MULTI-OBJECTIVE OPTIMISATION OF ASSEMBLY LINE BALANCING TYPE-E PROBLEM WITH RESOURCE CONSTRAINTS

MASITAH BINTI JUSOP

MASTER OF ENGINEERING (MECHANICAL)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : MASITAH BINTI JUSOP

Date of Birth : 21 APRIL 1991

Title : MULTI-OBJECTIVE OPTIMISATION OF ASSEMBLY LINE BALANCING TYPE-E PROBLEM WITH RESOURCE CONSTRAINTS

Academic Session : 2016/2017

I declare that this thesis is classified as:

- ☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- ☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- ☑ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserve the right as follows:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

____
(Student's Signature)

910421-11-5176
New IC/Passport Number
Date:

____
(Supervisor's Signature)

____
Name of Supervisor
Date:

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Engineering in Mechanical.

____

(Supervisor's Signature)

Full Name : DR. MOHD FADZIL FAISAE BIN AB. RASHID

Position : SENIOR LECTURER

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Author's Signature)

Full Name : MASITAH BINTI JUSOP

ID Number : MMM14044

Date :

MULTI-OBJECTIVE OPTIMISATION OF ASSEMBLY LINE BALANCING TYPE-E PROBLEM WITH RESOURCE CONSTRAINTS

MASITAH BINTI JUSOP

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Master of Engineering

Faculty of Mechanical Engineering

UNIVERSITI MALAYSIA PAHANG

NOVEMBER 2016

ACKNOWLEDGEMENT

I am grateful and would like to express my sincere gratitude to my supervisor Dr. Mohd Fadzil Faisae bin Ab Rashid for his germinal ideas, invaluable guidance, continuous encouragement and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, strong conviction for science, and his belief that a Master program is only the start of a life-long learning experience. I appreciate his consistent support from the 1st day I applied to graduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my naive mistakes, and his commitment on my future career. I also sincerely thank him for the time spent in proofreading and correcting many of my mistakes.

My sincere thanks goes to all my colleagues at Faculty of Mechanical Engineering, UMP who had helped me in many ways and made my stay at UMP a pleasant and unforgettable one. Thank you for their excellent cooperation, unwavering inspirations and support during this study.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dreams and sacrifice throughout my life. I acknowledge the sincerity of my beloved parents who had consistently encouraged me to pursue my higher studies in UMP. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals. Special thanks goes to all members from other universities. I would like to acknowledge their comments and suggestions, which was crucial for the successful completion of this study

ABSTRACT

Assembly Line Balancing (ALB) is an attempt to assign tasks to various workstations along a line so that the precedence relations are satisfied and some performance measures are optimised. In this research, a few tasks that use similar resources will be assigned in the same workstation by ensuring that it does not violate the precedence constraint and that the total processing time in each workstation is approximately the same and does not exceed the cycle time. Assumption by previous researches that any assembly task can be performed in any workstation encourages the author to focus on the resource usage in ALB. Limited number of resources in the industry also becomes a vital influencer to consider this constraint in ALB. Apart from that, Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) has not yet been implemented by previous researcher in the optimisation of Assembly Line Balancing Type-E (ALB-E) itself with resource constraints. The aim of this research is to establish a mathematical model for ALB-E with resource constraints (ALBE-RC). This research is proposed to be conducted in three main phases. After conducting literature review, the modelling phase will be performed. In the second phase of this research, an algorithm will be developed to optimise the problem. Later, the optimisation algorithm will be tested and verified using test problems from literature. The third phase of this research is, an industrial case study will be conducted for the purpose to validate the mathematical model and the optimisation algorithm. This research gap was identified when none of the previous research considered machine, tool, and worker constraint in ALB-E. In this research, a Genetic-based Algorithm was used as an optimisation approach. The Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) has been proposed to optimise ALBE-RC. The optimisation result indicated that the NSGA-II algorithm has better performance in finding non-dominated solution due to small error ratio and small generational distance as compared to other algorithms like Multi-Objective Genetic Algorithm (MOGA) and Hybrid Genetic Algorithm (HGA). The results indicate that NSGA-II has the ability to explore the search space and has better accuracy of solution towards Pareto-optimal front. The validation phase from the industrial case study concluded that the proposed methodology and algorithm can be implemented in industries. The cycle time of existing layout had been extensively decreased from 16.1 seconds to 13.1 seconds after the optimisation. The number of workstations was decreased after the optimisation from 17 workstations to nine (9) workstations. Meanwhile, the number of resources used were reduced from 43 resources to 40 resources. Apart from that, the percentage of line efficiency improved from 33.8% to 78.4%. These results indicated that the developed methodology and the proposed algorithm can reduce the utilisation of resources, workstations and cycle time. In fact, the aforementioned approach also can increase the efficiency of assembly process as well as enhance the industrial productivity.

ABSTRAK

Pengimbangan Rangkaian Pemasangan merupakan usaha untuk mengagihkan tugasan kepada pelbagai stesen kerja dalam rangkaian supaya tertib hubungan dipatuhi dan ukuran prestasi dapat dicapai pada tahap optimum. Dalam kajian ini, beberapa tugasan yang menggunakan sumber yang sama akan diagihkan dalam stesen kerja yang sama dengan mematuhi kekangan tertib hubungan serta memastikan jumlah masa pemprosesan di setiap stesen kerja adalah lebih kurang sama dan tidak melebihi masa kitaran. Penyelidik terdahulu yang mengandaikan bahawa mana-mana tugasan pemasangan boleh dilakukan di mana-mana stesen kerja telah mendorong penyelidik untuk mempertimbangkan penggunaan sumber dalam ALB. Jumlah sumber yang terhad dalam industri juga menjadi satu pengaruh yang penting untuk mengambil kira kekangan ini dalam ALB. Selain itu, *Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)* masih belum digunapakai oleh penyelidik terdahulu dalam pengoptimuman Pengimbangan Rangkaian Pemasangan Jenis-E (ALB-E) dengan kekangan sumber. Kajian ini bertujuan untuk menghasilkan satu model matematik untuk ALB-E dengan kekangan sumber (ALBE-RC). Kajian ini dicadangkan untuk dijalankan dalam tiga fasa. Selepas menjalankan kajian literasi, fasa memodelkan masalah akan dilakukan. Dalam fasa kedua kajian ini, penghasilan algoritma dilakukan untuk mengoptimumkan masalah. Kemudian, ujian dan pengesahan algoritma akan dilakukan menggunakan permasalahan daripada sumber literasi. Fasa ketiga kajian ini adalah menjalankan kajian kes industri bagi tujuan pengesahan model matematik serta algoritma. Masalah ini timbul apabila penyelidik terdahulu tidak mempertimbangkan kekangan mesin, sumber dan pekerja dalam ALB-E. Dalam kajian ini, pendekatan pengoptimuman berasaskan Algoritma Genetik telah digunakan. *Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)* telah dicadangkan untuk mengoptimumkan ALBE-RC. Hasil pengoptimuman menunjukkan bahawa algoritma NSGA-II mempunyai prestasi yang lebih baik dalam mencari penyelesaian tidak dominan dan mempunyai ralat kecil serta jarak generasi yang kecil berbanding dengan algoritma lain iaitu *Multi-Objective Genetic Algorithm (MOGA)* dan *Hybrid Genetic Algorithm (HGA)*. Ini menunjukkan bahawa NSGA-II mempunyai keupayaan untuk meneroka pencarian ruang serta mempunyai ketepatan penyelesaian yang lebih baik ke arah penyelesaian *Pareto* yang optimum. Fasa pengesahan daripada kajian kes industri menyimpulkan bahawa metodologi dan algoritma yang dicadangkan dapat diguna pakai dalam industri. Masa kitaran bagi susun atur yang sedia ada telah menunjukkan penurunan mendadak daripada 16.1 saat kepada 13.1 saat selepas pengoptimuman. Bilangan stesen kerja juga telah berkurang selepas pengoptimuman daripada 17 stesen kerja kepada sembilan stesen kerja. Di samping itu, bilangan sumberjuga menunjukkan penurunan daripada 43 sumberkepada 40 sumber. Selain itu, peratus kecekapan rangkaian juga meningkat daripada 33.8% kepada 78.4%. Keputusan ini menunjukkan bahawa metodologi dan algoritma yang dicadangkan dapat meningkatkan kecekapan proses pemasangan serta mempertingkatkan produktiviti industri.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

ALB	Assembly Line Balancing
ALB-E	Assembly Line Balancing Type-E
ALBE-RC	Assembly Line Balancing Type-E with Resource Constraint
GA	Genetic Algorithm
NSGA-II	Elitist Non-Dominated Sorting Genetic Algorithm
NSGA	Non-Dominated Sorting Genetic Algorithm
MOGA	Multi-Objective Genetic Algorithm
HGA	Hybrid Genetic Algorithm
E	Line efficiency
m	Number of workstations
c	Cycle time
n	Number of tasks
pt	Processing time
ct	Cycle time
R	Resource
NDS	Number of non-dominated solution
GD	Generational distance
ER	Error ratio
ST	Workstation
W	Worker
p_c	Crossover probability
p_m	Mutation probability
H_0	Null hypothesis
H_1	Alternative hypothesis

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter gives a short description of the research background including the problem statement, followed by research objectives, research scope and significant of research. In the next section, the structure of this thesis is briefly explained and in the last part, the chapter summary is presented.

1.2 Research Background

Assembly line is one of the industrial production systems used to produce finished goods in an industry. It has been widely employed in many production industries such as automotive, electronics and other consumer durable production to enhance the efficiency of production system (Mozdgir et al., 2013). The rapid development of manufacturing industry is caused by increasing customer demands. This scenario forced manufacturers to maximise the production output in order to meet customers' demands. This can be achieved by eliminating process inefficiencies (i.e. minimise the number of workstations and cycle time) as well as by utilising resources at an optimum level.

Assembly Line Balancing (ALB) problem is defined as assigning tasks to workstations to optimise some performance measures by reducing the percentage of idle time or balance delay of assembly line (Chen et al., 2006; Ranjan and Pawar, 2014). This research aims to maximise the production rate and achieve the number of workstations, while satisfying some particular constraints, such as (i) precedence constraints and (ii) the total processing time assigned to each workstation must not exceed the cycle time (Suwannarongsri and Puangdownreong, 2008).

Due to limited number of resources in industry, it is necessary to consider the problem in assembly optimisation. This research intends to focus on multi-objective optimisation of Assembly Line Balancing Type-E (ALB-E) problem of a simple model.

1.3 Problem Statement

The main problem with the existing research in ALB-E is the assumption that any assembly task can be performed at any workstation (Scholl & Becker, 2006; Zhang et al., 2007; Zhang et al., 2008; Hamta et al., 2013). However, each workstation has its own capabilities and specialisation. This situation has been highlighted as one of the serious problems in the industry (Ağpak and Gökçen, 2005; Bautista and Pereira, 2007). This finding is consistent with the study by Sungur and Yavuz, (2015) that emphasizes workers' assignment to be mandatorily based on their qualification.

Rapid growth in manufacturing becomes a vital influencer to consider the usage of resources due to limited number of machines and tools. Although a small number of researches considered resource constraint in their works, none of them focused on resource constraint in ALB-E especially in terms of machine, tool and worker constraints (Ağpak and Gökçen, 2005; Browning and Yassine, 2010; Corominas et al., 2011 Battaïa and Dolgui, 2013). It is important to consider these constraints due to the limited number of resources where the utilisation of these resources can be minimised.

In the past years, the Genetic Algorithm (GA) approach has attracted the attention of researchers to solve issues related to ALB (Gurevsky at al., 2013; Zacharia and Nearchou, 2013; Al-Hawari et al., 2014). This finding is consistent with the finding of past studies that used similar approach (Scholl & Becker, 2006; Suwannarongsri & Puangdownreong, 2008; Wei & Chao, 2011). Till date, to the best knowledge of the researcher, none of the published work employed Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) in the optimisation of ALB-E in terms of resource constraint.

For that reason, it is crucial to propose NSGA-II to address the research gap as it is capable to solve real-world optimisation problem for multi-objective functions (Chica et al., 2012; Guo et al., 2014; Zhao et al., 2015). In comparison with the old version of Non-Dominated Sorting Genetic Algorithm (NSGA), the NSGA-II implements elitism-preserving technique (Deb, 2001). The findings from past studies by Deb et al. (2002) and Zhao et al. (2015) concluded that NSGA-II has better convergence towards Pareto-optimal front. The solutions generated by NSGA-II are geared towards Pareto-optimal front. However, the existing algorithms such as Multi-Objective Genetic Algorithm (MOGA) usually have slow convergence (Fonseca and Fleming, 1993). Based on literature review, there are no studies available on the implementation of NSGA-II to optimise the ALB-E with resource constraint and this has motivated the researchers to conduct the present study.

1.4 Research Objective

The objectives of this research are:

i. To study the ALB-E problem and establish a mathematical model for ALB-E problem with resource constraints (ALBE-RC).
ii. To optimise the ALBE-RC using Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II).
iii. To validate the mathematical model and optimisation of algorithm through an industrial case study.

1.5 Research Objective

The scope of this research are stated as follow:

i. This research studies on the optimisation of Assembly Line Balancing Type-E problem with resource constraint (ALBE-RC). This problem is limited to a simple model of assembly problem.

ii. In this research, only GA-based algorithm is considered. For the optimisation of problem, the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) is proposed. The algorithm will be compared within GA-based algorithm.

iii. The validation method of the optimised algorithm is conducted using test problems from literature. In addition to that, the algorithm and mathematical model are validated through an industrial case study to ensure the applicability of the optimisation results. The industrial case study is only focused on and conducted in an electronic company.

1.6 Significance of Research

By achieving the aforementioned objectives, the research will increase the efficiency of assembly process. This research targets to reduce the number of workstation in an assembly as well as to minimise the cycle time by managing the number of resource used. Lower cycle time and number of workstation used will enhance the line efficiency. Apart from that, by considering the resource constraints, the resource usage will be significantly reduced in an assembly process.

By proposing an efficient way using the proposed algorithm and mathematical model to assemble a product, the long term implication of this research will be reflected on the enhanced industrial productivity. The modelling phase involves the steps to transform a product into a precedence diagram and also the steps on how to transform the precedence diagram into a digital format language that can be understood by a computer. Then, the NSGA-II will find the optimal solutions according to the objective functions to assemble a product.

1.7 Structure of Thesis

This thesis consists of six chapters. The summary of each chapter's content is detailed out as follow:

Chapter 1 introduces the research and clarifies the problem statement, research objectives and research scope. Lastly, the significance of research is discussed in this chapter.

Chapter 2 reviews the previous work on Assembly Line Balancing (ALB), specifically on a single model for Type-E problem. In this chapter, the review emphasises problem modelling, objective function and also the optimisation algorithm used in ALB-E. Apart from that, literature review is performed to identify research gaps.

Chapter 3 details the research methodology. It explains the idea of how the research is conducted. The flow of the research methodology begins with problem modelling, followed by algorithm development, testing, industrial data collection, discrete event simulation and finally validation.

Chapter 4 presents the development of optimisation algorithm, NSGA-II and also the finding of the research. This chapter validates the performance of the proposed algorithm. A computational experiment is done to test the performance of NSGA-II algorithm using generic problem from literature. Later, the performance of the proposed algorithm is compared with other comparison algorithms based on Genetic Algorithm.

Chapter 5 validates the mathematical model and the optimisation through an industrial case study. The industrial case study is performed to verify the proposed model and NSGA-II algorithm from the industrial experts. The collected data is modelled based on the proposed model as outlined in Chapter 3, section 3.1.

Chapter 6 summarises and concludes the research work and outlines the contribution of study to the knowledge pool. Apart from that, this chapter also discusses the limitations of research and provides some recommendations for future research.

1.8 Chapter Summary

In summary, this chapter explains the research background on Assembly Line Balancing. The problem statement is also highlighted in this chapter which leads to the discussion on research objectives. In addition to that, the scope and the significance of this research are also explained in this research. Last but not least, this chapter attempts to simplify the structure of the thesis by presenting the summary of each chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter serves as a platform to review literature on Assembly Line Balancing. Detailed descriptions of previous work on Assembly Line Balancing of Type-E problem (ALB-E) are also discussed. Literature review on Genetic Algorithm for ALB-E is also covered in this chapter followed by chapter summary.

2.2 Assembly Line Balancing

Assembly Line Balancing (ALB) is the decision problem of distributing assembly task among workstations with respect to some objectives to manufacture a finished product (Rashid et al., 2012; Grzechca and Foulds, 2015). It has been widely used in production system as it can increase the efficiency of the system (Nourmohammadi & Zandieh, 2011). The assembly task must be assigned to workstations without violating the precedence constraints (Sungur & Yavuz, 2015). The total assembly time taken at each workstation must also be approximately the same and should not exceed the cycle time (Mozdgir et al., 2013).

On the other hand, the number of workstations should also be optimised. These conditions must be respected in order to achieve line balance. ALB is a type of NP-hard optimisation problem (Otto et al, 2011; Wei & Chao, 2011; Rashid et al., 2012). Generally, this type of problem has an extremely large number of feasible solutions (Roshani et al., 2012; Battaïa and Dolgui, 2013; Hamta et al., 2013; Morrison et al., 2014) . However, it will be time consuming to find optimal solutions within the extremely large search space (Ranjan & Pawar, 2014). Therefore, an advanced approach of algorithm is necessary to solve the large-scale of problems.

ALB can be classified into two categories: Simple Assembly Line Balancing Problems (SALBP) and General Assembly Line Balancing Problems (GALBP) (Becker & Scholl, 2006; Boysen et al., 2007). The most notable assembly line is SALBP is employed when the same product is being manufactured from the line. This type of problem is classified into four groups with respect to the objective functions (Becker and Scholl, 2006).

i. Simple Assembly Line Balancing Type-1 (SALB-1) aims to minimise the number of workstation on the line for a fixed cycle time.

ii. Simple Assembly Line Balancing Type-2 (SALB-2) aims to minimise the cycle time for a fixed number of workstation on the line.

iii. Simple Assembly Line Balancing Type-E (SALB-E) aims to maximise the efficiency of the line by simultaneously minimising the number of workstation and the cycle time.

iv. Simple Assembly Line Balancing Type-F (SALB-F) aims to determine the feasible line with a combination of cycle time and number of workstation.

Other problems which are not included in the simple assembly line category are considered as Generalised Assembly Line Balancing (GALB) problems. GALB is a very large and is an extended type of Assembly Line Balancing problem. Mixed-model Assembly Line Balancing (MALB) or Mixed-model Sequencing Problem (MSP) and also U-line Balancing Problem (UALBP) are categorised as GALB problem (Boysen et al., 2007). The classification of Assembly Line Balancing problems is illustrated as in Figure 2.1.

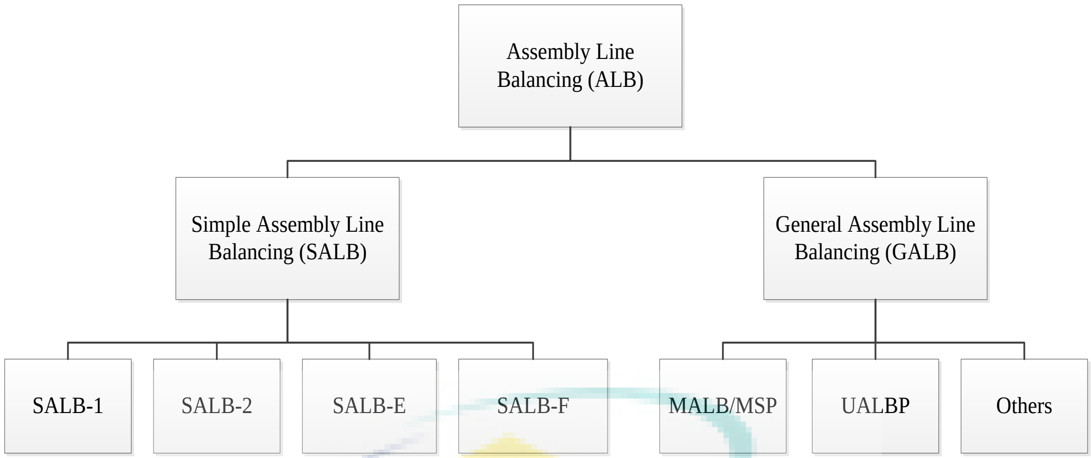


Figure 2.1 Classification of Assembly Line Balancing

Previous studies had primarily concentrated on SALB-1 (Emeke & Offiong, 2013; Armin Scholl & Becker, 2006). Research findings by Ponnambalam et al. (2000) and Chong et al. (2008) are also inclined towards the study on SALB-1. Meanwhile, the paper presented by Gu et al. (2007), Zhang et al. (2007) and Zhang et al. (2008) focused on SALB-2. Similarly, the work presented by Suwannarongsri et al. (2007) also discussed on SALB-2 problem.

However, little attention had been paid on SALB-E as it is more complicated as compared to SALB-1 and SALB-2 (Scholl & Becker, 2006; Suwannarongsri & Puangdownreong, 2008; Wei & Chao, 2011; Gurevsky et al, 2012). In fact, studying on SALB-E is more challenging (Su et al., 2014). Apart from that, the study on SALB-E also has been discovered in other research works (Zacharia & Nearchou, 2013; Al-Hawari et al., 2014; Esmaeilbeigi et al., 2015). Research on SALB-E need to consider multi-objective functions instead of single objective in both SALB-1 and SALB-2 (Esmaeilbeigi et al., 2015). Taking real manufacturing scenario into account, it is better to consider both parameters to minimise the number of workstation and the cycle time for the purpose of maximising assembly efficiency.

Simple Assembly Line Balancing problem is used when a single product is manufactured on the line (Battaïa & Dolgui, 2013). This research studies the Assembly Line Balancing Type-E problem (ALB-E) focusing on a single model. ALB-E aims to maximise the efficiency of line by simultaneously minimising the number of workstation and cycle time. In addition to that, ALB-E determines to identify the quality of line. The formulated solution for this problem can be expressed as described in Eq. (2.1).

$$E=\frac{\sum_{i=1}^{m} T_i}{mc} \times 100\% \tag{2.1}$$

where E : Line efficiency

m: Number of workstations

c : Cycle time

T_i: processing time of the i^{th} workstation

From Eq. (2.1), it is evident that the efficiency of the line, E increases independently with the number of workstations, m and cycle time, c. This relationship is proven true by Wang et al. (2012) who, in their research, pointed out that the productivity of the process is influenced by the effectiveness of the production line. The increasing market demand could only be achieved with a better line efficiency. In other words, the processing time in each workstation must be reduced so that the efficiency of the production line can be improved.

2.3 ALB-E Problem Modelling and Objective Function

A study on Assembly Line Balancing of Type-E problem (ALB-E) with different task processing times had been the focus of Gurevsky et al. (2012). The researchers attempted to study on the stability of feasible and optimal solutions for ALB-E. Two heuristic procedures were proposed and evaluated on certain targets in order to find a concession between the two goal functions. In order to compute the stability radius of feasible balances, the researchers had proposed polynomial time algorithm in the work.

A study by Scholl and Becker (2006) stated that there is no direct method to solve the ALB-E. Nevertheless, this type of model can be solved by a search method. The combination of the number of workstation, m and the cycle time, c feasible for the efficient line is chosen among the others or, the value of required line capacity as in Eq. (2.2) should be the least.

$$T = m.c \tag{2.2}$$

where T is line capacity.

An attempt to maximise the line efficiency and minimise the idle time had received great attention by Wei and Chao (2011) in ALB-E. These two objectives can be achieved by minimising the number of workstation and cycle time. Studies found that ALB-1 and ALB-2 models are combined by researchers in order to develop the ALB-E model. In ALB-1, the number of workstation is minimised with fixed cycle time. This model is re-defined as ALB-1-i with the intention of determining the minimum number of workstation. The goal of modified model ALB-2 is to achieve minimisation of cycle time, ct with a fixed number of workstations, m. The line efficiency, E is formulated as Eq. (2.3) which defines the efficiency of line and directly increases with total time of all tasks and inversely proportional to the number of workstation and cycle time.

$$E = \frac{t_{sum}}{m.ct} \tag{2.3}$$

where t_{sum} is the total time of all tasks

In order to maximise the line efficiency, the optimal number of workstation must be completed within a given maximum cycle time, ct_{max}. The value of ct_{max} must be less than or equal to the total task time, $\sum t_i$ and at the same time it also should be greater than or equivalent to the largest task time, max t_i in data. Only one workstation will be required whenever the value of ct_{max} exceeds or remains the same as total task time. No solution will be obtained as the value for ct_{max} is less than or equivalent to the largest task time in data. The following conditions are used for ct_{max}.

$\max t_i \leq ct_{max} \leq \sum t_i$

If $ct_{max} \geq \sum t_i$ then $m = 1$, $E = \frac{T_{total}}{1.T_{total}} = 1$ thus, Balance loss = 0

If $ct_{max} \geq \max t_i$, no solution

After the value of ct_{max} has been set, the optimal number of workstation, m can be attained by using the spreadsheet. The value of m lies between the minimum number of workstation, m_{min} and the maximum number of workstation, m_{max} which are reflected in Eq. (2.4) and Eq. (2.5) respectively.

$$m_{min} = \left\lceil \sum_{i=1}^{n} \frac{t_i}{ct_{max}} \right\rceil \tag{2.4}$$

$$m_{max} = \left\lceil \frac{\sum t_i}{\max t_i} \right\rceil \tag{2.5}$$

where $m_{min} \leq m \leq m_{max}$

The study by Zacharia and Nearchou (2013) examined the minimisation of number of workstation, m and cycle time, c using fuzzy task processing time which is also known as *f*-SALBP-E. The objective functions of the problem are to maximise the line efficiency by simultaneously minimising the number of workstation, m and cycle time, c. The fuzzy efficiency, $\check{e}$ of the line is linearly dependent on summation of fuzzy processing times of all the task, $\dot{t}_{sum}$. It can also be attained by minimising the number of workstation and fuzzy cycle time of the line. Eq. (2.6) represents the line efficiency function. According to the researchers, the maximisation of line efficiency is comparable to the maximisation of idle time, $\check{I}$ which is described in Eq. (2.7).

$$\check{e} = \frac{\dot{t}_{sum}}{m.\check{c}} \tag{2.6}$$

$$\check{I} = m \times \check{c} - \dot{t}_{sum} \tag{2.7}$$

where $\dot{t}_{sum}$: total sum of the fuzzy processing time of all tasks

$\check{c}$: fuzzy cycle time of the line

The uncertainty and variability of task processing time and cycle time are presented by triangular fuzzy numbers (TFNs). A heuristic method based on Genetic Algorithm (GA) had been developed to solve the *f*-SALBP-E as it is a type of NP-hard optimisation problem. A two-phase GA is used for the purpose of solving the problem. In this approach, the optimal solution found from the first run is used to generate the early population of the binary run. There is no resource constraint stated in the study. By considering the fuzzy processing time for single assembly line balancing problem, theformulated mathematical model is performed and thus minimised the number of workstation and the fuzzy cycle time on the line.

Another research presented by Al-Hawari et al. (2014) gave emphasize on minimisation of number of workstation, minimisation of workload variation, and maximisation of line efficiency, E as the objective functions in the study. Eq. (2.9) was used by researchers to define the line efficiency. As a matter of fact, line efficiency can be maximised by minimising both variables; the actual number of workstation, m and the actual cycle time of the assembly line, C_a is illustrated as in Eq. (2.8).

$$C_a = \max_{1 \le k \le m} \{t(S_k)\} \tag{2.8}$$

Meanwhile, the sum of handling time of task, i is fixed. The minimum number of actual workstation, m can be obtained using mathematical formulation as stated in Eq. (2.10).

$$\max E = \frac{\sum_{i=1}^{n} t_i}{m.C_a} \tag{2.9}$$

$$\min m = \sum_{k=1}^{M} \max_{1 \le i \le n} \{x_{ik}\} \tag{2.10}$$

where t_i : processing time of task i

n : number of tasks

k : workstation number; $k = 1,\ldots, m$

M : maximum number of workstation available ($m \le M \le n$)

$$x_{ik} = \begin{cases} 1 & \text{if task } i \text{ is assigned to station } k \\ 0 & \text{otherwise} \end{cases}$$

$t(S_k)$ = the total time assigned to workstation k

Gurevsky et al., (2012) studied Assembly Line Balancing of Type-E problem under different task processing time. The researchers had carried out a study on stability of feasible and optimal solutions for ALB-E by proposing a polynomial time algorithm as an approach. Two heuristic procedures were recommended and evaluated on certain targets in order to find a concession between the two goal functions.

Due to lack of studies on ALB-E, Esmaeilbeigi et al. (2015) took an initiative to focus on ALB-E problem in their research. The researchers aimed to minimise the cycle time, number of workstation as well as the smoothness index in their study. In order to achieve this target, the researcher proposed a mixed integer linear programming formulation. A computational test over the benchmark data set verified the effectiveness of the formulation.

2.4 ALB-E Optimisation Algorithm

Genetic Algorithm (GA) was introduced by John Holland in the year of 1975 and it is mainly used by researcher to optimise a large and complex problem specifically in ALB (Tasan & Tunali, 2008; Mohd Razali & Geraghty, 2011; Ranjan & Pawar, 2014). As mentioned previously, ALB falls in the classification of NP-hard optimisation problem. Therefore, the implementation of GA technique to solve problem in this classification is well-matched (Cheshmehgaz et al., 2012; Wei et al., 2015). This view is supported by Ranjan & Pawar (2014) who stated that GA uses a direct random search as the optimisation method for complex problems with the aim to find optimum solutions. GA execution usually needs longer time as it will globally search for optimal solutions (Ponnambalam et al., 2000).

GA also has the ability to find a set of optimum solutions in a single run (Triki et al., 2014). In addition to that, Gurevsky et al. (2012) proposed polynomial time algorithm in the study of ALB-E to compute the stability radius of feasible balances. Two heuristic procedures were proposed and evaluated on certain targets in order to find a concession between the two goal functions.

A new Genetic Algorithm was presented by Al-Hawari et al. (2014) to solve multi-objective optimisation problem. Multi-Assignment Genetic Algorithm (MA-GA) had been proposed by the researchers with the combination of forward, backward, and bidirectional methods. However, this approach can only provide many feasible solutions of task assignments by combining the three methods simultaneously instead of combining the forward method only.

Suwannarongsri et al. (2007) had proposed a combination of Tabu Search (TS) and Genetic Algorithm (GA) to identify the solution for simple assembly line balancing problem. The researchers used TSGA-based method which is the combination of TS and GA method to solve the problem. A test of all type of SALBP problems from literature against the proposed method was performed. The results showed that the proposed TSGA-based method is capable in producing better solutions as compared to conventional method.

In another work, Zacharia and Nearchou (2013) developed a heuristic method based on GA to solve *f*-ALB-E. A two-phase approach was used starting off with generating initial population, followed by implementing best solutions until it reached termination conditions. The optimal solution achieved from the first attempt was used as the source for early population in the binary part for the purpose of finding a better performance. The algorithm reached a good feasible solution which is approximately close to the exact solution in an acceptable time period.

Although most previous researchers used GA as an optimisation technique especially in ALB problem, only a small number of research focused on ALB-E (Gu et al., 2007; Zhang et al., 2007; Zhang et al., 2008). The studies by Sabuncuoglu et al. (2000) and also Gonçalves and Almeida (2002) focused on normal GA as an optimisation method in ALB-E. In fact, to the best knowledge of the researcher, no prior published work proposed NSGA-II in ALB-E itself.

2.5 Multi-Objective Optimisation

Deb (2001) defines optimisation as the finding of one or more feasible solutions with respect to one or more objectives. Multi-objective optimisation involves the task of finding single or more optimum solutions for an optimisation problem with more than one objective function. In other words, multi-objective optimisation problem deals with two or more objective functions that needed to be minimised or maximised (Gen & Cheng, 2000).

Herein, there must be multiple conflicting objectives and trade-off between each objective function that resulted in a set of optimal solutions known as Pareto-optimal solution (Deb, Pratap, et al., 2002; Taylor, 2008; Triki, Mellouli, & Masmoudi, 2014). There will be no single solution satisfying all performance criteria that is better than the others (Deb et al., 2000; Nourmohammadi & Zandieh, 2011). In other words, there is no existence of a unique optimal solution in multi-objective optimisation problem. This problem aims to find the nearest set of solutions to the Pareto-optimal front and also to find the diversity in a set of solutions (Deb, 2001).

To evaluate the performance of the proposed algorithms, five performance indicators were measured as proposed by (Deb, 2001).

i. Number of Non-Dominated Solution, *NDS*
ii. Error Ratio, *ER*
iii. Generational Distance, *GD*
iv. *Spacing*
v. Maximum Spread, $Spread_{max}$

The number of non-dominated solution was measured to identify the ability of the algorithm to explore the search space. Meanwhile, Error Ratio (*ER*) and Generational Distance (*GD*) metrics measure the accuracy of solution. *ER* measures the ratio of non-member of Pareto-optimal set to the number of non-dominated solution whereas *GD* measures the average distance of solution to the nearest Pareto set. *Spacing* metric measures the uniformity of solution. It measures the relative distance between each solution. Last but not least, the maximum spread is evaluated in order to determine the spread of solution. The distance between two extreme solutions in the corresponding objective space is calculated by the $Spread_{max}$.

2.5.1 GA-based Algorithm for Multi-Objective Optimisation

Various algorithms have been developed to optimise multi-objective optimisation problem. GAs are mainly used by researchers for the optimisation of multi-objective problems (Tasan & Tunali, 2008; Razali & Geraghty, 2011; Ranjan & Pawar, 2014;). GA uses a direct random search as the optimisation method for complex problems with the aim of finding optimum solutions (Ranjan& Pawar, 2014). GA has the ability to find a set of optimum solutions in a single run (Triki et al., 2014b). Vector Evaluated Genetic Algorithm (VEGA), Multi-Objective Genetic Algorithm (MOGA), Hybrid Genetic Algorithm (HGA), Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II), and Non-Dominated Sorting Genetic Algorithm (NSGA) are examples of GA-based approaches for multi-objective optimisation. In this research, MOGA and HGA were selected as the comparison algorithms as these GA approaches were developed as an optimisation tool for NP-hard optimisation problem and usually used by researchers as multi-objective optimisation methods (Al-Hawari et al., 2014; Triki et al., 2014).

2.5.1.1 Vector Evaluated Genetic Algorithm (VEGA)

The first multi-objective evolutionary algorithm was proposed by Schaffer (1985) to find Pareto-optimal solutions for multi-objective optimisation problem. A modification on simple GA was done by Schaffer by performing proportional selection cycle according to each objective function. However, no significant studies had been reported for nearly ten years after the establishment of the aforementioned method. VEGA was not able to obtain a good spread of solutions for some cases (Deb, 2001).

2.5.1.2 Multi-Objective Genetic Algorithm (MOGA)

Multi-objective Genetic Algorithm (MOGA) was introduced by Fonseca and Fleming (1993a). Non-dominated classification based on GA population was applied in the algorithm. The application of GA has been widely used to solve various assembly line balancing problems due to its ability to find a set of non-dominated solutions in a single run (Deb et al., 2002; Yang et al., 2013). MOGA has the abilities to find a diverse set of non-dominated solutions and to explore close to optimal set of solutions (Deb, 2001; Zhang et al., 2008). It has also been widely used in the real-world optimisation problems to solve ALB problems. In the research conducted by Ponnambalam et al. (2000), the researchers concluded that GA takes more time in finding the global optimal solutions.

2.5.1.3 Hybrid Genetic Algorithm (HGA)

A Hybrid GA (HGA) was proposed by Chen et al. (2002) to solve the Assembly Line Planning problem. In this work, the GA is combined with heuristics solution. The optimum assembly sequences generated from heuristics approaches were included in the initial population of the GA. The proposed GA is able to search for many feasible solutions in a short time. GA is an approach used in finding an optimal solution for a complex optimisation problem (Razali and Geraghty, 2011). Valls et al. (2008) stated that HGA is high in quality and is a fast algorithm that is better than all other state-of-the-art algorithms.

2.5.1.4 Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)

Deb et al. (2002) first introduced an Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) as an improved version of NSGA. The proposed method was developed to accommodate a complex and real-world optimisation problem for multi-objective functions (Chica et al., 2012; Bandyopadhyay & Bhattacharya, 2014). The capabilities of NSGA-II to find much better solutions and convergence of nearly true optimal Pareto-optimal front had been proven by other researchers (Deb, Pratap, et al., 2002; Zhao et al., 2015).

Apart from incorporating elitism, NSGA-II also has a better sorting algorithm. NSGA-II used a crowding distance approach that requires the population to be sorted in decreasing rank of level according to each objective function. Barbosa et al. (2012) clarified that NSGA-II differs from other methods by which the individuals of a given population are sorted based on the level of non-domination.

The aforementioned algorithm also implements an elitism-preserving technique as it will ensure the best solution found in each generation will never be lost until a better solution is discovered (Saravanan, 2006; Barbosa et al., 2012; Baykasoğlu & Özbakır, 2014). In other words, elitism will make sure the non-dominated solutions having the best dominance found during the search space will be retained in the current population (Zinflou et al., 2008).

2.5.1.5 Non-Dominated Sorting Genetic Algorithm (NSGA)

Non-Dominated Sorting Genetic Algorithm (NSGA) is the older version of NSGA-II pioneered by Goldberg in the year of 1989 (Deb, 2001). In multi-objective optimisation, no single solution is better than another one. Similarly, to NSGA-II, this older version of NSGA also employed the domination rank to determine the fitness of a solution. Rather than using elitist strategy, NSGA implements a sharing-preserving technique to preserve the diversity of the solutions. The sharing function method requires the distance of every single solution to be compared to one another from all the solutions available.

2.6 Summary of Literature Review

Table 2.1 summarised the literature review of ALB from recent years. There were different objectives and techniques used to optimise the ALB problem. The most important objective is to minimise the number of workstation as it had been the major concern in 16 cited papers followed by minimisation of workload variation highlighted in 12 cited papers. Minimisation of cycle time and production cost were studied in 7 and 6 cited research papers respectively. The aim to minimise workload variation will maximise the workload smoothness (Zacharia & Nearchou, 2012) whereas a shorter cycle time will increase the production rate.

The rest of the data shows the objective to minimise the station area and idle time that had been considered in 4 and 2 cited paper respectively. Only a small number of cited published works (6) that considered the objective to maximise line efficiency. In order to maximise the assembly line efficiency, the researcher need to consider both objectives: (i) minimise the number of workstations and (ii) minimise the cycle time. Ten out of 34 cited research used GA-based approach instead of other techniques. This technique is mainly used among researchers to optimise a large and complex problem particularly in ALB problem (Yu & Yin, 2010). In fact, it has the capability to find a set of optimum solution in a single run. The findings also reveal that no previous research implemented NSGA-II to optimise ALB-E problem.

In a nut shell, it can be concluded that only a few researchers focused on ALB-E in their research due to the complexity of the problem. Only one cited paper highlighted the resource constraint in Assembly Line Balancing Type-1 (Kao et al., 2010). However, none of the previous studies considered resource constraint on ALB-E itself. It is important to consider these constraints because of the limited number of resources in the industry. Through this, the resource utilisation and the production cost can be minimised.

Table 2.1 Summary of literature review for ALB

Author and year	Optimisation function		Goals							Method	Constraints										
	1	**2**	**1**	**2**	**3**	**4**	**5**	**6**	**7**		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
(Zacharia and Nearchou, 2016)		X	X	X	X	X				MOEA	X	X		X							
(Esmaeilbeigi et al., 2015)		X	X	X		X				Model	X	X		X							
(Sungur and Yavuz, 2015)	X							X		Model	X	X		X				X			
(Saif et al., 2014)		X		X		X				PBABC and Taguchi	X	X		X							
(Triki et al., 2014)	X			X						HGA	X	X	X								
(Al-Hawari et al., 2014)		X	X		X	X				MA-GA	X	X									
(Triki et al., 2014b)		X		X				X		Hybrid MOGA	X								X		
(Ranjan and Pawar, 2014)		X			X		X			GA	X										

Table 2.1 continued.

(Gurevsky et al., 2013)		X	X							Other soft computing		X		X	X						
(Zacharia and Nearchou, 2013)		X	X		X					GA	X										
(Mozdgir et al., 2013)	X					X				Taguchi	X	X		X							
(Hamta et al., 2013)		X		X		X		X		PSO algorithm	X	X		X							
(Sivasankaran and Shahabudeen, 2013)	X		X							Model	X	X		X							
(Tuncel and Topaloglu, 2013)	X		X							Other soft computing			X							X	
(Hazır and Dolgui, 2013)	X			X						ACO	X			X							
(Cheshmehgaz et al., 2012)	X		X							CGA	X	X		X							
(Wang et al., 2012)		X	X	X				X		HGA	X										
(Zacharia and Nearchou, 2012)		X		X		X				*Fuzzy*- MOGA	X	X		X							
(Chica et al., 2012)		X	X						X	Memetic algorithm	X	X				X					
(Tapkan et al., 2012)	X		X							Bee algorithm	X		X	X							
(Roshani et al., 2012)								X		SA	X	X								X	

Table 2.1 Continued

(Mutlu and Özgörmüş, 2012)		X	X			X				Other soft computing	X	X					X				
(Wei & Chao, 2011)		X			X		X			Model		X		X							
(Cakir et al., 2011)		X				X		X		Hybrid SA	X	X		X							
(Fattahi et al., 2011)	X		X							ACO	X	X	X	X				X			
(Chica et al., 2011)		X	X						X	MOACO		X		X		X					
(Chica et al., 2011)		X	X						X	MOGA	X	X				X					
(Nearchou, 2011)		X		X	X	X				PSO	X	X									
(Corominas et al., 2011)		X		X		X				Other soft computing				X			X				
(Kao et al., 2010)	X		X							Shortest path algorithm	X	X		X							X
(Yu and Yin, 2010)	X		X							GA	X			X							
(Kilincci, 2010)	X					X				Petri net	X										
(Chica et al., 2010)		X	X						X	ACORGS		X		X		X					
(Uğur Özcan, 2010)	X		X							SA	X			X							

Indicators:

- Optimisation function: 1- Single objective, 2- Multiple objectives
- Goal: 1- Minimise number of workstation, 2- Minimise cycle time, 3- Maximise line efficiency, 4- Minimise workload variation, 5- Minimise idle time, 6- Minimise cost, 7- Minimise station area
- Constraints: 1- Precedence, 2- Cycle time, 3- Zoning (position), 4- Workstations, 5- Capacity, 6- Space (area), 7- Workload, 8- Worker, 9- Compatibility (task and resource), 10- Task direction, 11- Resource
- Method: MOEA- Multi-Objective Evolutionary Algorithm, HGA- Hybrid Genetic Algorithm, MA-GA- Multi-Assignment Genetic Algorithm, MOGA- Multi-Objective Genetic Algorithm, PSO- Particle Swarm Optimisation, SA- Simulated Annealing, ACO- Ant Colony Optimisation, CGA- Cellular Genetic Algorithm, PBABC- Pareto Based Artificial Bee Colony Algorithm, MOACO- Multi-Objective Ant Colony Optimisation, ACORGS- Combination of Ant Colony Optimisation (ACO) and Random Greedy Search (RGS)

2.7 Chapter Summary

This chapter has reviewed the previous researches on Assembly Line Balancing (ALB). An overview of research in ALB-E as well as the optimisation algorithm used had been presented in this chapter. From the review of literature, it appears that more attention had been paid to ALB. Most of the previous researchers tend to focus on the minimisation of number of workstation in their studies since it is the simplest problem to resolve in ALB.

Meanwhile, studies relating to ALB-E have been relatively scanty due to the complexity of the problem. Researches on ALB-E need to consider multi-objective functions instead of single objective as in ALB-1 and ALB-2. It is better to consider multi-objective in order to achieve line balance. This study aims to minimise the cycle time and the number of workstation, and simultaneously maximise the line efficiency.

A rigorous study on the usage of resources in ALB-E problem has not been given great attention by researchers in the past. Due to the limited number of resources in the industry, it is crucial to be concerned on the resource constraints in this study. Generally, most of the researchers used GA-based technique to optimise ALB problem. It may, however. be noted that none of the studies were aimed in implementing the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) specifically in ALB-E as an optimisation approach. This study implements NSGA-II algorithm as an optimisation method as it accommodates the real-world multi-objective optimisation problem (Bandyopadhyay & Bhattacharya, 2014). Apart from that, NSGA-II also has the capabilities to find much better solutions and convergence of near-true optimal Pareto-optimal front which had been proven by other researchers (Deb et al., 2002; Chica et al., 2011; Zhao et al., 2015).

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter details the research methodology which is illustrated in Figure 3.1. The research methodology is proposed to be conducted in three main phases. The first phase; modelling stage including the objective functions are presented in this chapter. In the modelling stage, ALB-E problem with resource constraint (ALBE-RC) will be presented using a particular approach. In this research, three objective functions were considered in order to achieve line balance and maximum efficiency.

The latter phase is the development of algorithm and testing. Last but not least, an industrial case study and validation phase are performed. Suitable company for case study will be identified, and followed by data collection where the method used to collect data is explained. Further discussion on case study will be explained thoroughly in Chapter 5.

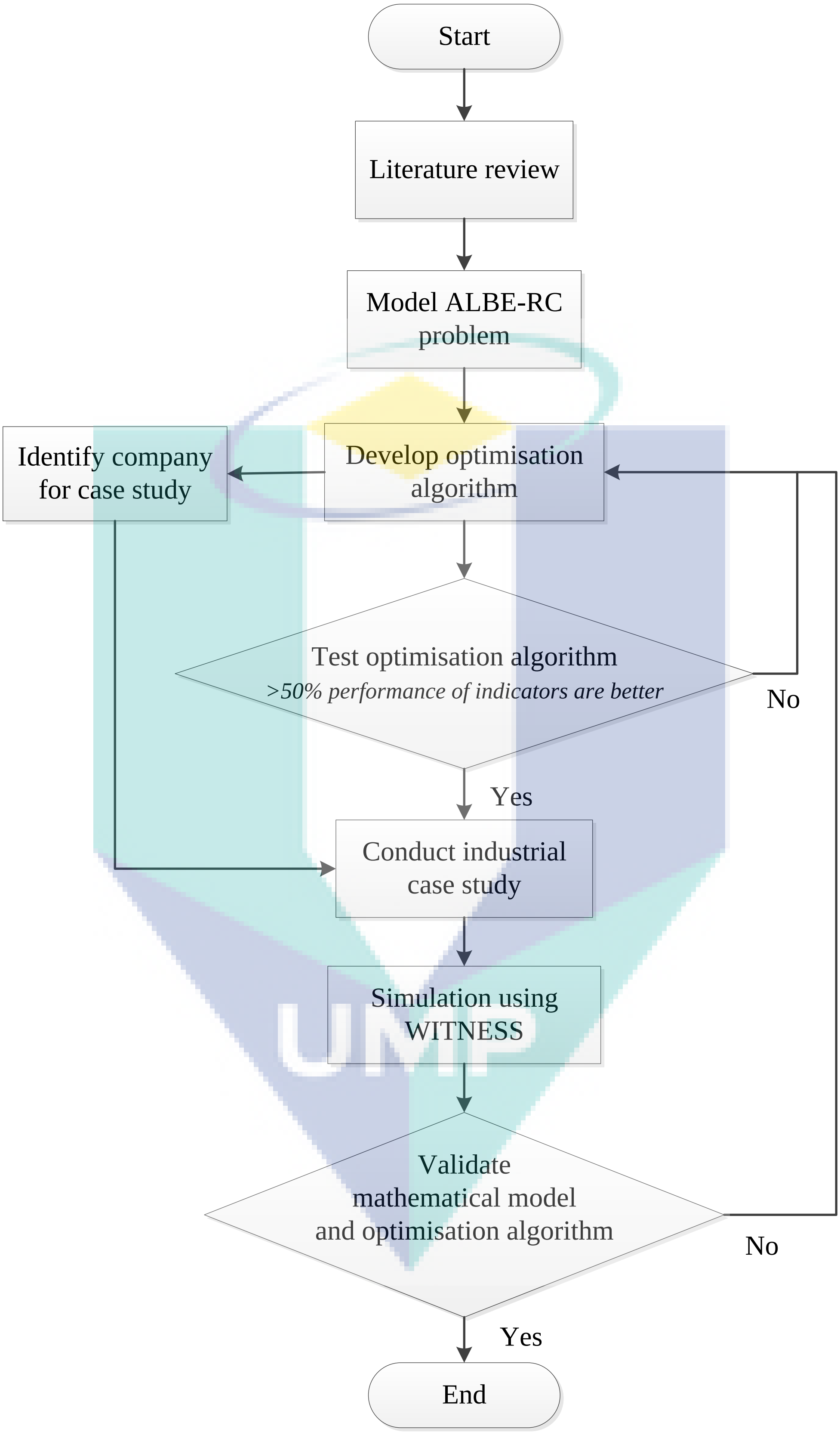


Figure 3.1 Flowchart of research methodology

3.1.1 First Phase of Research Methodology

The first phase of the research methodology started with conducting literature review. Later, the modelling phase of ALBE-RC problem was performed. The modelling of ALBE-RC problem was conducted using a particular approach. In this research, three objective functions were considered for the evaluation purpose.

i. Minimise the cycle time
ii. Minimise the number of workstation
iii. Minimise the number of resources

3.1.2 Second Phase of Research Methodology

The second phase in this research is the development of algorithm to optimise the ALBE-RC problem. The Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) was proposed as an optimisation approach to solve the problem. The activities in this phase started with establishing the solution procedure for ALBE-RC. Once the general solution procedure was established, the algorithm flow will be drafted by referring to the solution procedure. Next, the algorithm was coded into MATLAB software. The algorithm was modified to suit the problem studied. Then, the algorithm was tested and verified using generic problems from literature. In the meantime, suitable company to conduct the case study was identified.

Five performance indicators i.e. Number of Non-Dominated Solution (*NDS*), Error Ratio (*ER*), Generational Distance (*GD*), *Spacing*, and Maximum Spread ($Spread_{max}$) were used to evaluate the performance of the algorithm. According to Deb (2001), the algorithm can be classified as the best technique for multi-objective optimisation whenever it consistently produces better performance in at least two indicators.

3.1.3 Third Phase of Research Methodology

In the third phase of this research, an industrial case study was conducted for the purpose of validating the ALBE-RC model and the proposed NSGA-II algorithm. Related assembly data such as assembly task, cycle time, precedence constraint and resources were collected and manually recorded. The selected problem was modelled using the approach developed earlier. Then, the problem was optimised using the proposed algorithm. A simulation using WITNESS software was performed to simulate the assembly line. The optimisation results were validated and compared with the results of current layout. This phase was conducted in order to validate the mathematical model and the optimisation algorithm. The simulation results of the existing layout were compared with the simulation results after the optimisation to validate the significance of the optimisation parameters. A statistical test was performed to compare the different between the existing output and after the optimisation.

3.2 ALBE-RC Problem Modelling

After conducting literature review, the modelling phase was performed. The main activities in this phase were to establish problem representation, followed by modelling and evaluation procedures. The ALBE-RC was represented using a particular approach. A simple ALB problem was used at this stage. Besides that, an evaluation procedure to measure how good a generated solution is was also identified.

The modelling phase involved the steps to transform a product into a precedence diagram, steps on how to transform the precedence diagram into a digital format language that can be understood by a computer and also steps on how to evaluate the assembly sequence. This section demonstrates an approach for the optimisation of ALBE-RC problem through a simple diagram representation. Figure 3.2 shows the flowchart of the problem representation steps. Last but not least, an example of assembly problem was also presented in the last part of this section using the following approach.

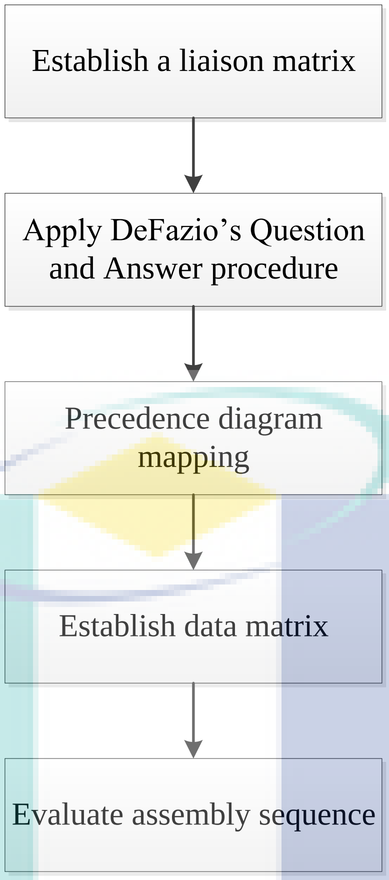


Figure 3.2 Flowchart of problem representation steps

3.2.1 Establishment of Liaison Matrix

In 1984, Bourjault had presented a liaison matrix approach to generate feasible assembly sequences (Fazio and Whitney, 1987). Let's say that the assembly product has r parts as a product of the relation between i^{th} and j^{th} parts presented in liaison matrix. If there is any relation between i and j, $L(i, j)=a_i$ where (i= 1, 2, 3,…, n), or else , $L(i, j)$ is left blank whereby n signifies the assembly task (number of liaison). In other words, liaisons represent the lines between nodes which correspond to the relations between parts wherein nodes are denoted as the parts. The establishment of liaison matrix determines the assembly relation.

3.2.2 Apply De Fazio's Question and Answer

Once the liaison matrix was established, DeFazio's question and answer procedure was applied for the purpose of identifying the existence of precedence relations in assembly tasks. There were two questions that must be considered and needed to be answered while evaluating each assembly task (Fazio and Whitney, 1987):

i. What tasks must be done prior to doing task *i*?
ii. What tasks must be left to be done after doing task *i*?

3.2.3 Precedence Diagram Mapping

The precedence diagram illustrates the relationship of predecessor-successor with assembly steps (Weigert et al., 2011). The aforementioned questions needed to be answered in order to determine the precedence constraint. The example of precedence diagram mapping is shown in Figure 3.3. The precedence constraint that can be identified from this figure is C[(1, 2), (1, 4), (2, 3), (4, 5)]. Referring to constraint (1, 2), task 1 needs to be done prior to doing task 2. Task 1 is known as predecessor while tasks 2 and 4 are the successors of task 1. In other words, an outgoing arc symbolises the predecessor task whereas an incoming arc represents the successor of the task. Each node with different numbers denotes the assembly tasks.

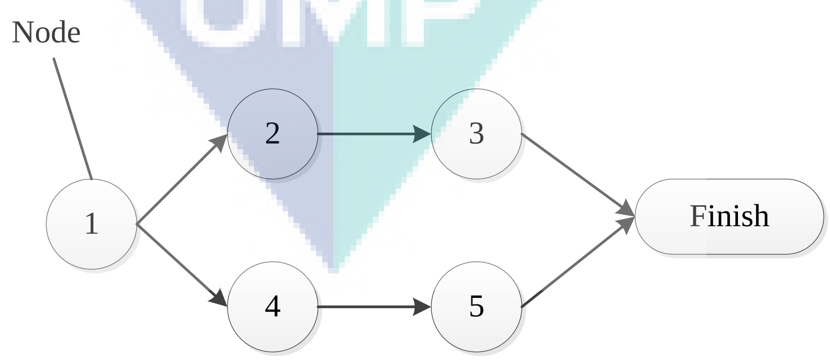


Figure 3.3 Precedence diagram mapping

3.2.4 Establishment of Data Matrix

The assembly data for proposed representation can be tabulated in $n \times 4$ table as illustrated in Figure 3.4 where n is the number of assembly task. The first column t denotes task time whereas the other three columns R_A, R_B and R_C represent resource A, resource B and resource C respectively. In this assembly process, numerical forms, e.g. 1, 2, and 3, denote the type of resource that was being used. For example; R_A=1, R_B=2 while R_C=3.

$$
\begin{array}{c|cccc|}
 & t & R_A & R_B & R_C \\
1 & t_1 & A_1 & B_1 & C_1 \\
2 & t_2 & A_2 & B_2 & C_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
n & t_n & A_n & B_n & C_n
\end{array}
$$

Figure 3.4 Data matrix representation

In this proposed model, there were three objective functions required to be measured: (i) to minimise cycle time (ii) to minimise number of workstations and (iii) to minimise the number of resources. These objective functions should be optimised by considering the resource constraint.

3.2.5 Evaluating Assembly Sequence

A feasible assembly sequence was evaluated according to three objectives: (i) to minimise cycle time (ii) to minimise the number of workstations (iii) to minimise the number of resources. According to Grzechca (2011), an interval when a task(s) can be assigned to the workstation is called cycle time. While grouping the tasks to the workstation, the total time taken to complete all tasks, also known as processing time, must not be greater than the cycle time. The current task was assigned to the next workstation whenever the processing time exceeds the cycle time. The first and second objectives were evaluated by considering the third objective. In this case, the tasks that use the same resources were assigned to one workstation.

The exemplary assembly presented in Figure 3.5 is the assembly of wall rack which consists of 6 metals and thirteen components. Two assembly tools and a machine were used for the purpose of assembling this product.

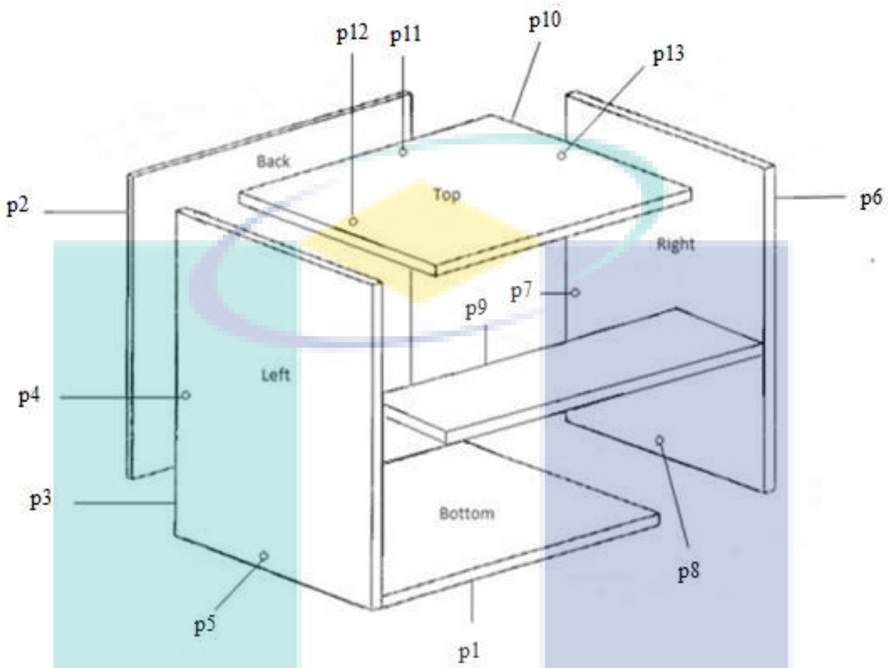


Figure 3.5 Assembly of wall rack

The first tool used to assemble parts (p1 and p2), (p1 and p3), (p1 and p5), (p1 and p6), (p1 and p8), (p2 and p3), (p2 and p4), (p2 and p6), (p2 and p7), (p2 and p10), (p3 and p9), (p3 and p10), (p6 and p9), (p6 and p10) and (p6 and p13) is a jig. The second tool is a blind rivet that is used to assemble parts (p1 and p5), (p1 and p8), (p2 and p4), (p2 and p7), (p2 and p11), (p3 and p12) and (p6 and p13). Meanwhile, a welding machine was used to assemble parts (p1 and p2), (p3 and p9) and (p6 and p9).

The establishment of liaison matrix in Table 3.1 was intended to record the assembly task for every part of the model. It is noted that the number in Table 3.1 now represent the assembly task. For instance, task 1 refers to the assembly relation between parts (p1 and p2). There is no assembly relation between parts (p1 and p4). Thus, the matrix is left empty.

Table 3.1 Liaison matrix for wall rack assembly

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14
p1	-	1	2		3	4		5						
p2		-	6	7		8	9			10	11			
p3			-						12	13		14		
p4				-										
p5					-									
p6						-			15	16			17	
p7							-							
p8								-						
p9									-					
p10										-				
p11											-			
p12												-		
p13													-	
p14														-

Meanwhile, De Fazio's question and answer (Q&A) procedure is applied for every task. For example, in order to perform task 2, task 1 must be done prior to performing task 2. Meanwhile, task 3 must be left undone until task 2 is completed. Table 3.2 records De Fazio's Question and Answer. The precedence constraints can be identified in the corresponding table. The precedence constraints for this product are stated as C[(1, 2), (1, 4), (2, 3), (12, 4), (4, 5), (6, 7), (12, 8), (8, 9), (9, 10), (10, 11), (6, 13), (13, 14), (12, 15), (8, 16), (16, 17)].

Table 3.2 De Fazio's Question and Answer for wall rack assembly

Task	Answer for question:	
	Q1	***Q2***
1	-	2, 4
2	1	3
3	2	-
4	12	5
5	4	-
6	-	7
7	6	-
8	12	9
9	8	-
10	9	11
11	10	-
12	-	-
13	6	14
14	13	-
15	12	-
16	8	17
17	16	-

Question 1, *Q1*: What tasks must be done prior to doing task *i*?

Question 2, *Q2*: What task must be left to be done after doing task *i*?

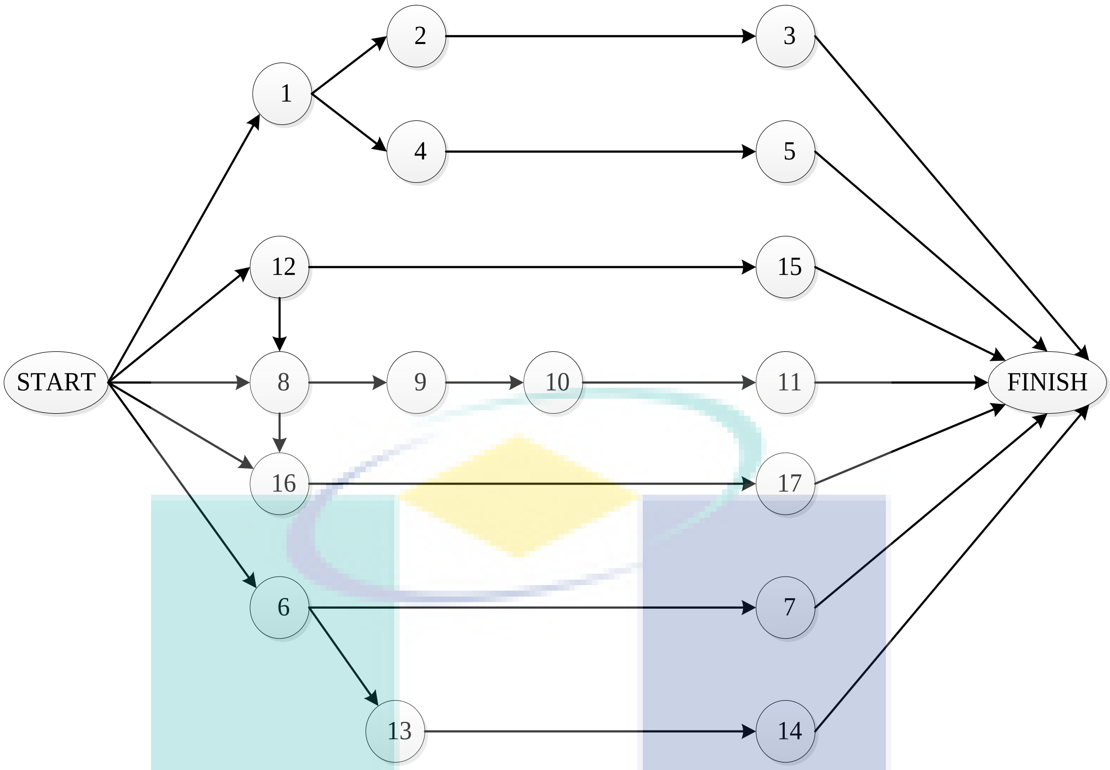


Figure 3.6 Precedence diagram for wall rack assembly

From the precedence constraints that were obtained from De Fazio's Q&A procedure, the precedence diagram is mapped as demonstrated in Figure 3.6. Table 3.3 shows the assembly data and information for the assembly problem. The resource, R represents the assembly tool and machine that are being used in the assembly process. For example, resource A (Jig) and C (Welding machine) are used to assemble task 1, whereas only resource B (Blind rivet) is required to assemble task 11.

Table 3.3 Data matrix for wall rack assembly

Task	Task time (minute)	Resource (R)
1	4	A, C
2	1	A
3	2	A, B
4	1	A
5	2	A, B
6	1	A
7	2	A, B
8	1	A
9	2	A, B
10	1	A
11	2	B
12	3	A, C
13	1	A
14	2	B
15	3	A, C
16	1	A
17	2	B

3.3 Objective Functions

While designing the line, the precedence constraints, number of workstation and required task time must be deliberated. The resource constraints should also be taken into consideration. These restrictions must be considered to achieve a balance line. The term 'cycle time' means maximum time allowed at each workstation. According to Grzechca (2011), cycle time is a period related to customer order and market demand. Cycle time can also be defined as the maximum time allowed at each workstation.

In this mathematical model, the first objective function is to minimise the cycle time represented by Eq. 3.1 and subjected to constraint in Eq. 3.4 which guarantees that the total task time in workstation j does not exceed the cycle time. The second objective for this problem is to minimise the number of workstations denoted by Eq. (3.2). Constraint (3.5) ensures that task i can only be assigned to workstation j. The last objective for this problem is to minimise the number of resources. The objective function is presented by Eq. (3.3), and subjected to constraint (3.6) to ensure that the number of resource units assigned to workstation does not exceed the available resource. Constraint (3.7) defines the precedence relationship between the tasks that should not be violated.

$$min\ f_1(x) = min\ ct \tag{3.1}$$

$$min\ f_2(x) = min\ m = \sum_{j=1}^{m_{max}} x_{ij} \tag{3.2}$$

$$min\ f_3(x) = min \sum_{r=1}^{R} S_{jr} \tag{3.3}$$

subjected to:

$$\sum_{i=1}^{n} pt_i . x_{ij} \leq ct \qquad \text{for } i=1,\ldots,m_{max} \tag{3.4}$$

$$\sum_{j=1}^{m_{max}} x_{ij} = 1 \qquad \text{for } i=1,\ldots,n \tag{3.5}$$

$$\sum_{r=1}^{R_{max}} S_{jr} \leq N_r \qquad \text{for } r=1,\ldots,R \tag{3.6}$$

$$\sum_{j=1}^{m_{max}} j.x_{ij} \leq \sum_{j=1}^{m_{max}} j.x_{kj} \qquad \text{for } (i,k) \in P \tag{3.7}$$

variables:

$$x_{ij} = \begin{cases} 1 & \text{if task } i \text{ is assigned to workstation } j \\ 0 & \text{else} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if there is task assigned to workstation } j \\ 0 & \text{else} \end{cases}$$

$S_{jr}=\begin{cases}1 & \text{if resource } r \text{ exists in workstation } j\\ 0 & \text{else}\end{cases}$

ct cycle time, $(i=1,\ldots, n)$

n number of task, $(j=1,\ldots, m)$

m number of workstation

pt_i processing time of task i

R number of resources, $(r=1,\ldots, R)$

P set of tasks (i,k) from a direct precedence relations

S_{jr} number of resource units of type r that assigned to workstation j

N_r number of available resource units of type r

3.3.1 Numerical Example

This section presents a numerical example of a feasible assembly sequence obtained from a precedence diagram for wall rack assembly (Figure 3.5). Some example of the sequences are: F_{seq1} [6, 1, 2, 13, 14, 7, 3, 12, 15, 4, 5, 8, 9, 10, 11, 16, 17] and F_{seq1} [1, 2, 3, 12, 15, 4, 5, 8, 9, 10, 11, 16, 17, 6, 13, 7, 14]. The sequences were evaluated based on three objectives: (i) to minimise cycle time (ii) to minimise the number of workstations (iii) to minimise the number of resources.

The assembly tasks were assigned to workstations by following the precedence and cycle time constraints. The total processing time (pt) in each workstation must not exceed the predetermined cycle time. In this example, the maximum allowable cycle time at each workstation, ct_{max} is 7 minutes. Figure 3.7 shows the task assignment of wall rack assembly for F_{seq1}.

Based on Figure 3.7, the first workstation (ST1) consists of 4 tasks with total processing time of 7 minutes. No other tasks can be assigned to ST1 as the total processing time in the workstation has achieved ct_{max}. Meanwhile, the total processing time for task 14, 7 and 3 in ST2 is 7 minutes. Whenever task 12 is included in ST2, the total processing time of current workstation will be 10 minutes, which exceeds the ct_{max}. Thus, task 12 must be assigned to a new workstation and similar approach is applied for the subsequent workstations.

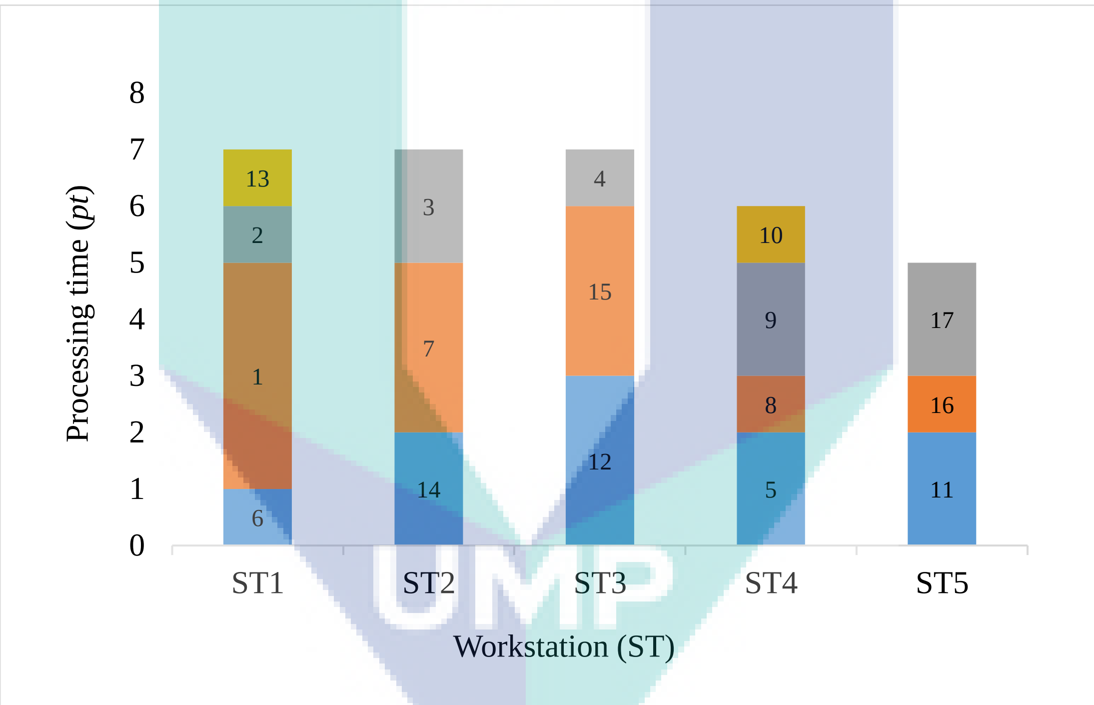


Figure 3.7 Task assignment for wall rack assembly (F_{seq1})

Table 3.4 and Table 3.5 present the task assignment for assembly of wall rack for F_{seq1} and F_{seq2} respectively. With reference to Table 3.4, task 6 requires resource A, task 1 needs resource A and C, whereas task 2 and 13 both require resource A. Therefore, there are 5 number of resources (4 units of resource A and 1 unit of resource C) needed to perform all tasks in ST1. The aim of this study is to minimise the number of resources used to assemble a product. Thus, the tasks that can be performed by the same resource will be assigned in the same workstation and will share the equipment. The number of resources used can be reduced to 2 (1 unit of resource A and 1 unit of resource C). This research objective will contribute to resource saving.

The similar approach is also applied for F_{seq1}in Table 3.5. In ST1, there is a total of 7 resources (4 units of resource A, 1 unit of resource B and 2 units of resources C) required to perform the corresponding tasks. By considering the aim of this research, the number of resources used can be reduced to 3 (1 unit of resource A, 1 unit of resource B and 1 unit of resource C).

Table 3.4 Example of assembly sequence evaluation for F_{seq1}

Name	**ST1**				**ST2**			**ST3**			**ST4**				**ST5**		
Task	6	1	2	13	14	7	3	12	15	4	5	8	9	10	11	16	17
ct	1	4	1	1	2	3	2	3	3	1	2	1	2	1	2	1	2
pt	7				7			7			6				5		
Type of resource (R)	A	A, C	A	A	B	A, B	A, B	A, C	A, C	A	A, B	A	A, B	A	B	A	B
Total R	2				2			2			2				2		

Table 3.5 Example of assembly sequence evaluation for F_{seq2}

Name	**ST1**				**ST2**					**ST3**							**ST4**
Task	1	2	3	12	15	4	5	8	9	10	11	16	17	6	13	7	14
ct	4	1	2	3	3	1	2	1	2	1	2	1	2	1	1	2	2
pt	10				9					10							2
Type of resource (R)	A, C	A	A, B	A, C	A, C	A	A, B	A	A, B	A	B	A	B	A	A	A, B	B
Total R	3				3					2							1

3.4 Algorithm Development and Testing

The latter phase in this research is the development of NSGA-II algorithm and testing. The representation style of the algorithm development should be compatible with the proposed model. In this method, permutation integer is applied to suit the ALB problem. NSGA-II has been chosen as an optimisation technique in this research since it accommodates to solve multi-objective optimisation problem (Deb et al., 2002; Bandyopadhyay and Bhattacharya, 2014; Zhao et al., 2015).

The NSGA-II algorithm is coded into the computer program. In this activity, MATLAB is used for the coding purpose. Later, the algorithm will be verified to ensure that the program gives the required output. For the numerical experiment purpose, five runs with different pseudo-random numbers are conducted for each problems and algorithms. The output obtained from each problem is combined and filtered to get non-dominated solutions. In addition to that, the following parameters have been used for algorithm testing.

Population size	: 20
Number of generations	: 200
Crossover probability, p_c	: 0.8
Mutation probability, p_m	: 0.3

The computational experiment was set up to test the proposed algorithm. In this work, six benchmark problems taken from open literature were used to test the performance of algorithm (Ponnambalam et al., 2000; Ağpak & Gökçen, 2005). Each problem consists of a number of task, cycle time and also task time. Since most problems do not have resource data, the data are randomly generated to suit the studied problem. Next, the performance of NSGA-II is compared within GA-based algorithm such as Hybrid Genetic Algorithm (HGA) and Multi-Objective Genetic Algorithm (MOGA). A detailed explanation on algorithm development and testing is provided in Chapter 4 in section 4.2 and section 4.5 respectively.

3.5 Industrial Data Collection

One of the stages involved in this research is industrial data collection. This section explaines the method used to collect data. In order to validate the ALBE-RC model and NSGA-II algorithm, data collection from practical application will be collected. The industrial data collection will be performed based on the following approach.

3.5.1 Observation of Assembly Plant

Firstly, an industrial visit that would enable data collection was organised. The visit was covered by a discussion with the Senior Manufacturing Engineer and co-workers related to the selected product and model. The other approach is the 'go-and-see' (*Genchi Genbutsu*) the actual situation in the production line (*Genba*). The motive behind these approaches is to have a better understanding of the real situation. In order to familiarize with the processes in each workstation, the assembly process is observed during this phase. Besides that, all components and tasks involved in each workstation are also identified.

3.5.2 Identification of Data Collection

Related assembly data such as assembly task, cycle time, precedence constraint and resources are collected. Apart from that, demand per day, working hour per day and number of employees are taken into account while collecting the data. The data collection was performed during the entire working hours. In average, six sets of cycle time data for each process were collected.

3.6 Simulation Phase

A simulation of existing layout was performed using WITNESS™ software to simulate the assembly line. The purpose of simulating existing layout is to validate the simulation model with actual layout. The simulation for each layout was carried out for 11 hours per shift and two shifts per day. The time unit for the simulation was set in second. Thus, the total simulation time for each layout is 79200 seconds.

3.7 Validation

This section presents the validation of the ALBE-RC model and the proposed NSGA-II algorithm. The validation of the proposed algorithm was divided into two stages. Firstly, a computational test was set up to test the performance of NSGA-II. The algorithm was tested using generic problems taken from open literature, which consist of different classes of problems. Later, the performance of the proposed algorithm was compared to the comparison algorithms. This approach is consistent with the approach used by Özcan and Toklu (2009), Ozbakir et al. (2011) and Wei and Chao (2011). Apart from that, Fettaka et al. (2012), Yang et al. (2013) and Triki et al. (2014) also used benchmark problems to test the algorithm. A detailed explanation on the NSGA-II test and results will be discussed in Chapter 4, section 4.5 and section 4.6 respectively.

Apart from the NSGA-II algorithm, the ALBE-RC model was also validated through an industrial case study. The validation stage was performed to get feedback from the industrial experts regarding the ALBE-RC model and the proposed algorithm, NSGA-II. Triki et al. (2014) used similar approach by performing a study in a manufacturing company to validate the proposed algorithm. Similarly, Chen et al. (2009) and Baykasoğlu and Akyol, (2012) validated the performance of the proposed method by performing an industrial case study in their works. Further discussion on this topic will be clarified in Chapter 5, section 5.2

3.8 Chapter Summary

This chapter had explained the research methodology that was performed throughout this research. In summary, this chapter had described the following points:

i. *Problem modelling*: This section presented an approach to establish ALBE-RC model. The modelling phase started with problem representation, followed by modelling procedure. Finally, steps on how to evaluate each assembly task was highlighted in this section.

ii. *Objective functions*: This part justified the objective function that were considered in this research to achieve a balanced line i.e. (i) to minimise the cycle time (ii) to minimise the number of workstations (iii) to minimise the number of resources. In fact, some restrictions such as precedence constraint, cycle time constraint and resource constraint also were put into consideration in order to achieve all objective functions.

iii. *Algorithm development and testing*: The development of the proposed algorithm, NSGA-II was briefly explained in this section. This part covered the representation style of the algorithm and coding phase. Last but not least, a validation phase was performed through a computational test to verify the performance of algorithm.

iv. *Industrial data collection*: This section discussed the method used to collect data. Two approaches were applied for this purpose i.e. (i) observation of assembly plant (ii) identification of data collection. Data from practical application were collected to validate the ALBE-RC model and NSGA-II algorithm.

v. *Simulation phase*: This part explained the simulation phase that was operated to get the overview on the assembly process. Besides that, the duration of simulation and also a few assumptions that were considered during the simulation were detailed out in this part.

vi. *Validation*: The validation methods for the proposed NSGA-II algorithm and the ALBE-RC model were clarified in this section. The proposed algorithm was justified by performing a computational experiment. Last but not least, the ALBE-RC model and NSGA-II algorithm were verified through an industrial case study.

CHAPTER 4

DEVELOPMENT OF OPTIMISATION ALGORITHM

`4.1 Introduction

This chapter proposes an algorithm to optimise the ALB-E problem with resource constraint (ALBE-RC). The activities in this phase are started off by establishing the solution procedure for ALBE-RC. Once the general solution procedure is established, the algorithm flow will be drafted by referring to the solution procedure. Next, the algorithm will be coded into computer program. In this stage, MATLAB software will be used for coding purpose. The algorithm will be modified to suit the optimisation problem. It is important to note that the algorithm will later be tested using generic problems from literature and then compared to the results from comparison algorithms. The algorithm in MATLAB code will later be tested and verified using test problems from literature.

4.2 NSGA-II Development

The second phase in this research is development of algorithm to optimise the ALBE-RC problem. The Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) was proposed as an approach to optimise the problem. NSGA-II procedure starts with initializing a random population P_i of size N_{pop}. The algorithm is then decoded into feasible sequences using topological sort. The fitness of feasible chromosomes is calculated by evaluating the objective functions.

Later, a non-dominated sorting approach is applied to generate Pareto-optimal set. The entire population is sorted using non-dominated sorting approach to identify the non-dominated set $F = (F_1, F_2, \ldots Fi)$. The parent population is filled by set F according to the non-domination rank. If $F > N_{pop}$, the last front will be selected based on higher crowding distance (*CD)*. Since NSGA-II uses the selection strategy based on crowding distance, it will give an estimation of the density of selected solutions.

The tournament competition between two random-pair of solutions from parent population is performed to determine the domination rank. The population will be sorted in decreasing rank according to each objective function. The solution with better rank is filled in parent pool. Meanwhile, the solution with same rank but remains in a less crowded area will be selected to be filled in parent pool. The tournament selection is repeated until the parent pool is fully occupied to generate children. New offspring population Q_i of size N_{pop} is generated from P_i by crossover and mutation operators. Later, P_i and Q_i are combined to form new population R_i of size $2N_{pop}$. The NSGA-II procedure is repeated until the termination criteria is met. The flowchart of NSGA-II is shown in Figure 4.1.

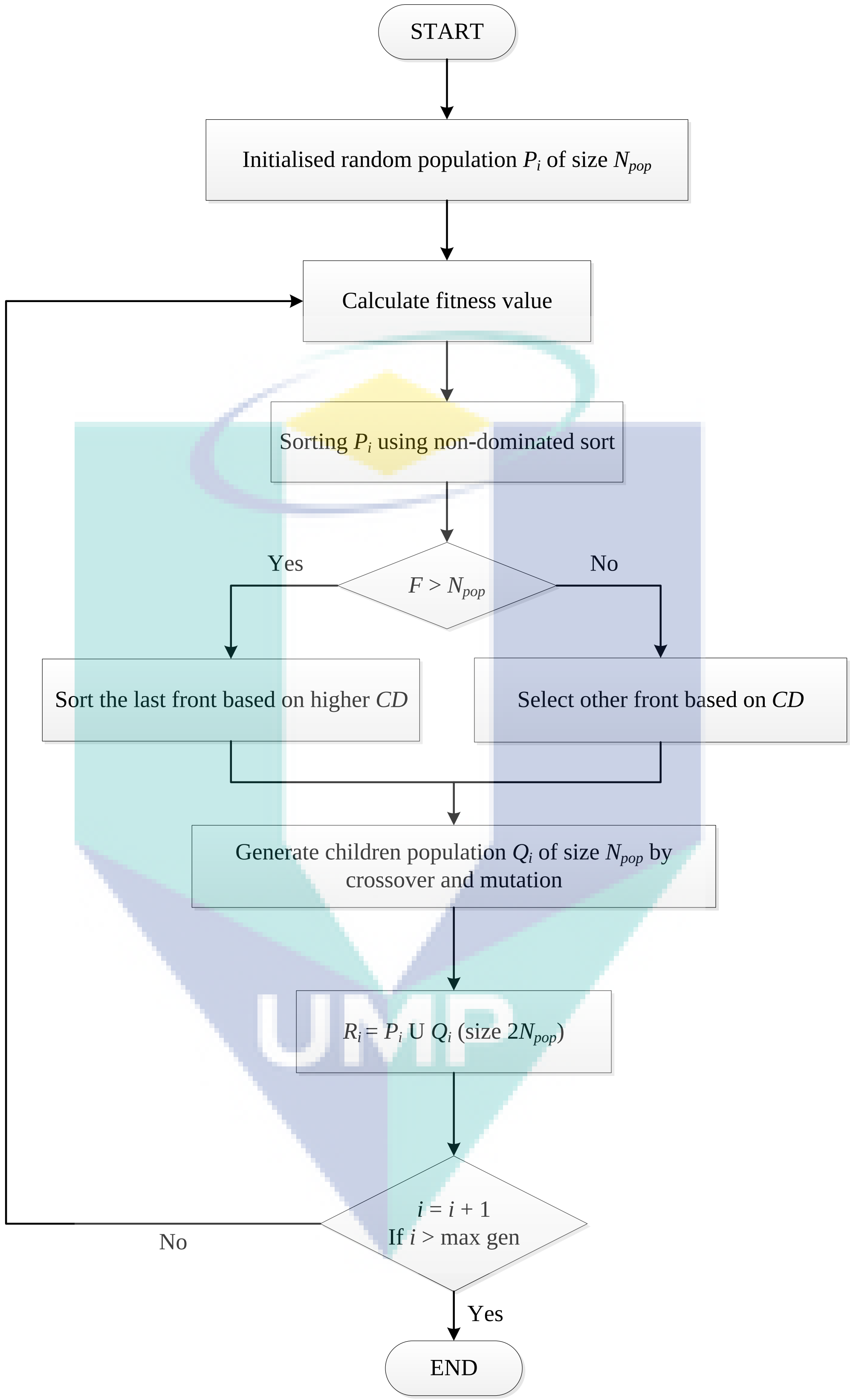


Figure 4.1 Flowchart of NSGA-II

4.3 Constraint Handling

This section explains a repairing strategy using topological sort to generate a feasible assembly sequence. Moreover, this approach had been adopted as a part of NSGA-II procedure to handle the precedence constraint. Topological sort is an ordering connection in a directed graph (Moon et al., 2002). This is also supported by Mohd Razali and Geraghty (2011) who used the repairing technique in GA.

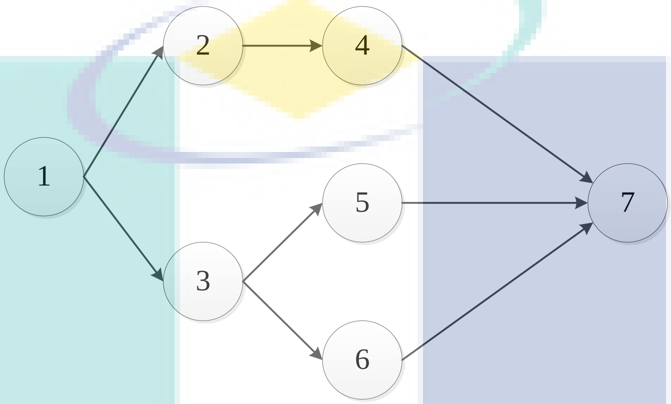


Figure 4.2 Example of precedence diagram for simple assembly line

Table 4.1 Details of assembly task

Task	Task time	Resource	
1	4	1	3
2	14	0	0
3	16	3	0
4	6	2	3
5	7	0	0
6	11	1	4
7	5	2	0

Figure 4.2 exhibits the example of precedence diagram for simple assembly line. The details of each task is tabulated in Table 4.1. Two infeasible populations obtained from the precedence diagram are considered: C_1[1, 4, 3, 7, 6, 2, 5] and C_2[7, 2, 5, 6, 4,

3, 1]. Table 4.2 shows a procedure to generate a feasible sequence from the precedence diagram. The repair mechanism of the infeasible populations using topological sort is clarified as following:

i. Identify available task from precedence diagram.
ii. Select an available task based on the first task that appears in chromosome.
iii. Remove the selected task from the precedence diagram.
iv. Repeat step (i) to step (iii) until all tasks are selected

Table 4.2 Procedure to generate a feasible sequence

C_1[1, 4, 3, 7, 6, 2, 5]		C_2[7, 2, 5, 6, 4, 3, 1]	
Available task(s)	**Selected task**	**Available task(s)**	**Selected task**
1	1	1	1
2, 3	3	2, 3	2
2, 5, 6	6	3, 4	4
2, 5	2	3	3
4, 5	4	5, 6	5
5	5	6	6
7	7	7	7
Feasible sequence, f_{seq1}: [1, 3, 6, 2, 4, 5, 7]		Feasible sequence, f_{seq2}: [1, 2, 4, 3, 5, 6, 7]	

4.4 Numerical Example of NSGA-II

This section presents a numerical example to explain how the NSGA-II works for ALBE-RC problem. NSGA-II procedure begins with the initialisation phase. It is followed by evaluation procedure, selection and finally reproduction phase.

4.4.1 Initialisation

In this step, the number of population (N_{pop}) is predetermined. The initial population with a random permutation from 1 to n (number of task) is produced. Originally, the NSGA-II is encoded by using either binary or real number (Deb et al., 2002; Fettaka et al., 2012). In this example, the N_{pop}= 4 whilst the initial population generated from the precedence diagram in Figure 4.2 are as follow:

$$\text{pop} = \begin{bmatrix} 1, 4, 3, 7, 6, 2, 5 \\ 7, 2, 5, 6, 4, 3, 1 \\ 1, 5, 7, 2, 6, 3, 4 \\ 1, 3, 7, 5, 2, 6, 4 \end{bmatrix}$$

Then, the algorithm is decoded to generate feasible sequences using topological sort as explained in section 4.3. The first available task found in the population will be selected. The following sequences represent the feasible sequences generated from the population after undergoing the repair mechanism.

$$\text{feasible sequence}=P_i=\begin{bmatrix} 1, 3, 6, 2, 4, 5, 7 \\ 1, 2, 4, 3, 5, 6, 7 \\ 1, 2, 3, 5, 6, 4, 7 \\ 1, 3, 5, 2, 6, 4, 7 \end{bmatrix}$$

4.4.2 Evaluation

In this step, the feasible sequence is evaluated according to objective functions in order to calculate the fitness of solution. The fitness will measure how good the solution is as well as the closeness of the chromosome to the optimal one. In this example, the predetermined maximum cycle time, ct_{max} is equivalent to 22 time unit. By using the evaluation approach as proposed in section 3.3.1, the fitness values for all feasible sequences obtained from the corresponding examples are shown in Table 4.3.

Table 4.3 Fitness values for feasible sequences

Feasible sequences	Cycle time, *ct*	Number of workstations, *nws*	Resources, *r*
1 3 6 2 4 5 7	20	4	7
1 2 4 3 5 6 7	22	4	7
1 2 3 5 6 4 7	18	4	6
1 3 5 2 6 4 7	22	3	6

4.4.3 Selection

Later, the non-dominated sorting approach is applied to generate Pareto-optimal set. The entire population is sorted using non-dominated sorting approach to identify the non-dominated set and solution front, $F = (F_1, F_2,\ldots, F_i)$. The best non-dominated solutions found from the population will belong to front 1, F_1. If the size of F_1 is smaller than N_{pop}, all solutions of F_1 will be chosen to be filled in selection pool. The remaining member of population will be chosen from F_2 and subsequent solution fronts based on non-domination level and *CD* until the parent population is filled. If $F_i > N_{pop}$, the solution from i^{th} front, F_i, will be selected based on higher *CD* value. Crowding distance gives an estimation of the density of particular solution. The following algorithm are used to calculate the *CD* as proposed by Deb (2001).

Step 1 Call the number of solution in F as $Q = |F|$. For each I in the set, first assign $d_i = 0$.

Step 2 For each objective function $m = 1, 2,\ldots, M$, sort the set in descending order of f_m.

Step 3 For $m = 1,2,\ldots$, M, assign maximum (m_{max}) and minimum (m_{min}) value of each objective.

Step 4 Calculate $d_i^m = \frac{Qi_{upper}^m - Qi_{lower}^m}{m_{max} - m_{min}}$ (4. 1)

where, Qi_{upper}^m is the upper value of the nearest m^{th} objective for solution i. Qi_{lower}^m represents the lower value of the nearest m^{th} objective for solution i.

Step 5 Calculate the total of d_i^m

$CD_i = \sum_{m=1}^{M} d_i^m$ (4. 2)

The following information were used for this example:

max ct=22 time unit

min ct= max task time in data=16 time unit

max nws= sum of task time/min ct =63/16=4

min nws= sum of task time/max ct= 63/22=3

max r= max nws×total r type=4×4=16

min r=number of r type=4

The following calculation measures the *CD* of solution 1:

$$d_1^1 = \frac{ct_{upper} - ct_{lower}}{ct_{max} - ct_{min}} = \frac{22-18}{22-16} = 0.667$$

$$d_1^2 = \frac{nws_{upper} - nws_{lower}}{nws_{max} - nws_{min}} = \frac{4-3}{4-3} = 1$$

$$d_1^3 = \frac{r_{upper} - r_{lower}}{r_{max} - r_{min}} = \frac{7-6}{16-4} = 0.083$$

$$CD_1 = \sum_{m=1}^{M} d_i^m = d_1^1 + d_1^2 + d_1^3 = 0.063 + 1 + 0.083 = 1.750$$

Table 4.4 presents the value of *CD* for each solution. The value of crowding distance obtained for solution 1 is 1.750, similar to solution 4. Solutions 2 and 3 recorded 1.416 and 0.416 of distance values respectively.

Table 4.4 Crowding distance for each solution

Solution	***CD***
1	1.750
2	1.416
3	0.416
4	1.750

Selection process is conducted using tournament selection approach. The purpose of this step is to select chromosomes from the selection pool to be the parent solutions. A random-pair of solution is generated from the selection pool. The tournament selection between a random-pair of solution is performed to determine the domination rank. Solution with a better rank (front) is chosen to be filled in parent pool. Meanwhile, solutions that have the same rank are compared to identify the ones with larger *CD*. The tournament competition is repeated until the parent pool is fully occupied (equivalent to N_{pop}) to generate new offspring.

The selection method for all solutions is tabulated in Table 4.5. From the table, it is obvious that solutions 3 and 4 are both selected from non-dominated set. Meanwhile, solutions 1 and 2 are selected based on their crowding distance. Based on the value of *CD* in Table 4.4, solution 1 has higher *CD* (1.750) as compared to solution 2 (1.416). Hence, solution 1 must be emphasized more than solution 2.

Table 4.5 Selection method

Selection method	Solution	Sequence
Non-dominated sorting	3 4	1, 2, 3, 5, 6, 4, 7 1, 3, 5, 2, 6, 4, 7
Crowding distance	1 2	1, 3, 6, 2, 4, 5, 7 1, 2, 4, 3, 5, 6, 7

4.4.4 Reproduction

New offspring population Q_i of size N_{pop} is generated from P_i by crossover and mutation operators. A process in which a new offspring or child is produced by taking two parent solutions is known as crossover (Sivanandam and Deepa, 2007). In this algorithm, Partially Matched Crossover (PMX) is applied. The following procedures illustrate how PMX works:

i. Consider the two parents, P_1 and P_2 as shown in Figure 4.3.
ii. The lines mark the selected cross-points.
iii. The middle sections are remained to produce offspring, C_1 and C_2.
iv. The blank positions of children 1, C_1 inherits P_2 from its left.
v. Only remaining numbers in P_2 will be filled in C_1.
vi. Inversely, the blank position of C_2 inherits P_1 from its left.
vii. The remaining numbers in P_1 is filled in C_2.
viii. The resulting children C_1 and C_2 are shown in Figure 4.3.
ix. Similar steps are applied for the subsequent parents P_3 and P_4 to produce children C_3 and C_4 as shown in Figure 4.4.

P_1	1	5	7	2	6	3	4
P_2	1	3	7	5	2	6	4
C_1	1	3	7	2	6	5	4
C_2	1	7	6	5	2	3	4

Figure 4.3 Partially matched crossover for P_1 and P_2

P_3	1	4	3	7	6	2	5
P_4	7	2	5	6	4	3	1
C_3	2	5	4	7	6	3	1
C_4	1	3	7	6	4	2	5

Figure 4.4 Partially matched crossover for P_3 and P_4

Mutation operation helps to maintain the diversity of the population by prevent the algorithm from getting trapped in a local minimum. In this study, swapping mutation is applied to the chromosome after crossover operation. A random position in the chromosome is chosen and the position of the chromosomes are swapped. For example, consider the children 1, C_1 as Figure 4.5.

C_1 = 1 3 7 | 2 6 5 4

C_1 = 2 6 5 4 1 3 7

Figure 4.5 Mutation of *C1*

Later, parent and offspring populations are combined to form population $R_i = P_i \cup Q_i$ od size $2N_{pop}$. The NSGA-II procedure is repeated until the termination criteria is met.

4.5 Computational Test

A computational test is set up to test and validate the proposed algorithm. The aim of the experiment is to test the performance of the proposed optimisation algorithm, NSGA-II as compared to Multi-Objective Genetic Algorithm (MOGA) and Hybrid Genetic Algorithm (HGA). The MOGA and HGA were selected because these GA approaches are developed as an optimisation tool for NP-hard optimisation problem as explained in Chapter 2, section 2.5.1. In this work, six benchmark problems are used to test the optimisation algorithm. Each problem consists of a number of tasks, cycle time and also task time except Problem 1 that has an extra information on the resource usage. All test problems excluding the problem by Ağpak and Gökçen (2005) have been modified by including resources but the original information are preserved. Table 4.6 shows the classification of problem size as proposed by Otto et al. (2011).

Table 4.6 Classification of problem size

Element	**Small**	**Medium**	**Large**
Number of tasks, n	$n \leq 20$	$20 \leq n \leq 70$	$n \geq 70$

Small size problems were taken from Ağpak and Gökçen (2005) and Ponnambalam et al., (2000) whereas medium and large problems are available on an online database for assembly line balancing research: www.assembly-line-balancing.de (Scholl, 1993). A comparable data sets were exploited by Gonçalves and Almeida (2002) and Özcan & Toklu (2009) in order to test the proposed algorithms. Wei and Chao (2011) and Tuncel and Topaloglu (2013) also used the same data sets in their research. The benchmark problems are summarised in Table 4.7.

Table 4.7 Summary of benchmark problems

Size of problem	Problem	Number of tasks, *n*	Cycle time, *ct* (s)
Small	Problem 1 (Ağpak & Gökçen, 2005)	11	7
	Problem 2 (Ponnambalam et al., 2000)	12	10
Medium	Buxey (Scholl, 1993)	29	37
	Kilbridge (Scholl, 1993)	45	69
Large	Wee-Mag (Scholl, 1993)	75	69
	Lutz (Scholl, 1993)	89	35

The proposed algorithm were coded in MATLAB and the experiments were executed on a Windows 8, Intel® Core™ i5-4210U CPU 1.70 GHz with 4 GB of RAM. The following parameters had been used to run the experiments:

Population size : 20
Number of generations : 200
Crossover probability, p_c : 0.8
Mutation probability, p_m : 0.3

To evaluate the performance of the proposed algorithms, five performance indicators were measured as proposed by Deb (2001). Furthermore, most of the indicators were used by Fettaka et al. (2012) and Yoosefelahi et al, (2012) to validate the performance of the algorithm. Further explanation on the following indicators was discussed in Chapter 2, section 2.6.

vi. Number of Non-Dominated Solution, *NDS*
vii. Error Ratio, *ER*
viii. Generational Distance, *GD*
ix. *Spacing*
x. Maximum Spread, $Spread_{max}$

4.6 Optimisation Results

The results of algorithm performances for MOGA, HGA and NSGA-II are presented in Table 4.8 and discussed in the following sections. The summary of each algorithm performance is as illustrated in the following graphical.

Table 4.8 Algorithms performance of MOGA, HGA and NSGA-II

Problem	Algorithm	*NDS*	*ER*	*GD*	*Spacing*	*Spread$_{max}$*
Problem 1	MOGA	1	0.5	0.5	**0**	1.7321
	HGA	1	0.5	0.5	**0**	1.7321
	NSGA-II	**3**	**0**	**0**	**0**	2.8284
Problem 2	MOGA	**3**	**0**	**0**	**0.4714**	**4.1231**
	HGA	**3**	**0**	**0**	**0.4714**	**4.1231**
	NSGA-II	**3**	**0**	**0**	**0.4714**	**4.1231**
Buxey	MOGA	5	0.1667	0.1667	1.2134	10.247
	HGA	5	0.5	0.6243	0.5	**13.9642**
	NSGA-II	**7**	**0**	**0**	**0.4949**	10.247
Kilbridge	MOGA	0	1	1.9255	**0.3499**	**15.3623**
	HGA	1	0.8333	0.9714	1.2134	14.0712
	NSGA-II	**6**	**0**	**0**	0.7454	11.5758
Wee-Mag	MOGA	1	0.875	2.903	**1.9365**	20.3224
	HGA	7	0.5882	1.1236	2.5219	**31.7805**
	NSGA-II	**10**	**0**	**0**	7.8358	29.0172
Lutz2	MOGA	5	0.6154	0.855	1.8138	33.8526
	HGA	6	0.5714	0.8257	1.2778	**34.7131**
	NSGA-II	**8**	**0.2**	**0.2**	**1.005**	27.2029

*The values in bold show the best results

The optimisation results in Table 4.8 revealed that NSGA-II demonstrated consistent performances in three indicators i.e. (i) Number of Non-Dominated Solutions, *NDS* (ii) Error Ratio, *ER* (iii) Generational Distance, *GD* for all test problems. It is apparent from this table that NSGA-II showed inconsistent performance to obtain better *Spacing* as compared to MOGA and HGA. In contrast to that, the optimisation results indicate that NSGA-II has poor performance in finding better spread function as compared to the other comparison algorithms.

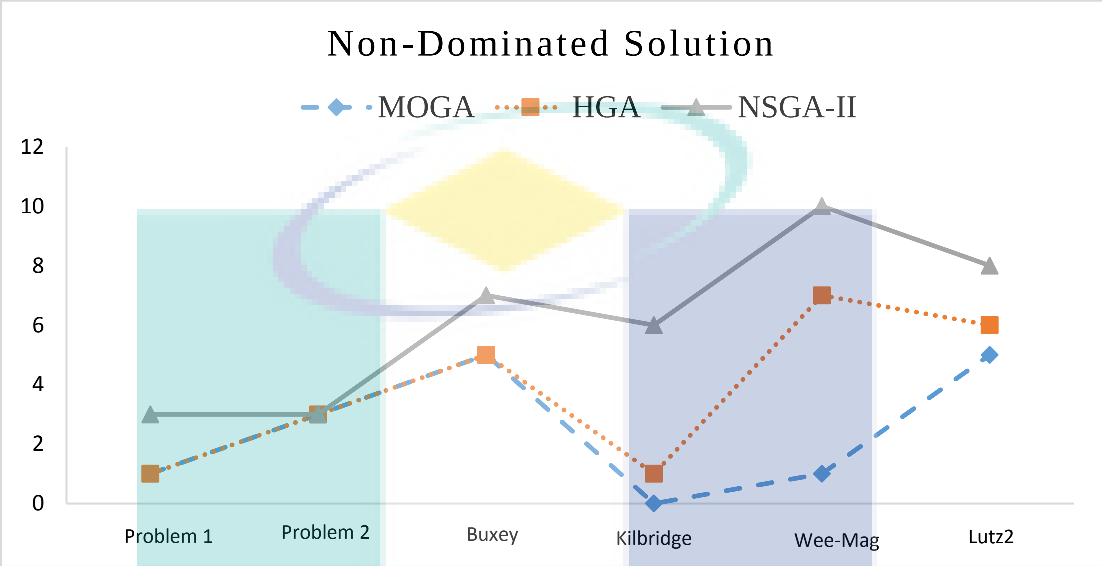


Figure 4.6 Number of Non-Dominated Solution for benchmark problems

Figure 4.6 exhibits the number of non-dominated solution for benchmark problem that were obtained by three comparison algorithms. The results from the computational test show that NSGA-II performed best in finding the non-dominated solutions for all six problems since NSGA-II used a crowding distance approach as the selection strategy. Hence, it can be concluded that NSGA-II has the ability to explore the search space as compared to other algorithms.

Graph of Error Ratio for six benchmark problems is illustrated in Figure 4.7. Small *ER* will increase the accuracy of the solution. The *ER* metric will take any number between 0 and 1. This implies that when the value of *ER* is zero, all solutions are member of Pareto-optimal set. However, none of the solution is a member of Pareto-optimal set as the value of *ER* equals to one. The results for all tested problems show that NSGA-II has the smallest *ER* among the other two comparison algorithms. Therefore, it can be concluded that NSGA-II has better accuracy of solution.

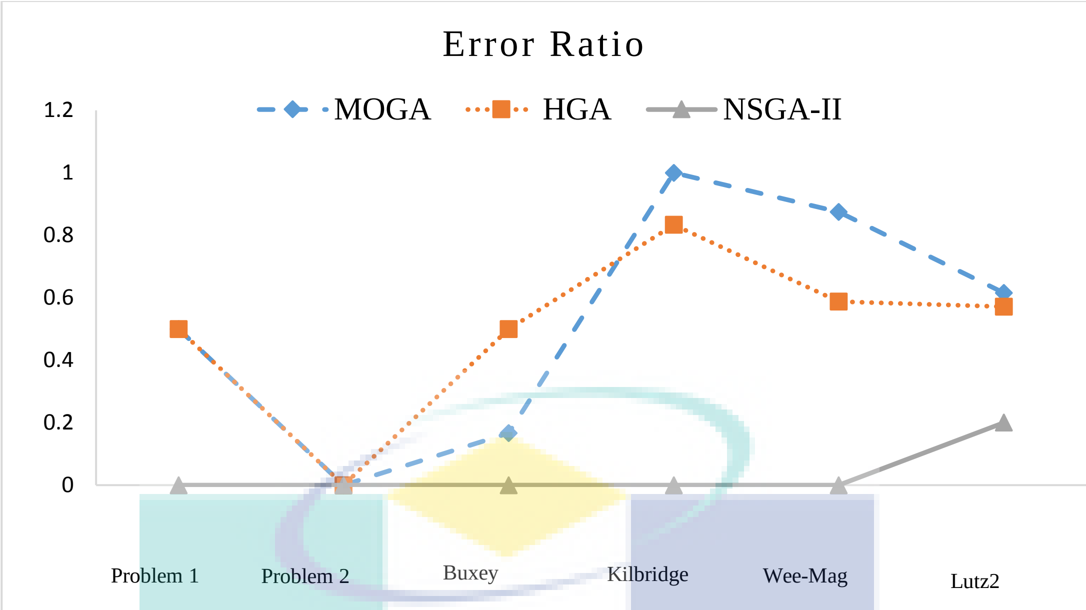


Figure 4.7 Graph of Error Ratio for benchmark problems

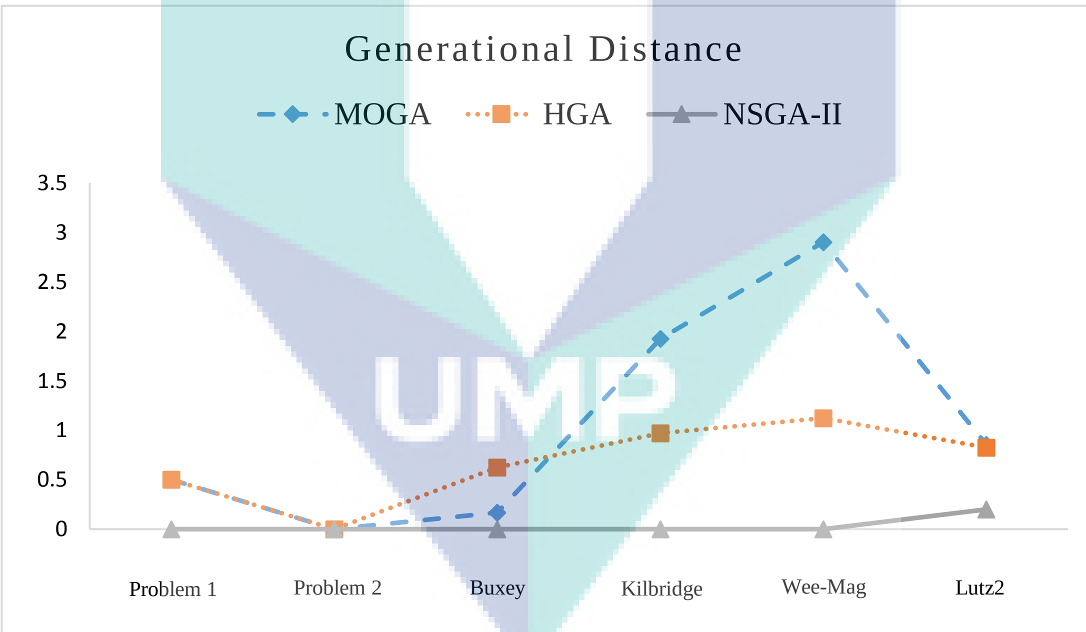


Figure 4.8 Graph of Generational Distance for benchmark problems

The graph plotted in Figure 4.8 shows the Generational Distance for six benchmark problems. Small *GD* leads to high accuracy of solution. In addition to the result of Error Ratio, the result of *GD* also showed that the performance of NSGA-II dominates the performance of MOGA and HGA in having high accuracy of solution. As expected, NSGA-II has the smallest value of *GD* for all size of problems as compared to MOGA and HGA.

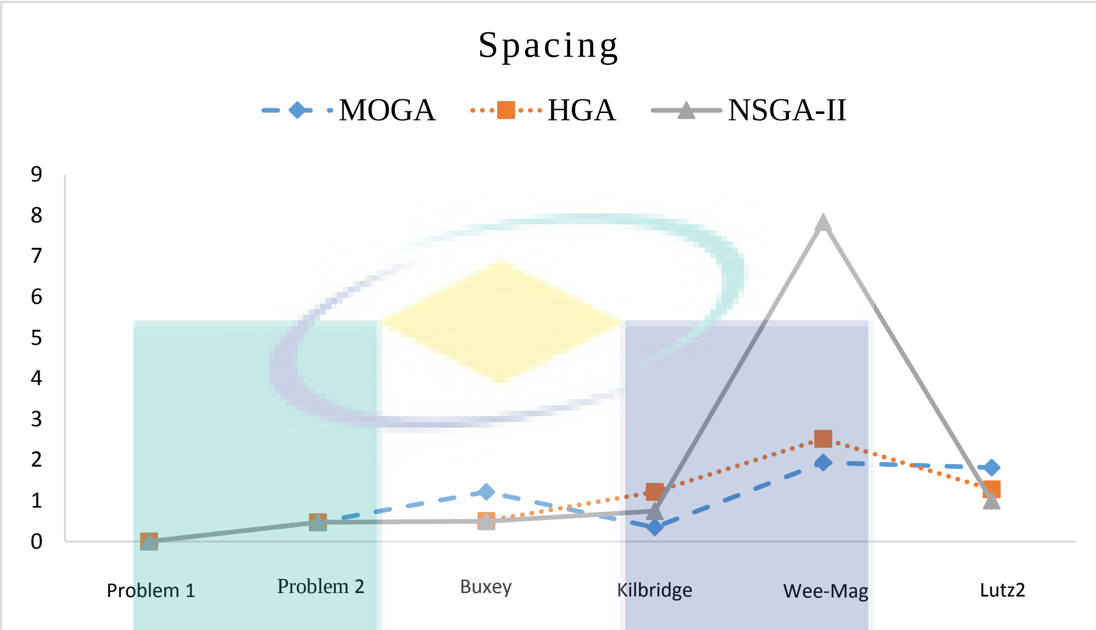


Figure 4.9 Graph of *Spacing* for benchmark problems

An algorithm that has a smaller *Spacing* is better instead of having larger *Spacing* as it can lead to a uniformed distribution of solution. Usually, algorithm that has higher number of solutions will produce better spacing. From Figure 4.9, it is apparent that NSGA-II only shows better *Spacing* in Buxey and Lutz2 problems. This is because, the number of solutions generated by NSGA-II is less than the number of solutions generated by MOGA and HGA. However, most of the solutions generated by NSGA-II lie on Pareto-optimal front. Since spacing metric will measure the distance of all solutions generated by the algorithm, when the number of solutions generated is small, the distribution of the solution would be worse.

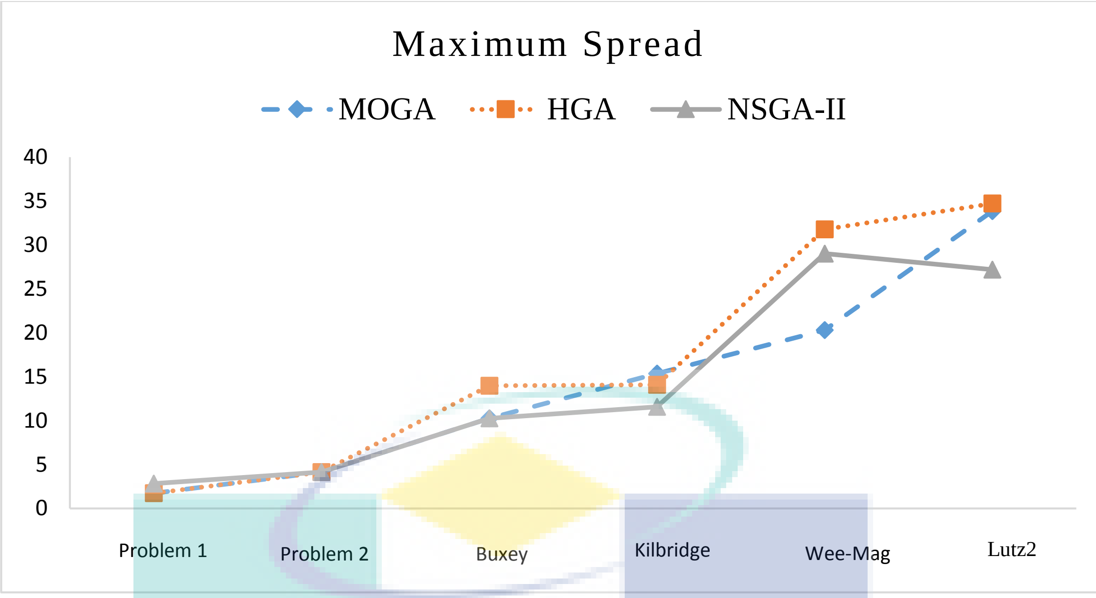


Figure 4.10 Graph of Maximum Spread for benchmark problems

Figure 4.10 presents the graph of Maximum Spread (*Spread*$_{max}$) of benchmark problems found by three algorithms. The spread of solution is determined by the maximum spread. The distance of two extreme solutions correspond from 1st objective function to m^{th} objective function will measure the performance of the algorithm. Large *Spread*$_{max}$ shows that the algorithm has better spread of solution. As a consequence of the computational experiment, NSGA-II only found better spread function in Problem 1. Other problems did not show any performance in this aspect. The reason behind this is that the solutions generated by the algorithm has the capability to move towards the Pareto-optimal front instead of lying on the extreme solution.

The results of the computational test revealed that NSGA-II has better performance as compared to MOGA and HGA since it has higher number of non-dominated solution (*NDS*), has smaller Error Ratio (*ER*) as well as smaller Generational Distance (*GD*). NSGA-II uses the selection strategy based on crowding distance. Thus, it gives an estimation of the density of selected solutions. In spite of that, the population will be sorted in decreasing rank of level according to each objective function. The individuals of a given population are sorted based on the level of non-domination and crowding distance.

In comparison with MOGA and HGA, NSGA-II has the smallest Error Ratio. *ER* metric is defined as the ratio of non-member of Pareto-optimal front to the number of non-dominated solutions. The results of the performance measure of the algorithms clearly show that all solutions generated by NSGA-II are members of the Pareto-optimal front. Thus, it can be concluded that NSGA-II has better convergence and the ability to come nearer to the Pareto-optimal front as compared to MOGA and HGA.

Since all solutions generated by NSGA-II lie on Pareto-optimal front, all test problems record the lowest value of Generational Distance than MOGA and HGA. The average distance of non-dominated solutions from Pareto-optimal front is equals to zero. NSGA-II produced better *Spacing* only in certain problem i.e. Buxey and Lutz2. Instead of having higher *NDS*, the distribution of solution will also influence the algorithm in achieving a better *Spacing*. The results of the performance measure of the algorithm indicate that NSGA-II shows an inconsistent performance in *Spacing* indicator. Similar trend of performance can be seen on MOGA.

On the other hand, for $Spread_{max}$, NSGA-II only show good performance in Problem 1. It is impossible to measure the distance between two extreme solutions in the objective space since the solutions generated by NSGA-II algorithm move towards the Pareto-optimal front. Nevertheless, both MOGA and HGA algorithms are inconsistently performed in $Spread_{max}$ indicator. From the computational test, it can be concluded that HGA is the worst algorithm to use to solve multi-objective optimisation problem. The results of the algorithm performance show that HGA is only good in a single performance ($Spread_{max}$) for Buxey and Wee-Mag problem. Meanwhile, MOGA algorithm shows an average performance in most indicators.

The results indicate that NSGA-II consistently performed in all test problems in three performance indicators i.e. (i) Number of Non-Dominated Solution, *NDS* (ii) Error Ratio, *ER* (iii) Generational Distance, *GD*. As a result, it is adequate to prove that the proposed NSGA-II outperform the other two comparison algorithms, MOGA and HGA for multi-objective optimisation problem.

4.7 Chapter Summary

The objective of this chapter is to explain the development of optimisation algorithm, NSGA-II. The development of NSGA-II begins with establishing the solution procedure for ALBE-RC. In this phase, MATLAB software has been used for the coding purpose. Following that, a computational test is conducted to investigate the performance of the proposed algorithm. The results tabulated in Table 4.4 show that NSGA-II performed better in finding the number of non-dominated solution (*NDS*), have smaller Error Ratio (*ER*) and Generational Distance (*GD*) in all test problems with different size. Out of the five performance measures that were used to compare the algorithms, NSGA-II performed better in three indicators. This indicates that NSGA-II has the ability to explore the entire search space and has better accuracy of solution towards the Pareto-optimal front.

CHAPTER 5

CASE STUDY AND VALIDATION

5.1 Introduction

This chapter presents an industrial case study that was conducted to validate the ALBE-RC model and NSGA-II algorithm. In the previous chapter, the NSGA-II was tested with generic problems from literature. The results from the computational test revealed that the proposed algorithm, NSGA-II performed in finding the non-dominated solution and have solution accuracy towards Pareto-optimal front.

Apart from that, this chapter also provides a brief background of company and product, current production layout, data collection, simulation of current layout, optimised layout and validated layout. Last but not least, a validation of the research is also presented in this chapter. The validation was performed to get feedback from the industrial expert on the ALBE-RC model and on the proposed algorithm, NSGA-II. The final section will summarise the objective of this chapter as well as the outcome of the validation phase.

5.2 Case Study

This section exhibits an industrial case study that had been conducted throughout this research. The industrial case study was performed in BI Technologies Sdn. Bhd, a company located in Kuantan, Malaysia. *HM72A-10* series model, which is a type of moulded inductor was chosen for the case study. *HM72A-10* was selected as the product running on line is a type of single model. Related assembly data such as assembly task, cycle time, precedence constraint and resources had been collected and manually recorded. The other essential data such as demand per day, output per day, working hour per day and number of employees were taken into account during data collection.

5.2.1 Product and Company Background

TT Electronics is a United Kingdom based manufacturer that produces sensing and control for industrial and car makers, advanced components and integrated manufacturing services (IMS). The advanced components provide engineered components solutions such as resistors, power and hybrid devices, magnetics and connectors. The magnetics components are handled by BI Technologies Corporation Sdn. Bhd. that is located in Kuantan, Malaysia. BI Technologies Corporation is a wholly owned subsidiary by TT Electronics.

Their product design team are focused on custom and semi-custom product based on customers' needs. The company's vision is to become one of the world's largest manufacturers of passive electronic components. The products produced by the company include magnetic components, power and signal, inductors SMD *(Surface Mount Device)* and through hole, moulded inductor, and lamination transformer.

The company is divided into several production section such as magnetic line, moulded inductor, and Agilent. For this case study, the moulded inductor production section is selected as the product running on the line is a type of single model. Only *HM72A-10* series model was running on the line during the data collection. *HM72A-10* series is a type of moulded inductor. This class of product is a high power low cost moulded SMD inductor which is typically used in electronic device such as computer.

5.2.2 Production Layout

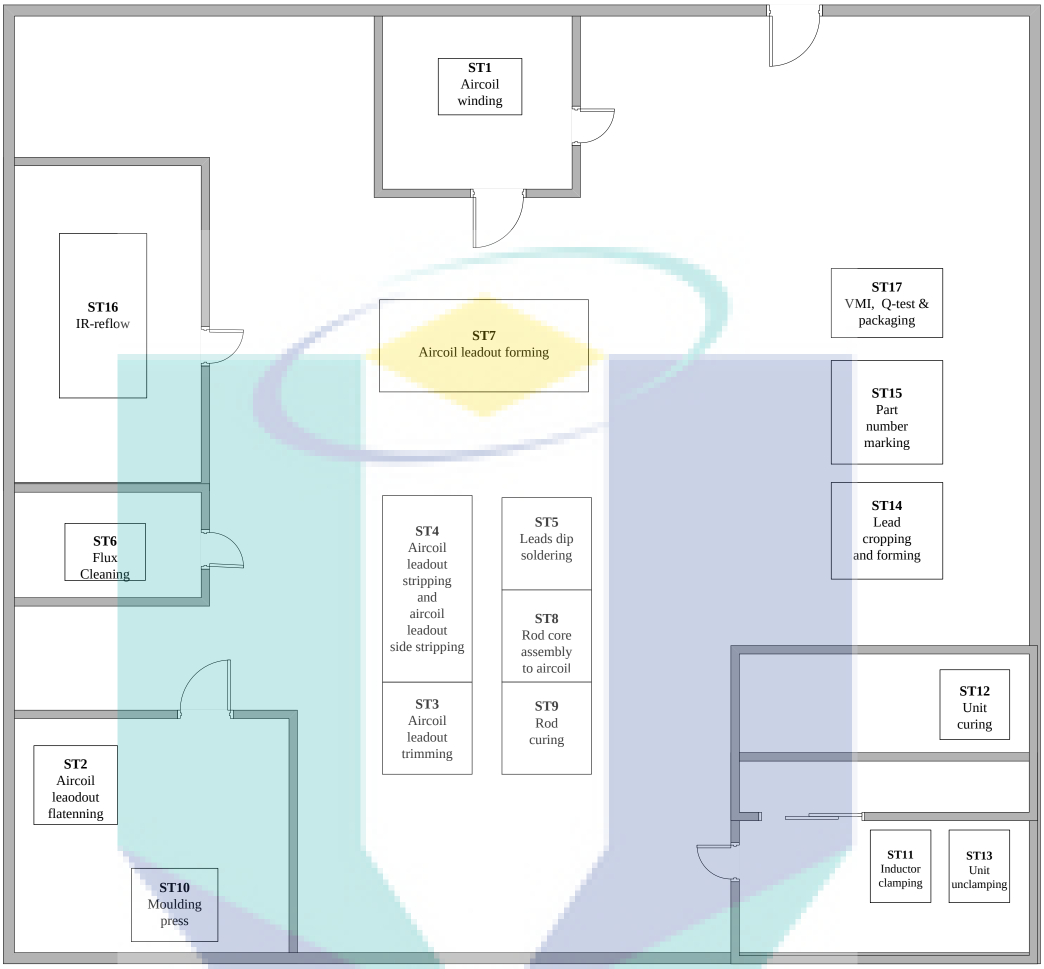


Figure 5.1 Current production layout

Figure 5.1 exhibits the current production layout. The production line is occupied with 18 workstations, consisting of 13 workers. There were 19 tasks that need to be performed to produce a single finished product (*HM72A-10*). Specifically, the assembly process starts with aircoil winding (a_1) performed in workstation 1 (ST1), by worker 1 (W1) with the average of cycle time being 5.1 seconds. Then, the process is continued with aircoil lead out flattening (a_2) by worker 2 (W2) in workstation 2 (ST2). The average cycle time that was recorded for a_2 is 7.8 seconds. Later, worker 3 (W3) run aircoil lead out trimming (a_3) with the average of cycle time recorded is 6.1 seconds was carried out in workstation 3 (ST3).

Aircoil leadout stripping (a_4) was performed by worker 4 (W4) meanwhile task a_5 which is aircoil leadout side stripping is run by worker 5 (W5). Both tasks were performed in workstation 4 (ST4) right after completing task a_3. An average of 8.3 seconds of cycle time was required to execute task a_4 whereas an average of 7.8 seconds of cycle time was needed to complete task a_5. Leads dip soldering (a_6), flux cleaning (a_7), aircoil lead out forming (a_8) are individually executed in workstation 5 (ST5), workstation 6 (ST6) and workstation 7 (ST7) with the average of cycle time being 4.5 seconds, 0.8 seconds and 4.4 seconds respectively. Worker 6 (W6) was assigned in ST6 and ST7 to perform both tasks. Task a_8 was performed by worker 7 (W7).

The next process continues in workstation 8 (ST8) by which task a_9 (rod core assembly to aircoil) was performed. The average cycle time required by the process was 4.6 seconds. After the rod core was assembled to aircoil, task a_{10} (rod curing) was operated in workstation 9 (ST9). The process took exactly 4.0 seconds of the average of cycle time. Similar worker; that is worker 8 (W8) was assigned to two workstations to conducted task a_9 and a_{10}. Worker 9 (W9) was assigned to workstation 10 (ST10) to perform task a_{11} (moulding press) with the average of cycle time equals to 8.1 seconds.

The process was continued with inductor clamping (a_{12}), running in workstation 11 (ST11) with average of cycle time being 3.2 seconds. After the inductor clamping was completed, the unit was then put into oven for curing process for exactly 9.0 seconds of cycle time. The task is labelled as a_{13} (unit curing) and it was conducted in workstation 12 (ST12). Workstation 13 (ST13) performed unit unclamping from tongs (a_{14}) with the average cycle time recorded being 1.6 seconds. These three tasks; a_{12}, a_{13} and a_{14} were operated by a single worker; worker 10 (W10).

The production of the aforementioned product was then continued with lead cropping and forming (a_{15}), part number marking (a_{16}) and IR-reflow (a17) that was individually performed in workstation 14 (ST14), workstation 15 (ST15) and workstation 16 (ST16). The average cycle time of task a_{15}, a_{16} as well as a_{17} were 4.7 seconds, 2.2 seconds and 2.3 seconds respectively. Worker 11 (W11) had been assigned to ST14 to run the corresponding task. Meanwhile, task a_{16} and a_{17} were done by worker 12 (W12). In the last workstation, ST17, two tasks were needed to be performed by worker 13 (W13) in order to finish the product; (i) VMI and Q-factor test (a_{18}) (ii) Packaging (a_{19}). Their respective average cycle time were 6.4 seconds and 1.5 seconds.

5.2.3 Data Collection

The data collection phase begun with an industrial visit to the company. During the visit, an interview and discussion session regarding the production of *HM72A-10* series model were conducted with the company expert. It is crucial to go and see the real situation in the production line to have a better understanding on the related product. For that purpose, two approaches had been applied to collect the data i.e. (i) *Genchi Genbutsu* (go and see) (ii) *Genba* (real place). Later, the data that were collected will be modelled according to the proposed model as explained in Chapter 3, section 3.1.

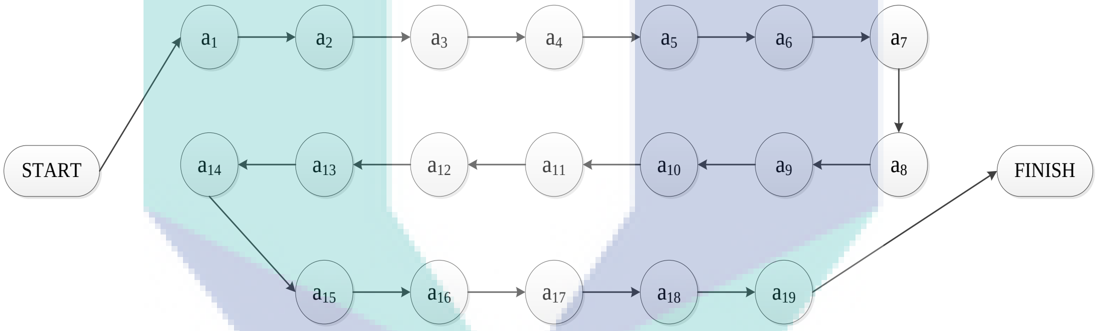


Figure 5.2 Precedence diagram for *HM72A-10*

Figure 5.2 shows the precedence diagram for *HM72A-10*. Meanwhile, Table 5.1 exhibits *HM72A-10* data collection. The data in the table clearly presents what type of resources were used while performing each task including the worker assignment. Besides that, the distribution of cycle time for each task is also tabulated in the Table 5.1. From the data that was collected, it shows that flux cleaning is the fastest process among other tasks with the average cycle time being 0.8 seconds. The longest processing time was recorded by task a_{13} (unit curing) with the average cycle time being 9.0 seconds. This task becomes the bottleneck of the line and thus, it will cause high idle time in other workstations.

Table 5.1 HM72A-10 assembly data

Work element		Resources		Cycle time (s)						Distribution
		Machine, tool	Worker	1	2	3	4	5	6	
a_1	Aircoil winding	- Auto CNC Aircoil Machine	W1	5.0	5.4	4.9	5.1	5.2	5.2	Log Normal (5.1,0.17)
a_2	Aircoil leadout flattening	- Pneumatic press 1	W2	8.2	7.3	6.8	8.5	8.2	7.8	Normal (7.8,0.64)
a_3	Aircoil leadout trimming	- Pneumatic press 2	W3	5.7	6.5	6.7	5.6	6.2	5.9	Log Normal (6.1,0.44)
a_4	Aircoil leadout stripping	- Stripping machine 1	W4	7.8	8.3	8.1	8.4	8.5	8.7	Normal (8.3,0.32)
a_5	Aircoil leadout side stripping	- Stripping machine 2	W5	7.7	7.6	7.5	8.3	7.3	8.4	Normal (7.8,0.45)
a_6	Leads dip soldering	Solder pot Tweezer - Flux	W6	4.2	4.5	4.8	4.5	4.6	4.4	Normal (4.5,0.14)

Table 5.1 Continued

a_7	Flux cleaning	- Dish washer	W6	0.8	0.8	0.8	0.8	0.8	0.8	Uniform (0.8,0)
a_8	Aircoil leadout forming	- Pneumatic Forming Machine	W7	4.2	4.6	4.4	4.6	4.1	4.5	Normal (4.4,0.21)
a_9	Rod core assembly to aircoil	- Bent tip tweezer - Varnish container	W8	4.4	4.5	4.9	4.8	4.1	4.9	Normal (4.6,0.32)
a_{10}	Rod curing	- Oven - Baking tray	W8	4.0	4.0	4.0	4.0	4.0	4.0	Uniform (4.0,0)
a_{11}	Moulding press	- Double acting compression moulding	W9	8.2	8.0	8.0	8.1	8.0	8.3	Log Normal (8.1,0.13)
a_{12}	Inductor clamping	- Tongs - Clamping machine	W10	3.0	3.2	2.9	3.1	3.4	3.6	Log Normal (3.2,0.26)
a_{13}	Unit curing	- Oven - Baking trolley - PC profiler	W10	9.0	9.0	9.0	9.0	9.0	9.0	Uniform (9.0,0)

Table 5.1 Continued

a_{14}	Unit unclamping from tongs	- Tongs - Clamping machine	W10	1.6	1.8	1.4	1.6	1.5	1.7	Normal (1.6,0.14)
a_{15}	Lead cropping and forming	- Semi-auto cropping and forming machine	W11	4.5	4.8	5.1	4.3	5.1	4.4	Log Normal (4.7,0.35)
a_{16}	Part number marking	- Video jet printer	W12	2.1	2.3	2.2	2.2	2.4	2.0	Normal (2.2,0.14)
a_{17}	IR-reflow	- IR-Reflow machine - Baking tray	W12	2.3	2.3	2.3	2.3	2.3	2.3	Uniform (2.3,0)
a_{18}	VMI + Q-factor test	- Mantis scope - Height gauge - LCR meter	W13	6.5	6.5	6.8	5.4	6.2	7.0	Normal (6.4,0.56)
a_{19}	Packaging	- Tape & reel machine	W13	1.5	1.4	1.5	1.6	1.6	1.5	Normal (1.5,0.08)

Data distribution test was performed using Minitab software to identify the distribution of each of assembly task. Figure 5.3 and Figure 5.4 present the example of data distribution test for task a_1 and task a_2 respectively. The highest P-Value indicates the distribution of data. The distribution of task a_1 was classified as Log Normal distribution since the P-Value is 0.829. Meanwhile, task a_2 is normally distributed.

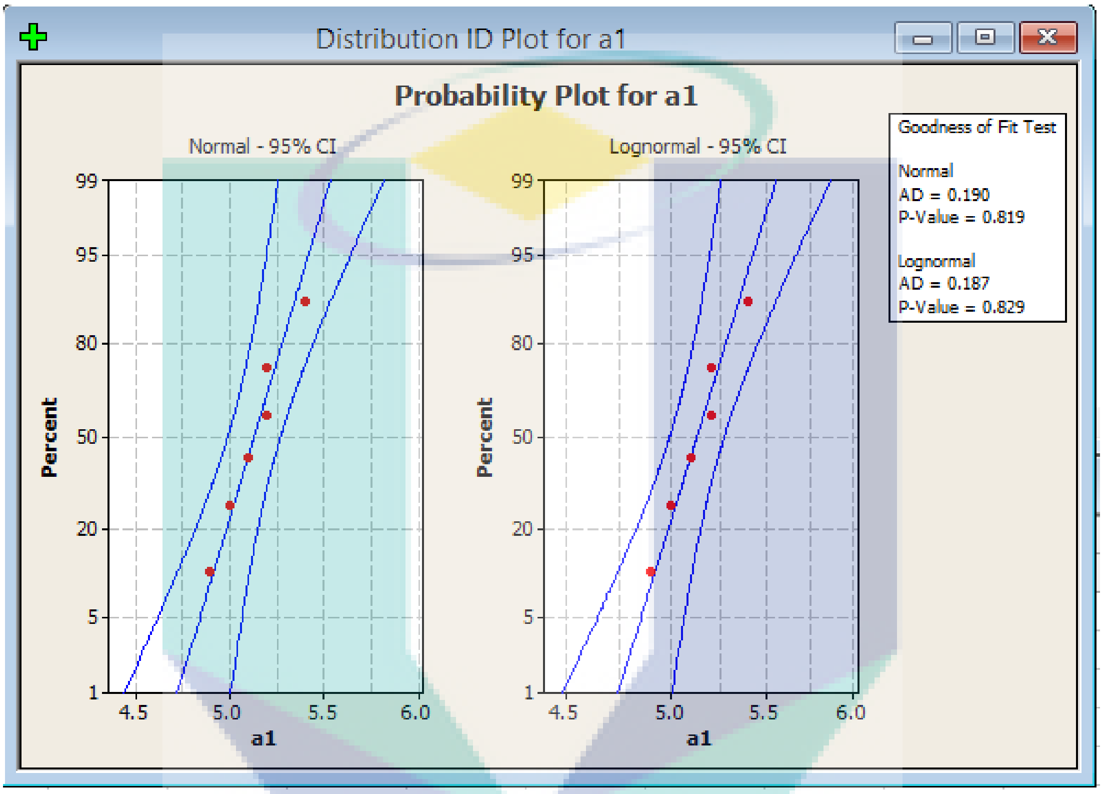


Figure 5.3 Example of data distribution test for a_1 task

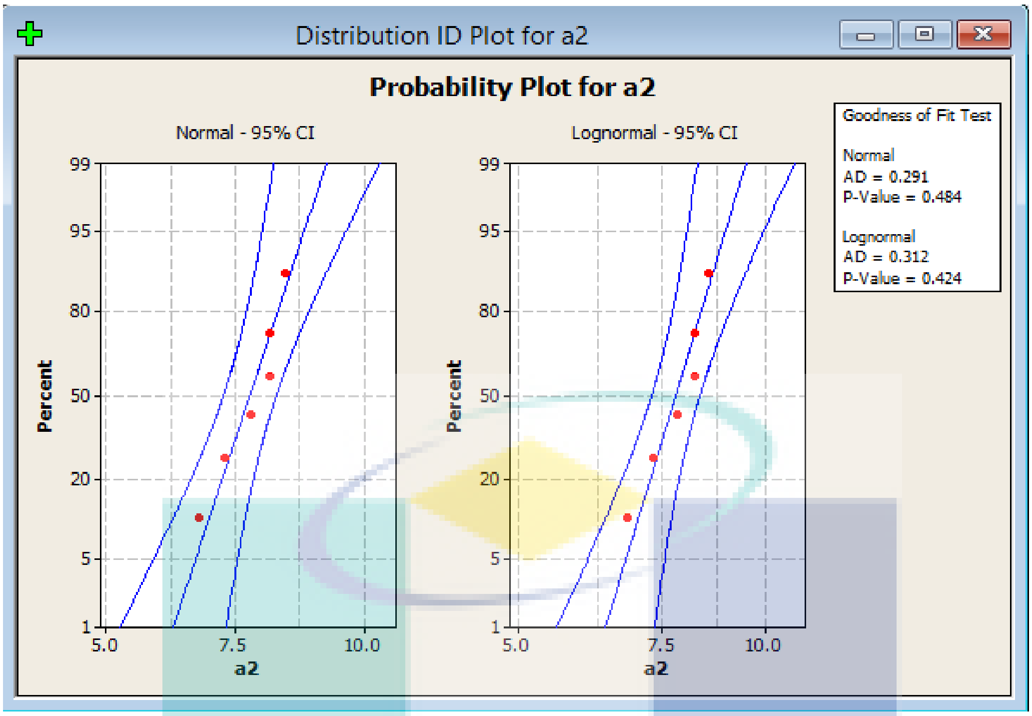


Figure 5.4 Example of data distribution test for a_2 task

5.2.4 Simulation of Current Layout

A simulation of existing layout was conducted using WITNESS™ software to simulate the assembly line. WITNESS™ is a simulation software that is commercially used to provide overall view on all the process in terms of busy, idle, blocked and output. The purpose of conducting existing layout simulation is to validate the simulation model with actual layout. The view of WITNESS™ software can be referred to Appendix A2. Meanwhile, Appendix A3 presents the simulation results of optimised layout from 1st run to 10th run.

Figure 5.5 exhibits the simulation model of current layout for the production of HM72A-10 series. A total of 19 tasks need to be carried out to produce a final product using 17 workstations to perform all 19 tasks. Each task was operated in each workstation except for task a_4 (aircoil leadout stripping) and task a_5 (aircoil leadout side stripping) which were performed in workstation 4 (ST4). Besides that, task a_{18} (VMI) and task a_{19} (Inductor, DCR, Q-factor tests, and packaging) were performed in the same workstation: workstation ST17.

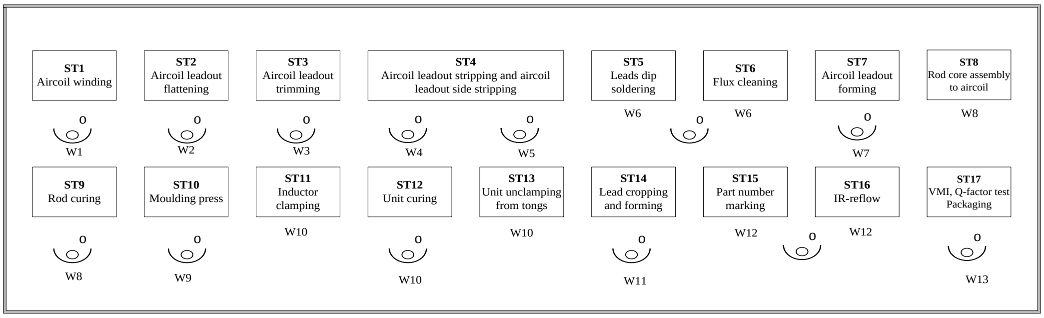


Figure 5.5 Current layout

The summary of the production line of HM72A-10 series model is presented in Table 5.2. A total of 13 workers were assigned to particular workstations to perform the related tasks with a total of 30 machines and tools that had been used throughout the process. The data in Table 5.2 indicated that workstation 4 (ST4) has the longest processing time; which is 16.1 seconds. Meanwhile, the shortest processing time (0.8 seconds) was recorded in workstation 6 (ST6).

Table 5.2 Summary of *HM72A-10* production

Work element		**ST**	**Resources**		**Processing time (s)**
			Machine, Tools	**Worker**	
a_1	Aircoil winding	ST1	- Auto CNC Aircoil Machine	W1	5.1
a_2	Aircoil leadout flatenning	ST2	- Pneumatic press 1	W2	7.8
a_3	Aircoil leadout trimming	ST3	- Pneumatic press 2	W3	6.1
a_4	Aircoil leadout stripping (upper side)	ST4	- Stripping machine 1	W4	8.3
a_5	Aircoil leadout side stripping (lower side)	ST4	- Stripping machine 2	W5	7.8
a_6	Leads dip soldering	ST5	- Solder pot - Tweezer - Flux	W6	4.5
a_7	Flux cleaning	ST6	- Dish washer	W6	0.8

Table 5.2 Continued

a_8	Aircoil leadout forming	ST7	- Pneumatic Forming Machine	W7	4.4
a_9	Rod core assembly to aircoil	ST8	- Bent tip tweezer - Varnish container	W8	4.6
a_{10}	Rod curing	ST9	- Oven - Baking tray	W8	4.0
a_{11}	Moulding press	ST10	- Double acting compression moulding	W9	8.1
a_{12}	Inductor clamping	ST11	- Tongs - Clamping machine	W10	3.2
a_{13}	Unit curing	ST12	- Oven - Baking trolley - PC profiler	W10	9.0
a_{14}	Unit unclamping from tongs	ST13	- Tongs - Clamping machine	W10	1.6
a_{15}	Lead cropping and forming	ST14	- Semi-auto crop & form machine	W11	4.7
a_{16}	Part number marking	ST15	- Video jet printer	W12	2.2
a_{17}	IR-reflow	ST16	- IR-Reflow machine - Baking tray	W12	2.3
a_{18}	VMI, Inductor/DCR + Q-factor	ST17	- Mantis scope - Height Gauge - LCR meter	W13	6.4
a_{19}	Packaging	ST17	- Tape & reel machine	W13	1.5

Indicator: ST = Workstation

The simulation time was set for 22 hours of working time; equivalent to 2 shifts. The objective of the simulation is to measure the changes of the layout based on the optimisation result. The simulation is only focused on normal operation and the following conditions are applied:

1. Simulation is conducted for the period of 22 hours per day, equivalent to 79200 seconds.
2. The warm up period is set for 180 minutes. The warm up period is set up to this point in order to eliminate the initial bias as the production line that started with empty system tend to flow at a faster rate.
3. The simulation is run for 30 times using different pseudo-random seeds so that the output generated from the simulation are non-repeating.
4. Buffer capacity is set to 300, based on the average capacity.
5. The reject rate is assumed to be 2%.
6. The arrival part is unlimited. Meanwhile, the inter-arrival time is set in distribution form.
7. The attendance rate of manpower is assumed to be >80%.

Table 5.3 presents the simulation result of output for existing layout. From the simulation result, the average of the output for the existing layout is 7318 units.

Table 5.3 Simulation result of output for existing layout

Run	Output
	Existing Layout
1	7313
2	7329
3	7319
4	7325
5	7327
6	7313
7	7323
8	7314
9	7317
10	7321
11	7312
12	7321
13	7323

Table 5.3 Continued

14	7322
15	7309
16	7320
17	7310
18	7312
19	7312
20	7325
21	7312
22	7317
23	7318
24	7314
25	7320
26	7311
27	7318
28	7323
29	7318
30	7308
Average	7318

Meanwhile, the actual output in industry is presented in Table 5.4. The output of the existing layout was compared with the actual industrial output using t-test in order to validate the acceptance of simulation model. Result of t-test is shown in Table 5.5.

Table 5.4 Actual industrial output

Actual industrial output
7200
7300
7200
7500
6600

Table 5.4 Continued

7300
7100
7200
7500
7200
6850
7300
7100
7400
7500
7000
7100

A t-test was performed using Minitab to validate the significance of existing layout simulation with the actual layout. The hypothesis for this t-test is as follow:

i. Null hypothesis, H_0 that assumes the mean of the two samples are equal.
ii. Alternative hypothesis, H_1 which assumes that there are difference of the mean between the two samples.

Table 5.5 shows result of t-test between actual industrial output and existing layout. Based on the t-test that was conducted, the calculated t-value is 2.120, while the critical t-value is the same which is 2.120. The critical t-value was obtained from T-table. There are two hypothesis that must be consider:

i. Null hypothesis, H_0 that assumes the mean of the two samples are equal.
ii. Alternative hypothesis, H_1 assumes that there are difference of the mean between the two samples.

After making the hypothesis, the level of confidence will be selected. Usually, the confidence level is 95%. The degree of freedom is obtained by taking the sample size to be reduced by one *(n-1)*. Later, the calculated value will be compared with the value from the T-table. When the calculated t-value is larger than critical t-value, the null hypothesis, H_0 will be rejected. On the other hand, if the calculated t-value is smaller than critical t-value, the null hypothesis is accepted. In this case, the t-test will accept null hypothesis, H_0 which means that there should be no difference between the two samples (i.e. output of existing layout and actual industrial output). Referring to Table 5.5, the t-value obtained from t-test is 2.12 similarly to the critical t-value from T-table.

As there are no difference between the mean of the actual industrial output and the output of the existing layout, the simulation model is deemed accepted to represent the actual assembly layout.

Table 5.5 Result of t-test between actual industrial output and existing layout

T-Test	Output
T-value	2.120
Critical t-value	2.120

Table 5.6 records the percentage of busy as well as percentage of idle for workstation of existing layout, obtained from simulation. It is apparent from this table that the average percentage of busy workstation obtained from the simulation is 45.25%. This percentage is consider as low due to large number of workstation as well as less tasks were performed in one workstation. In the meantime, the average idle workstation records an average of 43.78% idle. This value is consider as too high because of less tasks were performed by the workers.

Table 5.6 Percentage of busy and idle of workstation of existing layout

Run	**Workstation**	
	Busy (%)	**Idle (%)**
1	45.27	43.70
2	45.28	43.78
3	45.27	43.72

Table 5.6 Continued

4	45.27	43.80
5	45.31	43.66
6	45.23	43.85
7	45.24	43.91
8	45.21	43.87
9	45.20	43.94
10	45.27	43.63
11	45.18	43.97
12	45.29	43.56
13	45.26	43.86
14	45.25	43.88
15	45.20	43.99
16	45.28	43.62
17	45.23	43.77
18	45.22	43.84
19	45.23	43.79
20	45.33	43.50
21	45.23	43.72
22	45.19	44.11
23	45.28	43.59
24	45.24	43.84
25	45.25	43.81
26	45.23	43.70
27	45.23	43.85
28	45.28	43.66
29	45.27	43.64
30	45.21	43.72
Average	**45.25**	**43.78**

Table 5.7 shows the percentage of busy and idle of worker of existing layout. Based on Table 5.7, the simulation result for the existing layout reveals that the average of busy percentage of worker is 66.40%, meanwhile the average idle of the worker is 33.23%. The busy percentage of worker is less because most of the worker only perform single task. This situation leads to high idle percentage of the worker. The result of the simulation for the existing layout later will be compared with the simulation result after the optimisation using paired t-test to measure the improvement of the optimisation result in section 5.2.5.

Table 5.7 Percentage of busy and idle of worker of existing layout

Run	Worker	
	Busy (%)	Idle (%)
1	66.79	33.21
2	66.80	33.20
3	66.80	33.20
4	66.79	33.21
5	66.85	33.15
6	66.74	33.26
7	66.76	33.24
8	66.72	33.28
9	66.71	33.29
10	66.79	33.21
11	66.68	33.32
12	66.83	33.17
13	66.79	33.21
14	66.78	33.22
15	66.70	33.30
16	66.82	33.18
17	66.74	33.26
18	66.71	33.29
19	66.74	33.26
20	66.90	33.10

Table 5.7 Continued

21	66.74	33.26
22	66.68	33.32
23	66.79	33.21
24	66.75	33.25
25	66.79	33.21
26	66.75	33.25
27	66.76	33.24
28	55.83	33.17
29	66.79	33.21
30	66.70	33.30
Average	**66.40**	**33.23**

Table 5.8 shows the result of simulation for existing layout. It is apparent from this table that the number of workstation and resources used were the same as the previously mentioned figure, which were 17 and 43 respectively, with cycle time being 16.1 seconds. The percentage of line efficiency was obtained from the Eq. (2.1). The figure in Table 5.8 shows that the line efficiency of the existing layout is 33.8%. The simulation result shows the daily output achieved from the existing layout is 7318 units. Meanwhile, the actual average output is 7200 units per day. The result from t-test as in Table 5.5 provides evidence that the simulation model is acceptable.

Table 5.8 Result of simulation for existing layout

Data	**Existing**
Number of workstation	17
Cycle time (s)	16.1
Number of resources	43
Line efficiency (%)	33.8
Daily output	7318

5.2.5 Optimisation using NSGA-II and Simulation Results

In this research, there are three objective functions that needed to be optimised i.e. to minimise the number of workstations, to minimise the cycle time, and to minimise the number of resources. The optimisation for the case study data was done using NSGA-II approach and later followed by simulation.

The proposed layout from optimisation process is presented in Figure 5.6. The output of the optimisation process revealed that a total of nine workstations were needed to run all processes with one worker assigned at each workstation, except for the first workstation that has two workers. A few tasks that use the same resource were assigned to one workstation; respecting the precedence relationship and ensuring that the total processing time in each workstation does not exceed the cycle time.

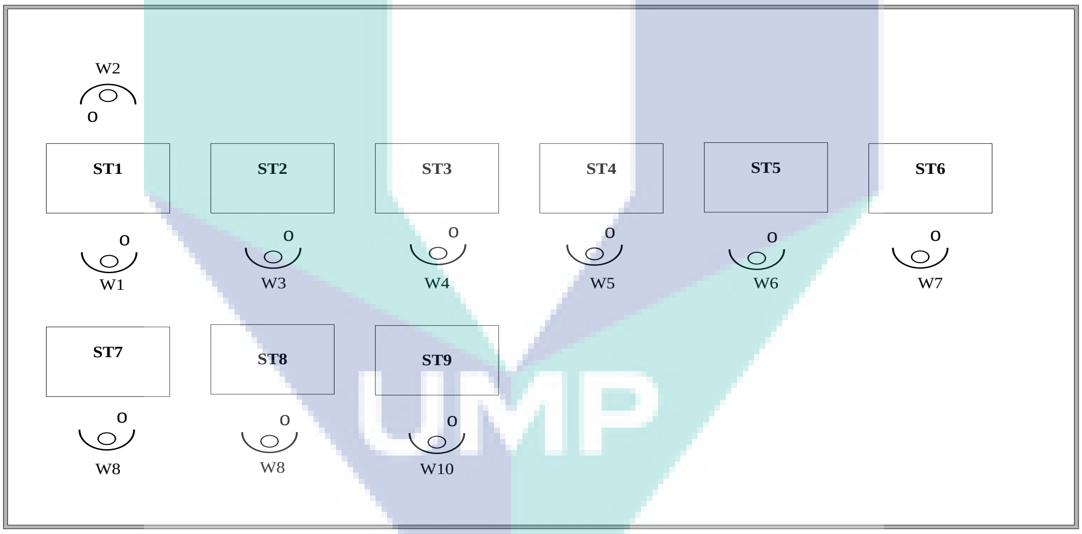


Figure 5.6 Proposed layout

Details on the tasks that have been assigned to each workstation and their respective total processing time are tabulated as in Table 5.9. The information in the table demonstrates that task a_1 and task a_2 were assigned to ST1 with total processing time, 12.9 seconds. Aircoil leadout trimming (a_3) was assigned in the second workstation, ST2 with the total processing time of 6.1 seconds. Meanwhile, aircoil leadout stripping (a_4) were allocated in the third workstation, ST3 with the total of processing time of 8.3 seconds. ST4 consists of three tasks; a_5 (aircoil leadout side stripping), a_6 (leads dip soldering), and a_7 (flux cleaning). The total processing time recorded was 13.1 seconds, which is the highest processing time. There were three tasks assigned to ST5 which were task a_8 (aircoil leadout forming), task a_9 (rod core assembly to aircoil) and task a_{10} (rod curing), with a total processing time of 13.0 seconds.

Two tasks were allocated in ST6: moulded pressing (a_{11}) and inductor clamping (a_{12}). A total of 11.3 seconds of processing time is required to carry out all these tasks. At the same time, task a_{13} (unit curing) and task a_{14} (unit unclamping from tongs) were assigned in ST7 with a total processing time of 10.6 seconds. The other tasks which lead to cropping and forming (a_{15}), part number marking (a_{16}), and IR-reflow (a_{17}) were assigned to workstation 8 (ST8). The total processing time recorded at the workstation is 9.2 seconds. The last workstation, ST9 consisted of two tasks i.e. (i) VMI, inductor/DCR + Q-factor (a_{18}) (ii) Packaging (a_{19}). Both tasks recorded 7.9 seconds of total processing time.

Table 5.9 Proposed task assignment based on NSGA-II optimisation result

ST	Task	Resources	Processing time (s)
ST1	a_1 – aircoil winding a_2 – aircoil leadout flattening	- W1, W2 - Auto CNC aircoil machine - Pneumatic press 1	12.9
ST2	a_3 – aircoil leadout trimming	- W3 - Pneumatic press 2	6.1
ST3	a_4 – aircoil leadout stripping	- W4 - Stripping machine 1	8.3
ST4	a_5 - aircoil leadout side stripping a_6 – leads dip soldering a_7 – flux cleaning	- W5 - Stripping machine 2 - Solder pot - Tweezer - Flux - Dish washer	13.1

Table 5.9 Continued

ST5	a_8 – aircoil leadout forming a_9 – rod core assembly to aircoil a_{10} – rod curing a_{12} – inductor clamping	- W6 - Pneumatic forming machine - Bent tip tweezer - Varnish container - Oven - Baking tray	13.0
ST7	a_{13} – unit curing a_{14} - unit unclamping from tongs	- W8 - Oven - Baking tray - PC profiler	10.6
ST8	a_{15} – lead cropping and forming a_{16} – part number marking a_{17} – IR-reflow	- W9 - Semi auto cropping and forming machine - Video jet printer - IR-reflow machine - Baking tray	9.2
ST9	a_{18} – VMI, inductor/DCR + Q-factor a_{19} – Packaging	- W10 - Mantis scope - Height gauge - LCR meter - Tape and reel machine	7.9

Table 5.10 Comparison of simulation results for the average of output of existing layout and after optimisation

Run	Output	
	Existing Layout	After Optimisation
1	7313	8073
2	7329	8079
3	7319	8072
4	7325	8076
5	7327	8068
6	7313	8079
7	7323	8084
8	7314	8082
9	7317	8076
10	7321	8074
11	7312	8072
12	7321	8078
13	7323	8072
14	7322	8081
15	7309	8079
16	7320	8076
17	7310	8070
18	7312	8080
19	7312	8077
20	7325	8071
21	7312	8079
22	7317	8080
23	7318	8084
24	7314	8079
25	7320	8072
26	7311	8081
27	7318	8074
28	7323	8080

Table 5.10 Continued

29	7318	8078
30	7308	8079
Average	**7318**	**8077**

The results in Table 5.10 demonstrate the comparison on the average of output of the existing layout and the proposed layout based on the optimisation result. It is apparent from the table that there are an average of 8077 units were produced after the optimisation compared to the existing layout that only produced 7318 units. In order to measure the improvement of the output between the existing layout and after the optimisation, a pair t-test was performed.

The pair t-test was conducted to compare the mean of two different samples. In this study, we want to figure out if there exist any improvement on the optimisation based on the case study that was performed in industry. There are two hypothesis that must be considered. First, null hypothesis (H_0) that assumes the mean of the two samples are equal. The second one is alternative hypothesis (H_1) which assumes that there are difference of the mean between the two samples.

The result of the test is presented in Table 5.11. From the results, we can see that the t-value (545.06) is much higher compared with critical t-value (2.000-2.021). The critical t-test value was obtained from T-table, with 53 degree of freedom and 95% confidence interval. Based on the t-value and critical t-value, we will reject the null hypothesis, H_0 since the calculated value is greater than the value from the table. Hence, it can be concluded that there exists significant improvement of the output after the optimisation.

Table 5.11 Result of pair t-test for output between the existing layout and after optimisation

Pair T-Test	Output
	After Optimization and Existing Layout
T-value	545.06
Critical t-value	2.000 – 2.021

Table 5.12 records the percentage of busy and idle of workstation after optimisation. From the table, an average of 61.74% denotes the busy of workstation meanwhile, 29.68% indicated an average of idle workstation. These values are improved compared to the existing condition as the number of workstations have been reduced from 14 workstations to eight (8) workstations after the optimisation. This situation caused the average of busy percentage to be higher than the existing layout. Apart from that, the average percentage of idle workstation also was reduced.

Table 5.12 Percentage of busy and idle of workstation after optimisation

Run	**Workstation**	
	Busy (%)	**Idle (%)**
1	61.31	29.28
2	61.75	29.66
3	61.74	29.73
4	61.76	29.71
5	61.71	29.73
6	61.77	29.68
7	61.79	29.65
8	61.78	29.66
9	61.75	29.65
10	61.76	29.70
11	61.70	29.71
12	61.74	29.68

Table 5.12 Continued

13	61.75	29.69
14	61.76	29.68
15	61.74	29.72
16	61.72	29.72
17	61.72	29.74
18	61.77	29.68
19	61.77	29.67
20	61.75	29.73
21	61.77	29.66
22	61.76	29.71
23	61.80	29.64
24	61.75	29.70
25	61.76	29.71
26	61.76	29.67
27	61.73	29.74
28	61.74	29.68
29	61.75	29.66
30	61.79	29.69
Average	**61.74**	**29.68**

Apart of that, the percentage of busy as well as the percentage of idle for worker are clearly stated in Table 5.13. In different with the existing layout, 69.16% of worker are busy and 35.39% of them are idle. The busy percentage of worker after the optimisation is higher compared to the existing layout. The result of simulation after the optimisation reveals that the idle percentage of the worker are much lower compared to the idle worker of the existing layout. However, the significant of the difference only can be confirmed once the t-test is conducted.

Table 5.13 Percentage of busy and idle of worker after optimisation

Run	Worker	
	Busy (%)	Idle (%)
1	69.17	30.83
2	68.57	31.43
3	69.18	30.82
4	69.20	30.80
5	69.14	30.86
6	69.19	30.82
7	69.22	30.78
8	69.22	30.78
9	69.16	30.84
10	69.20	30.80
11	69.10	30.90
12	69.16	30.84
13	69.16	30.84
14	69.18	30.83
15	69.16	30.84
16	69.16	30.84
17	69.15	30.85
18	69.19	30.81
19	69.20	30.80
20	69.17	30.83
21	69.20	30.80
22	69.19	30.81
23	69.22	30.79
24	69.19	30.81
25	69.18	30.82
26	69.18	30.82
27	69.15	30.85
28	69.17	30.83
29	69.19	30.84

Table 5.13 Continued

30	69.22	30.78
Average	**69.16**	**30.84**

A pair t-test of busy and idle workstation between the existing layout and optimised layout was performed. The results are demonstrated in Table 5.14. It is apparent from this table that the t-value of busy (283.82) surpasses the critical t-value (2.000-2.021). The result of pair t-test also revealed that the t-value of idle workstation are differ compared to the critical t-value. These imply that both parameters (i.e. busy and idle of workstation) have a significance improvement after the optimisation.

Table 5.14 Result of pair t-test of busy and idle workstation between the existing layout and after optimisation

Pair T-Test	**Workstation**	
	After Optimisation and Existing Layout	
	Busy	**Idle**
T-value	283.82	525.6
Critical t-value	2.000 - 2.021	2.000 - 2.021

Table 5.15 illustrates the result of pair t-test of busy and idle worker between the existing layout and after optimisation. The result highlights that the busy of worker has a slight improvement between the existing layout and after the optimisation. The t-value of busy worker (7.62) is slightly higher than the critical t-value (2.000-2.021). Interestingly, the result shows that the idle worker has better improvement between the existing layout and after the optimisation. In comparing the t-value (210.21) and the critical t-value (2.000-2.021), there exists much different between these values.

Table 5.15 Result of pair t-test of busy and idle worker between the existing layout and after optimisation

Pair T-Test	Worker	
	After Optimisation and Existing Layout	
	Busy	Idle
T-value	7.62	210.21
Critical t-value	2.000 - 2.021	2.000 - 2.021

Table 5.16 demonstrates the comparison of simulation results summary between the existing layout and that of after the NSGA-II optimisation. It is noteworthy that the results indicated the number of workstation decreased from 17 to nine (9) after the optimisation. The simulation results also highlighted that the cycle time decreased from 16.1 seconds to 13.1 seconds. Tasks that employ the same resources were assigned to one workstation by ensuring that the cycle time is not exceeded and the precedence constraints are not violated. The results from this table revealed that there were 3 resources less used after the optimisation.

The other important observation to note from this table is the efficiency of the line that increased to 78.4% after the optimisation. As mentioned previously, the value of line efficiency was calculated by using the Eq. (2.1). Therefore, the simulation result indicates that the most efficient line was the one that went through optimisation. Among the simulation results of the existing layout and that of after NSGA-II optimisation, the latter indicates that the daily output obtained from the simulation after the optimisation is 8077 units meanwhile, the existing output is 7318 units.

Table 5.16 Comparison of simulation results between existing layout and after NSGA-II optimisation

Data	Existing	After NSGA-II optimisation
Number of workstation	17	9
Cycle time (s)	16.1	13.1
Number of resource	43	40
Line efficiency (%)	33.8	78.4
Daily output	7318	8077

5.3 Validation

This section describes a validation phase that was performed with an industrial expert from BI Technologies Corporation to validate the applicability of the optimisation process. The validation stage was conducted through an interview and discussion session with their Senior Manufacturing Engineer in BI Technologies, Mr. Rashidi bin Jamaluddin. The results of current layout were compared to the results after the optimisation in order to get feedback from the expert. For validation purpose, some queries were raised during the interview and discussion session; (i) Is the proposed layout possible to implement in the production line? (ii) Does the line effectiveness meet the industrial target?

The findings from the validation suggested that the proposed layout should be modified because some of the equipment's are isolated and cannot be allocated to another place. The modification is also needed due to safety issue. With regards to the optimisation output in Table 5.9, the aircoil winding a_1 was assigned together with aircoil lead out flattening a_2 in ST1. This assignment will cause ear-splitting because the operation of CNC winding machine to conduct task a_1 will produce too much noise. Thus, task a1 and task a2 should be allocated in two different workstations as shown in Table 5.17.

Table 5.17 Task assignment after validation

ST	Task	Resources	Processing time (s)
ST1	a_1 – aircoil winding	- W1 - Auto CNC aircoil machine	5.1
ST2	a_2 – aircoil leadout flattening	- W2 - Pneumatic press 1	7.8
ST3	a_3 – aircoil leadout trimming	-W3 - Pneumatic press 2	6.1
ST4	a_4 – aircoil leadout stripping (upper side)	-W4 - Stripping machine 1	8.3
ST5	a_5 - aircoil leadout side stripping (lower side) a_6 – leads dip soldering a_7 – flux cleaning	- W5 - Stripping machine 2 - Solder pot - Tweezer - Flux - Dish washer	13.1
ST6	a_8 – aircoil leadout forming a_9 – rod core assembly to aircoil a_{10} – rod curing	- W6 - Pneumatic forming machine - Bent tip tweezer - Varnish container - Oven - Baking tray	13.0
ST7	a_{11} – moulding press a_{12} – inductor clamping	-W7 - Double acting compression moulding - Tong - Clamping machine	11.3
ST8	a_{13} – unit curing a_{14} - unit unclamping from tongs	-W8 - Oven - Baking tray - PC profiler	10.6
ST9	a_{15} – lead cropping and forming a_{16} – part number marking a_{17} – IR-reflow	-W9 - Semi auto cropping and forming machine - Video jet printer - IR-reflow machine - Baking tray	9.2

Table 5.17 Continued

ST10	a_{18} – VMI, inductor/DCR + Q-factor a_{19} – Packaging	-W10 - Mantis scope - Height gauge - LCR meter - Tape and reel machine	7.9

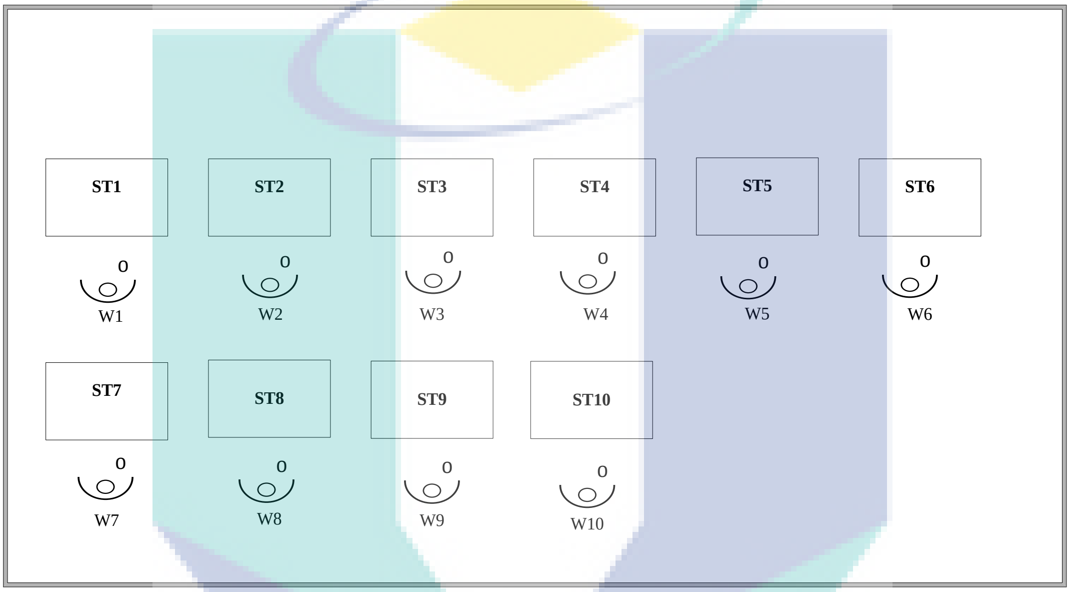


Figure 5.7 Layout after validation

Figure 5.7 exhibits the layout after validation whereas Table 5.17 provides the details on task assignment at each workstation and their respective total processing time. After the validation, some changes were made by taking into account the feedback from industrial expert. Task a_1 (aircoil winding) should be assigned in workstation 1, ST1 with processing time 5.1 seconds meanwhile, task a_2 (aircoil leadout flattening) is assigned in workstation 2, ST2 with the processing time of 7.8 seconds. The rest of the tasks remain unchanged as the optimisation result in Table 5.9.

Table 5.18 presents the comparison of simulation results for output of existing layout and after validation. The tabulated results show that the average number of output obtained after the validation (8056) is slightly higher than the existing layout (7318). On top of that, the output after the validation was compared with the existing layout by perform a pair t-test.

Table 5.18 Comparison of simulation results for output of existing layout and after validation

Run	Output	
	Existing Layout	**After Validation**
1	7313	8053
2	7329	8057
3	7319	8052
4	7325	8054
5	7327	8048
6	7313	8057
7	7323	8063
8	7314	8061
9	7317	8054
10	7321	8053
11	7312	8051
12	7321	8055
13	7323	8051
14	7322	8061
15	7309	8057
16	7320	8054
17	7310	8049
18	7312	8058
19	7312	8054
20	7325	8050
21	7312	8057

22	7317	8057
23	7318	8064
24	7314	8057
25	7320	8050
26	7311	8061
27	7318	8054
28	7323	8058
29	7318	8055
30	7308	8055
Average	**7318**	**8056**

The result of pair t-test for output between the existing layout and after validation is tabulated in Table 5.19. From here we can see that, the t-value (546.65) is excessively higher than critical t-value which is in the range of 2.000-2.021. The output after the validation is greater than the output of existing layout. In this case, the null hypothesis is rejected since there are significant difference of the mean between the output of the existing layout and after the validation.

Table 5.19 Result of pair t-test for output between the existing layout and after validation

Pair T-Test	**Output**
	After Validation and Existing Layout
T-value	546.65
Critical t-value	2.000 - 2.021

The percentage of busy and idle of workstation after the validation are recorded in Table 5.20. From the table, it is clearly stated that the average percentage of busy workstation is 63.73%, slightly higher than the value after the optimisation. In the meantime, the simulation result shows that 28.85% of workstations are idle. Following that, these simulation results are compared with the results of existing layout through a pair t-test.

Table 5.20 Percentage of busy and idle of workstation after validation

Run	Workstation	
	Busy (%)	Idle (%)
1	63.72	28.87
2	63.72	28.87
3	63.72	28.85
4	63.71	28.87
5	63.74	28.85
6	63.76	28.84
7	63.77	28.81
8	63.79	28.81
9	63.7	28.88
10	63.7	28.87
11	63.72	28.85
12	63.74	28.86
13	63.71	28.87
14	63.77	28.81
15	63.72	28.86
16	63.74	28.85
17	63.73	28.87
18	63.71	28.87
19	63.7	28.87
20	63.73	28.84
21	63.75	28.84
22	63.76	28.83
23	63.75	28.83
24	63.74	28.85
25	63.69	28.89
26	63.71	28.86

Table 5.20 Continued

27	63.72	28.86
28	63.73	28.84
29	63.76	28.83
30	63.76	28.84
Average	**63.73**	**28.85**

Table 5.21 presents the result of pair t-test of busy and idle workstation of existing layout and after validation. It appears from the table that the t-value of busy and idle are 317.68 and 559.77 correspondingly. These values are vary compare with the critical t-value which is in the range of 2.000-2.021. Since the t-value is greater than the critical t-value from the table, the null hypothesis (H_0) will be rejected. In this case, we will accept the alternative hypothesis (H_1) as there are difference of the mean of busy and workstation between the existing layout and after validation. In a simple word, it can be conclude that the validation phase out did the existing layout.

Table 5.21 Result of pair t-test of busy and idle workstation between the existing layout and after validation

Pair T-Test	**Workstation**	
	After Validation and Existing Layout	
	Busy	**Idle**
T-value	317.68	559.77
Critical t-value	2.000 - 2.021	2.000 - 2.021

Table 5.22 records the percentage of busy and idle of worker after the validation. The results in the table revealed that an average of 64.71% of workers are busy performing their tasks. Meanwhile, in average, 35.39% of the workers are in idle state. These situations might be caused by some of the workers have less jobs to conduct.

Table 5.22 Percentage of busy and idle of worker after validation

Run	Worker	
	Busy (%)	**Idle (%)**
1	65.24	34.76
2	65.24	34.76
3	58.72	41.28
4	58.72	41.28
5	58.74	41.26
6	65.29	34.71
7	65.29	34.71
8	65.31	34.69
9	65.22	34.78
10	65.22	34.78
11	65.24	34.76
12	65.26	34.74
13	65.24	34.76
14	65.30	34.70
15	65.25	34.75
16	65.26	34.74
17	65.25	34.75
18	65.24	34.76
19	65.23	34.77
20	65.27	34.73
21	65.28	34.72
22	65.29	34.71
23	65.28	34.72

Table 5.22 Continued

24	65.27	34.73
25	65.22	34.78
26	65.24	34.76
27	65.26	34.74
28	65.25	34.75
29	65.29	34.71
30	65.29	34.71
Average	**64.61**	**35.39**

Table 5.23 demonstrates the result of pair t-test of busy and idle worker between the existing layout and after validation. It is apparent from this table that very few different between the t-value of busy (3.37) and the critical t-value which is in the range of 2.000-2.021. It also appear from the table that the t-value of idle which is 5.90 has a small different with the critical t-value (2.000-2.021). Therefore, there is no much significance between the output of the existing layout and after the validation.

Table 5.23 Result of pair t-test of busy and idle worker between the existing layout and after validation

Pair T-Test	**Worker**	
	After Validation and Existing Layout	
	Busy	**Idle**
T-value	3.37	5.90
Critical t-value	2.000 - 2.021	2.000 - 2.021

The comparison of simulation result summary after NSGA-II optimisation and after validation is illustrated as in Table 5.24. Among the feedback from the validation stage, it is suggested that task a_1 (aircoil winding) and task a_2 (aircoil leadout flattening) should be assigned to two different workstations. As a result, the number of workstation after validation increased to 10 workstations from 9 after the optimisation. From Table 5.24, it is apparent that the average percentage of busy in workstation is 70.5%. Last but not least, the simulation result yields an average of 8077 units of daily output. Whereas, the actual average output for this product is 7200 units per day.

Table 5.24 Comparison of results after optimisation and validation

Data	**After NSGA-II optimisation**	**After validation**
Number of workstation	9	10
Cycle time (s)	13.1	13.1
Number of resource	40	40
Line efficiency (%)	78.4	70.5
Daily output	8077	8056

Table 5.25 exhibits the comparison of optimisation parameters for different stages i.e. existing layout, after optimisation as well as after validation. From the results, it is apparent that the cycle time was decreased from 16.1 seconds to 13.1 seconds both after the optimisation and validation. The results in Table 5.25 show that there are 17 workstations used in existing layout. In this research, a few tasks operated by similar resources are assigned to the same workstation given that it does not violate the precedence constraint and cycle time as well. Due to that, the number of workstations had drastically decreased to 9 workstations after the optimisation. However, this figure increased to 10 to include one additional workstation after the validation phase. The number of resources after the optimisation and validation are way different than the number of resources in existing layout.

Table 5.25 Comparison of optimisation parameters for different stages

	Existing Layout	After Optimisation	After Validation
Cycle time	16.1	13.1	13.1
Number of workstation	17	9	10
Number of resources	43	40	40
Assembly line efficiency (%)	33.8	78.4	70.5

The comparison results show that the number of resources used for existing layout is 43 resources. Meanwhile, the number of resources used after both, the optimisation and validation is 40 resources. The most striking observation from the comparative result is that the line efficiency after the optimisation phase (78.4%) surpassed the line efficiency of the existing layout (33.9%) and that of after validation (70.5%) as well. Apart from that, the feedback from the industrial expert also suggested that the optimisation technique is applicable in manufacturing field as it can reduce the utilisation of workstations, resources and time. Last but not least, the aforementioned approach also can enhance the line efficiency.

5.4 Chapter Summary

This chapter aims to provide detailed explanation on the case study that was performed in BI Technologies Sdn. Bhd. including some background of the company and product. Apart from that, the existing production layout is presented in this chapter with a clear description of the process flow. The industrial case study was conducted in order to validate the ALBE-RC model and NSGA-II algorithm. Related assembly data had been collected to be simulated in WITNESS™ Simulation Software.

The results from the statistical test between the actual industrial output and existing layout found that the calculated t-value and the critical t-value from the table are the same which is 2.120. As there are no different between the mean of these two conditions, the null hypothesis is accepted. Thus the simulation model is accepted to represent the actual layout.

The simulation result shows that the average daily output obtained after the optimisation is 8077. Meanwhile, the actual average output is 7200 units per day. The result from the statistical test proves that the H_0 is rejected. There exist major different among the two situations. The validation stage with the industrial expert was performed to validate the applicability of the optimisation process. The result from the statistical test also highlight that there are significant different of the mean between the output of the existing layout and after validation. The optimisation parameters i.e. number of workstations, cycle time and number of resources had recorded the minimum value that could be achieved using the proposed model and algorithm. In the meantime, the ALBE-RC model and the proposed algorithm was validated and can be implemented for industrial application. However, for the selected case study, a minor change in the optimisation result is needed due to isolated equipment.

CHAPTER 6

CONCLUSION AND RECOMMENDATION

6.1 Introduction

This chapter summarises the research works and highlights the contribution of the research to knowledge. Lastly, some recommendations for the future research are suggested.

6.2 Summary of Research

This section summarises the work that had been done throughout the research. The first phase in this research is conductance of literature review. The literature review presented the classification of assembly line balancing (ALB) which is divided to two categories (i) Simple Assembly Line Balancing Problem, SALBP (ii) General Assembly Line Balancing Problem, GALBP (Scholl and Becker, 2006, Boysen et al., 2007). Meanwhile, SALBP is classified into four groups i.e. ALB-1, ALB-2, ALB-E and ALB-F. However, very few studies were carried out on ALB-E.

In literature review, it is common to see previous researchers considering cycle time, workstation, precedence, zoning and other constraints in their works. The researches focuses on those constraints rather than resource constraint in ALB-E itself. Thus, this research is focused on ALB-E by considering the resource constraint (ALE-RC). The literature review reveals that most researchers used GA-based approach as the optimisation technique in ALB (Al-Hawari et al., 2014, Ranjan and Pawar, 2014, Zacharia and Nearchou, 2013, Mohd Razali and Geraghty, 2011). However, no reports so far were found using the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) for the optimisation of ALB-E itself.

This research had established a methodology to represent ALBE-RC. The optimisation of ALBE-RC problem is demonstrated by a few steps. Primarily, a liaison matrix is established to generate feasible assembly sequence and followed by the application of DeFazio's question and answer to identify the existence of precedence diagram mapping. Later, the matrix data is tabulated into respective table. Last but not least, the assembly sequence is evaluated according to three objective functions (i) to minimise the number of workstation (ii) to minimise the cycle time (iii) to minimise the number of resource.

The research work is continued with algorithm development to optimise the ALBE-RC. In this stage, NSGA-II is developed for the optimisation purpose and has been coded into a computer program, MATLAB. Then, the algorithm is verified to ensure the program provide the required output.

A computational test is conducted to test the performance of the algorithm. The performance of the proposed algorithm is compared to the other two comparative algorithms within Genetic Algorithm family; Multi-Objective Genetic Algorithm (MOGA) and Hybrid Genetic Algorithm (HGA). Out of the five performance indicators that had been used, the proposed algorithm, NSGA-II consistently performed in three indicators i.e. Number of Non-Dominated Solutions (*NDS*), Error Ratio (*E*R), and Generational Distance (*GD*).

Apart from that, an industrial case study is conducted in BI Technologies Sdn. Bhd. to validate the applicability of optimised algorithm as well as the mathematical model. The related assembly data had been collected to be simulated in WITNESS™ Simulation Software. The existing layout simulation is used to validate the simulation model with actual layout. The feedback from the industrial expert indicated that the output from the optimised algorithm cannot be fully implemented due to the fact that their equipment is isolated and cannot be moved to another place. Nevertheless, the industrial expert concluded that the proposed algorithm and model is applicable and can be implemented for industrial application as it can minimise the usage of resources and the number of workstation in production line. In fact, it also can enhance the line efficiency by minimising production time.

6.3 Conclusions

This section concludes the research objectives that lead to research contribution. Generally, this research reported on Assembly Line Balancing of Type-E problem focusing on single model. Specifically, this research consists of three objectives, upon achieving them, will result in research contribution.

The first objective is to study ALB-E problem and to establish a mathematical model for ALB-E problem with resource constraint. This objective is achieved by focusing on the Assembly Line Balancing Type-E problem to develop a mathematical model based on some constraints. In this work, the three objective functions need to be optimised. The first objective function is to minimise the cycle time. The next is to minimise the number of workstation and the last one is to minimise the number of resources used. These objective functions are subjected to the constraints detailed in Chapter 3.

The second objective; to optimise the ALB-E problem with resource constraint using Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II), is accomplished by developing an algorithm to optimise the problem. A computational test had been set to test the performance of NSGA-II. Six problems of different sizes taken from open literature were used to test the algorithm. The NSGA-II was compared with different versions of GA-based algorithms; MOGA and HGA. The computational test showed that NSGA-II performed well in finding the non-dominated solutions and has the ability to explore the search space. It also has higher accuracy of solution as discussed in Chapter 4.

The third objective; which is to validate the mathematical model and optimise algorithm through an industrial case study were achieved by collecting related data from industry to be optimised and simulated using software as reported in Chapter 5.

An interview and discussion session had been conducted with the industrial expert to validate the simulation model with the actual layout. The feedback from the expert suggested that the proposed algorithm and the ALBE-RC model are suitable for industrial application.

In summary, a methodology and algorithm for the optimisation of ALBE-RC problem using an Elitist Non-Dominated Sorting Genetic Algorithm had been successfully demonstrated in this research. The main contribution of this research are the establishment of ABLE-RC problem modelling and the implementation of NSGA-II for ABLE-RC which lead to the improvement of efficiency in assembly process. The accomplishment of the aforementioned objectives concluded that this research is capable to improve the industrial productivity by proposing an efficient way to assemble product. By accomplishing the research aim, it is proven that the proposed algorithm, NSGA-II is a good method to optimise the ALBE-RC problem. As discussed previously, this algorithm has dominated the other comparative algorithms such as MOGA and HGA in terms of the ability to explore the search space and also to obtain better accuracy of solutions.

6.4 Limitations and Recommendations

This part highlights the limitations that had been examined throughout the research. The NSGA-II only showed the best performance in three indicators (i.e. Non-Dominated Solutions, Error Ratio and Generational Distance) out of five indicators that were used while testing the algorithm. The NSGA-II algorithm demonstrated a restricted performance in achieving a uniformed solutions as well as better spread of solutions.

The other limitation in this research is the generic problems taken from literature which are randomly modified to suit the studied problem because the ALBE-RC problem had not been given great attention by researchers in the past. Apart from that, the industrial case study had only focused on and conducted in one company.

Based on the limitations of the research, some recommendations are proposed for future research. Primarily, a modification to improve NSGA-II to have better uniformity of solution and to obtain better solution spread is suggested. For validation stage, the industrial case study is also proposed to be conducted in a few companies to further validate the proposed method and algorithm.

REFERENCES

Ağpak, K., & Gökçen, H. (2005). Assembly line balancing: Two resource constrained cases. *International Journal of Production Economics*, *96*(1), 129–140.

Al-Hawari, T., Ali, M., Al-Araidah, O., & Mumani, A. (2014). Development of a genetic algorithm for multi-objective assembly line balancing using multiple assignment approach. *The International Journal of Advanced Manufacturing Technology*, *77*(5-8), 1419–1432.

Bandyopadhyay, S., & Bhattacharya, R. (2014). Solving a tri-objective supply chain problem with modified NSGA-II algorithm. *Journal of Manufacturing Systems*, *33*(1), 41–50.

Barbosa, M. Á., De Lima Neto, F. B., & Marwala, T. (2012). Optmizing Risk Management using NSGA-II. *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, 10–15.

Baril, C., Gascon, V., Miller, J., & Côté, N. (2016). Use of a discrete-event simulation in a Kaizen event: A case study in healthcare. *European Journal of Operational Research*, *249*(1), 327–339.

Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solutionapproaches. *International Journal of Production Economics*, *142*(2), 259–277.

Bautista, J., & Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, *177*(3), 2016–2032.

Baykasoğlu, A., & Akyol, Ş. D. (2012). Meta-heuristic Approaches for the Assembly Line Balancing Problem. *Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management*, 426–434.

Baykasoğlu, A., & Özbakır, L. (2015). Discovering task assignment rules for assembly line balancing via genetic programming. *The International Journal of Advanced Manufacturing Technology*, *76*(1-4), 417–434.

Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, *168*(3), 694–715.

Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, *183*(2), 674–693.

Browning, T. R., & Yassine, A. a. (2010). A random generator of resource-constrained multi-project network problems. *Journal of Scheduling*, *13*(2), 143–161.

Cakir, B., Altiparmak, F., & Dengiz, B. (2011). Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm. *Computers & Industrial Engineering*, *60*(3), 376–384.

Chen, G., Zhou, J., Cai, W., Lai, X., Lin, Z., & Menassa, R. (2006). A framework for an automotive body assembly process design system, *38*, 531–539.

Chen, J. C., Hsaio, M. H., Chen, C. C., & Sun, C. J. (2009). A grouping genetic algorithm for the assembly line balancing problem of sewing lines in garment industry. *Proceedings of the 2009 International Conference on Machine Learning and Cybernetics*, *5*(July), 2811–2816.

Chen, R.-S., Lu, K.-Y., & Yu, S.-C. (2002). A hybrid genetic algorithm approach on multi-objective of assembly planning problem. *Engineering Applications of Artificial Intelligence*, *15*(5), 447–457.

Cheshmehgaz, H. R., Desa, M. I., & Kazemipour, F. (2012). A cellular-rearranging of population in genetic algorithms to solve assembly-line balancing problem. *Journal of Mechanical Engineering and Automation*, *2*(2), 25–35.

Chica, M., Cordón, Ó., & Damas, S. (2011). An advanced multiobjective genetic algorithm design for the time and space assembly line balancing problem. *Computers & Industrial Engineering*, *61*(1), 103–117. doi:10.1016/j.cie.2011.03.001

Chica, M., Cordón, O., Damas, S., & Bautista, J. (2011). Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different Nissan scenarios. *Expert Systems with Applications*, *38*(1), 709–720.

Chica, M., Cordón, Ó., Damas, S., & Bautista, J. (2010). Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Information Sciences*, *180*(18), 3465–3487.

Chica, M., Cordón, Ó., Damas, S., & Bautista, J. (2012). Multiobjective memetic algorithms for time and space assembly line balancing. *Engineering Applications of Artificial Intelligence*, *25*(2), 254–273.

Chong, K. E., Omar, M. K., & Bakar, N. A. (2008). Solving Assembly Line Balancing Problem using Genetic Algorithm with Heuristics- Treated Initial Population, *II*, 1–5.

Corominas, A., Ferrer, L., & Pastor, R. (2011). Assembly line balancing: general resource-constrained case. *International Journal of Production Research*, *49*(12), 3527–3542.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. (R. Sheldon, Ed.) (1st ed.). New York: John Wiley & Sons, Ltd.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization : NSGA-II. *International Conference on Parallel Problem Solving From Nature*, 849–858.

Deb, K., Member, A., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm :, *6*(2), 182–197.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, *6*(2), 182–197.

Emeke Great, O., & Offiong, A. (2013). Productivity Improvement in Breweries Through Line Balancing using Heuristic Method. *International Journal of Engineering Science & Technology*, *5*(3), 475–486.

Esmaeilbeigi, R., Naderi, B., & Charkhgard, P. (2015). The type E simple assembly line balancing problem: A mixed integer linear programming formulation. *Computers & Operations Research*, *64*, 168–177.

Fattahi, P., Roshani, A., & Roshani, A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, *53*(1-4), 363–378.

Fazio, T. De, & Whitney, D. (1987). Simplified generation of all mechanical assembly sequences. *IEEE Journal on Robotics and Automation*, *3*, 640–658.

Fettaka, S., Thibault, J., & Gupta, Y. (2015). A new algorithm using front prediction and NSGA-II for solving two and three-objective optimization problems. *Optimization and Engineering*, *16*(4), 713–736.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization : Formulation , Discussion and Generalization. *ICGA*, *93*(July), 416–423.

Fonseca, C. M., & Fleming, P. J. (1993). Multiobjective genetic algorithms. *Genetic Algorithms for Control Systems Engineering, IEE Colloquium on*, 6–1.

Gen, M., & Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*. John Wiley & Sons, Inc.

Gonçalves, J. F., & De Almeida, J. R. (2002). A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics*, *8*(6), 629–642.

Grzechca, W. (2011). Cycle Time in Assembly Line Balancing Problem, (1), 4–7.

Grzechca, W., & Foulds, L. R. (2015). The Assembly Line Balancing Problem with Task Splitting: A Case Study. *IFAC-PapersOnLine*, *48*(3), 2002–2008.

Gu, L., Hennequin, S., Sava, A., & Xie, X. (2007). Assembly line balancing problems solved by estimation of distribution. In *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on* (pp. 123–127). IEEE.

Guo, W., Hua, L., & Mao, H. (2014). Minimization of sink mark depth in injection-molded thermoplastic through design of experiments and genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *72*(1-4), 365–375.

Gurevsky, E., Battaïa, O., & Dolgui, A. (2012). Balancing of simple assembly lines under variations of task processing times. *Annals of Operations Research*, *201*(1), 265–286.

Gurevsky, E., Battaïa, O., & Dolgui, A. (2013). Stability measure for a generalized assembly line balancing problem. *Discrete Applied Mathematics*, *161*(3), 377–394.

Hamta, N., Fatemi Ghomi, S. M. T., Jolai, F., & Akbarpour Shirazi, M. (2013). A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect. *International Journal of Production Economics*, *141*(1), 99–111.

Hazır, Ö., & Dolgui, A. (2013). Assembly line balancing under uncertainty: Robust optimization models and exact solution method. *Computers & Industrial Engineering*, *65*(2), 261–267.

Kao, H.-H., Yeh, D.-H., Wang, Y.-H., & Hung, J.-C. (2010). An optimal algorithm for type-I assembly line balancing problem with resource constraint. *African Journal of Business Management*, *4*(10), 2051–2058.

Kilincci, O. (2010). A Petri net-based heuristic for simple assembly line balancing problem of type 2. *International Journal of Advanced Manufacturing Technology*, *46*(1-4), 329–338.

Mohd Razali, N., & Geraghty, J. (2011). Biologically inspired genetic algorithm to minimize idle time of the assembly line balancing. *2011 Third World Congress on Nature and Biologically Inspired Computing*, 105–110.

Moon, C., Kim, J., Choi, G., & Seo, Y. (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research*, *140*(3), 606–617.

Morrison, D. R., Sewell, E. C., & Jacobson, S. H. (2014). An application of the branch , bound , and remember algorithm to a new simple assembly line balancing dataset. *European Journal of Operational Research*, *236*(2), 403–409.

Mozdgir, A., Mahdavi, I., Badeleh, I. S., & Solimanpur, M. (2013). Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing. *Mathematical and Computer Modelling*, *57*(1-2), 137–151.

Mutlu, Ö., & Özgörmüş, E. (2012). A fuzzy assembly line balancing problem with physical workload constraints. *International Journal of Production Research*, *50*(18), 5281–5291.

Nearchou, A. C. (2011). Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization. *International Journal of Production Economics*, *129*(2), 242–250.

Nourmohammadi, a., & Zandieh, M. (2011). Assembly line balancing by a new multi-objective differential evolution algorithm based on TOPSIS. *International Journal of Production Research*, *49*(10), 2833–2855.

Otto, A., Otto, C., Scholl, A., & Jena, F. (2011). SALBPGen – A systematic data generator for (simple) assembly line balancing. *Friedrich-Schiller-University Jena, School of Economics and Business Administration*, *5*.

Ozbakir, L., Baykasoglu, A., Gorkemli, B., & Gorkemli, L. (2011). Multiple-colony ant algorithm for parallel assembly line balancing problem. *Applied Soft Computing*, *11*(3), 3186–3198.

Özcan, U. (2010). Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research*, *205*(1), 81–97.

Özcan, U., & Toklu, B. (2009). A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems. *Journal of Intelligent Manufacturing*, *20*(1), 123–136.

Ponnambalam, S. G., Aravindan, P., & Mogileeswar Naidu, G. (2000). A Multi-Objective Genetic Algorithm for Solving Assembly Line Balancing Problem. *The International Journal of Advanced Manufacturing Technology*, *16*(5), 341–352.

Rajabalipour Cheshmehgaz, H., Ishak Desa, M., & Kazemipour, F. (2012). A Cellular-rearranging of Population in Genetic Algorithms to Solve Assembly-Line Balancing Problem. *Journal of Mechanical Engineering and Automation*, *2*(2), 25–35.

Ranjan, R., & Pawar, P. J. (2014). Assembly Line Balancing Using Real Coded Genetic Algorithm. *International Journal of Scientific Research in Computer Science and Engineering*, *2*(04), 1–5.

Rashid, M. F. F., Hutabarat, W., & Tiwari, A. (2012). A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. *The International Journal of Advanced Manufacturing Technology*, *59*(1-4), 335–349.

Roshani, A., Fattahi, P., Roshani, A., Salehi, M., & Roshani, A. (2012). Cost-oriented two-sided assembly line balancing problem: A simulated annealing approach. *International Journal of Computer Integrated Manufacturing*, *25*(8), 689–715.

Sabuncuoglu, I., Erel, E., & Tanyer, M. (2000). Assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing*, *11*(3), 295–310.

Saif, U., Guan, Z., Liu, W., Zhang, C., & Wang, B. (2014). Pareto based artificial bee colony algorithm for multi objective single model assembly line balancing with uncertain task times. *Computers & Industrial Engineering*, *76*, 1–15.

Saravanan, R. (2006). *Manufacturing Optimization through Intelligent Techniques*. Taylor & Francis Group.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international Conference on Genetic Algorithms* (pp. 93–100).

Scholl, A. (1993). Data of assembly line balancing problems. Retrieved from www.assembly-line-balancing.de

Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, *168*(3), 666–693.

Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to genetic algorithms* (1st ed.). New York: Springer Science & Business Media.

Sivasankaran, P., & Shahabudeen, P. (2013). Modelling hybrid single model assembly line balancing problem. *Udyog Pragati*, *37*, 26–36.

Su, P., Wu, N., & Yu, Z. (2014). A Petri net-based heuristic for mixed-model assembly line balancing problem of Type-E. *International Journal of Production Research*, *52*(5), 1542–1556.

Sungur, B., & Yavuz, Y. (2015). Assembly line balancing with hierarchical worker assignment. *Journal of Manufacturing Systems*, *37*, 290–298.

Suwannarongsri, S., Limnararat, S., & Puangdownreong, D. (2007). A new hybrid intelligent method for assembly line balancing. *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, 1115–1119.

Suwannarongsri, S., & Puangdownreong, D. (2008). Multi-Objective Assembly Line Balancing via Adaptive Tabu Search Method, *1*(3), 312–316.

Tapkan, P., Ozbakir, L., & Baykasoglu, A. (2012). Modeling and solving constrained two-sided assembly line balancing problem via bee algorithms. *Applied Soft Computing Journal*, *12*(11), 3343–3355.

Tasan, S. O., & Tunali, S. (2008). A review of the current applications of genetic algorithms in assembly line balancing. *Journal of Intelligent Manufacturing*, *19*(1), 49–69.

Taylor, P. (2008). International Journal of Production Multi-objective balancing of assembly lines by population heuristics, (December 2014), 37–41.

Triki, H., Mellouli, A., Hachicha, W., & Masmoudi, F. (2014). Assembly Line Balancing in an Automotive Cables Manufacturer Using a Genetic Algorithm Approach, (c), 297–302.

Triki, H., Mellouli, A., & Masmoudi, F. (2014). A multi-objective genetic algorithm for assembly line resource assignment and balancing problem of type 2 (ALRABP-2). *Journal of Intelligent Manufacturing*, 1–15.

Tuncel, G., & Topaloglu, S. (2013). Assembly line balancing with positional constraints, task assignment restrictions and station paralleling: A case in an electronics company. *Computers & Industrial Engineering*, *64*(2), 602–609.

Valls, V., Ballestin, F., & Quintanilla, S. (2008). A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, *185*(2), 495–508.

Wang, H. S., Che, Z. H., & Chiang, C. J. (2012). A hybrid genetic algorithm for multi-objective product plan selection problem with ASP and ALB. *Expert Systems with Applications*, *39*(5), 5440–5450.

Wei, H., Tang, X.-S., & Liu, H. (2015). A genetic algorithm (GA)-based method for the combinatorial optimization in contour formation. *Applied Intelligence*, *43*(1), 112–131.

Wei, N.-C., & Chao, I. M. (2011). A solution procedure for type E simple assembly line balancing problem. *Computers & Industrial Engineering*, *61*(3), 824–830.

Weigert, G., Henlich, T., & Klemmt, a. (2011). Modelling and optimisation of assembly processes. *International Journal of Production Research*, *49*(14), 4317–4333.

Yang, C., Gao, J., & Sun, L. (2013). Computers & Industrial Engineering A multi-objective genetic algorithm for mixed-model assembly line rebalancing. *Computers & Industrial Engineering*, *65*(1), 109–116. doi:10.1016/j.cie.2011.11.033

Yoosefelahi, A., Aminnayeri, M., Mosadegh, H., & Ardakani, H. D. (2012). Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model. *Journal of Manufacturing Systems*, *31*(2), 139–151.

Yu, J., & Yin, Y. (2010). Assembly line balancing based on an adaptive genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *48*(1-4), 347–354.

Zacharia, P. T., & Nearchou, A. C. (2012). Multi-objective fuzzy assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing*, *23*(3), 615–627.

Zacharia, P. T., & Nearchou, A. C. (2013). A meta-heuristic algorithm for the fuzzy assembly line balancing type-E problem. *Computers & Operations Research*, *40*(12), 3033–3044.

Zacharia, P. T., & Nearchou, A. C. (2016). A population-based algorithm for the bi-objective assembly line worker assignment and balancing problem. *Engineering Applications of Artificial Intelligence*, *49*, 1–9.

Zhang, R., Chen, D., Wang, Y., Yang, Z., & Wang, X. (2007). Study on Line Balancing Problem Based on Improved Genetic Algorithms. *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, (1), 2033–2036.

Zhang, W., Gen, M., & Lin, L. (2008). A multiobjective genetic algorithm for Assembly Line Balancing problem with worker allocation. *2008 IEEE International Conference on Systems, Man and Cybernetics*, 3026–3033.

Zhao, J., Cheng, G., Ruan, S., & Li, Z. (2015). Multi-objective optimization design of injection molding process parameters based on the improved efficient global optimization algorithm and non-dominated sorting-based genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *78*(9-12), 1813–1826.

Zinflou, A., Gagné, C., Gravel, M., & Price, W. L. (2008). Pareto memetic algorithm for multiple objective optimization with an industrial application. *Journal of Heuristics*, *14*(4), 313–333.

APPENDIX A1

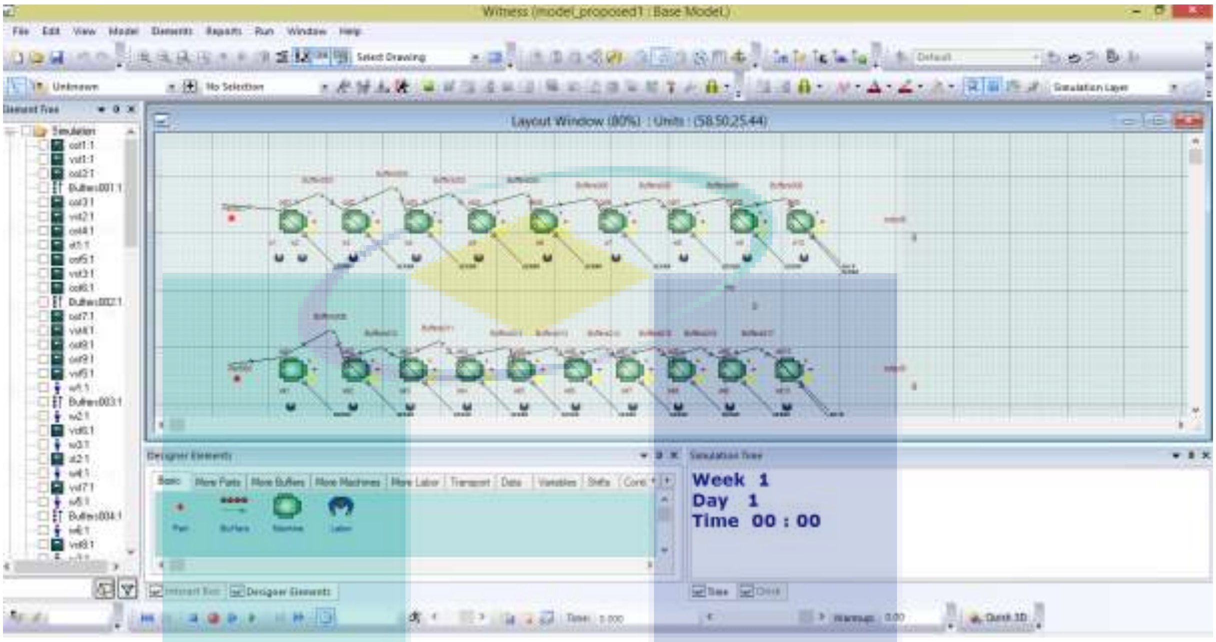
Main Program

```
function y = fitness_func2(x,n,ct_ref)
%n = 30; %size of problem
seq=x;
Tuncel_Topaloglu_4

r=zeros(5,n);
 j=1;
   for i=1:n
      r(1,i)= DM(seq(j,i),1);
      r(2,i)= DM(seq(j,i),2);
      r(3,i)= DM(seq(j,i),3);
      r(4,i)= DM(seq(j,i),4);
      r(5,i)= DM(seq(j,i),5);

   end

   ptime=0;
   w_s=1;
   for i=1:n
      if ptime+ r(1,i)<= ct_ref
         ptime=ptime+r(1,i);
         ws_assign(1,i) = w_s;
         ws_assign(2,i) = ptime;
         %ws_assign(3,i) =

      else
         w_s=w_s+1;
         ptime=r(1,i);
         ws_assign(1,i) = w_s;
```

```
        ws_assign(2,i) = ptime;
    end

end
ws_assign;
rs=[r;ws_assign];
res=zeros(ws_assign(1,n),n);
ws=1; rc=0;

for i=1:n
    if rs(6,i)==ws
        rc=rc+1;
        res(ws,rc)=rs(2,i);
        rc=rc+1;
        res(ws,rc)=rs(3,i);
         rc=rc+1;
        res(ws,rc)=rs(4,i);
         rc=rc+1;
        res(ws,rc)=rs(5,i);


    else
        ws=ws+1;
        rc=1;
        res(ws,rc)=rs(2,i);
        rc=rc+1;
        res(ws,rc)=rs(3,i);
```

APPENDIX A2

View of WITNESS™ software



APPENDIX A3

Name	% Idle	% Busy	% Filling	% Emptying	% Blocked	% Cycle Wait Labor	% Setup	% Setup Wait Labor	% Broken Down	% Repair Wait Labor	No. Of Operations
st1	6.75	6.84	0	0	86.4	0	0	0	0	0	10645
st2	0	100	0	0	0	0	0	0	0	0	10140
st3	23.45	76.55	0	0	0	0	0	0	0	0	9933
st4	1.44	98.56	0	0	0	0	0	0	0	0	9705
st5	45.91	54.09	0	0	0	0	0	0	0	0	9520
st6	95.3	4.7	0	0	0	0	0	0	0	0	9323
st7	49.21	50.79	0	0	0	0	0	0	0	0	9149
st8	46.33	52.08	0	0	0	1.59	0	0	0	0	8958
st9	77.87	22.13	0	0	0	0	0	0	0	0	8774
st10	12	88	0	0	0	0	0	0	0	0	8605
st11	10.32	34.01	0	0	0	55.67	0	0	0	0	8426
st12	38.13	47.23	0	0	0	14.64	0	0	0	0	8245
st13	53.75	16.32	0	0	0	29.93	0	0	0	0	8080
st14	52.99	47.01	0	0	0	0	0	0	0	0	7925
st15	78.46	21.54	0	0	0	0	0	0	0	0	7764
st16	89.04	10.96	0	0	0	0	0	0	0	0	7611
st17	62.32	37.68	0	0	0	0	0	0	0	0	7452
Average	43.72176	45.20529									

Name	% Busy	% Idle	Quantity	No. Of Jobs	No. Of Jobs	No. Of Jobs	No. Of Jobs	Avg Job Time

				Started	Ended	Now	Pre-empted	
worker1	6.84	93.16	1	10646	10645	1	0	0.51
worker2	100	0	1	10141	10140	1	0	7.81
worker3	76.55	23.45	1	9934	9933	1	0	6.1
worker4	98.56	1.44	1	9706	9705	1	0	8.04
worker5	98.56	1.44	1	9706	9705	1	0	8.04
worker6	58.78	41.22	1	18843	18843	0	0	2.47
worker7	50.79	49.21	1	9150	9149	1	0	4.4
worker8	74.21	25.79	1	17733	17732	1	0	3.31
worker9	88	12	1	8606	8605	1	0	8.1
worker10	97.56	2.44	1	24752	24751	1	0	3.12
worker11	47.01	52.99	1	7925	7925	0	0	4.7
worker12	32.51	67.49	1	15375	15375	0	0	1.67
worker13	37.68	62.32	1	7453	7452	1	0	4
Average	66.69615	33.30385						

Name	No. Entered	No. Shipped	No. Scrapped	No. Assembled	No. Rejected	W.I.P.	Avg W.I.P.	Avg Time
Part001	10536	8079	1589	0	68665	568	203.28	4534.9

APPENDIX A4

TT Electronics

BI Technologies Corporation Sdn. Bhd.
(21949-T)
A-1445, Jalan Tanjung Api,
25050 Kuantan,
Pahang Darul Makmur,
Malaysia.

T: +60 (9) 565 8888
F: +60 (9) 514 3555

BI/HRD/16/0198
26 April 2016

To whom it may concern,

Sir/ Madam

VALIDATION OF ALBE-RC AND NSGA-II IN BI TECHNOLOGIES

Referring to the above matter, we are pleased to confirm that the following researcher has conducted her case study in our company, BI Technologies Corporation Sdn Bhd.

Student's Name	: **Masitah Binti Jusop**
NRIC No	: **910421-11-5176**
Level of Study	: **Master in Engineering (Mechanical)**
Faculty/Institution	: **Faculty of Mechanical Engineering, University Malaysia Pahang**
Research Title	: **Multi-objective Optimisation of Assembly Line Balancing Type-E Problem with Resource Constraints**

2. The applicability of the proposed Assembly Line Balancing Type-E problem with resource constraints, *ALBE-RC* model and Elitist Non-Dominated Sorting Genetic Algorithm, *NSGA-II* method have been validated by our Sr. Manufacturing Engineer, Mr. Rashidi Jamaluddin. It can be concluded that the aforementioned approach is applicable for industrial application.

Thank you.

Best Regards,
BI Technologies Corporation Sdn Bhd

Mohd Podzi Mahmud
Operations Director, Magnetics

Figure A1. Validation letter for proposed method and model

APPENDIX A5

LIST OF PUBLICATIONS

Published Conference Paper

1. Jusop, M., & Ab Rashid, M. F. F. (2015). **A review on simple assembly line balancing type-e problem**. In *IOP Conference Series: Materials Science and Engineering* (Vol. 100, No. 1, p. 012005). IOP Publishing.
2. Jusop, M., & Ab Rashid, M. F. F. (2016). **Optimisation of Assembly Line Balancing Type-E with Resource Constraints Using NSGA-II**. In *Key Engineering Materials* (Vol. 701, pp. 195-199). Trans Tech Publications.