2006

Balancing mixed-model assembly line to reduce work overload in a multi-level production system

Pravin Y. Tambe
Louisiana State University and Agricultural and Mechanical College, ptambe1@lsu.edu

Recommended Citation

Tambe, Pravin Y., "Balancing mixed-model assembly line to reduce work overload in a multi-level production system" (2006). *LSU Master's Theses*. 3526.
https://digitalcommons.lsu.edu/gradschool_theses/3526

BALANCING MIXED-MODEL ASSEMBLY LINE TO REDUCE WORK OVERLOAD IN A MULTI-LEVEL PRODUCTION SYSTEM

A Thesis
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirement for the degree of
Master of Science in Industrial Engineering

in

The Department of Industrial Engineering

By
Pravin Y. Tambe

B.E., Doctor B. A. M. University, 2001
May 2006

ACKNOWLEDGEMENTS

First of all, I am grateful to my advisor and mentor, Dr. Warren Liao whose continual direction and support has led me to keep high standards throughout this thesis. I will always be indebted for his patience and encouragement and am thankful for the opportunity to work with him. The research that went into the creation of this subject has not only helped me in improving my analytical thinking but also refined my clarity in communication. I could not have come this far without his pertinent guidance.

I am thankful to Dr. Charles McAllister and Dr. Xiaoyue Jiang for their invaluable suggestions, which were indispensable in shaping up this thesis. I am especially thankful to Dr. Charles McAllister for critically evaluating my work and for his constructive criticisms that made this report more meaningful.

I thank all my friends over here in the United States and back home in India for their continuous support, encouragements and good wishes. I would like to take this opportunity to thank Mr. Vikas Singh for his enormous help in coding the algorithms.

Finally, I would like to thank almighty for everything that I am/have today.

I would like to dedicate this thesis to

my adorable brother and

my wonderful parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Generating the optimal production schedule for an assembly line, which will balance the workload at all the production stages, is a difficult task considering a variety of practical constraints. Varying customer demand is an important factor to be considered when designing an assembly line. In order to respond to varying customer demand, many companies are attempting to make their production system more flexible/agile or adaptable to change.

Due to the volatile nature of market, companies cannot afford to manufacture same type of product for long period of time and neither can maintain high inventory level; to tackle this problem we propose a new approach of balancing mixed-model assembly line in a multi-level production system.

The emphasis is on incorporating the effect of set-up times of lower production levels on the final assembly schedule. This will facilitate stabilized workload among and across the stations and effectively balance the production schedule at all production stages. As a result, the proposed model assures that workloads are balanced and setup times are reduced to such an extent that WIP and overall inventories are kept to a low level.

Key Words*:* Mixed-model assembly line balancing, scheduling mixed-model assembly, production planning and control, Just-in-time scheduling

CHAPTER 1. INTRODUCTION

Facilities are constructed to produce goods that can be sold to customers, while doing so capital, energy, human, information and raw material are acquired, transported, and consumed. Companies always aim for optimizing the resources consumed during this transformation. One way of optimum utilization is to reduce the non-value added cost associated with these resources. In a typical manufacturing industry there are eight major wastes (Non value added costs).

1.1 Non Value Added Costs

1.1.1 Overproduction

This is the most deceptive waste in today's variable demand scenario; this leads to unnecessary utilization of resources. Overproduction includes making more than what is required and making products earlier than required. The rationale behind this just-in-case thinking is undue use of automation.

1.1.2 Inventory

Higher inventory is not beneficial for any company in today's variable-demand business climate. The danger associated with high inventory is the chances of obsoleteness. In the case of obsolete inventory, all costs invested in the production of a part are wasted. Poor forecast, unbalanced workload, product complexity are the major reasons behind companies maintaining higher inventory.

1.1.3 Non-value Added Processing

Efforts that add no value to the desired product from a customer's point of view are considered as non-value added processing. Vague picture of customer requirements, communication flaws, inappropriate material or machine selection for the production are the reasons behind this type of waste.

1.1.4 Defects

Companies give much emphasis on defects reduction. However defects still remain the major contributors towards the non-value added cost. Cost associated with this is quality and inspection expenditure, service to the customer, warranty cost and loss of customer fidelity.

1.1.5 Transportation

Cost associated with material movement is a significant factor in the non-value added cost function. This consumes huge capital investment in terms of equipment required for material movement, storage devices, and systems for material tracking. Labor cost associated with the material movement also comes under transportation cost. Transportation does not add value towards the final product.

1.1.6 Motion

Any motion that does not add value to the product or service comes under non-value added cost; it may include man or machine movement. Time spent by the operators looking for a tool, extra product handling and heavy conveyor usage are the typical example of the motion waste. This is a result of improper design of the workplace, inconsistent work methods, and poor workplace organization and housekeeping.

1.1.7 Waiting

Inappropriate material flow selection is one of the reasons behind waiting time. The time spent on waiting for raw material, the job from the preceding work station, machine downtime, and the operator engaged in other operations and schedules are the major contributors in the waiting time.

1.1.8 People

Company culture, hiring practices, low pay, and less training are the reasons behind underutilizing the work force available. Proper utilization of human resources is one of the major tasks faced by the manufacturing industries.

1.2 Assembly Line Balancing

In our paper we are dealing with all these issues, but our primary focus is on minimizing waste related to the bottom four of the above-mentioned categories. Assembly Line Balancing (ALB) is one way to achieve that. Before going into the details of ALB we will see how the concept of assembly line evolved throughout history. Principle of interchangeability and division of labor brought in the concept of assembly line, primary aim of the assembly line was to facilitate mass production, standardization, simplification and specialization. Besides this, assembly line was also useful in dividing complex work structures into number of elemental tasks, which would simplify the complexity of assembly work. ALB also provides flexibility to the employees; thereby reducing the monotonous activity thus improved job satisfaction for the employees. From the manufacturers point of view foremost advantage of ALB is the ability to keep direct labors busy doing productive work. Historically assembly line is designed for high volume production of single item or similar family of items.

1.3 Mixed-Model Assembly Line Balancing

In practical, many items do not have sufficient demand to justify separate assembly line, but family of separate product might. We can use multiple product line to match the variable demand; however advantage of inventory reduction is lost. Recent attention to just-in-time manufacturing brings considerable focus on production of variety of models on the same assembly line, so is the need for Mixed-Model Assembly Line Balancing (MALB). Input given to the line is directly proportional to the demand rate of the particular model. The precise sequence of workstations is important considering the minimization of imbalance between the model types. Prenting and Thomopoulos are among the first to address the issue of MALB. Focus was to balance the workloads and construction of daily sequence to provide stable workloads on the assembly line. The stability of work station is particularly important from the company's point of view, stability allows the sufficient time for the workers so that they do not have to rush to complete the allocated task, this indirectly aid in quality improvement. In MALB products can generally be classified into two types, items that are either shorter or longer than average in all stations and items that have their own long and short stations. The goal here is to develop a task sequence such that it will minimize the variation for all the models that are assemble on the assembly line under consideration.

1.4 Why Mixed-model Assembly Line Balancing?

We use an example given in Miltenburg *et al* (1989) to explain further. A production shop consists of a fabrication area and an assembly area. Nine different parts are manufactured on 16 different machines in the fabrication area; 4 different products are assembled from the 9 parts on a mixed-model assembly line. Demand per product per day is shown in Table 1. Production manager has to decide a schedule such that there is uniform consumption of parts at all the production level to ensure lower inventory levels.

Table 1. Demand data per day per product.

Product	Mon	Tue	Wed	Thu	Fri
1	40	35	0	0	0
2	0	5	40	5	0
3	0	0	0	35	15
4	0	0	0	0	25

The machine requirements for each part are shown in Table 2. Because each part has different machine requirements, machines can be grouped into cells (group technology/cellular manufacturing) based on the production requirements and the capacity of the machine.

Table 2. Parts requirements on individual machines.

Machine **Parts**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	80	70	0	0	0	0	0	0	25	0
2	0	0	0	0	40	35	0	105	0	45
3	0	0	0	0	0	5	40	5	0	25
4	120	115	80	80	0	0	0	0	80	0
5	80	90	160	55	0	0	0	0	15	0
6	0	0	0	0	0	5	40	5	0	50
7	0	0	0	0	0	0	0	35	0	65
8	0	10	80	10	0	0	0	0	25	0
9	0	0	0	105	0	0	0	0	45	0

Grouping of the machines in two different cells can be seen in Table 3. Assume that the assembly line runs 8 hours per day, 5 days a week and has capacity of 5 units per hour. The total demand is 200 units and so sequence in which these 200 units will be assembled must be selected.

Table 3. Grouping Similar Machines in one cell
Traditional batch schedule (Sequence <S1> 75 units of product 1 – 50 units of product 2 – 50 units of product 3 – 25 units of product 4).

Part	**Cell-1**				
	Mon	**Tue**	**Wed**	**Thurs**	**Fri**
1	80	70	0	0	25
9	0	0	0	105	45
4	120	115	80	80	80
5	80	90	160	55	15
8	0	10	80	10	25
Total	**280**	**285**	**320**	**250**	**190**

Part	**Cell-2**				
	Mon	**Tue**	**Wed**	**Thurs**	**Fri**
2	40	35	0	105	45
3	0	5	40	5	25
6	0	5	40	5	50
7	0	0	0	35	65
Total	**40**	**45**	**80**	**150**	**185**

If we use batch production for the abovementioned situation, 75 units of product 1 followed by 50 units of product 2, 50 units of product 3 and finally 25 units of product 4, would be assembled as shown in Table 1. On Monday only product 1 will be assembled. The first 7 hours of Tuesday will be spent completing the requirement for product 1, after which the assembly line will switch over and produce 5 units of product 2, and so on. This continues as shown in Table 1. The resulting parts requirements are also shown in Table 2. For example, 160 units of part 5 are required on Wednesday (because each of the 40 units of product 2 requires 4 units of part 5). By Friday, a part 5 usage is down to 15 units. There is considerable variation in the daily usage of all parts, as well as the hourly usage. During first hour of Thursday 20 units of part 5 are needed because 5 units of product 2 are being assembled. The usage then drops to 5 units per hour for next 7 hours (because only 1 units of part 5 is needed for each unit of product 3).

Total daily usage for each cell also varies. Cell 1 must have capacity of 320 units per day, which is fully utilized only on Wednesday. The total daily usage for cell 2 ranges from 40 to 185 units, requiring capacity of 185 units. The machine usages also vary. For example, the daily usage on machine XII from Monday to Friday is 200, 205, 240, 240 and 140 units.

Clearly there is lot of variation in the fabrication area. While the schedule at level 1 (the assembly line) is simple, the resulting schedule at level 2 (fabrication area) is very complex. There will also be large inventories of finished products. For example, some of the products produced early in the week may not be required until late in the week, and some orders which require a unit of product 4 will be delayed until product 4 is assembled on Friday.

One balanced schedule would assemble each product every day – 15 units of product 1, followed by 10 units of 2, 10 units of 3 and 5 units of 4, (schedule S2 in Table 4). The daily usage of products, parts and machines are constant. The total daily usage in each cell is constant at 265 units for cell 1 and 100 units for cell 2. This is a characteristic of a balanced schedule i.e. smaller capacity machines are required.

Table 4 given below illustrates product utilization with modified batch production schedule S2 and balanced production schedule S3. Examine S2 more closely by looking at 4-hour time period Table 1. During the 4-hour morning 15 units of product 1 and 5 units of product 2 are assembled, requiring 145 parts from cell 1 and 25 parts from cell 2. During the 4-hour afternoon period product and part requirements change. Cell 2 production increases 300% to 75 units. (Recall we had computed cell 2s capacity to be 100 units per day. We see now that this consists of 25 units for the first 4-hour and 75 units for the next 4 hours). There is still some variability in S2.

Table 4. Modified schedules S2 and S3 for more balanced parts utilization.

Batch Schedule

						S2		S3	
Product	Mon	Tue	Wed	Thu	Fri	1st Half *	2nd Half**	1st Half	2nd Half
1	15	15	15	15	15	15	0	8	7
2	10	10	10	10	10	5	5	5	5
3	10	10	10	10	10	0	10	5	5
4	5	5	5	5	5	0	5	2	3

Part	Cell-1					1st Half	2nd Half	1st Half	2nd Half
1	35	35	35	35	35	30	5	18	17
9	30	30	30	30	30	0	30	15	15
4	95	95	95	95	95	55	40	48	47
5	80	80	80	80	80	50	30	41	39
8	25	25	25	25	25	10	15	12	13
Total	**265**	**265**	**265**	**265**	**265**	**145**	**120**	**134**	**131**

	Cell-2					1st Half	2nd Half	1st Half	2nd Half
2	45	45	45	45	45	15	30	23	22
3	15	15	15	15	15	5	10	7	8
6	20	20	20	20	20	5	15	9	11
7	20	20	20	20	20	0	20	9	11
Total	**100**	**100**	**100**	**100**	**100**	**25**	**75**	**48**	**52**

* 1st half refers to morning four working hours of the shift.
** 2nd half refers to evening four working hours of the shift.
S2- Modified Batch production schedule (15 of product 1 followed by 10 of product 2 followed by 10 of product 3 and 5 of product 4) per day.
S3- Balanced production schedule (In this case following 1-2-3-1-4-2-3-1 25 times).

A more balanced schedule is to sequence product in the order 1-2-3-1-4-2-3-1, and to repeat this sequence again and again. That is assembling one unit of product 1 followed by one unit of product 2, one unit of product 3, and so on. As we see in Table 3 (schedule S3) product, part and machine usage are as constant as they can possibly be.

Quick set-ups are both a requirement and goal for JIT systems. When set-up times are long the batch schedule is used. After setup times are reduced a balanced schedule, S2, is run. As setups are further reduced S3 is used.

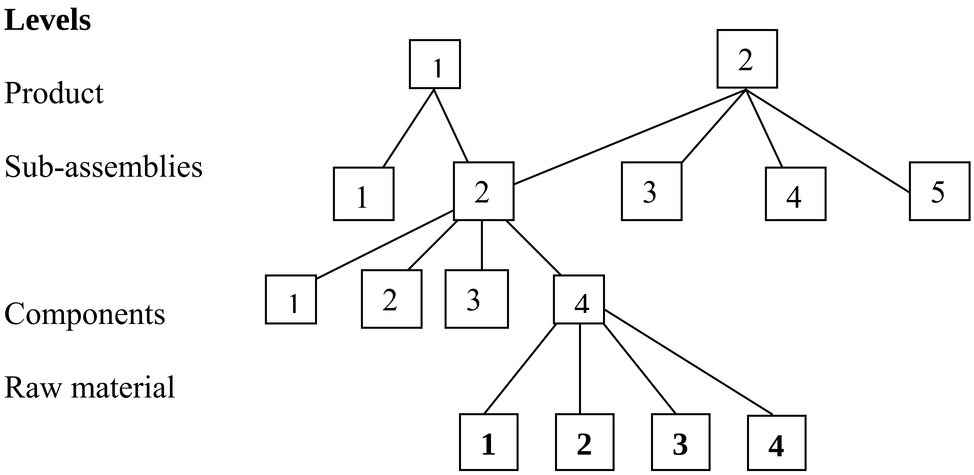


Figure 1. Production levels in a typical manufacturing environment.

In this research we are proposing a model, which will typically deal with a production system shown in Figure 1 which is the pictorial representation of different levels of product formation. Variety of products produced at the final assembly (product) level which will utilize parts produced at the subassembly, component and raw material level. Main objectives of our model are to determine the number of stations and sequence of the models on the final assembly. For the better results we are assuming that production system is a blend of batch and balanced schedule. Batch schedule will bring in the advantage of lower setup time whereas the outcome of balanced schedule will bring the advantage of lower inventory cost. We have assumed that the final assembly is a batch schedule, where the production schedule is dominated by the setup time. Whereas for subsequent level we use balanced schedule, with concurrent schedule we can make sure that inventory level is not high and part production is carefully synchronized with the other stages in the manufacturing. Our proposed model will consider not only physical demand but also the setup costs at the different production level and give us a product sequence at the final assembly level.

CHAPTER 2. LITERATURE REVIEW

2.1 Simple Assembly Line Balancing

Typically an assembly line consists of *n* number of work stations placed along the constantly moving conveyor belt. Value is added toward the desired product at each workstation. Raw material or semi-finished product enters at the one end and the desired product comes out from the other end of an assembly line. Time allocated at each workstation to complete its operations depends on the product demand, $cycletime = \frac{\text{number of hr. available for work}}{\text{demand during that particular period}}$. The decision problem of optimally partitioning the assembly work among the workstations with respect to some objective is known as Assembly Line Balancing Problem (ALBP) (Scholl, 2003).

Minimizing the cycle time for a fixed number of workstations and minimizing the number of work stations required in order to achieve that given output rate are the two main goals in Simple Assembly Line Balancing Problem (SALBP). Dynamic programming and branch and bound are typically the methods used to solve these problems. Canahan et al (2001) considered work fatigue in the assignment of tasks to the assembly line workstations. Physical measure like grip strength capacity, weight, bending required were considered while balancing an assembly line.

2.2 Mixed-Model Assembly Line Balancing

We can find literature on MALB way back to the 1960s Thomopoulos (1970) was the first to develop a heuristic on Mixed-Modeled Assembly line. He focused on general practices in mixed-model assembly line balancing like, to assign work to stations in a manner such that each station has an equal amount of work on a daily or a shift basis.

This paper illustrates how a modification to mixed-model line balancing algorithms can be used so that the stations are loaded more consistently on a model by model basis as well. The objective of the algorithm explained in this paper is to make assignments so that the precedence relation that is given in a precedence diagram is adhered and the idle time associated with station assignments is minimized. They also considered single model line balancing parameters and compared them with those of mixed-model line balancing. Cycle time *c* can be formulated as $c = N\sum_{k=1}^{K} t_k / T$, which can be seen as a desired amount of load time at each workstation. And $\sum_{i=}^{n} p_i = \sum_{k=1}^{K} t_k$ with objective to minimize the idle time associated with a set of stations assignment, $\sum_{i=1}^{n}(c - p_i)$. In the above equations, *n* denotes number of stations and p_i amount of time assigned to i^{th} station for producing each unit. The line balancing procedures used in this paper assigns elements in a serial fashion, searching for a feasible solution by combining elements until an acceptable combination is reached. Thomopoulos (1970) also gives details on how the proposed algorithm can be modified to yield better results in continuous assembly situation. However limitation of this paper was that it does not consider sequence to be followed by different models, as sequence will impact the average utilization of the line and in calculating the workload balance amongst and across the workstations.

Hadi. *et al* stated MALB problem as: Given *P* models, the set of tasks and a cycle time associated with each model, the performance times of the tasks, and the set of precedence relations which specify the permissible orderings of the tasks for each model, the problem is to assign the tasks to an ordered sequence of stations such that the precedence relations are satisfied and performance measures are optimized. To overcome the drawback of excessive number of variables and constraints the authors developed an integer programming model for mixed-model version of the problem in which they utilize properties that prevent the increase in number of variables. Another advantage of this model is that this algorithm can also be used as a validation tool for heuristic procedure developed for the mixed-model problem. In this paper they assumed that performance times, precedence relation, of each model are known and no WIP is allowed, common tasks of different models are assigned to the same stations and parallel stations are not allowed. Furthermore for the precedence relationship diagram they assumed that there are no conflicts in the precedence relations across the model. For example, if model *x* requires that task *b* should follow task *a*, then there should not be any model which requires that task *b* should precede task *a*.

2.2.1 Computer Assisted Mixed-model Assembly Line Balancing

From mid 1970's focus has been built around the performance of computer program, as computer has started to become important tool for computation. Macaskill (1972) says that effectiveness of algorithm depends on the choice of procedure for selecting the appropriate candidate for assignment. Two major factors to consider in this regard are balance efficiency and speed of computation; these factors are inversely proportional to each other. This paper deals with the assembly of different models of the same general product on one production line. Macaskill (1972) is the first to consider MALB concerning about allocation of work to operators and affect of model sequence on assembly performance.

Project scheduling is also an important planning function; it involves scheduling all activities such that total time to complete the entire project is minimized. All these activities must be scheduled in such a way that precedence relationships are not violated. However in real world every activity has resource constraints, sometimes with limited resources and many times we require a combination of these resources to complete any activity. DePuy *et al* (2000) presents the application of computer heuristic solution methodology called COMSOAL (Computer Method of Sequencing Operations for Assembly Lines) to the constrained resource allocation problem.

2.2.2 JIT and Mixed-model Assembly Line Balancing

Just in Time (JIT) has revolutionized the manufacturing world. In the late 1980s everyone was interested in implementing JIT to their manufacturing firm. JIT means producing necessary product of necessary quantity at necessary time. MALB is often used to cope up with the JIT requirements. Miltenburg (1989), talks about the scheduling the products to be assembled on the assembly line. Company goals like, inventory reduction, setup minimization and higher production have impact on the scheduling algorithm. The goals could be leveling the load on each station and keeping constant usage rate of every part used by the line. JIT system is effective only when there is constant rate of usage for all parts. To reduce the variation in the usage of each part it is desirable to sequence products in small lots. This paper assumes that products require approximately the same number and mixes of parts. The objective is to schedule a constant rate of production for each product. Miltenburg (1989) has considered $r_i = d_i / D_T$, where $D_T = \sum_{i=1}^{n} d_i$ is the total demand and d_i is the demand for individual product. Then objective of this schedule is to schedule the assembly line so that the proportion of product *i* produced over a given time period is as close to r_i as possible.

Bard (1989) used dynamic programming (DP) algorithm for solving an ALB problem with parallel work stations, this algorithm considers the tradeoff between minimum number of work stations required and cost of installing additional facility and it also considers unproductive time. Ghosh and Gagnon (1989), Yano and Bolat (1989) focused on job sequencing to support just-in-time delivery of component parts. Roberts and Vill (1970) are first to formulate mixed-modeled ALB to minimize idle time for fixed number of stations. Berger et al. (1992) develops method for solving the problem of minimizing number of stations for a fixed cycle time. Zante-de Fokkert and De Kok (1997) compared several heuristics for minimizing the number of stations. Thomopoulos (1970) considered effects of line balancing decision on the quality of achievable job sequences.

2.2.3 Sequencing Scheduling and Mixed-model Assembly Line Balancing

Merengo *et al* (1999) has discussed balancing and sequencing problems. Minimizing the rate of incomplete jobs, probability of blocking and starvation events, WIP and keeping constant parts usage are the goals of the sequencing process of this model. The authors consider manual assembly system processing time, which is constant for all the stations; that is why task time is considered as a random variable. The authors further assume that one operator performs all assembly tasks allocated to a given station; zoning constraints are not there.

Assembly line balancing refers to the procedure of assigning work to workstations in such a manner as to apportion the assembly work among the stations as evenly as possible without violating the precedence restrictions. Raouf *et el.* (1980) formulated a heuristic method which forms an initial balance and uses either backtracking methods of “trades and transfer” to give the maximum balance. The main feature of this paper was the assignment of priority of elements which means some elements were preferred over the others while assigning to the stations. The heuristic proposed by Raouf *et el.* (1980) consist of two phases: determination of the critical path and the assignment of priority to the work element. According to Nevis (1972) effectiveness of the algorithm depends upon the evaluation of relative merits of alternative paths and reaction to not so promising path. He has introduced a new search procedure, which resembles to branch and bound but is more effective than it from the computational point of view. This paper concentrates on finding out an assignment of tasks which minimizes the number of work stations needed to satisfy a given production rate without violating the technological constraints governing the vendor in which task may be performed.

2.2.4 Setup time and Mixed-model Assembly Line Balancing

Setup time is defined as the time it takes to go from the production of the last good piece of a prior run to the first good piece of a new production run (Trvino *et al.*,1993). Setup cost is a non-value added cost; that explains why many companies look to reduce the setup time. Trvino *et al* (1993) developed a total cost function, which can utilize the setup data. Recommendations were made to reduce the setup, and the information is applied to the total relative cost function to decide if setup time is economically feasible. A general equation was derived expressing relationship between percentage setup time reduction and required investment.

Lee *et al* (1994) applied goal programming to provide insight to setup time and lot size reduction. The objective of their model was to reduce production length, WIP cost and setup cost minimization. Rajendran *et al* (1997) considered static flowshop with sequence dependent setup time of jobs. The main objective of their heuristic is to minimize the sum of weighted flowtime in a sequence dependent setup time scenario. Genreau *et al* (2000) presented a heuristic for the multiprocessor scheduling problem with sequence dependent setup times. The goal of his algorithm was to minimize overall processing time by determining assignments of jobs to machine and cyclic sequence of jobs on each machine.

2.3 Limitations of the Existing Research

From the literature survey we observe that although mixed-model assembly line balancing has been given much deliberate attention, problem of sequencing the different models to reduce overall variability among the work assigned to workstation has not been given a proper justice. In this research we try to solve MALB with sequencing of different models, which will reduce workload variation at assembly level and also at the preceding levels.

Even though some researchers have worked on multi level production system, focus was mainly on the sequencing and scheduling. Miltenburg *et al* (1989) considers four level production systems where production is initiated by one level's requirement which is also an output requirement for the next level. As a result, the final assembly line is the focus of the control. Determination of cycle time, sequence of stations, line balancing and sequence schedule for producing different products on the line are the mains aim of their research.

Miltenburg *et al* (1989) further explains, if each product on final assembly line requires same sort of sub-assemblies, components and raw materials; then, variation at levels 2, 3 and 4 would be the same. Hence those levels can be ignored when developing the product schedule. On the other hand if we have significantly different sub- assembly, components and raw material requirements. Then variation must be considered while selecting a production schedule. While addressing the production schedule Miltenburg *et al* (1989) did not consider the impact of setup cost at sub-assembly and component level on the final assembly schedule.

Lee (1992) used goal programming for the decision making for conflicting objectives like lot size and setup time reduction, to decrease the inventory level and to increase the flexibility of the manufacturing system. Primary objective of their research is to develop a goal programming model for the lot size and setup time decision to choose the best solution out of multiple conflicting goals in a JIT environment. However they did not incorporate important factors like capacity, balancing and scheduling on their objective function.

Similarly from the above literature review we can summarize what has been done in the field of mixed-model assembly line balancing and how our research differs from the existing research. In this research we first balance work load across the final assembly level for all the product types then compute a sequence which will ensure uniform consumption of units at all the production levels. Then we use the closest insertion heuristic developed Ronald *et al* (1993) and modify according to our requirement to minimize the sequence dependent setup time at the final assembly level. Our final production schedule is blend of balanced and batch production as output from the closest insertion heuristic is semi-batch production at the final assembly level. This complete summary can be seen in Table 5.

Table 5. Comparison between existing research and our research

Author	**Line Balancing**	**Setup Time Minimization**	**Sequencing**	**Multilevel Production System**
Miltenburg (1989)	**X**	**--**	**X**	**X**
Lee (1992)	**--**	**X**	**--**	**--**
Fokkert (1997)	**X**	**--**	**--**	**X**
Mingzhou (2002)	**X**	**--**	**X**	**--**
Matanachai (2001)	**X**	**--**	**--**	**X**
Our Paper	**X**	**X**	**X**	**X**

CHAPTER 3. MODEL FORMULATION

From the literature review it is clear that not much attention is given to mixed-model assembly line balancing with multi-level scheduling and setup time reduction. Thomopoulos (1970) effectively demonstrated the benefits of balancing workload across stations for each model; Matanachai (2001) extended the original algorithm and addressed diversity of processing times among different stations, within-station diversity, and workload balance.

The assembly line of AC Manufacturing Company in Fort Smith, Arkansas is a typical example of a mixed model assembly line. This assembly line with its 19 workstations is capable of producing 16 different models simultaneously on the final assembly line. Figure 2 shows typical products produced at this facility. The company manufactures various models with different customer requirements and capacities; and it ships the products to domestic and international markets on daily basis. The manufacturing system consists of a traditional assembly line with semi-paced conveyors. The speed of the conveyor is set based on the cycle time of the models being assembled. Component parts or subassemblies are either built in-house or purchased from outside vendors, which are supplied to main assembly line with the help of forklifts.

The company follows a Make-to-Order production system, where customer/dealer orders guide the production schedule. Requirements for every product are determined at the start of every month based on the demand for that particular month. Recently with the implementation of “Lean Manufacturing” the company has shifted from batch production to blend of batch and balanced mixed-model assembly system. The new practice helps them keep plant load even and effectively manage the production and inventory.

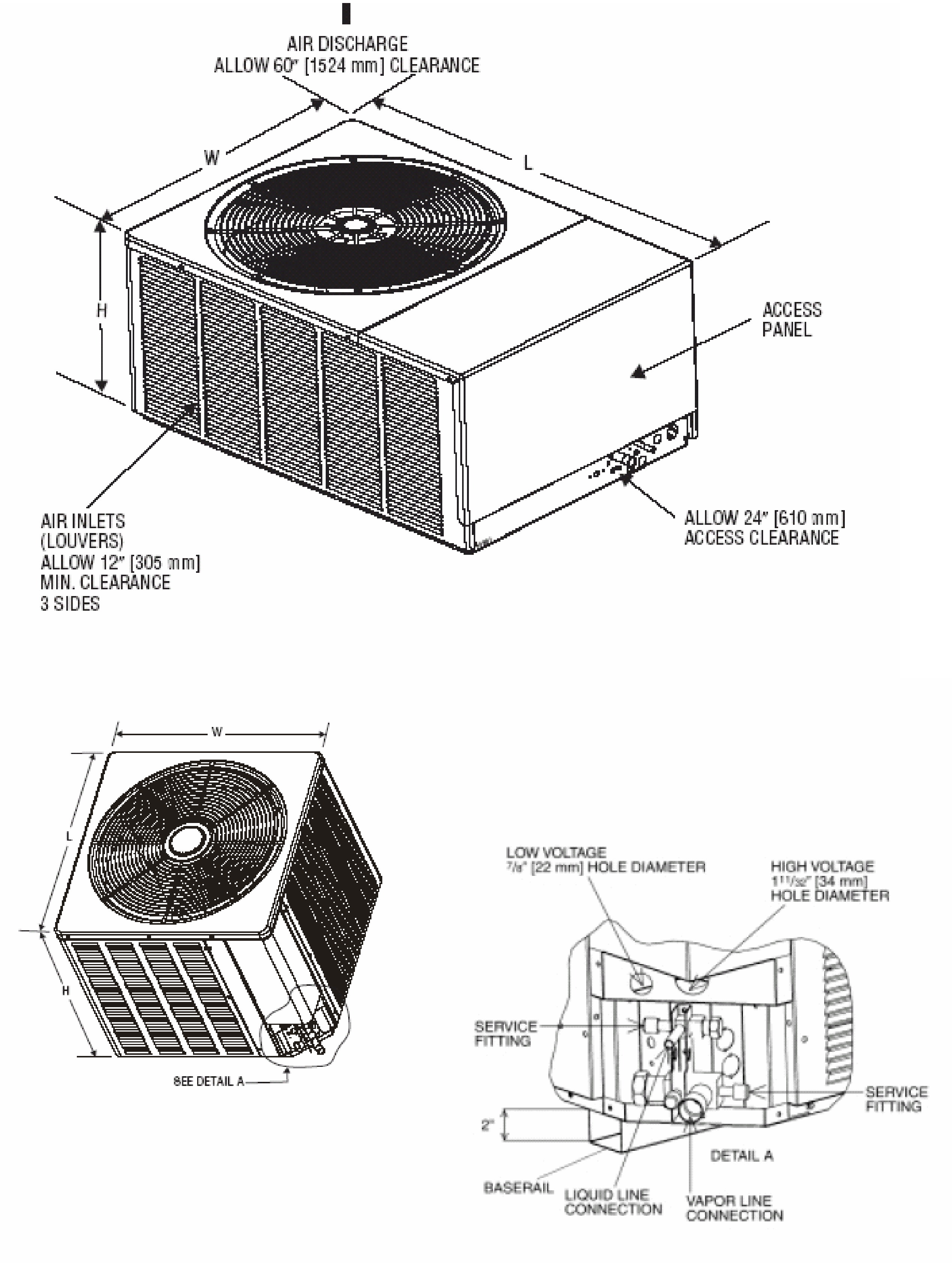


Figure 2. Product mixes which can be produced on the same assembly line.

3.1 Assumptions

- Each product (level 1) is made up of a variety of subassemblies (level 2) which are manufactured from different components (level 3) which, in turn are products of some kind of raw materials (level 4).
- Setup time depends on the sequence of the product, in addition to the product itself.
- Output quantity at the lower level is guided by the requirement of the upper levels.
- A batch schedule is used at the assembly level; it means that a product is produced in small batches so that production mix is synchronized with the market demand without increasing the inventory level.
- Products are produced in the same sequence, which is repeated *n* times.
- Single precedence diagram is used at the assembly level.
- One worker is assigned per station, and travel distance is limited for every work station, meaning that a worker cannot travel out of pre-specified boundaries even in the case of unfinished jobs.
- Work on unfinished job is done at separate or repair station.
- Maximum number of stations is specified exogenously.
- Cycle time is a function of demand.
- Assembly is done on the constantly paced moving conveyor; speed is guided by the cycle time.

3.2 Notations Used

We assume that n_i models will be produced on an assembly line, n_σ would be the number of tasks and K is the total number of stations. The following notation is used for further discussion at the mixed-model assembly line balancing stage.

y Level number (y = 1, Product; y = 2, sub-assembly; y = 3, component; y = 4, raw material);

i^y Product type at level y, $i^y = 1, 2, 3, \ldots, n_y$; for level = 1 we denote $i^y = i$;

$p_{\sigma i}$ Processing time for product i of task σ at level 1;

n_σ Total number of tasks required to produce a product at the final assembly level;

k Station number, $k = 1, 2, 3, \ldots, K$;

C Cycle time;

α Flexibility factor;

PW_σ Positional weight for each task;

$S(\sigma)$ Set of successors of task σ;

$IP(\sigma)$ Set of immediate predecessor of task σ;

PT Production time available;

L_i List of all tasks in non-increasing order according to their positional weights at the product level, for all product type i;

$m_{k,i}$ Number of tasks assigned to workstation k for product type i, $k = 1, \ldots, K$ and $i = 1, 2, 3, \ldots, n_1$;

$L_{opt,i}$ List of tasks been assigned;

Seq_i Candidate optimum sequence for product i, $i = 1, \ldots, n_1$;

var_i Sum of difference between cycle time and total time of all the task assigned to work station k, $k = 1, \dots, K$ over all work stations;

SE_{opt} Final optimum sequence;

3.3 Mixed-model Assembly Line Balancing Heuristic

This research is divided into three different stages with main objectives.

1. Balancing mixed-model assembly line.
2. Generating a schedule to minimize the inventory level.
3. Setup time reduction.

Based on the literature review and the observation of Matanachai (2001) we include the traditional objective function of minimizing the sum of absolute deviation of actual utilization at each work station in our model. Balancing workload among and across the workstations for all the models is our first objective function.

Our objective function of minimizing variation of sum of the task times of the tasks assigned at any station and the cycle time for all the products can be presented as,

$$min \sum_{i=1}^{n_1} \sum_{\sigma=1}^{n_\sigma} (CX_{i,\sigma} - p_{i,\sigma} X_{i,\sigma}) \tag{3.3}$$

st,

$$\sum_{i=1}^{n_1} p_{i,k} X_{i,k} \le C \qquad \forall k \tag{3.3.1}$$

$$\sum_{i=1}^{n_{i1}} p_{i^y,k} X_{i^y,k} \le \alpha n_{i^y} C \qquad \forall k \tag{3.3.2}$$

$$\sum_{k=1}^{K} X_{i,k} = 1 \qquad \text{For } i=1, 2, 3, \dots, n_1 \tag{3.3.3}$$

$$X_{vk} \leq \sum_{v=1}^{k} X_{ur} \qquad \forall i,k \text{ and } \forall r \in IP(\sigma) \qquad (3.3.4)$$

$$XT_{1k} = k \qquad \forall k \qquad (3.3.5)$$

Constraint 3.3.1 ensures that the sum of task times for the set of tasks assigned to each workstation does not exceed the cycle time. LHS of the constraints look at each task. If a task assigned to station *k,* its task time is added to the sum for that station. Final sum is compared to the allowed cycle time *C* to ensure time feasibility. Similarly 3.3.2 ensures the set of tasks assigned to work station *k* does not exceeds the product of numbers of different units assigned to that workstation (In our case it will be always n_i, as we assume that every product is assigned to every workstation) flexibility factor *α,* and cycle time, $\alpha * n_i * C$. Constraint 3.3.3 ensures that each product is assigned to exactly one station. Constraint 3.3.4 will make sure that no precedence relation is violated. If task *v* is assigned to workstation *k* then its immediate predecessor is assigned to somewhere between station 1 and *k.* For example if we have a case where task 3 must precede task 4 and we have three workstations to accommodate these task. For this precedence restriction equations can be written as follow.

$$x_{41} \leq x_{31}$$

$$x_{41} \leq x_{31} + x_{32}$$

$$x_{41} \leq x_{31} + x_{32} + x_{33}$$

Since 4 must be assigned to one of the three stations only one of the three equations will have nonzero LHS. Equation 3.3.5 ensures that we can assign exactly *k* products to *k* working stations.

To solve the above model we develop a heuristic named as Modified Ranked Positional Weight (MRPW). The MRPW procedure has two stages: the first stage produces the initial optimum solution for the individual product and the second stage refines the initial solution in order to find the best solution for all product models. Output of this procedure will give us the sequence of tasks which will minimize the variation between cycle time and sum of tasks assigned to all the workstations. The procedure is divided into two stages, as described below.

Modified Ranked Positional Weight Method (MRPW)

Stage-1

For each model,

Step1 – A task is weighted based on the cumulative assembly time associated with itself and its successors.

Step2 – Each task is arranged in non-increasing order of its weight.

Step3 – Assign the tasks to the available feasible station according to the sequence in *step-2*; continue assigning task to the station until further assignment would violate the cycle time constraint.

Step4 – Open a new workstation if there are no possible assignments for the current workstation. If no task is left for the further assignments we have the individual optimum sequence.

Stage-2

Step1 – Compute the total variation for each product summing over all the workstations according to their individual sequence. Select the sequence of a product with the least variation as the candidate optimal sequence.

Step2 – Compute the variation of each model based on the candidate optimal sequence. If this variation is less than or equal to the total variation associated with the individual sequence computed at ***stage-1*** for all models, make this candidate optimal sequence as your final sequence. Otherwise go to *step3*.

Step3 – Select the sequence with the least variation from the list of remaining individual sequences as the candidate optimal sequence and go to *step-2*. In the case that the list is empty and still no optimal sequence is found, choose the individual sequence which will give the least total variation over all products as the final sequence.

The input data required by the MRPW and the procedure for the heuristic**:**

- Number of different products produced at the final assembly level, n_1;
- Number of tasks required to produce one product at the final assembly level, n_σ.
- Cycle time (*C*), based on demand and production capacity.
- Precedence relationship. It is assumed that it is possible to draw a single, non cyclical precedence relationship which is common for all the products that are produced at the final assembly level. This is quite reasonable, if we consider fixed machine location for all the products which pass through the assembly line under consideration.
- Processing times ($p_{\sigma,i}$). The expected time required to perform each task on each product type i is known. The time required to perform a task on models which do not require that task equals zero.
- *S(σ)*, set of successors of task σ which can be derived from the precedence diagram. For example if $\sigma_2 \in S(\sigma_1)$ it indicates that σ_2 can not begin until σ_1 is

completed. Similarly $IP(\sigma)$ is a set of immediate predecessor of task σ. This relation is common for all the product types as we are assuming a common precedence diagram.

A more elaborated version of the MRPW procedure is presented below**.**

Stage-1

For each product type i^1 perform the following steps:

1. For all tasks $\sigma = 1, \ldots, n_\sigma$, compute $PW_\sigma = p_\sigma + \sum_{r \in S(\sigma)} p_r$.
2. List all the tasks in non-increasing order of their weights and let L be such a list.
3. Initialize workstation $k=1$, optimum sequence array $L_{opt,i}$ = Φ, temporary assignment array $Temp = \Phi$, and the number of tasks assigned to any workstation, $m_{k,i} = 0$.
4. Set the flag for all the elements in $L = 1;$
5. Select the first eligible task with flag = 1 from L, say task σ, and assign it to the first feasible workstation k by setting $X_{\sigma,k} = 1$ if IP(σ) ϵ L_{opt,i^1} and add σ to *Temp*.
6. Check cycle time constraint $\sum_{J_\sigma \in Temp} p_{J_\sigma} X_{J_\sigma,k} \leq C$ for $\forall k$, (*if* task σ is assigned to workstation k, $X_{\sigma,k} = 1$ and 0 otherwise). If satisfied delete task σ from L, add the same task to L_{opt,i^1} and increment m_{k,i^1} by one*,* m_{k,i^1} = $m_{k,i^1} + 1$. Otherwise set the flag of $\sigma = 0$. If $L \neq \{\}$ and not all elements in L having flag = 0, go to Step 5.

7. Open a new workstation, $k = k + 1$, reset $Temp = \Phi$ and go to *step-4.* Otherwise, task assignments are complete and name $L_{opt,i}$ as Seq_i as your optimum individual model sequence.
8. Store completed sequence for all the products in a new array, $Aseq = \{Seq_1, Seq_2, Seq_3, ..., Seq_{n1}\}$

Stage-2

1. Calculate the sum of difference of cycle time and sum of task times assigned to workstation *k* based on the sequence obtained at stage-1 over all work stations for each product type, $var_i = \sum_{k=1}^{K}\sum_{\sigma}(C - p_{\sigma,i}\, X_{\sigma,k})$, form an array and store the variations for all the products, $V_{ary} = \{ var_1, var_2, var_3, ..., \mathrm{var}_{n_{i1}} \}$. Make a duplicate copy of $V_{ary} = DV_{ary}$ and compute the total variation $CV_{\mathrm{ary}} = \Sigma_{\mathrm{i}}\mathrm{var}_{\mathrm{i}}$.
2. Select product *b* from DV_{ary} so that, $b = \{ i \mid var_i \leq var_j \,;\, j \neq i \;\&\; i, j = 1, ..., n_i\}$ and delete var_b from the array DV_{ary}.
3. Set sequence associated with *b* as your candidate optimal sequence for all the products, $SE_{can} = Seq_b$
4. Calculate new variation for all the products by following SE_{can} $var'_i = \sum_{k=1}^{K}\sum_{\sigma}(C - p_{\sigma,i}X_{\sigma,k})$, store new variations in another array $NV_{ary} = \{ \mathrm{var}'_1, \mathrm{var}'_2, \mathrm{var}'_3, ..., \mathrm{var}'_{n_i} \}$ and compute the new total variation over all products, CNV_{ary}. Compute $R_{ary} = NV_{ary} - V_{ary}$ if all elements in $R_{ary} \leq 0$, stop and Seq_b is the optimal sequence which gives you the lowest variation for all the models that

are produced on the assembly line. That is, $SE_{opt} = Seq_b$. Otherwise store values of CNV_{ary} in R'_{ary}. If $DV_{ary} \neq \Phi$. go to *step-2*.

5. If we do not find a product sequence which is equal to or better than the optimal solution at step-4 we will select product a such that, $a = \{i \mid R'_{ary,i} \leq R'_{ary,j};\ j \neq i\ \&\ i, j = 1, \ldots, n_i\}$, $i = a$ will give you the optimum variation for this assembly line. Make $SE_{opt} = Seq_a$ as your final optimal sequence. Stop.

We now have the solution to balance the mixed model on the assembly level. With this solution we try to put some practical scenario while assigning the tasks to the workstations. In a steady-paced moving conveyor assembly line cycle time allotted to the work station is controlled by the length of that work station. It is not always possible for a worker to complete his duties within the allotted cycle time, resulting in incomplete tasks. Those incomplete tasks are completed towards the end of the assembly line. Extra resources and time is consumed while doing that. To avoid this, companies prefer to give some flexibility to the worker by increasing the length of the workstations. We run experiments through our balancing heuristic with different flexibility factors and study its effect on number of incomplete tasks, variation and total cost function. The results are shown in the section 5.2.2.

3.4 Scheduling Heuristic

We now have mixed model balancing at the final assembly level and our second objective is to minimize inventory at the production levels, which can be achieved by uniform utilization of part supplied and utilized at the lower levels. To this end, a scheduling heuristic is developed.

In addition to the notations used at the mixed-model assembly line balancing we need the following notation for the scheduling heuristic:

s Index of stage required to sequence all the products that are produced at the final assembly level, $s = 1, 2, ..., S;$

n_y Number of output at level y, y -$1, 2, 3, 4;$

i^1 Index for product produced at final assembly, i^1- $1, 2, 3, ..., n_1;$

d_{i^1} Demand of product i at the final assembly level, $i^1 – 1, 2,..., n_1;$

t_{i^y,y,i^1} Number of units of output at particular level y used to produce one unit of product i^1 at the final assembly, $i^y=1, 2, ...,n_y$; $y = 2, 3, 4;$ $i^1 = 1, 2, ..., n_1$;

For level one that is $y = 1$

$$t_{i^1 1} - \begin{cases} 1 & \text{If } i^y = i^1 \\ 0 & \text{otherwise} \end{cases}$$

$d_{i^y y} = \sum_{h=1}^{n_1} t_{i^y,y,h} d_{h1}$ Demand for output i^y at level y, $i^y = 1, 2,..., n_y$; $y = 2, 3, 4$

$D_y = \sum_{i^y=1}^{n_y} d_{i^y,y}$ Total demand for production level y, $y = 1, 2, 3, 4;$

$r_{i^y y} = d_{i^y,y} / D_y$ Ratio of level y production devoted to output i^y, $i^y = 1, 2,..., n_y;$

w_y Weight given to particular level y, which is assumed as either 0 or 1 ;

$x_{i^1,1,s}$ Number of units of product i^1 produced during stages $1, 2, ..., S$;

$\mathrm{x}_{\mathrm{i}^{\mathrm{y}},\mathrm{y},\mathrm{s}} = \sum_{h=1}^{n_1} t_{i^y,y,h} x_{h,1,s}$ Number of units of output i^y at level y produced during stage s, s =1, 2, …, S ($x_{i^y,y,0} = 0$ as zero units are produced at stage 0);

XT_{ys} $\sum_{i^y=1}^{n_y} x_{i^y,1,s}$ Total production at level y during stage s, s = $1, 2, 3, \ldots, S$;

As we are looking for the blend of balanced and batch production schedule to take advantage of both lower inventory levels and setup time. We have developed two different production schedules; one for final assembly line and one for the subsequent production levels. At the final assembly where reduction of setup time is more significant we focus on reducing setup time using the closest insertion heuristic and we follow balanced schedule obtained through our heuristic for the subsequent production stages.

3.4.1 Scheduling for All the Production Levels

Scheduling products would be easy if the required product mix is approximately the same at every level of production. However that is not the case in a mixed-model scenario, in which there are only a few activities that are common, and variation in the product requirement at the assembly level affects the production schedule of the later levels. When we consider the case presented above, variation at all levels in the system must be considered while scheduling. We consider the scheduling heuristic developed by Miltenburg *et al* (1989) and modified it with an extension of balancing workload distribution across the workstations. While considering assignment at every workstation we will select a product which minimizes the variation in production with respect to demand at stage *s*. Stage here is different from workstation, it is virtual allocation of the products to the sequence array. Detailed explanation on how we calculate total number of stages is given in the next paragraph.

This research considers the impact of final assembly sequence on the bill of materials in a multi-level production system. Our objective function addresses the issue of synchronizing production with demand at all level. If production were strictly synchronized with demand then after *s* stages the total output $\mathrm{x}_{\mathrm{i}^{\mathrm{y}},\mathrm{y,s}}$ of part i^y at level y would be. $XT_{\mathrm{y,s}}\mathrm{r}_{\mathrm{i}^{\mathrm{y}},y}$ However, equality is not always possible so we strive to schedule the system in a way making $\mathrm{x}_{\mathrm{i}^{\mathrm{y}},\mathrm{y,s}}$ close to $XT_{\mathrm{y,s}}\mathrm{r}_{\mathrm{i}^{\mathrm{y}},y}$ for each i^y, y and s. To synchronize part supply and workload, we try to level the consumption of the parts in the sequence. The desired number of parts consumed in the first *s* positions (stages) at level *y* is $sr_{i^{y}y}$. Let *GC* be the greatest common factor of all $d_{i,y}$. In the JIT production system we aim for constant utilization of raw materials, we can achieve that by repeating the sequence with *S* products scheduled GC times. Where, $S = D_y/GC$. The cumulative part consumption for part i^y equals to $\sum_{s=1}^{n_y} x_{i^y,y,s}\, t_{i^y y i^1}$,

The objective function can be written as,

$$\text{Min } \sum_{s=1}^{S}\sum_{y=1}^{4}\sum_{i^y=1}^{n_y} w_y\,(x_{i^y,y,s} - XT_{y,s}r_{i^y})^2 \tag{3.4}$$

With this objective function we minimize the sum of squared deviation of actual production from the actual demand of any product for the level under considerations. Here, w_y is used to determine whether deviation of variation at the particular level is important or not.

The objective mentioned in Equation (3.4) is subjected to the following constraints:

$$\sum_{i\in[1,\ldots,n_i]} x_{i^y,y,s} = S \tag{3.4.1}$$

$$x_{i^y,y,s} - x_{i^y,y,s-1} \le 1 \qquad x_{i^y,y,0} = 0 \qquad \text{For } \forall i^y, \forall s \tag{3.4.2}$$

$$x_{i^y,y,s} - x_{i^y,y,s-1} \ge 0 \qquad \forall i^y, \forall s \tag{3.4.3}$$

$$0 \le x_{i^y,y,s} \le d_{i^y,y} \qquad \forall i^y, \forall s \tag{3.4.4}$$

Constraint 3.4.1 ensures that exactly *S* products are scheduled during stage *1* to *S*. Constraints shown by equations (3.4.2) and (3.4.3) ensure that it is not possible to schedule less than zero units and more than one unit or fraction of any unit i.e. we can schedule only one product at a time to any given workstation. Constraint (3.4.4) indicates that the number of the sequenced model i^y at any station *s* should be less than the total demand for that product.

For the scheduling purpose we have the following decision rule at each stage. We calculate variation for every level and schedule a product which will give a cumulative minimum variation for the stage under consideration. We use two different equations to calculate the variation. Equation (3.4.5) gives variation at the final assembly level,

$$V_{i^y,i^1,s} = w_1(x_{i,^y i^1,s-1} - sr_{i^y,1}) \tag{3.4.5}$$

However, our objective is to synchronize production with demand at all levels. Similarly for subsequent stages variation can be calculated as,

$$\sum_{y=2}^{4} \sum_{h=1}^{n_y} w_y [(x_{h,y,s-1} + t_{h,y,i^y}) - (XT_{y,s-1} + \sum_{h=1}^{n_y} t_{h,y,i^y}) r_{h,y}]^2 \tag{3.4.6}$$

We can combined these equations for the complete production system this function becomes,

$$CV_{i^y,y,s} = \sum_{i=1}^{n_i} x_{i^y y,s} t_{h,y,i^y} - sr_{i^y,y} + \sum_{y=2}^{4} \sum_{h=1}^{n_y} w_y [(x_{h,y,s-1} + t_{h,j,i^y}) - (XT_{y,s-1} + \sum_{h=1}^{n_y} t_{h,y,i^y}) r_{h,y}]^2$$

$$(3.4.7)$$

The weight, w_y determines the importance of variability at each stage. We can adjust the variability as per the importance of a particular level. For example if we put $w_4 = 0$ that means variability at this stage does not come into picture; hence, we can ignore that term. The mathematical expression for this decision rule is: schedule the product i with the lowest *CV*.

Heuristic Procedure

Input data required for the balancing heuristic

- Number of different products assembled at the final assembly level, i^1. $i^1 = 1, 2, 3, \ldots, n_1$;
- Demand for the products produced at the final assembly level, $\mathrm{d}_{i^1,1}$ for all i^1;
- Total number of product types produced at level y, n_y. For example for one unit at final assembly level we need 3 units from subassembly, 4 units from component and 3 from raw material. We will have $n_2 = 3$, $n_3 = 4$, $n_4 = 3$; $y = 2, 3, 4$;
- Number of units of output i^y at level y used to produce one unit of product i^1, t_{i^y,y,i^1}, $i_y = 1,2,3, \ldots, n_y$;
- Weights assigned as per the priority of the schedule, $w_y = \{0,1\}$, $y = 1, 2, 3, 4$;

Procedure

1. Initiate first stage for scheduling operations, $s=1$, calculate the variation between actual number of products produced and theoretical target at the final assembly for product i_y using $V_{i^1,1,s} = w_1(x_{i^1,1,s-1} - sr_{i^1,1})$, for $i_1 = 2, \ldots, n_1$. ($x_{i^1,1,s-1} = 0$, as there will be no product scheduled at 0th stage);

2. Initiate an array to store the total variation summed over all the levels for each individual product, $A_{cv,i} = \Phi$;

3. Similarly compute the variation for each subsequent level and for all the products that are produced on that level from the equation given below,

   $V_{i^y,y,s} = w_y [t_{i^y,y,i^1} - (\sum_{i^y=1}^{n_y} t_{i^y,y,i^1}) r_{i^y,y}]^2$ for $y = 2, 3, 4$ and $i^y = 1, 2, \ldots, n_y$, then compute the sum of variation for all the levels at stage 1 using $CV_{i^1,y,s} = \sum_{y=2}^{4} V_{i^1,y,s} + V_{i^1,1,s}$, and save the computed value in the array, $A_{cv,i}$;

4. Find the product type with the minimum variation for the stage under consideration, $Y = \{ i \mid A_{cv,i} \leq A_{cv,j} \; ; j \neq i \; \& \; i, j = 1, \ldots, n_y \}$ and add it to the final sequence array MA_{cv}, then open a new stage for the next allocation, i.e., $s = s+1$ and reset $A_{cv,i} = \Phi$;

5. For $s \geq 2$ Calculate the variation at the final assembly level using $V_{i^y,1,s} = w_1 (x_{i^y,1,s-1} - s r_{i^y,1})$ for all i^y, where $x_{i^y,1,s-1}$ is equal to number of times product i^y appeared in MA_{cy};

6. Similarly calculate the sum of variation at all the subsequent levels using the formula given below,

   $$V_{i^y,y,s} = w_y \sum_{y=2}^{4} \sum_{i^y=1}^{n_y} [(x_{i^y,y,s-1} + t_{i^y,y,i^1}) - (XT_{y,s-1} + \sum_{i^y=1}^{n_y} t_{i^y,y,i^1}) r_{i^y,y}]^2$$

   , for $i^y = 1, 2, 3, \ldots, n_y$. If $s \leq S$, compute the sum of variation $CV_{i^y,y,s} = \sum_{y=2}^{4} V_{i^y,y,s} + V_{i^y,1,s}$,store the computed value in $A_{cv,i}$ and go to *step 4*; otherwise, assignment is complete and stop.

We ensure smooth part consumption as an outcome of this heuristic but at the expense of frequent setup requirements. This method will have limited advantage in the scenario with a higher sequence dependent setup time. To set the balance between advantages of lower inventory with lower setup times we implement the closest insertion heuristic to minimize sequence-dependent setup time at the final assembly level. Detailed explanation of this heuristic is presented in the next section. We further discuss the reasoning behind this methodology in the example presented in the next chapter.

3.4.2 Scheduling for the Final Assembly Level

The JIT system is successful in both optimizing the material delivery timing and minimizes the inventory quantity by specifying delivery of materials only on an as needed basis. To decrease the inventory levels while increasing the manufacturing flexibility, JIT focuses on lot size and setup time reduction. In our model setup cost is expressed in terms of time units. In addition, we consider that if a job is followed by another job, a setup time independent of the machine is incurred. With the balanced schedule obtained through the scheduling heuristic discussed in the previous section we ensure smooth product consumption, resulting in lower inventory holding cost. However due to frequent product changes we incur extra cost in the form of time lost in frequent setups. So, the focus here is to reduce the number of setups while maintaining the advantage of lower inventory levels

We use the following assumptions while developing this model.

Assumptions:

- Number of products to be produced is known.
- Each work station can hold only one product at a time.
- Sequence of workstations is known and it is the same for all the products produced.
- Setup time given is the sum of setup times over all the work stations; an *n*-job *m*-machine problem is converted to an *n*-job *1*-machine problem.
- Machine breakdown is not considered.
- Preemptions are not allowed.

In addition to the notation used in the previous section we use the following set of notation for this heuristic. As we are considering setup minimization only at the final assembly level *i* will denote a product produced at the final production level.

$RT_{ik} = \sum_{\sigma=1}^{n_\sigma} p_{\sigma,k} X_{\sigma,k}$ Average runtime of one unit of product type *i* at work stage *k,*

where $X_{\sigma,k} = 1$ if task σ is assigned to station *k* and *0* otherwise.

$SET_{i,j}$ Setup time of product *i* if it directly precedes product *j;*

$$X_{i,j} - \begin{cases} 1 & \text{If product } j \text{ succeeds product } i \\ 0 & \text{Otherwise} \end{cases}$$

ST_i Planned setup time for product *i* at final assembly level, the sum of setup times over all the workstations;

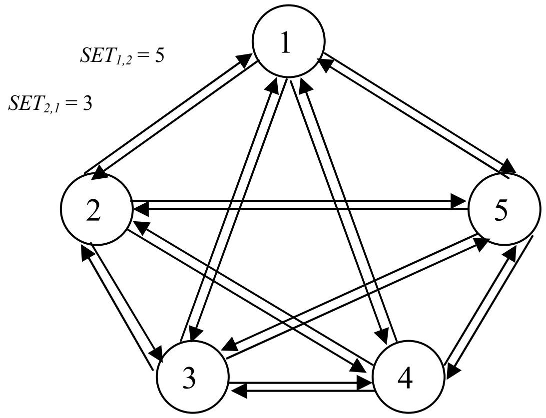
CST_i Current setup time of product *i;*

$MINST_i$ Minimum achievable setup time for product *i* over all the workstations*;*

CO_i Cost of raw material needed per unit of product i^y*;*

CML Current mixed-model production length;

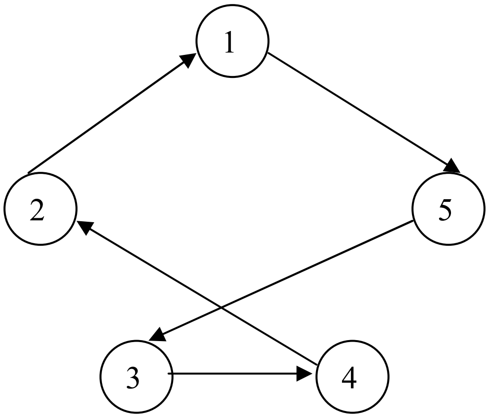
In this model total processing time is fixed by unit processing time and batch size. However, total setup time is dependent on the product sequence. For instance, if you take an example of wire drawing machine, copper wires generally require PVC coating of different color for every bundle. There is a setup time involved with every change in the color coating die. As an engineer you will have to decide which color to choose to replace the current color. For example white color is avoided to replace previous black, as it would incur higher setup (die cleaning) time and would waste larger material during the pilot run. This research focuses on choosing a right sequence to minimize the sequence dependent setup time. In Figure 3(a) we illustrate setup time involved in product change, in which a node represents a product. If product *i* is followed by product *j* then there is a directed link between them and it is denoted by $SET_{i,j}$. Figure 3(b) shows the possible sequence of jobs to be assigned on the assembly line to minimize the setup time. Total setup time for this schedule can be written as $SET_{1,5} + SET_{5,3} + SET_{3,4} + SET_{4,2} + SET_{2,1}$.



(a) Complete Setup time graph

Figure 3. Setup time illustration

Figure 3, Continued



(b) Possible Sequence

For the flow shop scheduling problem we use the closest insertion heuristic search method to sequence the products on the final assembly line in such a way that the resultant sequence will give us the least possible setup time among all the possible sequences.

Objective function shown in equation 3.4.2.1 will insure minimum setup cost for the sequence dependent setup time.

$$\text{Min} \sum_{i^y}^{n_y} \sum_{j=1}^{n_y} SET_{i,j} X_{i,j} \qquad \text{where } i \neq j \qquad 3.4.2.1$$

Only one product can precede the other product on the assembly line, this constraint can be written as,

$$\sum_{i}^{n_1} X_{i,j} = 1 \qquad \text{for } \forall i \qquad 3.4.2.2$$

Similarly only one product can succeed the other because one workstation can hold only one product.

$$\sum_{j}^{n_y} X_{i,j} = 1 \qquad \text{for } \forall j \qquad 3.4.2.3$$

Setup time reduction is a key practice at which almost all manufacturers must excel in their pursuit of manufacturing excellence. SMED (Single Minute Exchange of Dies) is the most promising technology which has been embraced by most of the modern manufacturing companies. SMED uses four key steps to reduce setup times,

1. Suppress useless operations. Convert internal setup operations to external setups.
2. Use automation for quick response to change.
3. Eliminate adjustments and trials, first time on target.
4. Simplify fittings and tightening.

From these characteristics it is evident that there are some engineering restrictions on setup time reduction. There is certain limit for setup time reduction; this constraint can be written as:

$$ST_{i,k} \geq MINST_{i,k} \qquad \text{for } \forall i; \forall k \qquad 3.4.2.4$$

As discussed earlier for each item i we must produce $d_{i,1}$ items per given period. GC is the greatest common factor of all $d_{i,1}$. We can achieve uniform utilization by repeating the cycle, $S = D_y/GC$, GC times to satisfy the total demand for the period under consideration. $S = \sum_{i=1}^{n_i} R_{i^1}$, items are produced at each cycle, where $R_i = d_{i,y} / GC$. Overall production requirements for the given period PT is considered as system constraints. Based on assembly line balancing constraint, the length of all the work stages should be identical. Equation 3.4.2.5 shows total production time per workstation.

$$\sum_{i=1}^{n_1} (ST_{i,k} + \sum_{\sigma=1}^{n_\sigma} p_{\sigma,k} * X_{\sigma,k}) \qquad \text{for } \forall k \qquad 3.4.2.5$$

As we discussed earlier production run repeats for *GC* cycles. Therefore, long run operation time for all work stations for one round of mixed model production should be identical. This constraint can be written as:

$$\sum_{i=1}^{n_i}(ST_{i,k} + \sum_{\sigma=1}^{n_\sigma} p_{\sigma,k} * X_{\sigma,k}) = \sum_{i=1}^{n_i}(ST_{i,k+1} + \sum_{\sigma=1}^{n_\sigma} p_{\sigma,k+1} * X_{\sigma,k}) \qquad \text{for } \forall k \qquad 3.4.2.6$$

A sequence dependent setup time problem becomes increasingly difficult to solve as the problem size increases. For large problems, optimal solutions are difficult to obtain. Heuristic is a common way to solve these kinds of problems. We start scheduling for this heuristic by scheduling the product with the highest demands first. We always schedule R_i units of the selected product to avoid frequent setup changes and by doing that we are also keeping inventory level to a reasonable level. We now have n_l-1 products to schedule on the final assembly, adding a new product at each stage. Thus sequence grows one product at each stage until we schedule all n_i products and then we repeat the same sequence *GC* times. We schedule the product to the next stage which will require least setup time if preceded by the product scheduled at the previous stage. The original heuristic is developed to optimize the route for a traveling salesman problem. Thus the original heuristic is mainly effective with symmetric cost matrix. Symmetric cost implies the same cost is incurred if event x is followed by the event y and vise versa; in our case it will mean $SET_{x,y} = SET_{y,x}$ which is not a practical consideration though. Hence, we modify the existing algorithm in order to consider different setup times and for a case where product y follows x. For increasing the optimality we repeat the procedure by selecting a different product to begin a sequence. Although this will increase the computational time but it will not be a huge factor unless we talk about scheduling large product mix on the assembly level. Detailed procedure for this heuristic is given below.

Modified Closest Insertion Heuristic Procedure

- Let S_a be the set of unassigned products at any stage s.
- Let S_p be the partial sequence in existence at any stage and is denoted as, $S_p = \{i_1, i_2, i_3, \ldots, i_n\}$ implying that product i_1 is immediately followed by i_2. For each unassigned product i we use c(i) to denote the product among all assigned in the partial sequence that has the lowest setup time if chosen to precede i. Here, [i] refers the current position of product i in the partial sequence.

Procedure

Step-1 Start by scheduling product 1 to a sequence. Hence $s=1$, $S_p = \{1\}$, and $S_a = \{2,\ldots,n_1\}$, which is the set of all the products which are available for sequencing, for $i = 2,\ldots,n_1$. $c(i) = 1$.

Step-2 Select a new product i^* = argmin $\{ SET_{c(i),i} \; ; i \in S_a \}$ and set $s=s+1$.

Step-3 Insert i^* to the sequence and update $c(i)$. Delete the assigned product from S_a, $S_a = S_a - i^*$. Find product $t^* \in S_p$ with setup time such that

$t^* = \arg\min_{[t]\in S_p} \{SET_{[t]i^*} + SET_{i^*,[t+1]} - SET_{[t],[t+1]}\}$. Update

$S_p = \{i_1,\ldots,t^*,i^*,t^*+1,\ldots,i_n\}$. For all $i \in S_a$, if $\min\{SET_{i,i^*}, SET_{t^*,t}\} < SET_{i,c(i)}$ then $c(i) = i^*$. If $s < n_1$, go to step 3.

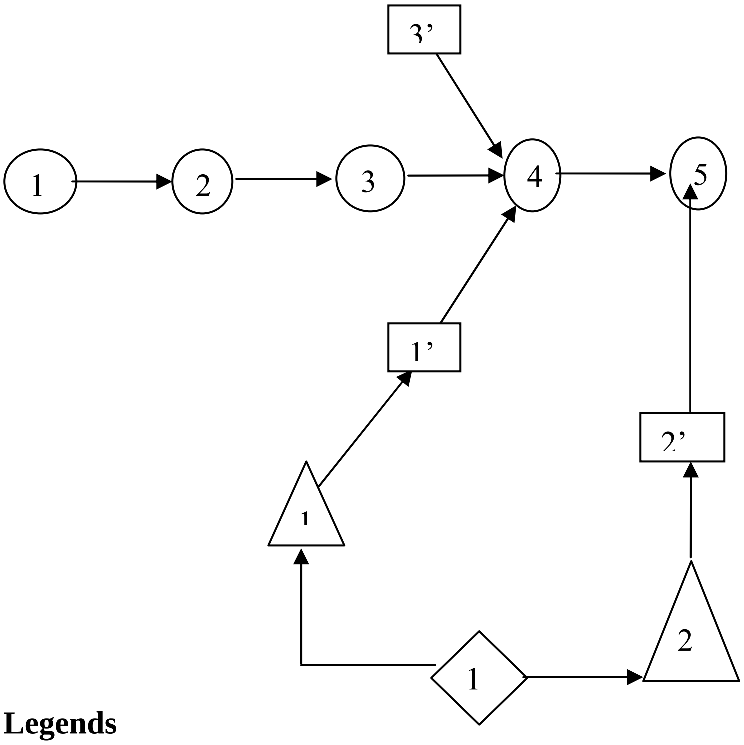
Step-4 Repeat all the steps by choosing a different product to begin with for the scheduling procedure and choose a sequence which will give the lowest setup time.

Our aim behind using separate scheduling procedures for assembly level and subsequent production levels is to get advantage of lower inventory while maintaining lower setup time. Although this heuristic will not give you the optimum solution it is good enough to produce the intended outcome. This heuristic is derived form the closest insertion heuristic.

CHAPTER 4. AN ILLUSTRATIVE EXAMPLE

4.1 Solution Procedure for Mixed-model Assembly Line Balancing

In this section, a simple example is used to illustrate our model. This example uses a 4-level production system: final assembly, subassembly, component, and raw material, to produce three different products: 1, 2, and 3. Figure 4 gives illustrations of typical production system. Here raw material is procured through outside supplier and stored in the inventory until needed by the component level production system similarly components are produced upon requirement from the subassembly level and products are sub-assembled depending on the demand from the final assembly level.



Legends

○- Level 1; □ - Level 2; △ - Level 3; ◇ - Level 4.

Figure 4. Assembly line with four production levels.

In the first half we concentrate on balancing the workload among and across the workstation for all the product types. As we mentioned before, the same precedence diagram is used for each product and a task that is not required by a product will takes a task time of zero. The precedence diagram is given in Figure 5.

Product-1

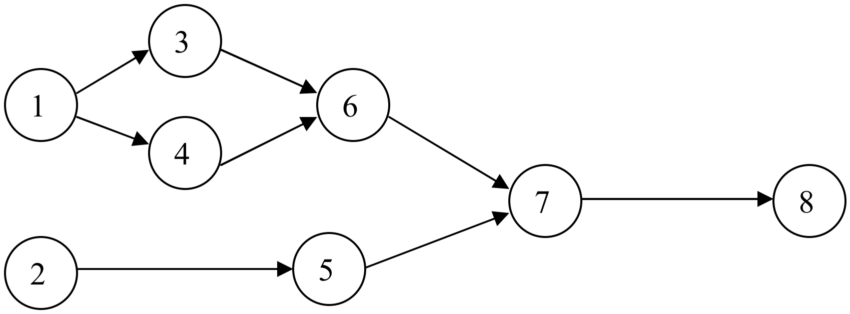


Figure 5. Precedence Diagram for Product 1.

Given the precedence diagram shown in Fig. 5 and the task times are: $p_{1,1}$-20, $p_{1,2}$-18, $p_{1,3}$-6, $p_{1,4}$-10, $p_{1,5}$-6, $p_{1,6}$-7, $p_{1,7}$-6, $p_{1,8}$-14, calculate ranked positional weights. The results are listed in Table 6.

Table 6. Positional weights and ranks of individual tasks product-1,

Task	**PW_i**	**Rank**
8	14+0=14	8
7	6+14=20	7
6	7+6+14=27	5
5	6+6+14=26	6
4	10+7+6+14=37	3
3	6+7+6+14=33	4
2	18+6+6+14=44	2
1	20+6+10+7+6+14=63	1

Arrange PW_i in non-increasing order to obtain the rank for each task, 1-2-4-3-6-5-7-8. The associated task times are 20 – 18 – 10 – 6 – 7 – 6 – 6 –14. According to Steps 3 and 4 of Stage 1 of the MRPW procedure, assign tasks to workstations. The results are given in Table 7.

Table 7. Summary of balancing for product 1.

Task	1-4	2-3-5	6-7-8	
Task time	(20,10)	(18,6,6)	(7,6,14)	
Work Stations	1	2	3	Total Variation
Variations	0	0	3	3

Product –2

Given the precedence diagram shown in Fig. 5 and task times: $p_{2,1}$-15,$p_{2,2}$-8, $p_{2,3}$-4, $p_{2,4}$-10,$p_{2,5}$-18, $p_{2,6}$-7, $p_{2,7}$-6, $p_{2,8}$-12, compute *PW* results are listed in Table 8.

Table 8. Positional weights and ranks of individual tasks product-2,

Task	PW_i	Rank
8	12+0=12	8
7	6+12=18	7
6	7+6+12=25	6
5	18+6+12=36	3
4	10+7+6+12=35	4
3	4+7+6+12=29	5
2	8+18+6+12=44	2
1	15+4+10+7+6+12=54	1

Arrange PW_i in decending order to obtain task ranks: 1 – 2 – 5 – 4 – 3 – 6 –7 –8 with associated task times: 15 – 8 – 18 – 10 – 4 – 6 – 6 –14. According to Steps 3 and 4 of Stage 1 of the MRPW procedure, assign tasks to workstations. The results are given in Table 9.

Table 9. Summary of balancing for product 2.

Task	1-2	5-4	3-6-7-8	
Task time	(15,8)	(18,10)	(4,7,6,12)	
Work Stations	1	2	3	Total Variation
Variations	7	2	1	10

Product –3

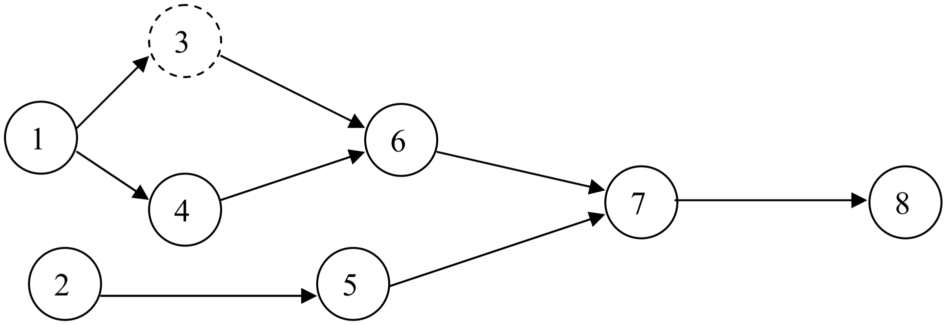


Figure 6. Precedence Diagram for Product 3.

Given the precedence diagram shown in Fig. 6 for product 3 task 3 is not present so we assign task time equal to zero and still maintain the original precedence diagram. Task times: $p_{3,1}$-8, $p_{3,2}$-12, $p_{3,3}$-0, $p_{3,4}$-7, $p_{3,5}$-8, $p_{4,6}$-4, $p_{4,7}$-12, $p_{4,8}$-14, compute PW.

Table 10. Positional weights and ranks of individual tasks product-3,

Task	PW_i	Rank
8	12+0=14	8
7	14+12=26	7
6	8+14+12=34	5
5	4+14+12=30	6
4	7+8+14+12=41	3
3	0+8+14+12=34	4
2	8+18+6+12=44	2
1	8+7+0+8+12+14=49	1

Arrange PW_i in descending order to rank the tasks: 1 – 2 – 4 – 3 – 6 – 5 –7 –8 with the associated task time: 8 – 12 – 0 – 7 – 8 – 4 – 12 –14. Following Steps 3 and 4 of Stage 1 of the MRPW procedure, assign tasks to workstations. The results are summarized in Table 11.

Table 11. Summary of balancing for product 3.

Task	1-2-3-4	5-6-7	8	
Task time	(8,12,0,7)	(8,4,12)	(14)	
Work Stations	1	2	3	Total variation
Variations	3	6	16	25

At this point, the first stage of MRPW is complete and we proceed to the 2nd stage. First, we compute the var_i values according to Step 1 of Stage 2 of the MRPW procedure. The results for total variations have been shown in Tables 7, 9, and 11, respectively.

Table 12. Summary of balancing for complete mixed model assembly.

Task	1-4	2-3-5	6-7-8	
Task time Product-1	(20,10)	(18,6,6)	(7,6,14)	
Task time Product-2	(15,10)	(8,4,18)	(7,6,12)	
Task time Product-3	(8,7)	(12,0,8)	(4,12,14)	
Work Stations	1	2	3	var_i
Variations Product-1	0	0	3	3
Variations Product-2	5	0	5	10
Variations Product-3	15	10	0	25

Choose the product with the lowest variation for further consideration. In this case, the selected product is 1 with variation of 3. Using the sequence of product 1, which has the following assignments (1, 4) (2, 3, 5) (6, 7, 8), we re-compute the variation for rest of the products and obtain new variations of 10 and 25 for products 2 and 3, respectively. In this case, the new variations are equivalent to the previous variations. Therefore, by selecting the sequence of product 1, one can obtain the best solution for all the models.

4.2 Product Schedule at All the Production Levels

Below is a bill of material for three different products with demand $d_{1,1}=10000$, $d_{1,2}=8000$, and $d_{1,3}=6000$ units with $n_1 = 3$, $n_2 = 3$, $n_3 = 4$ and $n_4 = 3$.

Table 13. Demand data for products at all production levels.

		Product		
		1	**2**	**3**
Sub-Assembly	**1**	1	0	0
	2	1	1	1
	3	0	0	4

Table 13, Continued

		Sub-Assembly		
		1	**2**	**3**
Component	**1**	1	0	1
	2	1	1	0
	3	0	1	0
	4	0	0	4

		Component			
		1	**2**	**3**	**4**
Raw Material	**1**	1	0	0	1
	2	0	1	0	1
	3	1	0	1	0

From the given data we will calculate the total demand at individual level and the ratio of individual demand to the total demand of all the products that are produced at that particular level. MATLAB program is used to compute Table 14, which gives us the bill of material considering all the production stages.

Table 14. Bill of material with ratio of demand of individual product to total demand.

Product	Sub-Assembly, *y=2*				Component, *y=3*				
y=1; i^1	**1**	**2**	**3**		**1**	**2**	**3**	**4**	
1	1	1	0		1	2	1	0	
2	0	1	0		0	1	1	0	
3	0	1	4		4	1	1	16	
				Total					**Total**
Demand $d_{i^y y}$ (in K)	10	24	24	**58**	34	34	24	96	**188**
Ratio $r_{i^y y}$	0.1724	0.4137	0.4137		0.1808	0.1808	0.1276	0.5106	

Raw Material, *y=4*			
1	**2**	**3**	
1	2	2	
0	1	1	
20	17	5	
			Total
130	130	58	**318**
0.4088	0.4088	0.1823	

Demand $d_{l,1}$	Ratios, $r_{l,1}$
10000	0.4166
8000	0.3333
6000	0.2500

We now have bill of material and input data required for the heuristic. We now solve this problem step by step to the scheduling heuristic presented in section 3.4.

Initiate stage 1, s=1 and calculate the variation between actual number of products produced and theoretical target for product 1 using, $V_{1,1,1} = w_1(x_{1,1,0} - sr_{1,1})$

$= 1\ (0 - 1 * 0.4166) = - 0.4166$

Similarly calculate the variations for the subsequent levels as

$$V_{1,2,1} = \{(x_{1,2,0} + t_{1,2,1}) - [XT_{2,0} + (t_{1,2,1} + t_{2,2,,1} + t_{3,2,1}]r_{1,2}\}^2$$

$$+ \{(x_{2,2,0} + t_{2,2,1}) - [XT_{2,0} + (t_{1,2,1} + t_{2,2,,1} + t_{3,2,1}]r_{2,2}\}^2$$

$$+ \{(x_{3,2,0} + t_{3,2,1}) + [XT_{2,0} + (t_{1,2,1} + t_{2,2,1} + t_{3,2,1})r_{3,2}]\}^2$$

$= \{(0+1) - [(0+2).1724]\}^2 + \{(0+1)-[(0+2)0.4134]\}^2 + \{(0+0)-[(0+2)0.4134]\}^2$

$= 1.144$

$$V_{1,3,1} = \{(x_{1,3,0} + t_{1,3,1}) - [XT_{3,0} + (t_{1,3,1} + t_{2,3,1} + t_{3,3,1} + t_{4,3,1})]r_{1,3}\}^2$$

$$+ \{(x_{2,3,0} + t_{2,3,1}) - [XT_{3,0} + (t_{1,3,1} + t_{2,3,1} + t_{3,3,1} + t_{4,3,1})]r_{2,3}\}^2$$

$$+ \{(x_{3,3,0} + t_{3,3,1}) - [XT_{3,0} + (t_{1,3,1} + t_{2,3,1} + t_{3,3,1} + t_{4,3,1})]r_{3,3}\}^2$$

$$+ \{(x_{4,3,0} + t_{4,3,1}) - [XT_{3,0} + (t_{1,3,1} + t_{2,3,1} + t_{3,3,1} + t_{4,3,1})]r_{4,3}\}^2$$

$=\{(0+1)-[0+4]\ 0.1808\}^2 + \{(0+2)-[0+4]\ 0.1808\}^2 + \{(0+1)-[0+4]\ 0.1276\}^2$

$+ \{(0+0)-[0+4]\ 0.5106\}^2 = 6.118$

$$V_{1,4,1} = \{(x_{1,4,0} + t_{1,4,1}) - [XT_{4,0} + (t_{1,4,1} + t_{2,4,1} + t_{3,4,1})]r_{1,4}\}^2$$

$$+ \{(x_{2,4,0} + t_{2,4,1}) - [XT_{4,0} + (t_{1,4,1} + t_{2,4,1} + t_{3,4,1})]r_{2,4}\}^2$$

$$+ \{(x_{3,4,0} + t_{3,4,1}) - [XT_{4,0} + (t_{1,4,1} + t_{2,4,1} + t_{3,4,1})]r_{3,4}\}^2$$

$= [(0+1)-(0+5)\ 0.4088]^2+[(0+2)-(0-5)\ 0.4808]^2+[(0+2)-(0-5)\ 0.1823]^2$

$= 2.276$

Then add all the above variations using the below equation

$$CV_{1,1,1} = w_1(x_{1,1,0} - sr_{1,1}) + \{w_2 \sum_{h=1}^{4}[(x_{1,2,0} + t_{1,2,1}) - (XT_{2,0} + \sum_{h=1}^{4} t_{1,2,1})r_{1,2}]^2\},$$

The results is the total variation summed over all production levels for product 1 at stage 1, $CV_{1,1,1}$ = -0.4166 + 1.144 + 6.118 + 2.276 = 9.1206.

Similarly for product 2 and 3,

$CV_{2,1,1}$ = 3.4252

$CV_{3,1,1}$ = 55.3395

At stage 1 we have total variations 9.1206, 3.4252, and 55.3395 for products 1, 2, and 3, respectively. We choose a product with the lowest variation, which is product 2 in this case. So we assign product 2 at stage one. According to Steps 4 & 5 of the scheduling heuristic, we decide a product to follow product 2 which will minimize the total variation at the next stage. Based on our calculation, we again assign product 2 as shown in Table 15.

We update the value of $x_{i^1,y,s-1}$ with every assignment of a product at any stage. We continue assigning the products till $s = S$. Further assignments are also shown in Table 15.

Table 15. Schedule with equal weights for all the production stages.

Stage #	Product	Variation					Product Scheduled
		Level-1	Level-2	Level-3	Level-4	Total	
1	1	-0.417	1.144	6.118	2.276	9.12069	
	2	**-0.333**	**0.545**	**2.136**	**1.105**	**3.45232**	**2**
	3	-0.250	5.615	34.858	15.116	55.3395	

Table 15. Continued

2	1	-0.833	2.350	14.584	6.454	22.5547	
	2	**0.333**	**2.178**	**8.543**	**4.421**	**15.476**	**2**
	3	-0.500	3.605	20.748	8.152	32.005	
3	1	-1.250	4.644	27.322	12.844	43.5601	
	2	1.000	4.901	19.222	9.947	35.0709	
	3	**-0.750**	**2.685**	**10.909**	**3.398**	**16.2418**	**3**
4	**1**	**-1.667**	**0.457**	**0.855**	**0.137**	**-0.21868**	**1**
	2	0.667	2.854	5.342	0.854	9.71659	
	3	0.000	14.421	82.991	32.608	130.02	
5	**1**	**-1.083**	**0.516**	**3.036**	**1.427**	**3.89556**	**1**
	2	0.333	1.287	1.618	0.667	3.90449	
	3	-0.250	7.498	44.104	17.663	69.0148	
6	1	-0.500	2.863	17.452	7.269	27.0845	
	2	**0.000**	**2.007**	**10.130**	**5.031**	**17.1671**	**2**
	3	-0.500	2.863	17.452	7.269	27.0845	
7	1	-0.917	5.015	30.876	13.946	48.9211	
	2	0.667	4.587	21.495	10.844	37.5933	
	3	**-0.750**	**1.800**	**8.300**	**2.803**	**12.1526**	**3**
8	**1**	**-1.333**	**0.114**	**0.214**	**0.034**	**-0.97134**	**1**
	2	0.333	1.826	3.419	0.547	6.12528	
	3	0.000	12.823	76.187	30.808	119.817	
9	1	-0.750	0.716	4.363	1.817	6.14613	
	2	**0.000**	**0.801**	**1.663**	**0.852**	**3.31641**	**2**
	3	-0.250	6.442	39.268	16.355	61.8151	
10	1	-1.167	2.064	12.143	5.708	18.7489	
	2	**0.667**	**2.578**	**7.384**	**3.880**	**14.5088**	**2**
	3	-0.500	4.576	24.471	9.103	37.6494	
11	1	-1.583	4.502	24.195	11.810	38.923	
	2	1.333	5.444	17.377	9.119	33.2725	
	3	**-0.750**	**3.798**	**13.946**	**4.061**	**21.055**	**3**
12	**1**	**-2.000**	**1.027**	**1.923**	**0.308**	**1.25797**	**1**
	2	1.000	4.109	7.692	1.230	14.0319	
	3	0.000	16.247	90.224	34.476	140.947	
13	**1**	**-1.417**	**0.545**	**2.136**	**1.105**	**2.36899**	**1**
	2	0.667	2.000	2.000	0.550	5.21657	
	3	-0.250	8.782	49.368	19.038	76.9385	
14	1	-0.833	2.350	14.584	6.455	22.5547	
	2	**0.333**	**2.178**	**8.543**	**4.421**	**15.476**	**2**
	3	-0.500	3.605	20.748	8.152	32.005	
15	1	-1.250	4.644	27.322	12.844	43.5601	
	2	1.000	4.901	19.222	9.947	35.0709	
	3	**-0.750**	**2.685**	**10.909**	**3.398**	**16.2418**	**3**
16	**1**	**-1.667**	**0.457**	**0.855**	**0.137**	**-0.21868**	**1**
	2	0.667	2.854	5.342	0.854	9.71658	
	3	0.000	14.421	82.991	32.608	130.02	
17	**1**	**-1.083**	**0.516**	**3.036**	**1.427**	**3.89557**	**1**
	2	0.333	1.287	1.618	0.667	3.90449	
	3	-0.250	7.498	44.104	17.663	69.0148	

Table 15. Continued

18	1	-0.500	2.863	17.452	7.269	27.0845	
	2	**0.000**	**2.007**	**10.130**	**5.031**	**17.1671**	**2**
	3	-0.500	2.863	17.452	7.269	27.0845	
19	1	-0.917	5.015	30.876	13.946	48.9211	
	2	0.667	4.587	21.495	10.844	37.5933	
	3	**-0.750**	**1.800**	**8.300**	**2.803**	**12.1526**	**3**
20	**1**	**-1.333**	**0.114**	**0.214**	**0.034**	**-0.97134**	**1**
	2	0.333	1.826	3.419	0.547	6.12528	
	3	0.000	12.823	76.187	30.808	119.817	
21	1	-0.750	0.716	4.363	1.817	6.14614	
	2	**0.000**	**0.801**	**1.663**	**0.852**	**3.31641**	**2**
	3	-0.250	6.442	39.268	16.355	61.8151	
22	1	-1.167	2.064	12.143	5.708	18.7489	
	2	**0.667**	**2.578**	**7.384**	**3.880**	**14.5088**	**2**
	3	-0.500	4.576	24.471	9.103	37.6494	
23	1	-1.583	4.502	24.195	11.810	38.923	
	2	1.333	5.444	17.377	9.119	33.2725	
	3	**-0.750**	**3.798**	**13.946**	**4.061**	**21.055**	**3**
24	**1**	**-2.000**	**1.027**	**1.923**	**0.308**	**1.25796**	**1**
	2	1.000	4.109	7.692	1.230	14.0319	
	3	0.000	16.247	90.224	34.476	140.947	

We assign equal weights, $w_y = 1$, for all the production stages to obtain the above sequence S <2-2-3-1-1-2-3-1-2-2-3-1-1-2-3-1-1-2-3-1-2-2-3-1>, which will be repeated *GC* = 10,000 times to satisfy cumulative yearly demand of $D_1 = 240{,}000$ for all the products with individual demands of $d_{1}^{1}{}_{,1} = 100{,}000$, $d_{1}^{2}{}_{,1} = 80000$ *and* $d_{1}^{3}{}_{,1} = 60000$ for products 1, 2, and 3, respectively. Table 16 further shows different schedules developed using the same scheduling heuristic but different values of w_y. In the second case in Table 16 variation at the final assembly is not considered to be important hence weight $w_1=0$. The reason behind this is in many cases production and inventory cost is higher at the subassembly and subsequent production level as compare to final assembly. In that case, as per the JIT system, focus will be on minimizing the variation at the subassembly and other subsequent levels. Similarly we can focus on the other levels by changing the weights assigned to them.

Table 16. Schedule based on the weights at the different production levels.

Sequ e-nce	w_y	**Desired sequence of the product mix**																							
1	1,1,1,1	2	2	3	1	1	2	3	1	2	2	3	1	1	2	3	1	1	2	3	1	2	2	3	1
2	0,1,1,1	2	2	3	1	2	2	3	1	1	2	3	1	1	2	3	1	2	2	3	1	1	2	3	1
3	0,0,1,1	2	2	3	1	2	2	3	1	1	2	3	1	2	2	2	3	1	2	2	3	1	1	2	3
4	0,0,0,1	2	2	3	1	2	2	3	1	2	2	3	1	1	2	3	1	2	2	2	3	1	2	2	3
5	1,1,1,0	2	2	3	1	1	2	3	1	2	2	3	1	1	2	3	1	1	2	3	1	2	2	3	1
6	1,1,0,0	2	1	3	1	2	3	1	2	1	3	1	2	2	1	3	1	2	3	1	2	1	3	1	2
7	1,0,0,0	1	2	3	1	2	1	3	2	1	3	2	1	1	2	3	1	2	1	3	2	1	3	2	1

From Table 16 we can see that there are 15 product changes at all the production levels, which result in high setup cost and time and hence high cost. As we mentioned before, to strike a balance between lower inventory and lower setup time we implement the closest insertion heuristic to minimize sequence-dependent setup time at the final assembly level. This example is continuously used to illustrate our setup heuristic procedure given in Section 3.4.2.

4.3 Closest Insertion Heuristic to Minimize the Setup Cost

As mentioned earlier we have sequence dependent setup times at the final assembly level. Table 17 shows setup times associated with the product sequence, from which we can see that if product 1 is followed by product 2 on the final assembly level, setup time of 3 seconds is required similarly setup time of 6 seconds is incurred when product 3 is followed by product 2. Detailed solution steps are presented below.

Table 17. Sequence dependent setup time.

Product

Produ ct		1	2	3
	1	-	3	5

Table 17. Continued

2	4	-	7
3	3	6	-

As we mentioned in the closest insertion heuristics presented in the last chapter, we start our scheduling procedure by scheduling the product with the highest demand as a first product in the sequence.

Step 1. Start with product 1. $s = 1$. $S_p = \{1\}$. $S_a = \{2, 3\}$.

$c(2) = c(3) = 1$, any of the products can be assigned after product 1.

Step2. Select new product,

For product 2, $c(2) = 1$; the associated time is $SET_{1,2} = 3$.

For product 3, $c(3) = 1$; the associated time is $SET_{1,3} = 5$.

Product 2 has minimum associated time and we set $i^* = 2$. Set $s = 2$.

Step3. Insert product 2, and update $c(3)$.

$S_a = \{3\}$. We place product 2 after product 1. The time increment after placing product 2 after product 1 will be, $SET_{12} + SET_{21} - SET_{11} = 3 + 0 - 0 = 3$.

Step 2. Select new product.

Only product 3 remains. $i^* = 3$.

Step 3. Insert product 3.

Updating $S_a = \{\}$. Insertion choices are

Product 3 after product 1: $SET_{13} + SET_{32} - SET_{12} = 5 + 6 - 3 = 8$

Product 3 after product 2: $SET_{23} + SET_{31} - SET_{21} = 7 + 3 - 4 = 6$

Insert product 3 after 2. The final sequence is {1, 2, 3} with setup time during every setup change is= $SET_{12} + SET_{23} = 3 + 7 = 10$

Step 2. S_a is empty. Assignment is complete and stop.

By repeating the same procedure where we schedule product 2 to begin the sequence, we get sequence S2 <2-1-3> with setup time = 4 +5 = 9; similarly if we begin the sequence with 3, S3 <3-1-2> with time = 3 + 3 = 6. Therefore, we choose the sequence S3 as our final schedule.

We now compare the results obtained through the scheduling heuristic in the previous stage and the output of the closest insertion heuristic. Table 18 shows the effect of different schedules on the total setup time and lost in production due to setup time at the final assembly level.

Table 18. Effect of sequence on total production output.

Criteria	**Sequence**																								**Setup Time/ Cycle**	**Total Product -ion lost**	**% loss**
W [1,1,1,1]	2	2	3	1	1	2	3	1	2	2	3	1	1	2	3	1	1	2	3	1	2	2	3	1	75	25000	10.5
Setup Time		0	7	3	0	3	7	3	3	0	7	3	0	3	7	3	0	3	7	3	3	0	7	3			
W [0,1,1,1]	2	2	3	1	2	2	3	1	1	2	3	1	1	2	3	1	2	2	3	1	1	2	3	1	75	25000	10.5
Setup Time		0	7	3	3	0	7	3	0	3	7	3	0	3	7	3	3	0	7	3	0	3	7	3			
CIM	3	3	3	3	3	3	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	6	2000	0.83
Setup Time		0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0			

From the Table 18 we can see that with the balanced production schedule there is 10.5 % lost in the total production due to frequent setups and higher setup time. However with the closest insertion heuristic with the same setup times we are able to reduce the lost in the production to mere 0.83%. With our approach we can still maintain the advantage of lower inventory level by following balanced schedule at lower production levels and lower setup time with a partial batched production at the final assembly level.

CHAPTER 5. VERIFICATION AND SENSITIVITY ANALYSIS

5.1 Verification

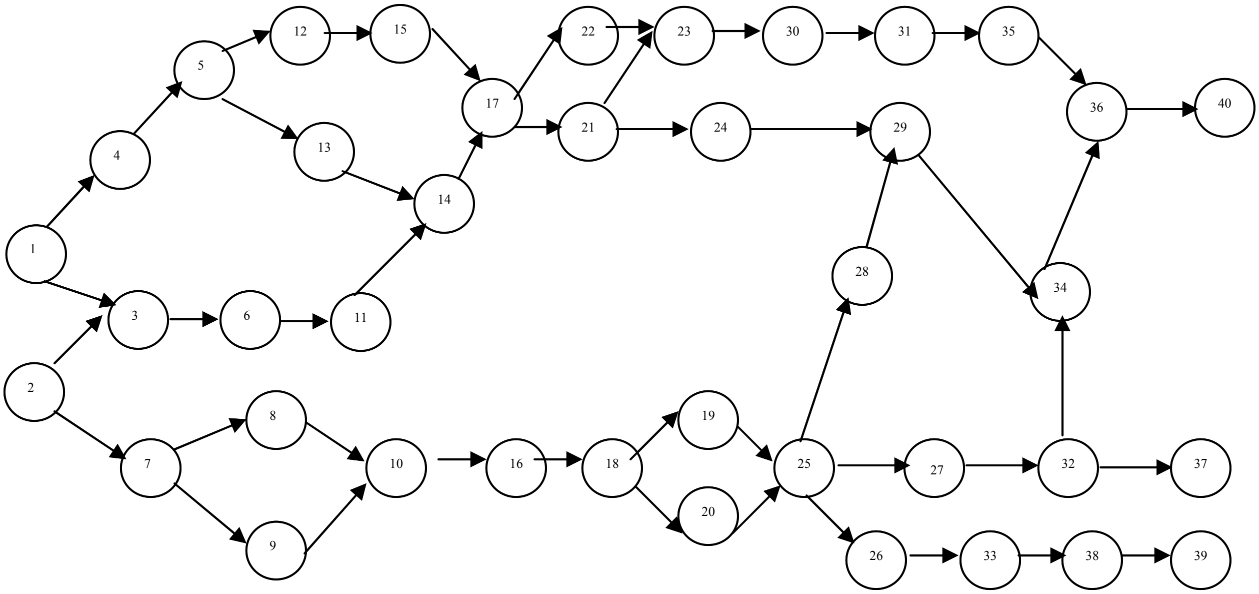


Figure 7. Precedence diagram for sensitivity analysis.

Table 19. Task time data of all the products for sensitivity analysis.

Task #	Product1	Product2	Product3	Task #	Product1	Product2	Product3
1	10	12	13	**21**	8	6	6
2	8	9	10	**22**	11	10	11
3	6	6	6	**23**	6	4	4
4	12	15	17	**24**	16	12	13
5	10	8	9	**25**	12	11	12
6	15	12	13	**26**	6	4	4
7	10	8	9	**27**	8	6	6
8	6	5	5	**28**	11	9	10
9	8	10	11	**29**	12	9	10
10	14	11	12	**30**	7	5	5
11	11	8	9	**31**	5	6	6
12	13	12	13	**32**	11	8	9
13	18	15	17	**33**	13	10	11
14	6	4	4	**34**	11	12	13
15	15	12	13	**35**	9	7	8
16	14	16	18	**36**	6	9	10
17	13	10	11	**37**	17	13	14
18	9	7	8	**38**	12	9	10
19	7	9	10	**39**	14	14	16
20	12	9	10	**40**	6	4	4

Experimental test for balancing the workload among and across the workstations was carried out. Effect of cycle time on number of workstations and total variation were observed. Precedence diagram for the test in shown in Figure 7, an assembly line with 3 different products and each requires 40 task to be completed for complete production is considered. We have assumed combined precedence diagram with different task times as shown in Table 19. For the data given above, output of our heuristic is shown in Table 20. We now compare the output with the theoretical optimum solution and the results of previous research in this area. Mixed-model assembly line balancing result is tested using cycle time = 30, flexibility factor as 1.05. Further discussion about flexibility factor is continued in the section 5.1.2.

Table 20. Mixed-model assembly line balancing results for sensitivity analysis.

Maximum time limit with flexibility factor is = 94.4
Cycle time constraint is = 90
Product which carry minimum variation = 3
Total minimum variation for product 3 is = 50

Station Number	**Time Allotted per Station**	**Variation per Station**
1	83	7
2	88	2
3	72	18
4	71	19
5	75	15
6	85	5
7	78	12
8	82	8
9	86	4
10	92	-2
11	89	1
12	89	1
13	67	23
14	83	7
15	44	46
	Total Variation	**166**

5.2 Sensitivity Analysis

5.2.1 Comparison- Actual and Theoretical Minimum Number of Workstations

The numbers of stations resulting from the suggested methodologies are compared to the minimum theoretical number of stations (*TK*). The expression of *TK* is a special case of task allocation with zero variation at all workstations, mathematical expression can be written as,

$$TK = \left\lceil \frac{\sum_{i=1}^{n_i} \sum_{\sigma=1}^{n_\sigma} p_{\sigma,i}}{n_i C} \right\rceil$$

We check the efficiency of balancing results in terms of percentage increase in number of work stations in a way the effectiveness of our heuristic. The expression used to calculate percentage increase is {[(Number of Stations - *TK*)/ *TK*]*100}; in our test we get 20, 25, 7.14, 8.33 and 0 percent increases in the number of actual workstations required as compare to theoretical minimum number of workstations with the cycle time of 20, 25, 30, 35 and 40 respectively. The results are shown in Table 21.

Table 21. Change in variation and number of workstations with change in cycle time.

Cycle Time	# of Stations	Theoretical Min # of Stations	% Variation	% Increase in # of workstations
20	24	20	18.19	20.00
25	20	16	21.07	25.00
30	15	14	12.44	7.14
35	13	12	13.63	8.33
40	10	10	4.58	0.00
		Average	13.98	12.10

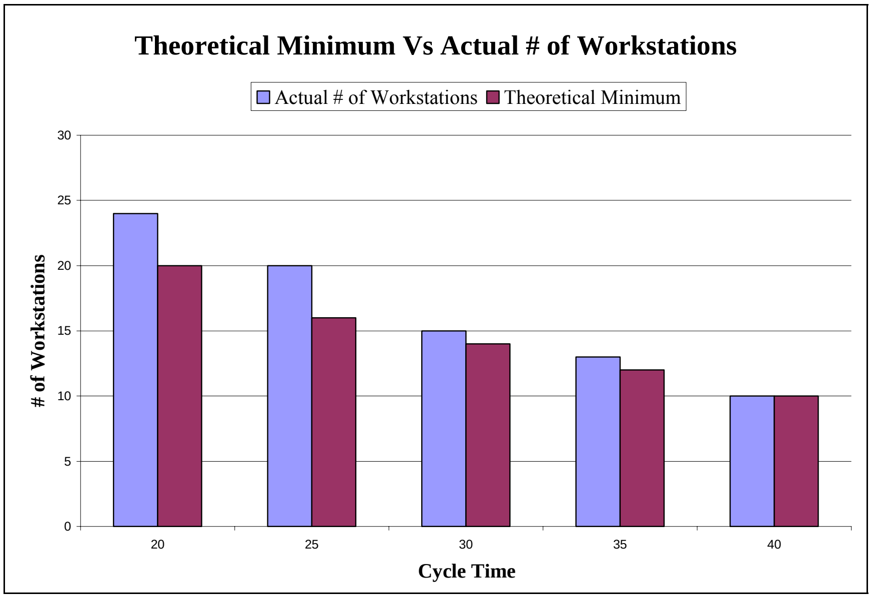


Figure 8. Comparison between actual and theoretical minimum on number of workstations required

The above figure shows that the best balance is with cycle time of 40, in which there is negligible variation. However, higher demand and increase in overhead cost might not permit us to set such a high cycle time of 40 for longer period of time in order to be profitable, as shown in Fig. 14 in the next section.

5.2.2 Effect of Flexibility Factor on Total Cost Function

In a steady-paced moving conveyor assembly line cycle time allotted to the work station is controlled by the length allotted to the work station. It is not always possible for a worker to complete his duties within the allotted time so we have to provide some flexibility in completing their tasks. Table 22 shows the effect of flexibility factor on total cost, which is a function of operating cost, cost associated with completing unfinished task on the final assembly level, lost production due to variation at each workstation, and overheads. Table 22 shows that although increase in flexibility factor reduces the total output, it is sometimes beneficial for the company as it reduces the number of incomplete tasks, thus eventually minimizes the cost associated with it. For example if we consider a case with cycle time of 20 and flexibility factor of 1.00 and compare it with flexibility factor of 1.05 we get profits of $2.41 and $2.83, respectively.

Lost production = (360000 – 342857) = 17,142

Total profit lost = 17142 * 2.41 = 41314.63

Gain in profit for flexibility factor 1.05 = (342857*2.83) - (360000*2.41)

= 1, 02,685

Net gain in profit after increasing the flexibility factor to 1.05 is = 102685.00 – 41314.63 = $61370.37 per year. This example clearly illustrates that increasing flexibility of mere 5% can give considerable rise in profit margin of the company with final assembly line in consideration.

Table 22. Effect of flexibility factor and cycle time on total cost function.

Flexibility Factor 1.00

Cycle Time	# of Stati-ons	Theoretical Min # of Stations	% of Incomp-lete tasks	Total Variation	Total Produc-tion	Operating Cost/ unit	Compl-etion cost	Lost Production	Overhe-ads	Total Cost	Profit / unit
20	24	20	12.5	262	360000	4.00	0.83	0.09	0.67	5.59	2.41
25	20	16	2.5	316	288000	4.17	0.17	0.13	0.83	5.31	2.69
30	15	14	2.5	168	240000	3.75	0.16	0.09	1.00	5.00	3.00
35	13	12	0	186	205714	3.79	0.00	0.12	1.17	5.08	2.92
40	10	10	7.5	55	180000	3.33	0.42	0.05	1.33	5.13	2.87

Flexibility Factor 1.05

Cycle Time	# of Statio-ns	Theoretical Min # of Stations	% of Incomplete tasks	Total Variation	Total Production	Operating Cost/ unit	Compl-etion cost	Lost Production	Overheads	Total Cost	Profit / unit
20	24	20	2.5	257	342857	4.20	0.18	0.09	0.70	5.17	2.83
25	20	16	0	316	274285	4.38	0.00	0.14	0.88	5.39	2.61
30	15	14	0	166	228571	3.94	0.00	0.10	1.05	5.08	2.92
35	13	12	0	181	195918	3.98	0.00	0.12	1.23	5.33	2.67
40	10	10	7.5	46	171428	3.50	0.44	0.04	1.40	5.38	2.62

Table 22. Continued

Flexibility Factor 1.1

Cycle Time	# of Stations	Theoretical Min # of Stations	% of Incomplete tasks	Total Variation	Total Production	Operating Cost/ unit	Completion cost	Lost Production	Overheads	Total Cost	Profit / unit
20	24	20	2.5	257	342857	4.20	0.18	0.09	0.70	5.17	2.83
25	20	16	0	316	274285	4.38	0.00	0.14	0.88	5.39	2.61
30	15	14	0	166	228571	3.94	0.00	0.10	1.05	5.08	2.92
35	13	12	0	181	195918	3.98	0.00	0.12	1.23	5.33	2.67
40	10	10	7.5	46	171428	3.50	0.44	0.04	1.40	5.38	2.62

Flexibility Factor 1.15

Cycle Time	# of Stations	Theoretical Min # of Stations	% of Incomplete Jobs	Total Variation	Total Production	Operating Cost/ unit	Completion cost	Lost Production	Overheads	Total Cost	Profit / unit
20	24	20	0	256	313043	4.60	0.00	0.10	0.77	5.47	2.53
25	20	16	0	316	250434	4.79	0.00	0.15	0.96	5.90	2.10
30	15	14	0	166	208695	4.31	0.00	0.11	1.15	5.57	2.43
35	13	12	0	196	178881	4.36	0.00	0.14	1.34	5.85	2.15
40	10	10	0	33	156521	3.83	0.00	0.03	1.53	5.40	2.60

For better understanding we plot the results shown in Table 22 by graphically representing the effect on output with change in some input parameters.

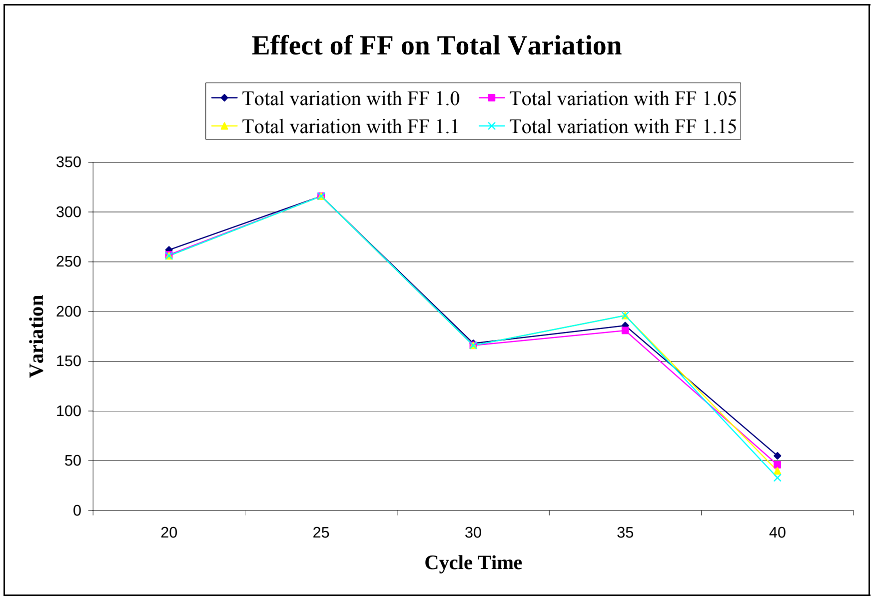


Figure. 9 Relationship between cycle time, flexibility factor and sum of variations.

From Figure 9 we can see that there is no considerable difference in the variation with change in flexibility factor. That shows small increase in flexibility factor does not reduce variation at the workstation. However, it helps in reducing number of incomplete jobs.

Advantage of having flexibility factor can be seen in Figure 10, only 5% increase in flexibility will reduce the percent of incomplete tasks from 12.5% to 2.5% thus reducing considerable resources needed to complete those incomplete tasks towards the end of the assembly line.

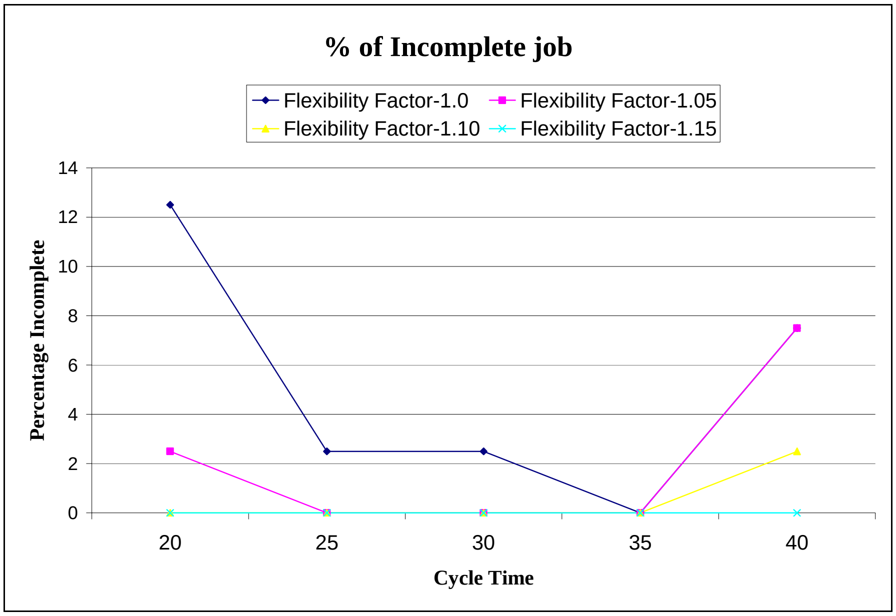


Figure 10. Effect of flexibility factor and cycle time on number of incomplete tasks.

It is obvious that increase in flexibility factor will reduce the total production output. Figure 11 shows how flexibility factor affect the final production output. Although there is a decrease in the output, in some cases it is advantageous in generating higher profit as can be seen in Figure 12, where an increase in flexibility factor from 1 to 1.05 results in an increase in the profit from 2.41 to 2.83 when the cycle is set at 20.

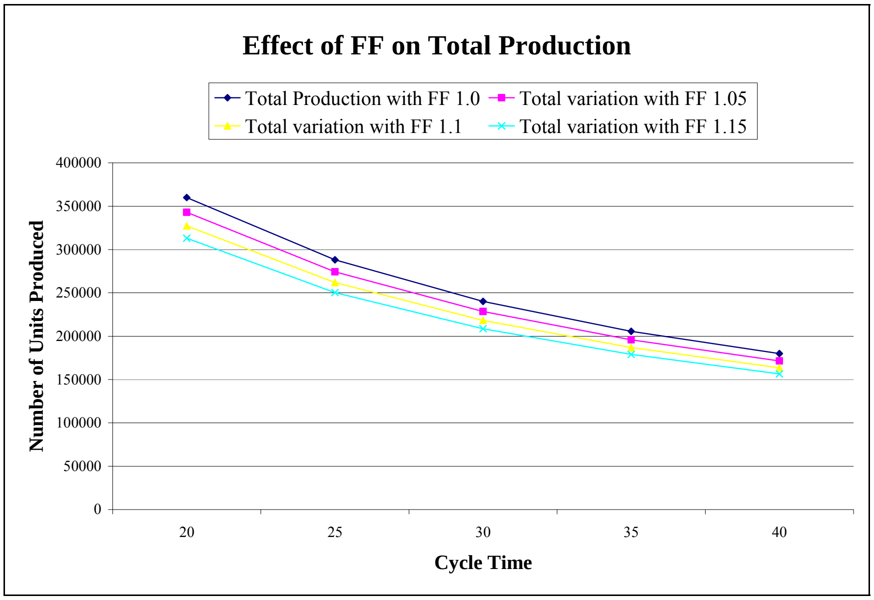


Figure 11. Change in total production with change in flexibility factor.

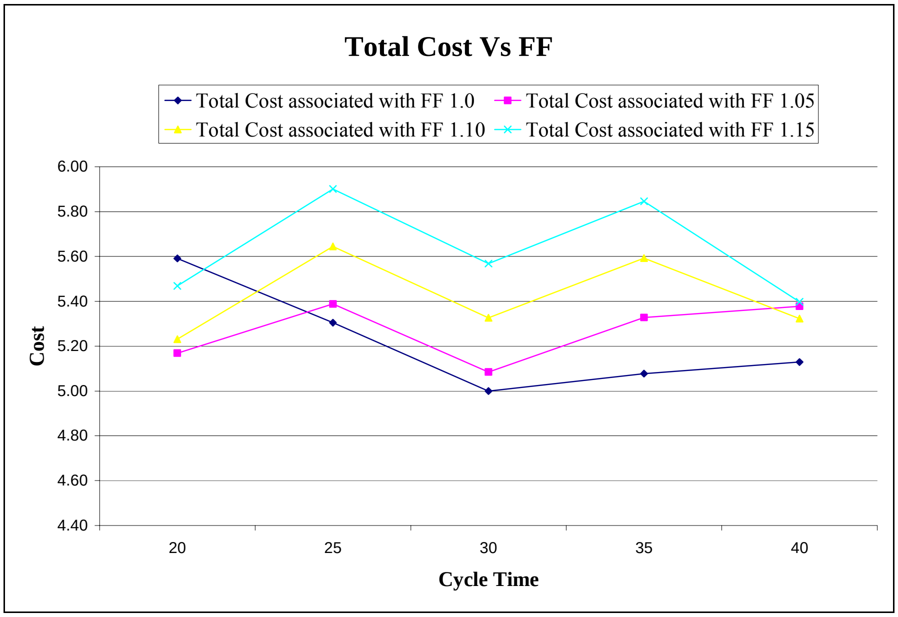


Figure 12. Change in total cost function with change in flexibility factor.

With increase overheads total cost function will be increased. Comparing results shown in Figure 13 with the one shown in Figure 12 we can see that Figure 13 shows a gradual increase in the total cost with an increase in flexibility factor and cycle time.

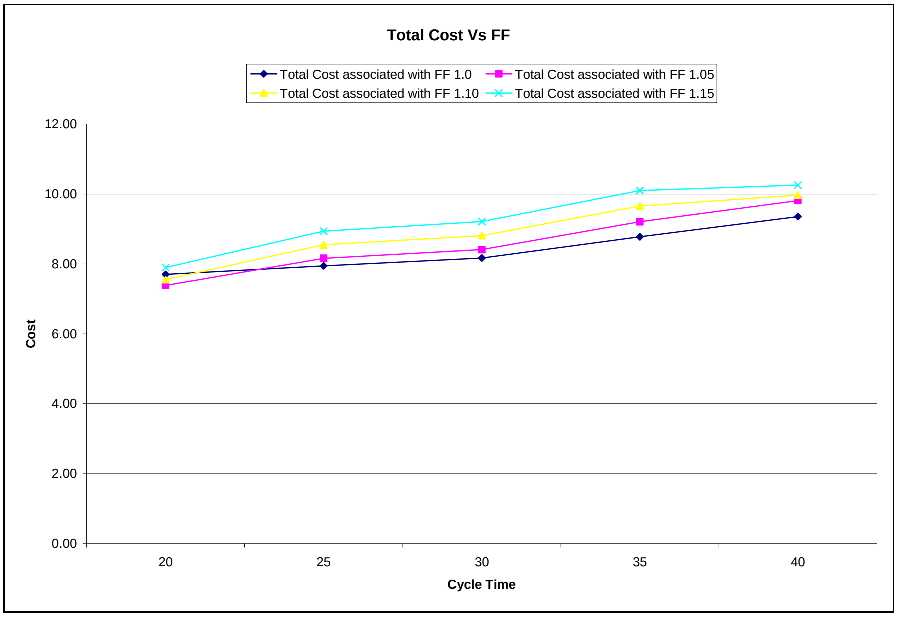


Figure 13. Change in cost function with increased overheads.

If we consider a case with higher overhead cost, profit will go down drastically with increase in cycle time. Hence, in a manufacturing organization where there is a higher cost associated with activities other than actual production we will have to ensure proper production rate to accommodate that extra cost. Also we can see from Figure 14 that higher flexibility factor is not recommended in such scenario. Company will have to carefully choose flexibility factor such that it will minimize the cost associated with the incomplete job and will not incur extra overhead cost to the company. From the analysis we can see that cycle time (*C*) = 20 with flexibility factor (FF) = 1.05 will give you the highest profit for the case under consideration.

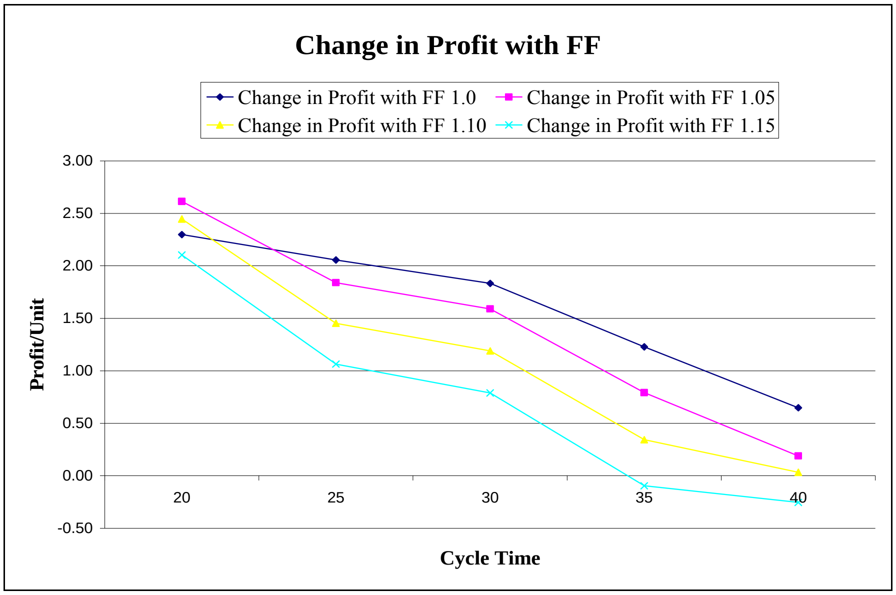


Figure14. Effect of change in cycle time and FF on the profit/unit.

CHAPTER 6. CONCLUSION

The basis for our model is the just-in-time manufacturing scenario where all the production levels are interdependent. Demand at the assembly level affects the production schedule of the sub-assembly, component and the raw material. This problem has been addressed in the past without much success as attention was given either only on balancing workload or sequencing mixed-model at assembly level. In this paper we overcome this drawback by balancing workload at every workstation and across the workstations at the final assembly level and effective sequence to be followed at all production level which will ensure lower inventory levels at all production levels.

Balancing mixed-model is a difficult task and sequencing of task even makes it more difficult for computation. Exact solution is computationally time intensive however our proposed heuristic will give you near optimal solution with significantly less amount of computation.

Even though this is good attempt obtain a balance between assembly line balancing and effective sequence for all levels there are few limitations for this model.

- *Myopic* as it does not consider the effect of its current decision on the variation in future cycles. That is, it may achieve low variability at stage s at the expense of higher variability at stage $s+1$.
- *Practical issues* such as reducing unnecessary setups and traveling distance should be considered.
- *Computational complexity,* even though proposed heuristic is computationally effective as compared to the previous algorithms, in practical there are hundreds of product produced on the assembly line with higher number of

sub-assemblies, component and raw material, data becomes vast as we move down in the production level. Our heuristic has limited advantage in such scenario as we consider impact of every product on sequencing the product at the final as well as subsequent levels.

Further, there is a scope to extend our heuristic by adding constraints on length of assembly line, workstation length, and workload should not exceed equipment capacity. Our model is of limited use when processing time is imprecise, vague or uncertain the line balancing problem in such scenario can be solved using fuzzy logic. Where, task for each model can be estimated in the form of fuzzy numbers. However considering effect of setup times on the product sequence at the final assembly is critical issue as we have considered sequence dependent setup time in our model.

REFERENCES

Askin R. G., Standridge C. R. Charity Robey (eds.) (1993), *Modeling and Analysis of Manufacturing Systems,* John Wiley & Sons, New York.

Bard Jonathan F. (1989), “Assembly line balancing with parallel work stations and dead time,” *International Journal of Production Research,* **27**(4), 1005-1018.

Berger, I., Bourjolly, J., and Laporte, G. (1992), “Branch-and-bound algorithms for the muli-product assembly line balancing problem,” *European Journal of Operational Research,* **38**, 215-222.

Bukchin, J. Dar-El, E.M. and Rubinovitz. J. (2002), “Mixed-model assembly line design in make-to-order environment,” *Computers and Industrial Engineering,* **41**, 405-421.

Candance, A.Y. and Rachmadugu, R., (1991), “Sequencing to minimize work overload in assembly lines with product options,” *Management Science,* **37**(5), 574-586.

DePuy, G.W. and Whitehouse, G.E. (2000), “Applying COMSOAL computer heuristic to the constrained resource allocation problem,” *Computers and Industrial Engineering,* **38**(2), 413-422.

Ding Fong-Yuen, and Tolani Ravi (2003), “Production planning to support mixed-model assembly,” *Computers and Industrial Engineering,* **45**, 375-392.

Erdal Erel, and Hadi, Gokcen. (1998), “Binary integer formulation for mixed-model assembly line balancing problem,” *Computers and Industrial Engineering*, **34**(2), 451-461.

Erdal Erel, and Hadi, Gokcen. (1999), “Shortest-route formulation of mixed-model assembly line balancing problem,” *European Journal of Operational Research*, **116**, 194-204.

Gendreau, M., Laporte G., and Guimaraes E. M. (2001) “A divide and merge heuristic for the multiprocessor scheduling problem with sequence dependent setup times,” *European Journal of Operations Research*, **133**, 183-189.

Hadi Gokcen, and Erdal Erel (1997), “Goal programming approach to mixed model assembly line balancing problem,” *International Journal of Production Economics*, **48**(2), 177-185.

Jin, M. and Wu. David. (2002), “A new heuristic method for mixed-model assembly line balancing problem,” *Computers and Industrial Engineering*, **44**, 159-169.

Josef Voros, (1999), “Lot sizing with quality improvement and setup time reduction,” *European Journal of Operational Research,* **113**, 194-204.

Lee, S. M., Everett, A. M. and Melkonian, J. H., (1994), “Optimizing lot size and setup time in a JIT environment: a multiple objective application,” *Production Planning and Control*, **5**(3), 308-319.

Lee, S. M., and Jung Hun-Joo., (1989), “Multi-objective production planning model in flexible manufacturing environment,” *International Journal of Production Reseach*, **27**(11), 1981-1992.

Lee, T.O., Kim, Yeongho, and Kim. Y. K. (2001), “Two-sided assembly line balancing to minimize work relatedness and slackness,” *Computers and Industrial Engineering*, **40**, 273-292.

Macaskill, J.L.C., (1972), “Production line balancing for mixed lines,” *Management Science*, **19**(4), 423-433.

Matanachai, S., and Candance, A.R., (2001), “Balancing mixed-model assembly lines to reduce work overload,” *IIE Transactions*, **33**, 29-42.

Merengo, C., Nava, F., and Pozzetti, A., (1999), “Balancing and sequencing manual mixed-model assembly line,” *International Journal of Production Research*, **33**(3), 2835-2860.

Miltenburg, J., (1989), “Level schedules for mixed-model assembly lines in just-in-time production systems,” *Management Science*, **35**(2), 193-207.

Miltenburg, J. and Sinnamon, G., (1989), “Scheduling mixed-model multi-level just-in-time production systems,” *International Journal of Production Research,* **27**(9), 1487-1509.

Mingzhou, J., and Wu, S. David., (2002), “A new heuristic method for mixed model assembly line balancing problem,” *Computers and Industrial Engineering*, **44**, 159-169.

Nevis, A.J., (1972), “Assembly line balancing using best bud search,” *Management Science*, **18**(9), 529-539.

Nguyen, V. H., (2006), “A heuristic solution for fuzzy mixed-model line balancing problem,” *European Journal of Operational Research,* **168**, 798-810.

Peng, S. O., and Morton T.H., (1989), “The single machine early/tardy problem,” *Management Science,* **35**(2), 177-191.

Rajendran, C. and Zeigler, H. (1997), “A heuristic for scheduling to minimize the sum of weighted flowtime of jobs in a flowshop with a sequence dependent setup times of jobs,” *Computers and Industrial Engineering,* 33(1), 281-284.

Raouf, A., Tusi, C.L. and El-Sayed, E.A. (1980), “A new heuristic approach to assembly line balancing,” *Computers and Industrial Engineering*,**4**, 223-234.

Roger Johnson, (1988), “Optimally balancing large assembly lines with ‘FABLE’,” *Management Science*, **34**(2), 241-253.

Sarker, B.R., and Shanthikumar J.G. (1983), “A Generalized approach for serial or parallel line balancing,” *International Journal of Production Research*, **21**(1), 109-133.

Sarker, B.R., and Haixu Pan. (2001), “Design configuration for a closed-station, mixed-model assembly line: a filing cabinet manufacturing system,” *International Journal of Production Research*, **39**(10), 2251-2270.

Talbot B.F., and Patterson J.H., (1984), “An integer programming algorithm with network cuts for solving the assembly line balancing problem,” *Management Science*, **30**(1), 85-99.

Trevino, B.J., (1993), “A mathematical model for the economic justification of setup time reduction,” *International Journal of Production Research*, **31**(1), 191-202.

Thomopoulos N. T., (1970), “Mixed-model line balancing with smoothed station assignment,” *Management Science*, **16**(9), 593-603.

Young, H. L. and Pinedo, M. (1997), “Scheduling jobs on parallel machines with sequence-dependent setup times,” *European Journal of Operational Research,* **100**, 464-474.

Zante-de Fokkert J.I. and Kok T.G. (1997), “The mixed and multi model assembly line balancing problem: a comparison”, *European Journal of Operational Research,* **100**, 399-412.

VITA

Pravin Tambe was born in the historic city of Aurangabad on 22nd of June 1980. Aurangabad is world famous for places like Ellora and Ajanta caves, highlights are the paintings spanning the period from 200 BC to 650 AD. The city also known for “Bibi-ka-Makbara” the replica of the Taj Mahal, one of the man made wonders of the world.

Pravin earned his Bachelor of engineering in Production Engineering from Dr. B.A. Marathwada University with Distinction in June 2001. After getting industrial exposure, Pravin joined the graduate program at Louisiana State University in August 2003. His areas of interest are manufacturing systems, operations optimization and supply chain management. He is a candidate for the degree of Master of Science in Industrial Engineering to be awarded at the commencement of May 2006.