002) An integer programming based formulation of the U-line balancing problem [R]. Working paper, The Wharton School of the University of Pennsylvania, Philadelphia, 5

Gokcen H, Agpak K (2006) A goal programming approach to simple U-line balancing problem [J]. Eur J Operat Res 171(2):577–585

Gokcen H, Agpak K, Gencer C, Kizilkaya E (2005) A shortest route formulation of simple U-type assembly line balancing problem [J]. Appl Math Model 29:373–380

Hwang RK, Katayama H, Gen M (2008) U-shaped assembly line balancing problem with genetic algorithm [J]. Int J Prod Res 46(10):4637–4649

Kara Y, Paksoy T, Chang CT (2009) Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing [J]. Eur J Operat Res 195(2):335–347

Miltenburg J (1998) Balancing U-lines in a multiple U-line facility [J]. Eur J Operat Res 109 (1):1–23

Miltenburg GJ, Wijngaard J (1994) The U-line line balancing problem [J]. Manage Sci 40 (10):1378–1388

Monden Y (1983) Toyota production system [M]. Industrial Engineering Press, Norcross

Sabuncuoglu I, Ozbakir L (2007) Stochastic U-line balancing using genetic algorithms [J]. Int J Adv Manuf Technol 32:139–147

Sabuncuoglu I, Erel E, Alp A (2009) Ant colony optimization for the single model U-type assembly line balancing problem [J]. Int J Prod Econ 120(2):287–3002

Scholl A, Klein R (1999) ULINO: optimally balancing U-shaped JIT assembly lines [J]. Int J Prod Res 37(4):721–736

Urban TL (1998) Note. Optimal balancing of U-shaped assembly lines [J]. Manage Sci 44 (5):738–741