RISK ORIENTED STOCHASTIC ASSEMBLY LINE BALANCING

A Thesis by

Prakhash Udayakumar

Bachelor of Engineering, Anna University, India, 2007

Submitted to the Department of Industrial and Manufacturing Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

August 2010

RISK ORIENTED STOCHASTIC ASSEMBLY LINE BALANCING

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Industrial Engineering.

____
Krishna K. Krishnan, Committee Chair

____
Don Malzahn, Committee Member

____
Vinod Namboodiri, Committee Member

ACKNOWLEDGMENTS

I am heartily thankful to my advisor, Dr. Krishna K. Krishnan for his guidance and support throughout my research. I also wish to express my thanks to Dr. Don Malzahn, and Dr. Vinod Namboodiri for serving on my thesis committee.

I owe my most sincere gratitude to my friend Prasanna Tamilselvan for his moral support and encouragement in the completion of this research.

I also thank facilities planning and logistics research group students for their help during the course of my thesis.

I would like to thank my grandparents, parents, uncle and sister for their understanding, care, blessing and support at all times throughout my life.

Lastly, I dedicate this thesis to my grandparents.

ABSTRACT

In a single model manual assembly line, product flows through series of workstations arranged in a sequential manner. Each workstation has a finite number of tasks and each task has probabilistic processing time. Due to the probabilistic nature of task time, the task times can exceed the expected standard task time at some instance. If a series of tasks exceeds in a particular station, then there is a risk that the product may exceed the cycle time. As a result, a small variability in task time can lead to large delays in the delivery lead time of the product.

Most of the line balancing approaches assume deterministic task times thereby ignoring the impact of task time variability on the system performance measures. The larger the variability of task time, the higher the risk associated with the station. In this paper, the impact of variability in task time is quantified in terms of risk. Risk is defined as potential loss caused when the product fails to complete within the specified station time. For line balancing, in addition to cycle time balancing, the risk should be balanced in order to improve the performance of the assembly line. In this research, a risk based assembly line balancing technique for highly variable task times is presented. The results from the case study show that the method increases the performance of the assembly line while balancing the risk of delays at each station.

TABLE OF CONTENTS

Chapter Page

1. INTRODUCTION 1

1.1 Assembly Line 1
1.2 Types of Assembly Line 1
1.3 Assembly Line Balancing 3
1.4 Simple Assembly Line Balancing Problem 4
1.5 Single Model Assembly Line 4
1.5.1 Single Model Stochastic ALBP 5
1.6 Research Purpose and Objectives 6
1.7 Chapter Outline 7

2. LITERATURE REVIEW 8

2.1 Introduction 8
2.2 Assembly Line Balancing Problem 8
2.3 ALBP Approaches 9
2.4 Simple Assembly Line Balancing Problem 11
2.4.1 SALBP – 1 12
2.4.2 SALBP – 2 13
2.4.3 SALBP – E and SALBP – F 13
2.5 Stochastic Single Model Assembly Lines 14
2.5.1 Solution Approaches 14
2.6 Summary 16

3. RISK ASSESSMENT 17

3.1 Introduction 17
3.2 Notations 18
3.3 Assumptions 19
3.4 Relationship Variability and Risk 20
3.5 Data Requirements 27
3.6 Risk Estimation Procedure 28
3.6.1 Risk Index Method 28
3.6.1.1 Delay Index 29
3.6.1.1 Contribution Ratio 29
3.6.1.1 Criticality Ratio 29
3.6.1.1 Risk Index Number 30
3.7 Risk Balancing Methods 30
3.7.1 Mathematical Formulation Method 30
3.7.2 Heuristic Method 31
3.8 Case Study 1 33

TABLE OF CONTENTS (continued)

Chapter Page

3.8.1 Simulation Method 37
3.8.1.1 Task Representation 38
3.8.1.2 Number of Replications 41
3.8.2 Results Generated from Simulation 41
3.9 Case Study 2 44
3.10 Conclusion 50

4. COST BASED RISK ANALYSIS 51

4.1 Introduction 51
4.2 Notations 51
4.3 Risk 52
4.3.1 Risk Estimation Methodology 53
4.4 Risk Balancing Methodology 55
4.4.1 Mathematical Formulation Method 55
4.4.2 Heuristic Method 57
4.5 Case Study 1 58
4.5.1 Validation of Heuristic 63
4.5.2 Results 64
4.6 Case Study 2 66
4.7 Conclusion 70

5. CONCLUSION AND FUTURE WORK 71

5.1 Conclusion 71
5.2 Future Work 72

REFERENCES 73

APPENDICES 78

A. Binary Precedence Table (Case Study 2) 79
B. Cost Function of 40 Tasks (Case Study 2) 80

LIST OF TABLES

Table .... Page

2.1 Different versions of SALBP 12

3.1 Classes of variability 20

3.2 Task times for T_1 and T_2 21

3.3 Statistical parameters of T_1 and T_2 22

3.4 Levels of risk 23

3.5 Low and high variability scenarios for different types of risk 26

3.6 Ratios‘‘ of risk index method 36

3.7 Task order - before and after reallocation 36

3.8 Continuous function parameters calculation for task 5 40

3.9 Continuous function for 16 tasks (case study 1) 40

3.10 Results (case study 1) 42

3.11 Ratios‘‘ of risk index method (case study 2) 45

3.12 Task order - before and after reallocation (case study 2) 46

3.13 Continuous function for 40 tasks (case study 2) 47

3.14 Results (case study 2) 48

4.1 Summary of cost function of 16 tasks (case study 1) 59

4.2 Summary of T_5 task times using simulation (50 Iterations) 60

4.3 Results of risk values of T_5 (50 Iterations) 62

4.4 Risk estimates of 16 tasks (case study 1) 62

4.5 Task order - before and after reallocation (case study 1) 63

LIST OF TABLES (continued)

Table Page

4.6 Results (case study 1).. 64

4.7 Risk Estimates of 40 Tasks (case study 2)... 66

4.8 Task order - before and after reallocation (case study 2)................................ 67

4.9 Results (case study 2).. 68

LIST OF FIGURES

Figure		Page
1.1	Different kinds of assembly lines	1
2.1	Precedence graph	9
3.1	Different regions of risk	23
3.2	Low risk region and low variability	25
3.3	Low risk region and high variability	25
3.4	Heuristic method flow chart	32
3.5	Assembly line layout for case study 1	33
3.6	RI balance chart (case study 1)	37
3.7	Flow chart for simulation methodology	38
3.8	Comparison of value & non-value added time (case study 1)	43
3.9	Throughput (case study 1)	43
3.10	Assembly line layout (case study 2)	44
3.11	RI balance chart (case study 2)	46
3.12	Comparison of value & non-value added time (case study 2)	49
3.13	Throughput (case study 2)	49
4.1	Risk estimation flow chart	55
4.2	Heuristic method flow chart	58
4.3	Risk balance chart (case study 1)	63
4.4	Reduction in non-value added cost (case study 1)	65
4.5	Reduction in total completion time (case study 1)	65

LIST OF FIGURES (continued)

Figure Page

4.6 Risk balance chart (case study 2) ... 67

4.7 Reduction in non-value added cost (case study 2) ... 69

4.8 Reduction in total completion time (case study 2) ... 69

CHAPTER 1

INTRODUCTION

1.1 Assembly Line

Assembly line is a manufacturing system, which is of great importance in large-scale and mass production. The product flows along the line by means of a material handling system, such as conveyors, where parts are added to the product in each workstation arranged successively in a sequential manner. However, in today‟s scenario, manufactures produce greater product variety to meet the customer demands. Under this circumstance, assembly lines traditionally viewed as mass production is now shifting its importance to low volume manufacturing environment.

1.2 Types of Assembly Lines

Boysen, Fliedner, and Scholl (2008) classified assembly lines based on the number of models manufactured in the line, line control, frequency, level of automation, and product types (Figure 1.1) and extended their work by listing the balancing methodology from the literature for all different kinds of assembly lines.

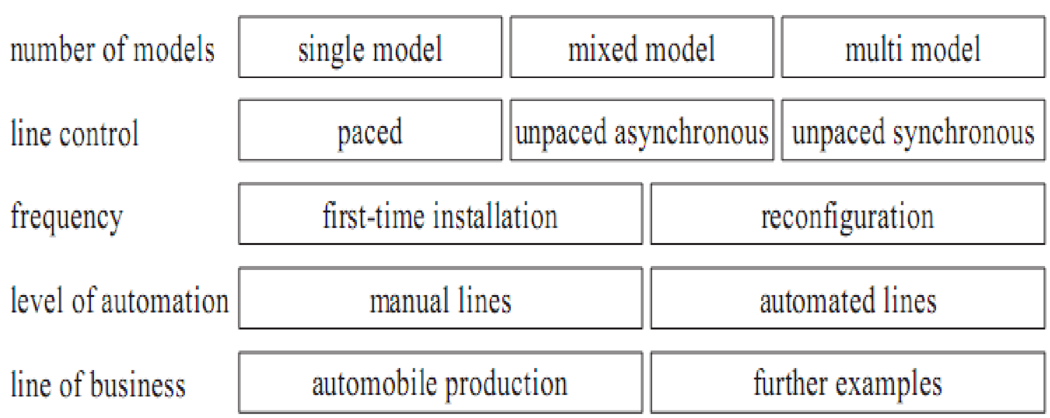


Figure 1.1. Different kinds of assembly lines (Boysen, Fliedner, & Scholl (2008))

Based on the number of models produced, assembly lines are further classified into single, mixed model, and multi model assembly lines. Single model assembly lines are used for high volume production where one product is manufactured in large quantities along the line. It can produce more than one product, when there are not too many variations in set-ups and task times. Mixed model assembly lines produce diverse product mix with similar characteristics in the same line. Multi model assembly lines produce different models with similar or completely different product characteristics in the same line.

Based on control strategies, assembly lines can be further classified into paced line, unpaced asynchronous line, and unpaced synchronous line. In a paced line, product moves continuously along the line in which the assembly workers complete task process before it reaches the end of the respective workstation. In an unpaced asynchronous line, product is transferred to the next workstation provided the task process is completed in the respective workstation and the succeeding workstation is waiting for the product. In an unpaced synchronous line, all the workstations transfer the products to the succeeding station at the same time. The next classification is in terms of frequency, which is further divided into first time installation and reconfiguration. The former denotes new assembly line is set up for newly developed product and latter denotes the assembly line is reconfigured when there is permanent change in the demand of products.

Automated assembly line is similar to paced line, where automated machines instead of assembly workers are used to complete task operations and hence, there is no variability in task times. In manual assembly lines, assembly workers are involved at workstations to perform tasks operations and hence, there exist high variability in task time. Based on product types, assembly lines were developed for the production of unique products such as automobile industry, aircraft manufacturing, etc.

1.3 Assembly Line Balancing

The common problem that arises in any assembly line is assembly line balancing problem (ALBP). ALBP involves optimally distributing the work among the workstations (allocation of tasks to workstations) without violating assignment restrictions based on an objective such as minimizing the total cost, number of workstations, cycle time, maximizing the throughput, efficiency, etc (Baybars, 1986).

The total work required to manufacture the product is partitioned into set of small operations called tasks. Each task takes processing or task time at a workstation to complete its operation without violating precedence constraints. Precedence constraints state that each task must be completed before the succeeding tasks can be started. A set of tasks is allotted to a workstation and the sum of their task times is the station time of the workstation. Cycle time is the fixed move rate for all workstations and is constant for all workstations. Line balancing is said to be feasible, when all the tasks were allocated to the workstations by considering the precedence relationship, provided the station time of all workstations never exceed the cycle time (Boysen, Fliedner, & Scholl, 2007).

Numerous research works have been conducted in this area of ALBP. Salveson (1955) was the first to develop mathematical formulation of ALBP that focused on allocation of tasks to workstations. Boysen, Fliedner, and Scholl (2007) did a comprehensive review of all the research works and proposed a new classification scheme of ALB. They further addressed the limitations in all methods and hence this paper proved to be valuable contribution to further research.

1.4 Simple Assembly Line Balancing Problem

Baybars (1986) defined ALBP as simple assembly line balancing problem (SALBP) which contains the following characteristics: 1) task times is assumed to be deterministic, 2) serial and paced assembly line is considered, 3) assembly line is dedicated to mass production of one product, 4) any task can be completed at any workstations without violating precedence constraints, 5) all workstations are equipped to complete any type of task, and 6) cycle time is fixed and is constant for all workstations.

Based on the objective, SALBP has different versions. Jackson (1956), Freeman (1968), Mastor (1970), Pinto, Dannenbring, and Khumawala (1978), Johnson (1983), Scholl, and Vob (1996), and Peeters and Degraeve (2006) solved SALBP-1 based on minimizing the number of stations along the line for the given cycle time. The next version, SALBP-2 is based on minimizing the cycle time to improve the throughput along the line for a given number of stations (Mastor, 1970, Scholl, & Vob, 1996, Ugurdag, Rachamadugu, and Papachristou, 1997). If both, cycle time and number of workstations can be minimized considering their interrelationship, SALBP-E maximizes line efficiency, which thereby minimizes idle times. Generally, line efficiency is defined as the ratio of total task time to the product of cycle time and number of stations (Boysen, Fliedner, & Scholl, 2007). SALBP-F seeks for feasible line balance for the given combinations of cycle time and number of stations (Scholl, 1999).

1.5 Single Model Assembly Line

Heuristics and optimal methods were developed to solve these different SALBP respectively. Most research works assume deterministic task times and the methods developed is not practical for application in a realistic environment. In manufacturing of products such as aircraft, there is a high variability in task times. This is because most tasks involve manual activities and as a result, task times vary from one unit to the next. This research work focuses on single model, manual assembly line with stochastic completion times for tasks.

1.5.1 Single Model Stochastic ALBP

Most research works performed on SALB problems assumed processing times of tasks to be deterministic. However, task times may vary from cycle to cycle and relatively less work is focused on stochastic ALB problems. Moodie and Young (1965) proposed a two-step heuristic procedure aimed at assigning tasks among workstations so to obtain minimum number of workstations for a given cycle time and also to reduce the probability of workstations not exceeding the cycle time in a moving line. The heuristic incorporates variability by considering normally distributed task times with given mean and variance. Reeve and Thomas (1973) developed and compared four solution procedures (trade and transfer, branch and bound, heuristic branch and bound, and BABTAT) aimed at assigning tasks among work such that the probability of line stoppage is minimized in a moving line. The four procedures handles variability by considering normally distributed task times with a given mean and variance.

Shin (1990) developed a heuristic procedure for solving stochastic SALB problem by considering normally distributed task times. This methodology finds an optimal task allocation to workstations by minimizing the total expected cost (operating cost and incompletion cost) for the given cycle time over a feasible range. Suresh and Sahu (1994) used a simulated annealing technique to solve stochastic SALB problems and compared their results with two other methods developed by Moodie and Young (1965), and Reeve (1971).

Nkasu and Leung (1995) proposed stochastic ALB algorithm that incorporates Monte-Carlo simulation procedure to handle stochastic cycle and task times with a choice of different probability distributions. This methodology aimed at minimizing cycle time, number of stations, balance delay and combination of two or more and the results presented used normally distributed task and cycle time. Lyu (1997) proposed single-run optimization algorithm for solving stochastic SALB problems and the algorithm used normally distributed task times. The proposed algorithm yielded better results when compared with two other algorithms (enumerative and shin2) in terms of cost. Sarin, Erel, and Dar-El (1999) developed a heuristic enumeration technique to minimize the total cost (labor cost and incompletion cost) for a single-model stochastic moving line SALB.

Scholl and Becker (2006) performed an extended survey of SALBP, which provided an up-to-date research on single model assembly lines. This paper reviewed the algorithms and heuristic procedures developed for solving both deterministic and stochastic single model assembly lines.

1.6 Research Purpose and Objectives

In a single model manual assembly line, product flows through series of workstations arranged in a sequential manner. Each workstation has a finite number of tasks and each task has probabilistic processing time. Due to the probabilistic nature of task time, the task times can exceed the expected standard task time at some instance. If a series of tasks exceeds in a particular station, then there is a risk that the product may exceed the cycle time. The larger the variability of task time, the higher the risk associated with the station. Risk is defined as potential loss caused when the product fails to complete within the specified station time. For line balancing, in addition to cycle time balancing, the risk should be balanced in order to improve the performance of the assembly line.

Hence, the objectives of this thesis are:

- Risk based Assembly Line Balancing

- Identify relationship between risk and variability
- Propose new method to measure risk
- Develop mathematical and heuristic method for risk balancing for an assembly line
- Investigate the impact on value and non-value added time

1.7 Chapter Outline

This thesis is categorized into five chapters. Chapter 2 details the literature review of all the research work existing in the field of assembly line balancing. Chapter 3 will provide development of risk measurement method. It also includes assessment of relationship between variability and risk, different levels of risk, risk based ALB formulation, and the solution approaches for risk balancing. Chapter 4 presents risk estimation procedure in terms of cost, risk based ALB formulation, and the solution approaches for risk balancing. Chapter 5 provides the conclusion and future work of this research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This research work addresses the assembly line balancing problem (ALBP) in a stochastic environment. This chapter provides literature review on all ALBP and the corresponding methodology proposed to solve ALBP. Section 2.2 explains ALBP and its characteristics. Section 2.3 lists all the approaches from literature that were used to solve ALBP. Section 2.4 discusses different versions of simple assembly line balancing problems (SALBP) and the various approaches for solving SALBP. Section 2.5 discusses researches on single model assembly lines. Finally, section 2.6 summarizes the literature review on ALBP.

2.2 Assembly Line Balancing Problem

ALBP is one of the important problems among the decision makers in a manufacturing environment. ALBP involves optimally distributing the work among the workstations (allocation of tasks to workstations) without violating assignment restrictions based on an objective such as minimizing the total cost, number of workstations, cycle time, maximizing the throughput, efficiency, etc (Baybars, 1986). Hence, the total work required to assemble the product is partitioned and balanced among all the workstation such that idle time is low in order to have high productivity.

The total work is split into set of small operations called tasks. Each task requires processing or task time at a workstation to complete its operation. Tasks cannot be allotted to a workstation in a random sequence due to technological constraints and cycle time constraints. Technological constraints include precedence constraints, which state that each task must be completed before the succeeding tasks can be started. They are represented graphically by a precedence graph (Figure 2.1). From Figure 2.1, the nodes represent task, node weights represent task time and arcs represent precedence relationship between tasks.

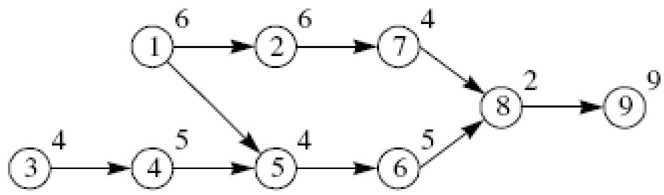


Figure 2.1 Precedence graph (Boysen, Fliedner, & Scholl (2007))

Another method of representing precedence constraints is precedence matrix (Hoffmann, 1963). Rows and columns represent task in a binary square matrix. If task i is precedent to task j, then ij^{th} element becomes 1. If there is no precedence between tasks, then ij^{th} element becomes 0. Cycle time constraints include sum of the task times allotted to a workstations which represents station time should never exceed cycle time of the assembly line. Cycle time represents the move rate of the workstation and is kept constant for all workstations.

Any assembly line looks for the feasible line balance to reduce the idle times. Assembly line balancing is said to be feasible, when all the tasks were allocated to the workstations without violating assignment restrictions, provided the station time of all workstations never exceed the cycle time (Boysen, Fliedner, & Scholl, 2007).

2.3 ALBP Approaches

Due to the intricacies of ALBP, solving the mathematical model by optimization procedures is not possible. Hence, by considering numerous simplifying assumptions, Salveson (1955) was the first to develop linear programming model of ALBP that focused on allocation of tasks to workstations. White (1961) and Thangavelu, and Shetty (1971) modeled the ALBP by changing linear programming to 0-1 program. Talbot and Patterson (1984) formulated the ALBP with integer programming by not considering binary variables. Enumerative techniques (branch and bound method, and dynamic programming methods) have been widely used for solving integer programming models developed for ALBP. Scholl and Becker (2006) presented a detailed survey on branch and bound procedures that were used for solving ALBP. Jackson (1956) first developed the other enumerative technique dynamic programming method for solving ALBP. Held, Karp, and Shareshian (1963), and Kao and Queyranne (1982) extended their work on dynamic programming procedures for solving ALBP.

Since most of the ALBP are known to be NP-hard (Baybars, 1986), these methods becomes impossible for optimization due to large number of work elements. This is because, reduction of ALB leads to a partition problem and partition problems are proved to be NP-hard (Karp, 1972). In such cases, heuristic method is the best method to handle line balancing problems.

A large variety of heuristic methods such as constructive methods, genetic algorithms, local search and meta strategies, simulated annealing, LP based improvement procedure and ant colony optimization approach were developed for solving SLBP. Constructive heuristic methods (Talbot & Patterson, 1986, Hackman, Magazine & Wee, 1989, and Scholl & Vob, 1996) include priority rule based procedure, incomplete enumeration procedure and search methods were used to find one or more feasible solutions. Genetic algorithms (Anderson & Ferris, 1994, Ponnambalam, Aravindan & Naidu, 2000, Sabuncuoglu, Erel, & Tanyer, 2000, and Goncalves & Almeida, 2002), meta strategies such as tabu search (Heinrici, 1994, Chiang, 1998, and Scholl & Vob, 1996) focuses on solving ALBP. Most of these approaches described above were employed for finding the optimal solutions from the initial solution found using heuristic methods.

Extensive research work in the field of assembly line balancing has been dedicated to simple assembly line balancing problem (SALBP) due to numerous simplifying assumptions. The next section will discuss about SALBP, different versions of SALBP and the procedures developed to solve SALBP.

2.4 Simple Assembly Line Balancing Problem

Baybars (1986) defined ALBP as simple assembly line balancing problem (SALBP) which contains the following characteristics: 1) task times is assumed to be deterministic, 2) serial and paced assembly line is considered, 3) assembly line is dedicated to mass production of one product, 4) any task can be completed at any workstations without violating precedence constraints, 5) all workstations are equipped to complete any type of task, and 6) cycle time is fixed and is constant for all workstations.

Based on an objective, SALBP has different versions. SALBP-1 is based on minimizing the number of stations along the line for the given cycle time. The next version, SALBP-2 is based on minimizing the cycle time to improve the throughput along the line for a given number of stations. If both, cycle time and number of workstations can be minimized considering their interrelationship, SALBP-E maximizes line efficiency, which thereby minimizes idle times. Generally, line efficiency is defined as ratio of total task time to the product of cycle time and number of stations (Boysen, Fliedner, & Scholl, 2007).

$$\text{Line Efficiency } E = \frac{t_{sum}}{m \times c} \tag{2.1}$$

Where,

m Total number of workstations

n Total number of tasks

c Cycle time

t task time

t_{sum} Total task time, $\sum_{j=1}^{n} t_j$

SALBP-F seeks for feasible line balance for the given combination of cycle time and number of workstations. Table 2.1 shows different versions of SALBP (Scholl, & Becker, 2006).

Table 2.1. Different versions of SALBP (Scholl, & Becker, 2006)

No. *m* of stations	Cycle Time *c*	
	Given	Minimize
Given	SALBP-F	SALBP-2
Minimize	SALBP-1	SALBP-E

Most of the research works focused on solving SALBP-1 and the frequently used approaches to obtain the exact solution include dynamic programming procedure, heuristic method and branch and bound method. SALBP-2 and SALBP-E mostly use direct heuristic methods for solutions similar to SALBP-1. They also employ search methods to obtain feasible line balance (SALBP-F) by repeated iteration using SALBP-1 methods.

2.4.1 SALBP – 1

The methods developed to solve SALBP-1 focus on minimizing the number of workstation along the assembly line for the given cycle time. Held, Karp and Shareshian (1963) modified dynamic programming method proposed by Jackson (1956) and the optimal solution is obtained by searching in stages in a forward recursion. If precedence constraints are ignored, SALBP-1 is reduced to bin packing problem (Baybars, 1986). Johnson (1988) developed a branch and bound algorithm „FABLE" for finding the optimal number of workstations and the algorithm is efficient as it can handle more than 1000 tasks with less computational time. Scholl and Vob (1996) developed unidirectional and bidirectional priority based heuristic method for solving SALBP-1. The solution obtained from the heuristic is further improved by the application of tabu search method.

Mcmullen and Frazier (1998) developed simulated annealing approach for SALBP -1 and the approach can handle parallel workstations and stochastic task times. Sprecher (1999) developed branch and bound method guided by precedence tree to solve SALBP-1. Bautista and Pereira (2002) developed an ant colony algorithm based on priority rule method to solve SALBP-1. Peeters and Degraeve (2006) proposed column generation algorithm to obtain the lower bound by means of Ddantzig-wolfe decomposition of integer programming formulation of SALBP-1.

2.4.2 SALBP – 2

The methods developed to solve SALBP-2 focus on minimizing the cycle time to improve the throughput along the assembly line for a given number of workstations. Dar-el and Rubinovitch (1979) developed a search method based on Mansoor-Yyadin algorithm to solve SALBP-2. Heinrici (1994) developed simulated annealing approach and tabu search method for solving SALPB-2. Scholl and Vob (1996) developed improvement procedure for SALBP-2 which is based on swaps and shifts. This method starts from a feasible solution and couples with tabu search to improve the given feasible solution. Ugurdag, Rachamadugu, and Papachristou (1997) proposed two step heuristic method with additional vertical balancing for solving SALBP-2. The method is based on integer programming and the initial solution is obtained by using priority rule search method. The initial solution is improved simplex type improvement method.

2.4.3 SALBP – E and SALBP – F

In the first case, if both cycle time and number of workstations can be minimized considering their interrelationship, SALBP-E maximizes line efficiency, which thereby minimizes idle times. SALBP-F seeks feasible line balance for the given combination of cycle time and number of workstations. In the case of SALBP-E, feasible combinations of number of workstations and cycle time (SALBP-F) are initially defined. SALBP-F is NP – hard and optimization techniques cannot be used to find the feasible combinations (Scholl, 1999). The heuristic procedures developed for SALBP-1 and SALBP-2 were employed to find the feasible combinations (SALBP-F). From the set of feasible combinations, the combination with maximum line efficiency becomes the optimal solution. Scholl and Vob (1995) employed search method for finding the optimal combination of cycle time and number of workstations.

2.5 Stochastic Single Model Assembly Lines

Most research works performed on SALBP assumed processing times of tasks to be deterministic and hence, heuristics and algorithms developed for performing line balancing is hardly practical for application in a realistic environment. In manufacturing of products such as aircraft, there is a high variability in task times. This is due to the fact that most tasks involve manual activities and as a result, task times vary from one unit to the next. This research work focuses on single model manual assembly line with stochastic completion times for tasks.

2.5.1 Solution Approaches

Moodie and Young (1965) was the first to address ALBP with stochastic task times. They proposed a two-step heuristic procedure aimed at assigning tasks among workstations so to obtain minimum number of workstations for a given cycle time (SALBP-1) and also to reduce the probability of workstations not exceeding the cycle time in a moving line. The heuristic incorporates variability by considering normally distributed task times with given mean and variance. Reeve and Thomas (1973) developed and compared four solution procedures (trade and transfer, branch and bound, heuristic branch and bound, and BABTAT) aimed at assigning tasks among work such that the probability of line stoppage is minimized in a moving line. The four procedures handles variability by considering normally distributed task times with a given mean and variance.

Kottas and Lau (1981) developed a heuristic procedure for assigning tasks to workstations which aimed at minimizing the cost which includes incompletion and labor cost. Shin (1990) developed a heuristic procedure for solving stochastic SALB problem by considering normally distributed task times. This methodology finds an optimal task allocation to workstations by minimizing the total expected cost (operating cost and incompletion cost) for the given cycle time over a feasible range. Suresh and Sahu (1994) used a simulated annealing technique to solve stochastic SALB problems in a moving line and compared their results with two other methods developed by Moodie, and Young (1965), and Reeve (1971).

Nkasu and Leung (1995) proposed stochastic ALBP algorithm that incorporates Monte-carlo simulation procedure to handle stochastic cycle and task times with a choice of different probability distributions. This methodology aimed at minimizing cycle time, number of stations, balance delay and combination of two or more and the results presented used normally distributed task and cycle time. Lyu (1997) proposed single-run optimization algorithm for solving stochastic SALB problems and the algorithm used normally distributed task times. The proposed algorithm yielded better results when compared with two other algorithms (enumerative and shin2) in terms of cost. Sarin, Erel, and Dar-El (1999) developed a heuristic enumeration technique to minimize the total cost (labor cost and incompletion cost) for a single-model stochastic moving line ALBP. Dynamic programming procedure is used to obtain the initial solution and the solution is improved by branch and bound enumeration technique. The results were compared with Kottas and Lau (1981) and the total costs obtained were 6.2% less than the other method.

Scholl and Becker (2006) performed an extended survey of SALBP, which provided an up-to-date research on single model assembly lines. This paper reviews the algorithms, heuristic procedures developed for solving both deterministic and stochastic single model assembly lines.

2.6 Summary

Even though extensive research works have been conducted in the area of ALB, there is no research work focused on solving ALB from the perspective of risk. Risk is defined as potential loss caused when the product fails to complete within the specified cycle time. Due to probabilistic nature of task time, the task can exceed the expected standard task time at some instance. If a series of tasks exceeds in a particular station, then there is a risk that the product may exceed the cycle time. The larger the variability of task time, the higher the risk associated with the station.

Hence, in addition to cycle time balancing, the risk should be balanced in order to reduce the non-value added time (waiting time and idle time) which in turn improves the productivity of the assembly line. In the following chapters, risk based assembly line balancing procedure is presented. It includes detailed assessment of relationship between variability in task time and risk, risk estimation procedure, risk based ALB formulation, and the solution approaches for risk balancing.

CHAPTER 3

RISK ASSESSMENT

3.1 Introduction

In general, simple assembly line balancing (SALB) problem seeks to achieve an optimal balance by distributing the workload along the line such that there is ample time to complete all tasks within a specific cycle time. Assignment of tasks to workstation is either based on objectives such as minimizing cycle time, number of workstations, and cost, or maximizing line efficiency. Heuristics and algorithms used for performing line balancing often assume deterministic task times and the line is dedicated to a single product.

In the manufacture of products such as aircraft manufacturing, these assumptions need to be relaxed in order to reflect practical assembly line. The most common of all the assumptions is deterministic task times. Since tasks can be manual, the stochastic nature of task times is to be considered. Because of this variability, tasks may not be completed in specified cycle time and this may lead to delay in product completion. The line may be balanced in terms of cycle time but if any workstation has more high variability tasks than other workstation, then there is a risk associated with that station. Modarres (2006) defined risk as “a measure of the potential loss occurring due to natural or human activities”. In this perspective, risk is defined as the potential loss caused when the product fails to complete within the specified station time.

In this chapter, relationship between variability and risk, and different levels of risk are established. A new risk index method is proposed to capture risk. An effective heuristic procedure is developed to balance the risk along the assembly line. Above all, an efficient assembly line is the ability to maintain its efficiency against the variability in task times.

3.2 Notations

Baybars (1986) presented the following notations for solving assembly line balancing and the same notations will be used in this research.

m Total number of tasks

n Total number of workstations

I Set of tasks, where $I = \{I_1, I_2, \dots \dots, I_m\}$

J Set of workstations, where $J = \{J_1, J_2, \dots \dots, J_n\}$

x_{ij} $\left.\begin{cases} 1 & \text{if task i is assigned to station j,} \\ 0 & \text{otherwise,} \end{cases}\right\} \forall\, i \in I \text{ and } j \in J$

t_i Standard task time for each task i

T Cycle time or fixed move rate at all workstations

S_j Total allotted station time, $S_j = \sum_{i=1}^{m} (t_i \times x_{ij})$

a_i Risk index of task i

B_j Risk index of the station j

P_{ij} Binary precedence table $\left.\begin{cases} 1, & \text{task i is precedent to task j} \\ 0, & \text{no precedence} \end{cases}\right\},$

$$P_{ij} = \begin{pmatrix} P_{11} & \cdots & P_{1m} \\ \vdots & \ddots & \vdots \\ P_{m1} & \cdots & P_{mm} \end{pmatrix} \forall\, i, j \in I$$

C_p Total cycle time of the product, $C_p = \sum_{i=1}^{m} t_i$

The following are the additional list of notations used in this research.

l Total number of historical data of the task

L Set of historical data (actual task time) for a task, where $1 \leq L \leq 1$

E Number of instances the task exceeds the standard task time

A_i Average task time

C_v Coefficient of variation

D Delay index ratio

d Standard delay index threshold value set by the user

k Standard K-factor threshold value set by the user

W_j Waiting time of station j

I_j Idle time of station j

N_j Non value added time of station j, $N_j = W_j + I_j$

K Number of intervals

W Interval width

R Range, maximum value – minimum value

N_r Number of replications

t T distribution with 95% confidence level

σ Standard deviation

h Half width of the population

T_h Throughput

C Contribution ratio

U Criticality ratio

RI Risk index number

3.3 Assumptions

The following assumptions are considered for the analysis of SALB:

1. Single product dedicated assembly line
2. Assembly line with no buffers
3. Task can be allocated to any workstation without violating precedence relationship
4. Serial line layout is considered
5. All task times follow continuous distribution
6. Workstation can have any number of task and no parallel tasking is allowed

3.4 Relationship Variability and Risk

Variability exists in processing times of all tasks in production environment due to operator delays, setup delays, and other external influences. It can have huge impact on performance of the production system (Hopp, & Spearman, 2001). Due to this variability, task may tend to exceed the standard task time and this will lead to delay in product completion. In order to analyze variability, coefficient of variation (C_v) is measured. C_v is the measure of dispersion of a random variable and it is represented by standard deviation divided by the mean. Based on C_v, variability is classified into three classes (Table 3.1):

Table 3.1. Classes of variability (Hopp, & Spearman, 2001)

Variability Class	Coefficient of Variation (C_v)
Low variability (LV)	$C_v < 0.75$
Medium Variability (MV)	$0.75 \leq C_v < 1.33$
High Variability (HV)	$C_v \geq 1.33$

In the context of line balancing under high variability, risk of the task is defined as the potential loss caused when the product fails to complete within the specified task time. There are two components to this risk: a) the number of times that the task times exceed the standard task time and b) the magnitude of the deviation of the completion time from the standard task time.

The first factor is addressed by the delay index. Delay index (D) is defined as the number of instances in which actual task time exceed the standard task time divided by total number of instances. The value of D ranges from 0 to 1.

$$\text{Delay Index (D)} = \frac{E}{l} \tag{3.1}$$

The second factor is defined by the K-factor, which is the ratio of average task time to standard task time.

$$\text{K-factor} = \frac{A_i}{t_i} \tag{3.2}$$

If the average task time is equal to standard task time, then the K-factor is 1. As K-factor value increases above one, it indicates that the average time taken to complete the job is greater than the standard time.

Under this definition of risk, a highly variable task does not necessarily indicate a high risk. To understand the relationship between risk and variability, consider the data shown in Table 3.2, which consists of two tasks and the actual time taken for performing them in the last 12 production runs. Table 3.3 shows the statistical parameters of the corresponding tasks such as C_v, standard task time, average task time, number of instances the actual task time exceed the standard task time from 12 production runs, K-factor, and delay index. The standard task time is the allotted time for completing the task.

Table 3.2. Task times for T_1 and T_2

Task	**Task Times**											
T_1	28.8	34.6	34.9	36.3	37.2	39.2	40.5	44	44.1	46.5	46.9	51.6
T_2	1.2	1.8	1.9	2.4	2.9	3	3	4.9	8.5	9.4	11.9	13.2

Table 3.3. Statistical parameters of T_1 and T_2

Task	t_i	A_i	C_v	E	K-factor	D
T_1	18	40.38	0.36	12	2.24	1
T_2	3.1	5.17	1.37	5	1.68	0.42

Average time taken to perform task 1 in the 12 production runs is 2.24 times the standard task time and delay index is 1. This is a high-risk task as all the values fall beyond the standard task time even though the C_v indicates that it has low variability. On the other hand, for task 2, delay index is 0.42 that is less than T1 and the average task time is 1.68 times the standard task time. The C_v indicates that task 2 is highly variable, but the delay index and K-factor indicate that the risk is not as high as T_1. Based on this analysis of the two tasks, it can be concluded that variability is not directly proportional to risk.

It is evident that variability measure is not sufficient to predict the risk of the task. Risk is measured by the K-factor and the delay index. Based on these two factors, new levels are proposed to categorize risk.

Level 1a: Delay Index (D) is less than d

Level 1b: Delay index (D) is greater than or equal to d

Level 2a: K-factor is less than or equal to 1

Level 2b: K-factor is greater than 1 but less than k

Level 2c: K-factor is greater than or equal to k

Based on these levels, three types of risk low, medium and high are defined. Medium risk is further classified into four levels, medium risk 1, 2, 3, and 4 respectively. Table 3.4 shows the levels of risk and Figure 3.1 shows segregation of different regions of risk bounded by the constraints.

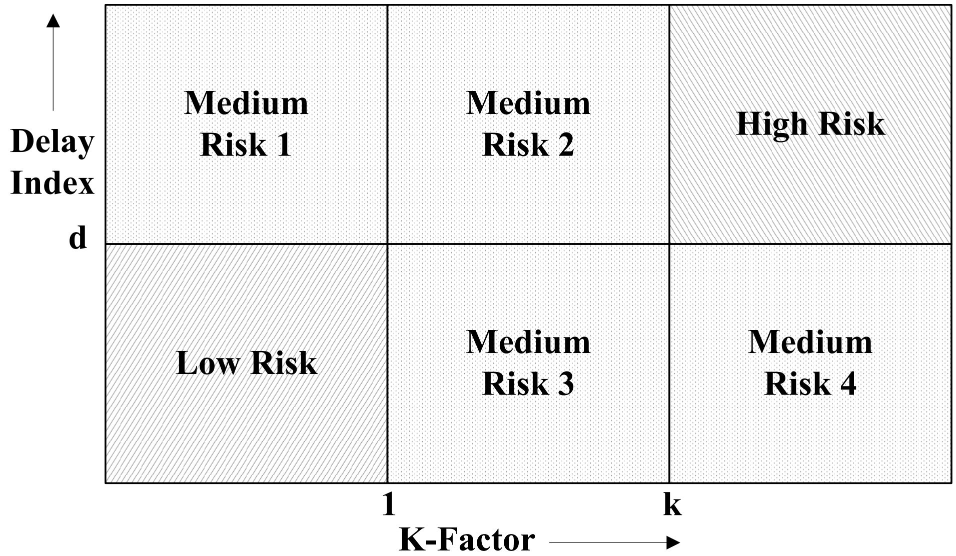


Figure 3.1. Different regions of risk

Table 3.4. Levels of risk

Types of Risk	**Level 1a**	**Level 1b**	**Level 2a**	**Level 2b**	**Level 2c**
Low Risk	True	False	True	False	False
Medium Risk 1	False	True	True	False	False
Medium Risk 2	False	True	False	True	False
Medium Risk 3	True	False	False	True	False
Medium Risk 4	True	False	False	False	True
High Risk	False	True	False	False	True

For each task, the value of D and K-factor is calculated and compared with standard values to determine the level of risk. The standard values include, delay index threshold value (d), and K-factor threshold value (k) are set by the user based on acceptable risk levels which the system can afford to allow. The delay index (d) increases with increase in number of instances in which the actual task time exceeds the allotted task time. If d is set close to zero, then the system will be considered at high risk even if the number of times that the allotted task time is exceeded is small. A low threshold value of d is used when the delay cost is very high. If the value of d is set close to one, then the system shows no indication of risk even if most of the actual task times exceed the allotted task time. A high value of d is used when the delay cost is low.

Similarly, K-factor shows the deviation of average task time from standard task time. When K-factor is less than or equal to 1 (Level 2a), it shows that that average task time is less than or equal to standard task time. This gives an indication that in most instances the tasks are completed on or before standard task time.

When k value is at level 2b and 2c, it represents the acceptable deviation level which the system can allow beyond standard task time. If the value of k is set close to 1, then the system will be considered at high risk even if a few actual task times exceed the allotted task time. If k is set much higher than 1, it indicates that the system is not under risk even if most of the actual task times exceed the allotted task time. i.e. the impact of the delay is not high in terms of cost. An appropriate p and k value is chosen based on the ability of the manufacturing system to absorb delay costs.

For demonstration, consider the same data in Table 3.2 and 3.3. User sets the standard values, d and k and in this case, it is assumed to be 0.50 and 1.3. For T_1, both the factors D and K-factor are greater than standard values, the task falls under the high risk category. Similarly, for the task T_2, the value of K-factor is greater than standard k value but the delay index (D) is less than standard d value. Hence, T_2 falls under the medium risk 4 category.

Each region of risk has all levels of variability. Once the risk is categorized, variability is analyzed by measuring coefficient of variation. Consider the low risk region with low and high variability scenarios shown in Figure 3.2 and Figure 3.3. In Figure 3.2, the measure of dispersion of task times is low which indicates task with low variability lies in the low risk region. On the other hand (Figure 3.3), the measure of dispersion of task times is spread over a wide area which indicates high variability but still the task lies in the low risk region.

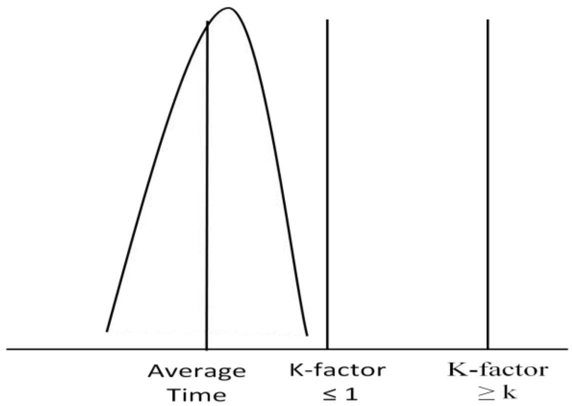


Figure 3.2. Low risk region and low variability

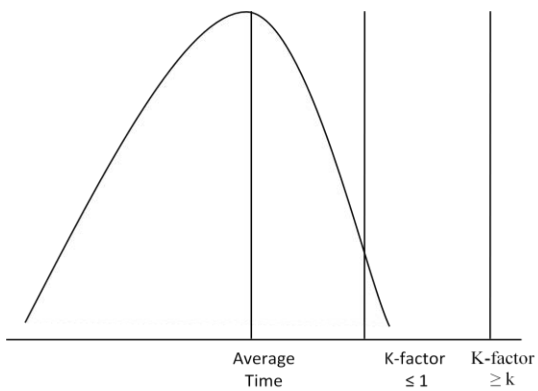
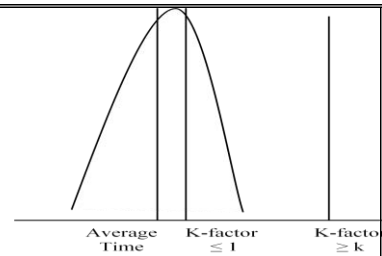
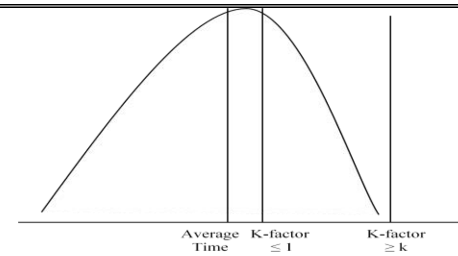
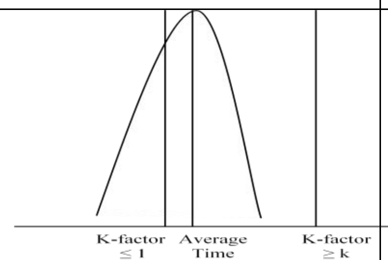
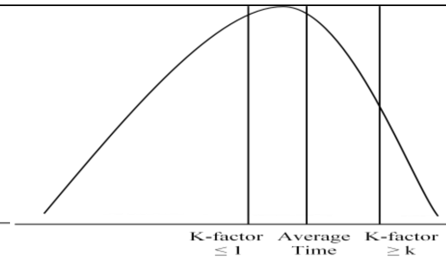
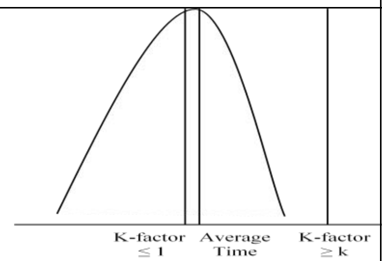
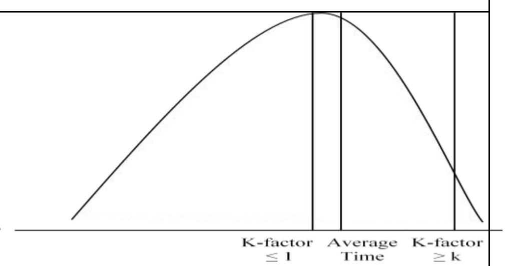
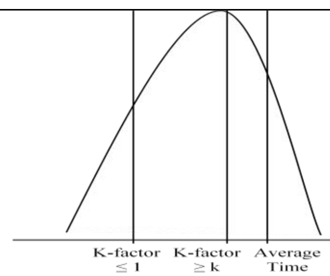
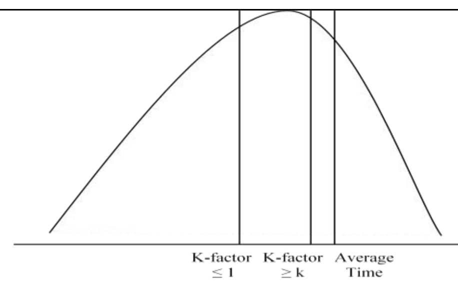
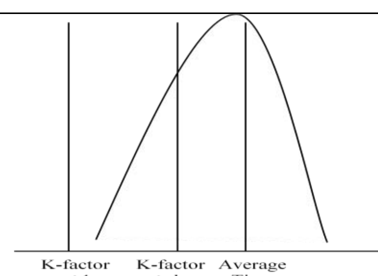
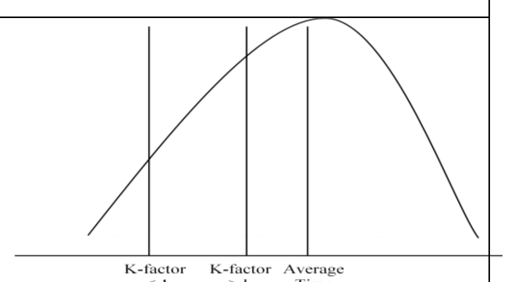


Figure 3.3. Low risk region and high variability

Similarly, Table 3.5 provides low and high variability scenarios for other regions of risk.

Table 3.5. Low and high variability scenarios for different types of risk

Types of Risk	**Low Variability**	**High Variability**

Medium risk 1		
Medium risk 2		
Medium risk 3		
Medium risk 4		
High risk		

Based on this analysis, it can be concluded that three classes of variability are possible on all regions of risk. Therefore, variability may or may not influence risk.

3.5 Data Requirements

To do risk analysis for an assembly line, data such as set of tasks and workstations, standard task time of all tasks, precedence constraints and a set of historical data of all tasks, are collected. They are represented mathematically as follows:

$$I = \{I_1, I_2, \ldots \ldots, I_m\} \tag{3.3}$$

$$J = \{J_1, J_2, \ldots \ldots, J_n\} \tag{3.4}$$

$$t = \{t_1, t_2, \ldots \ldots, t_m\} \tag{3.5}$$

In this research work, precedence constraints are the only assignment restriction for task allocation to workstations. They are represented as binary precedence table P_{ij}, as task i should be completed before task j.

$$P_{ij} = \begin{pmatrix} P_{11} & \cdots & P_{1m} \\ \vdots & \ddots & \vdots \\ P_{m1} & \cdots & P_{mm} \end{pmatrix} \forall\ i, j\ \in\ I \tag{3.6}$$

For a task, „L" historical set of actual task times are represented as shown below:

$$L_{xy} = \begin{pmatrix} L_{11} & \cdots & L_{i1} \\ \vdots & \ddots & \vdots \\ L_{1l} & \cdots & L_{il} \end{pmatrix} \forall\ x \in I, y \in L \tag{3.7}$$

Average task time is calculated by the summation of „L" historical actual task times for a task divided by number of „L" values and is represented as shown in equation 3.8.

$$A_i = \frac{\sum_{L=1}^{l} L_{Li}}{L}\ \forall\ i \in I \tag{3.8}$$

$$A = \{A_1, A_2, \ldots \ldots, A_m\} \tag{3.9}$$

All workstations have the same cycle time or fixed move rate. Total allotted station time (T_j) is the sum of the task times assigned to a workstation and therefore, allotted station time for all workstations should not exceed cycle time.

$$S_j = \sum_{i=1}^{m} (t_i \times x_{ij}) \quad \forall S_j \leq T \tag{3.10}$$

3.6 Risk Estimation Procedure

In the previous section, the data required to assess the risk is determined. In this section, a new method is proposed for the estimation of risk. As defined previously, there are two components of risk: a) the number of times that the task times exceed the standard task time and b) the magnitude of the deviation of the completion time from the standard task time. They were addressed by delay index and K-factor respectively. In addition to the above mentioned two components, processing or task time is also another factor that contributes to risk. Consider a task with a high processing time and lies in the high-risk region. Such a task amplifies the other two components and lead to much higher risk. For line balancing, in addition to cycle time balancing, the previously described three components of risk should be balanced in order to yield an increased throughput.

3.6.1 Risk Index Method

The objective of this work is to distribute the risk across the assembly line by reallocation of tasks among workstations based on newly developed risk index method. For each task, the method computes the risk index (RI) by multiplying three newly developed ratios‶, delay index, criticality ratio and contribution ratio. The three ratios‶ addresses the three components of risk. Delay index addresses the frequency, contribution ratio addresses the task time, and criticality ratio addresses the magnitude of deviation of task time away from the allotted task time. Hence, risk is captured in terms of RI. Risk distribution is defined as reassignment of tasks among workstations without violating precedence relationship by balancing the RI between workstations.

These three ratios‟ proposed for the calculation of RI is discussed in detail as follows.

3.6.1.1 Delay Index

As defined previously, delay index (D) is defined as the number of data points exceed the standard task time (E) divided by total number of data points (*l*). This ratio addresses the frequency that the task exceeds allotted task time.

$$\text{Delay Index (D)} = \frac{E_i}{l_i} \tag{3.11}$$

The value of D ranges from 0 (low) to 1 (high). If the value of D is low, then the task exceeds less frequently which in turn reduces the RI and thereby, shows that the task is at low risk. Similarly, if the value of D is high, then the task exceeds more frequently and contributes for an increase in RI.

3.6.1.2 Contribution Ratio

Contribution ratio (C) is defined as the standard task time (t_i) divided by total cycle time of the product (C_p). This ratio illustrates the contribution of each task to its total cycle time.

$$\text{Contribution Ratio (C)} = \frac{t_i}{C_p} \tag{3.12}$$

The value of C ranges from 0 (low) to 1 (high). When C is low, the task has a less processing time and the contribution to risk is also low. On the other hand, when C is high, the task has a high processing time and the contribution to risk is high.

3.6.1.3 Criticality Ratio

Criticality ratio (U) is defined as the ratio of standard task time (t_i) to average task time (A_i). This ratio addresses the magnitude of deviation away from the standard task time.

$$\text{Criticality Ratio (U)} = 1 - \frac{t_i}{A_i} \qquad (3.13)$$

The value of U will be less than 0 when average task time is less than the standard task time. U must be greater than 0. So, for instances, when U is negative, the value of U is set to 0.001. The value of U ranges from 0 (low) to 1 (high).

3.6.1.4 Risk Index Number

Risk index number is calculated by multiplying the above three ratios. It is given by,

$$RI = P \times C \times U \times 1000 \qquad (3.14)$$

3.7 Risk Balancing Methods

In this section, method used to balance the risk is addressed. Risk can be balanced by two methods: a) mathematical formulation method, and b) heuristic method. As mentioned earlier, risk balancing is distributing the RI among workstations by reassigning the tasks to workstations without violating the task assignment restrictions. Both the methods aim in minimizing the maximum difference in RI for all workstations.

3.7.1 Mathematical Formulation Method

The mathematical model for reallocation of tasks among workstation based on RI is shown below.

$$\text{Min Max} \, | B_j - B_k | \quad \forall\, j, k \in J, \text{ and } j \neq k \qquad (3.15)$$

Subject to:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i \in I \qquad (3.16)$$

$$\sum_{i=1}^{m} (t_i \times x_{ij}) \leq T \quad \forall j \in J \tag{3.17}$$

$$P_{ij} + P_{ji} \leq 1 \tag{3.18}$$

$$B_j = \sum_{i=1}^{m} (x_{ij} \times a_i) \quad \forall j \in J \tag{3.19}$$

$$P_{ij} = 0 \quad \forall i = j \tag{3.20}$$

$$x_{ij} = 0, 1 \tag{3.21}$$

Objective function (3.15) represents minimizing the maximum difference in RI between workstations. Constraint (3.16) ensures that every task is allotted to a workstation and each task is allotted only once to a workstation. Constraint (3.17) guarantees that station time of all workstations never exceed the cycle time. Constraint (3.18) represents precedence constraints. Constraint (3.19) represents the RI calculation for all workstations. Constraint (20) ensures the task cannot precede the same task. Constraint (3.21) represents x_{ij} can assume values either 0 or 1.

The model shown above is an ALB problem and it is known that ALB problems are NP-hard (Baybars, 1986). This is because, reduction of ALB leads to a partition problem and partition problems are proved to be NP-hard (Karp, 1972). Hence, the problem becomes impossible to solve due to the presence of large number of work elements, partition restrictions, and feasible different combination of task allocation between workstations. In such cases, heuristic method is the best method to handle line balancing problems.

3.7.2 Heuristic Method

The heuristic method is developed to balance the risk (minimizes the maximum difference in RI) between workstations without violating assignment restrictions (precedence constraints and cycle time constraints). Figure 3.4 demonstrates the procedure to balance the risk using the heuristic method

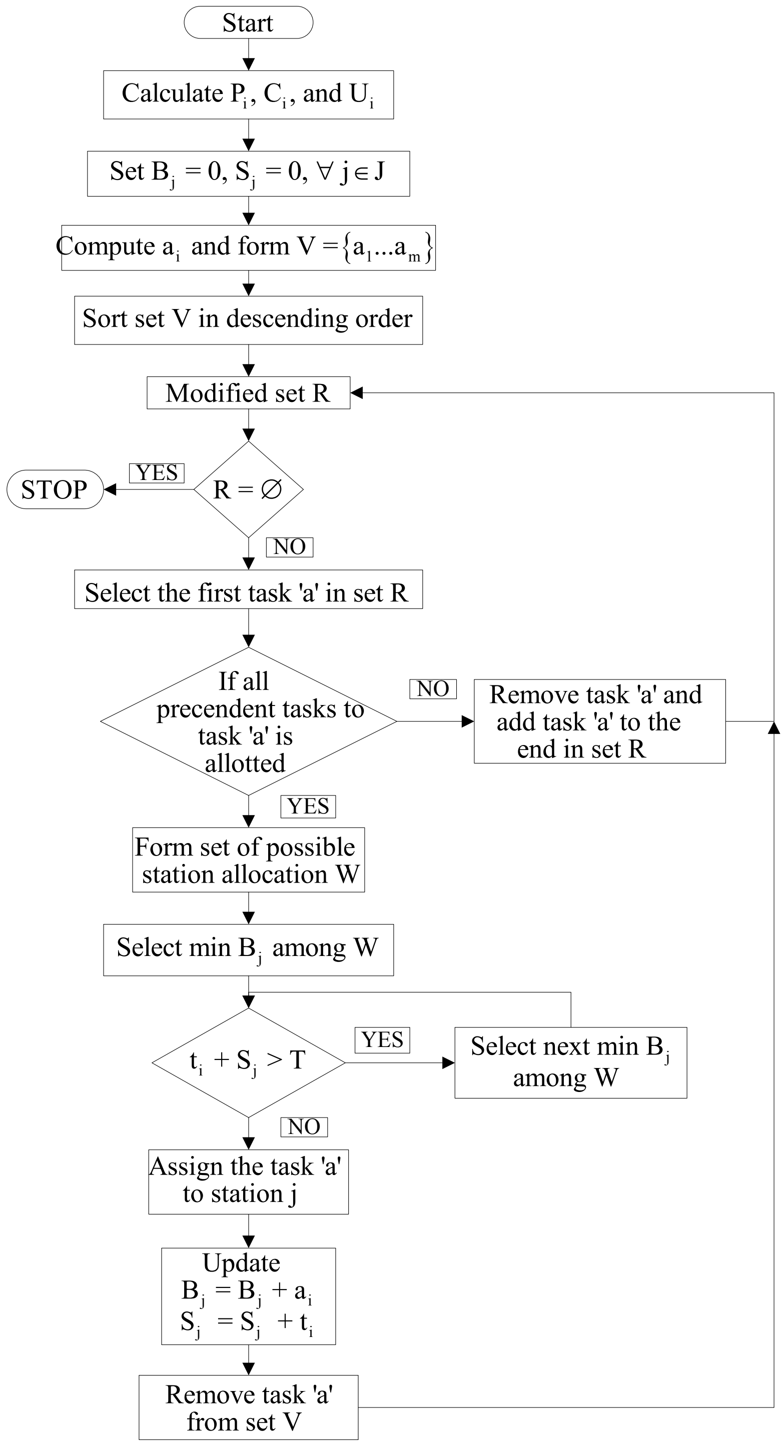


Figure 3.4. Heuristic method flow chart

The heuristic method is applied to the case study discussed in the next section.

3.8 Case Study 1

To balance risk, this section details the case study of single model manual unpaced asynchronous assembly line arranged in a serial manner as shown in Figure 3.5. The assembly line consists of four workstations with no buffers between workstations.

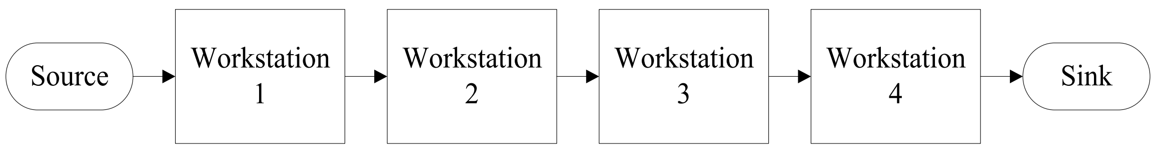


Figure 3.5. Assembly line layout for case study 1

The product is stationed at workstation 1 until all the task operations were completed and then moved to next workstation 2 by means of material handling device such as conveyors, AGV, etc. The given data includes number of tasks, number of workstations, cycle time (minutes), allotted station time (minutes), standard task time (minutes) and binary precedence table.

$I = \{I_1, I_2, \dots \dots, I_{16}\}$, where $m = 16$

$J = \{J_1, J_2, \dots \dots, J_4\}$, where $n = 4$

$T = 27$ minutes

$S_j = \{25, 24.7, 24.3, 24.7\}$

$t_i = \{5.1, 4, 6.9, 9, 5.8, 6, 6.9, 6, 6.8, 5.6, 5, 6.9, 5.8, 6.4, 6.5, 6\}$

Binary precedence table P_{ij}:

$$
P_{ij} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

The production data for 16 tasks is given below and average task time is calculated as follows:

$$
L_{xy} = \begin{pmatrix}
3.3 & 1.5 & 2.7 & 4.6 & 1.6 & 0.7 & 1.6 & 1.4 & 0.1 & 2.6 & 3.1 & 4.4 & 0.6 & 0.9 & 2.4 & 2.9 \\
3.5 & 2.3 & 3.2 & 4.7 & 2.6 & 3.2 & 2 & 2.1 & 4.3 & 3.9 & 4.9 & 4.6 & 4.2 & 2.4 & 3.3 & 4.3 \\
3.8 & 3.3 & 3.5 & 7 & 3.1 & 3.2 & 2 & 2.4 & 5.8 & 3.9 & 5 & 6.5 & 4.8 & 3.1 & 3.9 & 4.6 \\
4.7 & 4 & 3.9 & 8.5 & 3.2 & 3.2 & 3.3 & 2.7 & 6.4 & 4 & 5.7 & 7.6 & 6.1 & 3.9 & 4 & 4.8 \\
5.3 & 4.1 & 4.2 & 8.5 & 3.2 & 3.2 & 3.4 & 2.8 & 6.6 & 4.5 & 5.9 & 8.5 & 6.4 & 4.2 & 4 & 4.8 \\
5.3 & 4.5 & 4.6 & 9.5 & 3.7 & 3.4 & 4 & 3.2 & 6.7 & 4.9 & 6.3 & 8.5 & 6.7 & 4.3 & 4.3 & 4.9 \\
7.5 & 4.8 & 6.3 & 9.6 & 3.8 & 3.6 & 4.6 & 3.5 & 6.8 & 5 & 6.4 & 9.4 & 6.8 & 4.3 & 4.6 & 4.9 \\
7.5 & 5.1 & 6.8 & 10.6 & 3.9 & 4.7 & 4.6 & 4.7 & 7.1 & 6.3 & 6.5 & 9.9 & 7.1 & 4.9 & 4.7 & 5.1 \\
8.6 & 5.3 & 7.2 & 12.1 & 3.9 & 4.7 & 5.4 & 4.8 & 7.5 & 6.4 & 6.5 & 10 & 7.1 & 5.3 & 5 & 5.3 \\
8.6 & 5.7 & 9.5 & 12.3 & 3.9 & 4.7 & 5.4 & 5.2 & 7.7 & 6.6 & 6.5 & 10.3 & 7.1 & 5.3 & 5.6 & 5.3 \\
9.3 & 6 & 10.7 & 12.3 & 4.5 & 5 & 5.5 & 5.4 & 7.9 & 7.8 & 6.5 & 12.1 & 7.1 & 5.3 & 6.5 & 5.5 \\
10.5 & 6.4 & 11.3 & 13.6 & 4.6 & 5.2 & 6.4 & 5.7 & 8.6 & 8 & 6.6 & 12.2 & 7.8 & 6.6 & 7.7 & 5.6 \\
10.6 & 6.4 & 12.3 & 15.3 & 5.3 & 5.4 & 6.7 & 6.5 & 8.9 & 8.9 & 6.6 & 15.2 & 7.9 & 6.9 & 7.7 & 5.6 \\
11.8 & 6.7 & 12.8 & 15.9 & 6.1 & 7.1 & 10.2 & 7.1 & 10.3 & 10.6 & 7.6 & 15.2 & 7.9 & 8.3 & 7.9 & 5.8 \\
11.9 & 6.9 & 13.1 & 17.8 & 6.4 & 7.8 & 10.5 & 7.3 & 11.7 & 10.8 & 7.9 & 15.8 & 8 & 9 & 8.8 & 6.6 \\
13.2 & 7.3 & 13.2 & 19.8 & 6.5 & 9.9 & 11.9 & 7.7 & 15 & 11.8 & 8 & 16.5 & 8.5 & 9.2 & 10.6 & 6.9 \\
13.4 & 7.4 & 14.4 & 20.3 & 7.1 & 10 & 12.2 & 8.4 & 18.1 & 12 & 8.3 & 17.3 & 10.5 & 13.5 & 10.6 & 8.1 \\
13.9 & 7.5 & 16.5 & 24.4 & 8.9 & 11.7 & 12.4 & 10 & 20.6 & 14 & 9 & 23.3 & 10.7 & 16.5 & 14.7 & 13.8 \\
- & 8.5 & 19.7 & 24.6 & - & - & 16.3 & 11 & 21.3 & 15.3 & 9.3 & - & 14.4 & 18.7 & 14.8 & 14.3 \\
- & - & - & 26.8 & - & - & - & - & - & - & 9.6 & - & - & - & - & -
\end{pmatrix}
$$

Average task time is calculated from the historical data and is represented below.

$$A_i = \{8.5, 5.5, 9.3, 13.9, 4.6, 5.4, 6.8, 5.4, 9.5, 7.6, 6.8, 11.5, 7.4, 7, 6.9, 6.3\}$$

Delay index ratio, contribution ratio, criticality ratio, and risk index number are calculated as follows.

Consider task 5,

Number of instances the task exceeds the standard task time (E) = 5

Total number of historical data (l) = 18

$$\text{Delay Index } (D_5) = \frac{E_5}{l_5}$$

$$D_5 = \frac{5}{18} = 0.278$$

Standard task time (t_5) = 5.8

Total cycle time of the product (C_p) = 98.7

$$\text{Contribution Ratio } (C_5) = \frac{t_5}{C_p}$$

$$C_5 = \frac{5.8}{98.7} = 0.059$$

Average task time (A_5) = 4.6

$$\text{Criticality Ratio } (U_5) = 1 - \frac{t_5}{A_5}$$

$$U_5 = 1 - \frac{5.8}{4.6} = -0.26$$

Since the value of U_5 is negative, it is converted to 0.001.

$U_5 = 0.001$

$$RI_5 = D_5 \times C_5 \times U_5 \times 1000$$

$RI = 0.278 \times 0.059 \times 0.001 \times 1000$

$RI = 0.017$

Similarly, for all tasks, three ratios‟ are calculated and tabulated in Table 3.6.

Table 3.6. Ratios‟ of risk index method (case study 1)

Task	D	C	U	RI
T_1	0.778	0.052	0.400	16.076
T_2	0.789	0.041	0.273	8.726
T_3	0.579	0.070	0.258	10.445
T_4	0.750	0.091	0.353	24.109
T_5	0.278	0.059	0.001	0.017
T_6	0.278	0.061	0.001	0.017
T_7	0.316	0.070	0.001	0.023
T_8	0.368	0.061	0.001	0.023
T_9	0.632	0.069	0.284	12.367
T_{10}	0.632	0.057	0.263	9.431
T_{11}	0.850	0.051	0.265	11.399
T_{12}	0.833	0.070	0.400	23.303
T_{13}	0.842	0.059	0.216	10.7
T_{14}	0.421	0.065	0.086	2.341
T_{15}	0.421	0.066	0.058	1.608
T_{16}	0.263	0.061	0.048	0.762

The given assembly line is balanced in terms of cycle time and initial allocation of tasks without violating precedence relationship is shown in Table 3.7. Heuristic method is applied to the case study and the same Table 3.7 shows the task allocation order, values of risk, and station time before reallocation and after reallocation of tasks. Figure 3.6 shows that RI is balanced after reallocation.

Table 3.7. Task order - before and after reallocation (case study 1)

Workstation	J_1	J_2	J_3	J_4
Before Reallocation				
Task Allocation	$\{T_1, T_2, T_3, T_4\}$	$\{T_5, T_6, T_7, T_8\}$	$\{T_9, T_{10}, T_{11}, T_{12}\}$	$\{T_{13}, T_{14}, T_{15}, T_{16}\}$
RI	59.36	0.08	56.5	15.41
Station Time	25	24.7	24.3	24.7

After Reallocation				
Task Allocation	{T_4, T_2, T_{16}, T_8}	{T_{12}, T_3, T_5,T_6}	{T_1, T_{13}, T_{14}, T_{15}}	{T_9, T_{11}, T_{10}, T_7}
RI	33.62	33.782	30.725	33.22
Station Time	25	25.6	23.8	24.3

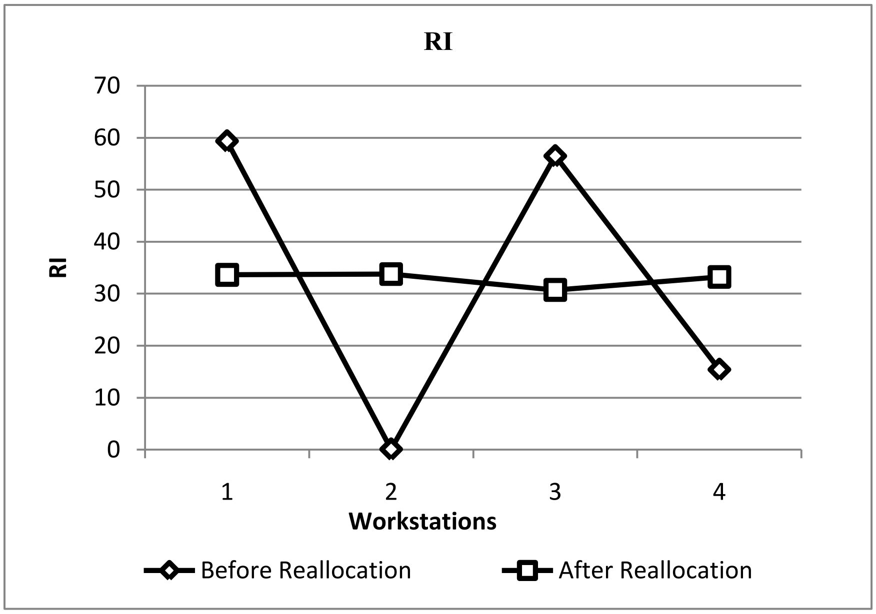


Figure 3.6: RI balance chart (case study 1)

3.8.1 Simulation Method

To validate the results of the heuristic approach for the case study, simulation has been used. Kelton, Sadowski, and Sadowski (2002) defined simulation as "A process of designing and creating a computerized model of a real or proposed system for the purpose of conducting numerical experiments to give us a better understanding of the behavior of that system for a given set of conditions". For our research, simulation software ARENA is used as it can easily handle stochastic input to models. Using ARENA, the real time assembly line is translated into a simulation model. Once the model assumptions are considered, stochastic task times are given as input to the model. The model is run for particular run length with „N_r" replications to observe the throughput. Figure 3.7 shows the procedure for simulation methodology.

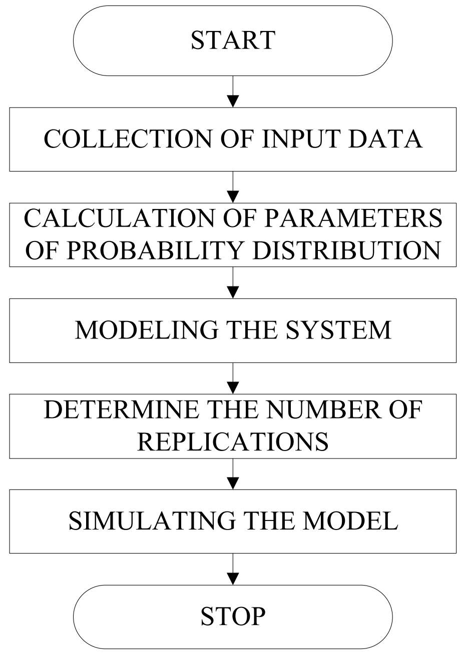


Figure: 3.7. Flow chart for simulation methodology

3.8.1.1 Task Representation

In this section, representation of task time into the model is illustrated. All tasks were assumed to follow continuous distribution function. The continuous function is used to incorporate observed values of a task directly into the model. This directly generates the data between the minimum and maximum value for simulation, which are consistent similar to the original data. In ARENA, the continuous function is represented by pairs of cumulative probabilities ($CumP_1$) and corresponding value (Val_1) as in equation 3.21.

$$\text{CONT}\left(\text{CumP}_1, \text{Val}_1, \ldots \ldots, \text{CumP}_n, \text{Val}_n\right) \qquad (3.21)$$

Continuous distribution function f(x) for a random variable x is represented below in equation 3.22 (Kelton, Sadowski, and Sadowski, 2002).

$$f(x)=\begin{cases} C_1 & \text{if } x = x_1 \\ C_j - C_{j-1} & \text{if } x_{j-1} \leq x < x_j, \text{ for } j=2,3,...,n. \\ 0 & \text{if } x < x_1 \text{ or } x \geq x_n \end{cases} \tag{3.22}$$

"The cumulative distribution function f(x) is piecewise linear with corners defined by f $(x_j) = C_j$ for j = 1, 2… n. Thus, for $j \geq 2$, the returned value will be in the interval (x_{j-1}, x_j) with probability $C_j - C_{j-1}$; given that it is in the interval, it will be distributed uniformly over it (Kelton, Sadowski, and Sadowski, 2002)". The value returned with cumulative probability C_j will be between x_1 and x_n but will be less than or equal to x_j. The value of C_j is between 0 and 1 and increases with j. The x_j increases with j and C_n must be 1. The value of C_1 is carefully specified to avoid mass of probability at x_1. To get truly continuous distribution, the value of C_1 is given as 0.

For a task, cumulative probabilities and the corresponding values are calculated by applying rule of thumb. It is used to determine the class intervals and interval width for calculating the frequency (probability) of the data points in a particular range. Number of intervals (K) is calculated by approximately the square root of the data points (L) and interval width (W) is determined by dividing the range by number of intervals (K). Range is the difference between maximum and minimum value from the data set. Count the number of data points that fall in each interval and calculate the probability of data points that fall in that interval. Cumulative probability of each interval is computed and the corresponding value is the average of all the data points that fall in that interval.

For demonstration, consider task 5 in the above-mentioned case study,

Number of historical data (l) = 18

Number of intervals (K) $=\sqrt{l} = \sqrt{18} = 4.24$

$K \cong 5$

Range (R) = maximum value – minimum value

R = 8.9 - 1.6 = 7.3

$$\text{Interval Width} = \frac{R}{K} = \frac{7.3}{5} = 1.46$$

The cumulative probability and the corresponding value for each interval is calculated and shown in Table 3.8.

Table 3.8. Continuous function parameters calculation for task 5

Intervals	Interval Boundaries	# of Data Points	Initial Probability	Cumulative Probability	Value
1	1.6 - 3.06	2	0.11	0.11	2.1
2	3.06 - 4.52	9	0.50	0.61	3.69
3	4.52 - 5.98	2	0.11	0.72	4.95
4	5.98 - 7.44	4	0.22	0.94	6.53
5	7.44 - 8.9	1	0.06	1.00	8.9

The continuous function for task 5 is,

$\text{CONT}\left(0, 1.6, 0.11, 2.1, 0.61, 3.69, 0.72, 4.95, 0.94, 6.53, 1.0, 8.9\right)$. The first interval (0, 1.6) is added along with above five intervals. This interval is set with probability 0 along with corresponding minimum value to avoid mass of probability at x_1.

Similarly, continuous function for all the 16 tasks in the case study is given below in Table 3.9.

Table 3.9. Continuous function for 16 tasks (case study 1)

Task	Continuous Function
T_1	CONT(0, 3.3, 0.33, 4.32, 0.44, 7.5, 0.61, 8.83, 0.72, 10.55, 1, 12.84)
T_2	CONT(0, 1.5, 0.11, 1.9, 0.26, 3.8, 0.53, 5.08, 0.79, 6.48, 1, 7.68)
T_3	CONT(0, 2.7, 0.32, 3.68, 0.53, 7.45, 0.74, 11.78, 0.89, 13.57, 1, 18.1)
T_4	CONT(0, 4.6, 0.25, 6.66, 0.55, 11.07, 0.75, 15.65, 0.85, 20.05, 1, 25.27)
T_5	CONT(0, 1.6, 0.11, 2.1, 0.61, 3.69, 0.72, 4.95, 0.94, 6.53, 1, 8.9)
T_6	CONT(0, 0.7, 0.06, 0.7, 0.61, 3.89, 0.78, 5.9, 0.83, 7.8, 1, 10.53)

T_7	CONT(0, 1.6, 0.32, 2.72, 0.68, 5.51, 0.74, 10.2, 0.95, 11.75, 1, 16.3)
T_8	CONT(0, 1.4, 0.32, 2.43, 0.53, 4.55, 0.74, 6.18, 0.89, 7.8, 1, 10.5)
T_9	CONT(0, 0.1, 0.11, 2.2, 0.58, 6.94, 0.79, 9.88, 0.84, 15, 1, 20)
T_{10}	CONT(0, 2.6, 0.37, 4.11, 0.53, 6.43, 0.68, 8.23, 0.89, 11.3, 1, 14.65)
T_{11}	CONT(0, 3.1, 0.05, 3.1, 0.2, 5.2, 0.65, 6.42, 0.85, 7.95, 1, 9.3)
T_{12}	CONT(0, 4.4, 0.22, 5.8, 0.56, 9.43, 0.78, 13.68, 0.94, 16.53, 1, 23.3)
T_{13}	CONT(0, 0.6, 0.05, 0.6, 0.21, 5.03, 0.84, 7.37, 0.95, 10.6, 1, 14.4)
T_{14}	CONT(0, 0.9, 0.37, 3.3, 0.68, 5.72, 0.84, 8.83, 0.89, 13.5, 1, 17.6)
T_{15}	CONT(0, 2.4, 0.42, 3.9, 0.58, 5.7, 0.79, 8.03, 0.89, 10.6, 1, 14.75)
T_{16}	CONT(0, 2.9, 0.42, 4.54, 0.84, 5.83, 0.89, 8.1, 1, 14.05)

After calculating the continuous function for all tasks, the assembly line is modeled in ARENA. Once the simulation model is developed, number of replications is determined. The next section explains the calculation of number of replications.

3.8.1.2 Number of Replications

Since the input to the model is specified by probability distribution, there will be randomness in the output. Therefore, single run may not be sufficient. So, the model is run for several independent runs called replications. For 95% confidence interval, approximate number of replications (N_r) is given by:

$$N_r = t^2_{\frac{\alpha}{2}, n-1} \frac{\sigma^2}{h^2} \qquad (3.23)$$

The warm-up period is determined for monitoring the waiting and idle time of the assembly line. The model is simulated for particular run length with „N_r" replications and the throughput is determined by calculating the average value of throughput for all replications.

3.8.2 Results Generated from Simulation

The model is simulated for 25 replications and each replication for 25000 minutes. Initial 5000 minutes is assumed as warm-up period and data obtained during this period is deleted. The summary of results, which includes value added time, waiting time, idle time, and non-value added time for all workstations in the assembly line, is shown in Table 3.10.

Table 3.10. Results (case study 1)

Workstation	J_1	J_2	J_3	J_4	Total	T_H
Before Reallocation						
Value Added Time	18385.03	10468.18	17915.95	13006.69		576
Waiting Time	1614.46	4862.1	481.49	0	6958.05	
Idle Time	0	4669.72	1602.56	6993.31	13265.59	
Non-Value Added time	1614.46	9531.82	2084.05	6993.31	20223.64	
After Reallocation						
Value Added Time	16341.8	15881.5	15468.58	15757.74		616
Waiting Time	3676.94	2648.41	1920.78	0	8246.13	
Idle Time	0	1470.09	2610.64	4242.26	8322.99	
Non-Value Added time	3676.94	4118.5	4531.42	4242.26	16569.12	

From Table 3.10, it can be observed that throughput is improved by 7% when RI is balanced. The increase in throughput is due to the reduction in non-value added time by 18.1%. Figure 3.8 compares the performance of before reallocation and after reallocation through value added time and non-value added time for all workstations, and Figure 3.9 shows the improvement in throughput.

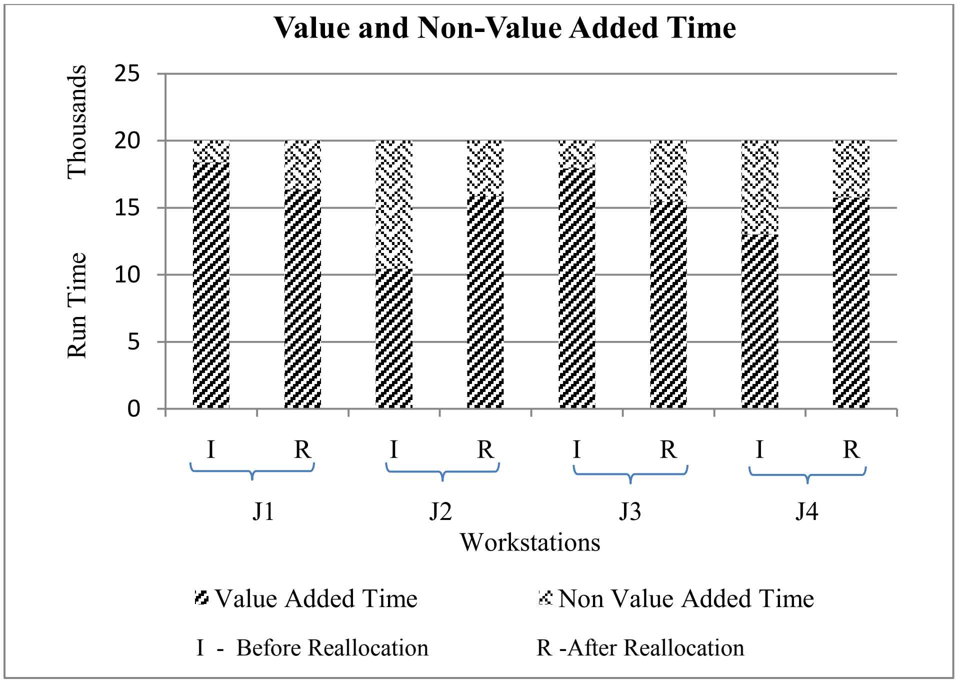


Figure: 3.8. Comparison of value & non-value added time (case study 1)

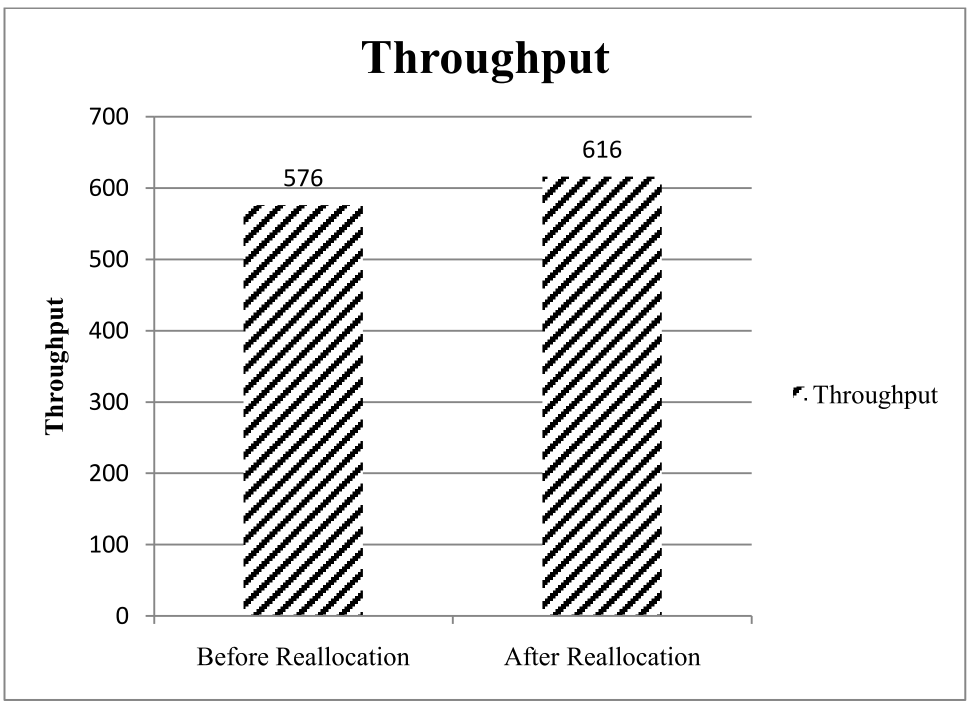


Figure: 3.9. Throughput (case study 1)

3.9 Case Study 2

This section details the case study 2 consists of single model manual assembly line arranged in a serial manner as shown in Figure 3.10.

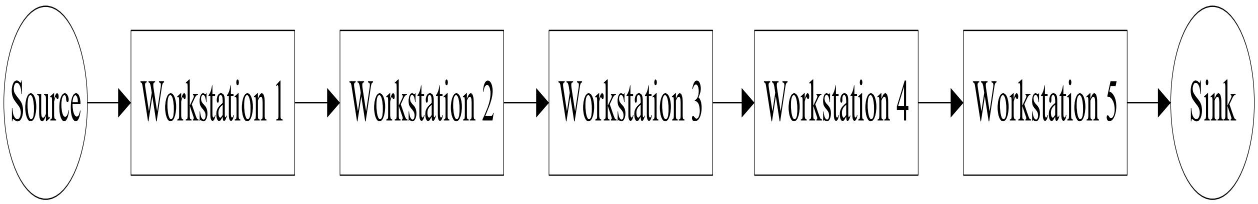


Figure: 3.10. Assembly line layout (case study 2)

The given data includes number of tasks, number of workstations, cycle time (minutes), allotted station time (minutes), standard task time (minutes), binary precedence table and historical data of task times for 20 products.

$I = \{I_1, I_2, \ldots \ldots, I_{40}\}$, where m = 40

$J = \{J_1, J_2, \ldots \ldots, J_5\}$, where n = 5

T = 53.2 minutes

$S_j = \{51.8, 52.3, 52.7, 52.4, 51.7\}$

$$t_i = \begin{Bmatrix} 3.1, 3.5, 5, 5.1, 6.9, 8.2, 9, 11, 3.2, 3.9, 4, 5.8, 6, 9, 10, 10.8, 3.1, 4, 4.1, 5.6, 6.8, \\ 6.9, 10.8, 11, 2.5, 3.1, 3.9, 6.9, 6, 7.7, 10.8, 11.5, 3.2, 4, 4.6, 5.8, 6, 6.4, 10.2, 11.5 \end{Bmatrix}$$

Average task time is calculated from the historical data and is represented below.

$$A_i = \begin{Bmatrix} 5.1, 5.2, 6.8, 8.5, 9.3, 12.3, 13.9, 14.6, 2.9, 3.3, 3.4, 4.6, 5.4, 7.2, 8.7, 7.6, 6.2, 5.5, 7, 7.6, \\ 9.5, 11.5, 15.5, 17.5, 1.8, 1.9, 3.4, 6.8, 5.4, 7.1, 5.3, 8.5, 4, 4.6, 5.2, 7.4, 6.3, 7, 9.7, 12.9 \end{Bmatrix}$$

Delay index ratio, contribution ratio, criticality ratio, and risk index number are calculated and tabulated in Table 3.11.

Table 3.11. Ratios" of risk index method (case study 2)

Task	D	C	U	RI	Task	D	C	U	RI
T_1	0.800	0.012	0.392	3.728	**T_{21}**	0.632	0.026	0.284	4.678
T_2	0.900	0.013	0.327	3.947	**T_{22}**	0.833	0.026	0.400	8.816
T_3	0.850	0.019	0.265	4.312	**T_{23}**	0.833	0.041	0.303	10.460
T_4	0.778	0.020	0.400	6.082	**T_{24}**	0.850	0.042	0.371	13.311
T_5	0.579	0.026	0.258	3.951	**T_{25}**	0.412	0.010	0.001	0.004
T_6	0.778	0.031	0.333	8.148	**T_{26}**	0.050	0.012	0.001	0.001
T_7	0.750	0.034	0.353	9.120	**T_{27}**	0.353	0.015	0.001	0.005
T_8	0.800	0.042	0.247	8.317	**T_{28}**	0.316	0.026	0.001	0.008
T_9	0.263	0.012	0.001	0.003	**T_{29}**	0.368	0.023	0.001	0.008
T_{10}	0.222	0.015	0.001	0.003	**T_{30}**	0.250	0.030	0.001	0.007
T_{11}	0.316	0.015	0.001	0.005	**T_{31}**	0.111	0.041	0.001	0.005
T_{12}	0.278	0.022	0.001	0.006	**T_{32}**	0.111	0.044	0.001	0.005
T_{13}	0.278	0.023	0.001	0.006	**T_{33}**	0.684	0.012	0.200	1.678
T_{14}	0.222	0.034	0.001	0.008	**T_{34}**	0.579	0.015	0.130	1.158
T_{15}	0.150	0.038	0.001	0.006	**T_{35}**	0.556	0.018	0.115	1.130
T_{16}	0.278	0.041	0.001	0.011	**T_{36}**	0.842	0.022	0.216	4.048
T_{17}	0.842	0.012	0.500	5.003	**T_{37}**	0.263	0.023	0.048	0.288
T_{18}	0.789	0.015	0.273	3.301	**T_{38}**	0.421	0.025	0.086	0.885
T_{19}	0.895	0.016	0.414	5.825	**T_{39}**	0.526	0.039	0.001	0.021
T_{20}	0.632	0.021	0.263	3.567	**T_{40}**	0.500	0.044	0.109	2.392

The given assembly line is balanced in terms of cycle time and initial allocation of tasks without violating precedence relationship is shown in Table 3.12. Heuristic method is applied to the case study and the same Table 3.12 shows the task allocation order, values of risk, and station time before reallocation and after reallocation of tasks. Figure 3.11 shows that RI is balanced after reallocation.

Table 3.12. Task order - before and after reallocation (case study 2)

Workstation	J_1	J_2	J_3	J_4	J_5
Before Reallocation					
Task Allocation	$\{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8\}$	$\{T_9, T_{10}, T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}\}$	$\{T_{17}, T_{18}, T_{19}, T_{20}, T_{21}, T_{22}, T_{23}, T_{24}\}$	$\{T_{25}, T_{26}, T_{27}, T_{28}, T_{29}, T_{30}, T_{31}, T_{32}\}$	$\{T_{33}, T_{34}, T_{35}, T_{36}, T_{37}, T_{38}, T_{39}, T_{40}\}$
RI	47.605	0.049	54.962	0.044	11.60
Allotted Station Time	51.8	52.3	52.7	52.4	51.7
After Reallocation					
Task Allocation	$\{T_{24}, T_{21}, T_{20}, T_{34}, T_{29}, T_{28}, T_{14}, T_9\}$	$\{T_{23}, T_{17}, T_5, T_{40}, T_{35}, T_{30}, T_{27}, T_{11}\}$	$\{T_7, T_{19}, T_{36}, T_{18}, T_{37}, T_{39}, T_{16}, T_{25}\}$	$\{T_{22}, T_4, T_3, T_1, T_{13}, T_{12}, T_{15}, T_{31}\}$	$\{T_8, T_6, T_2, T_{33}, T_{38}, T_{32}, T_{10}, T_{26}\}$
RI	22.742	22.954	22.618	22.960	22.985
Station Time	52.5	52.5	52.4	52.7	50.8

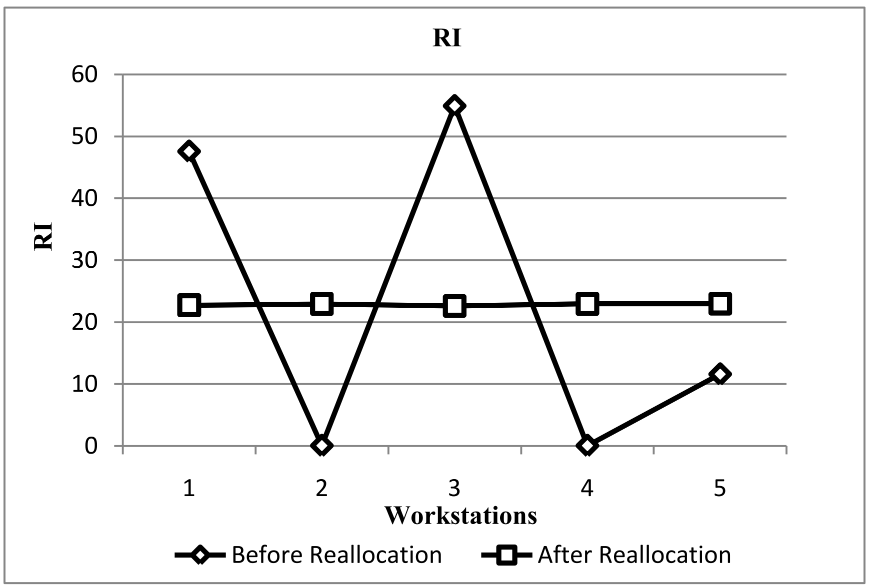


Figure 3.11. RI balance chart (case study 2)

To validate the results of the heuristic approach for the case study, simulation has been used. All the tasks were assumed to follow continuous distribution. The same procedure discussed in section 3.9.1 was used to calculate continuous distribution function. Table 3.13 shows the continuous distribution function for all the 40 tasks

Table 3.13. Continuous function for 40 tasks (case study 2)

Task	Continuous Function
T_1	CONT(0, 1.8, 0.2, 2.63, 0.5, 4.33, 0.7, 5.48, 0.95, 6.76, 1, 9.3)
T_2	CONT(0, 2.5, 0.1, 2.9, 0.4, 3.98, 0.65, 5.18, 0.85, 6.2, 1, 7.57)
T_3	CONT(0, 3.1, 0.05, 3.1, 0.2, 5.2, 0.65, 6.42, 0.85, 7.95, 1, 9.3)
T_4	CONT(0, 3.3, 0.33, 4.32, 0.44, 7.5, 0.61, 8.83, 0.72, 10.55, 1, 12.84)
T_5	CONT(0, 2.7, 0.32, 3.68, 0.53, 7.45, 0.74, 11.78, 0.89, 13.57, 1, 18.1)
T_6	CONT(0, 2.5, 0.16, 4.23, 0.47, 9.27, 0.79, 13.75, 0.95, 18.57, 1.0, 27.3)
T_7	CONT(0, 4.6, 0.25, 6.66, 0.55, 11.07, 0.75, 15.65, 0.85, 20.05, 1, 25.27)
T_8	CONT(0, 7.9, 0.2, 9.35, 0.5, 12.92, 0.75, 15.42, 0.9, 18.67, 1.0, 22.3)
T_9	CONT(0, 0.9, 0.11, 1.3, 0.58, 2.5, 0.89, 3.3, 1, 5.55)
T_{10}	CONT(0, 0.9, 0.39, 1.5, 0.78, 3.26, 0.89, 5.25, 0.94, 6.4, 1, 9.3)
T_{11}	CONT(0, 1.6, 0.32, 1.87, 0.42, 2.75, 0.68, 3.8, 0.89, 4.35, 1, 5.4)
T_{12}	CONT(0, 1.6, 0.11, 2.1, 0.61, 3.69, 0.72, 4.95, 0.94, 6.53, 1, 8.9)
T_{13}	CONT(0, 0.7, 0.06, 0.7, 0.61, 3.89, 0.78, 5.9, 0.83, 7.8, 1, 10.53)
T_{14}	CONT(0, 2, 0.33, 2.93, 0.78, 7.06, 0.89, 11.25, 0.94, 13.2, 1.0, 20.1)
T_{15}	CONT(0, 6, 0.05, 6, 0.25, 7.65, 0.8, 8.58, 0.85, 9.8, 1.0, 10.87)
T_{16}	CONT(0, 1.1, 0.33, 2.55, 0.5, 5.37, 0.56, 8.9, 0.83, 10.7, 1.0, 14.7)
T_{17}	CONT(0, 1.1, 0.32, 2.83, 0.58, 4.88, 0.84, 8.34, 0.95, 9.7, 1, 14.8)
T_{18}	CONT(0, 1.5, 0.11, 1.9, 0.26, 3.8, 0.53, 5.08, 0.79, 6.48, 1, 7.68)
T_{19}	CONT(0, 2, 0.21, 3.38, 0.58, 6, 0.95, 8.83, 1, 15.1)
T_{20}	CONT(0, 2.6, 0.37, 4.11, 0.53, 6.43, 0.68, 8.23, 0.89, 11.3, 1, 14.65)
T_{21}	CONT(0, 0.1, 0.11, 2.2, 0.58, 6.94, 0.79, 9.88, 0.84, 15, 1, 20)
T_{22}	CONT(0, 4.4, 0.22, 5.8, 0.56, 9.43, 0.78, 13.68, 0.94, 16.53, 1, 23.3)
T_{23}	CONT(0, 6.9, 0.26, 9.22, 0.63, 13.2, 0.79, 17.77, 0.89, 23.7, 1.0, 27.6)
T_{24}	CONT(0, 7, 0.15, 8.97, 0.4, 13.56, 0.8, 18.63, 0.85, 21.8, 1.0, 28.33)
T_{25}	CONT(0, 0.3, 0.53, 0.62, 0.59, 2, 0.82, 2.65, 0.94, 4, 1, 5)
T_{26}	CONT(0, 1, 0.45, 1.41, 0.85, 1.94, 0.95, 2.45, 1, 4.2)
T_{27}	CONT(0, 0.8, 0.24, 1.4, 0.41, 2.73, 0.82, 3.83, 0.88, 4.6, 1, 6.4)
T_{28}	CONT(0, 1.6, 0.32, 2.72, 0.68, 5.51, 0.74, 10.2, 0.95, 11.75, 1, 16.3)
T_{29}	CONT(0, 1.4, 0.32, 2.43, 0.53, 4.55, 0.74, 6.18, 0.89, 7.8, 1, 10.5)
T_{30}	CONT(0, 2.7, 0.25, 3.8, 0.65, 6.3, 0.8, 7.8, 0.9, 11.1, 1.0, 13.8)
T_{31}	CONT(0, 0.5, 0.17, 1.8, 0.67, 3.61, 0.72, 6, 1.0, 10.44)
T_{32}	CONT(0, 5.5, 0.56, 6.96, 0.78, 8.4, 0.89, 10.25, 0.94, 12.4, 1.0, 16.9)
T_{33}	CONT(0, 1.8, 0.21, 2.28, 0.53, 3.4, 0.63, 4.45, 0.89, 5.1, 1, 6.5)
T_{34}	CONT(0, 1.3, 0.11, 1.35, 0.32, 3.18, 0.53, 4.33, 0.74, 5.28, 1, 6.56)
T_{35}	CONT(0, 2.8, 0.28, 3.54, 0.67, 4.91, 0.89, 6.18, 0.94, 7.7, 1, 9.9)
T_{36}	CONT(0, 0.6, 0.05, 0.6, 0.21, 5.03, 0.84, 7.37, 0.95, 10.6, 1, 14.4)
T_{37}	CONT(0, 2.9, 0.42, 4.54, 0.84, 5.83, 0.89, 8.1, 1, 14.05)
T_{38}	CONT(0, 0.9, 0.37, 3.3, 0.68, 5.72, 0.84, 8.83, 0.89, 13.5, 1, 17.6)
T_{39}	CONT(0, 3.9, 0.21, 4.48, 0.42, 7.75, 0.63, 10.53, 0.84, 12.13, 1.0, 15.13)
T_{40}	CONT(0, 7.3, 0.28, 8.42, 0.56, 11.28, 0.83, 14.44, 0.89, 17.2, 1.0, 21.95)

After calculating the continuous function for all tasks, the assembly line is modeled in ARENA. The model is simulated for 30 replications and each replication for 25000 minutes.

Initial 5000 minutes is assumed as warm-up period and data obtained during this period is deleted. The summary of results, which includes value added time, waiting time, idle time, and non-value added time for all workstations in the assembly line, is shown in Table 3.14.

Table 3.14. Results (case study 2)

Workstation	J_1	J_2	J_3	J_4	J_5	Total	T_H
Before Reallocation							
Value Added Time	18734.82	10272.08	19441.59	9852.93	14140.9		285
Waiting Time	1265.18	6028.56	2.73	272.46	0	7568.93	
Idle Time	0	3699.36	555.68	9874.61	5859.1	19988.75	
Non-Value Added Time	1265.18	9727.92	558.41	10147.07	5859.1	22693.72	
After Reallocation							
Value Added Time	16796.77	18128.46	16451.64	15743.2	15985.04		326
Waiting Time	3203.23	1070.42	982.19	946.83	0	6202.67	
Idle Time	0	801.12	2566.17	3309.97	4014.96	10692.22	
Non-Value Added Time	3203.23	1871.54	3548.36	4256.8	4014.96	16894.89	

From Table 3.14, it can be observed that throughput is improved by 12.57% when RI is balanced. The increase in throughput is due to the reduction in non-value added time by 25.5%. Figure 3.12 compares the performance of before reallocation and after reallocation through value added time and non-value added time for all workstations, and Figure 3.13 shows the improvement in throughput.

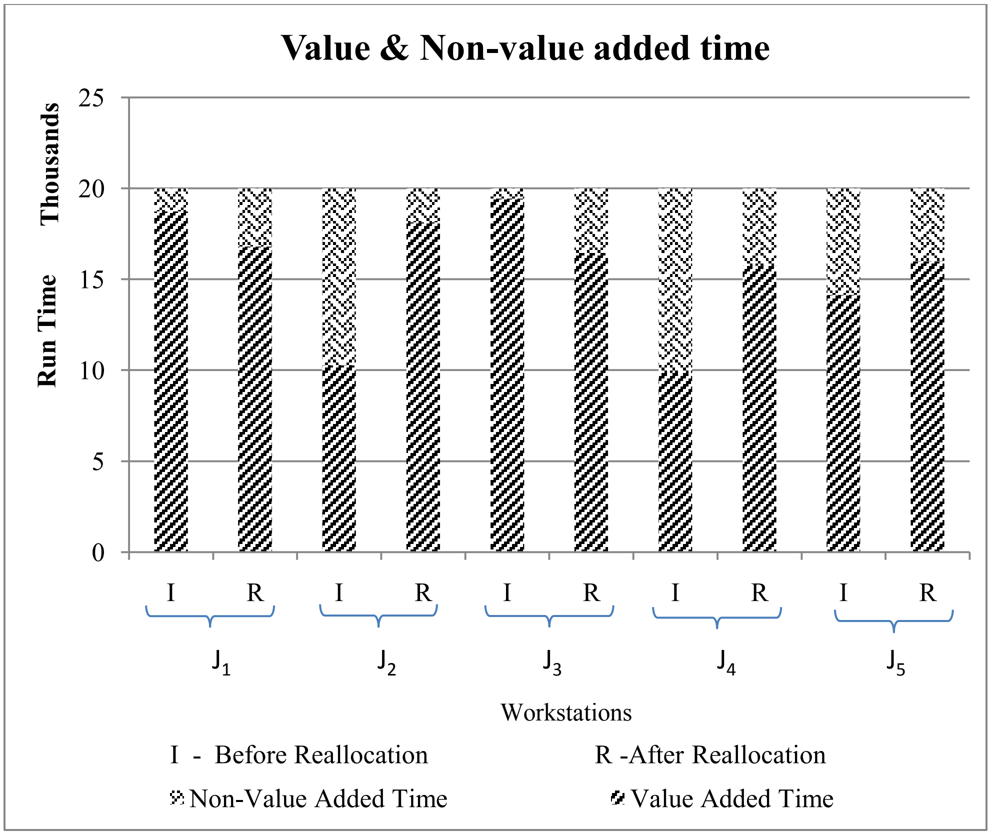


Figure: 3.12. Comparison of value & non-value added time (case study 2)

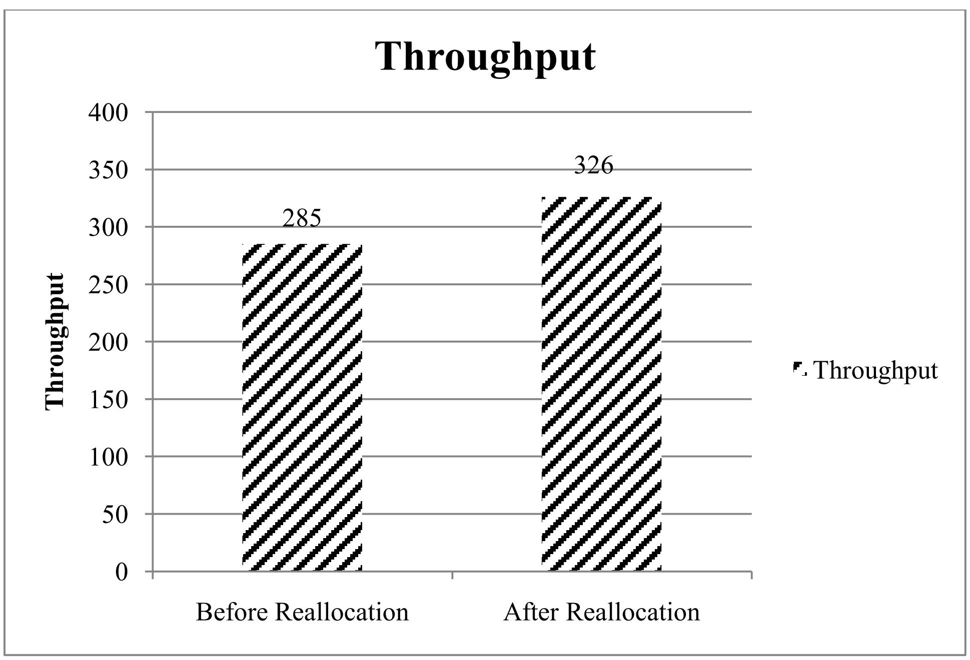


Figure: 3.13. Throughput (case study 2)

3.10 Conclusion

In this chapter, risk oriented assembly line balancing is established. The proposed risk index method captures risk in terms of RI. Risk based line balancing is achieved by balancing the RI between workstation. Hence, to balance the risk, two method were developed; 1) Mathematical formulation method and 2) Heuristic method. The heuristic method is used to reassign the tasks among workstation by minimizing the maximum difference in RI between workstations. The results proved that there is an increase in throughput when the line is balanced based on risk.

CHAPTER 4

COST BASED RISK ANALYSIS

4.1 Introduction

Chapter 3 details the development of risk index method and risk balancing procedures for an assembly line. In the risk index method, three components that contribute to risk are captured in terms of risk index. A heuristic method is proposed to balance the risk index between workstations in an assembly line. In this chapter, development of a cost based risk analysis method is presented. Risk is captured in terms of cost due to delay in product completion. The following sections describe the cost based risk measure, mathematical model, and heuristic method required for balancing the risk along the assembly line.

4.2 Notations

m Total number of tasks

n Total number of workstations

I Set of tasks, where $I = \{I_1, I_2, \ldots \ldots, I_m\}$

J Set of workstations, where $J = \{J_1, J_2, \ldots \ldots, J_n\}$

p_i Standard task time for each task i

a_i Observed or actual task time for each task i

t_i time exceeded from standard task time, $t_i = a_i - p_i$

x_{ij} $\left\{\begin{matrix} 1 & \text{if task i is assigned to station j,} \\ 0 & \text{otherwise,} \end{matrix}\right\} \forall\, i \in I \text{ and } j \in J$

T Cycle time or fixed move rate at all workstations

S_j Total allotted station time, $S_j = \sum_{i=1}^{m} (p_i \times x_{ij})$

P_{ij} Binary precedence table $\left\{\begin{array}{ll} 1, & \text{task i is precedent to task j} \\ 0, & \text{no precedence} \end{array}\right\}$,

$$P_{ij} = \begin{pmatrix} P_{11} & \cdots & P_{1m} \\ \vdots & \ddots & \vdots \\ P_{m1} & \cdots & P_{mm} \end{pmatrix} \forall\ i, j \in I$$

NV_j Non value added time of station j

N Total number of replications or iterations

C_F Fixed cost associated with task exceeding standard task time

F_z (t) Cost function due to delay for the interval „z"

R_i Risk of a task i, $\forall\ i \in I$

W_j Risk of a workstation j, $\forall\ j \in J$

4.3 Risk

In most assembly line balancing approaches, heuristics and algorithms developed assume deterministic processing or task times. However, in the manufacture of products such as aircraft, there is high variability in task times. Variability exists due to large amount of human activities involved in processing the task. Due to the presence of variability in task time, the task tends to exceed the standard task time. If a series of tasks in a workstation exceeds the allotted time, then there is a risk that the product may exceed the cycle time of the workstation. This leads to a potential risk in product delivery lead time.

In this study, the influence of variability in task time completion is evaluated using the risk cost factor method. Modarres (2006) termed risk as "a measure of the potential loss occurring due to natural or human activities" and it can be represented as follows:

$$\text{RISK} = \sum_i u_i c_i \tag{4.1}$$

where, u_i is the probability of event i and c_i denotes the associated consequence. For assembly line balancing, risk is defined as the expected cost when the task fails to complete within the standard task time. The probability of event is defined as the probability that the task exceeds the standard task time and the associated consequence can be viewed as an increase in cost due to delay in completion.

4.3.1 Risk Estimation Methodology

Traditionally, standard task time is estimated from the historical data or time study. Due to variability, the estimated standard task time may or may not be the same as the observed task time, which is obtained from the task time distribution. To analyze the variability, real time data from a local aircraft manufacturing company is collected. The data obtained is analyzed and the following observations were made: a) there is high variability in task times; b) number of data points collected is few due to low volume production; c) the data points are right skewed from the standard task time; and 4) a large variety of tasks is involved.

Parametric task time distribution cannot be used as the number of data points obtained is low. Single distribution function cannot be fitted to all tasks, as there is a large variety of tasks is involved at each station. In such cases, continuous distribution function of each task is determined and the observed task times obtained from the continuous function are used for the estimation of risk. Since there is more than one task in a workstation, cumulative distribution function of combined task time is to be estimated.

Risk can be estimated either by mathematical formulation method or by simulation method. Krishnan, and Jithavech (2009) calculated risk by both methods and showed that when the system needs more random inputs, mathematical method becomes tedious as the time taken to derive the cumulative distribution function can be very long. Simulation is the preferred method when dealing with complex systems. For accurate estimation of expected risk, the task is simulated for „N" replications using simulation software ARENA.

For each replication, risk cost factor analyses the observed task time, which is obtained from the continuous function with that of the standard task time. Due to variability in task times, these observed task times may exceed the standard task time, which in turn leads to a penalty cost due to delay in task completion. The cost due to delay is given in terms of a fixed cost and a variable cost function. It can be assumed that the cost function is different as the time increases. If the cost functions are assumed to be different for various ranges of delays, the generalized form of the cost function can be described as follows:

Interval 1: t ranges between $0 < t <= t_1$, $F_1(t) = t \times f_1(t)$

Interval 2: t ranges between $t_1 < t <= t_2$, $F_2(t) = [t_1 \times f_1(t)] + [(t - t_1) \times f_2(t)]$

……

Interval z: t ranges between $t_z < t < \infty$, $F_z(t) = [t_1 \times f_1(t)] + [(t_2 - t_1) \times f_2(t)] + \ldots[(t - t_z) \times f_z(t)]$

The risk value is an increase in the expected cost as the result of variability in task times. Risk of a task „R_i" is calculated as follows:

$$y_{ij} \quad \left\{ \begin{matrix} 1, & a_i > p_i \\ 0, & \text{Otherwise} \end{matrix} \right\} \ \forall\ i \in I \tag{4.2}$$

$$R_i = \frac{1}{N}\left[\sum_{j=1}^{N} y_{ij} \times \left(C_{Fij} + F_{zij}(t)\right)\right], \forall\ i \in I \tag{4.3}$$

Risk of a workstation „W_j" is calculated as follows:

$$W_j = \sum_{i=1}^{m} (x_{ij} \times R_i) \quad \forall\ j \in J \tag{4.4}$$

The risk value obtained gives a probable estimate of penalty cost due to delay in task completion. Figure 4.1 illustrates simple step-by-step procedure of risk estimation of an assembly line using simulation method.

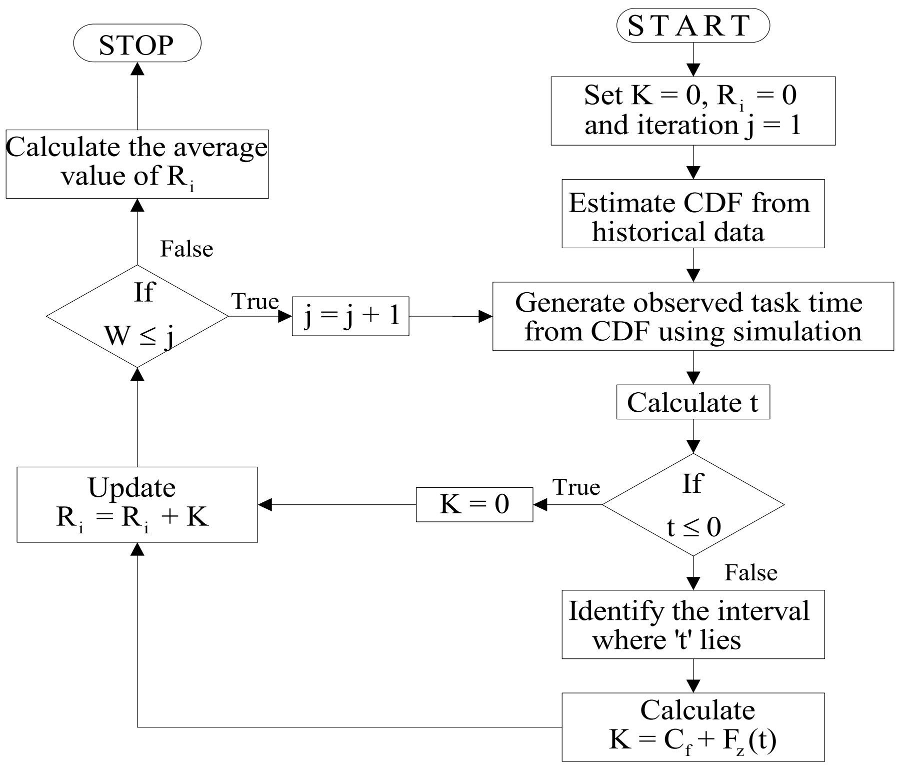


Figure 4.1. Risk estimation flow chart

4.4 Risk Balancing Methodology

In this section, the method used to balance risk is detailed. Risk can be balanced by two methods: a) mathematical formulation method, and b) heuristic method. As mentioned earlier, risk balancing is distributing the risk among workstations by reassigning the tasks to workstations without violating the task assignment restrictions. Both the methods aim in minimizing the maximum difference in risk for all workstations.

4.4.1 Mathematical Formulation Method

The mathematical model for reallocation of tasks among workstation based on risk is shown below.

$$\text{Min Max}\left[\frac{1}{N}\left\{\sum_{i=1}^{m}\sum_{j=1}^{N} x_{ija} \times y_{ija}\left(C_{Fija} + F_{zija}(t)\right)\right\} - \frac{1}{N}\left\{\sum_{i=1}^{m}\sum_{j=1}^{N} x_{ijb} \times y_{ijb}\left(C_{Fijb} + F_{zijb}(t)\right)\right\}\right]$$
$$\forall\ a, b \in J, \text{ and } a \neq b \tag{4.5}$$

<u>Subject to:</u>

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\ i \in I \tag{4.6}$$

$$\sum_{i=1}^{m} (t_i \times x_{ij}) \leq T \quad \forall\ j \in J \tag{4.7}$$

$$P_{ij} + P_{ji} \leq 1 \tag{4.8}$$

$$P_{ij} = 0 \quad \forall\ i = j \tag{4.9}$$

$$x_{ij} = 0, 1 \tag{4.10}$$

Objective function (4.5) represents minimizing the maximum difference in risk between workstations. Constraint (4.6) ensures that every task is allotted to a workstation and also each task is allotted only once to a workstation. Constraint (4.7) guarantees that station time of all workstations do not exceed the cycle time. Constraint (4.8) represents precedence constraints. Constraint (4.9) ensures the task cannot precede the same task. Constraint (4.10) represents x_{ij} can assume values either 0 or 1.

The model shown above is an ALB problem and it is known that ALB problems are NP-hard (Baybars, 1986). This is because, reduction of ALB leads to a partition problem and partition problems are proved to be NP-hard (Karp, 1972). However in this case, the formulation is even harder to solve and hence, the problem becomes impossible for optimization due to large number of work elements. In such cases, heuristic method is the best method to handle line balancing problems.

4.4.2 Heuristic Method

The heuristic method is developed to balance the risk (minimizes the maximum difference in RI) between workstations without violating assignment restrictions (precedence constraints and cycle time constraints). Figure 4.2 demonstrates the procedure to balance the risk using the heuristic method

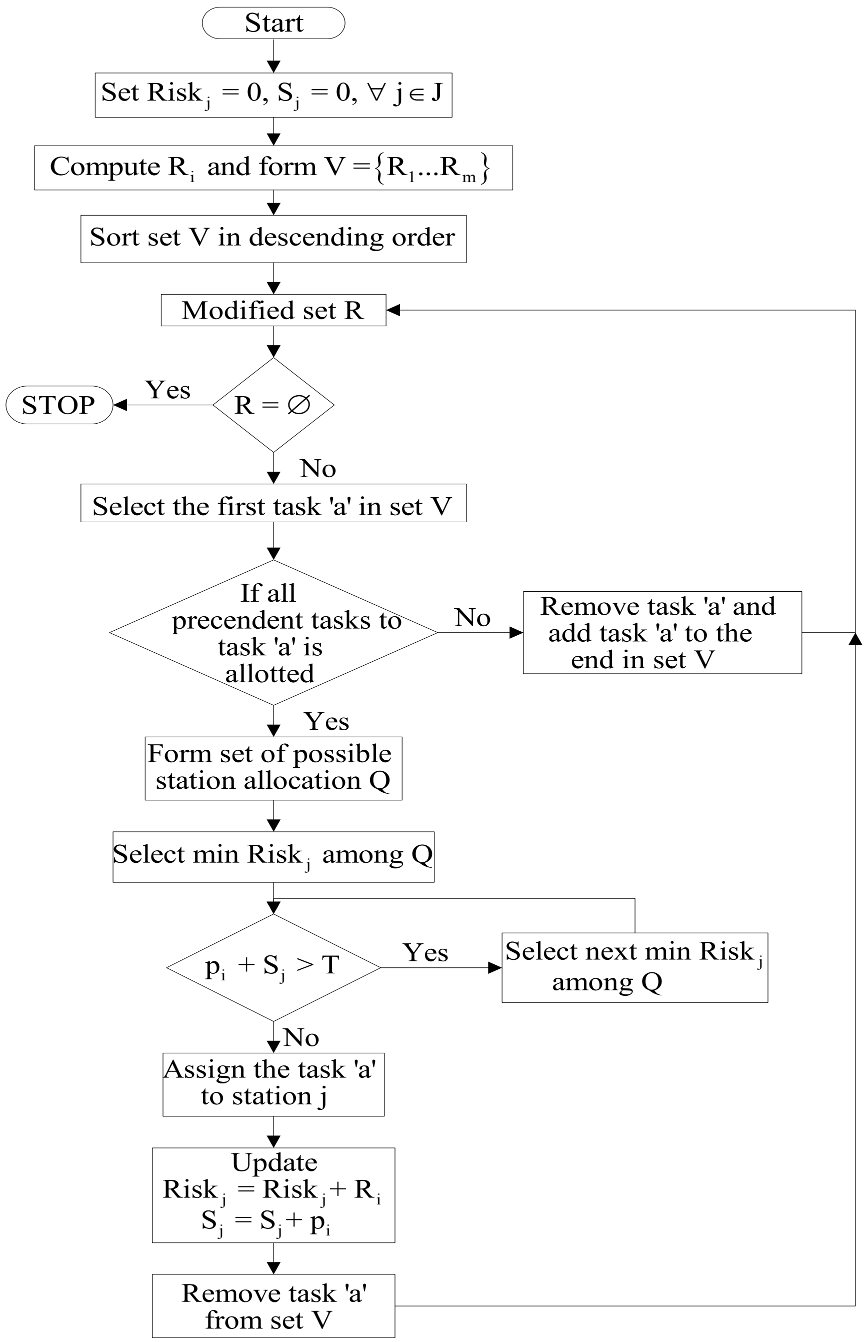


Figure 4.2. Heuristic method flow chart

4.5 Case Study 1

The data for the case study 1 is the same as the one used in chapter 3. The data includes number of tasks, number of workstations, cycle time (minutes), allotted station time (minutes), standard task time (minutes), binary precedence table and production data for 16 tasks are the same listed in the previous chapter. In addition to the data in chapter 3, Table 4.1 shows the summary of additional case study settings which includes, cost function for different intervals of „t"; standard time, and fixed cost. In this case, three intervals of „t" and the associated cost function is considered for all tasks.

Table 4.1. Summary of cost function of 16 tasks (case study 1)

Task	Standard Task Time	Fixed Cost	Intervals	Cost Function
T_1	5.1	100	0 < t <= 2.5	$20t$
			2.5 < t <= 5.5	$50+[30(t-2.5)]$
			5.5 < t < ∞	$140+[15(t-5.5)^2]$
T_2	4	80	0 < t <= 1	$10t^2$
			1 < t <= 3	$10+[10(t-1)^2]+[3(t-1)]$
			3 < t < ∞	$56+[15(t-3)^2]+[5(t-3)]$
T_3	6.9	90	0 < t <= 5	$15t$
			5 < t <= 7.5	$125+[10(t-5)^2]$
			7.5 < t < ∞	$187.5+[3e^{(t-7.5)}]$
T_4	9	125	0 < t <= 3	$5t^2$
			3 < t <= 7	$45+[5(t-3)^2]+[5(t-3)]$
			7 < t < ∞	$145+[5(t-7)^2]+[10(t-7)]$
T_5	5.8	75	0 < t <= 0.4	$10t+8$
			0.4 < t <= 1.5	$12+[15(t-0.4)]$
			1.5 < t < ∞	$28.5+[5(t-1.5)^2]$
T_6	6	100	0 < t <= 2	$2e^t$
			2 < t <= 4	$[2e^2]+[2e^{(t-2)}]+[5(t-2)]$
			4 < t < ∞	$4e^2+20+[3e^{(t-4)}]$
T_7	6.9	110	0 < t <= 0.5	$6t$
			0.5 < t <= 2	$3+[4(t-0.5)^2]$
			2 < t < ∞	$12+e^{(t-2)}$
T_8	6	50	0 < t <= 0.2	$2t^3$
			0.2 < t <= 1.5	$0.016+[12(t-0.2)^2]$
			1.5 < t < ∞	$1.706+[3(t-1.5)^3]+[7(t-1.5)]$

T_9	6.8	85	$0 < t <= 2$	$[5t]+9$
			$2 < t <= 3$	$19+[10(t-2)]$
			$3 < t < \infty$	$54+[9(t-3)]$
T_{10}	5.6	125	$0 < t <= 1.5$	$5t^2$
			$1.5 < t <= 4.5$	$2.25+[6(t-1.5)^2]+[5(t-1.5)]$
			$4.5 < t < \infty$	$71.25+[2e^{(t-4.5)}]$
T_{11}	5	100	$0 < t <= 0.8$	$[8t]+5$
			$0.8 < t <= 1.5$	$11.4+[12(t-0.8)]$
			$1.5 < t < \infty$	$19.4+[4(t-1.5)^2]$
T_{12}	6.9	80	$0 < t <= 4$	$5t$
			$4 < t <= 8$	$20+[4(t-4)^2]$
			$8 < t < \infty$	$84+e^{(t-8)}$
T_{13}	5.8	90	$0 < t <= 2.2$	$10t$
			$2.2 < t <= 5.3$	$22+[10(t-2.2)^2]$
			$5.3 < t < \infty$	$118.1+[3e^{(t-5.3)}]$
T_{14}	6.4	75	$0 < t <= 2.5$	$7t+9$
			$2.5 < t <= 5.5$	$31.5+[12(t-2.5)]$
			$5.5 < t < \infty$	$67.5+[10(t-5.5)^2]$
T_{15}	6.5	150	$0 < t <= 0.5$	$3t^2$
			$0.5 < t <= 1.4$	$0.75+[3(t-0.5)^2]+[5(t-0.5)]$
			$1.4 < t < \infty$	$7.68+[3(t-1.4)^2]+[10(t-1.4)]$
T_{16}	6	65	$0 < t <= 0.8$	$7t$
			$0.8 < t <= 1.5$	$5.6+[4(t-0.8)^2]$
			$1.5 < t < \infty$	$7.56+e^{(t-1.5)}+[7(t-1.5)]$

To demonstrate the usage of simulation method for risk estimation, task 5 (T_5) is simulated for 50 replications using ARENA. Table 4.2 shows the summary of the results from the simulation.

Table 4.2. Summary of T_5 task times using simulation (50 Iterations)

Iteration	T_5	**Iteration**	T_5	**Iteration**	T_5	**Iteration**	T_5	**Iteration**	T_5
1	2.9	**11**	5.83	**21**	3.52	**31**	3.2	**41**	6.53
2	5.24	**12**	3.16	**22**	3.64	**32**	3.03	**42**	2.91

3	1.62	**13**	3.51	**23**	3.23	**33**	3.03	**43**	2.97
4	8.69	**14**	3.18	**24**	3.26	**34**	1.96	**44**	6.18
5	6.45	**15**	2.13	**25**	3.47	**35**	4.33	**45**	3.6
6	5.71	**16**	3.59	**26**	7.89	**36**	2.3	**46**	1.76
7	6.1	**17**	2.95	**27**	2.43	**37**	6.52	**47**	7.23
8	1.98	**18**	2.63	**28**	4.36	**38**	1.69	**48**	8.15
9	7.34	**19**	2.45	**29**	4.75	**39**	2.78	**49**	3.07
10	3.13	**20**	2.64	**30**	1.79	**40**	2.23	**50**	1.94

Consider iteration 4,

Observed task time (a_5) = 8.69 minutes

Standard task time (p_5) = 5.8 minutes

Exceeded time (t_5) = 2.89

Fixed cost (C_F) = $75

Cost function for t_5 lies in the interval „3‟and it is given by:

$$F_3(t) = 28.5 + \left[5\left(t - 1.5\right)^2\right]$$

F_3 (t) = $38.16

Risk of a task „R_5‟ is calculated as follows:

$$y_{5j} \quad \begin{Bmatrix} 1, \ a_5 > p_5 \\ 0, \text{Otherwise} \end{Bmatrix}$$

$Y_{5j} = 1$

$$R_5 = \frac{1}{N}\left[\sum_{j=1}^{N} y_{5j} \times \left(C_{F5j} + F_z 5(t)\right)\right]$$

Since N = 1, R_5 is reduced to as follows:

$$R_5 = y_5 \times \left(C_{F5} + F_{z5}(t)\right)$$

$R_5 = 1 \times (75 + 38.16)$

R_5 = $113.16

Similarly, risk of the T_5 is calculated for all iterations and Table 4.3 shows the risk value of each replications.

Table 4.3. Results of risk values of T_5 (50 Iterations)

Iteration	R_5 ($)	**Iteration**	R_5 ($)	**Iteration**	R_5 ($)	**Iteration**	R_5 ($)	**Iteration**	R_5 ($)
1	0	**11**	83.3	**21**	0	**31**	0	**41**	91.95
2	0	**12**	0	**22**	0	**32**	0	**42**	0
3	0	**13**	0	**23**	0	**33**	0	**43**	0
4	113.16	**14**	0	**24**	0	**34**	0	**44**	86.8
5	90.75	**15**	0	**25**	0	**35**	0	**45**	0
6	0	**16**	0	**26**	105.24	**36**	0	**46**	0
7	86	**17**	0	**27**	0	**37**	91.8	**47**	102.45
8	0	**18**	0	**28**	0	**38**	0	**48**	107.11
9	103.51	**19**	0	**29**	0	**39**	0	**49**	0
10	0	**20**	0	**30**	0	**40**	0	**50**	0

Based on the results of risk values, the average of R_5 is estimated to be $21.24. For more accurate risk estimation, each task is run for 1000 replications and Table 4.4 shows the results of risk of all tasks.

Table 4.4. Risk estimates of 16 tasks (case study 1)

Task	**Risk ($)**	**Task**	**Risk ($)**
$\mathbf{T_1}$	141.92	$\mathbf{T_9}$	66.79
$\mathbf{T_2}$	75.17	$\mathbf{T_{10}}$	100.43
$\mathbf{T_3}$	100.33	$\mathbf{T_{11}}$	98.18
$\mathbf{T_4}$	181.74	$\mathbf{T_{12}}$	128.73
$\mathbf{T_5}$	15.49	$\mathbf{T_{13}}$	70.48
$\mathbf{T_6}$	27.56	$\mathbf{T_{14}}$	53.39
$\mathbf{T_7}$	61.40	$\mathbf{T_{15}}$	64.87
$\mathbf{T_8}$	20.81	$\mathbf{T_{16}}$	15.58

The given assembly line is balanced in terms of cycle time and initial allocation of tasks without violating precedence relationship is shown in Table 4.5. Heuristic method is applied to the case study and the same Table 4.5 shows the task allocation order, values of risk, and station time before reallocation and after reallocation of tasks. Figure 4.3 shows that risk is balanced after reallocation.

Table 4.5. Task order - before and after reallocation (case study 1)

Workstation	**J_1**	**J_2**	**J_3**	**J_4**
Before Reallocation				
Task Allocation	$\{T_1, T_2, T_3, T_4\}$	$\{T_5, T_6, T_7, T_8\}$	$\{T_9, T_{10}, T_{11}, T_{12}\}$	$\{T_{13}, T_{14}, T_{15}, T_{16}\}$
Risk ($)	499.16	125.26	394.13	204.32
Station Time	25	24.7	24.3	24.7
After Reallocation				
Task Allocation	$\{T_4, T_{13}, T_{14}, T_5\}$	$\{T_1, T_2, T_{15}, T_8\}$	$\{T_{12}, T_{11}, T_7, T_{16}\}$	$\{T_{10}, T_3, T_9, T_6\}$
Risk ($)	321.10	302.76	303.89	295.11
Station Time	27	21.6	24.8	25.3

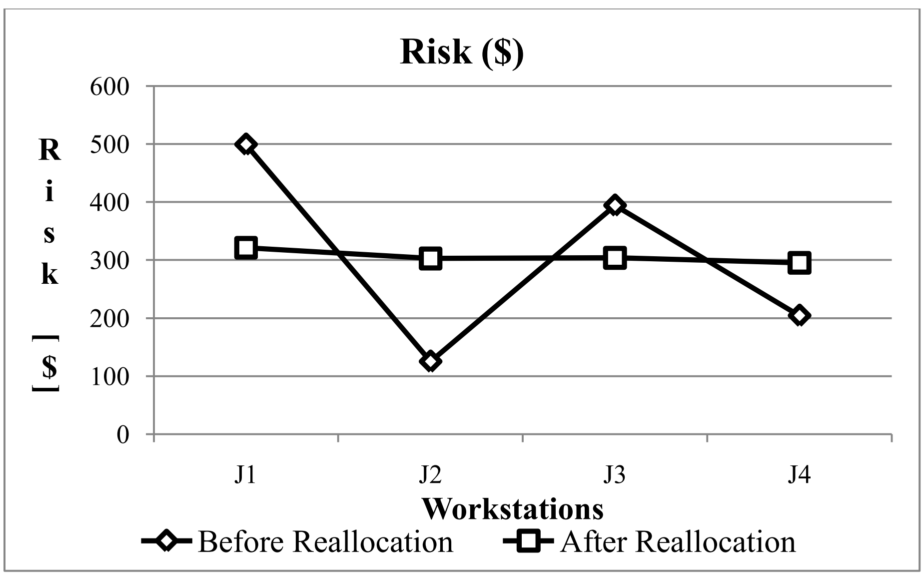


Figure 4.3. Risk balance chart (case study 1)

4.5.1 Validation of Heuristic

To validate the results of the heuristic approach for the case study, simulation has been used. For our research, simulation software ARENA is used as it can easily handle stochastic input to models. Using ARENA, the real time assembly line is translated into a simulation model. Once the model assumptions are considered, stochastic task times are given as input to the model. The model is run for completion of „F" products with „N"replications to observe the cycle time. The warm-up period is determined for monitoring the waiting and idle time of the assembly line. The cycle time is determined by calculating the average value of throughput for all replications.

4.5.2 Results

The model is simulated for 25 replications and each replication for 250 products. Initial 5000 minutes is assumed as warm-up period and data obtained during this period is deleted. Table 4.6 is the summary of results for all workstations in the assembly line. Results include, value added times (minutes), non-value added activity in time (minutes) and cost ($) for all workstations, and total cycle time for all products. Non-value added time is converted into non-value added cost by assuming cost per minute of non-value added activity is $5/min.

Table 4.6. Results (case study 1)

Workstation	**J_1**	**J_2**	**J_3**	**J_4**	**Completion Time (mins)**
Before Reallocation					
Value Added Time	8103.69	4391.39	7507.73	5605.82	9607.65
Non-Value Added Time (mins)	1503.96	5216.26	2099.92	4001.83	
Non-Value Added Cost ($)	7519.8	26081.3	10499.6	20009.15	
After Reallocation					
Value Added Time	6848.4	5771.79	6622.8	6400.47	9076.29
Non-Value Added Time (mins)	2227.89	3304.5	2453.49	2675.82	
Non-Value Added Cost ($)	11139.45	16522.5	12267.45	13379.1	

From Table 4.6, it can be observed that total completion time is reduced by 5.53% when risk is balanced. The reduction in total cycle time is due to the reduction in non-value added time. This eventually reduces the non-value added activity cost by 16.85%. Figure 4.4 shows the decrease in non-value added cost after reallocation and Figure 4.5 shows the reduction in total cycle time.

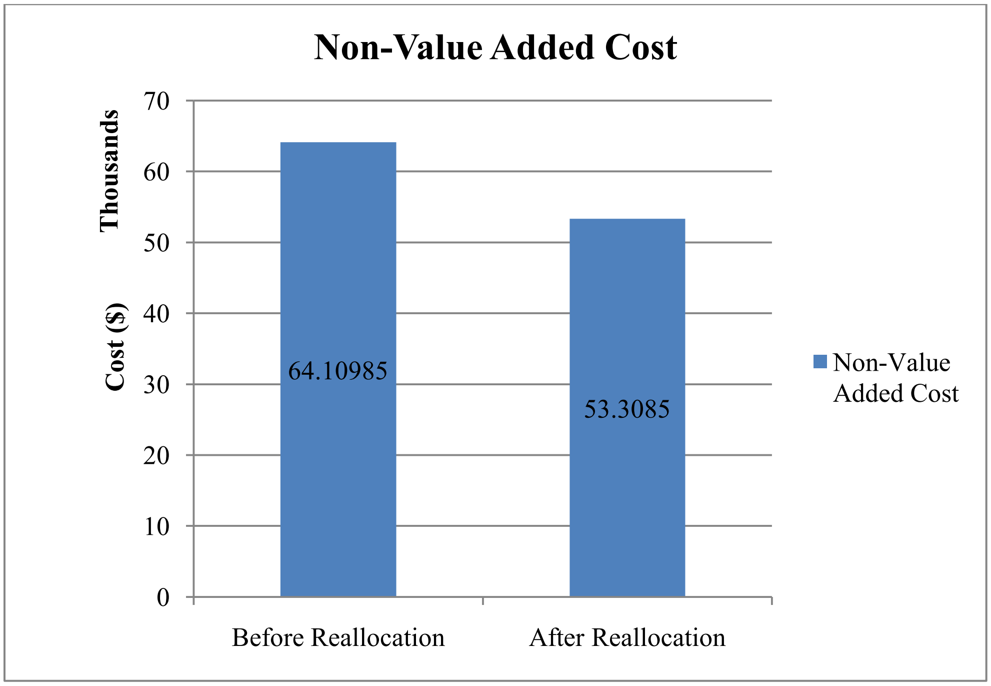


Figure 4.4. Reduction in non-value added cost (case study 1)

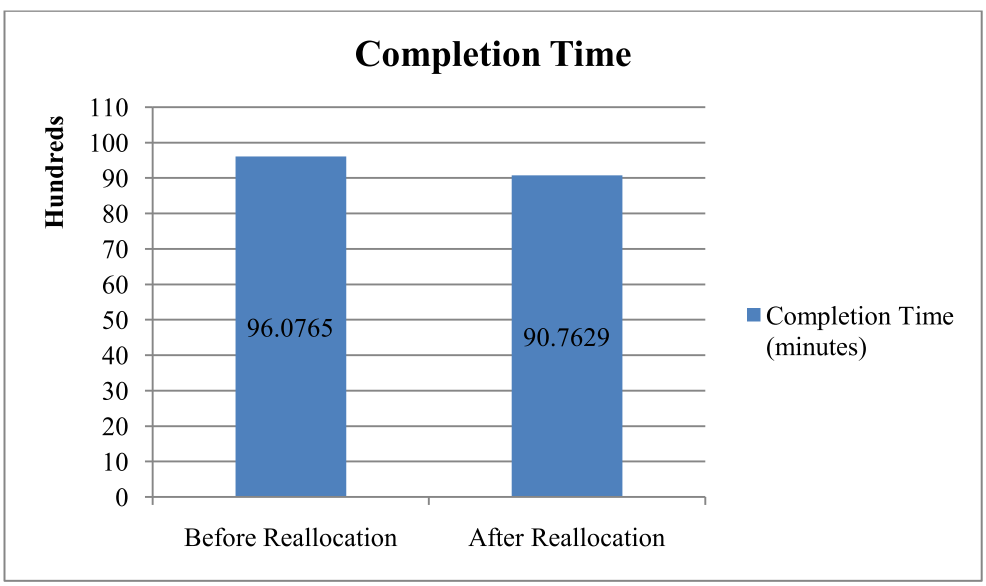


Figure 4.5. Reduction in total completion time (case study 1)

4.6 Case Study 2

The data for the case study 2 is the same as the one used in chapter 3. The data includes number of tasks, number of workstations, cycle time (minutes), allotted station time (minutes), standard task time (minutes), binary precedence table and production data for 40 tasks are the same listed in the previous chapter.

For more accurate risk estimation, each task is run for 1000 replications and Table 4.7 shows the results of risk of all tasks.

Table 4.7. Risk Estimates of 40 Tasks (case study 2)

Task	Risk ($)	Task	Risk ($)
T_1	100.39	T_{21}	66.79
T_2	74.01	T_{22}	124.38
T_3	98.18	T_{23}	110.77
T_4	141.92	T_{24}	128.73
T_5	100.33	T_{25}	22.71
T_6	177.18	T_{26}	3.24
T_7	181.74	T_{27}	27.84
T_8	104.14	T_{28}	61.40
T_9	15.22	T_{29}	20.81
T_{10}	27.56	T_{30}	36.06
T_{11}	26.72	T_{31}	0
T_{12}	15.49	T_{32}	8.26
T_{13}	27.5	T_{33}	52.60
T_{14}	26.60	T_{34}	47.36
T_{15}	18.82	T_{35}	32.60
T_{16}	13.08	T_{36}	70.48
T_{17}	70.94	T_{37}	15.58
T_{18}	75.17	T_{38}	53.39
T_{19}	72.04	T_{39}	37.58
T_{20}	100.43	T_{40}	92.86

The given assembly line is balanced in terms of cycle time and initial allocation of tasks without violating precedence relationship is shown in Table 4.8. Heuristic method is applied to the case study and the same Table 4.8 shows the task allocation order, values of risk, and station time before reallocation and after reallocation of tasks. Figure 4.6 shows that risk is balanced after reallocation.

Table 4.8. Task order - before and after reallocation (case study 2)

Workstation	**J_1**	**J_2**	**J_3**	**J_4**	**J_5**
Before Reallocation					
Task Allocation	$\{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8\}$	$\{T_9, T_{10}, T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}\}$	$\{T_{17}, T_{18}, T_{19}, T_{20}, T_{21}, T_{22}, T_{23}, T_{24}\}$	$\{T_{25}, T_{26}, T_{27}, T_{28}, T_{29}, T_{30}, T_{31}, T_{32}\}$	$\{T_{33}, T_{34}, T_{35}, T_{36}, T_{37}, T_{38}, T_{39}, T_{40}\}$
Risk ($)	977.89	170.99	749.25	180.32	402.45
Allotted Station Time	51.8	52.7	52.3	52.4	51.7
After Reallocation					
Task Allocation	$\{T_7, T_5, T_{19}, T_{38}, T_{30}, T_{13}, T_{15}, T_{26}\}$	$\{T_6, T_1, T_2, T_{28}, T_{35}, T_{14}, T_{12}, T_{16}\}$	$\{T_4, T_{20}, T_{18}, T_{17}, T_{33}, T_{10}, T_{29}, T_{32}, T_{31}\}$	$\{T_{24}, T_8, T_3, T_{21}, T_{34}, T_{11}, T_{37}, T_9\}$	$\{T_{22}, T_{23}, T_{40}, T_{36}, T_{39}, T_{27}, T_{25}\}$
Risk ($)	493.12	500.75	497.69	502.72	486.62
Station Time	53.2	51.9	53.2	51	51.6

Figure 4.6. Risk balance chart (case study 2)

To validate the results of the heuristic approach for the case study, simulation has been used. Assembly line is modeled using ARENA. The model is simulated for 30 replications and each replication for 250 products. Initial 5000 minutes is assumed as warm-up period and data obtained during this period is deleted. Table 4.9 is the summary of results for all workstations in the assembly line. Results include, value added times (minutes), non-value added activity in time (minutes) and cost ($) for all workstations, and total cycle time for all products. Non-value added time is converted into non-value added cost by assuming cost per minute of non-value added activity is $ 8/min.

Table 4.9. Results (case study 2)

Workstation	J_1	J_2	J_3	J_4	J_5	Completion Time (mins)
Before Reallocation						
Value Added Time	16453.41	9076.16	17067.58	8647.78	12410.66	17523.27
Non-Value Added Time (mins)	1069.86	8447.11	455.69	8875.49	5112.61	
Non Value Added Cost ($)	8558.88	67576.88	3645.52	71003.92	40900.88	
After Reallocation						
Value Added Time	12890.05	11554.99	11686.68	14174.52	13478.04	15279.32
Non Value Added Time (mins)	2389.27	3724.33	3592.64	1104.8	1801.28	
Non Value Added Cost ($)	19114.16	29794.64	28741.12	8838.4	14410.24	

From Table 4.9, it can be observed that total completion time is reduced by 12.80% when risk is balanced. The reduction in total cycle time is due to the reduction in non-value added time. This eventually reduces the non-value added activity cost by 47.36%. Figure 4.7 shows the decrease in non-value added cost after reallocation and Figure 4.8 shows the reduction in total cycle time.

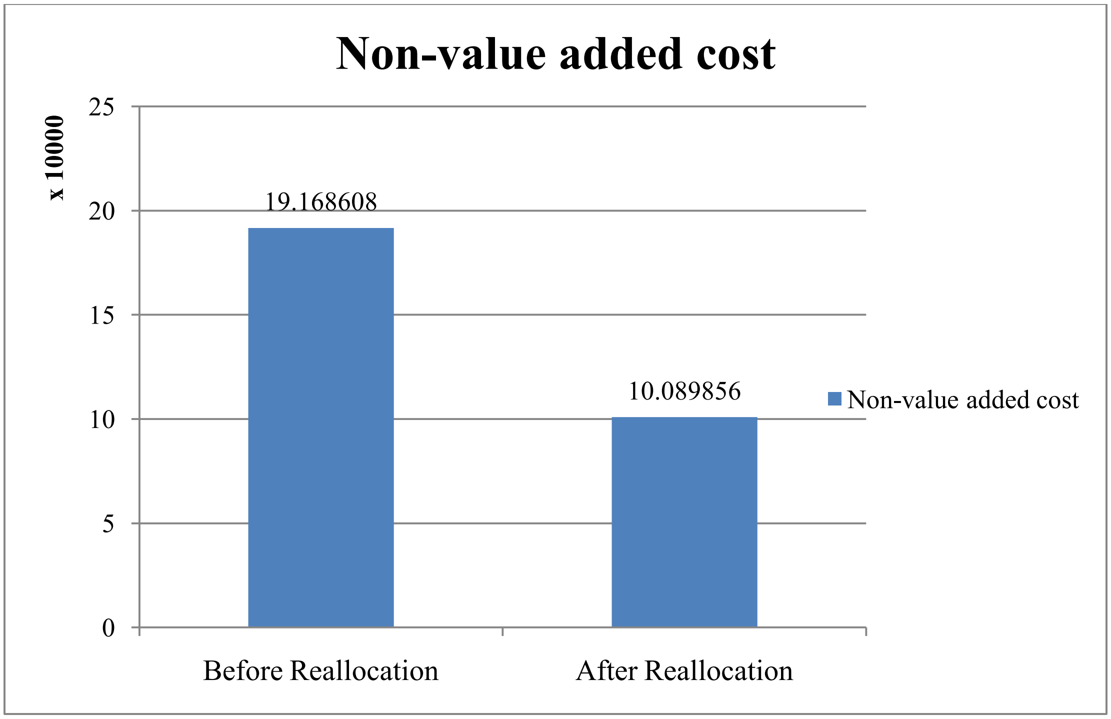


Figure 4.7. Reduction in non-value added cost (case study 2)

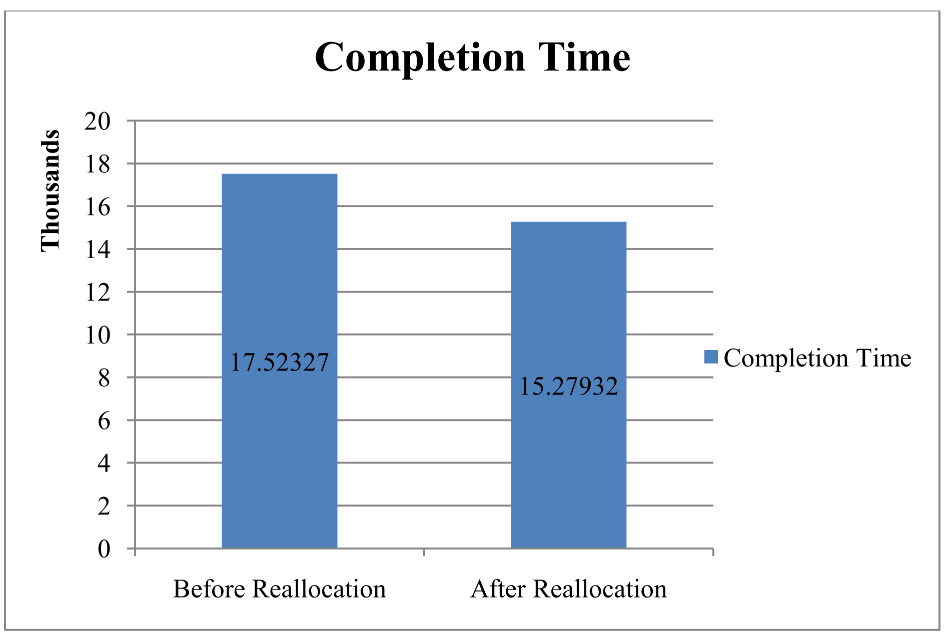


Figure 4.8. Reduction in total completion time (case study 2)

4.7 Conclusion

In this chapter, risk oriented assembly line balancing using a cost based approach is established. The proposed risk cost factor captures risk in terms of penalty cost due to delay in completion time. Risk based line balancing is achieved by balancing the risk between workstation. Hence, to balance the risk, two methods were developed; 1) Mathematical formulation method and 2) Heuristic method. The heuristic method is used to reassign the tasks among workstation by minimizing the maximum difference in risk between workstations. The results proved that there is a reduction in total completion time when the line is balanced based on risk.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The common problem that arises in any assembly line is assembly line balancing problem (ALBP). ALBP involves optimally distributing the work among the workstations (allocation of tasks to workstations) without violating assignment restrictions based on an objective such as minimizing the total cost, number of workstations, cycle time, maximizing the throughput, efficiency, etc. Assembly line balancing is said to be feasible, when all the tasks were allocated to the workstations without violating assignment restrictions, provided the station time of all workstations never exceed the cycle time. Due to probabilistic nature of task time, the task can exceed the expected standard task time at some instance. If a series of tasks exceeds in a particular station, then there is a risk that the product may exceed the cycle time. The larger the variability of task time, the higher the risk associated with the station. Risk is the potential loss caused when the product fails to complete within the specified station time. For line balancing, in addition to cycle time balancing, the risk should be balanced in order to increase the performance of the system.

Hence, the objective of this research work is to develop a methodology for risk based assembly line balancing. In Chapter 3, detailed assessment of relationship between variability in task time and risk was established. Case study conducted showed that variability measure is not sufficient to predict the risk of the task. Hence, new risk measures were proposed to identify the different levels of risk in order to predict the risk of the task. To quantify risk, risk index method was developed. With the estimated risk, mathematical formulation method was proposed to balance the risk in the assembly line. The objective of this approach is to minimize the maximum difference in risk between workstations. As any ALB problems, mathematical model is considered as an NP-hard problem. Hence, heuristic solution approach was developed for risk balancing. To validate the results of the heuristic approach, simulation has been used. Numerous case studies were conducted and the results showed that throughput is increased when risk is balanced. The increase in throughput is due to the reduction in non-value added time.

Chapter 4 discusses risk based assembly line balancing from the perspective of cost. A new risk cost factor method is proposed to measure the risk in terms of cost. The cost due to delay is given in terms of a fixed cost and a variable cost function. With the estimated risk, mathematical formulation method was proposed to balance the risk in the assembly line. As any ALB problems, mathematical model is considered as an NP-hard problem. Hence, heuristic solution approach was developed for risk balancing. To validate the results of the heuristic approach, simulation has been used. Numerous case studies were conducted and the results showed that total completion time is reduced when risk is balanced. The reduction in total completion time is due to the reduction in non-value added time.

5.2 Future Work

The risk based assembly line balancing is an efficient approach to deal variability in task time and also reduce its impact on station time, thereby reducing delay in product completion. The same problem can be extended to different assembly line types that include mixed-model assembly line, multi-model assembly line and zone-based assembly line. In addition to this, influence of paralleling of task within a workstation can be incorporated into risk based assembly line balancing approach.

REFERENCES

LIST OF REFERENCES

Anderson, E. J., and Ferris, M. C., (1994), "Genetic algorithms for combinatorial optimization: The assembly line balancing problem," *ORSA Journal of Computing*, Vol. 6, pp. 161-173.

Bautista, J., and Pereira, J., (2002), "Ant algorithms for assembly line balancing," *Lecture Notes in Computer Science*, Springer, Berlin, pp. 65-75.

Baybars, I., (1986), "A survey of exact algorithms for the simple assembly line balancing problems," *Management Science*, Vol. 32, No. 1, pp. 149-166.

Boysen, N., Fliedner, M., and Scholl, A., (2007), "A classification of assembly line balancing problems," *European Journal of Operational Research*, Vol. 183, No. 2, pp. 674-693.

Boysen, N., Fliedner, M., and Scholl, A., (2008), "Assembly line balancing: Which model to use when?," *International Journal of Production Economics*, Vol. 111, No. 2, pp. 509-528.

Chiang, W. C., (1998), "The application of a tabu search metaheuristic to the assembly line balancing problem," *Annals of Operations Research*, Vol. 77, pp. 209-227.

Dar-El, E. M., and Rubinovitch, Y., (1979), "MUST - A multiple solution technique for balancing single model assembly lines," *Management Science*, Vol. 25, pp. 1105-1114.

Freeman, D. R., (1968), "A general line balancing problem," *Proceedings of 19th Annual Conference AIIE,* Tampa, FLA., pp. 230-235.

Goncalves, J. F., and Almeida, J. R., (2002), "A hybrid genetic algorithm for assembly line balancing," *Journal of Heuristics*, Vol. 8, pp. 629-642.

Hackman, S. T., Magazine, M. J., and Wee, T. S., (1989), "Fast, effective algorithms for simple assembly line balancing problems," *Operations Research*, Vol. 37, pp. 916-924.

Heinrici, A., (1994), "A comparsion between simulated annealing and tabu search with an example from the production planning," *Operations Research Proceedings*, Springer, Berlin, pp. 498-503.

Held, M., Karp, R. M., and Shareshian, R., (1963), "Assembly line balancing – Dynamic programming with precedence constraints," *Operations Research*, Vol. 11, pp. 442-459.

Hopp, W. J., and Spearman, M. L., (2001), *Factory physics: Foundations of manufacturing management* (2 ed.), NY: Irwin/McGraw-Hill.

Jackson, J. R., (1956), "A computing procedure for a line balancing problem," *Management Science*, Vol. 2, No. 3, pp. 261-271.

Jackson, J. R., (1956), “A computing procedure for a line balancing problem,” *Management Science*, Vol. 2, pp. 261-271.

Johnson, R. V., (1983), “A branch and bound algorithm for assembly line balancing problems with formulation irregularities,” *International Journal of Production Research*, Vol. 29, No. 11, pp. 1309-1324.

Johnson, R. V., (1988), “Optimally balancing large assembly lines with “FABLE”,” *Management Science*, Vol. 34, pp. 240-253.

Kao, E. P. C., and Queyranne, M., (1982), “On dynamic programming methods for assembly line balancing,” *Operations Research*, Vol. 30, pp. 375-390.

Karp, R. M., (1972), “Reducibility among combinatorial problems,” *Complexity of Computer Applications*, Pleum, New York.

Kottas, J. F., and Lau, H. S., (1981), “A stochastic line balancing procedure,” *International Journal of Production Research*, Vol. 9, No. 2, pp. 177-193.

Lyu, J., (1997), “A single-run optimization algorithm for stochastic assembly line balancing problems,” *Journal of Manufacturing Systems*, Vol. 16, pp. 204-210.

Mastor, A. A., (1970), “An experimental investigation and comparative evaluation of production line balancing techniques,” *Management Science*, Vol. 16, No. 11, pp. 729-746.

Mcmullen, P. R., and Frazier, G. V., (1998), “Using simulated annealing to solve a multi objective assembly line balancing problem with parallel workstations,” *International Journal of Production Research*, Vol. 36, pp. 2717-2741.

Modarres, M., (2006), *Risk analysis in engineering, Techniques, Tools, and Trends*, New York, NY: Taylor & Francis.

Moodie, C. L., and Young, H. H., (1965), “A heuristic method of assembly line balancing for assumptions of constant or variable work element times,” *Journal of Industrial Engineering*, Vol. 16, pp. 23-29.

Nkasu, M. M., and Leung, K. H., (1995), “A stochastic approach to assembly line balancing,” *International Journal of Production Research*, Vol. 33, pp. 321-332.

Peeters, M., and Degraeve, Z., (2006), “An linear programming based lower bound for the simple assembly line balancing problem,” *European Journal of Operational Research*, Vol. 168, No. 3, pp. 716-731.

Pinto, P. A., Dannenbring, D. G., and Khumawala, B. M., (1975), “A branch and bound algorithm for assembly line balancing with paralleling,” *International Journal of Production Research*, Vol. 13, pp. 183-196.

Ponnambalam, S., Aravindan, G., and Naidu, M. G., (2000), “A multi-objective genetic algorithm for solving single model assembly line balancing problem,” *International Journal of Advanced Manufacturing Technology*, Vol. 16, pp. 341-352.

Reeve, N. R., and Thomas, W. H., (1973), “Balancing stochastic assembly lines,” *AIIE Transactions,* Vol. 5, pp. 223-229.

Sabuncuoglu, I., Erel, E., and Tanyer, M., (2000), “Assembly line balancing using genetic algorithms,” *Journal of Intelligent Manufacturing*, Vol. 11, pp. 295-310.

Salveson, M. E., (1955), “The assembly line balancing problem,” *The Journal of Industrial Engineering*, Vol. 6, No. 3, pp. 18-25.

Sarin, S. C., Erel, E., and Dar-El, E. M., (1999), “A methodology for solving single-model, stochastic assembly line balancing problem,” *Omega*, Vol. 27, pp. 525-535.

Scholl, A., (1999), *Balancing and sequencing assembly lines 2nd ed*, Physica, Heidelberg.

Scholl, A., and Becker, C., (2006), “State-of-the-art exact and heuristic solution procedures for simple assembly line balancing,” *European Journal of Operational Research*, Vol. 168, No. 3, pp. 666-693.

Scholl, A., and Vob, S., (1996), “Simple assembly line balancing – Heuristic approaches,” *Journal of Heuristics*, Vol. 2, pp. 217-244.

Shin, D., (1990), “An efficient heuristic for solving stochastic assembly line balancing problems,” *Computers and Industrial Engineering*, Vol. 18, No. 3, pp. 285-295.

Spreecher, A., (1999), “A competitive branch-and-bound algorithm for the simple assembly line balancing problem,” *International Journal of Production Research*, Vol. 37, pp. 1787-1816.

Suresh, G., and Sahu, S. (1994), “Stochastic assembly line balancing using simulated annealing,” *International Journal of Production Research*, Vol. 32, pp. 1801-1810.

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

Thangavelu, S. R., and Shetty, C. M., (1971), “Assembly line balancing by zero-one integer programming,” *AIIE Transactions*, Vol. 3, pp. 61-68.

Ugurdag, H. F., Rachamadugu, R., and Papachristou, C. A., (1997), “Designing paced assembly lines with fixed number of stations,” *European Journal of Operational Research*, Vol. 102, No. 3, pp. 488-501.

Wee, T. S., and Magazine, M. J., (1982), “Assembly line balancing as generalized bin packing”, *Operations Research Letters*, Vol. 1, pp. 56-58.

White, W. W., (1961), “Comments on a paper by Bowman,” *Operation Research*, Vol. 9, pp. 274-276.

APPENDICES

APPENDIX A

Binary Precedence Table (Case Study 2)

$$
P_{ij} = \left\{\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}\right\}
$$

APPENDIX B

Cost Function of 40 Tasks (Case Study 2)

Task	Standard Task Time	Fixed Cost	Intervals	Cost Function
T_1	3.1	100	0 < t <= 1.5	$10t$
			1.5 < t <= 2.5	$25+\left[20(t-1.5)\right]$
			2.5 < t < ∞	$45+\left[12(t-2.5)^2\right]$
T_2	3.5	80	0 < t <= 1	$8t^2$
			1 < t <= 1.5	$8+\left[8(t-1)^2\right]+\left[5(t-1)\right]$
			1.5 < t < ∞	$12.5+\left[10(t-1.5)^2\right]$
T_3	5	100	0 < t <= 0.8	$8t+5$
			0.8 < t <= 1.5	$11.4+12(t-0.8)$
			1.5 < t < ∞	$19.8+\left[4(t-1.5)^2\right]$
T_4	5.1	100	0 < t <= 2.5	$20t$
			2.5 < t <= 5.5	$50+\left[30(t-2.5)\right]$
			5.5 < t < ∞	$145+\left[5(t-7)^2\right]+\left[10(t-7)\right]$
T_5	6.9	90	0 < t <= 5	$15t$
			5 < t <= 7.5	$125+\left[10(t-5)^2\right]$
			7.5 < t < ∞	$187.5+3e^{(t-7.5)}$
T_6	8.2	100	0 < t <= 4	$6t^2+5t$
			4 < t <= 10	$116+\left[8(t-4)^2\right]$
			10 < t < ∞	$404+\left[2(t-10)^3\right]$
T_7	9	125	0 < t <= 3	$5t^2$
			3 < t <= 7	$45+\left[5(t-3)^2\right]+\left[5(t-3)\right]$
			7 < t < ∞	$145+\left[5(t-7)^2\right]+\left[10(t-7)\right]$
T_8	11	50	0 < t <= 2.4	$2t^2$
			2.4 < t <= 7.4	$11.52+\left[12(t-2.4)^2\right]$
			7.4 < t < ∞	$311.52+\left[3(t-7.4)^2\right]+\left[7(t-7.4)\right]$
T_9	3.2	85	0 < t <= 1	$\left[5t\right]+9$
			1 < t <= 3	$14+\left[10(t-1)\right]$
			3 < t < ∞	$59+\left[9(t-3)\right]$

T_{10}	3.9	125	$0 < t <= 1.5$	$5t^2$
			$1.5 < t <= 3.5$	$11.25+\left[6(t-1.5)^2\right]+\left[5(t-1.5)\right]$
			$3.5 < t < \infty$	$45.25+\left[2e^{(t-3.5)}\right]$
T_{11}	4	100	$0 < t <= 0.7$	$6t+4$
			$0.7 < t <= 2.2$	$8.2+\left[11(t-0.7)\right]$
			$2.2 < t < \infty$	$24.7+\left[7(t-2.2)^2\right]$
T_{12}	5.8	75	$0 < t <= 0.4$	$10t+8$
			$0.4 < t <= 1.5$	$12+\left[15(t-0.4)\right]$
			$1.5 < t < \infty$	$28.5+\left[5(t-1.5)^2\right]$
T_{13}	6	100	$0 < t <= 2$	$2e^{t}$
			$2 < t <= 4$	$14.78+\left[2e^{(t-2)}\right]+5(t-2)$
			$4 < t < \infty$	$39.56+\left[3e^{(t-4)}\right]$
T_{14}	9	75	$0 < t <= 2.5$	$7t+9$
			$2.5 < t <= 5.5$	$31.5+\left[12(t-2.5)\right]$
			$5.5 < t < \infty$	$67.5+\left[10(t-5.5)^2\right]$
T_{15}	10	150	$0 < t <= 1.5$	$2t^2$
			$1.5 < t <= 2.4$	$4.5+\left[2(t-1.5)^2\right]+\left[3(t-1.5)\right]$
			$2.4 < t < \infty$	$8.82+\left[2(t-2.4)^2\right]+\left[5(t-2.4)\right]$
T_{16}	10.8	65	$0 < t <= 0.8$	$10t$
			$0.8 < t <= 1.5$	$8+\left[4(t-0.8)^2\right]$
			$1.5 < t < \infty$	$9.96+e^{(t-1.5)}$
T_{17}	3.1	80	$0 < t <= 4$	$5t$
			$4 < t <= 8$	$20+\left[6(t-4)^2\right]$
			$8 < t < \infty$	$116+e^{(t-8)}$
T_{18}	4	80	$0 < t <= 1$	$10t^2$
			$1 < t <= 3$	$10+\left[10(t-1)^2\right]+\left[3(t-1)\right]$
			$3 < t < \infty$	$56+\left[15(t-3)^2\right]+\left[5(t-3)\right]$
T_{19}	4.1	75	$0 < t <= 3$	$10t$
			$3 < t <= 6$	$36+\left[5(t-3)\right]$
			$6 < t < \infty$	$51+(t-6)^2$

T_{20}	5.6	125	0 < t <= 1.5	$5t^2$
			1.5 < t <= 4.5	$11.25+\left[6(t-1.5)^2\right]+\left[5(t-1.5)\right]$
			4.5 < t < ∞	$80.25+2e^{(t-4.5)}$
T_{21}	6.8	85	0 < t <= 2	$5t+9$
			2 < t <= 3	$19+\left[10(t-2)\right]$
			3 < t < ∞	$54+\left[9(t-3)\right]$
T_{22}	6.9	80	0 < t <= 4	$5t$
			4 < t <= 8	$20+\left[4(t-4)^2\right]$
			8 < t < ∞	$84+e^{(t-8)}$
T_{23}	10.8	100	0 < t <= 6	$5t+2$
			6 < t <= 10	$32+8(t-6)$
			10 < t < ∞	$64+e^{(t-10)}$
T_{24}	11	80	0 < t <= 3.4	$10t$
			3.4 < t <= 7.4	$34+6(t-3.4)^2$
			7.4 < t < ∞	$130+8(t-7.4)^2$
T_{25}	2.5	90	0 < t <= 2	$6t$
			2 < t <= 5	$12+6(t-2)^2$
			5 < t < ∞	$66+e^{(t-5)}$
T_{26}	3.1	75	0 < t <= 2.5	$2t+9$
			2.5 < t <= 5.5	$19+7(t-2.5)$
			5.5 < t < ∞	$40+(t-5.5)^2$
T_{27}	3.9	150	0 < t <= 0.6	t^2
			0.6 < t <= 1.6	$0.36+(t-0.6)^2+5(t-0.6)$
			1.6 < t < ∞	$6.36+(t-1.6)^2+10(t-1.6)$
T_{28}	6.9	110	0 < t <= 0.5	$6t$
			0.5 < t <= 2	$3+4(t-0.5)^2$
			2 < t < ∞	$12+e^{(t-2)}$

T_{29}	6	50	0 < t <= 0.2	$2t^3$
			0.2 < t <= 1.5	$0.016+12(t-0.2)^2$
			1.5 < t < ∞	$20.296+3(t-1.5)^3+7(t-1.5)$
T_{30}	7.7	125	0 < t <= 1	$5t^2$
			1 < t <= 4.5	$5+6(t-1)^2$
			4.5 < t < ∞	$78.5+e^{(t-4.5)}$
T_{31}	10.8	100	0 < t <= 1.8	$4t+5$
			1.8 < t <= 2.5	$12.2+11(t-1.8)$
			2.5 < t < ∞	$19.9+(t-2.5)^2$
T_{32}	11.5	80	0 < t <= 2	$5t$
			2 < t <= 3	$10+(t-2)^2$
			3 < t < ∞	$11+e^{(t-3)}$
T_{33}	3.2	90	0 < t <= 2.5	$4t$
			2.5 < t <= 4	$10+5(t-2.5)^2$
			4 < t < ∞	$21.25+e^{(t-4)}$
T_{34}	4	75	0 < t <= 1.5	$t+9$
			1.5 < t <= 4.5	$10.5+2(t-1.5)$
			4.5 < t < ∞	$16.5+4(t-4.5)^2$
T_{35}	4.6	50	0 < t <= 0.5	$t+20$
			0.5 < t <= 1.5	$20.5+(t-0.5)^2+(t-0.5)$
			1.5 < t < ∞	$22.5+3(t-1.5)^2$
T_{36}	5.8	90	0 < t <= 2.2	$10t$
			2.2 < t <= 5.3	$22+\left[10(t-2.2)^2\right]$
			5.3 < t < ∞	$118.1+\left[3e^{(t-5.3)}\right]$
T_{37}	6	65	0 < t <= 0.8	$7t$
			0.8 < t <= 1.5	$5.6+4(t-0.8)^2$
			1.5 < t < ∞	$7.56+e^{(t-1.5)}+\left[7(t-1.5)\right]$

T_{38}	6.4	75	0 < t <= 2.5	$7t+9$
			2.5 < t <= 5.5	$31.5+12(t-2.5)$
			5.5 < t < ∞	$67.5+10(t-5.5)^2$
T_{39}	10.2	75	0 < t <= 3.5	$9t$
			3.5 < t <= 5.5	$31.5+5(t-3.5)^2$
			5.5 < t < ∞	$51.5+e^{(t-5.5)}$
T_{40}	11.5	150	0 < t <= 0.5	t^2
			0.5 < t <= 2	$0.25+3(t-0.5)^2$
			2 < t < ∞	$7+5(t-2)^2$