RISK BASED WORKER ALLOCATION AND LINE BALANCING

A Thesis by

Sathya Madhan Solaimuthu Pachaimuthu

Bachelor of Engineering, Anna University, India, 2008

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

December 2010

RISK BASED WORKER ALLOCATION AND 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

Michael Jorgensen, Committee Member

Ramazan Asmatulu, Committee Member

## ABSTRACT

In general, manufacturing systems could be classified into machine intensive manufacturing and labor intensive manufacturing. From the previous studies, we can infer that worker allocation plays an important role in determining efficiency of a labor intensive manufacturing system. Most of the research works in the previous literature is performed in a deterministic bed. But from the time study data that was obtained from a local aircraft company shows a high degree of variability in worker processing times.

Thus this research presents a worker allocation approach which also considers the uncertainty in worker processing times into account. Risk based worker allocation approach is developed for three different scenarios. First scenario is the single task per station balanced production line scenario, where workers are allocated to processes by minimizing the overall risk of delay due to workers. In the second scenario, in addition to worker allocation by minimizing the overall risk, multiple workers are allocated to processes to make the flow of products uniform in a single task per station unbalanced production line. Prior to implementing the final approach, a method for line balancing when variability is involved is studied and compared to the ranked positional-weight method. The final scenario developed is a simultaneous approach to balance and allocate workers in a multiple task per station production line. Case studies were simulated using QUEST software and the result indicates that risk based allocation has increased throughput and efficiency compared to deterministic worker allocation.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF TABLES (continued)

LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 Problem Background

Production line is a flow line manufacturing system in which the resources in a factory are efficiently organized to transform raw materials into finished products. In a production line, product flows through set of sequential value added activities called processes. In order to achieve high production efficiency, all the associated processes should be efficient. Often, production lines are employed in a high volume, low variety manufacturing environment.

In general, manufacturing systems could be classified into machine-intensive manufacturing and labor-intensive manufacturing (Suer, 1996). In machine intensive manufacturing, productivity of the system is primarily based on total number of machines available in the system. Involvement of worker is often limited in machine intensive manufacturing. Workers are bound to tasks such as loading/unloading parts from the machines, transferring products from one station to other, etc. On the contrary, in a labor intensive manufacturing system, performance of a system typically depends on worker involvement. Workers are equipped with small, inexpensive equipments performing the processes on products. Labor intensive manufacturing is more predominant in jewelry, apparel, leather, and sport goods manufacturing industries (Das and Kalita, 2009). A greater importance on worker allocation has to be imposed in order to obtain a strategic competitive advantage in manufacturing systems (Jordon, 1977). Thus, worker allocation plays an important role in determining the efficiency of a labor intensive manufacturing system.

## 1.2 Worker Allocation Problem

Worker allocation problem is defined as the allocation of best worker to the process in a manufacturing system, thereby increasing performance of the system. Traditionally, worker is allocated to processes based on several criteria such as experience, productivity, seniority, and often some arbitrary measures (Nembhard, 2001). In a competitive environment, production efficiency plays an important role in determining profit of a firm. For a system to be highly productive and efficient, worker should be allotted to the processes based on productivity measures. Processing time and quality level of worker associated with the processes are the dominant productivity measures in context of worker allocation. Thus, a worker with the lowest processing time and highest quality is more likely to be allotted to the process.

Worker allocation is done by assuming three different strategies, such as Single worker-Single process, Single worker-Multi process, and Multi worker-Multi process. In "Single worker-Single process", workers possess only one skill. In "Single worker-Multi process", workers possess multiple skills. The selected best worker moves between multiple processes and performs the operations. It is more common in U-Shaped production lines. In "Multi worker-Multi process", multiple workers possess multiple skills such that skills overlap within workers. Thus, worker with highest level of productivity for a particular skill is more likely to be selected.

## 1.3 Worker Profile

Worker profile could be defined as the operational characteristics associated with individual workers. Based on previous literature, worker allocation could be classified into:

1) Sole profile allocation
2) Multi profile allocation

### 1.3.1 Sole Profile Allocation

In sole profile allocation, all workers with similar skills are assumed to have same productivity measures. The differences in productivity due to inherent variability associated with workers are not considered. Several formulations were developed to solve sole profile worker allocation model and are shown in Table 1.1.

**Table 1.1 Formulations - Sole Profile Allocation**

| FORMULATIONS | AUTHORS |
|---|---|
| Mixed integer programming | Kuo and Yang (2005), Suer and Bera (1998), Davis and Mabert (2000), Min and Shin (1993) and Suer (1996) |
| Heuristic | Vembu and Srinivasan (1997), Bhaskar and Srinivasan (1997), and Nakade and Ohno (1999) |
| Network flow problem | Wittrock, 1992 |
| Data envelopment analysis | Ertay and Ruan, 2005 |
| Non linear programming | Davis and Mabert, 2000 |

Wittrock (1992) modeled sole profile operator assignment problem as a network flow problem with a lexicographic objective which tries to maximize the capacity in a machine intensive manufacturing system. Vembu and Srinivasan (1997) developed a heuristic approach for operator allocation and product sequencing in production lines with an objective of minimizing makespan. Bhaskar and Srinivasan (1997) used a heuristic approach to solve static and dynamic variety worker allocation problem.

Suer (1996) proposed a two-stage hierarchical methodology which simultaneously does operator allocation and cell loading in a labor intensive manufacturing system. Suer and Bera (1998) is an extension of previous work by Suer (1996). In this research, lot-splitting between cells are allowed and setup times for the products were included. Kuo and Yang (2005) implemented mixed integer formulation developed by Suer and Bera (1998) for operator staffing level decisions in a TFT-LCD inspection and packaging (I/P) process.

Min and Shin (1993) developed a multiple objective sole profile allocation model, which simultaneously forms machine and human cells in cellular manufacturing. Davis and Mabert (2000) presented two mathematical models for making order dispatching and worker assignment decisions in linked cellular manufacturing systems. Ertay and Ruan (2005) presented a data envelopment analysis approach for optimal number of worker allocation in cellular manufacturing. Nakade and Ohno (1999) proposed a heuristic which optimally selects minimum number of workers which minimizes overall cycle time thereby meeting demand in a U-shaped production line. In all the research papers discussed above, it is assumed that workers possess equal productivity. However, in a real world scenario, variability in worker productivity is predominant.

### 1.3.2 Multi Profile Allocation

In multi profile allocation, worker differences in terms of productivity are considered. Worker profile differences are modeled either based on multiple skill levels or individual workers. The worker profile difference based on individual workers is more realistic assumption. Several formulations such as were developed to solve multi profile worker allocation model and is shown in Table 1.2.

Nembhard (2001) developed a heuristic approach for multi profile worker allocation based on individual worker learning profiles for machine intensive manufacturing system. Chaves, Insa, and Lorena (2007) modeled an integer programming formulation for assembly line worker assignment and balancing problem (ALWABP) in sheltered work centers. This model assumes that individual workers have different deterministic processing time values. Miralles, Garcia, Andres, and Cardos (2008) extended the previous work by Chaves, Insa, and Lorena (2007) by providing a different solution methodology using branch and bound algorithm. Fowler, Wirojanagud, and Gel (2008) developed a mixed integer program for making decisions on hiring, training, and firing workers. Individual workers are assumed to have different profiles based on their general cognitive ability.

**Table 1.2 Formulations - Multi Profile Allocation**

| FORMULATIONS | AUTHORS |
|---|---|
| Mixed integer programming | Askin and Huang (2001), Chaves, Insa, and Lorena (2007), Miralles et al., (2008), Suer and Tummaluri (2008), Fitzpatrick and Askin (2005) , and Norman et al., (2002) |
| Heuristics | Fowler, Wirojanagud, and Gel (2008) and Nembhard (2001) |
| Non linear integer programming | Aryanezhad, Deljoo, and Al-e-Hasheem (2009) and Heimerl and Kolish (2009) |
| Particle swam optimization technique | Yaakob and Watada, 2009 |

Suer and Tummaluri (2008) developed a three stage operator allocation procedure for worker allocation in labor intensive manufacturing. This research assumes that the processing time of individual workers are different deterministic values depending on skill level of workers. Heimerl and Kolish (2009) modeled a non-linear integer program for assigning multi-skilled workers to tasks considering the worker learning, forgetting and company skill level targets.

Askin and Huang (2001) developed a mixed integer, goal programming model to form worker teams for cellular manufacturing based on psychological, organizational, and technical factors. Norman et al., (2002) modeled a mixed integer program which assigns workers to task in a cellular manufacturing environment considering both technical and human skills. Fitzpatrick and Askin (2005) formed multiple worker teams with multifunctional skill requirements for cellular manufacturing. The technical and inter-personnel skills of workers are considered to form worker teams. McDonald, Ellis, Aken, and Koelling (2009) proposed a mathematical model to assign cross trained workers to tasks in a lean manufacturing cell. This research considers both processing time and quality level of workers and assumes it as different deterministic values based on current skill depth level of workers. Aryanezhad, Deljoo, and Al-e-Hasheem (2009) formulated a non linear integer program for simultaneous dynamic cell formation and worker assignment problem (SDCWP). Yaakob and Watada (2009) developed a methodology for worker assignment in cellular manufacturing using particle swam optimization technique. In all the research discussed above, it is assumed that workers possess different deterministic productivity values. However, the time study data that we obtained from a local aircraft industry shows a high degree of uncertainty in worker productivity measures.

## 1.4 Research Purpose

Worker allocation is primarily based on the processing time and quality level of workers. All the researches in previous literature assume deterministic processing time and quality level of workers. A change in processing time or quality level of any worker will result in a modified relationship between the worker and the process, thus affecting the optimal worker allocation. From the time study data that we obtained from a local aircraft manufacturing company, we saw a high degree of variability in worker processing time. Task processing time details for a single task replicated for 20 times is shown in Table 1.3.

Thus, in real world scenarios, uncertainty is predominant in worker processing time and quality level. Therefore, an optimal worker allocation methodology should take into consideration the associated uncertainties.

**Table 1.3 Sample Task Processing Time Data**

| TASK | PROCESSING TIME (min) |
|---|---|
| 1 | 24.7 |
| 2 | 32.9 |
| 3 | 26.6 |
| 4 | 27.5 |
| 5 | 30.8 |
| 6 | 31 |
| 7 | 32.2 |
| 8 | 26.7 |
| 9 | 27.8 |
| 10 | 25.9 |
| 11 | 16.9 |
| 12 | 15.9 |
| 13 | 30.7 |
| 14 | 12.9 |
| 15 | 25 |
| 16 | 4.1 |
| 17 | 20 |
| 18 | 13.8 |
| 19 | 12.4 |
| 20 | 20 |

When there is an uncertainty associated with worker allocation process, it is essential to assess and quantify the risk associated with the workers. According to Modarres (2006), risk is defined as "a measure of the potential loss occurring due to natural or human activities". In the context of worker allocation, risk can be defined as the potential loss due to delay in process or due to bad quality of the product. When variability in processing time and quality level increases, risk due to delay in process and bad quality product also increases. In order to reduce risk in worker allocation process, advanced techniques have to be developed which captures the risk and minimizes its impact.

## 1.5 Research Objective

In general, production lines can be classified into: a) balanced line and unbalanced lines or b) production lines with single task per station and multiple tasks per station. Thus, the objective of this research is to develop and solve a risk-based worker allocation model which takes into account the associated uncertainties in production lines which has:

1. Single task per station - balanced line scenario
2. Single task per station - unbalanced line scenario
3. Multiple tasks per station scenario

In single task per station balanced line scenario, the objective of the risk based worker allocation is to allocate the best worker to the station by minimizing the overall risk in the production line. Two kinds of risks considered in this research are the processing time risk and quality level risk. In single task per station unbalanced line scenario, since the line is unbalanced, multiple workers have to be allocated in the bottleneck stations to ensure that the demand is met. Thus, our objective in single task per station unbalanced line scenario is to allocate multiple workers in bottleneck stations, thereby minimizing the overall risk in production line and to ensure that demand is met. In multiple tasks per station scenario, since each station has multiple tasks allocated to them, the efficiency is dependent upon both line balancing and worker allocation. Hence, the objective for multiple tasks per station scenario is to simultaneously balance the production line and to allocate the workers thereby minimizing the overall risk in the production line.

In order to achieve the above stated objective, this research is organized in to six chapters. Chapter 2 provides a detailed literature survey on various worker allocation approaches. Chapter 3 presents methodologies for risk based worker allocation in single task per station balanced and unbalanced line scenarios. Chapter 4 deals with a simplified case of Chapter 5 wherein a risk based methodology for line balancing for multiple tasks per station scenario is proposed. Chapter 5 presents a simultaneous approach for risk based line balancing and worker allocation approach for multiple tasks per station scenario. Lastly, Chapter 6 presents the conclusion for this research and possible extensions of this research.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

The purpose of this research was to address the worker allocation problem in a stochastic environment. This chapter provides a thorough review of previous literature in the field of worker allocation for a comprehensive understanding of worker allocation problem. Section 2.2 discusses the characteristics and types of production lines. Section 2.3 introduces the concept of worker allocation problem and its classification based on previous research. Section 2.4 explains the sole profile allocation and reviews the associated literature. Section 2.5 elaborates multi profile allocation and discusses the previous research. Section 2.6 summarizes the previous literature in the field of worker allocation and explains the motivation for this research.

## 2.2 Production Lines

Production line is a flow line manufacturing system in which raw material is transformed into finished goods by efficiently organizing resources in the factory. In a production line, set of sequential value added activities are performed on the products called processes. Production lines are preferred in high volume, low variety manufacturing.

Manufacturing systems could be generally classified into machine intensive manufacturing and labor intensive manufacturing (Suer, 1996). In a machine intensive manufacturing, productivity of the system depends on the number/capacity of machines available in the system. Involvement of workers is limited to tasks such as loading/unloading jobs and transferring jobs, etc. On the contrary, in labor intensive manufacturing, productivity of the system depends on the number/efficiency of workers available in the system. Workers are equipped with small, inexpensive equipment performing the processes on products. Thus, worker allocation plays an important role in determining the efficiency of a labor intensive manufacturing system.

## 2.3 Worker Allocation

Worker allocation could be defined as allocating the best worker to a process, thereby optimizing the performance of the system. In order to optimize the performance of the system, worker allocation has to be done based on productivity of the workers. Processing time and quality level of workers are the dominant productivity measures in context with worker allocation.

Worker profile could be defined as the operational characteristics of individual workers associated with processes. Based on the previous literature, worker allocation could be classified into:

1) Sole profile allocation
2) Multi profile allocation

## 2.4 Sole Profile Allocation

In sole profile allocation, all workers with similar skills are assumed to have the same productivity. The differences in productivity due to inherent variability associated with workers are not considered. Sole profile allocation is often preferred for simplicity in mathematical modeling. Since individual worker profiles are not considered in sole profile allocation, most of the researches determine the worker staffing level at each work station. The following sections provide an outline of sole profile worker allocation formulations and solution procedures.

### 2.4.1 Formulations in Sole Profile allocation

Conventionally, different kinds of formulations were developed to solve sole profile worker allocation problem as shown in Table 1.1.

In mixed integer programming, the objective of most of the approaches is to maximize the rate of production (Kuo and Yang, (2005), Suer, and Bera, (1998) and Suer, (1996)). Suer (1996) proposed a two-stage hierarchical methodology which simultaneously does operator allocation and cell loading in a labor intensive manufacturing system. The first stage in the proposed methodology used a mixed integer formulation to form alternative operator level allocation for all the products. The second stage was to ensure that the demand was met with minimum optimal operators and cell loading using an integer programming formulation.

Suer and Bera (1998) extended the previous work by Suer (1996), which is also a two-stage hierarchical methodology which does operator allocation and cell loading in a labor intensive manufacturing system. In this research lot-splitting between cells is allowed and setup times for products were included. Kuo and Yang (2005) implemented mixed integer formulation developed by Suer and Bera (1998) for operator staffing level decisions in a TFT-LCD inspection and packaging (I/P) process. Optimal operator staffing level decisions were made for three operations in I/P process.

Min and Shin (1993) developed a multiple objective model which simultaneously formed machine and human cells. In this method, workers with similar skills were grouped together to manufacture similar products. A mixed integer goal programming model was developed with various constraints which effectively formed machine-worker group. A heuristic procedure was also developed to efficiently solve the mixed integer goal programming model. The problem was solved in two steps. The first step was to form machine and part grouping using traditional cell formation techniques and the second step was to allot workers to the cells.

Davis and Mabert (2000) presented two mathematical models for making order dispatching and worker assignment decisions in linked cellular manufacturing systems. The first method used a mixed integer program to simultaneously make worker assignment and order dispatching decisions. The second method worked in two steps, where order dispatching was done using a mixed integer program and worker assignment was performed using a non-linear programming model. Workers were classified into three skill levels and the appropriate worker with required skill level for the operation was allotted. However, individual worker profile was not analyzed in this research.

Wittrock (1992) modeled operator assignment problem as a network flow problem with a lexicographic objective which tried to maximize the capacity of the system. The network flow problem was solved using "parametric preflow" algorithm devised by Gallo, Grigoriadis, and Tarjan. Ertay and Ruan (2005) presented a data envelopment analysis approach for optimal number of worker allocation in cellular manufacturing.

Although different mathematical methods were used to solve worker allocation problem, when the problem size increased, difficulty to solve for optimality also increased. So, efficient heuristic approaches are devised to solve for near-optimal solution (Vembu and Srinivasan, (1997), Bhaskar and Srinivasan, (1997), and Nakade and Ohno, (1999)).

In heuristic approaches, the objective of most of the algorithms was to minimize the makespan. Vembu and Srinivasan (1997) developed a heuristic approach for operator allocation and product sequencing in production lines with an objective of minimizing makespan. They divided a production line into cells with unidirectional flow. Heuristics selects the best optimal sequence of products with minimized makespan and finds the optimal worker level.

Bhaskar and Srinivasan (1997) used a heuristic approach to solve static and dynamic worker allocation problem. An assembly environment in machine intensive cellular manufacturing system was considered for their research. They developed a methodology for static worker allocation for cells in which a single variety of product was produced and extended by solving dynamic variety problem in which multiple variety of products were manufactured. The objective was to balance the work load among cells and simultaneously minimize makespan. However they did not address the problem of detailed worker allocation within the cell and assumed that all the workers performed at same levels. Nakade and Ohno (1999) proposed a heuristic which optimally selected the minimum number of workers which minimized overall cycle time thereby meeting the demand in a U-Shaped production line. The workers were assumed to have same abilities and deterministic processing and walking times.

In all the researches discussed above, it was assumed that all the workers possess equal productivity. However, in a real world scenario, variability in worker productivity is predominant. Since, the worker productivity was assumed to be equal, a detailed worker-to-task allocation was not performed in sole profile allocation.

## 2.5 Multi Profile Allocation

In multi profile allocation, worker differences in terms of productivity were considered. Worker profile differences were modeled either based on multiple skill levels or individual workers. In profile differences based on multiple skill levels, it was assumed that each skill had a different skill level and the workers within each skill level possess equal productivity. Worker profile difference based on individual workers was a more realistic assumption.

### 2.5.1 Formulations in Multi Profile allocation

Similar to sole profile allocation, several formulations for multi profile worker allocation problems were formulated and are shown in Table 1.2.

Askin and Huang (2001) developed a mixed integer, goal programming model to form worker teams based on psychological, organizational and technical factors. Worker teams were assigned to cells and further workers were assigned to tasks in the cells. Training schedule for the workers was also an output of the model developed. Mixed integer model developed was solved in various solution methods like greedy search heuristic, filtered beam search heuristic and simulated annealing technique. Filtered beam search heuristic was capable of producing near-optimal solution with a reasonable computational time. The performance of different algorithms with respect to lower bound is shown in Figure 2.1.

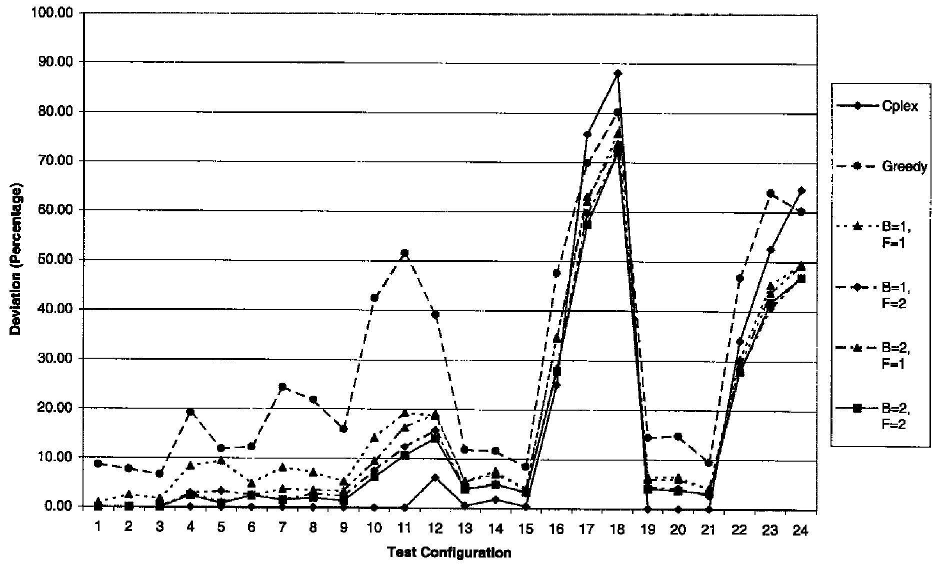


**Figure 2.1 Performance of algorithms (Askin and Huang, 2001)**

Norman et al., (2002) modeled a mixed integer program which assigned workers to tasks in a cellular manufacturing environment considering both technical and human skills. Skills of workers were categorized as different skill levels and individual worker profiles within the same skill levels were considered to be the same. Mixed integer model developed also decides on training schedule for workers to increase their skill levels.

Fitzpatrick and Askin (2005) formed multiple worker teams with multifunctional skill requirements for cellular manufacturing. The technical and inter personnel skills of workers were considered to form worker teams. The objective was to minimize the deviation from optimal klobe indices across all the worker teams. A heuristic was also proposed to solve and evaluate the same problem.

Chaves, Insa, and Lorena (2007) modeled an integer programming formulation for assembly line worker assignment and balancing problem (ALWABP) in sheltered work centers. The model assumed that individual workers have different deterministic processing time values due to inherent variability between workers. Objective of the IP model was to minimize the cycle time by assigning task to station and workers to task simultaneously. Clustering search approach was used to solve the IP model. Miralles, Garcia, Andres, and Cardos (2008) extended the previous work by Chaves, Insa, and Lorena (2007) by providing different solution methodology using branch and bound algorithm.

Suer and Tummaluri (2008) developed a three stage operator allocation procedure for worker allocation in labor intensive manufacturing. The first stage was to generate alternative worker staffing level configurations for all the products. The second stage was to simultaneously find optimal number of workers in cells and cell loading by assigning parts to cells. Two kinds of heuristics were proposed in the third step, namely, Max and Max Min to assign workers to task. They had assumed that the processing time of individual workers were different deterministic variables depending on the skill levels they possess. Operator skill levels were subject to change based on learning and forgetting nature which was determined by the number of times an operator was allotted to a particular operation.

Non-Linear Integer programming was also an extensive technique used to solve multi profile operator allocation problem. Aryanezhad, Deljoo, and Al-e-hasheem (2009) formulated a non linear integer program for Simultaneous Dynamic Cell formation and Worker assignment Problem (SDCWP). Workers ability was classified as different skill levels. The model simultaneously decided the number of machines to be purchased, relocated (or) to be removed in each cell, number of workers to be hired, fired, trained, and allocation of workers to machines for each period. Heimerl and Kolish (2009) modeled a non-linear integer program for assigning multi-skilled workers to tasks considering the worker learning, forgetting and company skill level target. Problem was solved using COIN-OR's Ipopt.

McDonald, Ellis, Aken, and Koelling (2009) proposed a mathematical model to assign cross trained workers to tasks in a lean manufacturing cell. The objective of the model was to minimize the net present cost. It was assumed that the processing time and quality level of workers were different deterministic values based on current skill depth level of workers. The output of model also identified training requirements and job rotation for workers.

Yaakob and Watada (2009) developed a methodology for worker assignment in cellular manufacturing using particle swarm optimization technique. The research considered multi function workers and evaluated relationship between workers and workers to task. Evaluation between workers was performed considering social, performance and mental factors.

Heuristic procedures were developed to reduce the complexity in mathematical models when the problem size increased. Nembhard (2001) developed a heuristic approach for worker allocation based on individual worker learning profiles. Worker–task allocation was examined for long and short production runs and it was observed that the learning profile based allocation increases productivity of the system. Simulation was done obtaining individual worker profile empirical data of learning/forgetting from the industry. Fowler, Wirojanagud, and Gel (2008) developed a mixed integer program for making decisions on hiring, training and firing workers. Individual workers were assumed to be different based on their general cognitive ability. The objective was to minimize the total cost due to hiring, training, firing and missed production costs over multiple periods. A genetic algorithm and two LP based heuristics were presented to solve the problem.

## 2.6 Research Motivation

Although some research works in worker allocation problem considered individual worker profiles, they assumed deterministic processing time and quality level for workers. In real world scenarios, uncertainty is predominant in worker processing time and quality level. A sample time study data obtained from local aircraft industry is shown in Table 1.3. Increase in uncertainty in worker processing time and quality level increases the risk due to delay in delivery of products and risk due to bad quality of products. Thus, the worker allocation methods in the literature are not suitable enough to address uncertain manufacturing environments.

In the following chapters, a risk based worker allocation methodology is proposed for single task per station balanced, single task per station unbalanced and multiple tasks per station production line scenarios. A methodology to determine optimal number of stations in an uncertain environment is also presented.

# CHAPTER 3

# RISK BASED WORKER ALLOCATION METHOD

## 3.1 Introduction

Manufacturing can be classified into machine intensive manufacturing and labor intensive manufacturing. In labor intensive manufacturing, increase in performance can be obtained by an optimal worker allocation method. In most worker allocation methods available in literature, allocation of workers to processes is carried out by assuming a deterministic processing time and quality level of workers. However, uncertainty in worker processing time and quality level is common in today's manufacturing industries. A change in processing time or quality of any worker would result in a modified relationship between the worker and the process, thus affecting the optimal worker allocation.

When there is an uncertainty associated with the worker processing time and quality, it is essential to assess and quantify the risk associated with the workers. According to Modarres, 2006, risk is defined as "a measure of the potential loss occurring due to natural or human activities". In the context of worker allocation, risk can be defined as the potential loss due to delay in process or due to the bad quality of the product. Thus, the objective of this research is to model and solve an efficient worker allocation method, which takes risk due to processing time and quality level of worker into account. A single product dedicated labor intensive manufacturing line was considered in this research.

## 3.2 Notations

The notations used in risk based worker allocation in balanced and un-balanced production line are follows:

$P_{ij}$ Processing time of worker i for process j

$P_{\sigma ij}$ Standard deviation in processing time of labor i for process j

$Q_{ij}$ Quality level of worker i for process j

$Q_{\sigma ij}$ Standard deviation in quality level of worker i for process j

$SP_j$ Standard processing time for process j

$SQ_j$ Standard quality level for process j

$DP_j$ Delay penalty for process j

$QP_j$ Quality penalty for process j

$\Pr_j$ Profit associated with process j

$Qr_j$ Quality cost due to bad quality in process j

$C_{ij}$ Cost of worker i for process j

$D$ Demand for the period

$T$ Time interval

$M$ Max number of workers allowed

$L_{ij}$ Worker i for process j (Binary variable)

$\mathrm{N_r}$ Number of replications

## 3.3 Worker Allocation Procedure

Worker allocation is defined as 'assigning the best worker' to the process from a pool of prequalified workers. Worker allocation can be performed in three kinds of manufacturing lines:

1) Single task per station balanced line
2) Single task per station unbalanced line
3) Multiple tasks per station

### 3.3.1 Worker Allocation in Single Task per Station Balanced Line

A line is said to be "balanced" when the cycle time associated with all processes are the same. There are various kinds of uncertainties associated with worker allocation, such as uncertainty in worker processing time, quality level, part arrival pattern, and demand. Among these, uncertainty in worker processing time and quality level of worker is more common and dominant. Before the uncertainty based model is developed, a deterministic model is developed. This is necessary to have a benchmark to validate against risk based worker allocation model. Thus, worker allocation in balanced line could be performed in two ways:

1) Worker allocation without considering uncertainty
2) Worker allocation considering uncertainty

#### 3.3.1.1 Worker Allocation without Considering Uncertainty

Worker allocation can be performed considering expected values for the processing time and quality level, while not taking into consideration the uncertainty associated with the workers. Thus, the objective for worker allocation without considering uncertainty was to maximize the profit by allocating the best worker to the process based on their deterministic processing time and quality level. A methodology for worker allocation without considering uncertainty is shown in the following section.

The assumptions for worker allocation without considering uncertainties are listed below:

- Expected processing time of each worker is known
- Expected quality level of each worker is known
- A worker can only be allocated to a single process
- At least one worker should be allotted to a process

#### 3.3.1.1.1 Model Input Requirements

The proposed worker allocation model for balanced line without considering uncertainty requires the following input parameters for testing the proposed approach:

1) Expected value of operation times of all workers
2) Expected value of quality level of all workers
3) Cost of workers for each process
4) Total number of processes required for the product
5) Standard time for each process
6) Standard quality level for each process
7) Revenue per process of the product
8) Quality loss per process of the product
9) Maximum number of workers in the line
10) Maximum number of workers per process

#### 3.3.1.1.2 Deterministic Mathematical Model

A Non Linear Integer Programming (NLIP) model was developed to find optimal worker allocation. The objective function and constraints are explained as follows;

*Objective function:*

$$Max\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\frac{T}{P_{ij}}\right]*\Pr_j*L_{ij}-\left(1-\frac{Q_{ij}}{100}\right)*\left[\frac{T}{P_{ij}}\right]*Qr_j*L_{ij}-\left[C_{ij}*\frac{T}{60}\right]*L_{ij} \quad (3.1)$$

The objective was to maximize the profit by allotting the best worker i, $i=1, 2, 3\ldots m$ to process j, $j=1, 2, 3\ldots n$ and is given in Equation (3.1). The first term in the objective function calculates the profit by multiplying total number of parts produced and revenue per part. The second term subtracts the loss due to bad quality from the profit, which is the product of total number of defective parts produced and their associated quality loss. The third term subtracts the cost of workers from the profit.

$\Pr_j$ is the profit associated with value added activities in process j. $Qr_j$ is the loss incurred due to bad quality in process j. In this research, quality level of a worker is defined as the number of products produced within the control limits of the respective process per 100 products and is given in Equation 3.2.

$$Quality\ level\ of\ process\ j = \frac{\text{No. of products within control limits in process j}}{100} \quad (3.2)$$

*Model Constraints:*

$$\sum_{j=1}^{n} L_{ij} \le 1 \qquad \forall i \in I \quad (3.3)$$

$$\sum_{i=1}^{m} L_{ij} \ge 1 \qquad \forall j \in J \quad (3.4)$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} L_{ij} \le M \quad (3.5)$$

Constraint (3.3) ensures that each worker i, is assigned to only one process. Constraint (3.4) assures that process j, has at least one worker. Constraint (3.5) places an upper bound for the total number of workers (M) allotted to the line.

The Non Linear Integer Programming (NLIP) model is solved using LINGO optimizer 12.0 software. Output from the LINGO software provides the optimal worker allocation. To demonstrate the worker allocation model, a sample case study is shown in the following section.

#### 3.3.1.1.3 Case Study–3-Process-6-Worker (Deterministic)

A balanced line with 3 processes and 6 workers is used as a sample case study to illustrate the proposed worker allocation model for the deterministic case. Standard processing time of all three processes was assumed to be 9 minutes.

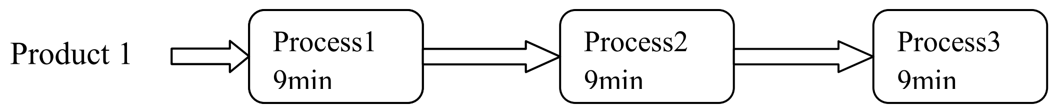


**Figure 3.1 Case Study-3-Process-6-Worker**

The processing time of all the workers for 3-process-6-worker case study are shown in Table 3.1. Expected values of worker processing time were obtained from previous historical data.

**Table 3.1 Processing Time (min) Chart for 3-Process-6-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 |
|---|---|---|---|
| Worker 1 | 8.8 | | 9.2 |
| Worker 2 | 8.9 | 8.9 | |
| Worker 3 | | 8.8 | 8.9 |
| Worker 4 | 9.0 | | 8.9 |
| Worker 5 | 9.1 | 9.1 | |
| Worker 6 | | 8.9 | 8.8 |

The quality levels of all the workers for 3-process-6-worker case study are shown in Table 3.2. Expected worker quality level is obtained from previous historical data.

**Table 3.2 Quality Level Chart for 3-Process-6-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 |
|---|---|---|---|
| **Worker 1** | 98.9 | | 99.2 |
| **Worker 2** | 99.3 | 98.7 | |
| **Worker 3** | | 98.3 | 98.8 |
| **Worker 4** | 99.0 | | 99.1 |
| **Worker 5** | 98.8 | 98.3 | |
| **Worker 6** | | 99.6 | 99.2 |

The costs of all workers associated with their respective processes for 3-process-6-worker case study are shown in Table 3.3.

**Table 3.3 Worker Cost ($) Chart for 3-Process-6-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 |
|---|---|---|---|
| **Worker 1** | 10.00/hr | | 9.37/hr |
| **Worker 2** | 8.75/hr | 10.00/hr | |
| **Worker 3** | | 11.80/hr | 10.62/hr |
| **Worker 4** | 10.62/hr | | 9.37/hr |
| **Worker 5** | 11.87/hr | 10.00/hr | |
| **Worker 6** | | 11.87/hr | 13.75/hr |

The $\mathrm{Pr}_j$ values for 3-process-6–worker-case study are given below:

Profit due to process 1 $\left(\mathrm{Pr}_1\right) = \$36$

Profit due to process 2 $\left(\mathrm{Pr}_2\right) = \$33$

Profit due to process 3 $\left(\mathrm{Pr}_3\right) = \$37.5$

The $Qr_j$ values for 3-process-6-worker case study are given as follows:

Quality loss due to process1 $(Q\text{r}_1) = \$33$

Quality loss due to process2 $(Q\text{r}_2) = \$36$

Quality loss due to process3 $(Q\text{r}_3) = \$39$

Maximum number of workers allowed in the 3-process-6-machine case study is set to 5 and time interval at 2880 minutes.

*LINGO model for 3-process-6-worker case study:*

Once the input variables are obtained, Non Linear Integer Programming (NLIP) model is solved using LINGO Optimizer 12.0 and the results are interpreted below.

*Objective function:*

$$Max\sum_{i=1}^{6}\sum_{j=1}^{3}\left[\frac{2880}{P_{ij}}\right]*\text{Pr}_j*L_{ij}-\left(1-\frac{Q_{ij}}{100}\right)*\left[\frac{2880}{P_{ij}}\right]*Qr_j*L_{ij}-\left[C_{ij}*\frac{2880}{60}\right]*L_{ij}$$

*Subject to:*

$$\sum_{j=1}^{3}L_{ij}\leq 1 \qquad \forall i=1,2,3,\ldots 6$$

$$\sum_{i=1}^{6}L_{ij}\geq 1 \qquad \forall j=1,2,3$$

$$\sum_{i=1}^{6}\sum_{j=1}^{3}L_{ij}\leq 5$$

The Worker allocation output of LINGO Optimizer 12.0 is shown below in Table 3.4.

**Table 3.4 Results - 3-Process-6-Worker Case Study**

| PROCESS | WORKER |
|---|---|
| Process 1 | Worker 1 |
| Process 2 | Worker 3 |
| Process 3 | Worker 6 |

#### 3.3.1.2 Worker Allocation Considering Uncertainty

Although most of the worker allocation methods in past literature were based on expected values of worker processing time and quality level, uncertainty is more common in real world scenario. Thus, the objective for worker allocation considering uncertainty is to allocate the best worker to the process, thereby minimizing the overall risk in the production line. A flexible approach for worker allocation considering uncertainties in processing time and quality level is developed and shown in the following sections.

##### 3.3.1.2.1 Worker Allocation Risk Assessment Procedure

In this research, influence of processing time and quality risk in worker allocation is assessed using risk assessment methodology. Modarres (2006) defined risk as;

$$RISK = \sum_{i} u_i c_i$$

Where, $u_i$ is the probability of event $i$ occurring and $c_i$ is the consequence associated with the event. For worker allocation, probability of occurrence is defined as the probability of delay in processing time and probability of bad quality. The consequence can be viewed as delay and quality penalty respectively. Thus, in order to overcome the problems associated with existing methodologies for worker allocation, risk based worker allocation methodology for balanced line is proposed in the following sections.

The assumptions for worker allocation in balanced line considering uncertainties are listed below:

- Processing time of each worker is stochastic and follows a normal distribution
- Quality of each worker is stochastic and follows a normal distribution
- A worker can only be allotted to a single process
- Minimum number of worker per process is one

- Demand is deterministic and known for each period
- Workers are constrained to perform certain operations

**3.3.1.2.2 Model Input Requirements**

The proposed risk based worker allocation model requires following input parameters for testing the proposed approach:

1) Operation times of all workers
2) Quality level of all workers
3) Cost of workers for each process
4) Total number of processes required for the product
5) Delay penalty associated with each process
6) Quality penalty associated with each process
7) Standard time for each process
8) Standard quality level for each process
9) Maximum number of workers in the line
10) Maximum number of workers per process

**3.3.1.2.3 Risk Based Mathematical Model**

A Non Linear Integer Programming (NLIP) model is developed for optimal risk based worker allocation, which, given an uncertain processing time and quality level of workers, allocates workers to the process thereby reducing the overall risk in the line. The objective function and the constraints are explained as follows;

*Objective Function*

$$Min\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\left[1-1/2\left[1+erf\left(\frac{SP_j-P_{ij}}{P_{\sigma ij}\sqrt{2}}\right)\right]\right]*DP_j*L_{ij}\right]+\left[\left[1/2\left[1+erf\left(\frac{SQ_j-Q_{ij}}{Q_{\sigma ij}\sqrt{2}}\right)\right]\right]*QP_j*L_{ij}\right]+\left[\left[C_{ij}*\frac{T}{60}\right]*L_{ij}\right] \quad (3.6)$$

The objective is to minimize the total risk due to worker processing time and quality level by allocating worker *i*=1,2,3…m to process *j*=1,2,3…n which is given in Equation (3.6). The first term in the objective function minimizes the risk due to worker processing time, which is the product of probability in exceeding standard processing time and delay penalty. Second term minimizes the risk due to worker quality level, which is the product of probability of quality below standard quality level and quality penalty. Third part minimizes the cost of hiring workers.

*Model Constraints*

$$\sum_{j=1}^{n} L_{ij} \leq 1 \qquad \forall i \in I \tag{3.7}$$

$$\sum_{i=1}^{m} L_{ij} \geq 1 \qquad \forall j \in J \tag{3.8}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} L_{ij} \leq M \tag{3.9}$$

In the above set of constraints, constraint (3.7) ensures that each worker i, is assigned to only one process. Constraint (3.8) assures that at least one worker is assigned to process j. Constraint (3.9) places an upper bound for the total number of workers (M) allotted to the line.

Delay penalty $(DP_j)$ is the penalty incurred if the processing time of certain process exceeds the standard processing time. Delay penalty will change as the criticality of the process changes. When a highly critical process is delayed, cost incurred is comparatively more than a low critical process. Quality penalty $(QP_j)$ is the penalty incurred if the quality level of certain process is below the standard quality level. Quality penalty will increase linearly when moving towards downstream of the line, since every process in downstream adds value to the product.

The Non Linear Integer Programming (NLIP) model is solved using LINGO optimizer 12.0 software. Output from the LINGO software provides the optimal worker allocation. To demonstrate the risk based worker allocation model, two sample case studies are given in the following sections.

**3.3.1.2.4 Case Study – 3-Process-6-Worker (Risk Based)**

The same 3-process-6-worker case study (Figure 3.1) is used to illustrate the proposed risk based worker allocation model. Standard processing time of all three processes is assumed to be 9 minutes.

It is assumed that the worker processing time and quality level is stochastic and follows normal distribution due to uncertainties in processing time and quality. The modified processing time values for 3-process-6-worker case study are given in Table 3.5.

**Table 3.5 Processing Time (min) Chart for 3-Process-6-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 |
|---|---|---|---|
| **Worker 1** | $8.8 \pm 1.05$ | | $9.2 \pm 0.39$ |
| **Worker 2** | $8.9 \pm 0.33$ | $8.9 \pm 0.66$ | |
| **Worker 3** | | $8.8 \pm 1.11$ | $8.9 \pm 0.68$ |
| **Worker 4** | $9.0 \pm 0.87$ | | $8.9 \pm 0.23$ |
| **Worker 5** | $9.1 \pm 0.48$ | $9.1 \pm 0.18$ | |
| **Worker 6** | | $8.9 \pm 0.36$ | $8.8 \pm 0.98$ |

The modified quality level values for 3-process-6-worker case study are given in Table 3.6.

**Table 3.6 Quality Level Chart for 3-Process-6-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 |
|---|---|---|---|
| **Worker 1** | $98.9 \pm 0.22$ | | $99.2 \pm 0.11$ |
| **Worker 2** | $99.3 \pm 0.14$ | $98.7 \pm 0.21$ | |
| **Worker 3** | | $98.3 \pm 0.54$ | $98.8 \pm 0.25$ |
| **Worker 4** | $99.0 \pm 0.08$ | | $99.1 \pm 0.08$ |
| **Worker 5** | $98.8 \pm 0.19$ | $99.3 \pm 0.18$ | |
| **Worker 6** | | $98.6 \pm 0.26$ | $99.2 \pm 0.34$ |

The costs of all workers associated with their respective processes for 3-process-6-worker case study are given in Table 3.3.

Delay penalties $\left(DP_j\right)$ for 3-process-6-worker case study are given below:

Delay penalty for process 1 $\left(DP_1\right)$: \$450

Delay penalty for process 2 $\left(DP_2\right)$: \$500

Delay penalty for process 3 $\left(DP_3\right)$: \$320

Quality penalties $\left(QP_j\right)$ for 3-process-6-worker case study are given below:

Quality penalty for process 1 $\left(QP_1\right)$: \$300

Quality penalty for process 2 $\left(QP_2\right)$: \$400

Quality penalty for process 3 $\left(QP_3\right)$: \$500

Maximum number of workers allowed in the 3-process-6-worker case study is set to 5 and standard quality level of all the processes is set at 99%.

*LINGO model for 3-process-6-worker case study:*

Once the input variables are obtained, Non Linear Integer Programming (NLIP) model was solved using LINGO Optimizer 12.0 and the results are interpreted below

*Objective function:*

$$Min\sum_{i=1}^{6}\sum_{j=1}^{3}\left[\left[1-1/2\left[1+erf\left(\frac{SP_j-P_{ij}}{P_{\sigma ij}\sqrt{2}}\right)\right]\right]*DP_j*L_{ij}\right]+\left[\left[1/2\left[1+erf\left(\frac{SQ_j-Q_{ij}}{Q_{\sigma ij}\sqrt{2}}\right)\right]\right]*QP_j*L_{ij}\right]+\left[\left[C_{ij}*\frac{2880}{60}\right]*L_{ij}\right]$$

*Subject to:*

$$\sum_{j=1}^{3}L_{ij}\leq 1 \qquad \forall i=1,2,3,\ldots 6$$

$$\sum_{i=1}^{6} L_{ij} \geq 1 \quad \forall j = 1,2,3$$

$$\sum_{i=1}^{6} \sum_{j=1}^{3} L_{ij} \leq 5$$

The worker allocation output of LINGO Optimizer 12.0 is shown below in Table 3.7.

**Table 3.7 Results –3-Process-6-Worker Case study**

| PROCESS | WORKER |
|---|---|
| Process 1 | Worker 2 |
| Process 2 | Worker 6 |
| Process 3 | Worker 4 |

### 3.3.1.2.5 Case Study – 4-Process-8-Worker

To further test the consistency of the proposed approach, a 4-process-8-worker case study is selected (Figure 3.2). The standard processing time for all the processes is assumed to be 14 minutes.

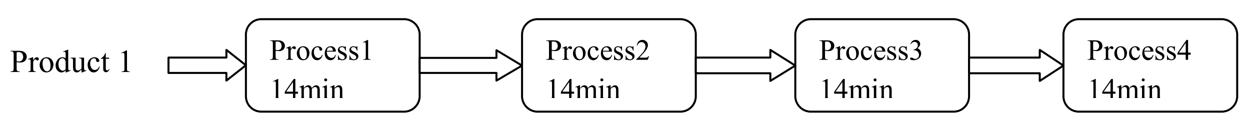


**Figure 3.2 Case Study - 4-Process-8-Workers**

The processing time of all the workers for 4-process-8-worker case study are shown in Table 3.8.

**Table 3.8 Processing Time (min) Chart for 4-Process-8-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 |
|---|---|---|---|---|
| **Worker 1** | $13.7 \pm 0.62$ | | $13.4 \pm 0.85$ | |
| **Worker 2** | | $14.2 \pm 0.58$ | | $13.6 \pm 0.66$ |
| **Worker 3** | $13.5 \pm 0.53$ | | $13.3 \pm 0.92$ | |
| **Worker 4** | | $13.8 \pm 0.76$ | | $13.7 \pm 0.48$ |
| **Worker 5** | $13.8 \pm 0.44$ | | $13.5 \pm 0.63$ | |
| **Worker 6** | | $14.0 \pm 0.56$ | | $13.9 \pm 0.42$ |
| **Worker 7** | $14.1 \pm 0.35$ | | $13.2 \pm 1.00$ | |
| **Worker 8** | | $13.9 \pm 0.82$ | | $13.8 \pm 0.53$ |

The quality levels of all the workers for 4-process-8-worker case study are shown in Table 3.9.

**Table 3.9 Quality Level Chart for 4-Process-8-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 |
|---|---|---|---|---|
| **Worker 1** | $98.6 \pm 0.16$ | | $99.1 \pm 0.26$ | |
| **Worker 2** | | $98.0 \pm 0.26$ | | $98.6 \pm 0.23$ |
| **Worker 3** | $99.0 \pm 0.33$ | | $98.4 \pm 0.13$ | |
| **Worker 4** | | $98.3 \pm 0.20$ | | $98.0 \pm 0.30$ |
| **Worker 5** | $98.5 \pm 0.20$ | | $99.2 \pm 0.16$ | |
| **Worker 6** | | $98.7 \pm 0.26$ | | $97.5 \pm 0.83$ |
| **Worker 7** | $97.0 \pm 1.00$ | | $98.7 \pm 0.28$ | |
| **Worker 8** | | $98.2 \pm 0.42$ | | $98.8 \pm 0.28$ |

The costs of all workers associated with their respective processes for 4-process-8-worker case study are given in Table 3.10.

**Table 3.10 Worker Cost ($) Chart for 4-Process-8-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 |
|---|---|---|---|---|
| **Worker 1** | 15/hr | | 8/hr | |
| **Worker 2** | | 13/hr | | 10/hr |
| **Worker 3** | 12/hr | | 14/hr | |
| **Worker 4** | | 10/hr | | 12/hr |
| **Worker 5** | 10/hr | | 15/hr | |
| **Worker 6** | | 13/hr | | 14/hr |
| **Worker 7** | 17/hr | | 16/hr | |
| **Worker 8** | | 11/hr | | 9/hr |

Delay penalties $(DP_j)$ for 4-process-8-worker case study are given below:

Delay penalty for process 1 $(DP_1)$: $650

Delay penalty for process 2 $(DP_2)$: $625

Delay penalty for process 3 $(DP_3)$: $725

Delay penalty for process 4 $(DP_4)$: $750

Quality penalties $(QP_j)$ for 4-process-8-worker case study are given below:

Quality penalty for process 1 $(QP_1)$: $550

Quality penalty for process 2 $(QP_2)$: $600

Quality penalty for process 3 $(QP_3)$: $650

Quality penalty for process 4 $(QP_4)$: $700

Maximum number of workers allowed in the 4-process-8-worker case study is set to 7 and standard quality level of all the processes is 98.5%.

*LINGO model for 4-process-8-worker case study:*

Once the input variables are obtained, Non Linear Integer Programming (NLIP) model was solved using LINGO Optimizer 12.0 and the results are interpreted below:

*Objective function:*

$$Min\sum_{i=1}^{8}\sum_{j=1}^{4}\left[\left[1-1/2\left[1+erf\left(\frac{SP_j-P_{ij}}{P_{\sigma ij}\sqrt{2}}\right)\right]\right]*DP_j*L_{ij}\right]+\left[\left[1/2\left[1+erf\left(\frac{SQ_j-Q_{ij}}{Q_{\sigma ij}\sqrt{2}}\right)\right]\right]*QP_j*L_{ij}\right]+\left[\left[C_{ij}*\frac{2880}{60}\right]*L_{ij}\right]$$

*Subject to:*

$$\sum_{j=1}^{4}L_{ij}\leq 1 \qquad \forall i=1,2,3,\ldots 6$$

$$\sum_{i=1}^{8}L_{ij}\geq 1 \qquad \forall j=1,2,3$$

$$\sum_{i=1}^{8}\sum_{j=1}^{4}L_{ij}\leq 7$$

The worker allocation output of LINGO Optimizer 12.0 is shown Table 3.11.

**Table 3.11 Results – 4-Process-8-Worker Case Study**

| PROCESS | WORKER |
|---|---|
| Process 1 | Worker 3 |
| Process 2 | Worker 6 |
| Process 3 | Worker 1 |
| Process 4 | Worker 8 |

### 3.3.1.3 Validation

Worker allocation results obtained for 3-process-6-worker case study by deterministic method and risk based methodology for single task per station balanced line are compared for validation purposes. Simulation is selected as a tool to compare the methods.

In context with production lines, costs could be classified as,

1) Internal costs

2) External costs

Internal costs are incurred if processing time of a certain process exceeds the standard processing time, such as cost for re-scheduling jobs. External costs are penalty incurred if the cycle time of a product exceeds the desired cycle time (or) costs due to poor quality of the product.

Since the processing time and quality level of workers are assumed to follow a normal distribution, a single run may not be sufficient to eliminate randomness in output. Hence, the model is replicated several times. Number of replications is calculated using Equation (3.10). Where 'α' denotes the confidence interval, 'σ' denotes the standard deviation and 'h' denotes the accuracy.

$$N_r = t^2_{\frac{\alpha}{2}, n-1} \frac{\sigma^2}{h^2} \qquad (3.10)$$

Simulation is carried out using discrete event simulation software QUEST with the worker allocation results obtained using deterministic and risk based worker allocation techniques and the outputs were compared. Simulation is run for 2880 minutes. The simulation model developed using worker allocation results from the deterministic worker allocation model is replicated for 24 runs and risk based worker allocation model is replicated for 12 runs. It is assumed that internal cost is \$20/min, external cost associated with cycle time is \$100/min and external cost associated with bad quality is \$500/part. The results of the simulation are shown in Tables 3.12 and 3.13.

**Table 3.12 Cost Comparison Between Deterministic and Risk Based Method**

| | Deterministic method | | Risk based Method | | Percent improvement |
|---|---|---|---|---|---|
| **Throughput** | 286.58 | | 303.41 | | 5.8% |
| | **Internal Cost** | **External Cost** | **Internal Cost** | **External Cost** | - |
| **Time exceeded by process 1 (min)** | 93.02x20 =$1,860.4 | - | 25.84x20 =$516.8 | - | 72.22% |
| **Time exceeded by process 2 (min)** | 98.38x20 =$1,867.6 | - | 28.83x20 =$576.6 | - | 70.69% |
| **Time exceeded by process 3 (min)** | 80.43x20 =$1,608.6 | - | 14.34x20 =$286.8 | - | 82.17% |
| **Time exceeded by cycle time (min)** | - | 330.80x100 =$33,080 | - | 121.74x100 =$12,174 | 63.19% |
| **Total bad quality parts** | - | 11.91x500 =$5,955 | - | 9.25x$500 =$4,625 | 22.33% |
| **TOTAL COST** | **$44,371.6** | | **$18,179.2** | | |

**Table 3.13 VA/NVA Comparison Between Deterministic and Risk Based Method**

| Process | Process 1 | Process 2 | Process 3 | Total |
|---|---|---|---|---|
| **Deterministic method** | | | | |
| **Value Added Time** | 2646.059 | 2606.869 | 2539.582 | 7792.51 |
| **Idle Time** | 0 | 149.456 | 340.418 | 489.874 |
| **Waiting Time** | 233.941 | 123.674 | 0 | 357.615 |
| **Non-Value Added Time** | 233.941 | 273.131 | 340.418 | 847.49 |
| **Risk based Method** | | | | |
| **Value Added Time** | 2804.605 | 2770.01 | 2732.069 | 8306.684 |
| **Idle Time** | 0 | 68.663 | 147.931 | 216.594 |
| **Waiting Time** | 75.394 | 43.77 | 0 | 119.164 |
| **Non-Value Added time** | 75.394 | 112.437 | 147.931 | 335.762 |

From the above tables, it can be observed that the risk based worker allocation outperforms deterministic worker allocation methodology. Throughput increased by 5.8% and simultaneously non-value added time is reduced by 60.38%. Figure 3.3 compares the value added and non-value added times of deterministic and risk based methodology.

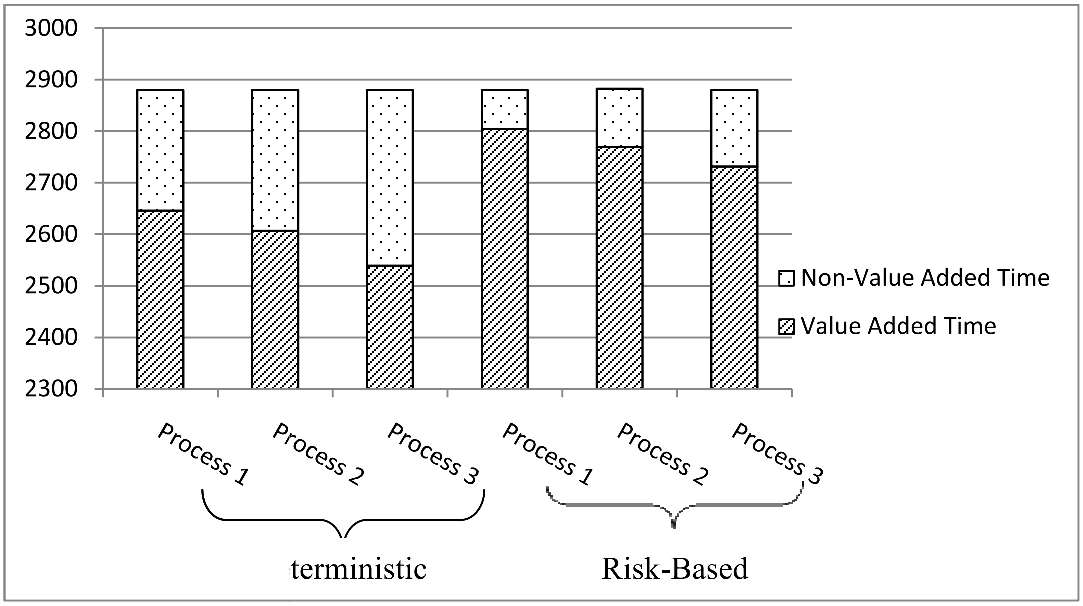


**Figure 3.3 VA/NVA Comparison Chart**

### 3.3.2 Worker Allocation in Single Task per Station Un-Balanced Line

A production line in which the sequential operations have different processing times is called an unbalanced line. An unbalanced production line can be balanced by allocation of multiple workers to the bottleneck processes. Thus in a single task per station unbalanced production line, proposed risk based worker allocation methodology allocates multiple workers to the bottleneck processes to meet the demand. An optimal risk based worker allocation methodology for unbalanced line is described in the following sections.

#### 3.3.2.1 Model Input Requirements

The proposed risk based worker allocation model for unbalanced line requires following input parameters for testing the proposed approach:

1) Operation times of all workers for their associated processes
2) Quality level of all workers for their associated processes

3) Cost of workers for each process respectively

4) Planning horizon

5) Demand of the product for given planning horizon

6) Total number of processes required for the product

7) Delay penalty associated with each process

8) Quality penalty associated with each process

9) Standard time for each process

10) Standard quality level for each process

11) Maximum number of workers in the line

12) Maximum number of workers per process

**3.3.2.2 Mathematical Model for Unbalanced Line**

A Non Linear Integer Program is developed to solve the risk based worker allocation model for unbalanced line, which, given an uncertain processing time and quality of workers, allocates the best workers to the processes. The objective function and the constraints in the proposed model is explained in the following section

*Objective function:*

$$Min\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\left[1-1/2\left[1+erf\left(\frac{SP_j-P_{ij}}{P_{\sigma ij}\sqrt{2}}\right)\right]\right]*DP_j*L_{ij}\right]+\left[\left[1/2\left[1+erf\left(\frac{SQ_j-Q_{ij}}{Q_{\sigma ij}\sqrt{2}}\right)\right]\right]*QP_j*L_{ij}\right]+\left[C_{ij}*\left(\frac{T}{60}\right)*L_{ij}\right] \quad (3.11)$$

The objective function minimizes the total risk due to worker processing time and quality level by allocating worker i, $i$=1,2,3…m to process j, $j$=1,2,3…n (Equation 3.11). The first part of the objective function minimizes the risk due to processing time of worker which is the product of probability in exceeding standard processing time and delay penalty. Second part minimizes the risk due to quality level of the worker which is the product of probability in quality of product below standard quality level and quality penalty. Third part includes the worker cost into objective function.

*Model constraints:*

$$\sum_{j=1}^{n} L_{ij} \leq 1 \qquad \forall i \in I \tag{3.12}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} L_{ij} \leq M \tag{3.13}$$

$$\left[ \sum_{i}^{m} \frac{(Q_{ij} - 3Q_{\sigma ij})}{(P_{ij} + 3P_{\sigma ij})} * L_{ij} \right] > \frac{D}{T} \quad \forall j \in J \tag{3.14}$$

$$DP_j = \frac{1}{\sum_{i=1}^{n} \left[ \frac{L_{ij}}{P_{ij}} \right] - \frac{D}{T}} * DP \quad \forall j \in J \tag{3.15}$$

In the above set of constraints, Equation (3.12) ensures that worker i, *i*=1, 2, 3,…n, can perform at most one process in a given time period. Equation (3.13) provides an upper bound for the total number of workers (M) allotted to the line. Equation (3.14) allots multiple workers to the processes and makes sure that the line is producing more than the demand. Demand constraint was met by ensuring that each process produces greater than the takt time of the line even in the worst case scenario. Equation (3.15) allots dynamic delay penalty values to the processes based on the idle time of the process, such that the busiest process will have more delay penalty in comparison with other processes.

The Non Linear Integer Programming (NLIP) model is solved using LINGO 12.0 optimizer for minimum risk after worker allocation. Output of LINGO optimizer gives the binary value for $L_{ij}$. if $L_{ij}$ equals 1 then it is inferred that worker *i* is allocated to process *j* and if equals 0 then it is inferred that worker *i* is not allocated to process *j*. In order to demonstrate the proposed

Non Linear Integer Programming (NLIP) model, two sample case studies are given in the following section.

**3.3.2.3 Case Study - 5-Process-10-Worker**

A case study with 5 processes and 10 workers in an unbalanced line is used to illustrate the proposed risk based worker allocation model. The standard processing times for all five processes are different as shown in Figure 3.4.

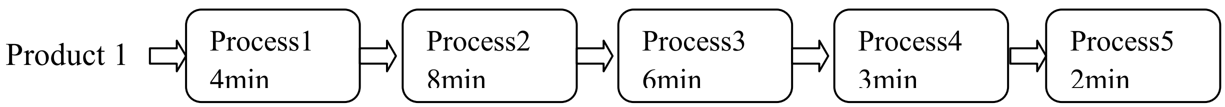


**Figure 3.4 Case Study - 5-Process-10-Workers**

It is assumed that the worker processing time follows normal distribution which is collected from previous historical values and workers are constrained to certain operations. Processing time chart for 5-process-10-workers case study are shown in Table 3.14.

**Table 3.14 Processing Time (min) Chart for 5-Process-10-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 | Operation 5 |
|---|---|---|---|---|---|
| **Worker 1** | 3.5±0.16 | | 6.2±0.1 | | |
| **Worker 2** | | 7.0±0.6 | | 3.0±0.16 | |
| **Worker 3** | 4.5±0.06 | | | | 1.5±0.33 |
| **Worker 4** | | 7.5±0.33 | | 3.5±0.16 | |
| **Worker 5** | | | 5.8±0.2 | | 2.0±0.2 |
| **Worker 6** | 4.2±0.6 | | 6.0±0.16 | | |
| **Worker 7** | | 6.8±0.16 | | 2.8±0.2 | |
| **Worker 8** | 4.0±0.13 | | | | 2.2±0.06 |
| **Worker 9** | | 6.5±0.26 | | 2.5±0.33 | |
| **Worker 10** | | | 6.5±0.16 | | 2.5±0.16 |

In this research, quality level of a worker is defined as the number of products produced within the control limits of the respective process per 100 products. It is assumed that the worker quality level follows normal distribution which was obtained from previous historical data. Quality level chart for 5-process-10-workers case study are shown in Table 3.15.

**Table 3.15 Quality Level Chart for 5-Process-10-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 | Operation 5 |
|---|---|---|---|---|---|
| **Worker 1** | $98 \pm 0.1$ | | $99 \pm 0.16$ | | |
| **Worker 2** | | $98 \pm 0.5$ | | $99 \pm 0.1$ | |
| **Worker 3** | $99 \pm 0.13$ | | | | $97 \pm 0.5$ |
| **Worker 4** | | $98.5 \pm 0.33$ | | $99.5 \pm 0.16$ | |
| **Worker 5** | | | $98 \pm 0.06$ | | $98 \pm 0.16$ |
| **Worker 6** | $98.5 \pm 0.1$ | | $98 \pm 0.16$ | | |
| **Worker 7** | | $99.5 \pm 0.1$ | | $98 \pm 0.13$ | |
| **Worker 8** | $98 \pm 0.16$ | | | | $98.5 \pm 0.33$ |
| **Worker 9** | | $99.5 \pm 0.13$ | | $98.6 \pm 0.13$ | |
| **Worker 10** | | | $98.8 \pm 0.16$ | | $99 \pm 0.33$ |

Demand for the product 1 in 5-process-10 worker case study is assumed to be 600 products and the associated planning horizon is 2880 minutes. The costs of all workers associated with their respective processes for 5-process-10-worker case study are shown in Table 3.16.

**Table 3.16 Worker Cost ($) Chart for 5-Process-10-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 | Operation 5 |
|---|---|---|---|---|---|
| **Worker 1** | 12.50/hr | | 13.33/hr | | |
| **Worker 2** | | 11.66/hr | | 15.00/hr | |
| **Worker 3** | 13.33/hr | | | | 14.16/hr |
| **Worker 4** | | 15.83/hr | | 12.50/hr | |
| **Worker 5** | | | 11.66/hr | | 15.00/hr |
| **Worker 6** | 14.16/hr | | 13.33/hr | | |
| **Worker 7** | | 18.33/hr | | 14.16/hr | |
| **Worker 8** | 12.50/hr | | | | 13.33/hr |
| **Worker 9** | | 18.33/hr | | 15.83/hr | |
| **Worker 10** | | | 13.33/hr | | 14.16/hr |

*Dynamic delay penalty:*

Delay penalty is the penalty incurred if the processing time of certain process exceeds the standard processing time. Delay penalty will change as the criticality of the process changes. Thus a dynamic delay penalty constraint is introduced which allots delay penalty based on the idle time of the process. The busiest process in the production line is allotted the highest delay penalty to make sure the best worker was allotted to the busiest process. The delay penalty calculations for the 5-process-10-worker case study are explained below:

Delay penalty for process 1 $= \dfrac{1}{\sum_{i=1}^{10}\left[\dfrac{L_{ij}}{P_{ij}}\right] - \dfrac{D}{T}} * DP \quad \forall j = 1$

$$= \frac{1}{\left[\frac{1}{3.5} + \frac{0}{4.5} + \frac{0}{4.2} + \frac{0}{4.0}\right] - \left[\frac{600}{2880}\right]} * 40$$

$= \$516.92$

Delay penalty for process 2 $$= \frac{1}{\sum_{i=1}^{10}\left[\frac{L_{ij}}{P_{ij}}\right] - \frac{D}{T}} * DP \quad \forall j = 2$$

$$= \frac{1}{\left[\frac{0}{7.0} + \frac{1}{7.5} + \frac{1}{6.8} + \frac{0}{6.5}\right] - \left[\frac{600}{2880}\right]} * 40$$

$= \$555.10$

Delay penalty for process 3 $$= \frac{1}{\sum_{i=1}^{10}\left[\frac{L_{ij}}{P_{ij}}\right] - \frac{D}{T}} * DP \quad \forall j = 3$$

$$= \frac{1}{\left[\frac{0}{6.2} + \frac{1}{5.8} + \frac{0}{6.0} + \frac{1}{6.5}\right] - \left[\frac{600}{2880}\right]} * 40$$

$= \$339.19$

Delay penalty for process 4 $$= \frac{1}{\sum_{i=1}^{10}\left[\frac{L_{ij}}{P_{ij}}\right] - \frac{D}{T}} * DP \quad \forall j = 4$$

$$= \frac{1}{\left[\frac{0}{3.0} + \frac{0}{3.5} + \frac{0}{2.8} + \frac{1}{2.5}\right] - \left[\frac{600}{2880}\right]} * 40$$

$= \$208.69$

Delay penalty for process 5 $= \dfrac{1}{\sum_{i=1}^{10}\left[\dfrac{L_{ij}}{P_{ij}}\right] - \dfrac{D}{T}} * DP \quad \forall j = 5$

$$= \frac{1}{\left[\frac{0}{1.5} + \frac{0}{2.0} + \frac{1}{2.2} + \frac{0}{2.5}\right] - \left[\frac{600}{2880}\right]} * 40$$

$= \$162.46$

Quality penalty is incurred if the quality level of certain process is below the standard quality level. Quality penalty will increase linearly when moving towards downstream of the line. Quality Penalties for 5-process-10-worker-case study is given below:

Quality penalty for process 1: $300

Quality penalty for process 2: $400

Quality penalty for process 3: $500

Quality penalty for process 4: $600

Quality penalty for process 5: $700

Maximum number of workers allowed in the 5-process-10-worker case study is set to 9 and Standard quality level of all the processes is 98%.

*LINGO model for 5-process-10-machine case:*

Once the input variables were decided, Non Linear Integer Programming (NLIP) model was solved using LINGO Optimizer 12.0 and the results are interpreted below:

Objective function:

$$Min \sum_{i=1}^{10} \sum_{j=1}^{5} \left[ \left[ 1 - 1/2 \left[ 1 + erf\left( \frac{SP_j - P_{ij}}{P_{\sigma ij}\sqrt{2}} \right) \right] \right] * DP_j * L_{ij} \right] + \left[ \left[ 1/2 \left[ 1 + erf\left( \frac{SQ_j - Q_{ij}}{Q_{\sigma ij}\sqrt{2}} \right) \right] \right] * QP_j * L_{ij} \right] + \left[ C_{ij} * L_{ij} \right]$$

*Subject to:*

$$\sum_{j=1}^{5} L_{ij} \leq 1 \qquad \forall i = 1,2,3.....10$$

$$\sum_{i=1}^{10}\sum_{j=1}^{5} L_{ij} \leq 9$$

$$\left[\sum_{i}^{10} \frac{(Q_{ij} - 3Q_{\sigma ij})}{(P_{ij} + 3P_{\sigma ij})} * L_{ij}\right] \geq \frac{600}{2880} \qquad \forall j = 1,2,....5$$

$$DP_j = \frac{1}{\sum_{i=1}^{10}\left[\frac{L_{ij}}{P_{ij}}\right] - \frac{600}{2800}} * 40 \quad \forall j = 1,2,...5$$

The worker allocation output obtained from LINGO Optimizer 12.0 is shown in Table 3.17.

**Table 3.17 Results – 5-Process-10-Worker Case Study**

| PROCESS | WORKER |
|---|---|
| Process 1 | Worker 1 |
| Process 2 | Worker 2, Worker 4 |
| Process 3 | Worker 5, Worker 10 |
| Process 4 | Worker 9 |
| Process 5 | Worker 8 |

The results obtained from the LINGO model for 5-process-10-worker case study is simulated using QUEST software. Simulation is replicated for16 times and the average throughput is found to be 622.4 products.

### 3.3.2.4 Case Study - 5-Process-15-Worker

For further testing of the methodology explained above for unbalanced line, another case study has been tested in the following section. An unbalanced line with 5 processes and 15 workers was considered as case study to further test the proposed methodology for risk based worker allocation in unbalanced production line.

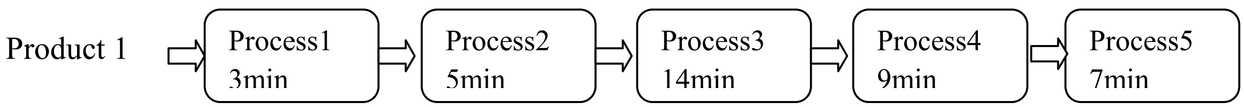


**Figure 3.5 Case Study -5-Process-15-Worker**

The processing time chart for 5-process-15-workers case study is given in Table 3.18. The quality level chart for 5-process-15-workers case study is given in Table 3.19. Demand for the product 1 in 5-process-15-worker case study is assumed to be 500 products and the associated planning horizon was 2880 minutes. The cost of all workers associated with their respective processes for 5-process-15-worker case study is given in Table 3.20.

**Table 3.18 Processing Time (min) Chart for 5-Process-15-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 | Operation 5 |
|---|---|---|---|---|---|
| **Worker 1** | $2.8 \pm 0.17$ | | $13.5 \pm 0.33$ | | |
| **Worker 2** | | $5.1 \pm 0.07$ | | $8.9 \pm 0.13$ | |
| **Worker 3** | $2.9 \pm 0.13$ | | | | $6.6 \pm 0.20$ |
| **Worker 4** | | $4.8 \pm 0.20$ | | $8.8 \pm 0.20$ | |
| **Worker 5** | | | $13.8 \pm 0.23$ | | $6.8 \pm 0.23$ |
| **Worker 6** | $2.5 \pm 0.33$ | | $14.1 \pm 0.17$ | | |
| **Worker 7** | | $4.6 \pm 0.27$ | | $9.1 \pm 0.07$ | |
| **Worker 8** | $3.0 \pm 0.10$ | | | | $7.2 \pm 0.10$ |
| **Worker 9** | | $4.7 \pm 0.20$ | | $8.7 \pm 0.20$ | |
| **Worker 10** | | | $13.7 \pm 0.27$ | | $7.0 \pm 0.17$ |
| **Worker 11** | $3.1 \pm 0.10$ | | $13.9 \pm 0.23$ | | |
| **Worker 12** | | $5.0 \pm 0.20$ | | $8.8 \pm 0.23$ | |
| **Worker 13** | $2.7 \pm 0.20$ | | | | $6.9 \pm 0.17$ |
| **Worker 14** | | $5.2 \pm 0.27$ | | $9.0 \pm 0.17$ | |
| **Worker 15** | | | $14.0 \pm 0.13$ | | $6.7 \pm 0.27$ |

**Table 3.19 Quality Level Chart for 5-Process-15-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 | Operation 5 |
|---|---|---|---|---|---|
| **Worker 1** | $98.0 \pm 0.67$ | | $98.6 \pm 0.20$ | | |
| **Worker 2** | | $98.7 \pm 0.20$ | | $99.0 \pm 0.17$ | |
| **Worker 3** | $99.2 \pm 0.03$ | | | | $98.8 \pm 0.90$ |
| **Worker 4** | | $98.5 \pm 0.33$ | | $98.3 \pm 0.53$ | |
| **Worker 5** | | | $98.5 \pm 0.33$ | | $99.2 \pm 0.10$ |
| **Worker 6** | $97.5 \pm 1.0$ | | $98.8 \pm 0.17$ | | |
| **Worker 7** | | $99.0 \pm 0.17$ | | $98.5 \pm 0.33$ | |
| **Worker 8** | $98.8 \pm 0.20$ | | | | $99.0 \pm 0.17$ |
| **Worker 9** | | $97.0 \pm 1.0$ | | $99.5 \pm 0.20$ | |
| **Worker 10** | | | $99.1 \pm 0.10$ | | $98.0 \pm 0.50$ |
| **Worker 11** | $99.0 \pm 0.23$ | | $99.5 \pm 0.20$ | | |
| **Worker 12** | | $98.8 \pm 0.27$ | | $98.7 \pm 0.20$ | |
| **Worker 13** | $99.1 \pm 0.13$ | | | | $98.9 \pm 0.10$ |
| **Worker 14** | | $98.7 \pm 0.33$ | | $99.3 \pm 0.20$ | |
| **Worker 15** | | | $98.9 \pm 0.23$ | | $99.1 \pm 0.13$ |

**Table 3.20 Worker Cost ($) Chart for 5-Process-15-Worker Case Study**

| Operation/Worker | Operation 1 | Operation 2 | Operation 3 | Operation 4 | Operation 5 |
|---|---|---|---|---|---|
| **Worker 1** | 10.87/hr | | 12.87/hr | | |
| **Worker 2** | | 9.43/hr | | 9.25/hr | |
| **Worker 3** | 12.87/hr | | | | 11.00/hr |
| **Worker 4** | | 9.56/hr | | 12.37/hr | |
| **Worker 5** | | | 13.12/hr | | 10.75/hr |
| **Worker 6** | 13.25/hr | | 11.93/hr | | |
| **Worker 7** | | 10.25/hr | | 11.62/hr | |
| **Worker 8** | 9.37/hr | | | | 12.81/hr |
| **Worker 9** | | 11.50/hr | | 9.43/hr | |
| **Worker 10** | | | 11.06/hr | | 9.93/hr |
| **Worker 11** | 9.12/hr | | 13.06/hr | | |
| **Worker 12** | | 10.93/hr | | 10.18/hr | |
| **Worker 13** | 12.37/hr | | | | 10.12/hr |
| **Worker 14** | | 13.56/hr | | 8.43/hr | |
| **Worker 15** | | | 11.56/hr | | 12.00/hr |

Delay penalty for the processes are calculated using the dynamic delay penalty constraint and their values are shown below:

Delay penalty for process 1 = $175.21

Delay penalty for process 2 = $685.24

Delay penalty for process 3 = $660.82

Delay penalty for process 4 = $572.05

Delay penalty for process 5 = $244.49

Quality Penalty input values given for 5-process-15-worker case study is given below

Quality penalty for process 1 = $300

Quality penalty for process 2 = $400

Quality penalty for process 3 = $500

Quality penalty for process 4 = $600

Quality penalty for process 5 = $700

Maximum number of workers allowed in the 5-process-15-machine case study is set to 14 and standard quality level of all the processes is set at 99%. Input values were fed on to the LINGO Optimizer 12.0 to obtain optimal worker allocation and the results are as follows in Table 3.21.

**Table 3.21 Results – 5-Process-10-Worker Case study**

| PROCESS | WORKER |
|---|---|
| Process 1 | Worker 3 |
| Process 2 | Worker 7 |
| Process 3 | Worker 1, Worker 10, Worker 11 |
| Process 4 | Worker 2, Worker 9 |
| Process 5 | Worker 5, Worker 15 |

The results obtained from the LINGO model for 5-process-15-worker case study is simulated using QUEST software. Simulation is replicated for 18 times and the average throughput is 516.8 products.

### 3.4 Conclusion

In this chapter, definition of risk from the perspective of worker allocation was presented. There were two classes of risks associated with worker allocation - processing time risk and quality level risk. Processing time risk was defined as the increase in delay penalty cost due to uncertainty in worker processing time. Quality level risk was defined as the increase in quality penalty cost due to uncertainty associated with worker quality level. The proposed approach helps to allocate workers to the processes by minimizing overall risk. Non Linear Integer Programming (NLIP) model was developed which found the optimal worker allocation by minimizing the risk in labor intensive single task per station balanced/unbalanced production lines. In the next chapter, an extension of risk based worker allocation in a production line with multiple tasks at each station is presented.

# CHAPTER 4

# RISK BASED LINE BALANCING

## 4.1 Introduction

In Chapter 3, risk based worker allocation method for single task per station balanced/unbalanced production line scenarios were presented. The risk based worker allocation method aims to allocate the workers to the processes by minimizing the overall risk. In a production line with multiple tasks performed at each station, the products flow in a sequential order through the workstations. Each workstation has a number of tasks allocated to it. The performance of a multiple task production line primarily depends upon balancing the workload between stations and the quality of workers allotted to the workstations. Thus, in this chapter, a risk based line balancing problem is considered. Further, the risk based line balancing is extended into simultaneous risk based line balancing and worker allocation problem in Chapter 5 for multiple tasks per station scenario.

## 4.2 Notations

The notations used in the risk based line balancing approach are as follows:

$P_j$ Mean processing time of task j

$P_{\sigma j}$ Standard deviation in processing time of task j

$STP_j$ Standard processing time of task j

$T$ Time interval

$S_{\min}$ Minimum number of stations

$S_{\max}$ Maximum number of stations

$D$ Demand during time interval T

$L_{jk}$ Task j allocated to station k (Binary variable)

$RBP$ Risk balancing penalty

$LBP$ Line balancing penalty

$\Pr_{fg}$ Binary precedence variable that task $f$ has to be done before task $g$

## 4.3 Assumptions

The assumptions for risk based line balancing are listed below:

- Processing time of each task is known and follows normal distribution
- Standard processing time for each task is known
- A single product dedicated multiple task production line is considered
- Demand is deterministic and is known for each period

## 4.4 Model Input Requirements

The proposed Risk Based Line Balancing (RLB) model required the following input parameters for testing the approach.

1) Mean processing times of all tasks
2) Standard deviation in processing time for all tasks
3) Expected processing time for each task
4) Precedence constraints for all tasks

## 4.5 Mathematical Model

A Non Linear Integer Programming (NLIP) model is developed for solving risk based line balancing problem. The objective of the proposed model is to balance the workload and processing time risk between stations. The objective function and constraints for the proposed model for multiple tasks per station scenario are explained as follows:

*Objective Function*

$$Min\sum_{k=1}^{z}\left[\sum_{l=k+1}^{z}\left[\left(\left|RP_k - RP_l\right| * RBP\right) + \left(\left|SP_k - SP_l\right| * LBP\right)\right]\right] \tag{4.1}$$

*Wherein*

$$RP_k = \left[1 - \frac{1}{2}\left[1 + erf\left(\frac{S_k - SP_k}{SP\sigma_k\sqrt{2}}\right)\right]\right] \qquad \forall k \in K \tag{4.2}$$

$$SP_k = \sum_{j=1}^{m} P_j * L_{jk} \qquad \forall \text{k} \in K \tag{4.3}$$

$$SP\sigma^2{}_k = \sum_{j=1}^{m} P\sigma^2{}_j * L_{jk} \qquad \forall \text{k} \in K \tag{4.4}$$

$$S_k = \sum_{j=1}^{m} STP_j * L_{ijk} \qquad \forall \text{k} \in K \tag{4.5}$$

The objective function (Equation 4.1) minimizes the difference in risk and workload between stations by allocating task *j* to station *k*. The first term in objective function minimizes the difference in processing time risk between stations. The second term in the objective function balances the work load between stations by minimizing the difference in mean workload between stations. Equation 4.2 through 4.5 expands the terms in objective function. The constraints for the risk based assembly line balancing problem are shown below.

$$\sum_{k=1}^{z} L_{jk} = 1 \qquad \forall \text{j} \in J \tag{4.6}$$

$$\sum_{j=1}^{m} L_{jk} \geq 1 \qquad \forall k \in K \tag{4.7}$$

$$\text{Precedence} = \begin{cases} L_{fk} \leq \sum_{h=k}^{z} L_{gh} \quad \forall k \in K & if \quad \Pr_{fg} = 1 \\ No\ const. & if \quad \Pr_{fg} = 0 \end{cases} \qquad \forall \text{k} \in \text{K} \tag{4.8}$$

In the above set of constraints, Constraint 4.6 ensures that a task can only be allotted to one station. Constraint 4.7 ensures that a station will have a minimum of one task. Constraint 4.8 ensures that the precedence relationships within tasks are met. The binary precedence variable

$Pr_{fg}$ restricts the task assignment to workstation. $Pr_{fg}$ =1 represents that task *f* has to be allocated before task *g*. The Non Linear Integer Programming (NLIP) model was solved using LINGO optimizer 12.0 software. The output obtained from the LINGO software provides the optimal task-station allocation.

### 4.6 Case Study – 9-Task-3-Station

A multiple task per station production line with 9 tasks and 3 stations is used as a sample case study to illustrate the proposed RLB model. The expected values of processing time for 9 task, 3 station case study is shown in Table 4.1.

**Table 4.1 Expected Processing Time for 9-Task-3-Station Case Study**

| Task | Expected Processing Time |
|---|---|
| 1 | 15 |
| 2 | 26 |
| 3 | 8 |
| 4 | 12 |
| 5 | 14 |
| 6 | 10 |
| 7 | 23 |
| 8 | 28 |
| 9 | 16 |

Actual operation times of all the tasks in the 9 task - 3 station case study is shown in Table 4.2. The actual task processing time values are obtained from previous historical time study.

**Table 4.2 Processing Time (min) Chart for 9-Task–3-Station Case Study**

| Task | Processing Time |
|---|---|
| 1 | 13±1.56 |
| 2 | 25.8±1.34 |
| 3 | 7.6±1.15 |
| 4 | 11.1±0.94 |
| 5 | 14.4±1.22 |
| 6 | 10.2±1.14 |
| 7 | 22.8±0.85 |
| 8 | 27.85±0.66 |
| 9 | 15.8±0.96 |

The binary precedence variable $\Pr_{fg}$ for 9 task – 3 station case study is shown below:

$$\Pr_{fg} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The time interval (T) is set to 480 hours. Minimum number of stations ($S_{\min}$) is set as 3 and the demand is assumed as 700 products. Risk balancing penalty and line balancing penalties are $1000 for the 9 task, 3 station case study.

*LINGO model for* 9 task and 3 station case study*:*

Once the input variables are obtained, NLIP model is solved and the results are shown below:

*Objective Function*

$$Min\sum_{k=1}^{3}\left[\sum_{l=k+1}^{3}\left[\left(\left|RP_k - RP_l\right|*1000\right)+\left(\left|SP_k - SP_l\right|*1000\right)\right]\right]$$

*Wherein*

$$RP_k = \left[1-\frac{1}{2}\left[1+erf\left(\frac{S_k - SP_k}{SP\sigma_k\sqrt{2}}\right)\right]\right] \qquad \forall k \in K$$

$$RQ_k = \frac{1}{2}\left[1+erf\left(\frac{Q_k - SQ_k}{SQ\sigma_k\sqrt{2}}\right)\right] \qquad \forall k \in K$$

$$SP_k = \sum_{j=1}^{9} P_j * L_{jk} \qquad \forall \mathrm{k} \in K$$

$$SP\sigma^2{}_k = \sum_{j=1}^{9} P\sigma^2{}_j * L_{jk} \qquad \forall \mathrm{k} \in K$$

$$S_k = \sum_{j=1}^{9} STP_j * L_{ijk} \qquad \forall \mathrm{k} \in K$$

$$\sum_{k=1}^{3} L_{jk} = 1 \qquad \forall \mathrm{j} \in J$$

$$\sum_{j=1}^{9} L_{jk} \geq 1 \qquad \forall k \in K$$

$$\text{Precedence} = \begin{cases} L_{fk} \leq \sum_{h=k}^{3} L_{gh} \quad \forall k \in K & if \quad \Pr_{fg} = 1 \\ No\ const. & if \quad \Pr_{fg} = 0 \end{cases} \qquad \forall \mathrm{k} \in \mathrm{K}$$

The output from the LINGO Optimizer 12.0 is obtained (Table 4.3).

**Table 4.3 Results –9-Task–3-Station-Case Study-Risk Based Allocation**

| STATION | Station 1 | Station 2 | Station 3 |
|---|---|---|---|
| 3 STATION | Task 4<br>Task 7<br>Task 9 | Task 3<br>Task 5<br>Task 8 | Task 1<br>Task 2<br>Task 6 |

## 4.7 Validation

In order to validate the risk based line balancing approach, a deterministic line balancing is performed using rank positional weight method for the same 9-task-3-station case study and results are shown in Table 4.4.

**Table 4.4 Results –9-Task–3-Station-Case Study-Deterministic Allocation**

| STATION | Station 1 | Station 2 | Station 3 |
|---|---|---|---|
| **3 STATION** | Task 8<br>Task 2 | Task 7<br>Task 9<br>Task 1 | Task 5<br>Task 4<br>Task 6<br>Task 3 |

Simulation is selected as the tool to compare deterministic and risk based allocation. Simulation is conducted using Delmia QUEST V5 R18 software. Simulation is run for 480 hours. Since the processing time tasks are assumed to follow normal distribution, a single run may not be sufficient enough to eliminate randomness in output. Hence, the model is replicated several times. Number of replications was calculated using Equation (4.9).

$$N_r = t^2_{\frac{\alpha}{2}, n-1} \frac{\sigma^2}{h^2} \quad (4.9)$$

The optimal results obtained from the LINGO optimizer is replicated 5 times for risk based allocation and 3 times for deterministic allocation. Comparison of the outputs from simulation is shown in Table 4.5.

**Table 4.5 Comparison Between Deterministic and Risk Based Method**

| Description | Deterministic Method | Risk based Method | Percent Improvement |
|---|---|---|---|
| **Throughput** | 531.9 | 561.4 | 5.54 |
| **Value Added Time** | 79190.68 | 83602.3 | 5.57 |
| **Non-Value Added Time** | 7209.32 | 2797.68 | 51.19 |

Figure 4.1 shows a comparison of cycle time between deterministic and risk based line balancing methodology.

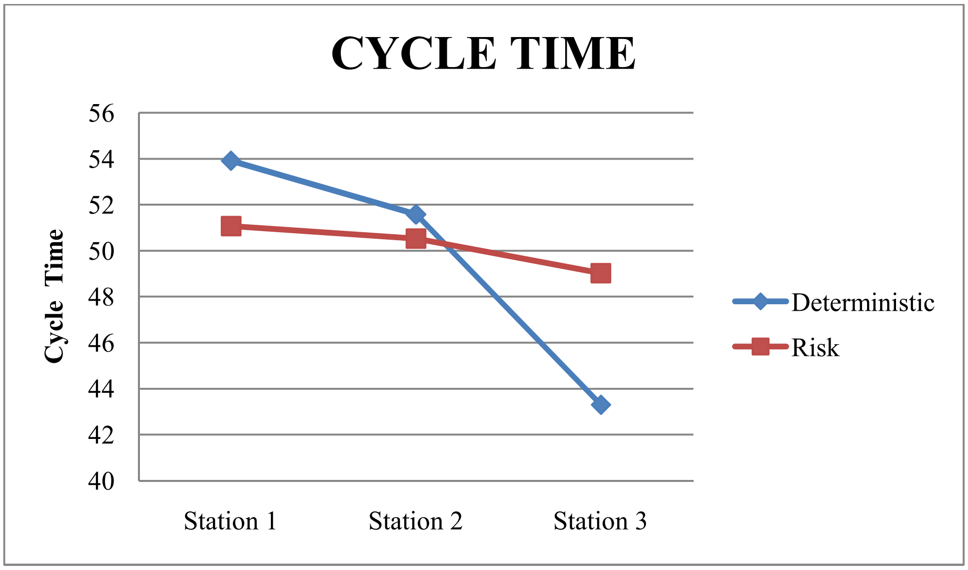


**Figure 4.1 Cycle Time Comparison Chart**

Thus, from the above table and chart, it is evident that the risk based line balancing produces a better balance between the work stations in terms of time taken to complete the tasks assigned to the stations. This results in an increased throughput of 5.54% and simultaneously value added time is also improved.

## 4.8 Optimum Number of Station Determination

The throughput of a multiple tasks per station production line is dependent on the cycle time of the stations. When the number of stations in production line increases/decreases, the cycle time of the stations also increases/decreases respectively. Thus, it is also vital to make decisions on number of stations when balancing a multiple tasks per station production line. An algorithm to make decisions on number of stations when balancing a production line is presented below.

*Algorithm for optimal number of station determination:*

STEP 1: Start

STEP 2: Set Z = minimum number of stations ( $S_{min}$ )

STEP 3: Run RLB LINGO model to obtain optimal result

STEP 4: Determine throughput by simulation

STEP 5: If Throughput ≥ Demand go to STEP 7

STEP 6: Set Z= Z+1 go to STEP 3

STEP 7: Stop

A flowchart for optimal number of station determination in an assembly line is shown in Figure 4.2.

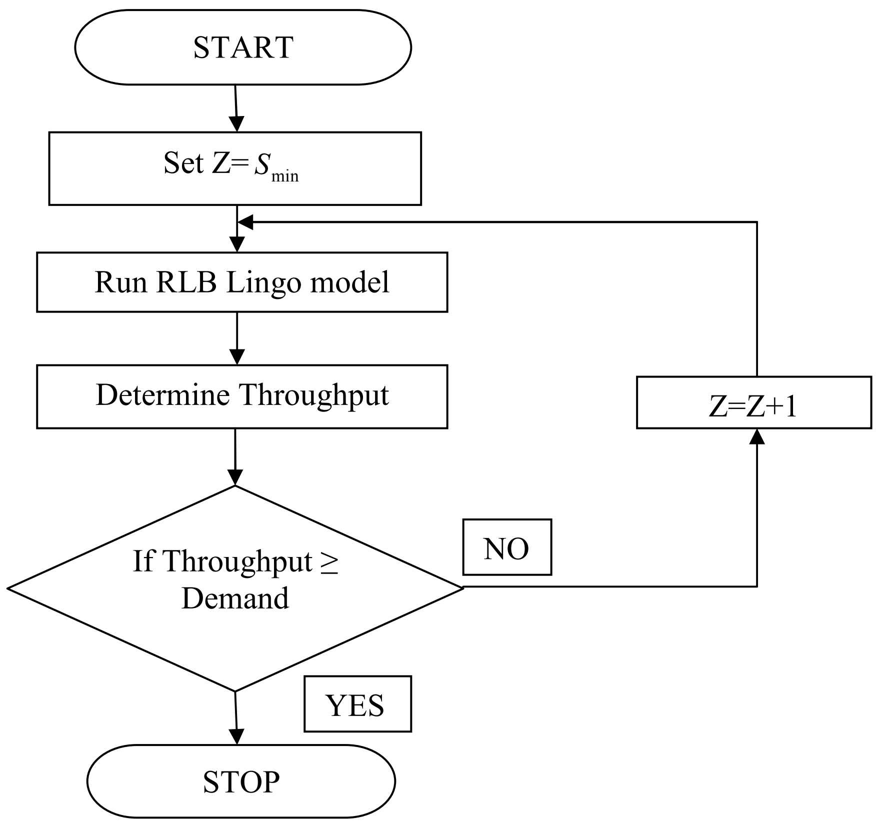


**Figure 4.2 Flowchart for optimal number of station determination - RLB**

### 4.9 Case Study – 9 Task

To test the proposed methodology for optimal number of station determination, same 9 station case study is selected. Demand is assumed as 700 products and minimum number of stations ($S_{\min}$) is set as 3. The RLB LINGO model is solved for 3 stations and results are shown in Table 4.6.

**Table 4.6 Results –9-Task–3-Station-Case Study**

| STATION | Station 1 | Station 2 | Station 3 | Avg. Throughput |
|---|---|---|---|---|
| **3 STATION** | Task 4<br>Task 7<br>Task 9 | Task 3<br>Task 5<br>Task 8 | Task 1<br>Task 2<br>Task 6 | 561.4 |

Simulation is performed with the obtained result to find the average throughput. The simulation is replicated 6 times for the 3 station case. Time interval for simulation is assumed to be 480 hours. The average throughput obtained for 3 stations-9 task case study is 561.4.

Since the demand of 700 products could not be met by 3 stations, number of stations is increased by one and RLB model is solved for optimality.

*LINGO model for* 9 task and 4 station case study*:*

Once the input variables were obtained, NLIP model was solved and the results are shown below:

*Objective Function*

$$Min\sum_{k=1}^{4}\left[\sum_{l=k+1}^{4}\left[\left(\left|RP_k - RP_l\right|*1000\right)+\left(\left|SP_k - SP_l\right|*1000\right)\right]\right]$$

*Wherein*

$$RP_k = \left[1 - \frac{1}{2}\left[1 + erf\left(\frac{S_k - SP_k}{SP\sigma_k\sqrt{2}}\right)\right]\right] \qquad \forall k \in K$$

$$RQ_k = \frac{1}{2}\left[1 + erf\left(\frac{Q_k - SQ_k}{SQ\sigma_k\sqrt{2}}\right)\right] \qquad \forall k \in K$$

$$SP_k = \sum_{j=1}^{9} P_j * L_{jk} \qquad \forall \mathrm{k} \in K$$

$$SP\sigma^2{}_k = \sum_{j=1}^{9} P\sigma^2{}_j * L_{jk} \qquad \forall \mathrm{k} \in K$$

$$S_k = \sum_{j=1}^{9} STP_j * L_{ijk} \qquad \forall \mathrm{k} \in K$$

$$\sum_{k=1}^{4} L_{jk} = 1 \qquad \forall \mathrm{j} \in J$$

$$\sum_{j=1}^{9} L_{jk} \geq 1 \qquad \forall k \in K$$

$$\text{Precedence} = \begin{cases} L_{fk} \leq \sum_{h=k}^{4} L_{gh} \quad \forall k \in K & if \quad \Pr_{fg} = 1 \\ No\ const. & if \quad \Pr_{fg} = 0 \end{cases} \qquad \forall \mathrm{k} \in \mathrm{K}$$

The result for 9-task-4 station case study is shown in Table 4.7.

**Table 4.7 Results –9-Task–4-Station-Case Study**

| STATION | Station 1 | Station 2 | Station 3 | Station 4 | Avg. Throughput |
|---|---|---|---|---|---|
| **4 STATION** | Task 7<br>Task 9 | Task 4<br>Task 8 | Task 1<br>Task 2 | Task 3<br>Task 5<br>Task 6 | 716.7 |

Simulation is performed with the obtained allocations of workers and tasks to find the average throughput. The simulation is replicated 6 times for the 4 station case. Time interval for simulation is assumed to be 480 hours. The average throughput obtained for 4 stations-9 task case study is 716.7. Thus demand of 700 products could be met by 4 stations. Hence, 4 stations is determined to be the optimal number of stations for the 9 task case study.

### 4.10 Conclusion

In this chapter, the necessity of risk based line balancing in multiple tasks per station scenario was presented. An NLIP optimization methodology was proposed which allocates tasks to the stations by balancing the risk and the mean processing times between stations. The proposed approach is validated against a deterministic approach (Rank Positional Weight Method) and a 5.6 percent improvement in throughput was observed. A methodology to determine the optimal number of stations in an assembly line was also presented. Efficiency of a multiple tasks per station production line depends on line balancing and worker allocation. Hence in chapter 5, the proposed risk based line balancing approach was extended to a simultaneous risk based line balancing and worker allocation problem.

# CHAPTER 5

# SIMULTANEOUS RISK BASED LINE BALANCING AND WORKER ALLOCATION

## 5.1 Introduction

In Chapter 4, risk based line balancing approach was presented. The risk based line balancing approach aims to allocate tasks to workstations by balancing the risk and mean processing time between stations. The performance of a multiple tasks per station production line primarily depends upon balancing the workload between stations and the quality of workers allotted to the workstations. Thus, an approach for simultaneously balancing the workstation and allocating best worker to the workstation is presented in this chapter. A single model un-paced asynchronous production line is considered in this research.

## 5.2 Notations

The notations used in the simultaneous risk based line balancing and worker allocation is as follows:

$P_{ij}$ Processing time of worker i for task j

$P_{\sigma ij}$ Standard deviation in processing time of labor i for task j

$Q_{ij}$ Quality level of worker i for task j

$Q_{\sigma ij}$ Standard deviation in quality level of worker i for task j

$SP_j$ Standard processing time of task j

$SQ_j$ Standard quality level of task j

$DP_k$ Delay penalty for station k

$QP_k$ Quality penalty for station k

$C_{ij}$ Cost of worker i for task j

$T$ Time interval

$S_{\min}$ Minimum number of stations

$S_{\max}$ Maximum number of stations

$D$ Demand during time interval T

$L_{ijk}$ Worker i for task j in station k (Binary variable)

$RBP$ Risk balancing penalty

$LBP$ Line balancing penalty

$\Pr_{fg}$ Binary precedence variable that task f has to be done before task g

$V_{ik}$ Binary variable equals 1 if labor i is allotted to station k, 0 otherwise.

## 5.3 Assumptions

The assumptions for simultaneous risk based line balancing and worker allocation are listed below:

- Processing time of each worker is known and follows normal distribution
- Quality level of each worker is known and follows normal distribution
- Standard processing time of each task is known
- Standard quality level of each task is known
- A worker can only be allocated to a single station
- A station has only one worker

## 5.4 Model Input Requirements

The proposed Simultaneous Risk Based Line Balancing and Worker Allocation (SRLW) model requires the following input parameters for testing the approach:

1) Processing times of all workers to their respective task

2) Quality level of all workers to their respective task

3) Cost of workers for each task

4) Expected processing time for each task

5) Expected quality level for each task

6) Delay penalty associated with each station

7) Quality penalty associated with each station

8) Precedence constraints for all tasks

### 5.5 Mathematical Model

A Non Linear Integer Programming (NLIP) model is developed for solving SRLW problem. The proposed model balances the workload, risk between stations, and allocates the best worker to the station simultaneously. The objective function and constraints are explained as follows:

*Objective Function*

$$Min\sum_{k=1}^{z}\left[RP_k * DP_k + RQ_k * QP_k + CL_k + \sum_{l=k+1}^{z}\left[\left(\left|RP_k - RP_l\right| * RBP\right) + \left(\left|RQ_k - RQ_l\right| * RBP\right) + \left(\left|SP_k - SP_l\right| * LBP\right)\right]\right] \tag{5.1}$$

*Wherein*

$$RP_k = \left[1 - \frac{1}{2}\left[1 + erf\left(\frac{S_k - SP_k}{SP\sigma_k\sqrt{2}}\right)\right]\right] \qquad \forall k \in K \tag{5.2}$$

$$RQ_k = \frac{1}{2}\left[1 + erf\left(\frac{Q_k - SQ_k}{SQ\sigma_k\sqrt{2}}\right)\right] \qquad \forall k \in K \tag{5.3}$$

$$SP_k = \sum_{j=1}^{m}\sum_{i=1}^{n} P_{ij} * L_{ijk} \qquad \forall k \in K \tag{5.4}$$

$$SP\sigma^2{}_k = \sum_{j=1}^{m}\sum_{i=1}^{n} P\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K \qquad (5.5)$$

$$SQ_k = \sum_{j=1}^{m}\sum_{i=1}^{n} Q_{ij} * L_{ijk} \qquad \forall k \in K \qquad (5.6)$$

$$SQ\sigma^2{}_k = \sum_{j=1}^{m}\sum_{i=1}^{n} Q\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K \qquad (5.7)$$

$$S_k = \sum_{j=1}^{m}\sum_{i=1}^{n} STP_j * L_{ijk} \qquad \forall k \in K \qquad (5.8)$$

$$Q_k = \sum_{j=1}^{m}\sum_{i=1}^{n} STQ_j * L_{ijk} \qquad \forall k \in K \qquad (5.9)$$

$$CL_k = \sum_{j=1}^{m}\sum_{i=1}^{n} \left( C_{ij} * \frac{T}{60} \right) * L_{ijk} \qquad \forall k \in K \qquad (5.10)$$

The objective function (Equation 5.1) minimizes the risk due to worker allocation, the difference in workload between stations, and the difference in risk between stations by allocating worker *i* to task *j* and task *j* to station *k*. The first term in the objective function minimizes the risk due to station processing time, which is the product of probability in exceeding the expected processing time and station delay penalty. The second term minimizes the risk due to station quality level, which is the product of probability in quality level below expected quality level and station quality penalty. The third term minimizes the cost of workers. The fourth term minimizes the difference in processing time risk between stations. The fifth term minimizes the difference in quality risk between stations. The sixth term balances the work load between stations by minimizing the difference in mean workload between stations. Equations 5.2 through 5.10 expand the terms in objective function. The constraints for the SRLW problem are shown below.

*Constraints*

$$\sum_{j=1}^{m}\left[L_{ijk}\right]=\sum_{i=1}^{n}\sum_{j=1}^{m}\left[L_{ijk}*V_{ik}\right] \qquad \forall i\in I,k\in K \tag{5.11}$$

$$\sum_{i=1}^{n}V_{ik}=1 \qquad \forall k\in K \tag{5.12}$$

$$\sum_{k=1}^{z}V_{ik}\leq 1 \qquad \forall i\in I \tag{5.13}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{z}L_{ijk}=1 \qquad \forall j\in J \tag{5.14}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m}L_{ijk}\geq 1 \qquad \forall k\in K \tag{5.15}$$

$$\text{Precedence}=\begin{cases}\sum_{i=1}^{n}L_{igk}\leq\sum_{i=1}^{n}\sum_{h=1}^{k}L_{ifh} & \forall k\in K \quad \textit{if} \quad \Pr_{fg}=1\\ \textit{No const.} & \textit{if} \quad \Pr_{fg}=0\end{cases} \qquad \forall \text{k}\in \text{K} \tag{5.16}$$

Constraints 5.11 and 5.12 ensure that only one worker can be allocated to a station. Constraint 5.13 ensures that a worker can only be allotted to a maximum of one station. Constraint 5.14 ensures that a task can only be allotted to one worker and one station. Constraint 5.15 ensures that at least one task is allocated to a station. Constraint 5.16 ensures that the precedence relationships within tasks are not violated.

Station delay penalty $(DP_k)$ is the penalty incurred if the processing time of a certain station exceeds the expected station processing time. Station delay penalty will change as the criticality of the station changes. Station quality penalty $(QP_k)$ is the penalty incurred if the quality level of a certain station is below the expected station quality level. The binary precedence variable $\Pr_{fg}$ restricts the task assignment to workstation. $\Pr_{fg}=1$ represents that task *f* has to be allocated before task *g*. The Non Linear Integer Programming (NLIP) model is solved using LINGO optimizer 12.0 software. The output obtained from the LINGO software provides the optimal worker-task-station allocation.

The throughput of a multiple task production line is dependent upon the cycle time of the stations. When number of stations in an assembly line increases/decreases, the cycle time of the stations also increases/decreases respectively. Thus, it is also vital to make decisions on number of stations when balancing an assembly line. An algorithm to make decisions on number of stations when balancing an assembly line is presented below.

*Algorithm for optimal number of station determination:*

STEP 1: Start

STEP 2: Set Z = minimum number of stations ( $S_{\min}$ )

STEP 3: Run SRLW LINGO model to obtain optimal result

STEP 4: Determine throughput by simulation

STEP 5: If Throughput ≥ Demand go to STEP 7

STEP 6: Set Z= Z+1 go to STEP 3

STEP 7: Stop

A flowchart for optimal number of station determination in an assembly line is shown in Figure 5.1.

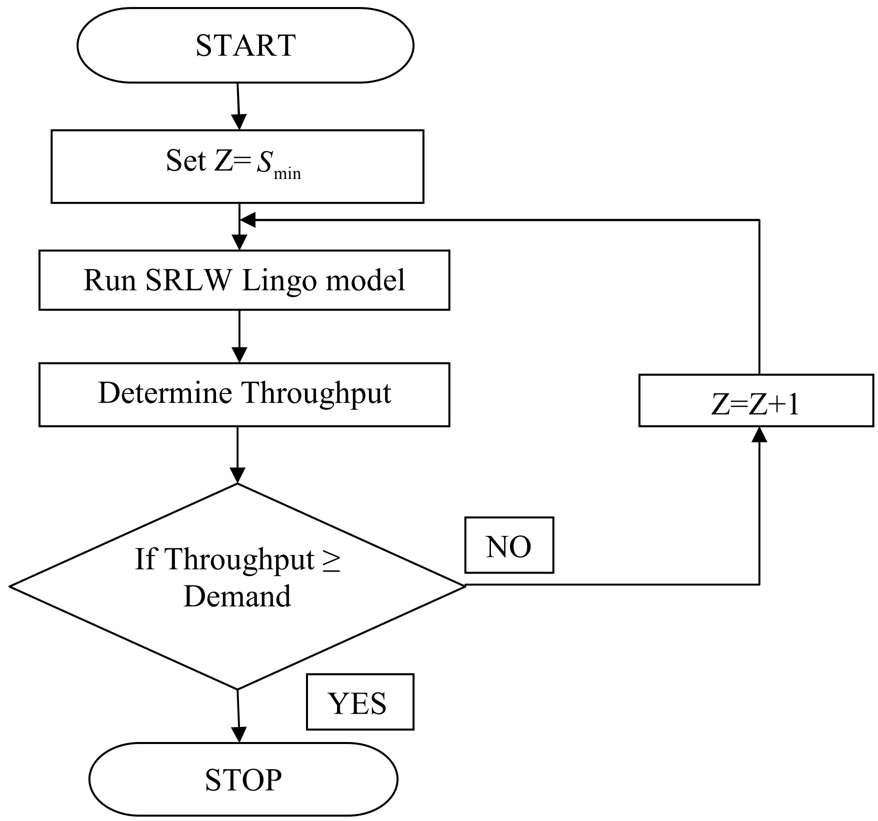


**Figure 5.1 Flowchart for optimal number of station determination - SRLW**

To demonstrate the simultaneous risk based line balancing and worker allocation model, sample case studies are shown in the following sections.

## 5.6 Case Study – 9 Task – 6 Worker

A multiple tasks per station production line with 9 task and 6 workers is used to illustrate the proposed model. The expected values of processing time and quality level for all the tasks in the 9-task-6-worker case study are shown in Table 5.1.

**Table 5.1 Expected Processing Time (min)/ Quality Level Chart for**

**9-Task–6-Worker Case Study**

| TASK | PROCESSING TIME | QUALITY |
|---|---|---|
| 1 | 15 | 98 |
| 2 | 8 | 98.5 |
| 3 | 26 | 99 |
| 4 | 12 | 98 |
| 5 | 23 | 97 |
| 6 | 10 | 97.5 |
| 7 | 14 | 99 |
| 8 | 28 | 98 |
| 9 | 16 | 99.5 |

Actual operation times of all the workers in the 9-task-6-worker case study are shown in Table 5.2. The worker processing time values are obtained from previous historical time study.

**Table 5.2 Processing Time (min) Chart for 3-Station-9-Task–6-Worker Case Study**

| | TASK1 | TASK2 | TASK3 | TASK4 | TASK5 |
|---|---|---|---|---|---|
| Worker 1 | 14.5±0.89 | 7.9±0.36 | 25.2±1.0 | 11.8±0.62 | 22.5±0.72 |
| Worker 2 | 15.2±0.32 | 7.6±1.05 | 26.2±0.40 | 12.1±0.41 | 23.1±0.41 |
| Worker 3 | 13.0±1.5 | 8.0±0.83 | 25.8±0.62 | 11.5±0.82 | 22.0±1.00 |
| Worker 4 | 14.8±0.78 | 8.1±0.52 | 24.0±1.50 | 11.0±1.00 | 22.8±0.80 |
| Worker 5 | 15.0±0.52 | 7.5±1.21 | 24.5±1.20 | 12.2±0.30 | 23.0±0.52 |
| Worker 6 | 13.5±1.00 | 7.8±0.38 | 26.0±0.33 | 11.0±0.92 | 23.2±0.30 |

| | TASK6 | TASK7 | TASK8 | TASK9 |
|---|---|---|---|---|
| Worker 1 | 9.9±0.66 | 14.0±0.41 | 27.0±1.00 | 15.8±0.76 |
| Worker 2 | 9.6±1.11 | 13.8±0.73 | 27.5±0.75 | 15.5±0.92 |
| Worker 3 | 10.1±0.35 | 14.1±0.30 | 28.0±0.42 | 16.0±0.41 |
| Worker 4 | 10.0±0.50 | 13.5±0.93 | 28.1±0.30 | 16.2±0.30 |
| Worker 5 | 9.8±0.73 | 13.4±1.11 | 26.5±1.55 | 15.2±1.32 |
| Worker 6 | 10.2±0.30 | 13.7±0.84 | 27.8±0.63 | 15.9±0.50 |

Actual quality level of all the workers in the 9-task-6-worker case study is shown in Table 5.3. The worker quality level values are obtained from previous historical time study.

**Table 5.3 Quality Level Chart for 3-Station-9-Task–6-Worker Case Study**

| | TASK1 | TASK2 | TASK3 | TASK4 | TASK5 |
|---|---|---|---|---|---|
| Worker 1 | $97.5 \pm 0.8$ | $98.0 \pm 0.26$ | $98.0 \pm 0.65$ | $97.7 \pm 0.34$ | $96.5 \pm 0.34$ |
| Worker 2 | $97.8 \pm 0.24$ | $98.3 \pm 0.24$ | $99.2 \pm 0.03$ | $98.2 \pm 0.48$ | $97.1 \pm 0.18$ |
| Worker 3 | $98.0 \pm 0.53$ | $99.2 \pm 0.22$ | $97.5 \pm 0.8$ | $97.5 \pm 0.26$ | $96.8 \pm 0.24$ |
| Worker 4 | $97.8 \pm 0.32$ | $98.8 \pm 0.30$ | $98.8 \pm 0.20$ | $98.0 \pm 0.50$ | $97.2 \pm 0.21$ |
| Worker 5 | $98.2 \pm 0.30$ | $98.0 \pm 0.66$ | $99.0 \pm 0.23$ | $98.1 \pm 0.21$ | $97.0 \pm 0.30$ |
| Worker 6 | $98.1 \pm 0.30$ | $98.6 \pm 0.16$ | $99.1 \pm 0.13$ | $97.8 \pm 0.36$ | $96.7 \pm 0.41$ |

| | TASK6 | TASK7 | TASK8 | TASK9 |
|---|---|---|---|---|
| Worker 1 | $98.0 \pm 0.33$ | $99.0 \pm 0.18$ | $97.5 \pm 0.33$ | $99.0 \pm 0.33$ |
| Worker 2 | $98.1 \pm 0.26$ | $98.8 \pm 0.38$ | $98.1 \pm 0.30$ | $99.4 \pm 0.10$ |
| Worker 3 | $97.5 \pm 0.32$ | $98.3 \pm 0.53$ | $97.7 \pm 0.21$ | $99.2 \pm 0.26$ |
| Worker 4 | $97.2 \pm 0.26$ | $98.5 \pm 0.33$ | $98.3 \pm 0.24$ | $99.0 \pm 0.33$ |
| Worker 5 | $96.5 \pm 0.83$ | $99.5 \pm 0.15$ | $97.0 \pm 0.52$ | $98.0 \pm 0.66$ |
| Worker 6 | $99.3 \pm 0.22$ | $98.7 \pm 0.24$ | $97.6 \pm 0.28$ | $99.3 \pm 0.22$ |

The cost for all workers associated with their respective processes in the 9-task-6 worker case study is given in Table 5.4.

**Table 5.4 Worker Cost ($) Chart for 3-Station-9-Task–6-Worker Case Study**

| | TASK1 | TASK2 | TASK3 | TASK4 | TASK5 |
|---|---|---|---|---|---|
| Worker 1 | $9.48/hr | $11.83/hr | $10.04/hr | $10.35/hr | $10.65/hr |
| Worker 2 | $9.41/hr | $8.40/hr | $11.81/hr | $10.98/hr | $11.98/hr |
| Worker 3 | $10.66/hr | $14.93/hr | $13.12/hr | $11.08/hr | $11.83/hr |
| Worker 4 | $9.37/hr | $13.61/hr | $14.15/hr | $9.05/hr | $14.22/hr |
| Worker 5 | $12.43/hr | $11.43/hr | $14.87/hr | $11.88/hr | $10.71/hr |
| Worker 6 | $9.57/hr | $9.26/hr | $10.60/hr | $13.12/hr | $8.47/hr |

| | TASK6 | TASK7 | TASK8 | TASK9 |
|---|---|---|---|---|
| Worker 1 | $13.48/hr | $10.38/hr | $10.10/hr | $11.73/hr |
| Worker 2 | $12.69/hr | $10.41/hr | $13.41/hr | $13.95/hr |
| Worker 3 | $14.98/hr | $13.67/hr | $11.22/hr | $10.71/hr |
| Worker 4 | $12.17/hr | $11.63/hr | $12.81/hr | $11.80/hr |
| Worker 5 | $8.44/hr | $12.50/hr | $13.93/hr | $8.73/hr |
| Worker 6 | $10.38/hr | $12.66/hr | $10.91/hr | $13.36/hr |

The binary precedence variable $\mathrm{Pr}_{fg}$ for 9-task-6 worker case study is shown below.

$$
Pr_{fg} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Station delay penalties $(DP_k)$ for the 9-task-6 worker case study are given below:

Delay penalty for station 1 $(DP_1)$: \$450

Delay penalty for station 2 $(DP_2)$: \$550

Delay penalty for station 3 $(DP_3)$: \$500

Station quality penalties $(QP_k)$ for the 9 task-6-worker case study are given below:

Quality penalty for station 1 $(DP_1)$: \$375

Quality penalty for station 2 $(DP_2)$: \$350

Quality penalty for station 3 $(DP_3)$: \$400

The time interval (T) is set to 2880 minutes. Minimum number of stations ( $S_{\min}$ ) is set at 3 and the demand is assumed to be 70 products. Risk balancing penalty and line balancing penalties are \$1000 for the 9-task-6 worker case study.

*LINGO model for* 3 station, 9 task and 6 worker case study*:*

Once the input variables were obtained, NLIP model is solved and the results are shown below:

*Objective function:*

$$Min\sum_{k=1}^{3}\left[RP_k * DP_k + RQ_k * QP_k + CL_k + \sum_{l=k+1}^{3}\left[\left(\left|RP_k - RP_l\right| * 1000\right) + \left(\left|RQ_k - RQ_l\right| * 1000\right) + \left(\left|SP_k - SP_l\right| * 1000\right)\right]\right]$$

*Wherein*

$$RP_k = \left[1 - \frac{1}{2}\left[1 + erf\left(\frac{S_k - SP_k}{SP\sigma_k\sqrt{2}}\right)\right]\right] \qquad \forall k \in K$$

$$RQ_k = \frac{1}{2}\left[1 + erf\left(\frac{Q_k - SQ_k}{SQ\sigma_k\sqrt{2}}\right)\right] \qquad \forall k \in K$$

$$SP_k = \sum_{j=1}^{9}\sum_{i=1}^{6} P_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SP\sigma^2{}_k = \sum_{j=1}^{9}\sum_{i=1}^{6} P\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SQ_k = \sum_{j=1}^{9}\sum_{i=1}^{6} Q_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SQ\sigma^2{}_k = \sum_{j=1}^{9}\sum_{i=1}^{6} Q\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K$$

$$S_k = \sum_{j=1}^{9}\sum_{i=1}^{6} STP_j * L_{ijk} \qquad \forall k \in K$$

$$Q_k = \sum_{j=1}^{9}\sum_{i=1}^{6} STQ_j * L_{ijk} \qquad \forall k \in K$$

$$CL_k = \sum_{j=1}^{9}\sum_{i=1}^{6}\left(C_{ij} * \frac{T}{60}\right) * L_{ijk} \qquad \forall k \in K$$

*Subject to*

$$\sum_{j=1}^{9}\left[L_{ijk}\right] = \sum_{i=1}^{6}\sum_{j=1}^{9}\left[L_{ijk} * V_{ik}\right] \qquad \forall i \in I, k \in K$$

$$\sum_{i=1}^{6} V_{ik} = 1 \qquad \forall k \in K$$

$$\sum_{k=1}^{3} V_{ik} \leq 1 \qquad \forall i \in I$$

$$\sum_{i=1}^{6} \sum_{k=1}^{3} L_{ijk} = 1 \qquad \forall j \in J$$

$$\sum_{i=1}^{6} \sum_{j=1}^{9} L_{ijk} \geq 1 \qquad \forall k \in K$$

$$\text{Precedence} = \begin{cases} \sum_{i=1}^{6} L_{igk} \leq \sum_{i=1}^{6} \sum_{h=1}^{3} L_{ifh} & \forall k \in K \quad if \quad \Pr_{fg} = 1 \\ No\ const. & if \quad \Pr_{fg} = 0 \end{cases} \qquad \forall \mathrm{k} \in \mathrm{K}$$

The output from the LINGO Optimizer 12.0 is obtained and simulation is conducted using Delmia QUEST V5 R18 software. Since the processing time and quality level of workers are assumed to follow normal distribution, a single run may not be sufficient to eliminate randomness in output. Hence, the model is replicated several times. Number of replications is calculated using Equation (5.17).

$$N_r = t^2_{\frac{\alpha}{2}, n-1} \frac{\sigma^2}{h^2} \tag{5.17}$$

The optimal results obtained from the LINGO optimizer were replicated for 3 times for 3 station case and average throughput for 2880 minutes is calculated to be 51.55 products (Table 5.5).

**Table 5.5 Results – 3-Station-9-Task–6-Worker Case Study**

| STATION | | Station 1 | Station 2 | Station 3 | Throughput |
|---|---|---|---|---|---|
| 3 STATION | TASKS | Task 2<br>Task 7<br>Task 8 | Task 4<br>Task 5<br>Task 9 | Task 1<br>Task 3<br>Task 6 | 51.55 |
| | WORKER | Worker 6 | Worker 3 | Worker 5 | |

Since the average throughput obtained for 3 stations is not enough to meet the demand of 70 products, the number of stations is increased by one and the same case study is solved in LINGO optimizer for 4 stations. The LINGO model for 4 station case is shown below.

*LINGO model for* 4 station, 9 task and 6 worker case study*:*

*Objective function:*

$$Min\sum_{k=1}^{4}\left[RP_k * DP_k + RQ_k * QP_k + CL_k + \sum_{l=k+1}^{4}\left[\left(\left|RP_k - RP_l\right| * 1000\right) + \left(\left|RQ_k - RQ_l\right| * 1000\right) + \left(\left|SP_k - SP_l\right| * 1000\right)\right]\right]$$

*Where in*

$$RP_k = \left[1 - \frac{1}{2}\left[1 + erf\left(\frac{S_k - SP_k}{SP\sigma_k\sqrt{2}}\right)\right]\right] \qquad \forall k \in K$$

$$RQ_k = \frac{1}{2}\left[1 + erf\left(\frac{Q_k - SQ_k}{SQ\sigma_k\sqrt{2}}\right)\right] \qquad \forall k \in K$$

$$SP_k = \sum_{j=1}^{9}\sum_{i=1}^{6} P_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SP\sigma^2{}_k = \sum_{j=1}^{9}\sum_{i=1}^{6} P\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SQ_k = \sum_{j=1}^{9}\sum_{i=1}^{6} Q_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SQ\sigma^2{}_k = \sum_{j=1}^{9}\sum_{i=1}^{6} Q\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K$$

$$S_k = \sum_{j=1}^{9}\sum_{i=1}^{6} STP_j * L_{ijk} \qquad \forall k \in K$$

$$Q_k = \sum_{j=1}^{9}\sum_{i=1}^{6} STQ_j * L_{ijk} \qquad \forall k \in K$$

$$CL_k = \sum_{j=1}^{9}\sum_{i=1}^{6}\left(C_{ij} * \frac{T}{60}\right) * L_{ijk} \qquad \forall k \in K$$

*Subject to*

$$\sum_{j=1}^{9}\left[L_{ijk}\right]=\sum_{i=1}^{6}\sum_{j=1}^{9}\left[L_{ijk}*V_{ik}\right] \qquad \forall i\in I, k\in K$$

$$\sum_{i=1}^{6}V_{ik}=1 \qquad \forall k\in K$$

$$\sum_{k=1}^{4}V_{ik}\le 1 \qquad \forall i\in I$$

$$\sum_{i=1}^{6}\sum_{k=1}^{4}L_{ijk}=1 \qquad \forall j\in J$$

$$\sum_{i=1}^{6}\sum_{j=1}^{9}L_{ijk}\ge 1 \qquad \forall k\in K$$

$$\text{Precedence}=\begin{cases}\sum_{i=1}^{6}L_{igk}\le\sum_{i=1}^{6}\sum_{h=1}^{4}L_{ifh} & \forall k\in K \quad \text{if} \quad \Pr_{fg}=1\\ \text{No const.} & \text{if} \quad \Pr_{fg}=0\end{cases} \qquad \forall k\in K$$

The model for the 4 station case is solved using LINGO Optimizer 12.0 and the optimal result for 4 station case was obtained. The number of replications is calculated using the replication size formula mentioned in Equation 5.17. The 4 station case was simulated using Delmia QUEST software for 11 replications and the average throughput for 2880 minutes is 66.85 products (Table 5.6).

**Table 5.6 Results – 4-Station-9-Task–6-Worker Case Study**

| STATION | | Station 1 | Station 2 | Station 3 | Station 4 | Throughput |
|---|---|---|---|---|---|---|
| 4 STATION | TASKS | Task 5<br>Task 7 | Task 3<br>Task 4 | Task 6<br>Task 8 | Task 1<br>Task 2<br>Task 9 | 66.85 |
| | WORKER | Worker 2 | Worker 6 | Worker 1 | Worker 3 | |

Since the average throughput obtained for 4 stations is not enough to meet the demand of 70 products, the number of stations is increased by one and the same case study is solved in LINGO optimizer for 5 stations. The LINGO model for 5 station case is shown below.

*LINGO model for* 4 station, 9 task and 6 worker case study*:*

*Objective function:*

$$Min\sum_{k=1}^{5}\left[RP_k * DP_k + RQ_k * QP_k + CL_k + \sum_{l=k+1}^{5}\left[\left(\left|RP_k - RP_l\right| * 1000\right) + \left(\left|RQ_k - RQ_l\right| * 1000\right) + \left(\left|SP_k - SP_l\right| * 1000\right)\right]\right]$$

*Wherein*

$$RP_k = \left[1 - \frac{1}{2}\left[1 + erf\left(\frac{S_k - SP_k}{SP\sigma_k\sqrt{2}}\right)\right]\right] \qquad \forall k \in K$$

$$RQ_k = \frac{1}{2}\left[1 + erf\left(\frac{Q_k - SQ_k}{SQ\sigma_k\sqrt{2}}\right)\right] \qquad \forall k \in K$$

$$SP_k = \sum_{j=1}^{9}\sum_{i=1}^{6} P_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SP\sigma^2{}_k = \sum_{j=1}^{9}\sum_{i=1}^{6} P\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SQ_k = \sum_{j=1}^{9}\sum_{i=1}^{6} Q_{ij} * L_{ijk} \qquad \forall k \in K$$

$$SQ\sigma^2{}_k = \sum_{j=1}^{9}\sum_{i=1}^{6} Q\sigma^2{}_{ij} * L_{ijk} \qquad \forall k \in K$$

$$S_k = \sum_{j=1}^{9}\sum_{i=1}^{6} STP_j * L_{ijk} \qquad \forall k \in K$$

$$Q_k = \sum_{j=1}^{9}\sum_{i=1}^{6} STQ_j * L_{ijk} \qquad \forall k \in K$$

$$CL_k = \sum_{j=1}^{9}\sum_{i=1}^{6}\left(C_{ij} * \frac{T}{60}\right) * L_{ijk} \qquad \forall k \in K$$

*Subject to*

$$\sum_{j=1}^{9}\left[L_{ijk}\right] = \sum_{i=1}^{6}\sum_{j=1}^{9}\left[L_{ijk} * V_{ik}\right] \qquad \forall i \in I, k \in K$$

$$\sum_{i=1}^{6} V_{ik} = 1 \qquad \forall k \in K$$

$$\sum_{k=1}^{5} V_{ik} \leq 1 \qquad \forall i \in I$$

$$\sum_{i=1}^{6}\sum_{k=1}^{5} L_{ijk} = 1 \qquad \forall j \in J$$

$$\sum_{i=1}^{6}\sum_{j=1}^{9} L_{ijk} \geq 1 \qquad \forall k \in K$$

$$\text{Precedence} = \begin{cases} \sum_{i=1}^{6} L_{igk} \leq \sum_{i=1}^{6}\sum_{h=1}^{5} L_{ifh} & \forall k \in K \quad \textit{if} \quad \Pr_{fg} = 1 \\ \textit{No const.} & \textit{if} \quad \Pr_{fg} = 0 \end{cases} \qquad \forall \mathrm{k} \in \mathrm{K}$$

The model for 5 station case is solved using LINGO Optimizer 12.0 and the optimal result for 5 station case is obtained. The number of replications is calculated using the replication size formula mentioned in Equation 5.17. The 4 station case is simulated using Delmia QUEST software for 25 replications and the average throughput for 2880 minutes is 73.79 products (Table 5.7).

**Table 5.7 Results – 5-Station-9-Task–6-Worker Case Study**

| **STATION** | | **1** | **2** | **3** | **4** | **5** | **Throughput** |
|---|---|---|---|---|---|---|---|
| **5 STATION** | **Tasks** | Task 7<br>Task 9 | Task 2<br>Task 5 | Task 8 | Task 3<br>Task 6 | Task 1<br>Task 4 | 73.79 |
| | **Worker** | Worker 6 | Worker 3 | Worker 2 | Worker 4 | Worker 5 | |

Since the 5 station configuration's throughput is greater than the demand of 70 products, the 5 station is the optimal number of stations for the 9-tasks-6-worker case study.

## 5.7 Conclusion

In this chapter, the necessity for a simultaneous approach for balancing assembly line and worker allocation was presented. The two kinds of risks considered in this research are the processing time risk and the quality level risk. Processing time risk was the increase in delay penalty cost due to uncertainty in worker processing time. Quality risk was the increase in quality penalty cost due to uncertainty associated with worker quality level. The proposed approach simultaneously minimizes the risk due to worker, the difference in risks between stations and the difference in mean workload of stations. A methodology to determine the optimal number of stations in an assembly line was also presented. Non Linear Integer Programming (NLIP) model, which simultaneously balances and allocates the best worker to the workstation in a multiple tasks per station production line has been developed.

# CHAPTER 6

# CONCLUSION AND FUTURE RESEARCH

## 6.1 Conclusion

In general, a manufacturing system can be classified into machine intensive manufacturing and labor intensive manufacturing. In labor intensive manufacturing, worker allocation plays an important role in determining the productivity and efficiency of the production line. Worker allocation problem deals with allocation of optimal worker to the process to maximize the efficiency of the production line. Worker allocation is performed based on worker characteristics such as worker processing time and worker quality level. Previous researches in worker allocation could be classified into sole profile allocation and multi profile allocation. In sole profile allocation, all the workers within the system are assumed to have equal processing time and quality level. In multi profile allocation, individual workers are assumed to have different processing times and quality level for the same task. Multi profile allocation is a more realistic approach since it takes into consideration the inherent differences between the workers.

From the time study data obtained from a local aircraft manufacturing company, it was evident that a high degree of variability exists in worker processing times (Table 1.3). Thus, variability in worker processing time and quality is more common in labor intensive manufacturing systems. In all previous approaches for worker allocation, variability in worker processing time and quality level is not addressed. If the worker allocation is done assuming a deterministic processing time and quality level for workers, it may degrade the performance of the line. Thus, a research void was found that there was no efficient worker allocation methodology in the literature, which takes into account the uncertainty in worker processing time and quality level. Hence, the objective of this research was to model and solve risk based worker allocation methodology in which variability in worker processing time and quality was also taken into consideration.

In general, production lines can be classified into: a) balanced line and unbalanced lines or b) production lines with single task per station and multiple tasks per station. Thus, risk based worker allocation methodology was applied to three different scenarios: single task per station balanced line, single task per station unbalanced line, and multiple tasks per station scenarios. Two kinds of risks considered in this research are processing time risk and quality level risk. Processing time risk is defined as the probability that the workers' processing time will exceed the standard processing time. Quality level risk is the probability that the workers' quality level is below standard quality level.

In Chapter 3, risk based worker allocation methodology was modeled and solved for single task per station balanced line and single task per station unbalanced line scenarios. In single product balanced line scenario, a NLIP model was developed which allocates the best worker to a station, thereby minimizing the overall risk in the production line. A deterministic worker allocation methodology for single task per station balanced line scenario was also developed as a benchmark for the problem being addressed. Then, risk based worker allocation for balanced line is validated against deterministic worker allocation methodology, in which risk based methodology outperformed deterministic methodology with an increased throughput. In single task per station unbalanced line scenario, in addition to allocation of best worker to the process, it also allocates multiple workers to the processes in order to balance the bottleneck stations and to meet the demand. The methodology that was developed was validated using simulation.

In a multiple task per station scenario, the productivity is dependent on both worker allocation and line balancing. Thus, in chapter 4, an approach for risk based line balancing was first developed. Results from risk based line balancing were compared with rank positional weight method for the same case study and an improvement in throughput is observed. In chapter 5, the methodology from chapter 4 was expanded to include worker allocation along with line balancing. The simultaneous approach, which balances the risk between the stations and allocates best worker to the station was developed. This methodology also determines the optimal number of stations in multiple task per station production line. Thus in this research, risk based worker allocation methodology was developed and solved for three different scenarios.

## 6.2 Intellectual Merit

The contributions of this research to the literatures in the field of worker allocation and line balancing are as follows:

- Although worker allocation problem was extensively investigated assuming different scenarios in the literature, all previous research assumed a deterministic test bed. Assuming a fixed worker profile, especially in low-volume-high-variability production systems, may lead to solutions that are not optimal and can be detrimental to the successful allocation of workers. This is the first research that has incorporated the stochastic nature of worker processing time and quality level to the worker allocation problem. This along with the use of simulation to determine the best worker allocation strategy represents a significant addition to the body of knowledge in terms of its application to real world manufacturing scenarios.
- In the existing line balancing approaches, the methodology was to balance the mean processing time between stations by assuming deterministic task processing

times. This is the first research that has developed an optimization approach for balancing mean processing time and risk between stations in an uncertain scenario. This takes into consideration the expected time to manufacture in highly variable systems and helps in determining optimal allocation of tasks.

- As an extension of the previous methodologies in this research, worker allocation problem was further extended to simultaneous worker allocation and line balancing problem in an uncertain multiple task per station production line. This is very relevant to aircraft industry wherein the allocation of tasks and workers play a critical role in determining the expected completion time
- The research developed in this thesis is more relevant to low-volume-high-variability production systems. If variability is low, existing strategies may provide reasonably good solutions. However, when dealing with highly variable systems such as aircraft industry or ship manufacturing, the research methodologies developed in this research will be extremely relevant.

This research will serve as a basis to extend the worker allocation problem in various dimensions. Some of the possible extensions of this research are presented in the following section.

## 6.3 Future Research

Risk based worker allocation approach was shown to be an efficient tool to incorporate uncertainty in worker characteristics into worker allocation problem. The problem considered for this research is a single product problem. The methodologies for balancing product lines and assigning workers when multiple products are involved are tedious and currently there are no methods that can address this issue. The current research could be used a starting point to developing multi-product line-balancing and worker allocation issues.

The methodology used in this research was to develop optimization approaches to the worker allocation problem. The largest case study that was conducted in this research is with 5 processes and 15 workers. Since, the formulation is a NLIP, as the problem size increases, the formulation and computational time increases exponentially. Hence, new heuristics that can adapt to large size problems have to be developed. Some of the heuristics that can be applied for larger size problems are genetic algorithms, tabu-search, ANT Colony algorithms etc. The efficiency of each of these heuristics and the implementation should be investigated.

In the current research, the worker profiles are assumed to be static. However, with the learning curve effect, the worker profile can be made dynamic. When including dynamic worker profile the problem of worker allocation also becomes dynamic and the improvements in workers have to be taken into consideration in the solution of worker allocation problems. However, this could make the problem size larger and will require more aggressive heuristics to be developed.

# REFERENCES

## REFERENCES


Aryanezhad, M, B., Deljoo, V., and Al-e-hashem, S, M, J, M., (2009), "Dynamic cell formation and worker assignment problem: a new model," *International Journal of Advanced Manufacturing Technology*, Vol. 41, No. 3-4, pp. 329-342.

Askin, R, G., and Huang, Y., (2001), "Forming effective worker teams for cellular manufacturing," *International Journal of Production Research*, Vol.39, No. 11, pp. 2431-2451.

Bhaskar, K., and Srinivasan, G., (1997), "Static and dynamic operator allocation problems in cellular manufacturing systems," *International Journal of Production Research*, Vol. 35, No. 12, pp. 3467-3481.

Chaves, A, A., Insa, C, M., and Lorena, L, A., (2007), "Clustering search approach for the assembly line worker assignment and balancing problem," *Proceedings of the 37th International Conference on Computers and Industrial Engineering,-*Alexandria, Egypt., pp. 1469-1478.

Davis, D. J., and Mabert, V. A., (2000), "Order dispatching and labor assignment in cellular manufacturing systems," *Decision Sciences*, Vol. 31, No. 4, pp. 745-770.

Ertay, T., and Ruan, D., (2004), "Data envelopment analysis based decision model for optimal operator allocation in CMS," *European Journal of Operational Research*, Vol.164, No. 3, pp. 800-810.

Fitzpatrick, E, L., and Askin, R, G., (2005), "Forming effective worker teams with multi-functional skill requirements," *Computers and Industrial Engineering*, Vol. 48, No. 3, pp.593-608.

Fowler, J, W., Wirojanagud, P., and Gel, E., (2008), "Heuristics for workforce planning with worker differences," *European Journal of Operational Research*, Vol. 190, No. 3, pp. 724-740.

Heimerl, C., and Kolisch, R., (2009), "Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets," *International Journal of Production Research*, Vol. 48, No. 13, pp. 3759-3781.

Kuo, Y., and Yang, T., (2006), "A case study on the operator allocation decision for TFT-LCD inspection and packaging process," *Journal of Manufacturing Technology Management*, Vol. 17, No. 3, pp. 363-375.

Mcdonald, T., Ellis, K, P., Aken, E, M, V., and Koelling, C, P., (2009), "Development and application of a worker assignment model to evaluate a lean manufacturing cell," *International Journal of Production Research*, Vol.47, No. 9, pp. 2427-2447.

Min, H., and Shin, D., (1993), "Simultaneous formation of machine and human cells in group technology: a multiple objective approach," *International Journal of Production Research*, Vol. 31, No. 10, pp. 2307-2318.

Miralles, C., Garcia, J, P., Andres, C., and Cardos, M., (2008), “Branch and bound procedures for solving the assembly line worker assignment and balancing problem. Application to sheltered work centers for disabled,” *Discrete Applied Mathematics*, Vol. 156, No. 3, pp. 352-367.

Nakade, K., and Ohno, K., (1999), “An optimal worker allocation problem for a U-shaped production line,” *International Journal of Production Economics*, Vol. 60-61, pp 353-358.

Nembhard, D, A., (2001), “Heuristic approach for assigning worker to tasks based on individual learning rates,” *International Journal of Production Research*, Vol. 39, No. 9, pp. 1955-1968.

Norman, B, A., Tharmmaphornphilas, W., Needy, K, L., Bidanda, B., and Warner, R, C., (2002), “Worker assignment in cellular manufacturing considering technical and human skills,” *International Journal of Production Research*, Vol. 40, No. 6, pp. 1479-1492.

Suer, G, A., and Tummaluri, R, R., (2008), “Multi-period operator assignment consideringskills, learning and forgetting in labour –intensive cells,” *International Journal of Production Research*, Vol. 46, No. 2, pp. 469-493.

Suer, G. A., (1996), “Optimal operator assignment and cell loading in labor-intensive manufacturing cells,” *Computers and Industrial Engineering*, Vol. 31, No. ½, pp. 155-158.

Suer, G. A., and Bera, I. S., (1998), “Optimal operator assignment and cell loading when lot-splitting is allowed,” *Computers and Industrial Engineering*, Vol. 35, No. 3-4, pp.431-434.

Vembu, S., and Srinivasan, G., (1997), “Heuristics for operator allocation and sequencing in product-line-cells with manually operated machines,” *Computers and Industrial Engineering*, Vol. 32, No. 2, pp. 265-279.

Wittrock, R. J., (1992), “Operator assignment and the parametric preflow algorithm,” *Management Science*, Vol. 38, No. 9, pp.1354-1359.

Yakoob, S, B., and Watada, J., (2009), “Particle swarm optimization for multi-function worker assignment problem,” *Knowledge-Based and Intelligent Information and Engineering Systems*, Vol. 5712/2009, pp. 203-211.

# APPENDICES

## Appendix A

### LINGO model for worker allocation without considering uncertainty -3-process-6-worker

```
model:
max =((((t/P11)*pr1*L11)-(((1-q11)/100)*(t/P11)*qr1*L11)-(c11*L11))
+(((t/P13)*pr3*L13)-(((1-q13)/100)*(t/P13)*qr3*L13)-(c13*L13))
+(((t/P21)*pr1*L21)-(((1-q21)/100)*(t/P21)*qr1*L21)-(c21*L21))
+(((t/P22)*pr2*L22)-(((1-q22)/100)*(t/P22)*qr2*L22)-(c22*L22))
+(((t/P32)*pr2*L32)-(((1-q32)/100)*(t/P32)*qr2*L32)-(c32*L32))
+(((t/P33)*pr3*L33)-(((1-q33)/100)*(t/P33)*qr3*L33)-(c33*L33))
+(((t/P41)*pr1*L41)-(((1-q41)/100)*(t/P41)*qr1*L41)-(c41*L41))
+(((t/P43)*pr3*L43)-(((1-q43)/100)*(t/P43)*qr3*L43)-(c43*L43))
+(((t/P51)*pr1*L51)-(((1-q51)/100)*(t/P51)*qr1*L51)-(c51*L51))
+(((t/P52)*pr2*L52)-(((1-q52)/100)*(t/P52)*qr2*L52)-(c52*L52))
+(((t/P62)*pr2*L62)-(((1-q62)/100)*(t/P62)*qr2*L62)-(c62*L62))
+(((t/P63)*pr3*L63)-(((1-q63)/100)*(t/P63)*qr3*L63)-(c63*L63)));
P11=8.8;
P13=9.2;
P21=8.9;
P22=8.9;
P32=8.8;
P33=8.9;
P41=9.0;
P43=8.9;
P51=9.1;
P52=9.1;
P62=8.9;
P63=8.8;
q11=98.9;
q13=99.2;
q21=99.3;
q22=98.7;
q32=98.3;
q33=98.8;
q41=99.0;
q43=99.1;
q51=98.8;
q52=98.6;
q62=99.3;
q63=99.2;
c11=480;
c13=450;
c21=420;
c22=480;
c32=570;
c33=510;
c41=510;
c43=450;
c51=570;
c52=480;
c62=570;
c63=660;
pr1=36;
qr1=33;
pr2=33;
```

```
qr2=36;
pr3=37.5;
qr3=39;
t=2880;
@bin(L11);
@bin(L12);
@bin(L13);
@bin(L21);
@bin(L22);
@bin(L23);
@bin(L31);
@bin(L32);
@bin(L33);
@bin(L41);
@bin(L42);
@bin(L43);
@bin(L51);
@bin(L52);
@bin(L53);
@bin(L61);
@bin(L62);
@bin(L63);
L11+L13<=1;
L21+L22<=1;
L32+L33<=1;
L41+L43<=1;
L51+L52<=1;
L62+L63<=1;
L11+L13+L21+L22+L32+L33+L41+L43+L51+L52+L62+L63<=6;
L11+L21+L41+L51=1;
L22+L32+L52+L62=1;
L13+L33+L43+L63=1;
```

## Appendix B

### LINGO model for worker allocation considering uncertainty -3-process-6-worker

```
SETS:
Labor/1 2 3 4 5 6/:Lab;
Process/1 2 3/:SP,SQ,QP,DP;
LP(Labor,Process):P,Q,PS,QS,RP,RQ,X,C;
END SETS

DATA:
SP= 9 9 9;
SQ= 0.990 0.990 0.990;
QP= 300 400 500;
DP= 450 500 320;
T= 2880;

!substituted 20 for 0;
P= 8.8 20 9.2
8.9 8.9 20
20 8.8 8.9
9.0 20 8.9
9.1 9.1 20
20 8.9 8.8;

!substituted 1 for 0;
PS= 1.05 1 0.39
0.33 0.66 1
1 1.11 0.68
0.87 1 0.23
0.48 0.18 1
1 0.36 0.98;

!substituted 0.50 for 0;
Q= 0.989 0.5 0.992
0.993 0.987 0.5
0.5 0.983 0.988
0.990 0.5 0.991
0.988 0.986 0.5
0.5 0.993 0.992;

!substituted 0.0001 for 0;
QS= 0.0022 0.0001 0.0011
0.0014 0.0021 0.0001
0.0001 0.0054 0.0025
0.0008 0.0001 0.0008
0.0019 0.0026 0.0001
0.0001 0.0018 0.0034;

!substituted 1000 for 0;
C= 10 1000 9.37
8.75 10 1000
1000 11.8 10.62
10.62 1000 9.37
```

```
11.87 10 1000
1000 11.87 13.75;

END DATA

@for(LP(i,j):@bin(X(i,j)));


min=@sum(Labor(i):
       @sum(Process(j):
                (RP(i,j)*DP(j)*X(i,j))+(RQ(i,j)*QP(j)*X(i,j))+
(C(i,j)*(T/60)*X(i,j))));



@for(Labor(i):
       @for(Process(j):
                   RP(i,j)=(1 - @PSN((SP(j)-P(i,j))/PS(i,j)));
                   RQ(i,j)=(@PSN((SQ(j)-Q(i,j))/QS(i,j)));
));


@for(Labor(i):@sum(Process(j):X(i,j))<=1);

@sum(Labor(i):@sum(Process(j):X(i,j)))<=5;

@for(Process(j):@sum(Labor(i):X(i,j))=1);
```

## Appendix C

### LINGO model for worker allocation considering uncertainty -4-process-8-worker

```
SETS:
Labor/1 2 3 4 5 6 7 8/:Lab;
Process/1 2 3 4/:SP,SQ,QP,DP;
LP(Labor,Process):P,Q,PS,QS,RP,RQ,X,C;
END SETS

DATA:
SP= 14 14 14 14;
SQ= 0.985 0.985 0.985 0.985;
QP= 550 600 650 700;
DP= 650 625 725 750;
T= 2880;

!substituted 20 for 0;
P= 13.7 20 13.4 20
20 14.2 20 13.6
13.5 20 13.3 20
20 13.8 20 13.7
13.8 20 13.5 20
20 14 20 13.9
14.1 20 13.2 20
20 13.9 20 13.8;

!substituted 1 for 0;
PS= 0.62 1 0.85 1
1 0.58 1 0.66
0.53 1 0.92 1
1 0.76 1 0.48
0.44 1 0.63 1
1 0.56 1 0.42
0.35 1 1.0 1
1 0.82 1 0.53;

!substituted 0.50 for 0;
Q= 0.986 0.5 0.991 0.5
0.5 0.98 0.5 0.986
0.99 0.5 0.984 0.5
0.5 0.983 0.5 0.980
0.985 0.5 0.992 0.5
0.5 0.987 0.5 0.975
0.97 0.5 0.987 0.5
0.5 0.982 0.5 0.988;

!substituted 0.0001 for 0;
QS= 0.0016 0.0001 0.0026 0.0001
0.0001 0.0026 0.0001 0.0023
0.0033 0.0001 0.0013 0.0001
0.0001 0.002 0.0001 0.003
0.002 0.0001 0.0016 0.0001
0.0001 0.0026 0.0001 0.0083
0.1 0.0001 0.0028 0.0001
0.0001 0.0042 0.0001 0.0028;
```

```
!substituted 1000 for 0;
C= 15 1000 8 1000
1000 13 1000 10
12 1000 14 1000
1000 10 1000 12
10 1000 15 1000
1000 13 1000 14
17 1000 16 1000
1000 11 1000 9;

END DATA

@for(LP(i,j):@bin(X(i,j)));


min=@sum(Labor(i):
       @sum(Process(j):
                (RP(i,j)*DP(j)*X(i,j))+(RQ(i,j)*QP(j)*X(i,j))+
(C(i,j)*(T/60)*X(i,j))));



@for(Labor(i):
       @for(Process(j):
                   RP(i,j)=(1 - @PSN((SP(j)-P(i,j))/PS(i,j)));
                   RQ(i,j)=(@PSN((SQ(j)-Q(i,j))/QS(i,j)));
));


@for(Labor(i):@sum(Process(j):X(i,j))<=1);

@sum(Labor(i):@sum(Process(j):X(i,j)))<=7;

@for(Process(j):@sum(Labor(i):X(i,j))=1);
```

## Appendix D

## LINGO model for worker allocation unbalanced line -5-process-10-worker

```
SETS:
Labor/1 2 3 4 5 6 7 8 9 10/:Lab;
Process/1 2 3 4 5/:SP,SQ,QP,DP;
LP(Labor,Process):P,Q,PS,QS,RP,RQ,X,C;
END SETS

DATA:
SP= 4 8 6 3 2;
SQ= 0.980 0.980 0.980 0.980 0.980;
QP= 300 400 500 600 700;
T= 2880;
D=600;
U=40;

!substituted 20 for 0;
P= 3.5 20 6.2 20 20
20 7.0 20 3.0 20
4.5 20 20 20 1.5
20 7.5 20 3.5 20
20 20 5.8 20 2.0
4.2 20 6.0 20 20
20 6.8 20 2.8 20
4.0 20 20 20 2.2
20 6.5 20 2.5 20
20 20 6.5 20 2.5;

!substituted 1 for 0;
PS= 0.16 1 0.1 1 1
1 0.6 1 0.16 1
0.06 1 1 1 0.33
1 0.33 1 0.16 1
1 1 0.2 1 0.2
0.6 1 0.16 1 1
1 0.16 1 0.2 1
0.13 1 1 1 0.06
1 0.26 1 0.33 1
1 1 0.16 1 0.16;

!substituted 0.50 for 0;
Q= 0.980 0.5 0.990 0.5 0.5
0.5 0.980 0.5 0.990 0.5
0.990 0.5 0.5 0.5 0.970
0.5 0.985 0.5 0.995 0.5
0.5 0.5 0.980 0.5 0.980
0.985 0.5 0.980 0.5 0.5
0.5 0.995 0.5 0.980 0.5
0.980 0.5 0.5 0.5 0.985
0.5 0.995 0.5 0.986 0.5
0.5 0.5 0.988 0.5 0.990;

!substituted 0.0001 for 0;
QS= 0.0010 0.0001 0.0016 0.0001 0.0001
```

```
0.0001 0.0050 0.0001 0.0010 0.0001
0.0013 0.0001 0.0001 0.0001 0.0050
0.0001 0.0033 0.0001 0.0016 0.0001
0.0001 0.0001 0.0006 0.0001 0.0016
0.0010 0.0001 0.0016 0.0001 0.0001
0.0001 0.0010 0.0001 0.0013 0.0001
0.0016 0.0001 0.0001 0.0001 0.0033
0.0001 0.0013 0.0001 0.0013 0.0001
0.0001 0.0001 0.0016 0.0001 0.0033;

!substituted 1000 for 0;
C= 12.5 1000 13.33 1000 1000
1000 11.66 1000 15 1000
13.33 1000 1000 1000 14.16
1000 15.83 1000 12.50 1000
1000 1000 11.66 1000 15
14.16 1000 13.33 1000 1000
1000 18.33 1000 14.16 1000
12.5 1000 1000 1000 13.33
1000 18.33 1000 15.83 1000
1000 1000 13.33 1000 14.16;

END DATA

!X(i,j) is a binary variable for labor selection;
@for(LP(i,j):@bin(X(i,j)));


min=@sum(Labor(i):
       @sum(Process(j):
                (RP(i,j)*DP(j)*X(i,j))+(RQ(i,j)*QP(j)*X(i,j))+
(C(i,j)*(T/60)*X(i,j))));



@for(Labor(i):
       @for(Process(j):
                   RP(i,j)=(1 - @PSN((SP(j)-P(i,j))/PS(i,j)));
                   RQ(i,j)=(@PSN((SQ(j)-Q(i,j))/QS(i,j)));
));


@for(Labor(i):@sum(Process(j):X(i,j))<=1);

@sum(Labor(i):@sum(Process(j):X(i,j)))<=10;

@for(Process(j):
    @sum(Labor(i):
               ((Q(i,j)-(3*QS(i,j)))/(P(i,j)+(3*PS(i,j))))*X(i,j))>(D/T));

@for(Process(j):
       DP(j)=(1/(@sum(Labor(i):(X(i,j)/P(i,j)))-(D/T)))*U);
```

## Appendix E

## LINGO model for worker allocation unbalanced line -5-process-15-worker

```
SETS:
Labor/1 2 3 4 5 6 7 8 9 10 11 12 13 14 15/:Lab;
Process/1 2 3 4 5/:SP,SQ,QP,DP;
LP(Labor,Process):P,Q,PS,QS,RP,RQ,X,C;
END SETS

DATA:
SP= 3 5 14 9 7;
SQ= 0.990 0.990 0.990 0.990 0.990;
QP= 300 400 500 600 700;
T= 2880;
D=500;
U=30;

!substituted 20 for 0;
P= 2.8 20 13.5 20 20
20 5.1 20 8.9 20
2.9 20 20 20 6.6
20 4.8 20 8.8 20
20 20 13.8 20 6.8
2.5 20 14.1 20 20
20 4.6 20 9.1 20
3.0 20 20 20 7.2
20 4.7 20 8.7 20
20 20 13.7 20 7.0
3.1 20 13.9 20 20
20 5.0 20 8.8 20
2.7 20 20 20 6.9
20 5.2 9.0 20 20
20 20 14.0 20 6.7;

!substituted 1 for 0;
PS= 0.17 1 0.33 1 1
1 0.07 1 0.13 1
0.13 1 1 1 0.20
1 0.20 1 0.20 1
1 1 0.23 1 0.23
0.33 1 0.17 1 1
1 0.27 1 0.07 1
0.1 1 1 1 0.1
1 0.20 1 0.20 1
1 1 0.27 1 0.17
0.1 1 0.23 1 1
1 0.2 1 0.23 1
0.20 1 1 1 0.17
1 0.27 0.17 1 1
1 1 0.13 1 0.27;

!substituted 0.50 for 0;
Q= 0.980 0.5 0.986 0.5 0.5
0.5 0.987 0.5 0.990 0.5
0.992 0.5 0.5 0.5 0.988
```

```
0.5 0.985 0.5 0.983 0.5
0.5 0.5 0.985 0.5 0.992
0.975 0.5 0.988 0.5 0.5
0.5 0.990 0.5 0.985 0.5
0.988 0.5 0.5 0.5 0.990
0.5 0.970 0.5 0.995 0.5
0.5 0.5 0.991 0.5 0.980
0.990 0.5 0.995 0.5 0.5
0.5 0.988 0.5 0.987 0.5
0.991 0.5 0.5 0.5 0.989
0.5 0.987 0.5 0.993 0.5
0.5 0.5 0.989 0.5 0.991;

!substituted 0.0001 for 0;
QS= 0.0067 0.0001 0.0020 0.0001 0.0001
0.0001 0.0020 0.0001 0.0017 0.0001
0.0003 0.0001 0.0001 0.0001 0.0090
0.0001 0.0033 0.0001 0.0053 0.0001
0.0001 0.0001 0.0033 0.0001 0.0010
0.0100 0.0001 0.0017 0.0001 0.0001
0.0001 0.0017 0.0001 0.0033 0.0001
0.0020 0.0001 0.0001 0.0001 0.0017
0.0001 0.0100 0.0001 0.0020 0.0001
0.0001 0.0001 0.0010 0.0001 0.0050
0.0023 0.0001 0.0020 0.0001 0.0001
0.0001 0.0027 0.0001 0.0020 0.0001
0.0013 0.0001 0.0001 0.0001 0.0010
0.0001 0.0033 0.0001 0.0020 0.0001
0.0001 0.0001 0.0023 0.0001 0.0013;

!substituted 1000 for 0;
C= 10.87 1000 12.87 1000 1000
1000 9.43 1000 9.25 1000
12.87 1000 1000 1000 11.00
1000 9.56 1000 12.37 1000
1000 1000 13.12 1000 10.75
13.25 1000 11.93 1000 1000
1000 10.25 1000 11.62 1000
9.37 1000 1000 1000 12.81
1000 11.50 1000 9.43 1000
1000 1000 11.06 1000 9.93
9.12 1000 13.06 1000 1000
1000 10.93 1000 10.18 1000
12.37 1000 1000 1000 10.12
1000 13.56 1000 8.43 1000
1000 1000 11.56 1000 12.00;

END DATA

!X(i,j) is a binary variable for labor selection;
@for(LP(i,j):@bin(X(i,j)));


min=@sum(Labor(i):
       @sum(Process(j):
                (RP(i,j)*DP(j)*X(i,j))+(RQ(i,j)*QP(j)*X(i,j))+
(C(i,j)*(T/60)*X(i,j))));
```

```
@for(Labor(i):
       @for(Process(j):
                   RP(i,j)=(1 - @PSN((SP(j)-P(i,j))/PS(i,j)));
                   RQ(i,j)=(@PSN((SQ(j)-Q(i,j))/QS(i,j)));
));


@for(Labor(i):@sum(Process(j):X(i,j))<=1);

@sum(Labor(i):@sum(Process(j):X(i,j)))<=15;

@for(Process(j):
    @sum(Labor(i):
               ((Q(i,j)-(3*QS(i,j)))/(P(i,j)+(3*PS(i,j))))*X(i,j))>(D/T));

@for(Process(j):
       DP(j)=(1/(@sum(Labor(i):(X(i,j)/P(i,j)))-(D/T)))*U);
```

## Appendix F

## LINGO model for risk based line balancing-9-task-3-station

```
SETS:
task/1 2 3 4 5 6 7 8 9/:sp,pm,ps;
station/1 2 3/:RP,ST,STS,S;
TS(task, station): X;
LT(task);
STN(station);
TSK(task);
endsets
Data:
sp= 15 26 8 12 14 10 23 28 16;
pm= 13 25.8 7.6 11.1 14.4 10.2 22.8 27.85 15.8;
ps= 1.56 1.34 1.15 0.94 1.22 1.14 0.85 0.66 0.96;
RBP= 1000;
End Data

@for(TS:@bin(X));

min = RBP*((@ABS(RP(1)-RP(2)))+(@ABS(RP(1)-RP(3)))+(@ABS(RP(2)-RP(3)))
+ (@ABS(ST(1)-ST(2)))+(@ABS(ST(1)-ST(3)))+(@ABS(ST(2)-ST(3)))) ;

@for(station(k):
ST(k) = @sum(LT(j): pm(j)*X(j,k)) ;
STS(k) = (@sum(LT(j): ps(j)^2*X(j,k)))^0.5 ;
S(k) = @sum(LT(j):sp(j)*X(j,k)) ;
RP(k) = (1 - @PSN((S(k)-ST(k))/STS(k))) ;
) ;

@for(task(j):
@sum(STN(k):X(j,k))=1
) ;

@for(station(k):
@sum(TSK(j):X(j,k))>=1
) ;

X(7,1)<=X(3,1)+X(3,2)+X(3,3);
X(7,2)<=X(3,2)+X(3,3);
X(7,3)<=X(3,3);

X(9,1)<=X(1,1)+X(1,2)+X(1,3);
X(9,2)<=X(1,2)+X(1,3);
X(9,3)<=X(1,3);
```

# Appendix G

## LINGO model for risk based line balancing-9-task-4-station

```
SETS:
task/1 2 3 4 5 6 7 8 9/:sp,pm,ps;
station/1 2 3 4/:RP,ST,STS,S;
TS(task, station): X;
LT(task);
STN(station);
TSK(task);
endsets
Data:
sp= 15 26 8 12 14 10 23 28 16;
pm= 13 25.8 7.6 11.1 14.4 10.2 22.8 27.85 15.8;
ps= 1.56 1.34 1.15 0.94 1.22 1.14 0.85 0.66 0.96;
RBP= 1000;
End Data

@for(TS:@bin(X));

min = RBP*((@ABS(RP(1)-RP(2)))+(@ABS(RP(1)-RP(3)))+(@ABS(RP(2)-RP(3)))
+ (@ABS(ST(1)-ST(2)))+(@ABS(ST(1)-ST(3)))+(@ABS(ST(2)-ST(3)))) ;

@for(station(k):
ST(k) = @sum(LT(j): pm(j)*X(j,k)) ;
STS(k) = (@sum(LT(j): ps(j)^2*X(j,k)))^0.5 ;
S(k) = @sum(LT(j):sp(j)*X(j,k)) ;
RP(k) = (1 - @PSN((S(k)-ST(k))/STS(k))) ;
) ;

@for(task(j):
@sum(STN(k):X(j,k))=1
) ;

@for(station(k):
@sum(TSK(j):X(j,k))>=1
) ;

X(7,1)<=X(3,1)+X(3,2)+X(3,3)+X(3,4);
X(7,2)<=X(3,2)+X(3,3)+X(3,4);
X(7,3)<=X(3,3)+X(3,4);
X(7,4)<=X(3,4);

X(9,1)<=X(1,1)+X(1,2)+X(1,3)+X(1,4);
X(9,2)<=X(1,2)+X(1,3)+X(1,4);
X(9,3)<=X(1,3)+X(1,4);
X(9,4)<=X(1,4);
```

## Appendix H

### LINGO model for risk based line balancing-9-task-5-station

```
SETS:
task/1 2 3 4 5 6 7 8 9/:sp,pm,ps;
station/1 2 3 4 5/:RP,ST,STS,S;
TS(task, station): X;
LT(task);
STN(station);
TSK(task);
endsets
Data:
sp= 15 26 8 12 14 10 23 28 16;
pm= 13 25.8 7.6 11.1 14.4 10.2 22.8 27.85 15.8;
ps= 1.56 1.34 1.15 0.94 1.22 1.14 0.85 0.66 0.96;
RBP= 1000;
End Data

@for(TS:@bin(X));

min = RBP*((@ABS(RP(1)-RP(2)))+(@ABS(RP(1)-RP(3)))+(@ABS(RP(2)-RP(3)))
+ (@ABS(ST(1)-ST(2)))+(@ABS(ST(1)-ST(3)))+(@ABS(ST(2)-ST(3)))) ;

@for(station(k):
ST(k) = @sum(LT(j): pm(j)*X(j,k)) ;
STS(k) = (@sum(LT(j): ps(j)^2*X(j,k)))^0.5 ;
S(k) = @sum(LT(j):sp(j)*X(j,k)) ;
RP(k) = (1 - @PSN((S(k)-ST(k))/STS(k))) ;
) ;

@for(task(j):
@sum(STN(k):X(j,k))=1
) ;

@for(station(k):
@sum(TSK(j):X(j,k))>=1
) ;

X(7,1)<=X(3,1)+X(3,2)+X(3,3)+X(3,4)+X(3,5);
X(7,2)<=X(3,2)+X(3,3)+X(3,4)+X(3,5);
X(7,3)<=X(3,3)+X(3,4)+X(3,5);
X(7,4)<=X(3,4)+X(3,5);
X(7,5)<=X(3,5);

X(9,1)<=X(1,1)+X(1,2)+X(1,3)+X(1,4)+X(1,5);
X(9,2)<=X(1,2)+X(1,3)+X(1,4)+X(1,5);
X(9,3)<=X(1,3)+X(1,4)+X(1,5);
X(9,4)<=X(1,4)+X(1,5);
X(9,5)<=+X(1,5);
```

## Appendix I

## LINGO model for SRLW-9-task-6-worker-3-station

```
SETS:
labor/1 2 3 4 5 6/:lab;
task/1 2 3 4 5 6 7 8 9/:SP,ql;
station/1 2 3/:stn,DP,QP,RP,RQ,ST,STS,SQ,SQS,S,Q,L;
LT(labor,task):pm,ps,qm,qs,c,X;
TS(task,station):Y;
LS(labor, station):V;
LTS(labor, task, station): XY ;
endsets
CALC:
  ! Set the linearization on ;
  @APISET( 214, 'int', 3);
ENDCALC
Data:
SP= 15 8 26 12 23 10 14 28 16;
ql=0.98 0.985 0.99 0.98 0.97 0.975 0.99 0.98 0.995;
M=3;
T=2880;
D=140;
QP=375 350 400;
DP=450 550 500;
RBP=1000;

pm= 14.5 7.9 25.2 11.8 22.5 9.9 14.0 27.0 15.8
15.2 7.6 26.2 12.1 23.1 9.6 13.8 27.5 15.5
13.0 8.0 25.8 11.5 22.0 10.1 14.1 28.0 16.0
14.8 8.1 24.0 11.0 22.8 10.0 13.5 28.1 16.2
15.0 7.5 24.5 12.2 23.0 9.8 13.4 26.5 15.2
13.5 7.8 26.0 11.0 23.2 10.2 13.7 27.8 15.9;

ps=0.87 0.36 1.0 0.62 0.72 0.66 0.41 1.0 0.76
0.32 1.05 0.4 0.41 0.41 1.11 0.73 0.75 0.92
1.5 0.83 0.62 0.82 1.0 0.35 0.3 0.42 0.41
0.78 0.52 1.5 1.0 0.8 0.5 0.93 0.3 0.3
0.52 1.21 1.2 0.30 0.52 0.73 1.11 1.55 1.32
1.0 0.38 0.33 0.92 0.3 0.3 0.84 0.63 0.5;

qm=0.975 0.98 0.98 0.977 0.965 0.98 0.99 0.975 0.99
0.978 0.983 0.992 0.982 0.971 0.981 0.988 0.981 0.994
0.98 0.992 0.975 0.975 0.968 0.975 0.983 0.977 0.992
0.978 0.988 0.988 0.98 0.972 0.972 0.985 0.983 0.99
0.982 0.98 0.99 0.981 0.97 0.965 0.995 0.97 0.98
0.981 0.986 0.991 0.978 0.967 0.993 0.987 0.976 0.993;

qs=0.008 0.0026 0.0065 0.0034 0.0034 0.0033 0.0018 0.0033 0.0033
0.0024 0.0024 0.0003 0.0048 0.0018 0.0026 0.0038 0.0030 0.001
0.0053 0.0022 0.008 0.0026 0.0024 0.0032 0.0053 0.0021 0.0026
0.0032 0.003 0.002 0.005 0.0021 0.0026 0.0033 0.0024 0.0033
0.003 0.0066 0.0023 0.0021 0.003 0.0083 0.0015 0.0052 0.0066
0.003 0.0016 0.0013 0.0036 0.0041 0.0022 0.0024 0.0028 0.0022;

c=9.48 11.83 10.04 10.35 10.65 13.48 10.38 10.10 11.73
```

```
9.41 8.40 11.81 10.98 11.98 12.69 10.41 13.41 13.95
10.66 14.93 13.12 11.08 11.83 14.98 13.67 11.22 10.71
9.37 13.61 14.15 9.05 14.22 12.17 11.63 12.81 11.80
12.43 11.43 14.87 11.88 10.71 8.44 12.50 13.93 8.73
9.57 9.26 10.60 13.12 8.47 10.38 12.66 10.91 13.36;
enddata


@for(LT(i,j):@bin(X(i,j)));
@for(TS(j,k):@bin(Y(j,k)));
@for(LS(i,k):@bin(V(i,k)));
@for(LTS:@bin(XY));



min = (@sum(station(k):RP(k)*DP(k) + RQ(k)*QP(k) + L(k)) +
RBP*((@ABS(RP(1)-RP(2)))+(@ABS(RP(1)-RP(3)))+(@ABS(RP(2)-RP(3)))+(@ABS(RQ(1)-
RQ(2)))+(@ABS(RQ(1)-RQ(3)))
+(@ABS(RQ(2)-RQ(3))) + (@ABS(ST(1)-ST(2)))+(@ABS(ST(1)-ST(3)))+(@ABS(ST(2)-
ST(3))))) ;


@for(station(k):
ST(k) = @sum(LT(i,j): pm(i,j)*XY(i,j,k)) ;
STS(k) = (@sum(LT(i,j): ps(i,j)^2*XY(i,j,k)))^0.5 ;
SQ(k) = @sum(LT(i,j): qm(i,j)*XY(i,j,k)) ;
SQS(k) = (@sum(LT(i,j): qs(i,j)^2*XY(i,j,k)))^0.5 ;
S(k) = @sum(LT(i,j):sp(j)*XY(i,j,k)) ;
Q(k) = @sum(LT(i,j):ql(j)*XY(i,j,k)) ;
L(k) = @sum(LT(i,j):c(i,j)*(T/60)*XY(i,j,k)) ;
RP(k) = (1 - @PSN((S(k)-ST(k))/STS(k))) ;
RQ(k) = @PSN((Q(k)-SQ(k))/SQS(k)) ;
) ;


@for(station(k):
     @for(labor(i):
            @sum(task(j):XY(i,j,k))= (@sum(LT(i,j):XY(i,j,k)))*V(i,k));
     @sum(labor(i): V(i,k)) = 1 ;
);

@for(labor(i): @sum(station(k): V(i,k)) <= 1) ;
@for(task(j): @sum(LS(i,k): XY(i,j,k))=1);


   @sum(labor(i): XY(i,3,1))<= @sum(labor(i):XY(i,7,1));
   @sum(labor(i): XY(i,3,2))<= @sum(labor(i):XY(i,7,1)+XY(i,7,2));
   @sum(labor(i): XY(i,3,3))<= @sum(labor(i):XY(i,7,1)+XY(i,7,2)+XY(i,7,3));

   @sum(labor(i):XY(i,1,1))<= @sum(labor(i):XY(i,9,1));
   @sum(labor(i):XY(i,1,2))<= @sum(labor(i):XY(i,9,1)+XY(i,9,2));
   @sum(labor(i):XY(i,1,3))<= @sum(labor(i):XY(i,9,1)+XY(i,9,2)+XY(i,9,3));


END
```

## Appendix J

## LINGO model for SRLW-9-task-6-worker-4-station

```
SETS:
labor/1 2 3 4 5 6/:lab;
task/1 2 3 4 5 6 7 8 9/:SP,ql;
station/1 2 3 4/:stn,DP,QP,RP,RQ,ST,STS,SQ,SQS,S,Q,L;
LT(labor,task):pm,ps,qm,qs,c,X;
TS(task,station):Y;
LS(labor, station):V;
LTS(labor, task, station): XY ;
endsets
CALC:
  ! Set the linearization on ;
  @APISET( 214, 'int', 3);
ENDCALC
Data:
SP= 15 8 26 12 23 10 14 28 16;
ql=0.98 0.985 0.99 0.98 0.97 0.975 0.99 0.98 0.995;
M=4;
T=2880;
D=140;
QP=375 350 400 425;
DP=450 550 500 525;
RBP=1000;

pm= 14.5 7.9 25.2 11.8 22.5 9.9 14.0 27.0 15.8
15.2 7.6 26.2 12.1 23.1 9.6 13.8 27.5 15.5
13.0 8.0 25.8 11.5 22.0 10.1 14.1 28.0 16.0
14.8 8.1 24.0 11.0 22.8 10.0 13.5 28.1 16.2
15.0 7.5 24.5 12.2 23.0 9.8 13.4 26.5 15.2
13.5 7.8 26.0 11.0 23.2 10.2 13.7 27.8 15.9;

ps=0.87 0.36 1.0 0.62 0.72 0.66 0.41 1.0 0.76
0.32 1.05 0.4 0.41 0.41 1.11 0.73 0.75 0.92
1.5 0.83 0.62 0.82 1.0 0.35 0.3 0.42 0.41
0.78 0.52 1.5 1.0 0.8 0.5 0.93 0.3 0.3
0.52 1.21 1.2 0.30 0.52 0.73 1.11 1.55 1.32
1.0 0.38 0.33 0.92 0.3 0.3 0.84 0.63 0.5;

qm=0.975 0.98 0.98 0.977 0.965 0.98 0.99 0.975 0.99
0.978 0.983 0.992 0.982 0.971 0.981 0.988 0.981 0.994
0.98 0.992 0.975 0.975 0.968 0.975 0.983 0.977 0.992
0.978 0.988 0.988 0.98 0.972 0.972 0.985 0.983 0.99
0.982 0.98 0.99 0.981 0.97 0.965 0.995 0.97 0.98
0.981 0.986 0.991 0.978 0.967 0.993 0.987 0.976 0.993;

qs=0.008 0.0026 0.0065 0.0034 0.0034 0.0033 0.0018 0.0033 0.0033
0.0024 0.0024 0.0003 0.0048 0.0018 0.0026 0.0038 0.0030 0.001
0.0053 0.0022 0.008 0.0026 0.0024 0.0032 0.0053 0.0021 0.0026
0.0032 0.003 0.002 0.005 0.0021 0.0026 0.0033 0.0024 0.0033
0.003 0.0066 0.0023 0.0021 0.003 0.0083 0.0015 0.0052 0.0066
0.003 0.0016 0.0013 0.0036 0.0041 0.0022 0.0024 0.0028 0.0022;

c=9.48 11.83 10.04 10.35 10.65 13.48 10.38 10.10 11.73
```

```
9.41 8.40 11.81 10.98 11.98 12.69 10.41 13.41 13.95
10.66 14.93 13.12 11.08 11.83 14.98 13.67 11.22 10.71
9.37 13.61 14.15 9.05 14.22 12.17 11.63 12.81 11.80
12.43 11.43 14.87 11.88 10.71 8.44 12.50 13.93 8.73
9.57 9.26 10.60 13.12 8.47 10.38 12.66 10.91 13.36;
enddata
@for(LT(i,j):@bin(X(i,j)));
@for(TS(j,k):@bin(Y(j,k)));
@for(LS(i,k):@bin(V(i,k)));
@for(LTS:@bin(XY));

min = (@sum(station(k):RP(k)*DP(k) + RQ(k)*QP(k) + L(k)) +
RBP*((@ABS(RP(1)-RP(2)))+(@ABS(RP(1)-RP(3)))+(@ABS(RP(1)-RP(4)))+(@ABS(RP(2)-
RP(3)))+(@ABS(RP(2)-RP(4)))+(@ABS(RP(3)-RP(4)))
+(@ABS(RQ(1)-RQ(2)))+(@ABS(RQ(1)-RQ(3)))+(@ABS(RQ(1)-RQ(4)))+(@ABS(RQ(2)-
RQ(3)))+(@ABS(RQ(2)-RQ(4)))+(@ABS(RQ(3)-RQ(4)))
+ (@ABS(ST(1)-ST(2)))+(@ABS(ST(1)-ST(3)))+(@ABS(ST(1)-ST(4)))+ (@ABS(ST(2)-
ST(3)))+(@ABS(ST(2)-ST(4)))+(@ABS(ST(3)-ST(4)))));

@for(station(k):
ST(k) = @sum(LT(i,j): pm(i,j)*XY(i,j,k)) ;
STS(k) = (@sum(LT(i,j): ps(i,j)^2*XY(i,j,k)))^0.5 ;
SQ(k) = @sum(LT(i,j): qm(i,j)*XY(i,j,k)) ;
SQS(k) = (@sum(LT(i,j): qs(i,j)^2*XY(i,j,k)))^0.5 ;
S(k) = @sum(LT(i,j):sp(j)*XY(i,j,k)) ;
Q(k) = @sum(LT(i,j):ql(j)*XY(i,j,k)) ;
L(k) = @sum(LT(i,j):c(i,j)*(T/60)*XY(i,j,k)) ;
RP(k) = (1 - @PSN((S(k)-ST(k))/STS(k))) ;
RQ(k) = @PSN((Q(k)-SQ(k))/SQS(k)) ;
) ;

@for(station(k):
     @for(labor(i):
            @sum(task(j):XY(i,j,k))= (@sum(LT(i,j):XY(i,j,k)))*V(i,k));
     @sum(labor(i): V(i,k)) = 1 ;
);

@for(labor(i): @sum(station(k): V(i,k)) <= 1) ;
@for(task(j): @sum(LS(i,k): XY(i,j,k))=1);

   @sum(labor(i): XY(i,3,1))<= @sum(labor(i):XY(i,7,1));
   @sum(labor(i): XY(i,3,2))<= @sum(labor(i):XY(i,7,1)+XY(i,7,2));
   @sum(labor(i): XY(i,3,3))<= @sum(labor(i):XY(i,7,1)+XY(i,7,2)+XY(i,7,3));
   @sum(labor(i): XY(i,3,4))<=
@sum(labor(i):XY(i,7,1)+XY(i,7,2)+XY(i,7,3)+XY(i,7,4));

   @sum(labor(i):XY(i,1,1))<= @sum(labor(i):XY(i,9,1));
   @sum(labor(i):XY(i,1,2))<= @sum(labor(i):XY(i,9,1)+XY(i,9,2));
   @sum(labor(i):XY(i,1,3))<= @sum(labor(i):XY(i,9,1)+XY(i,9,2)+XY(i,9,3));
   @sum(labor(i):XY(i,1,4))<=
@sum(labor(i):XY(i,9,1)+XY(i,9,2)+XY(i,9,3)+XY(i,9,4));


END
```

# Appendix K

## LINGO model for SRLW-9-task-6-worker-5-station

```
SETS:
labor/1 2 3 4 5 6/:lab;
task/1 2 3 4 5 6 7 8 9/:SP,ql;
station/1 2 3 4 5/:stn,DP,QP,RP,RQ,ST,STS,SQ,SQS,S,Q,L;
LT(labor,task):pm,ps,qm,qs,c,X;
TS(task,station):Y;
LS(labor, station):V;
LTS(labor, task, station): XY ;
endsets
CALC:
  ! Set the linearization on ;
  @APISET( 214, 'int', 3);
ENDCALC
Data:
SP= 15 8 26 12 23 10 14 28 16;
ql=0.98 0.985 0.99 0.98 0.97 0.975 0.99 0.98 0.995;
M=5;
T=2880;
D=140;
QP=375 350 400 425 450;
DP=450 550 500 525 575;
RBP=1000;

pm= 14.5 7.9 25.2 11.8 22.5 9.9 14.0 27.0 15.8
15.2 7.6 26.2 12.1 23.1 9.6 13.8 27.5 15.5
13.0 8.0 25.8 11.5 22.0 10.1 14.1 28.0 16.0
14.8 8.1 24.0 11.0 22.8 10.0 13.5 28.1 16.2
15.0 7.5 24.5 12.2 23.0 9.8 13.4 26.5 15.2
13.5 7.8 26.0 11.0 23.2 10.2 13.7 27.8 15.9;

ps=0.87 0.36 1.0 0.62 0.72 0.66 0.41 1.0 0.76
0.32 1.05 0.4 0.41 0.41 1.11 0.73 0.75 0.92
1.5 0.83 0.62 0.82 1.0 0.35 0.3 0.42 0.41
0.78 0.52 1.5 1.0 0.8 0.5 0.93 0.3 0.3
0.52 1.21 1.2 0.30 0.52 0.73 1.11 1.55 1.32
1.0 0.38 0.33 0.92 0.3 0.3 0.84 0.63 0.5;

qm=0.975 0.98 0.98 0.977 0.965 0.98 0.99 0.975 0.99
0.978 0.983 0.992 0.982 0.971 0.981 0.988 0.981 0.994
0.98 0.992 0.975 0.975 0.968 0.975 0.983 0.977 0.992
0.978 0.988 0.988 0.98 0.972 0.972 0.985 0.983 0.99
0.982 0.98 0.99 0.981 0.97 0.965 0.995 0.97 0.98
0.981 0.986 0.991 0.978 0.967 0.993 0.987 0.976 0.993;

qs=0.008 0.0026 0.0065 0.0034 0.0034 0.0033 0.0018 0.0033 0.0033
0.0024 0.0024 0.0003 0.0048 0.0018 0.0026 0.0038 0.0030 0.001
0.0053 0.0022 0.008 0.0026 0.0024 0.0032 0.0053 0.0021 0.0026
0.0032 0.003 0.002 0.005 0.0021 0.0026 0.0033 0.0024 0.0033
0.003 0.0066 0.0023 0.0021 0.003 0.0083 0.0015 0.0052 0.0066
0.003 0.0016 0.0013 0.0036 0.0041 0.0022 0.0024 0.0028 0.0022;

c=9.48 11.83 10.04 10.35 10.65 13.48 10.38 10.10 11.73
```

```
9.41 8.40 11.81 10.98 11.98 12.69 10.41 13.41 13.95
10.66 14.93 13.12 11.08 11.83 14.98 13.67 11.22 10.71
9.37 13.61 14.15 9.05 14.22 12.17 11.63 12.81 11.80
12.43 11.43 14.87 11.88 10.71 8.44 12.50 13.93 8.73
9.57 9.26 10.60 13.12 8.47 10.38 12.66 10.91 13.36;
enddata


@for(LT(i,j):@bin(X(i,j)));
@for(TS(j,k):@bin(Y(j,k)));
@for(LS(i,k):@bin(V(i,k)));
@for(LTS:@bin(XY));



min = (@sum(station(k):RP(k)*DP(k) + RQ(k)*QP(k) + L(k)) +
RBP*((@ABS(RP(1)-RP(2)))+(@ABS(RP(1)-RP(3)))+(@ABS(RP(1)-RP(4)))+(@ABS(RP(1)-
RP(5)))+(@ABS(RP(2)-RP(3)))+(@ABS(RP(2)-RP(4)))+(@ABS(RP(2)-RP(5)))
+(@ABS(RP(3)-RP(4)))+(@ABS(RP(3)-RP(5)))+(@ABS(RP(4)-RP(5)))+(@ABS(RQ(1)-
RQ(2)))+(@ABS(RQ(1)-RQ(3)))+(@ABS(RQ(1)-RQ(4)))+(@ABS(RQ(1)-RQ(5)))
+(@ABS(RQ(2)-RQ(3)))+(@ABS(RQ(2)-RQ(4)))+(@ABS(RQ(2)-RQ(5)))+(@ABS(RQ(3)-
RQ(4)))+(@ABS(RQ(3)-RQ(5)))+(@ABS(RQ(4)-RQ(5)))
+ (@ABS(ST(1)-ST(2)))+(@ABS(ST(1)-ST(3)))+(@ABS(ST(1)-ST(4)))+(@ABS(ST(1)-
ST(5)))+ (@ABS(ST(2)-ST(3)))+(@ABS(ST(2)-ST(4)))+(@ABS(ST(2)-
ST(5)))+(@ABS(ST(3)-ST(4)))
+(@ABS(ST(3)-ST(5)))+(@ABS(ST(4)-ST(5)))));


@for(station(k):
ST(k) = @sum(LT(i,j): pm(i,j)*XY(i,j,k)) ;
STS(k) = (@sum(LT(i,j): ps(i,j)^2*XY(i,j,k)))^0.5 ;
SQ(k) = @sum(LT(i,j): qm(i,j)*XY(i,j,k)) ;
SQS(k) = (@sum(LT(i,j): qs(i,j)^2*XY(i,j,k)))^0.5 ;
S(k) = @sum(LT(i,j):sp(j)*XY(i,j,k)) ;
Q(k) = @sum(LT(i,j):ql(j)*XY(i,j,k)) ;
L(k) = @sum(LT(i,j):c(i,j)*(T/60)*XY(i,j,k)) ;
RP(k) = (1 - @PSN((S(k)-ST(k))/STS(k))) ;
RQ(k) = @PSN((Q(k)-SQ(k))/SQS(k)) ;
) ;


@for(station(k):
     @for(labor(i):
            @sum(task(j):XY(i,j,k))= (@sum(LT(i,j):XY(i,j,k)))*V(i,k));
     @sum(labor(i): V(i,k)) = 1 ;
);

@for(labor(i): @sum(station(k): V(i,k)) <= 1) ;
@for(task(j): @sum(LS(i,k): XY(i,j,k))=1);
!new constraint;
@for(station(k): @sum (LT(i,j): XY(i,j,k))>=1);


   @sum(labor(i): XY(i,3,1))<= @sum(labor(i):XY(i,7,1));
   @sum(labor(i): XY(i,3,2))<= @sum(labor(i):XY(i,7,1)+XY(i,7,2));
   @sum(labor(i): XY(i,3,3))<= @sum(labor(i):XY(i,7,1)+XY(i,7,2)+XY(i,7,3));
```

```
   @sum(labor(i): XY(i,3,4))<=
@sum(labor(i):XY(i,7,1)+XY(i,7,2)+XY(i,7,3)+XY(i,7,4));
   @sum(labor(i): XY(i,3,5))<=
@sum(labor(i):XY(i,7,1)+XY(i,7,2)+XY(i,7,3)+XY(i,7,4)+XY(i,7,5));

   @sum(labor(i):XY(i,1,1))<= @sum(labor(i):XY(i,9,1));
   @sum(labor(i):XY(i,1,2))<= @sum(labor(i):XY(i,9,1)+XY(i,9,2));
   @sum(labor(i):XY(i,1,3))<= @sum(labor(i):XY(i,9,1)+XY(i,9,2)+XY(i,9,3));
   @sum(labor(i):XY(i,1,4))<=
@sum(labor(i):XY(i,9,1)+XY(i,9,2)+XY(i,9,3)+XY(i,9,4));
   @sum(labor(i):XY(i,1,5))<=
@sum(labor(i):XY(i,9,1)+XY(i,9,2)+XY(i,9,3)+XY(i,9,4)+XY(i,9,5));

END
```