AN MIP APPROACH TO THE U-LINE BALANCING PROBLEM WITH PROPORTIONAL WORKER THROUGHPUT

Reyhan Erin
Magna PowerTrain
Troy, Michigan 48048. USA

Andres L. Carrano
Department of Industrial and Systems Engineering,
Rochester Institute of Technology,
Rochester, NY 14623. USA.

Moises Sudit
Center for Multisource Information Fusion
Department of Industrial and Systems Engineering.
University at Buffalo.
Buffalo, NY 14260. USA.

Abstract

One of the major challenges faced by manufacturing companies is to remain competitive in dynamic environments, where fluctuations in customer demand and production rates require systems capable of adapting in a practical and economical way. A U-shaped production cell is considered one of the most flexible designs for adapting the workforce level to varying conditions. However, re-balancing efforts are time consuming and often require a new work allocation and line design. In this paper, a two-stage MIP model to determine the best cell design under varying workforce levels is proposed. The model seeks to maintain proportionality between throughput and the number of workers. Computational experiments considering various line configurations (up to 19 stations) and workloads (up to 79 tasks) are performed. The results show the proposed algorithm provides excellent results for all small and medium size problems addressed in this study, as well as for certain configurations of large problems. This approach can be used to generate lookup tables of line designs to help with quick reallocation of worker assignments on the shop floor and with minimal disruption.

1. Introduction

U-shaped production lines, where workers handle one or more machines, are widely used in cellular manufacturing and lean production systems. The main advantages derive from a flexible line with cross-trained workers capable of adapting to changes in demand and production pace. In lean systems, production lines typically strive to meet a pre-determined, demand-driven cycle time denoted as *takt time*. This cycle time is achieved through a line balancing exercise that involves grouping tasks into stations and assigning workers to tend one or more stations. When the demand rate or workforce levels change, the line balancing is revisited and modified with new task groupings and worker assignments. This process of adapting the production resources to the changing demand patterns is known as Shojinka [1]. On the other hand, the implications of not appropriately addressing line balancing span across a wide range of issues, from production control (e.g. overproduction, idle workers, etc.) to accounting (e.g. variable direct labour cost per unit, etc.).

Current re-balancing practices in industry often involve trial-and-error approaches. Such iterative approaches, however, can be tedious and time consuming. Additionally, most line balancing algorithms for U-shaped lines proposed in literature tend to address stable environments and assume little or no variation with respect to workforce levels. Some efforts have addressed workforce issues but have typically sought to minimize the number of workers for a given workload condition [2]. These algorithms may not be practical in a dynamic environment where very frequent, daily sometimes, U-line redesigns are needed. The reasons for such frequent redesigns are typically found whenever a volatile demand environment is coupled with an unstable workforce – in turn a consequence of absenteeism and high levels of temporary workers. This paper was motivated by the needs of a tier-one automotive supplier with a similar environment.

The objective of this paper is to develop a model for designing and balancing U-shaped production lines that maximizes the linearity of worker throughput. Such model aims to produce the design that is most conducive to line balancing while facing a varying number of available line workers.

2. Literature Review

Extensive research has been done on line balancing for traditional production lines over the past four decades. However, research efforts on U-shaped lines started with the widespread adoption of JIT and lean manufacturing concepts and philosophy in the early 1990s. Previous research mainly considers two types of line balancing problems: (*i*) minimization of the required number of stations for a fixed cycle time or, (*ii*) minimization of the maximum cycle time for a fixed number of stations.

Miltenberg and Wijngaard [3] introduced the U-line assembly line balancing problem (ULB) where assignments can be done on both sides of the line. In this work, a dynamic programming model was developed for small size (up to 11 tasks) problems to minimize the required number of stations for a given cycle time. For medium size problems, a "maximum ranked positional weight" heuristic procedure was proposed. This work, however, did not consider walking times or crossover issues on walk-paths.

Several efforts have since addressed the ULB with different approaches and various degrees of success. Urban [4] used an integer programming model with a branch and bound preliminary search to solve problems with more than 21 tasks. Ajenblit and Wainwright [5] used a genetic algorithm approach with six different task assignment methods on larger problems (up to 111 tasks). Scholl and Klein [6] developed a branch and bound procedure for simple assembly line balancing. In this work, models with up to 297 tasks were solved with optimal or best solutions. Erel *et al.* [7] proposed solving the ULB problem with a simulated annealing algorithm. Martinez and Duff [8] proposed a genetic algorithm approach to improve results obtained from 10 different heuristics. Gokcen and Agpak [9] proposed a goal programming approach for simple U-line balancing problems. In this work, up to 30 tasks with conflicting goals, including minimization of the number of work-stations, cycle time and the number of tasks, were considered.

There is archival literature on production lines that have considered walking times, waiting times and walk-paths. Nakade and Ohno [10] developed a procedure to calculate cycle times under optimal worker allocation scenario while considering walking and waiting times. Nakade and Ohno [11] proposed a model for deterministic walking and process time scenarios. In this paper, the minimum number of workers under a given cycle time is first determined. Then, an optimal worker allocation with a minimum number of operators is obtained. Stockton *et al.* [12] modelled flexible walk cycles that included walking times using genetic algorithms. The objectives were to minimize the number of operators and to reduce the smallest operator cycle time. Shewchuk [2] addressed the worker allocation problem for U-shaped lines with objectives of minimizing the number of workers while maximizing full work. The proposed model incorporates circular walking paths where crossovers are not permitted.

The majority of the existing research on line balancing has been developed for stable production environments where the rebalancing frequency is very low. Dynamic environments, due to demand volatility and workforce instability, require engaging on more frequent line balancing of U-lines, thus the chosen approach should be easy and economical. This work addresses this gap.

3. Production System Description

Figure 1 illustrates a simple u-shaped production line. Multiple stations are closely located in a u-shape to form a production cell. Tasks are allocated to stations according to precedence relationships. Typically, once tasks are allocated to stations, then assignments of workers are established. Multiple stations can be assigned to the same worker as long as crossover and cycle time constraints are satisfied. The term "walk-path" is used to define the area of responsibility of each worker. Figure 1 represents a U-shaped system with 7 stations and 4 workers (A, B, C and D). With *M* representing the number of stations and *N* representing the number of workers, the system in Figure 1 can be described as (*M*=7, *N*=4). In this Figure, worker A is responsible for stations 1 and 7, worker B is responsible for stations 2 and 3, worker C is responsible for stations 4 and 5, while worker D only tends to station 6.

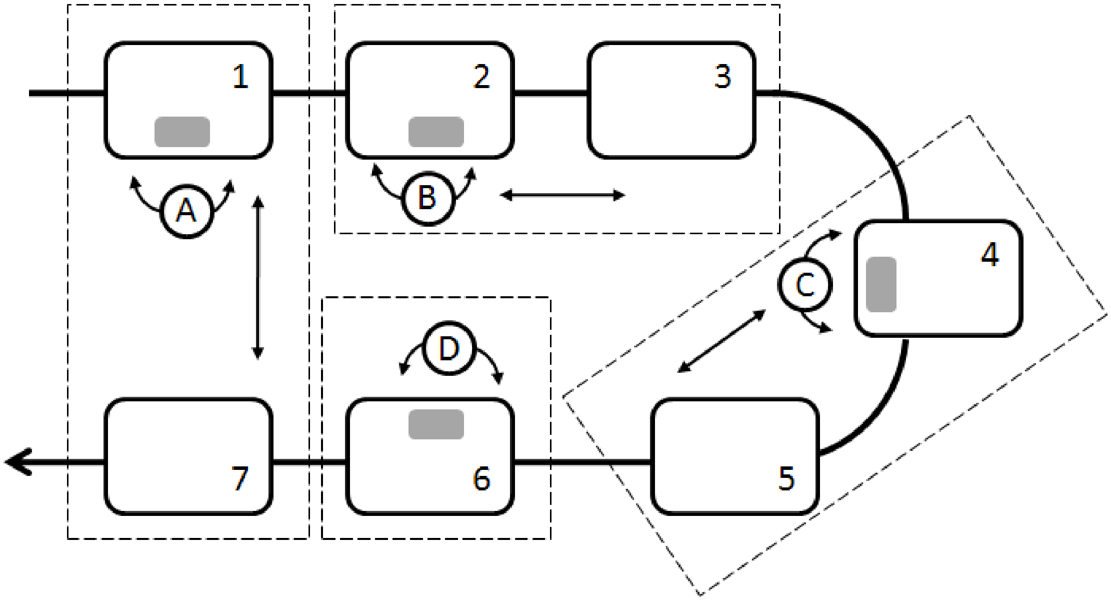

Figure 1: U-shaped production line (*M*=7, *N*=4)

In a U-shape cell, parts flow sequentially through the stations in the system. After processing at station 1 is completed, worker A delivers the part to worker B for processing at station 2. After that is completed, and since worker B is responsible for station 3 as well, he then walks the assembly down to station 3 to perform further processing. After delivering the part to worker C at station 4, worker B then returns to station 2. A crossover occurs when a worker, by design, systematically intrudes into the work zone (as defined by the walk-path) of another worker. This would occur if, for example, in Figure 1, worker A were in charge of stations 1 and 3, while worker B was in charge of stations 2 and 6. Crossovers are undesirable for several reasons, including safety and productivity considerations.

4. Mathematical model

As mentioned before, the objective of this model is to determine the cell configuration that, for a variable number of workers, allows for maximum linearity of worker throughput. This is accomplished with a two-stage MIP model. The first stage seeks to allocate the tasks with the objective of balancing the workload across the stations. In the second stage, workers are assigned to stations with the goal of balancing walk-path times (i.e. task time + walking time). The second stage is run for different levels of workforce, from 2 workers to M-1 (a practical maximum) and iterated until the proposed linearity metric is acceptable. During this iterative process, the objective value is relaxed (stage 1) and the operators are reassigned (stage 2), until the target linearity is achieved. Although not part of the MIP model, a third stage is used to calculate the linearity of the results. Figure 2 shows the logic of the model.

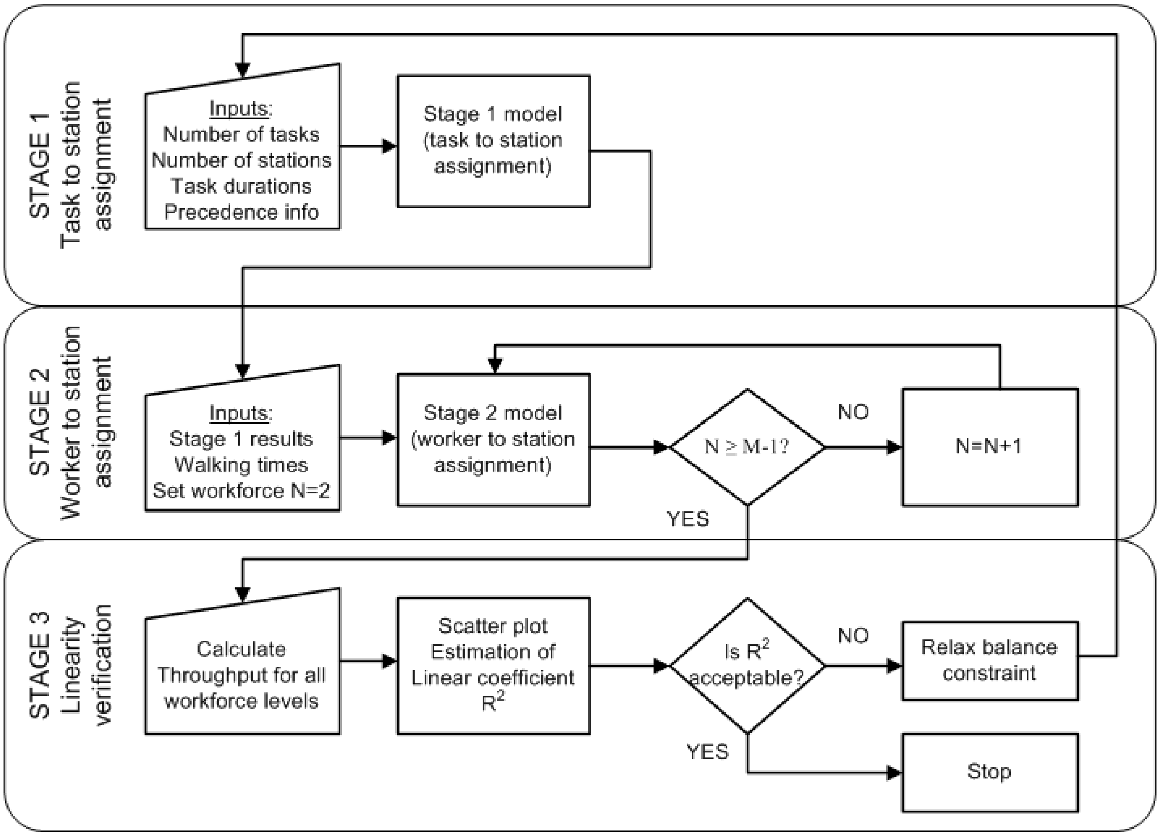

Figure 2: Flowchart of the two-stage model

In this model, the following assumptions are made:

1) Task processing times and worker walking time between stations are considered deterministic.
2) Workers are cross-trained on all tasks.
3) The production protocol is that of one-piece flow for a single model.
4) Operators are not allowed to chase each other or to share stations.
5) Machine breakdowns are not considered.
6) There is no machine waiting time.
7) There is an infinite supply of parts in front of the first station.

Also, it is assumed that the stations are located in a grid arrangement and that it takes t units of time to travel to adjacent stations (in the same row) or directly across. This assumes equal and constant travel speed with no significant acceleration/deceleration. It was also assumed that it would take twice that amount of time ($2t$) to reach a station located diagonally across. This assumed a worst case scenario (since $2t > \sqrt{2}\,t$). These travel assumptions are mapped to other stations located farther away. Figure 3 depicts the travel times for a u-cell configurations with even number of stations.

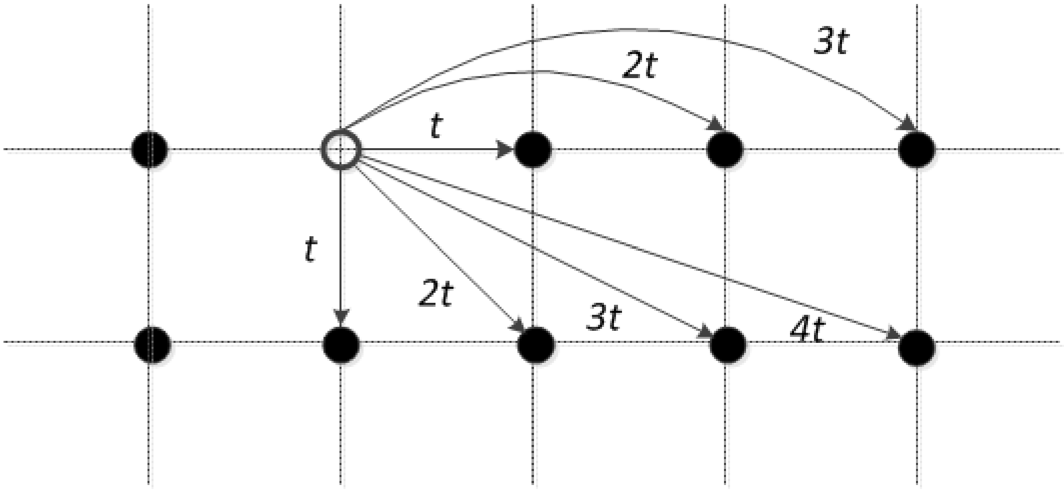

Figure 3: Travel times to stations (even number of stations, M=10)

4.1 Stage-1 mathematical model

The model in this stage seeks to distribute the tasks to stations without initially considering worker assignments. The required inputs for this stage are: (*i*) number of stations, (*ii*) number of tasks, (*iii*) precedence relationships, and (*iv*) task processing times.

Stage-1 nomenclature

T = Total number of tasks

M = Total number of stations

i = Index for tasks (i = 1, 2, ……, T)

j = Index for stations (j = 1, 2, 3……, M)

D_i = Standard time to process the task i (deterministic)

P (i, h) = A set of precedence of tasks, P={(i, h) / task i must be completed before task h}

$Stime_j$ = Sum of all the task processing times assigned to station j

Maxf = Maximum of the time difference between station times. ($Stime_j$)

Stage-1 decision variables

To establish the allocation of the tasks to the stations, define

$$X_{ij} = \begin{cases} 1, & \text{if task } i \text{ is assigned to station } j \\ 0, & \text{otherwise} \end{cases}$$

Stage-1 objective function:

The objective function in this stage seeks to balance the workload among stations by minimizing the maximum difference between station times (*$Stime_j$)* pairs.

$$\text{Minimize } Z_1 = Maxf$$

Stage-1 constraints:

[C1] Each task can be assigned to *only one* station.

$$\sum_{j=1}^{M} X_{ij} = 1 \qquad \forall_{i} = 1...T$$

[C2] All stations must have an assignment. Dummy stations are not allowed.

$$\sum_{i=1}^{T} X_{ij} \geq 1 \qquad \forall j = 1...M$$

[C3] This constraint ensures that precedence relations are maintained. P = {(*h*,*i*) / task *h* must be completed before task *i*} . If task *h* is assigned to station *j*, task *i* can only be assigned to station *k*, $k \in M \quad M = \{k / k = j, j+1, j+2,...M\}$.

$$Xhj \leq \sum\nolimits_{k=j}^{M} Xik \qquad \forall (h,i) \in P$$

$$\forall j = 1...M$$

[C4] The following constraints are used to compare the times (*Stime$_j$*) between stations. The variable *Maxf* gets assigned to the largest value of the gaps. The two left terms in the constraints each give the total processing time at station *j* and *g*. *Maxf* is consequently minimized through the objective function.

$$\sum\nolimits_{i=1}^{T} D_i * X_{ij} - \sum\nolimits_{i=1}^{T} D_i * X_{ig} \leq Maxf \qquad \forall j = 1...(M-1) \qquad (a)$$

$$\forall g = (j+1)...M$$

$$\sum\nolimits_{i=1}^{T} D_i * X_{ig} - \sum\nolimits_{i=1}^{T} D_i * X_{ij} \leq Maxf \qquad \forall j = 1...(M-1) \qquad (b)$$

$$\forall g = (j+1)...M$$

Stage-1 output:

$$Stime_j = \sum\nolimits_{i=1}^{T} Di * Xij \qquad \forall j = 1...\mathrm{M}$$

4.2 Stage-2 Mathematical Model

This stage is used to create walk-paths by allocating workers to the stations and with the aim of balancing walk-path times. A walk-path time is the sum of the task processing plus walking times assigned to the same worker. The model for this phase considers two major constraints when developing the walk-paths: (*i*) path crossover and, (*ii*) balanced workloads. Workers cannot be assigned to the stations that require crossing the walk-path of another worker. In order to generate a measure of throughput linearity, Stage-2 is repeated for different numbers of workers *e*: $\forall e,\ e \in N,\ N = \{e / e = 1,2,3,...M-1\}$.

Stage-2 nomenclature

N = Total number of workers

M = Total number of stations

e = Index for workers (i = 1, 2, ……, N)

j = Index for stations (j = 1, 2, 3..…, M)

Stime $_j$ = Total processing time at station j

L_{ij} = Walking time between station i and j.

$LS_{\max}$ = An input used to prevent crossovers.

$$LS_{\max} = \begin{cases} M/2 \quad , & \text{if } M \text{ is } e\text{ven number} \\ \lfloor M/2 \rfloor, & \text{if } M \text{ is odd number} \end{cases}$$

$HS_{\max}$ = An input used to prevent crossovers.

$$HS_{\max} = \begin{cases} (M/2)+1 \ , & \text{if } M \text{ is } e\text{ven number} \\ \lceil M/2 \rceil + 1 \ , & \text{if } M \text{ is odd number} \end{cases}$$

Stage-2 decision variables:

1) This variable is used to represent station assignments to workers.

$$Y_{je} = \begin{cases} 1, & \text{if station } j \text{ is assigned to worker } e \\ 0, & \text{otherwise} \end{cases}$$

2) This variable is used to determine the walk-path for each worker.

$$W_{jhe} = \begin{cases} 1, \text{ if station } j \text{ and station } h \text{ are assigned to same worker } e \\ 0, \text{ otherwise} \end{cases}$$

3) Variable IM_{jhe} is referred as *immediate station.* It is used to determine the sequence of stations through which worker *e* moves in his walk-path.

$$IM_{jhe} = \begin{cases} 1, \text{ if station } j \text{ and } h \text{ are assigned to same worker } e, \\ \quad \text{and if worker } e \text{ moves to station } h \text{ immediately after station } j \\ 0, \text{ otherwise} \end{cases}$$

4) Variable FS_{je} denotes the first station visited in the walk-path**.**

$$FS_{je} = \begin{cases} 1, \text{ if station } j \text{ is the } \textit{first} \text{ station in the walkpath of worker } e \\ 0, \text{ otherwise} \end{cases}$$

5) Variable LS_{he} denotes the last station visited in the walk-path**.**

$$LS_{he} = \begin{cases} 1, \text{ if station } h \text{ is the } \textit{l}\text{ast station in the walkpath of worker } e \\ 0, \text{ otherwise} \end{cases}$$

6) Variable B_{jhe} is created to determine the end of the cycle. If $B_{jhe} = 1$, then *j* is the last station and *h* is the first station in the sequence.

$$\mathrm{B}_{\mathrm{jhe}} = \begin{cases} 1, \text{ if station j and h are assigned to same worker e}, \\ \quad \text{and if worker e goes back from the last station (j) to the first station (h)} \\ 0, \text{ otherwise} \end{cases}$$

7) $Wtime_e$ = Sum of all the task processing times and walking times between stations assigned to worker *e,* also referred as walk-path time of worker *e.*

8) maxT = Maximum of walk-path times

Stage-2 objective function:

The objective function of the second stage seeks to balance walk-path times by minimizing the maximum walk-path time.

Minimize Z$_2$ = $maxT$

Stage-2 constraints:

[C6] Each station can only be assigned to one worker.

$$\sum_{e=1}^{N} Y_{je} = 1 \qquad \forall j = 1 \ldots M$$

[C7] One worker can be assigned to many stations if the number of workers is less than the number of stations. The following constraints determine the stations in the walk-path of each worker.

$$1 + W_{jge} \geq Y_{ge} + Y_{je} \qquad \forall j = 1 \ldots (M-1) \quad \forall g = (j+1) \ldots M \quad \forall e = 1 \ldots N \qquad \text{(a)}$$

$$W_{jge} \leq Y_{ge} \qquad \forall j = 1 \ldots (M-1) \quad \forall g = (j+1) \ldots M \quad \forall e = 1 \ldots N \qquad \text{(b)}$$

$$W_{jge} \leq Y_{je} \qquad \forall j = 1 \ldots (M-1) \quad \forall g = (j+1) \ldots M \quad \forall e = 1 \ldots N \qquad \text{(c)}$$

[C8] The following constraints prevent crossovers. Neither diagonal nor horizontal crossovers are allowed in this model. Constraint (C8-a) prevents diagonal crossovers (across the main aisle in the cell) while constraints C8 (b, c) prevent horizontal crossovers (along the row of stations on the same side of the cell).

$$3 - Y_{je} - Y_{ge} \geq Y_{hf} + Y_{rf} \qquad \forall j = 1 \ldots M-1 \quad \forall g = (j+1) \ldots M \quad \forall h = (j+1) \ldots (g-1) \quad \forall r = (g+1) \ldots M \quad \forall e = 1 \ldots N \quad \forall f = 1 \ldots N\text{, } e \neq f \qquad \text{(a)}$$

$$(g-j-1)*(Y_{je}+Y_{ge}-1) \leq \sum_{k=(j+1)}^{(g-1)} Y_{ke} \qquad \forall j=1\ldots(LS_{\max}-2) \quad \forall g=(j+2)\ldots LS_{\max} \quad \forall e=1\ldots N \qquad (b)$$

$$(g-j-1)*(Y_{je}+Y_{ge}-1) \leq \sum_{k=(j+1)}^{(g-1)} Y_{ke} \qquad \forall j=HS_{\max}\ldots(M-2) \quad \forall g=(j+2)\ldots M \quad \forall e=1\ldots N \qquad (c)$$

[C9] The following constraints are used to determine the sequence of stations in each walk-path. Constraints C9 (a, b, c) determine the *next immediate* station in the sequence for worker *e* (if multiple stations are assigned to him). Constraints C9 (d, e, f) are used to decide the *last station* in the sequence to complete one cycle. Constraints C9 (g, h, i) are created to determine the *first station* in the sequence. Return time from last station to first station is calculated through constraints C9 (j, k, l).

$$IM_{jge} \geq W_{jge} - \sum_{h=(j+1)}^{(g-1)} W_{jhe} \qquad \forall j=1\ldots(M-1) \quad \forall g=(j+1)\ldots\mathrm{M} \quad \forall e=1\ldots N \qquad (a)$$

$$W_{jge} \geq IM_{jge} \qquad \forall j=1\ldots M-1 \quad \forall g=(j+1)\ldots M \quad \forall e=1\ldots N \qquad (b)$$

$$\sum_{h=(j+1)}^{(g-1)} W_{jhe} \leq M*(1-IM_{jge}) \qquad \forall j=1\ldots(M-1) \quad \forall g=(j+1)\ldots\mathrm{M} \quad \forall e=1\ldots N \qquad (c)$$

$$\sum_{g=1}^{M}(g * LS_{ge}) \geq (j * Yj_e) \qquad \forall j = 1...M \quad \forall e = 1...N \qquad \text{(d)}$$

$$\sum_{j=1}^{M} LS_{je} = 1 \qquad \forall e = 1...N \qquad \text{(e)}$$

$$LS_{je} \leq Y_{je} \qquad \forall e = 1...N \quad \forall j = 1...M \qquad \text{(f)}$$

$$\sum_{g=1}^{M}(M-g) * FS_{ge} \geq (M-j) * Y_{je} \qquad \forall j = 1...M \quad \forall e = 1...N \qquad \text{(g)}$$

$$\sum_{j=1}^{M} FS_{je} = 1 \qquad \forall e = 1...N \qquad \text{(h)}$$

$$FS_{je} \leq Y_{je} \qquad \forall e = 1...N \quad \forall j = 1...M \qquad \text{(i)}$$

$$B_{jge} + 1 \geq FS_{ge} + LS_{je} \qquad \forall j = 2...M \quad \forall g = 1...(j-1) \quad \forall e = 1...N \qquad \text{(j)}$$

$$B_{jge} \leq FS_{ge} \qquad \forall e = 1...N \quad \forall j = 2...M \quad \forall g = 1...(j-1) \qquad \text{(k)}$$

$$B_{jge} \leq LS_{je} \qquad \begin{array}{l} \forall e = 1...N \\ \forall j = 2...M \\ \forall g = 1...(j-1) \end{array} \qquad (1)$$

[C10] This constraint is developed to balance walk-path times by driving worker time to a minimum. The terms at the left side of the constraint represent the task processing times, the walking times from station to station, and the return time after completion of one cycle. The sum of all three gives the overall walk-path time for each worker.

$$\sum_{j=1}^{M} Stime_j * Y_{je} + \sum_{j=1}^{M-1} \sum_{g=(j+1)}^{M} L_{jg} * IM_{jge} + \sum_{j=2}^{M} \sum_{g=1}^{(j-1)} L_{gj} * B_{jge} \leq \max T \qquad \forall e = 1...N$$

Stage-2 output:

$$\text{Wtime}_{(e)} = \sum_{j=1}^{M} Stime_j * Y_{je} + \sum_{j=1}^{M-1} \sum_{g=(j+1)}^{M} L_{jg} * IM_{jge} + \sum_{j=2}^{M} \sum_{g=1}^{(j-1)} L_{gj} * B_{jge} \qquad \forall e = 1...N$$

5. Computational experiments

The data used to test this model was obtained from two distinct sources. The first-tier automotive supplier, which motivated this work, provided one set of data. The other sets were obtained from archival literature available in the “Homepage for Assembly Line Optimization Research” [13], which hosts a collection of data sets used in simple-line and U-line balancing problems. In this paper, a total of 8 different data sets were used. Each data set consisted of a collection of task processing times and precedence diagrams. Table 1 presents a classification of the data sets. All eight data sets were tested with small and medium size configurations (M≤16), while four data sets were used to test large configurations (M>16). For practical reasons, the extreme points in the number of workers (i.e. N=1 and N=M) were excluded from the analysis. Total of 108 different configurations were tested for all levels of workforce between [2, M-1], for a total of 1008 experiments. All experiments were conducted on a 3.20GHz processor with 1 GB of RAM. The proposed model was solved with OPL Studio using a CPLEX program. A branch and cut model was used in searching for an optimal solution.

Table 1: Summary of data sets (number of tasks in parenthesis)

Small / Medium Size	**Large Size**
(5-16 stations)	**(17-19 stations)**
Mitchell (21)	Sawyer (30)
Roszieg (25)	Warnecke (58)
Sawyer (30)	Tonge (70)
Kilbrid (45)	Automotive Supplier (79)
Warnecke (58)	
Tonge (70)	
Wee-Mag (75)	
Automotive Supplier (79)	

5.1. Linearity measurement

With the output from the model, a scatter plot between throughput and the number of workers is developed. A linear regression with a best fit line that intercepts at the origin is obtained and its corresponding coefficient of determination R^2 then calculated. In this work, R^2 is used to measure the linearity of throughput per worker.

6. Results and discussion

Table 2 presents the computational results. A cursory examination of the summary data suggests that the proposed procedure performs well within the boundaries of the experiment.

Table 2: Computational results (R^2)

Number of stations (M)	Number of data sets	Number of tasks (T)	Number of workers (N)	Problem size	Computational results (R^2)	
					Average	**Range**
5	8	21-79	2-4	S/M	**0.95**	**0.91-0.98**
6	8	21-79	2-5	S/M	**0.99**	**0.96-0.99**
7	8	21-79	2-6	S/M	**0.95**	**0.92-0.96**
8	8	21-79	2-7	S/M	**0.97**	**0.95-0.97**
9	8	21-79	2-8	S/M	**0.89**	**0.85-0.95**
10	8	21-79	2-9	S/M	**0.95**	**0.92-0.97**
11	8	21-79	2-10	S/M	**0.97**	**0.96-0.98**
12	8	21-79	2-11	S/M	**0.95**	**0.94-0.97**
13	8	21-79	2-12	S/M	**0.95**	**0.92-0.98**
14	8	21-79	2-13	S/M	**0.93**	**0.90-0.96**
15	8	21-79	2-14	S/M	**0.95**	**0.92-0.97**
16	8	21-79	2-15	S/M	**0.92**	**0.83-0.97**
17	4	30-79	2-16	L	**0.89**	**0.82-0.93**
18	4	30-79	2-17	L	**0.86**	**0.71-0.95**
19	4	30-79	2-18	L	**0.80**	**0.67-0.98**

All the scenarios in the small/medium size range were solved with high average linearity (R^2 ϵ [0.89-0.99]) while the large size scenarios presented a satisfactory coefficient of determination (R^2 ϵ [0.80-0.89]). However, it can be observed that overall performance of the algorithm degrades with the size of the problem (i.e. T and M). Based on these results, it appears the algorithm works very well for all small and medium size problems and for some configurations of large problems.

Table 3 shows the workload assignments obtained from the model and with the provided data sets. The percentages shown in the table depict of the amount of work to be allocated to each station such that, after accounting for walking times, the linearity (R^2) falls in the ranges shown in Table 2. This allocation can be used as a general guideline for a permanent work distribution among stations when the linearity of the throughput per worker is paramount.

Table 3: Best solutions for work allocation (percentages by station)

Number of Stations														
5	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
23	15	20	17	16	9	12	9	9	9	7	9	9	6	6
11	23	10	11	7	11	12	3	11	7	6	7	7	6	7
24	22	8	17	7	13	7	11	3	7	4	4	4	4	4
12	23	9	17	6	9	11	7	5	7	6	10	7	6	7
30	4	17	7	8	7	6	8	7	6	6	9	6	5	3
	13	17	11	12	7	13	5	6	5	8	5	6	6	6
		19	11	8	8	5	9	3	9	8	4	5	6	6
			9	18	11	13	8	9	8	8	5	4	5	3
				18	13	11	10	8	7	6	5	7	6	5
					12	5	11	11	6	6	5	8	6	4
						5	10	6	8	5	4	4	7	10
							9	10	10	7	10	5	7	4
								12	6	5	6	4	6	5
									5	9	8	4	5	5
										9	4	5	8	4
											5	10	2	6
												5	8	6
													1	5
														4

Finally, once all tasks are allocated, the various worker assignments can be depicted for all scenarios of workforce levels between 2 and M-1.

7. Conclusions

A 2-stage MIP model is developed for U-shaped production lines for environments with unstable workforce. The model seeks to optimize the linearity of throughput per worker while maintaining the line balancing within acceptable limits. The algorithm is able to solve problems up to 79 tasks and 19 stations. The method tabulates the best work allocation by station that is conducive to both high throughput-worker linearity and line balancing. Tables of line design for varying number of stations and workforce levels can be derived. These look-up tables can be very useful on the shop floor, where a quick redesign response may be needed to adapt to daily changes in workforce levels.

References

[1] Y. Monden. Toyota Production System. 2nd Ed. Industrial Engineering and Management Press, Institute of Industrial Engineers. Norcross, GA. 1993.

[2] J.P. Shewchuk. Worker allocation in lean U-shaped production lines. International. Journal of Production Research (2008). 46(13), 3485-3502.

[3] J. Miltenberg, J. Wijngaard. The U-line balancing problem. Management Science (1994). 40(10), 1378-1388.

[4] T. Urban. Optimal balancing of U-shaped assembly lines, Management Science (1998). 44(5), 738-741.

[5] D.A. Ajenblit, R.L. Wainwright. Applying genetic algorithms to the U-shape assembly line balancing problem. Proceedings of the 1998 IEEE Conference on Evolutionary Computation, ICEC. (1998). Anchorage, Alaska, May 4-9. 96-101.

[6] Scholl, R. Klein. ULINO: Optimally balancing U-shaped JIT assembly lines, International Journal of Production Research (1999). 37(4), 721-736.

[7] E. Erel, I. Sabuncuoglu, B.A. Aksu. Balancing of U-type assembly systems using simulated annealing. International Journal of Production Research (2001). 39(13), 3003-3015.

[8] U. Martinez, W. Duff. Heuristic approaches to solve the U-shaped line balancing problem augmented by genetic algorithms. IEEE Systems and Information Engineering Design Symposium (2004).Charlottesville, Virginia, April 16. 287-293.

[9] H. Gokcen, K. Agpak. A goal programming approach to simple U-line balancing problem. European Journal of Operational Research (2006), 171(2), 577-585.

[10] K. Nakade, K. Ohno. Stochastic analysis of a U-shaped production line with multiple workers. Computers & Industrial Engineering (1997), 33 (3-4), 809-812.

[11] Nakade, K. and Ohno, K. An optimal worker allocation problem for a U-shaped production line. International Journal of Production Economics (1999), 60-61, 353-358.

[12] D.J. Stockton, J. Ardon-Finch, R. Khalil. Walk cycle design for flexible manpower lines using genetic algorithms. International Journal of Computer Integrated Manufacturing (2005), 18(1), 15-26.

[13] S. Armin. Homepage for assembly line optimization research [online]. URL: http://www.wiwi.uni-jena.de/Entscheidung/alb/index.htm. [accessed March 2007].