Jurnal Teknik Industri, Vol. 11, No. 1, Juni 2009, pp. 33-42
ISSN 1411-2485

MULTI OBJECTIVE MODEL FOR BALANCING U-TYPE ASSEMBLY LINE WITH PERMANENT AND TEMPORARY WORKERS

Gede Agus Widyadana
Department of Industrial and System Engineering, Chung Yuan Christian University
Chung Li 320, Taiwan
Faculty of Industrial Technology, Department of Industrial Engineering, Petra Christian University
Jl. Siwalankerto 121-131 Surabaya 60236, Indonesia
Email: gede@petra.ac.id

ABSTRACT

Assembly line balancing is one of the most important management decisions for reducing production costs. So far, many researches have been done in this area. In this research, we expand the U-type line-balancing using goal programming for multi objective model with two goals, i.e., minimized the number of temporary workers and number of cycle time in each station. Additionally, an example to illustrate the model, as well as some analyses is presented. Different amount of time for temporary worker to accomplish their tasks were generated. For some cases, the U-type assembly line is more efficient than the straight one. However, the goal 1 and goal 2 are conflicted

Keywords: line balancing, U-type line, permanent and temporary workers, goal programming

1. INTRODUCTION

As a medium term decision, line balancing is applied when management want to have an efficient production line. An assembly line consists of some workstations. Every workstation does some consecutive tasks in a determined cycle time and process flow restrictions. The cycle time is determined by customers, so it can vary among different production planning periods. When the cycle time is changed, the number of workstations may change either. These changes force the management to hire temporary workers to fulfill the demand, especially in a peak demand period. Usually, temporary workers are less skillful so as need longer processing time compared to permanent workers. Thus a good line balancing model of temporary and permanent workers plays an important role to get a more efficient production line.

The U-line is a production line with U-shape layout. One uniqueness of U-line balancing problem is that one workstation can do task which is not in sequence. In reality, U-shapes type has been utilized in many production lines. Many researches in the U-type line balancing problem have been developed continuously in recent years. Miltenburg (1998) developed dynamic programming model to solve U-line balancing problem. In his problem, more than one U-line assembly lines in one production line are considered. He found an optimal solution when individual U-lines do not have more than 22 tasks and do not have wide, sparse precedence graphs. A heuristic method to solve U-shape line balancing problem was developed by Nakade and Ohno (1999). They assumed workers have to be allocated to machines so that the walking routes of workers are not crossed. In their model, waiting times and walking times between machines are considered and assumed deterministic. Kara *et al.* (2007) studied the mixed model U-lines utilized in just-in-time (JIT). Mixed-model production is a production environment where more than one product with similar production characteristic is produced on the same line. They solved the problem using simulated annealing method. Nakade and Nishiwaki (2008) developed an algorithm to solve U-shaped production line problem with different workers skill. They assumed processing, operation and walking times are deterministic and different among workers. Learning effect in simple assembly line balancing problem and U-type line balancing problem was studied by Toksari *et al.* (2008). They found that more computational effort is needed to solve line balancing with learning effect than without learning effect problem.

Some researchers have focused on developing model for multi objective U-line assembling problem. Baykasoglu (2005) proposed multi objective U-line model by integrating COMSOAL method, task assignment heuristic and simulated algorithm to solve simple line and U-type assembly line balancing problem. He developed an algorithm to find optimal solutions in a short computational time. A goal programming model for simple U-line balancing problem was developed by Gokcen and Akpag (2006). They concluded that goal programming approach is a realistic and natural approach for solving U-line balancing problem. Their model increased flexibility to decision maker in evaluating different alternatives. Kara *et al.* (2009) developed models for balancing straight and U-shaped assembly lines with fuzzy goals. Ozcan and Toklu (2009) developed a mathematical model, a pre-emptive goal programming model for precise goals and a fuzzy goal programming model. They applied their model for solving two-sided assembly line balancing.

Corominas *et al.* (2008) presented method of rebalancing line at a motorcycle company. The company had to hire temporary workers due to customer demand and they wanted to minimize number of temporary worker that will be hired. They developed a mixed integer programming model and solved it using ILOG CPLEX. In addition to minimization temporary workers, there are also many goals should be considered such as minimizing quality deviation because of the skill of temporary workers. So, it is reasonable in some situation to consider some goals instead of one optimizing goal. However computation time of mixed integer programming method sometime is not reasonable, so heuristic method is preferred. The comparison of performance and computation time of mixed integer programming and heuristic method for scheduling problem was compared by Widyadana (2001). Many researchers have developed heuristic methods to solve assembly line balancing problems. Some of them are Agrawal (1985), Shin and Min (1991) and Erel *et al.* (2005).

The literature search above shows there are no researchers developed a U-line balancing model considering multi objectives and temporary workers. In this study, goal programming models for U-line and straight line balancing problem with permanent and temporary workers are developed. The first goal is minimizing deviation of workers before and after temporary workers is added. The second goal is minimizing quality deviation. Quality deviation can occur if skill of temporary workers is not similar than permanent workers. As the skill of permanent workers is better than the skill of the temporary workers, the permanent workers can be assigned to more tasks than the temporary workers. The third goals is minimizing the deviation of the workstation cycle time which is determined by customer demand. When goal programming model has been developed then the model will be solved using LINGO mathematical programming language.

The organization of this paper is as follows. In the first section, background of the study and some literatures review are presented. The developing goal programming model for U-line and straight line balancing are explained briefly in Section 2. Some illustrative examples will be shown to illustrate how the model works and the trade offs among the goals in Section 3 and in the last section, some conclusions and future researches recommendation are derived.

2. GOAL PROGRAMMING MODELING

Line balancing problem is a problem of assigning number of tasks to some workstations such that the total task time does not bigger than the predetermined cycle time, and the precedence relations are satisfied. U-line balancing problem has different characteristic than straight line one. In U-line balancing problems, forward and backward task assignment must be allowed and the first and last task can be assigned in the same workstation.

Urban (1998) carried out the forward and backward assignment by introducing a “phantom” network and add this network to the original network. The example of precedence network with phantom network that was introduced by Urban (1998) is shown in Figure 1. The task assignment to workstation can be made forward through original network or backward through phantom network or simultaneously in both directions.

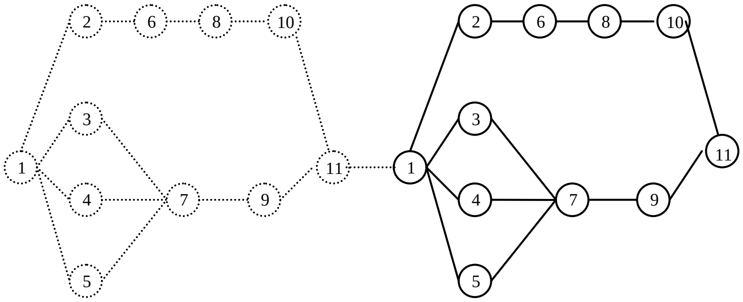

Figure 1. Precedence diagram with phantom network (Urban, 1998)

The assumptions:

1. Permanent and temporary workers tasks times are known and constant.
2. Operators walking times are ignored.
3. The line work continuously and there is no buffer between workstation.
4. Precedence relations between the tasks are known.
5. There is no extra cost for permanent workers.

The notations used in this paper are as follows:

i index of tasks
j index of stations
N total number of tasks
K_{max} maximum number of workstation
C cycle time
P subset of all precedence tasks in original and phantom diagram
PW number of permanent workers
M_P number of tasks allows to be assigned to workstation with a permanent worker
M_T number of tasks allows to be assigned to workstation with a temporary worker
TP_i task time for permanent workers to process task i
TT_i task time for temporary workers to process task i
C_{BT} temporary workers cost per worker per day
C_{AP}^{-} cost of more task for permanent workers in each work station per task per day

C_{AP}^{+} cost of less task for permanent workers in each work station per task per day
C_{AT}^{-} cost of more task for temporary workers in each work station per task per day
C_{AT}^{+} cost of less task for temporary workers in each work station per task per day
C_{CP}^{+} cost of over cycle time in each work station per minute for permanent workers
C_{CP}^{-} cost of under cycle time in each work station per minute for permanent workers
C_{CT}^{+} cost of over cycle time in each work station per minute for temporary workers
C_{CT}^{-} cost of under cycle time in each work station per minute for temporary workers

Decision variables:
V_{ij} 1, if task *i* in phantom precedence diagram with a permanent worker is assigned to workstation *j*; 0, otherwise, $i = 1,..,N, j = 1,\ldots,K_{max}$
X_{ij} 1, if task *i* in original precedence diagram with a permanent worker is assigned to workstation *j*; 0, otherwise, $i = 1,..,N, j = 1,\ldots,K_{max}$
U_{ij} 1, if task *i* in phantom precedence diagram with a temporary worker is assigned to workstation *j*; 0, otherwise, $i = 1,..,N, j = 1,\ldots,K_{max}$
Y_{ij} 1, if task *i* in original precedence diagram with a temporary worker is assigned to workstation *j*; 0, otherwise, $i = 1,..,N, j = 1,\ldots,K_{max}$
PW_j 1, if there is a permanent worker assigned to workstation *j*; 0, otherwise, $j = 1,\ldots,K_{max}$
TW_j 1, if there is a temporary worker assigned to workstation *j*; 0, otherwise, $j = 1,\ldots,K_{max}$

2.1 U-line balancing model

The U-line goal programming balancing problem is developed first and then in the second part straight assembly line balancing model is constructed. The goals of the line balancing problem are:

Goal 1

Number of temporary worker should be minimized to reduce cost.

$$\sum_{j=1}^{K_{\max}} TW_j - d_{BT}^{\ +} = 0 \tag{1}$$

Goal 2

To assure the product quality, one worker should not do too many jobs, so number of tasks that can be done by one worker should be restricted. Number of tasks assigns in each workstation for permanent worker and temporary worker should not difference to the number of tasks allowed.

$$\sum_{i=1}^{N} V_{ij} + \sum_{i=1}^{N} X_{ij} + d_{jAP}^{-} - d_{jAP}^{+} = PW_j.M_p \qquad j = 1,\ldots,K_{\max} \tag{2}$$

$$\sum_{i=1}^{N} U_{ij} + \sum_{i=1}^{N} Y_{ij} + d_{jAT}^{-} - d_{jAT}^{+} = TW_j.M_T \qquad j = 1,\ldots,K_{\max} \tag{3}$$

Goal 3

The workstation time should be as closely as possible to the cycle time

$$\sum_{i=1}^{N} TP_i.V_{ij} + \sum_{i=1}^{N} TP_i.X_{ij} + d_{jCP}^{-} - d_{jCP}^{+} = PW_j.C \qquad j = 1,\ldots,K_{\max} \tag{4}$$

$$\sum_{i=1}^{N} TT_i.U_{ij} + \sum_{i=1}^{N} TT_i.Y_{ij} + d_{jCT}^{-} - d_{jCT}^{+} = TW_j.C \qquad j = 1,..., K_{\max} \tag{5}$$

The objective function can be derived from (1) to (5) as follows:

$$\begin{aligned} Min \quad Z = C_{BT} d_{BT}^{+} + C_{AP}^{-} \sum_{j=1}^{K_{\max}} d_{jAP}^{-} + C_{AP}^{+} \sum_{j=1}^{K_{\max}} d_{jAP}^{+} + C_{AT}^{-} \sum_{j=1}^{K_{\max}} d_{jAT}^{-} + C_{AT}^{+} \sum_{j=1}^{K_{\max}} d_{jAT}^{+} \\ C_{CP}^{-} \sum_{j=1}^{K_{\max}} d_{jCP}^{-} + C_{CP}^{+} \sum_{j=1}^{K_{\max}} d_{jCP}^{+} + C_{CT}^{-} \sum_{j=1}^{K_{\max}} d_{jCT}^{-} + C_{CT}^{+} \sum_{j=1}^{K_{\max}} d_{jCT}^{+} \end{aligned} \tag{6}$$

Constraints:

1. Each task is assigned only in one workstation. The model is:

$$\sum_{j=1}^{K_{\max}} V_{ij} + \sum_{j=1}^{K_{\max}} X_{ij} + \sum_{j=1}^{K_{\max}} U_{ij} + \sum_{j=1}^{K_{\max}} Y_{ij} = 1 \qquad i = 1,..., N \tag{7}$$

2. A workstation can be assigned only with one type of worker. The constraint is modeled using three inequalities as follows:

$$\sum_{i=1}^{N} U_{ij} + \sum_{i=1}^{N} Y_{ij} \le M.TW_j \qquad j = 1,..., K_{\max} \tag{8}$$

$$\sum_{i=1}^{N} V_{ij} + \sum_{i=1}^{N} X_{ij} \le M.PW_j \qquad j = 1,..., K_{\max} \tag{9}$$

$$PW_j + TW_j \le 1 \qquad j = 1,..., K_{\max} \tag{10}$$

3. Number of workstations assigned with a permanent worker is equal with number of available permanent workers. This constraint can be modeled as follows:

$$\sum_{j=1}^{K_{\max}} PW_j = PW \qquad j = 1,..., K_{\max} \tag{11}$$

4. Precedence constraint for phantom and original network diagram are:

$$\sum_{j=1}^{K_{\max}} (K_{\max} - j + 1)(V_{sj} - V_{rj} + U_{sj} - U_{rj}) \ge 0 \qquad (r, s) \in P \tag{12}$$

$$\sum_{j=1}^{K_{\max}} (K_{\max} - j + 1)(X_{rj} - X_{sj} + Y_{rj} - Y_{sj}) \ge 0 \qquad (r, s) \in P \tag{13}$$

5. Open workstation from lowest number can be modeled as:

$$PW_j + TW_j \ge PW_{j+1} + TW_{j+1} \qquad j = 1,..., K_{\max} - 1 \tag{14}$$

2.2. Straight line balancing model

Straight line balancing problem is developed by modifying U-line balancing problem. The phantom network variables are eliminated and then (2) and (3) can be modified as (15) and (16).

$$\sum_{i=1}^{N} X_{ij} + d_{jAP}^{-} - d_{jAP}^{+} = PW_j.M_p \qquad j = 1,..., K_{\max} \tag{15}$$

$$\sum_{i=1}^{N} Y_{ij} + d_{jAT}^{-} - d_{jAT}^{+} = TW_j.M_T \qquad j = 1,...,K_{\max} \qquad (16)$$

In straight line balancing model, (4) and (5) can be formulated as:

$$\sum_{i=1}^{N} TP_i.X_{ij} + d_{jCP}^{-} - d_{jCP}^{+} = PW_j.C \qquad j = 1,...,K_{\max} \qquad (17)$$

$$\sum_{i=1}^{N} TT_i.Y_{ij} + d_{jCT}^{-} - d_{jCT}^{+} = TW_j.C \qquad j = 1,...,K_{\max} \qquad (18)$$

Constraints in (7), (8), and (9) can be remodeled as equation (19) to (21).

$$\sum_{j=1}^{K_{\max}} X_{ij} + \sum_{j=1}^{K_{\max}} Y_{ij} = 1 \qquad i = 1,...,N \qquad (19)$$

$$\sum_{i=1}^{N} Y_{ij} \le M.TW_j \qquad j = 1,...,K_{\max} \qquad (20)$$

$$\sum_{i=1}^{N} X_{ij} \le M.PW_j \qquad j = 1,...,K_{\max} \qquad (21)$$

Since (12) is equation for phantom network, so in a straight line model, the equation is eliminated. The entire equation in a straight line balancing model is similar with the U-line balancing model.

3. NUMERICAL EXAMPLE AND SENSITIVITY ANALYSIS

In this section, an example is shown to illustrate the goal programming model. The permanent workers task times are cited from Gokcen and Agpak (2006) and the temporary workers task times are generated randomly. The temporary workers task times are generated from Uniform (1.1..2.0) times permanent workers task times. The details of task times are shown in Table 1. Effective working hour in a day is assumed to be 8 hours and the manufacture has to produce 40 units in a day.

The priority of the three goals that is used in the formulation is as follows:

Goal with priority level 1 (*P*1)*:*

Number of maximal tasks in one workstation for a permanent or a temporary worker should not exceed 5 and 3.

Goal with priority level 2 (*P*2)*:*

Number of cycle time in each workstation should not exceed 12.

Goal with priority level 3 (*P*3)*:*

Number of temporary workers should be minimized

Table 1. Task times

Task	Task times (minutes)	
	Permanent workers	Temporary workers
1	6	11
2	2	4
3	5	8
4	7	13
5	1	2
6	2	4
7	3	4
8	6	7
9	5	7
10	5	8
11	4	8

The problem was solved using LINGO mathematical programming language on Pentium 4-1.8 GHz. The task assignment of the problem above is given in Table 2. If a workstation consists of a phantom network, then the workstation is allocated with the succeeded task first. However, if a workstation consists of an original network then the workstation is allocated with the precedence task first. The optimal line balancing solution is shown in Figure 2.

Table 2. Task assignment

Workstation	Tasks	WS cycle time	Worker status	Network
1	1, 2, 11	12	Permanent	Original & phantom
2	7, 9	11	Temporary	Phantom
3	8, 10	11	Permanent	Phantom
4	4, 5	8	Permanent	Original & phantom
5	3, 6	12	Temporary	Original & phantom

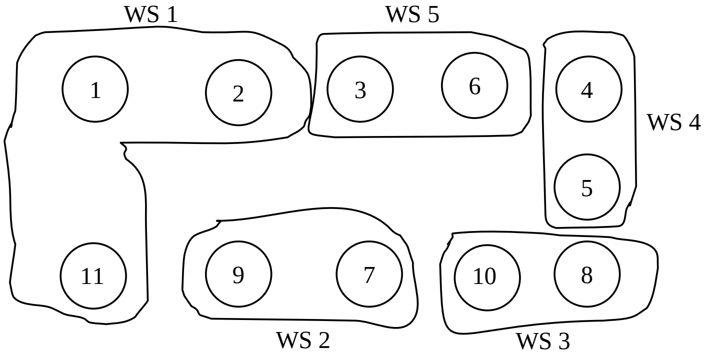

Figure 2. U-line balancing result

First, the sensitivity of the number of temporary workers in various maximal tasks is analyzed. Table 3 shows number of temporary workers in the different number of permanent and temporary workers maximal tasks. The table shows that in this case, number of temporary workers is not sensitive in different of permanent and temporary workers maximal tasks.

Table 3. Sensitivity analysis on worker maximal tasks

Maximal worker tasks		Number of temporary workers (TW)
Permanent	Temporary	
3	3	2
4	3	2
5	3	2
2	2	3
3	2	2
4	2	2
5	2	2

In the next analyses, the goal priorities are rotated and the result can be seen in Table 4. The table shows that the goals can not be satisfied simultaneously. The number of cycle time in each station goal (P2) and the number of temporary workers goal (P3) are conflicting and the number of maximal task in a workstation goal (P1) and P3 are not conflicting. When P2 is satisfied, P3 can not be satisfied and vice versa. Table 4 also supports analysis that in this problem. P1 always can be satisfied and when P2 has the highest priority then the solution is derived only by adding temporary workers.

Table 4. Sensitivity analysis on sequence of priorities

Rotations	Priority 1	Priority 2	Priority 3	Unsatisfied goals	Value of non zero	Cycle time	# of TW
1	P1	P2	P3	P3	${d_{BT}}^{+} = 2$	12	2
2	P1	P3	P2	P2	$d_{1CP}^{+} = 1$ $d_{2CP}^{+} = 9$	21	0
3	P2	P1	P3	P3	${d_{BT}}^{+} = 2$	12	3
4	P2	P3	P1	P3	${d_{BT}}^{+} = 2$	12	2
5	P3	P2	P1	P2	$d_{1CP}^{+} = 6$ $d_{3CP}^{+} = 4$	18	0
6	P3	P1	P2	P2	$d_{1CP}^{+} = 2$ $d_{2CP}^{+} = 5$ $d_{3CP}^{+} = 3$	17	0

In the next step, performance of U-line balancing model is compared with straight line balancing performance. The precedence diagram in Figure 1 and permanent workers tasks time in Table 1 are used. Ten cases are generated by randomizing the temporary workers task times. The goal programming solutions for the cases can be seen in Table 5. The table shows that U-line balancing has better performance than the straight line balancing in three cases (case 2, 3, and 7) and in seven cases, U-line and straight line balancing have equal performances.

Table 5. Performance comparison of U-line and straight line balancing

Case	Number of temporary workers	
	U-line balancing	Straight line balancing
1	2	2
2	1	2
3	1	2
4	1	1
5	2	2
6	2	2
7	1	2
8	2	2
9	2	2
10	1	1

4. CONCLUSIONS

Goal programming U-line and straight line balancing models have been developed and some sensitivity analyses are shown. The cycle time in each station goal and the number of temporary workers goal are conflicting goals. When one goal has a higher priority, then the other one will be unsatisfied. The result also shows that in some cases U-line balancing model has better performance than straight line balancing model and in some cases both of them are equal.

This study shows that the U-line balancing has more benefit than the straight line balancing, but the U-line balancing could be not to interesting since it needs more walking time. In the future research, considering operator walking time will be an interesting and a useful research.

ACKNOWLEDGEMENTS

The author greatly appreciates the anonymous referees for their valuable and helpful suggestions.

REFERENCES

Agrawal, P. K., 1985. “The Related Activity Concept in Assembly Line Balancing.” *International Journal of Production Research*, Vol. 23, pp. 403 – 421.

Baykasoglu, A., 2006. “Multi-Rule Multi Objective Simulated Annealing Algorithm for Straight and U Type Assembly Line Balancing Problems.” *Journal Intelligent Manufacturing,* Vol. 17, pp. 217 – 232.

Corominas, A., Pastor, R., and Plans, J., 2008. “Balancing Assembly Line with Skilled and Unskilled Workers.” *Omega,* Vol.36, pp. 1126-1132.

Erel, E., Sabuncuoglu, I., and Sederci, 2005. “Stochastic Assembly Line Balancing Using Beam Search.” *International Journal of Production Research*, Vol. 43, pp. 1411 – 1426.

Gokcen, H. and Agpak, K., 2006. “A Goal Programming Approach to Simple U-Line Balancing Problem.” *European Journal of Operational Research*, Vol. 170, pp. 577-585.

Kara, Y., Ozcan, U., and Peker, A., 2007. “Balancing and Sequencing Mixed-Model Just-in-Time U-Lines with Multiple Objectives.” *Applied Mathematics and Computation*, Vol. 184, pp. 566-588.

Kara, Y., Paksoy, T., and Chang, C. T., 2009. “Binary Fuzzy Goal Programming Approach to Single Model Straight and U-Shaped Assembly Line Balancing.” *European Journal of Operational Research*, Vol. 195, pp. 335-347.

Miltenburg J., 1998. “Balancing U-lines in a Multiple U-Line Facility.” *European Journal of Operational Research,* Vol. 109, pp. 1-23.

Nakade, K., and Ohno, K., 1999. “An Optimal Worker Allocation Problem for a U-Shaped Production Line.” *International Journal of Production Economics,* Vol. 60-61, pp.353-358.

Nakade, K., and Nishiwaki, R., 2008. “Optimal Allocation of Heterogeneous Workers in a U-Shaped Production Line.” *Computers & Industrial Engineering,* Vol. 54, pp. 432-440.

Ozcan, U., and Toklu, B., 2009. “Multiple-Criteria Decision-Making in Two-Sided Assembly Line Balancing: A Goal Programming and a Fuzzy Goal Programming Models.” *Computers & Operations Research,* Vol. 36, pp. 1955-1965.

Shin, D., and Min., 1991. “Uniform Assembly Line Balancing with Sthocastic Task Times in Just-in-time Manufacturing.” *International Journal of Operations and Production Management,* Vol 11, No. 8, pp. 32 – 34.

Toksari, M. D., Isleyen, S. K., Guner, E. and Baykoc, O. F., 2008. “Simple and U-type Assembly Line Balancing Problems with a Learning Effect.” *Applied Mathematical Modeling,* Vol. 32, pp. 2954 – 2961.

Urban, T. L., 1998. “Note: Optimal Balancing of U-Shaped Assembly Lines.” *Management Science*, Vol. 44, pp. 738-741.

Widyadana, G. A., 2001. “Comparison between Mixed Integer Programming with Heuristic Method for Job Shop Scheduling with Separables Sequence-Dependent Setups.” *Jurnal Teknik Industri,* Vol. 3, No. 1, pp. 9-17.