Multi-Objective Genetic Algorithm for an Integrated Inspection Allocation and Flow Shop Scheduling with Sequence Dependent Setup Time

by

Ammar Sulaimani

A Thesis
presented to
The University of Guelph

In partial fulfilment of requirements
for the degree of
Master of Applied Science
in
Engineering

Guelph, Ontario, Canada

ABSTRACT

MULTI-OBJECTIVE GENETIC ALGORITHM FOR AN INTEGRATED INSPECTION ALLOCATION AND FLOW SHOP SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME

Ammar Sulaimani
University of Guelph, 2022

Advisor:
Professor F.M. Defersha

One of the most critical elements in manufacturing is production scheduling. Hence, any improvement in the scheduling system has significantly impacts time, quality, productivity, flexibility, and total cost. Improving these factors can easily increase the customer satisfaction. Multi-objective flow-shop scheduling with sequence-dependent set up time is considered as an NP-hard problem (nondeterministic polynomial time). Integrating inspection allocation with flow-shop scheduling problem results in a complex scheduling problem. For this degree of complexity, genetic algorithm is an excellent choice. Adding many inspection points in the production line can help to avoid reject items. However, the operation cost and productivity will be impacted negatively. The main objectives in this thesis are minimizing the inspection cost, minimizing the cost of processing defective items, and minimizing penalty cost by using genetic algorithm. Moreover, establishing a balance between cost and quality is extremely important in modern manufacturing management to avoid waste according to lean manufacturing methodology.

Keywords: *Flow shop scheduling; genetic algorithm; inspection allocation; optimization*

Dedicated to my lovely family members.

ACKNOWLEDGEMENTS

First, I want to express my appreciations to my advisor, Dr. Fantahun Defersha, for welcoming me to Guelph university and giving me the golden opportunity to do this project. I am very grateful for his guidance and support in completing this thesis. Without his suggestions and directions, I would not be successful so far. Furthermore, his knowledge and expatriates have inspired me throughout my academic study. I would also like to mention that he was a great supporter of me in all of my difficult moments such as COVID-19 pandemic days.

I would like to thank my academic advisory committee member, Dr.Sheng Yang, for his academic input and inspiration.

I am grateful for my parent's support and encouragement throughout writing this thesis and my life. I want to thank my parents for their support and for their daily encouragement to complete my thesis. To them I owe my success. Furthermore, I must express my gratitude to my wife, Dr. Ghyda Hashim, for her continuous support and understanding throughout my study. She is a great best friend and companion.

Finally, I am grateful to my kids, Adnan and Kinda, for being in my life and being patient when I was busy writing this thesis. This milestone would not be possible without my lovely family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

J Number of jobs where jobs are indexed as $j = 1,\ 2,\ \cdots J$;

D_j Requirement (demand or shipment size) for job j;

N Number of stages in the flow line where stages are indexed as $n = 1,\ 2,\ \cdots,\ N$;

Q Number of identical quality inspection stations located nearby the production line where inspection stations are indexed as $q = 1,\ 2,\ \cdots,\ Q$;

K Potential number of inspection on an inspection stations (K should be set to reasonable number);

$\Theta_{j,n}$ Defect rate of job j at stage n;

$I_{j,n}$ Inspection cost of job j at stage n;

$P_{j,n}$ Penalty cost of the processing of a defective items of job j that passed through Stage $n-1$ and processed in Stage n if there is no inspection after stage $n-1$ for job j;

C_j Penalty cost of per defective item of job j reaching the customer.

$T_{j,n}$ Processing time per unit for job j at stage n;

$U_{j,n}$ Inspection time per unit for job j following its processing at stage n;

$S_{j,n}$ Setup time for the processing job j at Stage-n if it is the first job processed at this stage;

$S_{j,n,j'}$ Setup time for the processing job j at Stage-n if the processing of job j immediately follows job j' at this stage;

M Large positive number;

$p_{j,r,n}$ A binary variable that takes the value 1 if the If job j is the r^{th} job processed by the machine at stage n, 0 otherwise.

$y_{j,n}$ A binary variable that takes the value 1 If job j is inspected by one of the inspection stations immediately after being processed at stage n, 0 otherwise.

$x_{j,k,q,n}$ A binary variable that takes the value 1 If job j is the k^{th} job inspected at station q after being processed at stage n , 0 otherwise.

$z_{k,q}$ A binary variable that takes the value 1 If the k^{th} inspection is conducted at station q , 0 otherwise.

b_j The batch size of job j needed to deliver a total of D_j units at the end of the production line ($b_j > D_j$ as some units will be rejected along the production line if they do not meet quality characteristics).

$e_{j,n}$ The completion time of the processing (machining) of job j at Stage n.

$v_{r,n}$ The completion time of the r^{th} processing (machining) operation at Stage n.

$f_{j,n}$ The completion time of the inspection of job j following its processing at stage n.

$w_{k,q}$ The completion time of the k^{th} inspection of stations q.

$g_{j,n}$ Number of good items of job j leaving stage n.

$d_{j,n}$ Number of defective items of job j leaving stage n if inspection is not conducted using one of the inspection stations.

k_q An inspection run counter which increase by one every time an inspection of a job is completed on inspection station q.

$\gamma_{j,n}$ The time at which job j is ready for its processing on stage n.

ψ_n The sequence (ordered list of jobs) by with the jobs are presented to stage n.

$\psi_n(r)$ The index of the job at location r of the ordered list ψ_n.

LIST OF ACRONYMS

TQM	Total Quality Management
NVA	Non-Value Added
SDST	Sequence Dependent Setup Times
QMS	Quality management systems
SMED	Single Minute Exchange of Dies
JIT	Just in Time
QA	Quality Assurance
QC	Quality Control
PC	Prevention costs
IOT	Internet of things
FIFO	first in first out
HFS	Hybrid Flow Shop
GA	Genetic Algorithm
ML	Machine Learning
SA	Simulated Annealing
WIP	Work In Progress
ACO	Ant Colony Optimization
MILP	Multi Objective Mixed Integer Linear Programming
FSSP	Flow Shop Scheduling Problem
HFSP	Hybrid Flow Shop Scheduling Problem
FFSSP	Flexible Flow Shop Scheduling Problem
TS	Tabu Search
TSP	Traveling Salesman Problem

RHS	Right Had Side
LHS	Left Hand Side
SPC	Single Point Crossover

Chapter 1

Introduction

The manufacturing process has become increasingly challenging and costly due to the emergence of new technologies. Therefore, reducing defects is a global strategy for many organizations worldwide. Many quality philosophies and management strategies focus on reducing or avoiding defects, including total quality management (TQM) and quality assurance. Lean manufacturing is the methodology that aims to reduce waste and non-value added (NVA) Shou et al. (2020). According to this methodology, defects and rework are considered waste. Any waste in the process can increase lead-time and manufacturing costs. On the other hand, avoiding waste and NVA could reduce total operation time and manufacturing costs. A rejected item is a product that not within the two control limits, upper and lower control limits Goh (1989). Many factors can lead to defects or product rejecting, including poor maintenance and weak process control Al-Najjar (1996). In a pure flow-shop scheduling problem articles are generally concerned about makespan. Nonetheless, in modern operations management, other factors may be important too. For example, total delivery time and quality are important reasons of growing the competition on the markets. Research in the area of production scheduling started in the 1950s, and it is considered one of the oldest research area in engineering. The industry has to provide a huge variety of final products or semi-final products, while providing clients with acceptable quality with affordable prices. Consequently, the demand is high for multi-objective problems in which all objectives can be achieved. In this regard, flow shop with sequence-dependent setup times (SDST) is the most diffcult class of scheduling problems. Additionally, multi-objective flow-shop scheduling with sequence dependent set up time is considered a complicated or NP-hard problem. Therefore, integrating scheduling problem with inspection allocation poses an extremely difficult challenge. Adding an inspection operation in the production line is one of the best steps to control the variation Winchell (1996). There are two types of process variation: the first one is the common case variation, whereas the second one is assignable causes variation Adler et al. (2011). The aim of the inspection is to identify nonconforming products before they are delivered to the end-user Winchell (1996). Additionally, the inspection process may help in corrective action Rabinowitz and Emmons (1997). Moreover, adding many inspection points can help to reduce the percentage of defective items that may reach the end-user. If company strategy aims to produce excellent quality and zero reject items, then a 100% inspection method is the easiest solution in this case. In other words, inspection point after each station during manufacturing operations can help ensure that companies provide clients products with zero defects. However, establishing a high number of inspection points is costly in many cases. It requires extra recourses such as inspection machines, tools, space, and workers. Therefore, optimizing inspection allocation and flow-shop scheduling could provide a good balance between inspection cost and quality. Moreover, scheduling is very important especially in manufacturing and logistic operations.

1.1. The History of Quality

There are many different definitions of quality. Overall, quality means freedom from errors Ito (1995). The concept of this topic can be traced back to medieval Europe. Quality control methods were used in many systems during the Manufacturing Revolution, during the 17th-19th centuries Montagna (2015). Due to the increasing number of workers and products, the defective products were either scrapped or reworked. Quality management systems (QMS) were first developed in the early 20th century, more precisely in the 1920s. Walter A. Shewhart introduced a quality control methodology known as the statistical sampling technique Hossain et al. (2010). During this period, a decline in quality control was observed because of an increased demand for more productivity Joseph M. Juran and W. Edwards Deming were the pioneer experts who developed many techniques in total quality management, and today many industries still rely on these techniques. Juran refined his theories on quality control at the New York University, and he wrote Quality Control Handbook in 1951 . Since then, the seventh editions of Juran's handbook have been printed. He was invited to Japan to talk about his theories concerning quality control, and he managed to change Japanese industries' attitude towards quality control. As a result, Japanese industry started producing higher quality products Herzog and Dworkin (2002). Dr. Juran, another key figure whose contributions to quality management are still admired today is W. Edwards Deming. Born in 1900, he earned degree in physics, mathematics, and engineering as well as a doctorate from Yale in mathematical physics. He lectured in the fields of maths, statistics, and physics for about 10 years. During this period, he also studied the statistical quality control principles of Walter Shewart with a view to expand Shewart's technique to administrative and management activities that were earlier limited to manufacturing only Fienberg and Stigler (2001). Slowly and gradually, the quality difference between the Japanese and American products started surfacing and, by the early 1980s, the difference became very obvious. In 1970, the new quality revolution has began. At that time, the quality of Japanese products exceeded those of Europe and the US. This was because the Japanese invented many tools and methodologies that could increase productivity and reduce the total number of defects Leksic et al. (2020). Poka-Yoke is a Japanese concept that means avoiding mistakes during production Leksic et al. (2020). On the other hand, JIT is a methodology that focuses on minimizing flow time and inventory cost. By the 1990s, Motorola had developed a new concept which is known as ”Six Sigma”. Some goals of this concept are reducing defects and problem-solving by applying DMAIC methodology Pyzdek (2003). After that, lean manufacturing and six sigma combined together and this integration became known as ‘Lean Six Sigma’ in the 2000s Pyzdek (2003). Finally, Quality 4.0 was initiated with the Smart Manufacturing-Industry Revolution 4.0 in the 2010s.

The Importance of Quality

The American Heritage Dictionary defines quality as, quality means a level or grade of excellence Teli and Bhushi (2010). Usually, users reject products or services that are of low-grade of quality, and they avoid dealing with companies that provide products with lower quality control as it can result in poor customer satisfaction, thus leading to low sales rate. Therefore, no company can persist in the market without producing a high-quality finished product. One of the consequences of poor quality is that each corrective action such as rework requires more raw material, operation time, and manpower. A weak process control can significantly increase rework and scrap rate. Additionally, poor quality means high operation costs. Nowadays, many companies around the world are focusing on improvement strategies to reduce manufacturing cost. To maintain operation costs, companies need to achieve high quality standards. Reaching that level requires them to provide the best possible service or final product to their end-users. To improve products' quality, industries are implementing various tools and techniques such as 5S, kaizen, lean six sigma and lean manufacturing. These tools help them manage their processes with increased efficiency.

Quality Control (QC)

QC is a management method employed to decide if items are acceptable or not according to manufactured quality standards. QC aims to prevent defects using inspection operations and efficient feedback. The efficient feedback can also be used as input in corrective plans. The goals of QC are as follows:

- To provide consumers with more acceptable products to increase profit.
- To produce good quality at optimal price.
- To confirm that all services and goods are satisfactory to customers.
- To measure the variation during manufacturing operations.

1.2. Quality Cost

Good-Quality Cost

This type can be defined as the price of providing an acceptable end-product from the first time without any additional actions such as rework Harrington (1987). Examples of prevention costs are continuous improvement, cost of quality management system, maintenance cost, and training cost.

Poor-Quality Cost

Poor-quality cost is a price or expense resulting from a rejected item. According to the article by Kumar et al. (2018), applying the six sigma methodology is an efficient way to decrease rejected quality cost. Additionally, this methodology helps to increase process capability as evident in the following figure:

Sigma Level	Defect Rate (PPM*)	Yield in %	Cost of Poor Quality (% of sales)	Competitive Level
6σ	3.4	99.99966	<10%	World Class
5σ	233	99.97670	10 to 15%	
4σ	6210	99.37900	15 to 20%	Industry Average
3σ	66807	93.31930	20 to 30%	
2σ	308537	69.14620	30 to 40%	Non-competitive
1σ	690000	31.00000	>40%	

Figure 1.1: Six Sigma Methodology and Poor Quality Kumar et al. (2018)

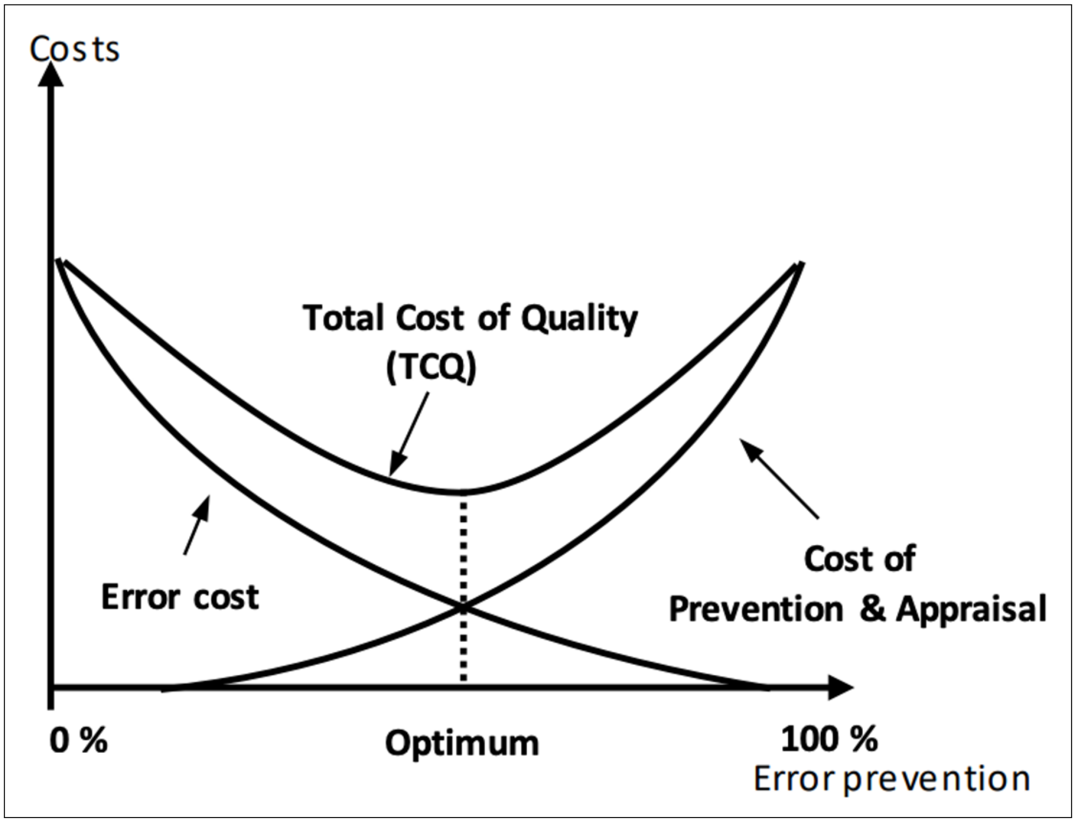

Figure 1.2: COPQ Stief et al. (2018)

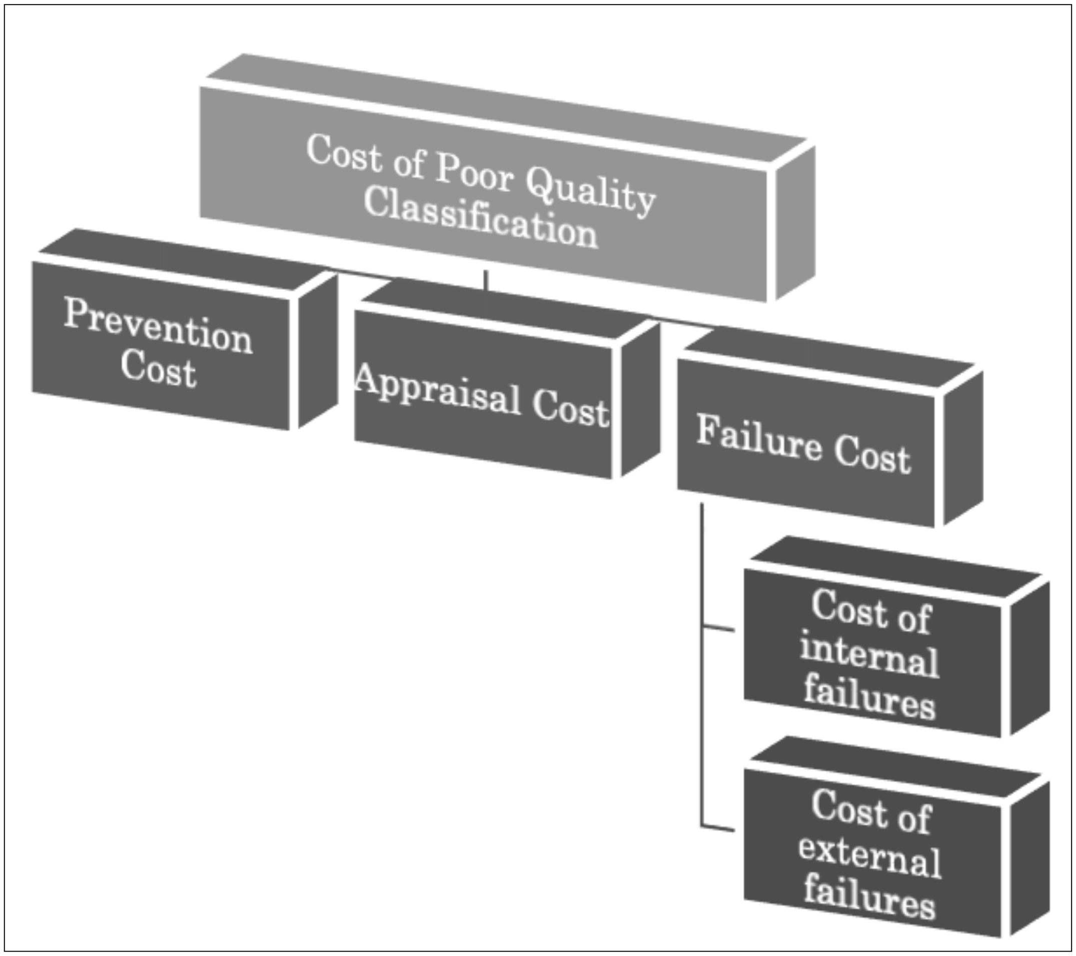

Figure 1.3: COPQ Stief et al. (2018)

1.3. Introduction To the Inspection Process

Inspection is the process that is used to decide whether materials, final products, and semi-final products are within an acceptable range Teli and Bhushi (2010). Inspection operation is an important task during manufacturing steps to filter out non-conforming items before sending to the consumers. Depending on the final product, a defective item can either be reworked or scrapped. Inspection, can be done after every stage during the manufacturing process. However, quality inspection in multistage production systems is not an easy task because these systems present different possibilities for defect rates, thus leading to increase in operation costs as all the stages can produce rejected items. Nonetheless, this process is highly critical for every business to maintain quality control. Thus, to achieve excellent quality of final products, companies should improve their inspection strategy. High non-conformance rate can cause costly rework and lead to customers dissatisfaction. To control customer complaints, businesses have developed good QC plans to filter out non-conforming products. Inspection planning is often the most challenging part in quality assurance and control (QA/QC). It involves identifying quality characteristics and inspection strategies. Full inspection is inefficient as it is time consuming and increases the total operation cost. However, without inspection, the quality of the products is not guaranteed. Consequently, academics have studied sampling methods to decrease manufacturing budget. Measurement or testing of products is a part of the inspection process. The outputs from any inspection process can result in passed or rejected products. One article notes that large sample size can lead to an increase in operation costs Ryan (2006).

Inspection Objectives

1. To measure the performance of the production line by collecting information such as scrap or defective ratio.
2. To maintain quality standards by sorting out poor-quality products.
3. To detect bad-quality products before sending to end-users.
4. To improve or recheck the final design of products.

1.4. Inspection Procedures

Inspection operations can be done in many ways. Floor inspection, centralized inspection, and combined inspection are some of examples of inspection techniques.

Floor Inspection

In this inspection procedure, inspectors must check materials or parts of the product during the manufacturing operation randomly. Floor inspection is an important inspection procedure to reduce material handling costs and avoid delays. However, it requires a highly skilled team of inspectors.

Strengths

1. Random inspection can be more effective than batch testing.
2. There is no delay during the manufacturing processes in floor inspection.
3. It saves time and inspection costs.
4. Inspectors can observe and report any problem encountered during manufacturing.

Weaknesses

1. There is a high possibility of having defects due to lack of experience.
2. It places pressure on inspector.

Centralized Inspection

Raw materials and product items may be tested at a centralized inspection centre. The centralized inspection can be done at a single inspection station or many stations during manufacturing operations.

Strengths

1. Stronger quality supervision.
2. Less pressure on the inspectors.
3. Systematic production flow.

Weaknesses

1. More material handling costs.
2. Delays and high waiting time.

Combined Inspection

The main target of this type of inspection is to avoid rejected items by combining the two previous inspection techniques. Through combined inspection methods, quality costs can be easily maintained.

1.5. Inspection Strategies

Ultimately, inspection is a process used to confirm the product quality during production. If the outputs regarding a product are not within the quality standard, then it will leave the system as scrap or rework. If the number of rejected items is high, operation management can take an action to accept or reject all the produced quantities. Inspections may help top management to make key decisions and control the costs Menipaz (1978). Therefore, it is important to realise the several constraints on quality inspection, including product quantity, and product budget Dudek-Burlikowska and Szewieczek (2009). Another article reports that mainly two types of inspection are used to inspect a final product or semi-final product: conformity inspection and monitoring inspection. In this regard, conformity inspection is a process that involves testing and certification Hinrichs (2011). The aim of this type is to determine if a product meets the requirements based on the production plan. On the other hand, monitoring inspection is necessary to achieve the best productivity and excellent quality Kurniati et al. (2015). It is also beneficial to identify the risks related to damaged tools and machines in the production line.

Full Inspection and Sampling Inspection

Issues encountered in inspection have motivated many researchers to develop mathematical models that can help to optimize the inspection process. In this thesis, a multi-quality features flow-shop manufacturing system is optimized by assigning different inspection allocations, which can lower inspection costs and minimize the total production time. 100% inspection method is included at all workstations to check in detail if items should pass or reject. However, this method requires many workers. On the other hand, the sampling method is faster and easier. In 1997, two authors set several inspection plans, 100%, or 0% inspection Rabinowitz and Emmons (1997). They believe that full inspection is more useful compared with zero inspection. Also, they note that full inspection can prevent defective products from reaching end users Rabinowitz and Emmons (1997). Moreover, full inspection is costly while no inspection is inefficient. The inspection decision in this thesis is based on a final product or production plan where each product can follow different quality plans. Additionally, inspection plans can be designated to minimize the total inspection budget. Instead, passing the undetected defect items can increase rework cost. Overall, our objective is to reduce cost and time based on the inspection policy for each final product. According to Al.Shayea et al., there are two inspection techniques, Sorting and

Process control Al-Shayea et al. (2020). The following are examples of products requiring full inspection:

- Health and scientific equipment.
- Airplanes.
- Jet engines.

Non-Conforming Strategy

After a rejected product is found through inspection operations, the decision of reworking depends on the cost, time, and many other factors. In this thesis, any defective item will exit the system after the inspection operation straightaway. In another way, a rework process has not been considered in the mathematical model.

1.6. Inspection Allocation

There is a link between defects rate and customer satisfaction. Therefore, every production line contains several inspection stations. Recently, the manufacturing costs have significantly increased due to the complexity of final products and quality requirements. According to the lean manufacturing theory, one of the techniques to minimize manufacturing costs is by speeding up the manufacturing process Winchell (1996). The time of inspection depends on the type of product: either it can be before, after , or during manufacturing. Therefore, achieving a balance between the inspection locations and defect rate is important in maintaining the operation cost. Industrial sector has always tried to provide good-quality final products with minimum operation cost. Thus, final product price and quality are two critical concerns in any production system. Any improvement in the quality can reduce the manufacturing cost. Many authors have explored some topics related to inspection allocation and its optimization. One of the objectives of optimizing inspection allocation is to reduce the operation budget. Bishop and Lindsay developed an inspection model to reduce inspection cost and the total number of non-conforming items Rabinowitz and Emmons (1997).

Quality Inspection Locations

Many researchers have discussed where to add the inspection process in the production system. Peters and Williams summarised the main rules to consider in the following five points Peters and Williams (1984):

1. An inspection process should be added after the operation or machine that may have the highest chance of causing defect items.
2. An inspection process should be done before the most costliest step in the production line. This step will help to avoid preforming high-cost operations on items that are considered non-conforming parts.
3. An inspection process should be performed after finishing all operations in the production line.
4. An inspection should be done before the most expensive machining process and after finishing the last operation.
5. A Sequence of inspection operations must be done after every machine usage the manufacturing processes. In this rule, inspection activities may a take long time. Hence, the percentage of rejected items in the last stage is low.

1.7. Manufacturing Processes and Classification

The manufacturing process involves transforming raw material into a final product. Some final products could be very complex. Therefore, many product providers have used advanced technology such as the internet of things (IOT), depending on the final product. Based on the quantity and variety of final products, production systems are divided into four main classifications: continuous, mass, batch and job-shop production system. However, this thesis focuses more on the pure flow-shop production system.

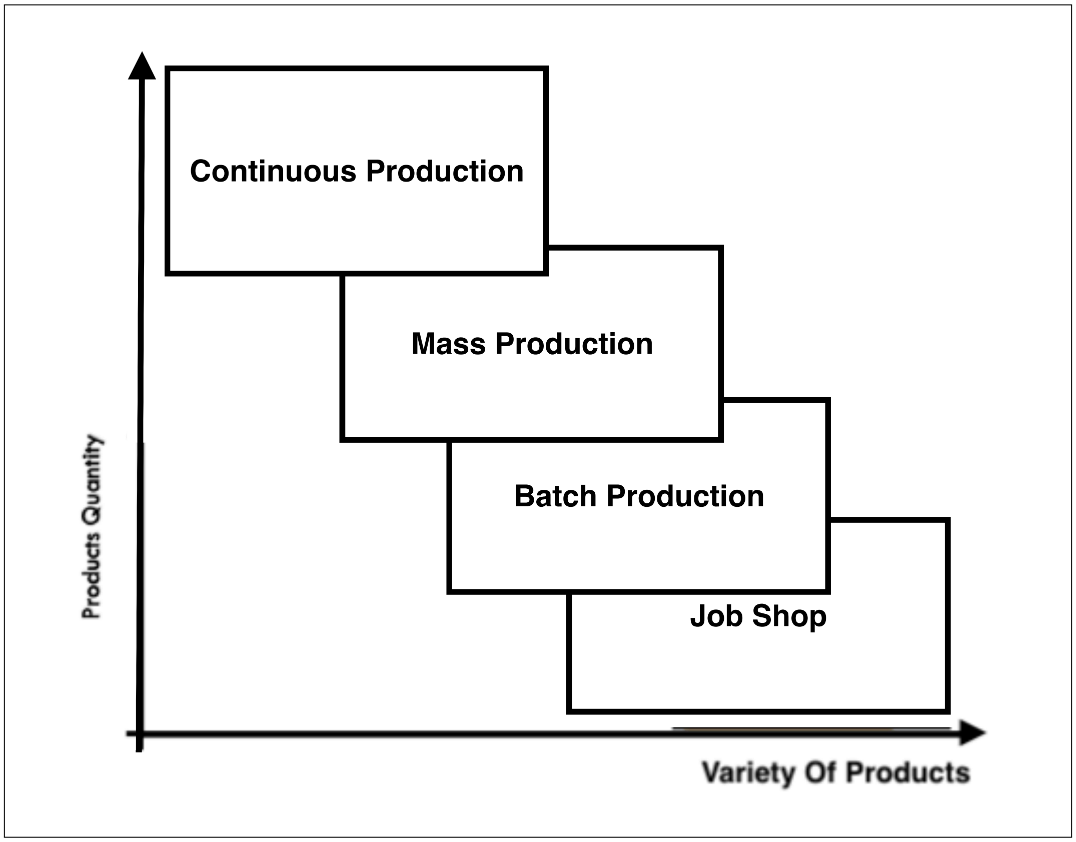

Figure 1.4: Manufacturing Processes and Classification

1.8. Introduction to Scheduling

In 1981, the concept of production scheduling was first defined by Graves. It involves the allocation of resources according to the requirements of a project Rodammer and White (1988). One of the most core factors in manufacturing environments is the production scheduling. It involves deciding the completion time, date, and task order where all processes should be finished within the shortest time Sangaiah et al. (2019). A study conducted in 2008 revealed that the concept of scheduling problems first emerged during the 1950s Allahverdi and Al-Anzi (2008). Furthermore, they noted that the number of flexible and challenging problems has increased significantly. Currently, it is possible to find a significant number of articles which cover deeply many scheduling problems with different levels of complexity. This kind of problem is characterized by three elements: a group of restrictions, machine environments, and objectives. Scheduling problems can be characterized as static versus dynamic and deterministic versus stochastic problems. If all information about the planning is complete at the time of planning, the problem is categorized as static. Conversely, in dynamic problems some parameters are unknown in advance. An excellent scheduling system can positively impact cost, efficiency, and customer satisfaction. Furthermore, end-user demand for different types of goods has contributed to an increase in product complexity. As such, more effort is needed to improve scheduling and production. Through scheduling, we can easily control some resources such as manpower and machines. Manufacturing scheduling is usually connected with planning of jobs on a variety of machines to manage the resources efficiently.

Significance of Scheduling

One of the objectives of scheduling process is to improve one or more targets, allocation of resources, and the total completion time. The tasks and resources in any business can take many formulaes based on final products or services. The resources may be machines in a production line and manpower. Additionally, every job order could have a definite urgency level. The objective function can also take many different forms. Optimizing of the number of jobs or the makespan, are some examples of scheduling optimization. Scheduling plays a significant part in manufacturing and service systems. The consequences of poor scheduling can be summarized in the following points:

- Increased cost of production.
- Delays and waste.

Terms in Scheduling

A job is a number of steps that need to be managed by several workstations. Every task has processing time related to each machine.

Processing Time is the amount of time that a job needs to spend on a specific machine in order to be processed.

Idle time is the time when a machine doesn't have any job or task for processing.

Makespan is the total duration time needed to finish all jobs.

Some Scheduling Tools

A Gantt chart

A Gantt chart or a bar chart is a kind of diagram that was made by Henry Gantt in 1910 to 1915. A Gantt chart is used as a visual diagram to assist in planning and scheduling. It can demonstrate the relationship between all orders and total duration. Figure (1.5) is an example of Gantt chart.

1.9. Machine environment

The machine environment plays a dynamic role in shaping the complexity of production scheduling problems. Therefore, it is important to analyze the results of the various methodologies and techniques used in the development of these problems.

Single-Machine Problems

The most common type of machine environment is the single-machine situation. It is considered to be the easiest of all the machine environments. It involves only one machine to process many jobs . The purpose of the single-machine problem is to schedule many jobs on a single machine, where jobs can be either independent or dependent. The single-machine case is recognized to be NP-complete. In this scheduling problem, the makespan is the same for all the sequences. Therefore, it is not a part of the list of measures of performance.

Parallel-Machines Problems

A parallel-machine scheduling problem is a type of single-machine problem as we mentioned previously, and it can be used in combination with other problems such as shop scheduling. Due to its practical importance, this type has received significant consideration. In addition to being used in real-world situations, the techniques for solving this problem can also be utilized in multi-stage systems decomposition procedures Liou and Hsieh (2015).

Open-Shop Problems

In this type of scheduling problem, every job has its own operation time and only needs a single machine to finish the work. Yet, no boundaries are set on the sequence of the whole operation. This allows arbitrary selection of the job operations. Each machine is able to deal with one job in the same period of time. This type of problem aims to create a schedule of operations that will allow the machines to perform their tasks in a certain order. It can also minimize the overall time it takes to accomplish a job. This type of problem first emerged in 1976 by Sahni and Gonzalez Gonzalez and Sahni (1976).

1.10. Scheduling Problems

Flow Shop

If all machines in production line are arranged in series, the environment is classified as a flow shop. In this environment there are many machines in series and every job has to be processed on each one of the machines. Every job should follow the same sequence. Every job must be processed on first machine, then the second machine, and so on until it is completed on the last machine in the production line. Keeping the sequence of all jobs as first in first out (FIFO) is the most common assumption in this kind of scheduling problem. Additionally, reducing the completion time of jobs in a flow shop with two workstations is considered to be NP-complete.

Job Shop

Job Shop scheduling (1.6) is similar to a typical combinatorial optimization problem except that whole jobs could or could not pass through some of the workstations in the manufacture system. For example, job 1 can be processed in $machine - 2$, $machine - 5$, $machine - 6$, $machine - 8$ and $machine - 10$ only. While, job 2 needs to be processed through the all workstations ($frommachine - 1tomachine - 10$) to be get completed. Each workstation performs a different category of jobs. Ultimately, job shop is a complex problem. In this type of problem, the organization of orders impacts the makespan directly.

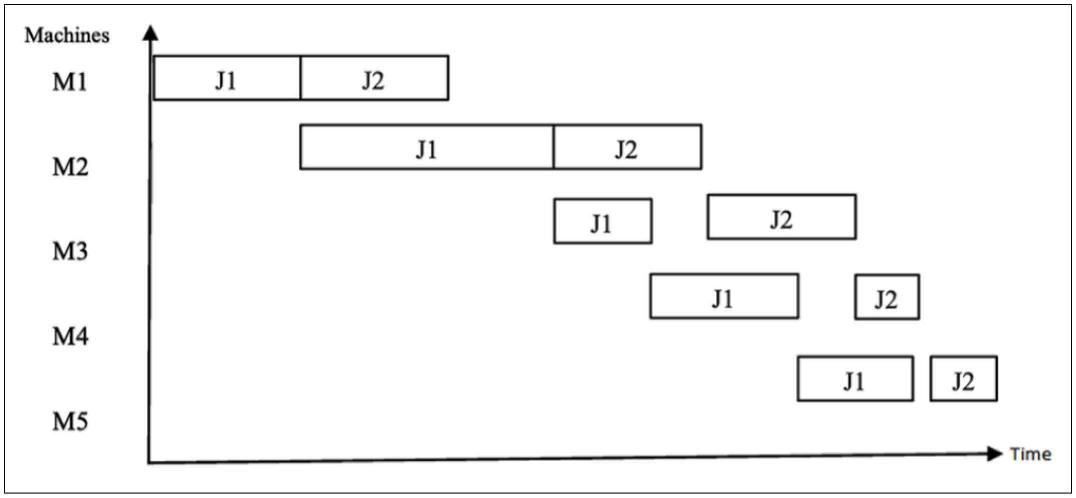

Figure 1.5: Gantt chart

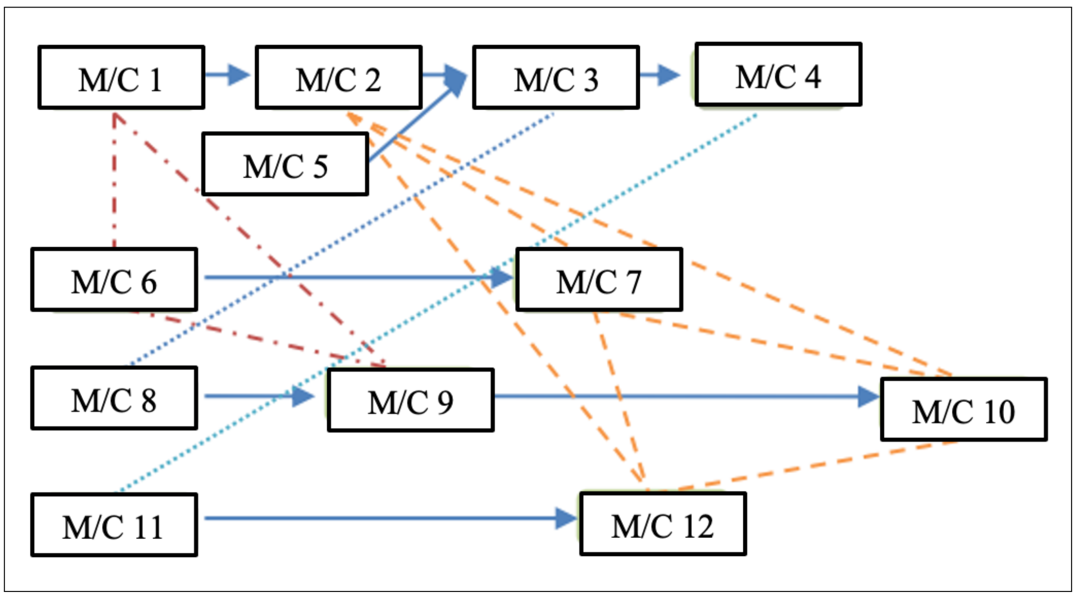

Figure 1.6: Job Shop

1.11. Optimization Algorithms to Solve Scheduling problems

Before going further in describing modeling tools and methods that are used in $job/flow$-shop problem, the word optimization and other related definitions in scheduling have to be introduced. Optimization can be expressed as a method that manages resources efficiently to achieve the greatest goals in a professional and efficient way. Optimization tools can easily provide accurate and efficient information. Optimization algorithms are divided into two groups: deterministic and stochastic algorithms. A deterministic algorithm considers a finite number to optimize any complex problem, while the stochastic technique is used for worldwide optimization problems with bound constrained and unconstrained. Additionally, stochastic approaches are combined to obtain best global solution. It ignores repeating similar solutions. Many articles note that, if clustering is required for any problem set, the stochastic methods are the best optimization option in this case Boender et al. (1982). The stochastic algorithm is further split into two types: heuristic and metaheuristic.

1.12. Search Algorithms

Some global search techniques are briefly explained in the following two sections. More specific descriptions about genetic algorithm technique can be found in Chapter 4.

Genetic Algorithm

Over the last decade, the applications of genetic algorithm have been significantly increasing in different areas such as operations management, and production scheduling. Genetic algorithm (GA) was initially developed by John Holland Mitchell (1995). As one of the most widely used in optimization techniques, it is genetic algorithms, which are designed to analyze the genetic structure of a biological individual. Each of the individual's chromosomes contains several genes. Due to its popularity, genetic algorithms have been successfully used in various optimization problems over the last 30 years. GA offers great potential in solving hard problems by decreasing the execution time. The rise of GA has been regarded as a significant development in the field of adaptive search techniques. Currently, there is a positive sign that GA is useful for optimization and dealing with difficult problems Gupta and Stafford (2006). Furthermore, increasing attention has been recently paid to integrating machine learning (ML) problems and GA Mitchell (1995). As we mentioned previously, GA has been commonly used in job-shop and flow-shop scheduling. However, its use is not confined to this field only. A recent study demonstrates that GA can be used to optimize electrical circuit designs. Additionally, in 1995, Mitchell provided some examples of using GA including, economic models, immune system models, ecological models, automatic programming, and machine learning (ML) Mitchell (1995).

Fundamentals of Genetic Algorithm

Typically, in engineering and mathematics, the optimization problems are first formulated as mathematical expressions, and then an optimal solution is found. Additionally, GA involves information transfer in natural organisms through a sequence of genes known as chromosomes. Through the transfer of information, a certain organism can excel against other competing organisms. This allows the surviving organisms to reproduce. GA is commonly used in scheduling problems due its ability to find optimal solutions or near-optimal solutions in a short time. Hence, it has gained an increasing popularity as a search tool that can be used for complex optimization problems. The flow chart in fig 1.7 illustrates the basic steps of GA.

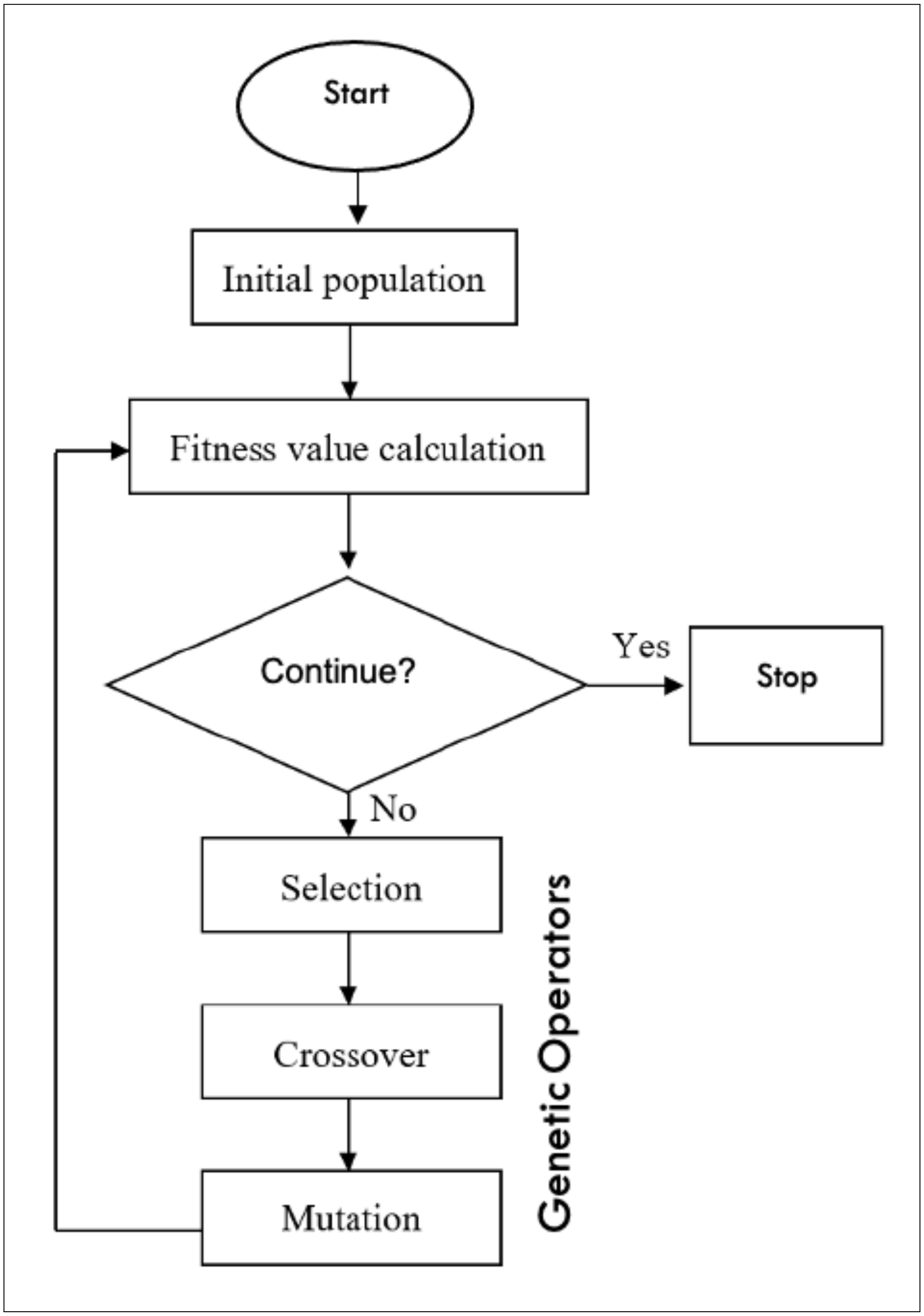

Figure 1.7: GA Operators

Selection

There are various types of selection operators that were developed over the past years, including k-way tournament selection, rank-based selection, and roulette-wheel-selection. For example, the roulette wheel selection operation is a straight forward selection model where solutions are selected based on the fitness. However, in this thesis work, we have considered the k-way tournament selection. In this type of selection process, the individual k is selected randomly. Also, it may be adjusted depending on the solution. The k individual with the highest fitness value is more likely to be the winner. Lastly, this process continues until the number of individuals equal to the population size is equal to the size of the population or reach to stop condition.

Crossover

There are many crossover techniques, including the one point crossover, two-point crossover, k-point crossover, shuffle crossover and uniform crossover.

Mutation

This genetic operator is essential to maintain genetic diversity from one generation to another. It can slightly change of the new child's chromosomes to get a better solution. As with the previous operators, several mutation operators can be used in GA including Flip, Swap, Inversion, and Shift mutations.

1.13. Chapter Summary

This chapter offers background information on quality, inspection process, scheduling, and some common methodologies for solving scheduling problems. First, a brief background on the history of quality is provided in Section 1.1, followed by quality control in Section 1.7. General information about inspection process is provided in Section 1.7. Section 1.10 discusses scheduling problems, and section 1.17.1 is about optimization algorithms used to solve scheduling problems.

1.14. Organization of Remaining Chapters

After the introduction about quality, inspection process, and scheduling, Chapter 2 provides a review about flow-shop scheduling problem and inspection allocation. Also, in this chapter we discussed the difficulty of integrating flow-shop scheduling with sequence-dependent setup time and inspection allocation. Chapter 3 centers around the mathematical model. The three mathematical models are as follows: inspection allocation model, sequence dependent model and instigated inspection allocation with scheduling model. Additionally, all the variables are defined and explained in Chapter 3. Chapter 4 explains the solution procedure and how it works. The numerical examples are explained in Chapter 5. Chapter 6 concludes by providing a comprehensive view of the thesis and reviews suggested ideas for future work.

Chapter 2

Literature Review

2.1. Introduction

Certainly, any business in the market focuses on customers satisfaction as raising customer satisfaction increases business profitability. In this regard, cost and quality are the most critical factors that directly impact customer satisfaction. Therefore, many industries have invented some methodologies to reduce output error. For example, Toyota has added a tool to implement lean manufacturing, and it is known as Poka Yoke Dudek-Burlikowska and Szewieczek (2009). The ultimate objective of this concept is to avoid mistakes during the operation; fewer mistakes lead to fewer defects. Six sigma and Motorola are other examples that demonstrate the relation between cost, defects, and customer satisfaction as discussed in Chapter 1. Nevertheless, quality improvement is not an easy task. In many cases, if the quality is upgraded, operations costs will increase due to inspection costs and other factors. As such, businesses require a balanced quality strategy to help in cost reduction as well. Another tool used to lower the ratio of rejected items is optimization in the scheduling of production. Scheduling is a very critical factor in production management. Perfect scheduling plan means increasing in efficiency and capacity; in other words, any improvement in production scheduling leads to an increase in total profit. Based on the final goal, there are many ways to optimize production scheduling, including job shop and flow shop. A pure-flow shop problem focuses mainly on reducing total operation time of all jobs. Yet, in modern manufacturing systems, on-time delivery and cost saving are important factors too. Businesses have to provide diverse products, and clients expect the final products to be delivered on time. Consequently, there is high demand on multi-objective scheduling. A flow-shop problem with sequence-dependent set up time is NP-hard with greater complexity and cannot be solved in a reasonable time Tavakkoli-Moghaddam et al. (2008). One goal of this thesis, is to find the optimal inspection allocation based on many factors such as cost, completion time, and risk.

2.2. Inspection Allocation

Inspection policy can be selected based on cost and other factors. For example, implementing inspection processes after each station in flow shop can be a good technique to control the defect rate; however, the operation cost will be high in this case. High operation costs lead to increased products prices. Thus, it is crucial to optimize the inspection allocation to control the whole manufacturing chain. In the recent years, some articles have discussed topics related to optimization inspection allocation. In particular, an author provided a complete review of mathematical models to improve inspection processes in production line with multiple stages Raz (1986). He notes that the mathematical models can be used in both parallel and serial production systems. In 1981, Yum and Mcdowell developed a mathematical model for non-serial manufacturing systems Ohta (1974). They solved the model by using nonlinear integer programming. This study reported many assumptions; for example, only one inspection process was performed after each operation. Automating the inspection process is another way to improve inspection strategies Ohta (1974). This helps detect the defective items faster compared with regular inspection process. Consequently, Raz and Thomas developed an algorithm to provide optimal inspection order Raz and Thomas (1983). E. Trovato studied several inspection strategies and analyzed them Trovato et al. (2010). One of the purposes of this study is to minimize cost of scrap and rework. Moreover, Chen in 2013 explored optimal inspection allocation to minimize rework cost Chen (2013). Another article explored different methods of inspection to produce high-quality final products with low price Peters and Williams (1984). In 1982, two authors established a model to find the optimal spot of inspection processes in production lines with in multiple stages. The model dealt with several errors, both predictable and erratic. They noted that this model can provide information to management about the optimal number of inspection stations Ballou and Pazer (1982). Furthermore, another article studied inspection allocation problems You et al. (2021). They developed a mixed integer-programming model to decrease inspection costs in every workstation. Mandroli et al. conducted a literature review on inspection process in manufacturing Mandroli et al. (2006). A further study proposed a model is to allocate the inspection station in a production line, and it is also known as the shortest route model White (1969). This model considers that inspection should occur after each stage. After the model is implemented, the optimization strategy focuses on finding the configuration that decreases the cost. Another study developed a stochastic search algorithm to find the best locations of an inspection process in a manufacturing line Viswanadham et al. (1996). Cost of the process is computed based on various factors such as inspection cost, processing cost, and scrapping cost. To find the optimal allocation of an inspection station in a multistage process, dynamic programming was used by Eppen and Hurst in 1974 Eppen and Hurst (1974). This method considers the various factors that affect the cost structure of the process. Parts that fall within the area of inspection are then allocated according to their related revenue. In some studies, the interrelationship of various quality features was also taken into account. It was assumed that failure rates of each stage can't be evaluated independently. The production layout of different stages was also studied based on this assumption. In 2000, Veatch explored the inspection allocation problem to find the best strategies for the inspection process in multiple workstations Veatch (2000). The study developed an economic model that considers the various factors that affect the inspection process. He stated that sampling inspection can be cost effective only if there is a huge variation between lots and defect rate Veatch (2000). Verduzco et al. studied inspection allocation in the electronics assembly line Verduzco et al. (2001). This study aimed to determine which parts should be tested and at which inspection station. The authors have developed a greedy heuristic technique to find the optimum solutions Verduzco et al. (2001).

2.3. Elements Impact Inspection Process

Production Line Structure

Multistage production systems are commonly used to process products. These systems involve the transformation of raw materials into finished products. Production systems can be classified into serial or non-serial systems and assembly systems Rezaei-Malek et al. (2019). The raw materials used in a serial production system go through a sequence of steps before they are finished. Conversely, in an assembly system, the finished products are assembled or fixed by the other processing lines. A non-serial system is a type of production system that doesn't involve assembly Taneja and Viswanadham (1994). It is more challenging to identify defective components in an assembly line compared with a serial system because various serial lines in a production system join together in one single line. Therefore, rejected parts in an assembly line can affect the total number of finished products.

Time of Inspection Process

To control the overall budget of manufacturing, the time that it takes to inspect a part can also affect the quality of the inspection. To understand the performance of various inspection stations, a study performed a simulation model analysis Taneja and Viswanadham (1994). In one study, the authors found that the time spent inspecting a part was the most influential factor when it came to the selection of heuristics rules Lee and Unnikrishnan (1998). In another article, the authors discovered that inspection strategies can be reduced by considering inspection process time Shin et al. (1995). They also ascertained that by increasing the number of inspections stations, they can maintain low bottleneck time. In production, time of inspection can play a key role in the operation cost. High inspection time can lead to an increase in work in progress (WIP). Furthermore, time of inspection for every inspection position may be represented by cost of inspection.

2.4. Flow-shop Scheduling Problem

In any marketplace, distribution speed has become a way to minimize the competitive gap. Thus, scheduling plays a major role in logistic supply chains or manufacturing processes overall. The processing routes are very similar for all work orders in a classic flow shop scheduling Solimanpur et al. (2004). The number of studies on flow shop is small compared with other fields of manufacturing. However, scheduling problems in flow shops are among the most common issues that affect the operations of businesses. Numerous studies have been conducted on this issue Murata et al. (1996). One study explored the improvement and achievable solutions in the last 50 years of flow-shop scheduling problems Gupta and Stafford (2006). In another study in 2005, authors solved a flow-shop scheduling using algorithms M'Hallah and Bulfin (2005). In this problem, the total number of the machines is two. The study found that, decreasing the weighted number of tardy orders is the objective function. Grabowski and Pempera discovered the flow-shop problem with no wait to optimise the total completion time. To optimize this problem, they developed different local search algorithms M'Hallah and Bulfin (2005). Other authors tried to optimize the flow-shop scheduling problem by optimizing the makespan of the two machines Wang et al. (2006). In 1996, Akpan studied a new method of job-shop scheduling problems Akpan (1996). The method was formulated on a network scheduling technique. The goal was to reach a minimum operation period. In this regard, hybrid genetic algorithm is another way to optimize job-shop or flow-shop scheduling problems. In another article, the authors developed an efficient method to solve job-shop scheduling problem based on two types of GA: single and parallel genetic algorithm Park et al. (2003). They reported that hybrid genetic algorithm provides a better solution compared to the traditional GA. Wang et al. provided an excellent comparison between tabu search and simulated annealing to deal with complex scheduling problems Wang et al. (2006). The authors believe that, in complex problems, simulated annealing becomes more useful than tabu search. A branch and bound method was developed by Sayin and Karabati to solve scheduling problems. The problem consists of two machines in a flow-shop environment Sayin and Karabati (1999). The objective of optimization method is to reduce makespan and total completion time. In the same sense, the study by Park et al. provides another example of optimization flow-shop optimization using the branch-and-bound technique Park et al. (2003). Another paper proposed different optimization techniques that can deal with complex flow-shop scheduling problem, including the multi-objective simulated annealing algorithm to optimize the flow-shop scheduling problem Jarosław et al. (2013). Recently, the just in time methodology has been applied in flow shop scheduling problem to minimize costs Fuchigami et al. (2018). On the other hand, another paper studied the GA approach to achieve minimum completion time for hybrid flow shops Xiao et al. (2000). The first branch-and-bound algorithms were developed by Schrage and Ignall in 1965 to solve flow-shop problems. The goal of this study was to minimized makespan Brah and Hunsucker (1991). After that, in 1972, Lockett and Muhlemann proposed an algorithm for scheduling jobs with sequence dependent setups to optimize tool changes during production Hwang and Sun (1998). In 1986, Gupta applied the same optimization technique, bound and branch algorithm, to minimize the setup cost for flow shops with several number of jobs and machines. He found that this method can be used for small problems Gupta (1986). In 2004, Toktas et al. also discussed a branch-and-bound method to optimize flow-shop scheduling problems, aiming to minimize the makespan and maximize the earliness Toktaş et al. (2004). The authors noted that the branch-and-bound procedure is very effective in solving flow-shop scheduling problems with a maximum of 25 jobs. In 1992, Chaudhuri and Rajendran proposed a heuristic algorithm that can reduce the flow time of a flow-shop scheduling problem by considering the various factors that affect the efficiency of the system. The first criterion focuses on the total number of idle times, while the second one takes into account the waiting times Liu and Reeves (2001). Finally, the third criterion considers the various stages of the schedule. In 2004, the two researchers optimized flow shop scheduling by using two ACO algorithms, aiming to reduce the total makespan and flow time Rajendran and Ziegler (2004). Other researchers developed a meta-heuristic-based ant colony optimization to solve flow-shop scheduling problems. They assumed the flow-shop to consist of different workstations to process many jobs in a fixed order. The final goal of this problem was to minimize the makespan Ying and Liao (2004). Furthermore, neural networks can also be used to optimize flowshop scheduling problems. In this regard, a neural network is a type of mathematical model that is based on the principles of the biological nervous system. It consists of neurons. Typically, a neural network has at least two layers: an output layer, and an input layer. The back propagation network is one of the most common models. This network consists of an output layer, input layer, and hidden layers. In 2005, Tang et al. applied a neural network-based model for a hybrid flow-shop. The neural network model in this problem consisted of many layers. They used simulation for training purposes Tang et al. (2005).

2.5. Integrating Inspection Allocation and Scheduling

With the growth of manufacturing around the world, the rate of defective items has become a main topic in manufacturing management. Although publications related to inspection operation are on the rise, more effort should be done in this field to reduce the gap between inspection allocation and scheduling Galante and Passannanti (2007). Inspection operation is neglected in many mathematical models including pure flow-shop and job-shop scheduling problems. However, inspection time might be affecting the total processing time. Therefore, adding inspection processes to any production model can be a significant step toward making production lines more efficient. Usually in industry, quality policy is selected independently from quality information feedback. Also, in many cases, inspection allocations are chosen according to the age of the machines or quality assurance estimation without considering many factors such as inspection cost and total cycle time. However, as an alternative, we can optimize the inspection process in multistage production systems without impacting the final product quality. For this purpose, increasing production line efficiency can be achieved either by avoiding unnecessary inspection processes that can increase waiting time or by avoiding processing rejected items. Moreover, inspection allocation is important to prevent machines from producing more rejected products, thus improving quality and reducing inspection cost of the production system. In 1991, Kaspi and Raz have studied the sequencing and location problems for several inspection stages. The production line in this article consisted of many stages. Authors developed a nonlinear mixed integer programming model to solve this problem. The study aimed to decrease inspection cost per element Raz and Kaspi (1991). A mathematical model was developed by G. Galante and G. Passannanti to combine inspection and scheduling Galante and Passannanti (2007). They found that selecting inspection points without cooperating with the scheduling part could lead to incorrect management decisions; additionally, the authors noted that this is the first paper to integrate scheduling and inspection in a job-shop manufacturing system. They used GA to optimize the cost in this problem. Suryadhini et al. developed a model to integrate flow shop and inspection activities. The flow shop in this research consisted of three stages. The target was to decrease total flow time. The authors applied a heuristic algorithm that developed by Halim et al in 1994 with some adjustments to solve the scheduling problem for a multiple-stages flow-shop manufacturing system Suryadhini et al. (2021). In 2020, a study developed a model that integrated scheduling and inspection planning for different number of jobs. The authors noted that each job had a unique operation time. additionally, they mentioned that there were only two workstations in the production line. Overall, this article aimed to minimize the cost of inspection and achieve the best sequence for all jobs Sinisterra and Cavalcante (2020). Furthermore, another study conducted in 2017 combined the scheduling and inspection process. The authors developed a mathematical model to discover the best solution for this problem. This study involved two inspection policies and three production stages. The authors optimized the problem by using a hybrid bee colony algorithm Wang et al. (2017). Yumbe et al. optimized an inspection scheduling plan for facilities of power distribution facilities using a heuristic algorithm, aiming to minimize the cost of inspection process. They reported that the cost was minimized by 14 percent. It is noteworthy that some of the input data in this problem is based on inspection history and customer feedback Yumbe et al. (2017). Elizabeth M.Jewkes in 1995, integrated inspection efforts and scheduling for one stage workstation. The author has added an extra load on inspection, aiming to determine the optimal inspection effort Jewkes (1995).

2.6. Research Motivation

The increasing complexity of production systems and the need for faster response times and lower costs are some of the factors that have exerted competitive pressures on the whole logistic supply chains. The multi-objective scheduling problem with sequence-dependent setup times has been an object of investigations for years. Most of literature on solving flow-shop scheduling problems are limited in the number of machines and the objective functions Eren (2009), Wang et al. (2017), Sinisterra and Cavalcante (2020),and M'Hallah and Bulfin (2005). Moreover, many articles do not consider the inspection process in their models Galante and Passannanti (2007). As discussed in the previous chapters, adding many inspection points during the production line leads to increased lead time and operation costs. However, determining optimum operation time and cost are essential requirements in manufacturing. Moreover, from the literature, we can observe that a limited number of articles are related to optimizing inspection allocation in job shop or flow shop. Therefore, we tried to achieve a balance between delivery time, defect rate, and cost to improve production flow 2.1. Furthermore, these kind of problems are considered as NP-hard problems Tavakkoli-Moghaddam et al. (2008). Integrating inspection allocation with complex problems such as flow-shop scheduling with sequence dependent setup time is a challenge. This encouraged us to do research in the area of combining optimization in inspection allocation and flow-shop scheduling problem by using GA. Accordingly, in this thesis, we aim to bridge the research gap between flow-shop scheduling and inspection location. We propose a mathematical model for flow-shops where each job has a different inspection plan. We believe that the mathematical model and solution techniques proposed in this thesis can be easily used in various manufacturing applications with some modifications. As we mentioned in Chapter 1, there are different scheduling problem types. One of the easiest scheduling problems is pure flow shop, as shown in fig 2.2. However, scheduling becomes complex when sequence-dependent setup time is considered. The following points are summarised in our research motivation:

- Few articles have considered inspection allocation in manufacturing systems with different types of quality.
- To the best of our knowledge, there is no paper that integrates multiple-stages flow shop and inspection allocation with sequence dependent setup time.
- No paper has considered inspection cost, penalty cost of processing of a defective items and makespan to be optimized in one problem.
- We believe that developing more effective metaheuristic algorithms can help us to solve more complex problems in the future, and an example of these algorithms is the flexible flow shop integrated with inspection operation.

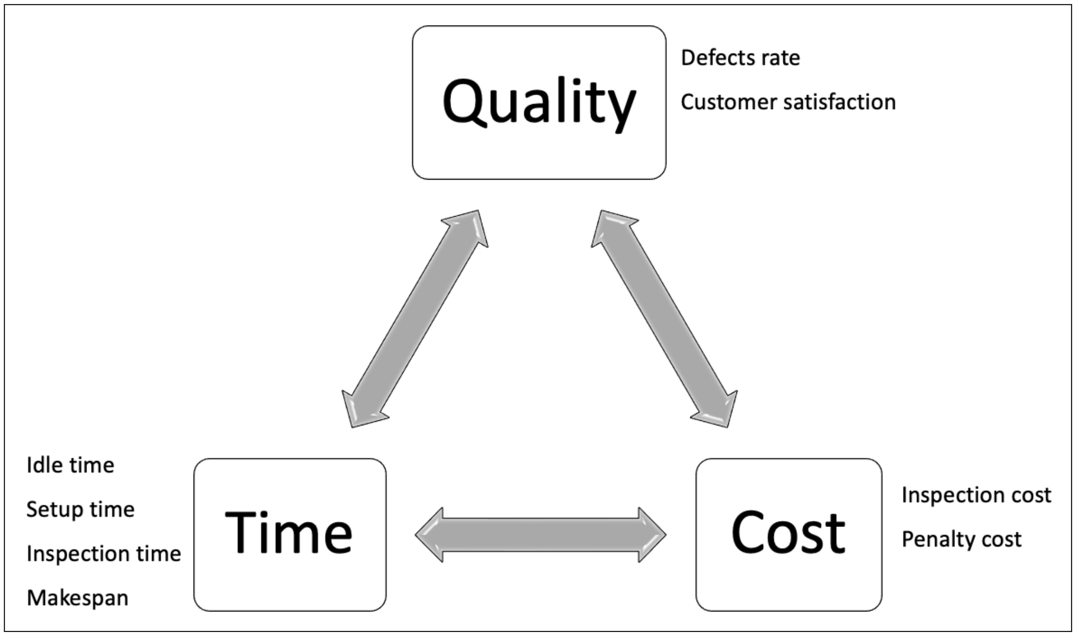

Figure 2.1: Thesis Main Goals

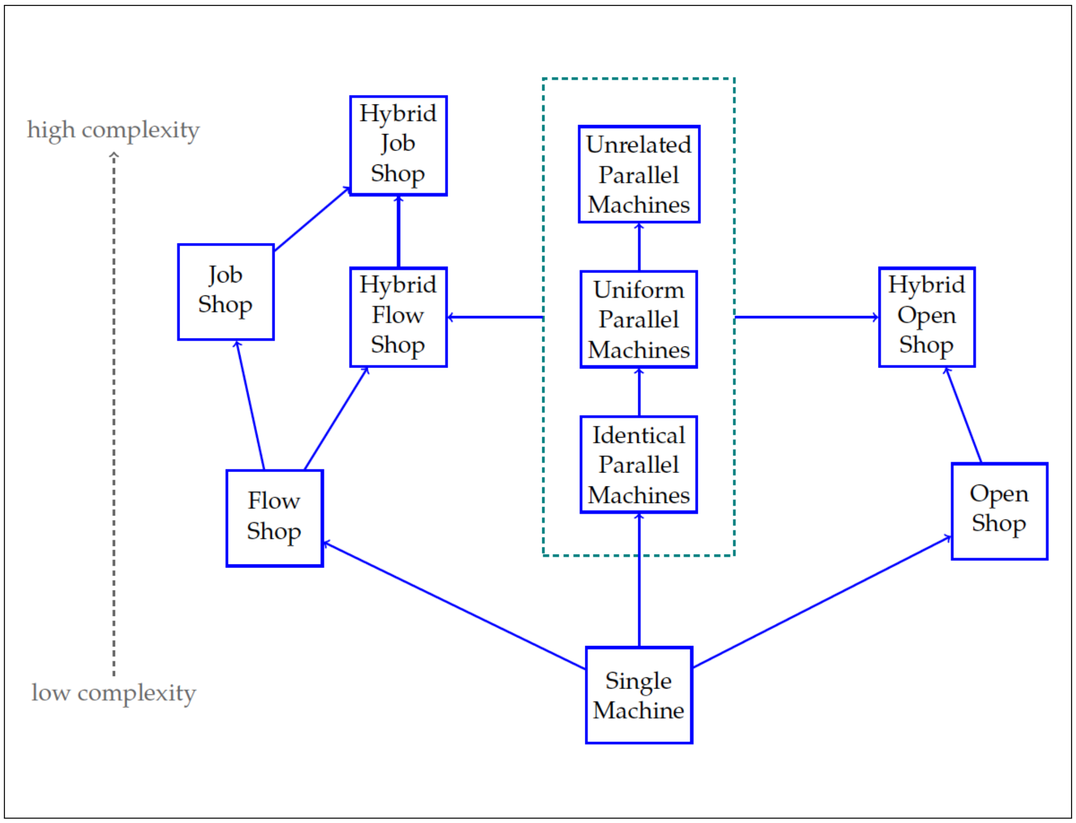

Figure 2.2: Complexity Of Scheduling Problems Framinan et al. (2014)

Chapter 3

Mathematical Model

3.1. Introduction

The process of scheduling tasks is commonly used in various industries. It involves making decisions about how to allocate resources. Industrialists have always been committed to producing products that are both cost effective and environmentally friendly. As such, Toyota has become a global brand for car manufacturing. Through its continuous improvement efforts, Toyota has been able to reduce the non-value added components in its products. By doing so, they can send their products to their customers without any defects. However, this method increases the cost of the product and can lead to increased rework. Additionally, due to the increasing examination costs, the company needs to balance the quality and cost of its products. In this context, this thesis proposes a balanced inspection strategy that involves station allocation and multi-quality characteristics in a flow-shop scheduling system. Regarding manufacturing, scheduling is the method of finding the best balance between limited resources and several jobs. In a pure flow-shop system, a set of jobs needs to be processed on a sequential set of machines. Each job requires several operations, where each operation must be performed on a number of machines. The scheduling of jobs is very important for various service and manufacturing industries to maintain low costs and maximize their profit. However, it can be very challenging to implement and manage these procedures due to the complexity of the task. This section discusses the following points:

(A) Inspection Allocation Model.

(B) Model for Sequence Dependent FSP

(C) Integrated Inspection and Scheduling Model.

3.2. Problem Definition and Assumptions

Consider a flow-shop consisting of multiple production stages to process several jobs, where each job must go through the processing in every machine on the shop floor. Each job has to be processed in the first stage ($n = 1$), then the second stage, and so on. Additionally, each stage in the production receives a batch of row material. The jobs have different processing times and different inspection plans. There is a sequence-dependent setup time on each available machine. Accordingly, in this case, arranging the order of all jobs is important to minimize makespan. A possible inspection point is followed by an actual manufacturing operation. Each workstation in the production line may introduce defective items. When the manufacturing process is finished, all items are supplied to the customers. Defective items can only be discovered through an inspection process. Otherwise, a defective item will continue through the next stages and will be shipped to the customer. There are two outputs after each inspection operation: pass or reject products (no rework). The percentage of defective items in the system is based on the total defect rate. Also, the production line has several inspection stations, and each inspection station can receive any job order. To further reduce the number of rejected items, more inspection processes should be added; however, this option is costly as we mentioned in previous chapters. Therefore, the objective is to find the optimum inspection allocation for each job or product. The assumptions are as follows:

- The availability of the machine is 100% (no break or failure event).
- Each job is available at time $t = 0$.
- There is no transportation time (the transportation time between stages is negligible).
- The number of inspection machines are limited.
- All machines are capable of handling one job order only at a time (only one machine available for each operation).
- Every final product needs to be processed continuously.
- The maximum number of jobs is represented by J $(1, 2, 3, 4, 5, 6, 7, , J)$.
- N represents the last station $(1, 2, 3, 4, 5, 6, , , , N)$.

3.3. Inspection Allocation Model

The inspection allocation model (3.3.2) can be used in production to minimize rework and raw materials costs by improving the quality during manufacturing operation. The goal of improving the quality of a product is usually motivated by the cost considerations. In some cases, the contractual agreement between the producer and the customer may also specify a minimum or acceptable quality level that the company is expected to meet. Without a mathematical model, it can be difficult to estimate or analyze the defect rate.

3.3.1. Notation

We define various notations that are used to describe the model:

Problem Parameters:

D Requirement (demand or shipment size) for the product.

N Number of stages in the serial line where stages are indexed by $n = 1,\ 2,\ \cdots,\ N$;

Θ_n Defect rate at stage n.

I_n Inspection cost at stage n.

P_n Penalty cost associated with the processing of defective items generated in Stage $n-1$ and processed in Stage n if there is no inspection after stage $n-1$.

C Penalty cost per defective item reaching the customer.

M Large positive number.

Continuous Decision Variables:

b The batch size needed to deliver a total of D units at the end of the production line ($b > D$ as some units will be rejected along the production line if they do not meet quality characteristics).

Binary Decision Variables:

$$y_n = \begin{cases} 1 & \text{If inspection is allocated at stage } n; \\ 0 & \textit{Otherwise.} \end{cases}$$

Auxiliary Continuous Variables:

g_n Number of good items leaving stage n;

d_n Number of defective items leaving stage n if inspection is not conducted;

3.3.2. MILP Formulation

Minimize:

$$Z = \left(\sum_{n=1}^{N} I_n \cdot y_n\right) + \left(\sum_{n=2}^{N} P_n \cdot d_{n-1}\right) + C \cdot d_N \tag{3.1}$$

Subject to:

$$g_1 = (1 - \Theta_1) \cdot b \tag{3.2}$$

$$g_n = (1 - \Theta_n) \cdot g_{n-1}; \quad \forall (n > 1) \tag{3.3}$$

$$d_1 \leq \Theta_1 \cdot b + M \cdot y_1 \tag{3.4}$$

$$d_1 \geq \Theta_1 \cdot b - M \cdot y_1 \tag{3.5}$$

$$d_n \leq d_{n-1} + \Theta_n \cdot g_{n-1} + M \cdot y_n; \quad \forall (n > 1) \tag{3.6}$$

$$d_n \geq d_{n-1} + \Theta_n \cdot g_{n-1} - M \cdot y_n; \quad \forall (n > 1) \tag{3.7}$$

$$g_N + d_N = D \tag{3.8}$$

$$b \geq 0; \quad g_n \geq 0; \text{ and } d_n \geq 0 \tag{3.9}$$

$$y_n \in \{0, 1\} \tag{3.10}$$

The objective function in Eq. (3.1) represents the minimization of the sum of (i) the inspection cost, (ii) the cost of wasted effort in processing items that are already defective, and (iii) the penalty cost of defective item reaching the customer. The constraints in Eqs. (3.2) and (3.3) calculate the number of good items, g_n, leaving stages $n = 1$ and $n > 1$, respectively. The number of defective items d_1 leaving stage-1 is set to $\Theta_1 \cdot b$ by the constraints in Eqs. (3.4) and (3.5) if $y_1 = 0$. The value of d_1 will be forced to zero by the objective function if $y_1 = 1$. Similarly, the number of defective items, d_n, leaving stage $n > 1$ are determined by the constraints in Eq. (3.6), Eq. (3.7) and the objective function. The constraint in Eq. (3.8) equates the number of items leaving stage-N to that of the customer requirement D. Non-negativity and binary constraints are in Eq. (3.9) and Eq. (3.10), respectively.

3.4. Model for Sequence-Dependent FSP

A limited number of articles deal with flow-shop scheduling problems with sequence-dependent setup time Luh et al. (1998). Adding sequence-dependent set up time to any scheduling problem can increase its difficulty. According to many studies, one workstation scheduling problem with sequence-dependent setup time is considered an NP-hard problem Anghinolfi and Paolucci (2009). The authors note that to improve the production line performance, it is important to manage the sequence dependent setup time Anghinolfi and Paolucci (2009). Additionally, they note that around 70% of production scheduling employees had to deal with sequence-dependent set up time. In 1975, Ergin Uskup and Spencer B.Smith developed an algorithm to find the optimum solution for two stage production line with sequence dependent setup time Uskup and Smith (1975). In this thesis, scheduling with sequence-dependent setups has been applied on a flow-shop manufacturing system. However, in the future we can apply this concept with flexible flow shop, job shop, and flexible job shop. In this sense, in 2009, Fantahun Defersha and Mingyuan Chen developed a model to solve a flexible job-shop scheduling problem incorporating sequence-dependent setup time by using

GA Defersha et al. (2009). Additionally, a comprehensive study conducted by Ali Allahverdi et al. in 2008 demonstrated some examples related to sequence-dependent set up times in flow-shop scheduling problems. The authors explored many types of shop environments such as open shop, no-wit flow shop, parallel machine, single machine, and job shop Allahverdi et al. (2008). As we noted previously, machines require cleaning or reset between jobs. If the length of the setup depends on the job just completed and on the one about to be started, then the setup times are sequence dependent. If job "j" is the first job on machine at stage "n", then the setup time is $S_{j,n}$. If job j+1 is followed by job j on machine at stage n then the setup time is $S_{j,n,j'}$. To understand this concept, consider a yarn production line with multiple coolers. To produce different types of colors, the changeover is required many times during the operation. Using any new color, all machines in the production line need to be cleaned. Measuring the cleanup time is based on the previous color and the color about to be used. Processing light colors first followed by dark color is one of the best ways to reduce waste in yarn manufacturing because the light colors cleanup process is faster than the dark colors. Considering many valuables such as labor cost, machine time, waste of raw material, and setup-time are important factors to improve any scheduling problem. In the following model, there are extra notations and variables such as $S_{j,n}$, $S_{j,n,j'}$, and $e_{j,n}$.

3.4.1. Notations

Problem Parameters:

J Number of jobs where jobs are indexed as $j = 1, 2, \cdots J$;

D_j Requirement (demand or shipment size) for job j;

N Number of stages in the flow line where stages are indexed as $n = 1, 2, \cdots, N$;

R Number of production runs on a stage where runs are indexed as $r = 1,\ 2,\ \cdots,\ J$;

$T_{j,n}$ Processing time per unit for job j at stage n;

$S_{j,n}$ Setup time for the processing job j at Stage-n if it is the first job processed at this stage;

$S_{j,n,j'}$ Setup time for the processing job j at Stage-n if the processing of job j immediately follows job j'

M Large positive number;

Covinous Variables:

$e_{j,n}$ The completion time of the processing (machining) of job j at Stage n.

$v_{r,n}$ The completion time of the r^{th} processing (machining) operation at Stage n.

Binary Decision Variables:

$$p_{j,r,n} = \begin{cases} 1 & \text{If job } j \text{ is the } r^{th} \text{ job processed by the} \\ & \text{machine at stage } n; \\ 0 & \text{Otherwise.} \end{cases}$$

$$Z = c_{max} \tag{3.11}$$

$$c_{max} \geq e_{j,N};\ \forall j \tag{3.12}$$

3.4.1.1 Constraints

In this section, the various constraints of the proposed model are presented. These constraints are categorized as (i) Constraints Relating Finish Times, (ii) Capacity

Constraints, (iii) Sequencing Constraints, and (iv) Logical Constraints on binary variables.

Constraints Relating Finish Times

Eq. (3.11) is the objective function which focus on minimize the makespan. The makespan, c_{max}, of the schedule is greater than all the completion times of the operations of jobs by the last stage ($e_{j,N}$) as enforced by the constraint Eq. (3.12). The constraints in Eqs. (3.13) and (3.14) set $e_{j,n} = v_{r,n}$, if job j is processed by the r^{th} run of stage n (or in other words if $p_{j,r,n} = 1$).

$$e_{j,n} \leq v_{r,n} + M(1 - p_{j,r,n});\ \forall(j, r, n) \tag{3.13}$$

$$e_{j,n} \geq v_{r,n} - M(1 - p_{j,r,n});\ \forall(j, r, n) \tag{3.14}$$

Capacity Constraints

The constraints in Eqs. (3.15) and (3.17) are capacity constraints on the resources (the processing stages). These constraints are needed because a resource can handle only one job at a time and it cannot be made available before $t = 0$. The constraint in Eq. (3.15) state that the starting time of the first run ($r = 1$) of stage-1 cannot be less than zero. The starting time of run $r > 1$ of a processing stage has to be greater or equal to the finish time of its previous run (run $r - 1$). This capacity constraint is enforced by Eq. (3.16) for stage-1 and by Eq. (3.17) and for stage-$n > 1$.

$$v_{1,1} - S_{j,1} - T_{j,1} \cdot D_j + M(1 - p_{j,1,1}) \geq 0;\quad \forall j \tag{3.15}$$

$$v_{r,1} - S_{j,1,j'} - T_{j,1} \cdot D_j + M(2 - p_{j,r,1} - p_{j',r-1,1}) \geq v_{r-1,1};$$
$$\forall (j, j', r) | r > 1, \ \& \ j \neq j' \tag{3.16}$$

$$v_{r,n} - S_{j,n,j'} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) +$$
$$M(2 - p_{j,r,n} - p_{j',r-1,n}) \geq v_{r-1,n};$$
$$\forall (j, j', r, n) | r > 1 \ \& \ n > 1 \ \& \ j \neq j' \tag{3.17}$$

Sequencing Constraints

The constraint in Eq. (3.18) states that the processing of job j on stage n cannot begin before its processing in stage $n-1$ is completed. This constraint is applicable for the first run ($r = 1$) of stage-n. The constraint in Eq. (3.19) is similar to that in Eq. (3.18) except that Eq. (3.19) is applicable for run $r > 1$ of stage-n.

$$v_{1,n} - S_{j,n} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) +$$
$$M(2 + y_{j,n-1} - p_{j,1,n} - p_{j,r',n-1}) \geq v_{r',n-1};$$
$$\forall (j, n, r') | n > 1 \tag{3.18}$$

$$v_{r,n} - S_{j,n,j'} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) +$$
$$M(3 + y_{j,n-1} - p_{j',r-1,n} - p_{j,r,n} - p_{j,r',n-1}) \geq v_{r',n-1};$$
$$\forall (j, j', n, r, r') | n > 1, \ r > 1, \ \& j \neq j' \tag{3.19}$$

Logical Constraints

A job has to be assigned to exactly one production run of a stage (Eq. (3.20)) and a prodcution run of a stage can be assigned to exactly one job (Eq. (3.21)).Non-negativity and binary constraints are in Eq. (3.22) and Eq. (3.23), respectively.

$$\sum_{r=1}^{J} p_{j,r,n} = 1;\ \forall (j,n) \tag{3.20}$$

$$\sum_{j=1}^{J} p_{j,r,n} = 1;\ \forall (r,n) \tag{3.21}$$

$$b_j,\ d_j,\ e_{j,n},\ v_{r,n} \geq 0;\quad \forall (j,r,n) \tag{3.22}$$

$$p_{j,r,n} \in \{0,1\};\quad \forall (j,r,n) \tag{3.23}$$

3.5. Integrated Inspection and Scheduling Model

Many inspection allocation models for multi-stage manufacturing systems have been proposed in the literature with great degrees of complexity. Integrating these complex models with a scheduling problem (which is complex in itself) is a challenge. In this section, we propose a simplified model, which we believe can be easily integrated with multi-stage discrete manufacturing systems with scheduling problems such as in flow shop, job shop, and their variants.

3.5.1. Additional Notations

Problem Parameters:

J Number of jobs where jobs are indexed as $j = 1,\ 2,\ \cdots J$;

D_j Requirement (demand or shipment size) for job j;

N Number of stages in the flow line where stages are indexed as $n = 1,\ 2,\ \cdots,\ N$;

Q Number of identical quality inspection stations located nearby the production line where inspection stations are indexed as $q = 1,\ 2,\ \cdots,\ Q$;

K Potential number of inspection on an inspection stations (K should be set to reasonable number);

$\Theta_{j,n}$ Defect rate of job j at stage n;

$I_{j,n}$ Inspection cost of job j at stage n;

$P_{j,n}$ Penalty cost of the processing of a defective items of job j that passed through Stage $n-1$ and processed in Stage n if there is no inspection after stage $n-1$ for job j;

C_j Penalty cost of per defective item of job j reaching the customer.

$T_{j,n}$ Processing time per unit for job j at stage n;

$U_{j,n}$ Inspection time per unit for job j following its processing at stage n;

$S_{j,n}$ Setup time for the processing job j at Stage-n if it is the first job processed at this stage;

$S_{j,n,j'}$ Setup time for the processing job j at Stage-n if the processing of job j immediately follows job j' at this stage;

M Large positive number;

Binary Decision Variables:

$$p_{j,r,n} = \begin{cases} 1 & \text{If job } j \text{ is the } r^{th} \text{ job processed by the} \\ & \text{machine at stage } n; \\ 0 & \text{Otherwise.} \end{cases}$$

$$y_{j,n} = \begin{cases} 1 & \text{If job } j \text{ is inpected by one of the inspection} \\ & \text{stations imidiately after being processed at} \\ & \text{stage } n; \\ 0 & \text{Otherwise.} \end{cases}$$

$$x_{j,k,q,n} = \begin{cases} 1 & \text{If job } j \text{ is the } k^{th} \text{ job inpected at station} \\ & q \text{ after being processed at stage } n \\ 0 & \text{Otherwise.} \end{cases}$$

$$z_{k,q} = \begin{cases} 1 & \text{If the } k^{th} \text{ inpection is conducted at station } q; \\ 0 & \text{Otherwise.} \end{cases}$$

Continuous Decision Variables:

b_j The batch size of job j needed to deliver a total of D_j units at the end of the production line ($b_j > D_j$ as some units will be rejected along the production line if they do not meet quality characteristics).

$e_{j,n}$ The completion time of the processing (machining) of job j at Stage n.

$v_{r,n}$ The completion time of the r^{th} processing (machining) operation at Stage n.

$f_{j,n}$ The completion time of the inspection of job j following its processing at stage n.

$w_{k,q}$ The completion time of the k^{th} inspection of stations q.

$g_{j,n}$ Number of good items of job j leaving stage n;

$d_{j,n}$ Number of defective items of job j leaving stage n if inspection is not conducted using one of the inspection stations;

3.5.2. Formulation

3.5.2.1 Objective Function

The formulation presented in this section is a multi-objective mixed-integer linear programming (MILP) where the weighted sum of the costs of inspection policy, z_1, and makespan of the schedule , z_2, is to be minimized as shown in equation Eq. (3.24). The weight factors, W_1 and W_2, are set by decision-makers to reflect the relative importance of the two objective function terms. The values of z_1 and z_2 are set by the equations in Eqs. (3.25) and (3.26), respectively.

$$z = W_1 \cdot z_1 + W_2 \cdot z_2 \tag{3.24}$$

$$\begin{aligned} z_1 = & \sum_{j=1}^{J} \sum_{n=1}^{N} I_{j,n} \cdot y_{j,n} + \sum_{j=1}^{J} \sum_{n=2}^{N} P_{j,n} \cdot d_{j,n-1} \\ & + \sum_{j=1}^{J} C_j \cdot d_{j,N} \end{aligned} \tag{3.25}$$

$$Z_2 = c_{max} \tag{3.26}$$

3.5.2.2 Constraints

There are many common constraints from the previous model 3.4.1. However, additional constraints related to inspection allocation in this model.

Inspection Allocation Constraints

The constraints in Eqs. (3.27) and (3.33) are inspection allocation constraints. These constraints are similar to those discussed in Eqs. (3.2) and (3.8) with the exception that those in Eqs. (3.27) and (3.33) are for the case in which multiple jobs (products) are considered.

$$g_{j,1} = (1 - \Theta_{j,1}) \cdot b_j;\ \forall j \tag{3.27}$$

$$g_{j,n} = (1 - \Theta_{j,n}) \cdot g_{j,n-1};\quad \forall (j,n)|n > 1) \tag{3.28}$$

$$d_{j,1} \leq \Theta_{j,1} \cdot b_j + M \cdot y_{j,1};\ \forall j \tag{3.29}$$

$$d_{j,1} \geq \Theta_{j,1} \cdot b_j - M \cdot y_{j,1};\ \forall j \tag{3.30}$$

$$d_{j,n} \leq d_{j,n-1} + \Theta_{j,n} \cdot g_{j,n-1} + M \cdot y_{j,n}; \forall (j,n)|n > 1) \tag{3.31}$$

$$d_{j,n} \geq d_{j,n-1} + \Theta_{j,n} \cdot g_{j,n-1} - M \cdot y_{j,n}; \forall (j,n)|n > 1) \tag{3.32}$$

$$g_{j,N} + d_{j,N} = R_j;\ \forall j \tag{3.33}$$

Constraints Relating Finish Times

The constraints in Eqs. (3.34) and (3.39) interrelates the makespan and the various completion times ($c_{max},\ e_{j,n},\ f_{j,n},\ w_{k,q}$). In this model, the makespan, c_{max}, of the schedule is greater than all the completion times of the operations of jobs by the last stage ($e_{j,N}$) and the completion times of all the inspections that might have occurred after jobs are processed by the last stage ($f_{j,N}$) as enforced by the constraints in Eqs. (3.34) and (3.35), respectively. Along with these constraints, the objective function will enforce the value of c_{max} to take the value of the largest completion time, which is to be minimized through the optimization process. The constraints in Eqs. (3.36) and (3.37) are similar to constraints in Eqs. (3.13) and (3.14). If job j is allocated inspection after it completes processing on stage n and if its inspection is the k_{th} inspection at inspection station q (i.e., $x_{j,k,q,n} = 1$), then $w_{k,q} = f_{j,n}$ as dectated by the constraints in Eqs. (3.38) and (3.39).

$$c_{max} \geq e_{j,N};\ \forall j \tag{3.34}$$

$$c_{max} \geq f_{j,N};\ \ \forall (j,q);\ \forall (j,q) \tag{3.35}$$

$$e_{j,n} \leq v_{r,n} + M(1 - p_{j,r,n});\ \forall (j,r,n) \tag{3.36}$$

$$e_{j,n} \geq v_{r,n} - M(1 - p_{j,r,n});\ \forall (j,r,n) \tag{3.37}$$

$$w_{k,q} \leq f_{j,n} + M(1 - x_{j,k,q,n});\ \forall (j,k,q,n) \tag{3.38}$$

$$w_{k,q} \geq f_{j,n} - M(1 - x_{j,k,q,n});\ \forall (j,k,q,n) \tag{3.39}$$

Capacity Constraints

The constraints Eqs. (3.15), (3.16) and (3.17) are similar to Eqs. (3.40), (3.41) and (3.44), respectively. The capacity constraint on inspection stations are in Eqs. (3.43) and (3.44). These constraints prohibit the overlapping of the $(k-1)^{th}$ and the k^{th} inspections of inspection station q. Eq. (3.43) is applicable if the k^{th} inspection is for a job that comes from stage-1, whereas Eq. (3.44) is for a job that comes from stage-$n > 1$.

$$v_{1,1} - S_{j,1} - T_{j,1} \cdot b_j + M(1 - p_{j,1,1}) \geq 0;\ \ \forall j \tag{3.40}$$

$$v_{r,1} - S_{j,1,j'} - T_{j,1} \cdot b_j + M(2 - p_{j,r,1} - p_{j',r-1,1}) \geq v_{r-1,1};$$
$$\forall (j, j', r) | r > 1, \ \& \ j \neq j' \quad (3.41)$$

$$v_{r,n} - S_{j,n,j'} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) + M(2 - p_{j,r,n} - p_{j',r-1,n}) \geq v_{r-1,n};$$
$$\forall (j, j', r, n) | r > 1 \ \& \ n > 1 \ \& \ j \neq j' \quad (3.42)$$

$$w_{k,q} - U_{j,1} \cdot b_j + M(1 - x_{j,k,q,1}) \geq w_{k-1,q}; \forall (j, k, q) | (k > 1) \quad (3.43)$$

$$w_{k,q} - U_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) + M(1 - x_{j,k,q,n}) \geq w_{k-1,q};$$
$$\forall (j, n, k, q) | (n > 1) \& (k > 1) \quad (3.44)$$

Sequencing Constraints

The constraints in Eqs. (3.45) and (3.50) are sequencing constraints. These constraints are needed because consecutive activities on a given job cannot be overlapped. If inspection is not allocated, the completion time of an operation of a job on give stage should be smaller than the starting time of the operation of the same job in the next stage. If inspection is allocated for a job after its processing in a given stage, the completion time of the job in that stage should be less than the start time of the inspection process on the selected inspection station. Moreover, the finish time of the inspection should be less than the start time of the operation of the same job on the next stage. The constraint in Eq. (3.45) states that the processing of job j on stage n cannot begin before its processing in stage $n-1$ is completed if inspection is not allocated for the job after its processing in stage $n-1$ (i.e., $y_{j,n-1} = 0$). This constraint is applicable for the first run ($r = 1$) of stage-n. The constraint in Eq. (3.46) is similar to that in Eq. (3.45) except that Eq. (3.46) is applicable for run $r > 1$ of stage-n. If an inspection is allocated for a job after being processed at stage-n, its upcoming inspection cannot begin before its operation is completed on stage-n as enforced by Eq. (3.47) for $n = 1$ and by Eq. (3.48) for $n > 1$. The constraints Eqs. (3.49) and (3.50) are post-inspection sequencing constraint. These constraints guarantee that an upcoming operation of a job cannot begin before inspection is completed, if inspection was allocated for the previous operation of the job.

$$v_{1,n} - S_{j,n} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) + M(2 + y_{j,n-1} - p_{j,1,n} - p_{j,r',n-1}) \geq v_{r',n-1}; \quad \forall (j, n, r') | n > 1 \tag{3.45}$$

$$v_{r,n} - S_{j,n,j'} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) + M(3 + y_{j,n-1} - p_{j',r-1,n} - p_{j,r,n} - p_{j,r',n-1}) \geq v_{r',n-1}; \quad \forall (j, j', n, r, r') | n > 1,\ r > 1,\ \& j \neq j' \tag{3.46}$$

$$w_{k,q} - U_{j,1} \cdot b_j + M(2 - x_{j,k,q,1} - p_{j,r,1}) \geq v_{r,1}; \forall (j, r, q, k) \tag{3.47}$$

$$w_{k,q} - U_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) + M(2 - x_{j,k,q,n} - p_{j,r,n}) \geq v_{r,n}; \quad \forall (j, n, r, q, k) | n > 1 \tag{3.48}$$

$$v_{1,n} - S_{j,n} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) + M(2 - x_{j,k,q,n-1} - p_{j,1,n}) \geq w_{k,q}; \quad \forall (j, n, k, q) | n > 1 \tag{3.49}$$

$$v_{r,n} - S_{j,n,j'} - T_{j,n} \cdot (g_{j,n-1} + d_{j,n-1}) + \\ M(3 - x_{j,k,q,n-1} - p_{j',r-1,n} - p_{j,r,n}) \geq w_{k,q}; \\ \forall (j, j', n, r, k, q) | n > 1 \ \& \ r > 1 \quad (3.50)$$

Logical Constraints

The logical constraints on the various integer variables are in Eqs. (3.51) to (3.55). The constraints in Eqs. (3.20) and (3.21) similar to Eqs. (3.51) and (3.52). The constraint in Eq. (3.53), states that if an inspection is allocated to job j after its processing on stage-n (i.e., $y_{j,n} = 1$), its inspection has to be assigned to exactly one inspection run on one of the inspection stations ($\sum_{q=1}^{Q} \sum_{k=1}^{K} x_{j,k,q,n} = 1$). If the k^{th} inspection run of inspection station occurs ($z_{k,q} = 1$), it should be for exactly one job ($\sum_{j=1}^{J} \sum_{j=1}^{J} x_{j,k,q,n} = 1$) as stated by Eq. (3.54). The constraint in Eq. (3.55) states that k^{th} inspection run by inspection station-q can occur if its $(k-1)^{th}$ inspection run has occurred. Nonnegativity and binary constraints are in Eqs. (3.56) and (3.57), respectively.

$$\sum_{r=1}^{J} p_{j,r,n} = 1; \ \forall (j, n) \quad (3.51)$$

$$\sum_{j=1}^{J} p_{j,r,n} = 1; \ \forall (r, n) \quad (3.52)$$

$$\sum_{q=1}^{Q} \sum_{k=1}^{K} x_{j,k,q,n} = y_{j,n}; \ \forall (j, n) \quad (3.53)$$

$$\sum_{j=1}^{J} \sum_{n=1}^{N} x_{j,k,q,n} = z_{k,q; \ \forall (k,q)} \quad (3.54)$$

$$z_{k,q} \leq z_{k-1,q}; \ \forall (k, q) | k > 1 \quad (3.55)$$

$$b_j,\ d_j,\ e_{j,n},\ f_{j,n},\ v_{r,n},\ w_{k,q} \geq 0;$$
$$\forall (j,r,n,k,q) \tag{3.56}$$

$$y_{j,n}, p_{j,r,n},\ x_{j,k,q,n},\ z_{k,q} \in \{0,1\};$$
$$\forall (j,r,n,k,q) \tag{3.57}$$

3.6. Chapter Summary

This chapter presents three models that deal with flow-shop system. The three models are inspection allocation model, sequence-dependent model, and integrated inspection and scheduling model. By using the equations in this chapter, we can easily calculate the number of detectives items for each job during the production and the total makespan. Therefore, analyzing the solution is immediate, and no simulation is needed. The equations also allow us to study or analyze the behavior of the system, i.e.,adding more inspection machines to the system. Moreover, based on the operation cost and other factors, production management then can take the right decisions.

Chapter 4

Solution Procedure

This chapter starts with an introduction and then covers the basics of GA. Then we discuss the Solution representation and initialization. Section 4.3.3 explains the evaluation step. Section 4.3.4 discussed multi-objective fitness function. In Section 4.3.5, the genetic operators are explained, and it includes mutation, crossover, and selection.

4.1. Introduction

Several mathematical techniques are commonly used in scheduling. However, these mathematical formulations are very complex and time-consuming to implement especially, in multi objective problems. One of the reasons why these techniques are not perfect tool to deal with complex scheduling problems is due to the difficulty to find an optimum solution. In many cases of scheduling problems such as FSSP, HFSP, and FFSSP, many alternative solutions are available. Recently, there has been an increased interest in the use of simulation for scheduling Drake et al. (1995). In the case of simulation techniques, any model is generated based on assumptions and observations to represent of the behavior of the real system. It can provide valuable details of the manufacturing systems, limited only by cost and time. However, for complex systems scheduling problems, simulation softwares can be costly. Besides, validating any complex model can be challenging. Additionally, scheduling problems are known to be NP-hard combinatorial problems. The heuristics can solve difficult problems by reducing the number of evaluations and getting solutions in short periods. On the other hand, it is difficult to evaluate the performance of the solution Foulds (1983). Furthermore, while heuristics, can be efficient for some problems, an optimal solution is not guaranteed. Due to the above factors, optimization for flow shop, flexible flow shop, job shop, and flexible job shop is a complex task. Therefore, more powerful search techniques are required in the future. Using high-level optimization tools such as metaheuristics, can lead to better results in very short periods compared with other optimization methods. Additionally, the literature has discussed many of meta-heuristics that deal with flow-shop scheduling problems.

4.2. The Proposed Genetic Algorithm

As discussed previously, GA is one of the best optimization tools to solve NP-hard problems. For example, GA is used to solve the traveling salesman problem (TSP). Some of the goals of solving this problem using GA is to create an effective plan. According to that plan we can easily optimize the total time and cost of each travel. Furthermore, GA is useful in many fields such as finances, design, and DNA analysis. Another important point is that, solving-multi stages flow-shop scheduling problems with inspection allocation by using optimization packages such as LINGO is extremely challenging or impossible in many cases. In this regard, a study found that LINGO software is not able to find optimum solution even after more than 20 hours Chung et al. (2011). Another article noted that solving a problem by using LINGO may take more than 112 hours. Conversely, algorithms can produce the optimal solutions for the same problem in less than 12 seconds Chang et al. (2013). Considering the multiple workstations and inspection stages that need to be optimized in this thesis, it is important to select an efficient optimization technique that can provide a solution in short period of time.

4.2.1. Solution Representation

Solution representation is the first and the most important step in developing a GA. It is a technique that maps the problem to a searchable solution space. Its design impacts the way initialization, evaluation, cross-over, and mutation are implemented. The solution representation adopted to solve the proposed problem is shown in Figure 4.1. As it can be seen from this figure, the representation has two segments, namely, the right-hand-side (RHS) segment and the left-hand-side (LHS) segment. The RHS segment has subsegments corresponding to each job. Each subsegment is a binary vector of size equal to the number of processing stages. The binary vector of a subsegment determines the inspection allocation for the job associated to it. For instance, in Figure 4.1, the details of the subsegment-J4 are shown. This subsegment representees the inspection allocation of job-4. The LHS segment is simply the permutation of the jobs to be scheduled on the system, and it represents the sequence by which the jobs are presented to the first stage. The sequence of processing in the other stages also largely depends on the LHS segment. The actual sequence in the remaining stages is determined by the decoding procedure that takes into account the information obtained from both the LHS and the RHS segments.

4.2.2. Initialization

As stated in the previous section, the LHS segment represents the permutation of the jobs by which they are admitted to the first stage. Any permutation of

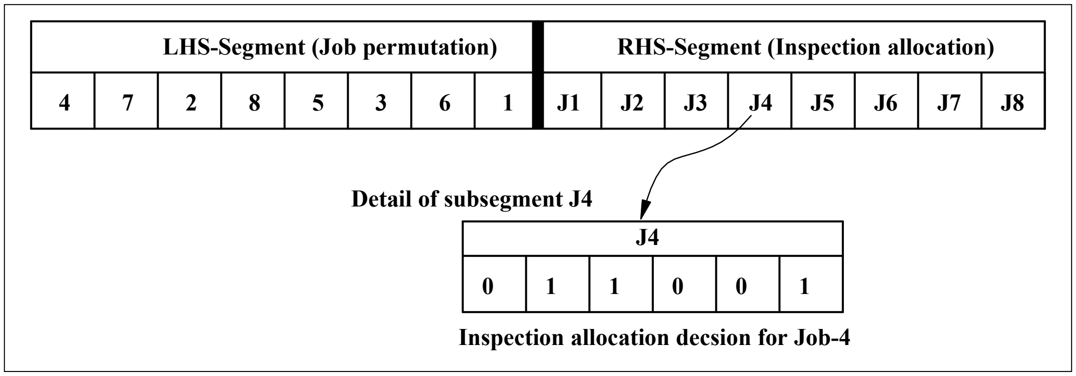

Figure 4.1: Solution representation assuming eight jobs and six manufacturing stages.

jobs can be a candidate solution in flow-shop scheduling with sequence-dependent setup time and makespan criterion. Hence, the LHS segment of a solution has to be initialized by a randomly generated permutation of the jobs. The RHS segment can also be initialized by a random process. However, its initialization must avoid solutions with excessive or too little inspection allocations. Hence, we suggest a parameter $\alpha \in (0, 1)$ that can be set to control the level of inspection allocation in such a way that a lower or a higher value of α corresponds to a fewer or greater inspection allocation, respectively. This can be achieved by initializing the binary values of the RHS segment with a function that takes α as its argument and returns zero or one, as shown in Eq. (4.1). In this equation, RHS.Segment$[j]$.Gene$[n]$ represents the inspection allocation of job j after its processing by stage n, whereas $rand()$ represents a uniform random number generator (between 0 and 1).

$$\text{RHS.Segment}[j].\text{Gene}[n] = \begin{cases} 1 & \text{If } \alpha < rand() \\ 0 & \text{Otherwis} \end{cases} \tag{4.1}$$

4.2.3. Solution Decoding and Fitness Evaluation

4.2.3.1 Decoding $y_{j,n}$, b_j, $g_{j,n}$, $d_{j,n}$, and Z_1

Decoding $y_{j,n}$ is a straight forward process as its value can be directly read from the solution representation. For instance, from the detail of the subsegment J4 in Figure 4.1, $y_{4,1} = 0$, $y_{4,2} = 1$, $y_{4,3} = 1$, $y_{4,4} = 0$, $y_{4,5} = 0$, and $y_{4,5} = 1$. The values of $y_{j,n}$ corresponding to the other jobs can be obtained in a similar fashion.

Given $y_{j,n}$, the determination of the initial batch size b_j, needed to meet the demand R_j, requires knowing the values of $g_{j,n}$ and $d_{j,n}$ using a backward recursive calculation from the last stage to the first stage, which is difficult to accomplish. On the other hand, if b_j was given, $g_{j,n}$ and $d_{j,n}$ can easily be determined by a forward recursion using Eqs. (3.27) to (3.32). However, b_j is not a given parameter, instead it is variable. To alleviate this difficulty, we devise an alternative approach outlined in Algorithm-1. In describing this algorithm, let us first define $\hat{g}_{j,n}$ and $\hat{d}_{j,n}$ as the rate of good and defective items, respectively, leaving stage n per unit of job j supplied to the first stage under a given inspection allocation decision (i.e., given $y_{j,1},\ y_{j,2},\ \cdots,\ y_{j,N}$). The values of these new variables can be determined using a forward recursive calculation as shown from line-2 to line-6. Then, the corresponding rate of the final output $\hat{R}_j$ per unit input of job j can be calculated as $\hat{g}_{j,N} + \hat{d}_{j,N}$ (see line-8). Once $\hat{R}_j$, $\hat{g}_{j,n}$, $\hat{d}_{j,n}$ are determined, the values for b_j, $g_{j,n}$, and $d_{j,n}$ can be calculated as $b_j = R_j/\hat{R}_j$, $g_{j,n} = \hat{g}_{j,n} \times (R_j/\hat{R}_j)$ and $d_{j,n} = \hat{d}_{j,n} \times (R_j/\hat{R}_j)$ as shown from lines 10 through 16 of Algorithm-1. Those calculations are valid because of the linear relationship of the variable in the constraints from Eqs. (3.27) to (3.32). Having determined all the variables related to the inspection allocation ($y_{j,n}$, b_j, $g_{j,n}$ and $d_{j,n}$), the inspection policy cost Z_1 can be recursively calculated as shown in the for-loop in line-18 to line-24.

4.2.3.**2** **Decoding completion times and makespan** z_2

Once all the variables related to inspection allocation are determined using Algorithm-1, the various completion times ($e_{j,n},\ f_{j,n},\ v_{r,n},\ w_{k,n}$) are determined by Algorithm-2 along with other two algorithms (Algorithms-3 and -4). These algorithms determine the completion times and the makespan by starting the manufacturing operations and the allocated inspections of the jobs as soon as possible (without intentional delay) while at the same time respecting the capacity and the sequencing constraints. In describing these algorithms, we first introduce additional notation given below.

Additional Notation

k_q An inspection run counter which increases by one every time an inspection of a job is completed on inspection station q.

$\gamma_{j,n}$ The time at which job j is ready for its processing on stage n. The value of $\gamma_{j,n}$ is zero for $n = 1$ whereas its value for $n > 1$ is either the jobs competition time $e_{j,n-1}$ in stage $n-1$ (if no inspection after stage $n-1$) or the competition time of its inspection $f_{j,n-1}$ (if the job is inspected after its processing in stage $n-1$).

ψ_n The sequence (ordered list of jobs) by with the jobs are presented to stage n. For $n = 1$, ψ_n is obtained from the LHS-Segment of the solution representation, whereas for the remaining stages ψ_n is obtained by sorting the jobs based on non-decreasing order of their ready time $\gamma_{j,n}$.

$\psi_n(r)$ The index of the job at location r of the ordered list ψ_n.

Given the above notations, the algorithms are described step by step hereunder.

- The run counter k_q is initialized (Algorithm-2, line-1)
- The order of processing of the jobs on stage-1, ψ_1, is obtained from the LHS-Segment of the solution under evaluation (Algorithm-2, line-2).
- The job index at location r of the list ψ_n is obtained (Algorithm-2, line-5).
- The completion time of job j on the r^{th} run of stage-n is calculated using Algorithm-3 under one of the four different cases given below.

  1. $n = 1$ and $r = 1$

     In this case, job j is ready for its first operation at time zero ($\gamma_{j,n} = 0$) and the sage is also available at time zero for its firs run ($r = 1$). Hence, the first run starts at time zero and completes at $v_{r,n} = S^*_{j,n} + (T_{j,n} \times b_j)$. Note that the size of the batch of job j that will be processed on stage $n = 1$ is equal to b_j, the initial batch size.

  2. $n = 1$ and $r > 1$

     Similar to (1),$\gamma_{j,n} = 0$. However, the stage is available at $v_{r-1,n}$ which is the completion time of its previous run. The jobs that was processed in run $r - 1$ is job $j' = \psi_n(r - 1)$, which is used to obtain the value of the sequence dependent setup time $S_{j,n,j'}$. Hence, run r starts at $v_{r-1,n}$ and completes at $v_{r,n} = v_{r-1,n} + S_{j,n,j'} + T_{j,n} \times b_j$.

  3. $n > 1$ and $r = 1$

     In this case, $\gamma_{j,n} = f_{n-1,j}$ if $y_{j,n-1} = 1$ or at $\gamma_{j,n} = e_{n-1,j}$ if $y_{j,n-1} = 0$. Similar to (1), the stage is available at time zero for its first run ($r = 1$). Hence, processing can begin as soon as the job is available. The completion time of the run will then be $v_{r,n} = f_{n-1,j} + S^*_{j,n} + T_{j,n} \times g_{j,n-1}$ if $y_{j,n-1} = 1$ or $v_{r,n} = e_{n-1,j} + S^*_{j,n} + T_{j,n} \times (g_{j,n-1} + d_{j,n-1})$ if $y_{j,n-1} = 0$. Note that the size of the batch of job j that will be processed on stage $n > 1$ is equal to $g_{j,n-1}$ if $y_{j,n-1} = 1$ or $g_{j,n-1} + d_{j,n-1}$ if $y_{j,n-1} = 0$.

4. $n > 1$ and $r > 1$

   In this case, $\gamma_{j,n}$ is similar to (3) and the availability of the stage is similar to (2). Run r can start at the job ready time or at the completion time of run $r-1$, whichever larger. Therefore, the completion time of run r is equal to $\max\{f_{n-1,j}, v_{r-1,n}\} + S_{j,n,n'} + T_{j,n} \times g_{j,n-1}$ if $y_{j,n} = 1$ or $\max\{e_{n-1,j}, v_{r-1,n}\} + S_{j,n,j'} + T_{j,n} \times (g_{j,n-1} + d_{j,n-1})$ if $y_{j,n} = 0$. The index of the job that was processed in run $r-1$ is obtained in a similar fashion as (2).

- Once the completion time $v_{r,n}$ is computed, its value is assigned to the completion time $e_{j,n}$ as job j is processed on the r^{th} run of stage n (Algorithm-2, line-8).

- If inspection is allocated for job j after its processing on stage n (i.e., $y_{j,n} = 1$), Algorithm-2 (at line-11) calls Algorithm-4 to determine the completion time of inspection.

- Algorithm-4 first determines the inspection station that will be available soon using the for-loop from line-2 to line-12. When the for-loop breaks at line-4 or exits at line-12, the variable "*selected*" will assume the index of the selecting inspection station. This station is the first station with run counter $k_q = 1$ (i.e., the first station available at time zero if any) or the one with the smallest completion time $w_{k_q-1,q}$ of its $(k_q - 1)^{th}$ inspection activity.

- The value of the variable "*selected*" is assigned to an index q^* on line-13 of Algorithm-4. The algorithm then determines the finish time of the $k_{q^*}^{th}$ inspection activity of inspection station q^* under one of the following four cases.

  1. $n = 1$ and $k_{q^*} = 1$

In this case, the batch size of job j to be inspected is b_j. The inspection station is available at time zero for its first inspection activity (since $k_{q^*} = 1$). The inspection can be started as soon as the job's operation is finished (at $e_{j,n}$). Thus, the completion time of the inspection is $w_{k_{q^*},q^*} = e_{j,n} + U_{j,n} \times b_j$.

2. $n = 1$ and $k_{q^*} > 1$

   Similar to (1), the batch size of job j to be inspected is b_j. The inspection station is available after it finishes its $(k_{q*} - 1)^{th}$ inspection activity. Thus the inspection for job j can begin either at $e_{j,n}$ or $w_{k_{q^*}-1,q^*}$, whichever larger. Thus the completion time of k_{q*}^{th} inspection is $w_{k_{q^*},q^*} = \max\{e_{j,n}, w_{k_{q^*}-1,q^*}\} + U_{j,n} \times b_j$.

3. $n > 1$ and $k_{q^*} = 1$

   In this case, the batch size of job j to be inspected is $g_{j,n-1} + d_{j,n-1}$. Similar to (1), the inspection station is available at time zero. The inspection can be started as soon as the job's operation is finished (at $e_{j,n}$). Thus, the completion time of the inspection is $w_{k_{q^*},q^*} = e_{j,n} + U_{j,n} \times (g_{j,n-1} + d_{j,n-1})$

4. $n > 1$ and $k_{q^*} > 1$

   Similar to (3), the batch size of job j to be inspected is $g_{j,n-1} + d_{j,n-1}$. Similar to (2), the inspection station is available after it finishes its $(k_{q*} - 1)^{th}$ inspection activity. The inspection can be started at $e_{j,n}$ or $w_{k_{q^*}-1,q^*}$, whichever larger. Thus, the completion time of the inspection is $w_{k_{q^*},q^*} = \max\{e_{j,n}, w_{k_{q^*}-1,q^*}\} + U_{j,n} \times (g_{j,n-1} + d_{j,n-1})$.

- Algorithm-2 assigns the value of $w_{k_{q^*},q^*}$ to $f_{j,n}$ at line-12 as the $k_{q^*}^{th}$ inspection activity of station q^* is for job j following the job's processing on stage n. Then, the run counter k_{q^*} is increased by one on line-13.

- Job j's ready time for its processing on stage $n+1$ is determined on line-17 as $\gamma_{j,n+1} = e_{j,n} \times (1 - y_{j,n}) + f_{j,n} \times y_{j,n}$, which takes the value of $e_{j,n}$ if $y_{j,n} = 0$ or $f_{j,n}$ or $y_{j,n} = 1$.

- When Algorithm-2 exits the inner for-loop on line-19, ready time $\gamma_{j,n+1}$ is known for all the jobs. If $n < N$, the algorithm constitutes the order of processing of the jobs in the next stage (ψ_{n+1}) by sorting the jobs in a non-increasing order of their ready times (see lines 20 to 23).
- The outer for-loop of Algorithm-2 continuous until all the processing and inspection operations are assigned and their completion times are computer.
- The value of largest completion time is assigned to z_2 on line 25, which is the makespan of the integrated inspection allocation, inspection scheduling, and operation scheduling model.

4.2.3.3 Multi-objective fitness evaluation

In literature, there are several advanced methods for multi-objective optimization. However, as the primary objective of this thesis is merely to introduce an integrated approach in inspection allocation and scheduling, we adopted a weighted sum approach because of its simplicity and computational efficacy. In this approach, multiple objectives are aggregated into a single objective using weights. The weights represent the relative importance of the objective function terms set by decision-makers. Nevertheless, because of dissimilarities in measuring units of the objective function terms and their differences in magnitudes, scaling is needed to be consistent with the preferences of the decision makers. We used a scaling mechanism in such a way that, in the initial population of the GA, the magnitude of the maximum values of inspection policy cost z_1 will be the same as the maximum value of the makespan z_2. In describing this procedure, let us first define F_1 and F_2 as the scaling factors for z_1 and z_2, respectively. Moreover, let

$IniMax(z_1)$ and $IniMax(z_2)$ represent the maximum values of the two objective functions terms in the initial population of the GA. The factors F_1 and F_2 are then selected in such a way that $F_1 \times IniMax(z_1) = F_2 \times IniMax(z_2)$. In this scaling, without loss of generality, one can choose $F_1 = 1$ and $F_2 = \frac{IniMax(z_1)}{IniMax(z_2)}$. After scaling factors are set, the weight factors W_1 and W_2 can be selected by the decision maker only to reflect the relative importance of the two objective function terms. This scaling technique and the corresponding weighted objective function (fitness) can be mathematically represented ass shown in Eqs. (4.2), (4.3) and (4.4). The value of the weighted objective function in Eq. (4.4) can then be used as fitness function for the proposed genetic algorithm.

$$F_1 = 1 \tag{4.2}$$

$$F_2 = \frac{IniMax(z_1)}{IniMax(z_2)} \tag{4.3}$$

$$z = \{W_1 \cdot F_1 \cdot z_1\} + \{W_2 \cdot F_2 \cdot z_2\} \tag{4.4}$$

4.2.4. Selection Operator

The survival of the fittest is a phrase that originated from Darwinian evolutionary theory to describe the mechanism of natural selection. As per this theory, fitness is the reproductive success of an organism to leave the most copies of itself in successive generations. A genetic algorithm mimics this natural process using its selection operator. The operator can be applied in several different schemes. Among those schemes, K-ways tournament is the most frequently used scheme in the literature as it is easy to implement and adjust selection pressure to balance exploration and exploitation of the search space. Hence, it is also a choice of method in this thesis. In this selection mechanism, k individuals are randomly selected, and the one with the highest fitness is added to the next generation. The selection process continues (with replacement) until the required population size is reached. The tournament size k can be adjusted to regulate the selection pressure, where a smaller value corresponds to lesser selection pressure.

4.2.5. Crossover Operators

Crossover operators are the mechanism by which a genetic algorithm combines information contained in two solutions (parents) to create two other new solutions (offsprings). The operators need to be tailored to the solution representation adopted. Three different crossover operators were used in the proposed genetic algorithm (***SPCO1***, ***SPCO2*** and ***SPCO3***). These operators are illustrated in Figure 4.5. SPCO1 exchanges partial genetic material contained in the LHS segment between parent chromosomes. The operator ensures that there is no repeating or missing job in the LHS segment of the offspring. SPXO2 exchange partial information of the RHS segment, whereas SPXO3 exchange the complete RHS segment between parent chromosomes. In creating offsprings from a given pair of parents, only one of these operators will be arbitrarily selected and applied with certain probability. The steps involved in applying these operators are described hereunder.

SPCO1

1. Arbitrarily create a cross-over point on the LHS-segment of the parent chromosomes.

2. Create a fragmented chromosome from each parent by removing the part between the cross-over point and the junction point between LHS and RHS segments.

3. Complete the missing genes of a fragment chromosome from one parent with the order in which the genes appear on the second parent.

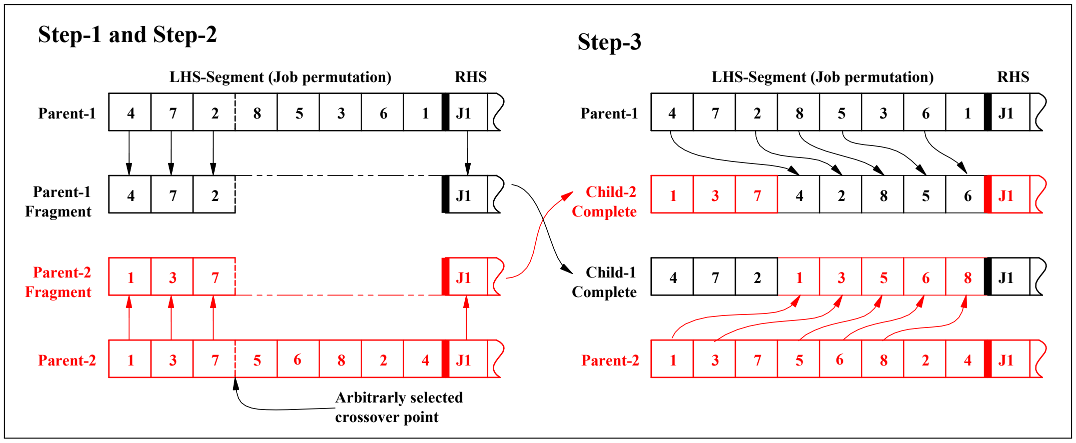

Figure 4.2: SPCO1

SPCO2

1. Arbitrarily create a cross-over point between any two sub-segments (J1, J2, ..., JN) of the RHS-segment of the parent chromosomes.

2. Create a fragmented chromosome from each parent by removing all the sub-segments of the RHS-segment to the right of the cross-over point.

3. Complete the missing sub-segments of a fragment chromosome from one parent with the sub-segments from the second parent.

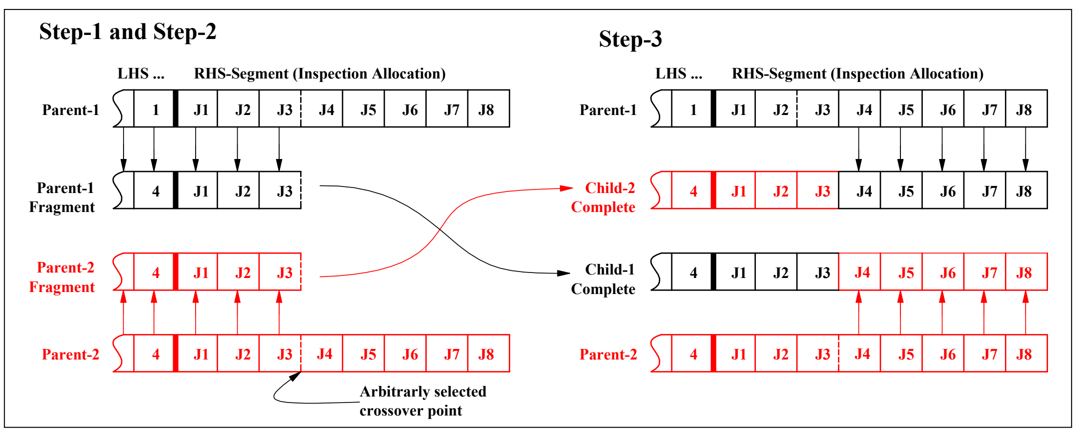

Figure 4.3: SPCO2

SPCO3

1. Set the cross-over point at the junction between the LHS and the RHS segments of the parent chromosomes.

2. Create a fragmented chromosome from each parent by removing the RHS segment.

3. Complete the fragment chromosome from one parent with the RHS segments from the second parent.

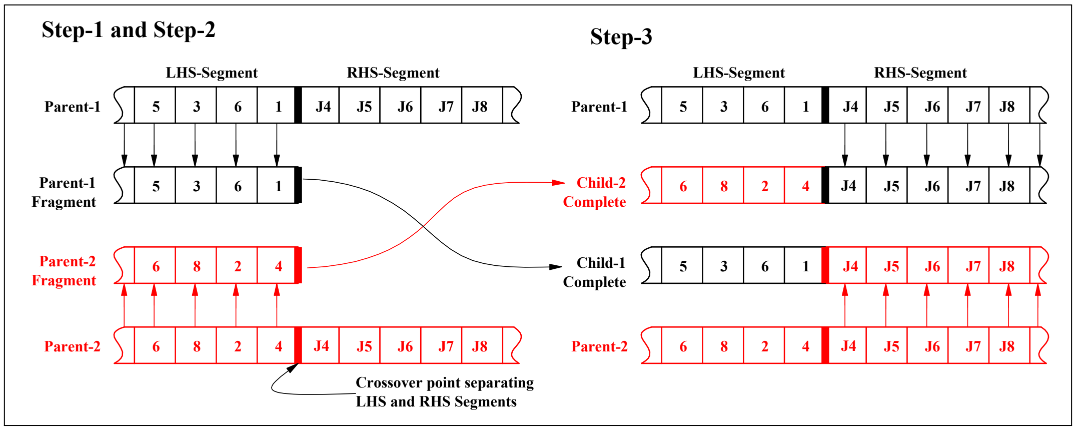

Figure 4.4: SPCO3

4.2.6. Mutation Operators

Mutation is a genetic operator aimed at introducing new genetic materials, restoring lost ones, and maintaining population diversity. Three mutation operators were used in the proposed genetic algorithm. We refer to these operators as Mutation Operator 1, 2 and 3 (***MuOp1***, ***MuOp2***, and ***MuOp3***). MuOp1 arbitrarily selects two locations on the LHS segment of an individual solution and swaps the genetic materials contained in these locations. MuOp2 arbitrarily selects one location on the LHS segment and shifts the genetic material to another arbitrarily selected location. MuOp3 is applied to each binary gene of the RHS segment and alters its value (from 0 to 1 or from 1 to 0). Each of these three operators is applied to a given solution with its set probability. The probability for MuOp3 must be very small as this operator is tried on each gene of the RHS segment.

4.3. Chapter Summary

This chapter offers information on problem definition and assumptions, inspection allocation model, model for sequence-Dependent FSP and integrated inspection and scheduling model. First, a problem definition and assumptions is provided in Section 3.2, followed by inspection allocation model 3.3. The model for sequence-Dependent FSP is provided in Section 3.4. Section 3.5 is about integrated inspection and scheduling model.

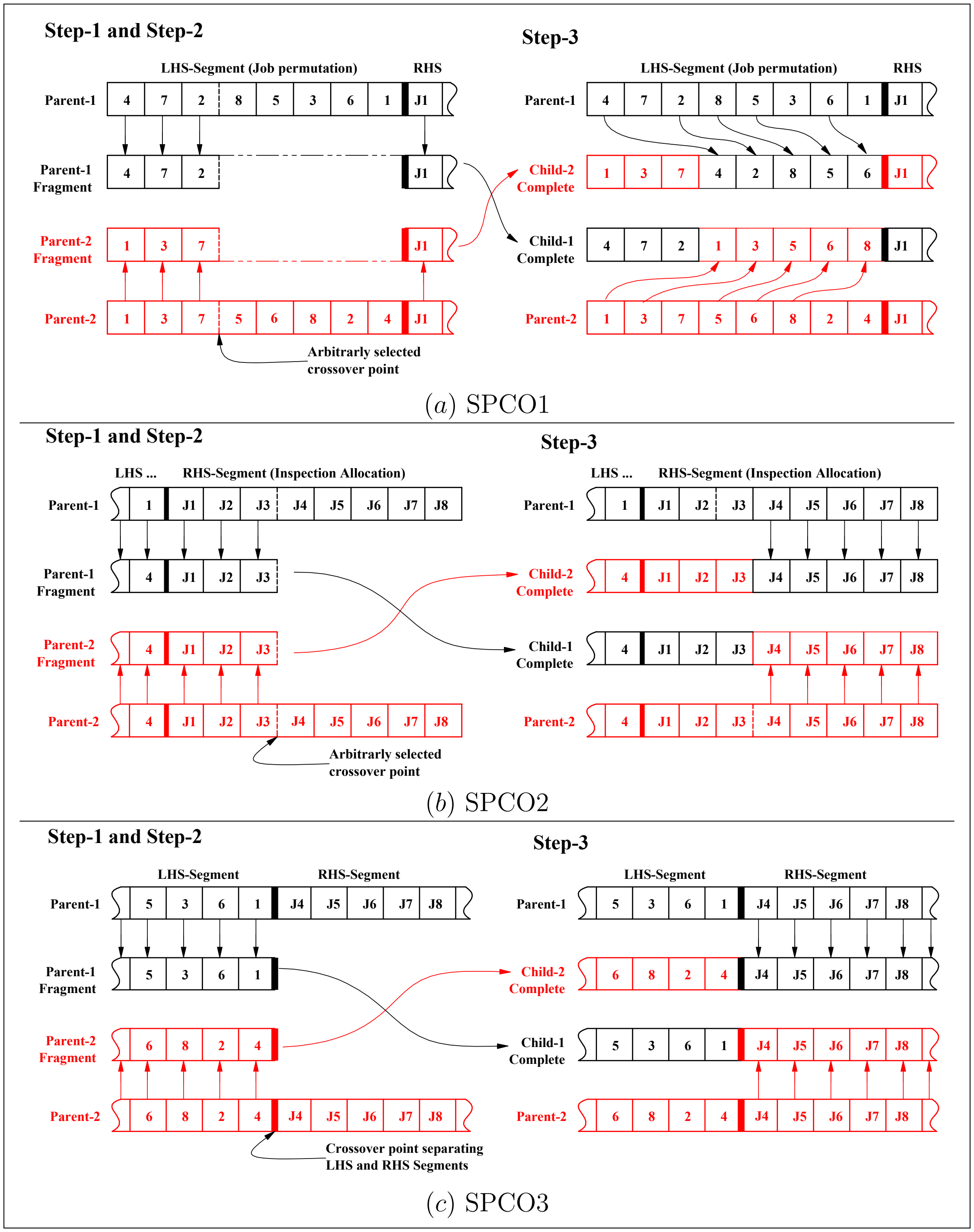

Figure 4.5: The three single point cross-over operators.

Algorithm 1: Given $y_{j,n}$, detemine b_j, $g_{j,n}$, $d_{j,n}$, and Z_1

Input : Inspection allocation decision $y_{j,n}$ for all (j, n)
Output: Values for b_j, $g_{j,n}$, $d_{j,n}$ and Z_1

for $j = 1$ **to** $j = J$ **do**
  /* **Calculate** $\hat{g}_{j,1}$ **and** $\hat{d}_{j,1}$ **for** $n = 1$ */
  $\hat{g}_{j,1} = 1 - \Theta_j$
  $\hat{d}_{j,1} = \Theta_j \times (1 - y_{j,1})$
  /* **Calculate** $\hat{g}_{j,n}$ **and** $\hat{d}_{j,n}$ **for** $n = 2,\ 3,\ \cdots,\ N$ */
  for $n = 2$ **to** $n = N$ **do**
    $\hat{g}_{j,n} = (1 - \Theta_j) \times \hat{g}_{j,n-1}$
    $\hat{d}_{j,n} = \left(\hat{d}_{j,n-1} + \Theta_j \times \hat{g}_{j,n-1}\right) \times (1 - y_{j,n})$
  end for
  $\hat{R}_j = \hat{g}_{j,N} + \hat{d}_{j,N}$
end for
/* **Calculate** b_j**,** $g_{j,n}$ **and** $d_{j,n}$ */
for $j = 1$ **to** $j = J$ **do**
  $b_j = R_j / \hat{R}_j$
  for $n = 1$ **to** $n = N$ **do**
    $g_{j,n} = \hat{g}_{j,n} \times (R_j / \hat{R}_j)$
    $d_{j,n} = \hat{d}_{j,n} \times (R_j / \hat{R}_j)$
  end for
end for
/* **Calculate** Z_1 */
Set $Z_1 = 0$
for $j = 1$ **to** $j = J$ **do**
  $Z_1 = Z_1 + \{I_{j,1} \times y_{j,1}\}$
  for $n = 2$ **to** $n = N$ **do**
    $Z_1 = Z_1 + \{I_{j,n} \times y_{j,n}\} + \{P_{j,n} \times d_{j,n-1}\}$
  end for
  $Z_1 = Z_1 + C_j \times d_{j,N}$
end for

Algorithm 2: Given $y_{j,n}$, b_j, $g_{j,n}$, $d_{j,n}$, and sequence of jobs from the LHS-Segment of chromosome, determined $e_{j,n}$, $f_{j,n}$, $c_{r,j}$, $w_{k,q}$ and z_2

Input : Values of b_j, $y_{j,n}$, , $g_{j,n}$, $d_{j,n}$ for all (j, n)
Input : Sequence of jobs from the LHS-Segment
Output: Completion times $e_{j,n}$, $f_{j,n}$, $c_{r,j}$, $w_{k,q}$ and makespan Z_2

Set $k_q = 1$ for all q
Set ψ_1 same as the LHS-Segment of the chromosome under evaluation.
for $n = 1$ **to** $n = N$ **do**
  for $r = 1$ **to** $r = J$ **do**
    $j = \psi_n(r)$
    /* **Determine $v_{r,n}$ using Algorithm-3** */
    $v_{r,n} =$ Algorithm-3$(n,\ r,\ j)$
    $e_{j,n} = v_{r,n}$
    if $y_{j,n} = 1$ **then**
      /* **Select an inspection station $q*$ that finishes the inspection soon and determined $w_{k_{q^*},q^*}$** */
      $w_{k_{q^*},q^*} =$ Algorithm-4$(n,\ j)$
      $f_{j,n} = w_{k_{q^*},q^*}$
      $k_{q^*} = k_{q^*} + 1$ /* **Increase by 1** */
    end if
    /* **Calculate job's ready time for its processing in stage $n + 1$** */
    if $n < N$ **then**
      $\gamma_{j,n+1} = e_{j,n} \times (1 - y_{j,n}) + f_{j,n} \times y_{j,n}$
    end if
  end for
  if $n < N$ **then**
    $\psi_{n+1} =$ **sort**$(\gamma_{1,n+1}, \gamma_{2,n+1}, \cdots \gamma_{J,n+1})$
    /* **The function sort returns an ordered list of jobs based on non-increasing order of their ready time $\gamma_{j,n+1}$** */
  end if
end for
$z_2 = \max\{e_{j,N},\ f_{j,N} \text{ for all } j\}$

Algorithm 3: Given stage index n, run index r, and job index j determined $v_{r,n}$

Input : Indices n, r and j
Output: Completion times $v_{r,n}$ and $e_{j,n}$

if $n = 1$ *and* $r = 1$ **then**
  $v_{r,n} = S^*_{j,n} + (T_{j,n} \times b_j)$
else if $n = 1$ *and* $r > 1$ **then**
  $j' = \psi_n(r-1)$
  $v_{r,n} = v_{r-1,n} + S_{j,n,j'} + T_{j,n} \times b_j$
else if $n > 1$ *and* $r = 1$ **then**
  if $y_{n-1} = 1$ **then**
    $v_{r,n} = f_{n-1,j} + S^*_{j,n} + T_{j,n} \times g_{j,n-1}$
  end if
  $v_{r,n} = e_{n-1,j} + S^*_{j,n} + T_{j,n} \times (g_{j,n-1} + d_{j,n-1})$
else
  /* **If** $n > 1$ **and** $r > 1$ */
  $j' = \psi_n(r-1)$
  if $y_{n-1} = 1$ **then**
    $v_{r,n} = \max\{f_{n-1,j},\ v_{r-1,n}\} + S_{j,n,j'} + T_{j,n} \times g_{j,n-1}$
  end if
  $v_{r,n} = \max\{e_{n-1,j},\ v_{r-1,n}\} + S_{j,n,j'} + T_{j,n} \times (g_{j,n-1} + d_{j,n-1})$
end if
$p_{j,r,n} = 1$

Algorithm 4: Given stage index n and job index j, select an inspection station q^* that finish the inspection soon and determine $w_{k_{q^*},q^*}$

Input : Indices n, and j
Output: Completion times $w_{k_{q^*},q^*}$

Set $M = 100000$ /* M is large number */
for $q = 1$ **to** $q = Q$ **do**
  if $k_q = 1$ **then**
    $Selected = q$
    break /* break the for loop */
  else
    if $w_{k_q-1,q} < M$ **then**
      $M = w_{k_q-1,q}$
      $Selected = q$
    end if
  end if
end for
$q^* = Selected$ /* Get the index of the selected inspection station. */
if $n = 1$ *and* $k_{q^*} = 1$ **then**
  $w_{k_{q^*},q^*} = e_{j,n} + U_{j,n} \times b_j$
else if $n = 1$ *and* $k_{q^*} > 1$ **then**
  $w_{k_{q^*},q^*} = \max\{e_{j,n}, w_{k_{q^*}-1,q^*}\} + U_{j,n} \times b_j$
else if $n > 1$ *and* $k_{q^*} = 1$ **then**
  $w_{k_{q^*},q^*} = e_{j,n} + U_{j,n} \times (g_{j,n-1} + d_{j,n-1})$
else
  /* if $n > 1$ and $k_{q^*} > 1$ */
  $w_{k_{q^*},q^*} = \max\{e_{j,n}, w_{k_{q^*}-1,q^*}\} + U_{j,n} \times (g_{j,n-1} + d_{j,n-1})$
end if
$f_{j,n} = w_{k_{q^*},q^*}$
$x_{j,k_{q^*},q^*,n} = 1$; $z_{k_{q^*},q^*} = 1$ /* for the record */
$k_{q^*} = k_{q^*} + 1$ /* Increase counter by 1 */

Chapter 5

Numerical Example

Numerical examples are important to check the accuracy and efficiency of the mathematical model. Accuracy refers to how the final result agrees with other optimization packages such as LINGO. On the other hand, efficiency refers to the period of time needed to solve any problem. In this Chapter, two examples are presented to explain the behavior of the proposed models in Chapter 3. Moreover, the examples in this chapter show how the proposed models solve the multi-stages flow-shop scheduling problem integrated with inspection allocation. Furthermore, the following sections will provide more details regarding the investigated results.

5.1. Inspection Allocation Prototype

This example is based on the inspection allocation model 3.3.2 without scheduling or sequence-dependent setup time. The demand D is equal to 190. Consider a flow-shop with 5 machines need to finish a single job only. Table 5.1 represent $\Theta_{j,n}$, $P_{j,n}$, $I_{j,n}$, and $y_{j,n}$. The requirements in this problem are to determine the initial batch size b_j, needed to meet the demand D_j, I_N,P_N,and ,C. To solve this kind of problem we need to use a backward recursive calculation from the last stage to the first stage, which is difficult to accomplish as we mentioned in

Chapter 4. Table 5.2, represents the values of $\hat{g}_{j,n}$ and $\hat{d}_{j,n}$ for each stage. The values of these two variables are important to calculate $\hat{R}_j$. Once $\hat{R}_j$, $\hat{g}_{j,n}$, $\hat{d}_{j,n}$ are determined, the values for b_j, Z, and, C can be easily calculated as shown in table 5.3. The alternative approach is valid because of the linear relationship between the demand and the batch size. Fig 5.1 shows the steps for solving inspection allocation prototype

Table 5.1: Data Inputs For Inspection Allocation Prototype

j	$(n1)$	$(n2)$	$(n3)$	$(n4)$	$(n5)$
			$\Theta_{j,n}$		
1	0.03	0.01	0.03	0.02	0.01
			$P_{j,n}$		
1	0	30.00	20.00	10.00	10.00
			$I_{j,n}$		
1	32.00	32.00	32.00	17.00	17.00
			$Y_{j,n}$		
1	1	0	1	0	1

Table 5.2: The rate of good and defective items for each stage

j	$(n1)$	$(n2)$	$(n3)$	$(n4)$	$(n5)$
			$\hat{g}_{j,n}$		
1	0.97	0.9603	0.931491	0.91286118	0.90373257
			$\hat{d}_{j,n}$		
1	0	0.0097	0	0.01862982	0

Table 5.3: Solutions for Inspection Allocation Prototype

$\hat{R}_j$	b_j	Z_1	C
0.90373257	210	402.52	0

	Description
	Given $y_{j,n}$, $I_{j,n}$, $P_{j,n}$, and $\Theta_{j,n}$ (Table 5.1)
	Determine b_j, $g_{j,n}$, $d_{j,n}$, and Z_1
1	$j = 1$
2	**Step-1 calculate $\hat{g}_{j,1}$ and $\hat{d}_{j,1}$ for $n = 1$**
3	$\hat{g}_{j,1} = 1 - \Theta_1 =$ (1-0.03)$=$ 0.97
4	$\hat{d}_{j,1} = \Theta_1 \times (1 - y_{j,1}) = 0.03 \times (1-1) = 0$
5	**Step-2 calculate $\hat{g}_{j,n}$ and $\hat{d}_{j,n}$ for $n = 2 - n = 5$**
7	$\hat{g}_{j,n} = (1 - \Theta_j) \times \hat{g}_{j,n-1}$
	(n = 2) $\hat{g}_{1,2} = 0.96$ (n = 3) $\hat{g}_{1,3} = 0.93$ (n = 4) $\hat{g}_{1,4} = 0.91$ (n = 5) $\hat{g}_{1,5} = 0.90$
8	$\hat{d}_{j,n} = \left(\hat{d}_{j,n-1} + \Theta_j \times \hat{g}_{j,n-1}\right) \times (1 - y_{j,n})$
	(n = 2) $\hat{d}_{1,2} = 0.009$ (n = 3) $\hat{d}_{1,3} = 0$ (n = 4) $\hat{d}_{1,4} = 0.018$ (n = 5) $\hat{d}_{1,5} = 0$
	Step-3 calculate $\hat{R}_j$
10	$\hat{R}_j = \hat{g}_N + \hat{d}_N = (0.903 + 0) = 0.903$
	Step-4 calculate b_j
14	$b_j = R_j/\hat{R}_j =$ (190/0.9037) =210.239
	Step-5 calculate $g_{j,n}$ and $d_{j,n}$
16	$g_{j,n} = \hat{g}_{j,n} \times (R_j/\hat{R}_j)$
	$g_{1,1} = 203.9 * g_{1,2} = 201.8 * g_{1,3} = 195.8 * g_{1,4}$=191.9 $*$ $g_{1,5}$=190
17	$d_{j,n} = \hat{d}_{j,n} \times (R_j/\hat{R}_j)$
	(n = 1) $d_{1,1}$ =0 (n = 2) $d_{1,2}$=2.039 (n = 3) $d_{1,3}$=0 (n = 4) $d_{1,4}$=3.91 (n = 5) $d_{1,5}$= 0
	Step-6 Calculate Z_1
21	$Z_1 = 0$
22	$j = 1$ **to** $j = 5$
23	$Z_1 = Z_1 + \{I_{j,1} \times y_{j,1}\}$= 32
24	$n = 2$ **to** $n = N$
25	$Z_n = Z_1 + \{I_{j,n} \times y_{j,n}\} + \{P_{j,n} \times d_{j,n-1}\}$
	(n = 2) $Z_1 = 32$ (n = 3) $Z_1 = 104.78$ (n = 4) $Z_1 = 104.78$ (n = 5) $Z_1 = 160.95$
27	$Z_1 = Z_1 + C_j \times d_{j,N} =$ (402.52)+(0*0) $=$ 402.52

Figure 5.1: Steps To Solve Inspection Allocation Prototype

5.2. A Prototype of Flow-Shop Scheduling Problem with Sequence Dependent Setup Time Integrated with Inspection Allocation

This problem consists of 4 jobs that need to be processed in a total of five stages using a maximum of one machine in each stage and two inspection stations. For each job, there is a sequence-dependent setup time. The problem involves the values of $\Theta_{j,n}$, $P_{j,n}$, $I_{j,n}$, $U_{j,n}$, $S_{j,n}$, $S_{j,n,j'}$, R_j, and ,C_j whereas all values are provide in tables 5.4 and 5.5. In this example we assumed that all five machines are available from $t = 0$. However each machine has a different setup time. Table 5.6 contains the start-time and the end-time for each job. On the other hand, table 5.7 contains inspection start-time and inspection end-time for each job. Table 5.8, 5.9, 5.10, and, 5.11 are additional information for the scheduling problem. Through solving this problem we can easily find the best sequence of all jobs in each stage and completion time of the last job which is known by makespan. Figure 5.2 is a Gantt chart for problem 5.2.

5.3. Performance Comparing CPLEX and GA

In this part of the Chapter, we provide four examples (5.12). The first two examples are small problems size, while example 3 and example 4 are complex problems. The target of adding these examples is to demonstrate the performance of GA and compare between CPLEX and GA. The algorithms that have been used in this thesis are coded through C++ programming language. Figure 5.3 - 5.6 offer information on problem-1, figure 5.7 - 5.10 on problem-2, figure 5.11 - 5.14 on problem-3, and figure 5.15 - 5.18 on problem-4.

Table 5.4: Data set-1 for Problem-2

j	$n1$	$n2$	$n3$	$n4$	$n5$
			$\Theta_{j,n}$		
1	0.04	0.03	0.03	0.03	0.04
2	0.04	0.02	0.02	0.04	0.01
3	0.01	0.04	0.03	0.02	0.03
4	0.04	0.03	0.01	0.01	0.01
			$P_{j,n}$		
1	0	9.00	9.00	9.00	6.00
2	0	12.00	6.00	12.00	6.00
3	0	6.00	9.00	12.00	9.00
4	0	12.00	12.00	9.00	9.00
			$I_{j,n}$		
1	32.00	32.00	32.00	32.00	32.00
2	17.00	17.00	17.00	32.00	17.00
3	32.00	32.00	17.00	17.00	32.00
4	32.00	32.00	17.00	17.00	17.00
			$T_{j,n}$		
1	0.80	0.70	0.80	0.50	1.50
2	0.90	0.50	1.20	0.60	0.50
3	0.80	0.90	1.30	1.40	0.90
4	0.80	0.50	0.90	0.70	0.50
			$U_{j,n}$		
1	2.40	2.60	2.80	3.00	3.80
2	2.60	1.20	3.00	4.00	3.80
3	1.20	1.40	2.20	2.60	1.40
4	1.40	3.60	1.00	3.60	1.20
			$S_{j,n}$		
1	60.00	90.00	30.00	75.00	45.00
2	30.00	75.00	60.00	75.00	60.00
3	75.00	75.00	90.00	45.00	75.00
4	45.00	75.00	75.00	45.00	90.00

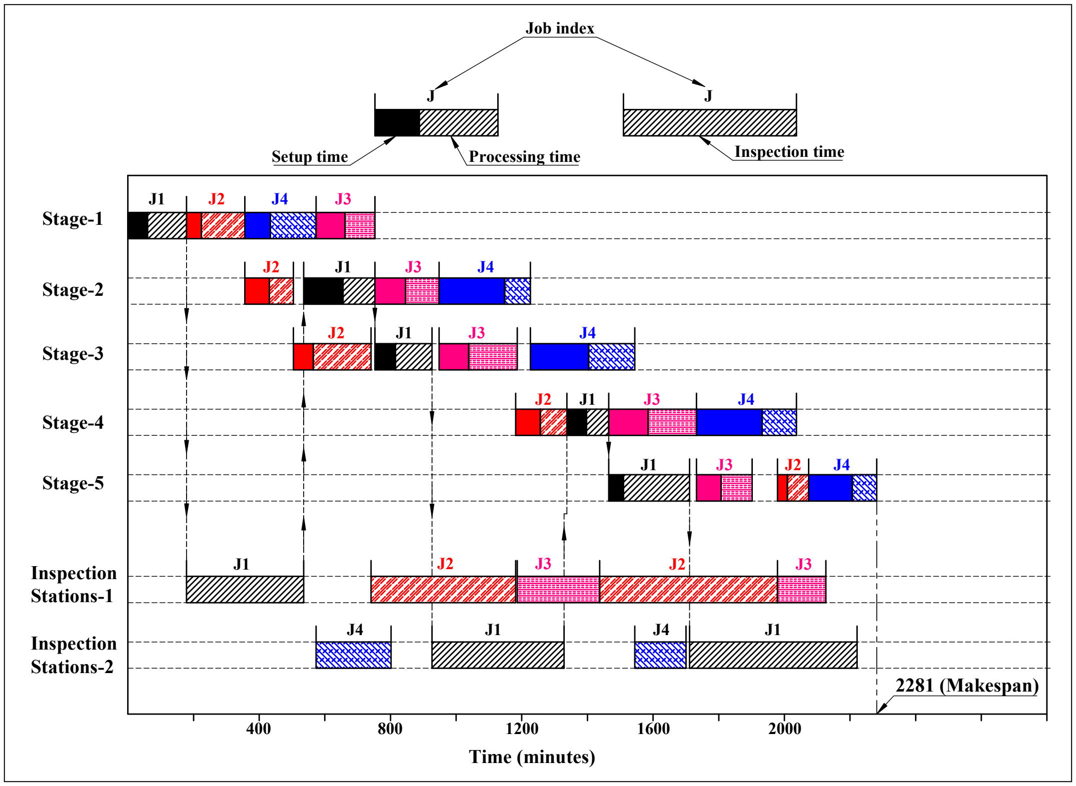

Figure 5.2: Inspection and operation scheduling for Problem-2.

5.3.1. The Performance Summary

The results of the four cases show that our GA performed better than branch-and-bound algorithm (5.13). We note that the GA has solved the four examples in a very short duration of time. Therefore, using GA is an efficient way to solve any complex scheduling problem.

Table 5.5: Data set-2 for Problem-2

				$S_{j,n,j'}$			
j	R_j	C_j	n	$j' = 1$	$j' = 1$	$j' = 1$	$j' = 1$
1	125	14	1	0	105.00	120.00	155.00
			2	0	120.00	90.00	87.50
			3	0	60.00	90.00	155.00
			4	0	60.00	105.00	87.50
			5	0	75.00	75.00	177.50
2	130	18	1	45.00	0	45.00	177.50
			2	90.00	0	45.00	200.00
			3	90.00	0	60.00	87.50
			4	60.00	0	30.00	132.50
			5	45.00	0	30.00	132.50
3	100	20	1	45.00	60.00	0	87.50
			2	90.00	90.00	0	200.00
			3	90.00	45.00	0	87.50
			4	120.00	75.00	0	110.00
			5	75.00	120.00	0	65.00
4	150	18	1	110.00	87.50	177.50	0
			2	177.50	155.00	200.00	0
			3	132.50	155.00	177.50	0
			4	200.00	155.00	200.00	0
			5	155.00	132.50	177.50	0

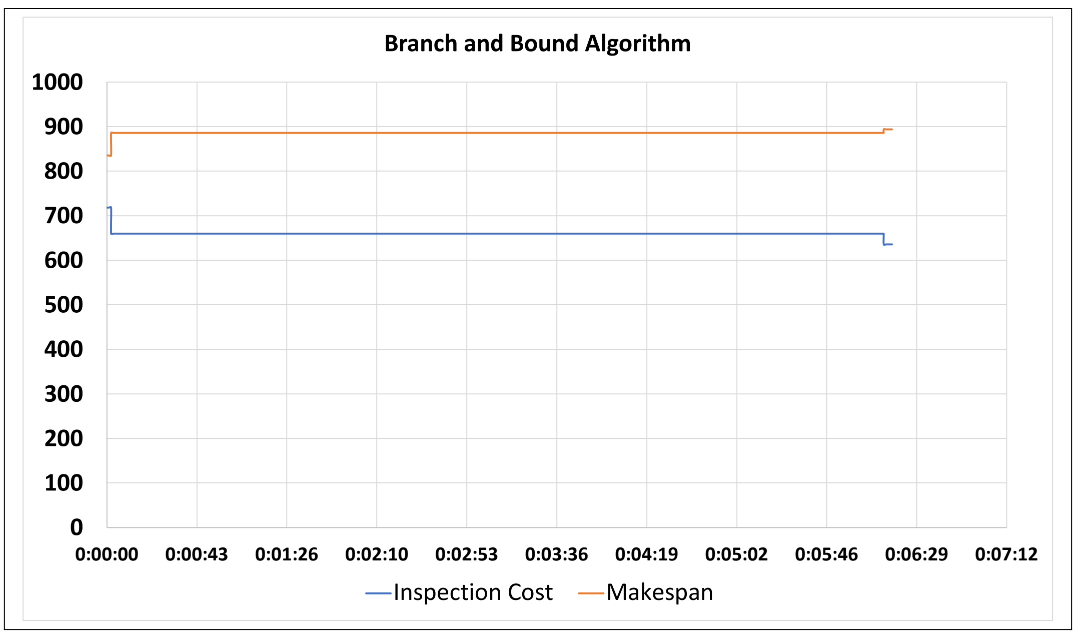

Figure 5.3: Graph-1 for Problem-1

Table 5.6: Detail schedule on the five processing stages of Problem-2

Stage n	Run r	Job j	SS	SE/PB	PE
1	1	1	0	60	179
	2	2	179	224	356
	3	4	356	444	574
	4	3	574	661	753
2	1	2	356	431	505
	2	1	536	656	755
	3	3	755	845	948
	4	4	948	1148	1226
3	1	2	505	565	741
	2	1	755	815	930
	3	3	948	1038	1186
	4	4	1226	1404	1544
4	1	2	1181	1256	1338
	2	1	1338	1398	1465
	3	3	1465	1585	1732
	4	4	1732	1932	2037
5	1	1	1465	1510	1711
	2	3	1732	1807	1902
	3	2	1979	2009	2074
	4	4	2074	2207	2282

SS = Setup Starts; SE/PB = Setup ends and Processing Begins; PE = Processing Ends.

Table 5.7: Detail schedule on the two inspection stations of Problem-2

Inspection Station n	Inspection run r	Job j	From Station n	IB	IE
1	1	1	1	179	536
	2	2	3	741	1181
	3	3	3	1186	1438
	4	2	4	1438	1979
	5	3	5	1979	2126
2	1	4	1	574	802
	2	1	3	930	1329
	3	4	3	1544	1701
	4	1	5	1711	2221

IS = Inspection Starts; IE = Inspection Ends.

Table 5.8: Additional solution information-1.

j	b_j		n = 1	2	3	4	5
1	148.6	$y_{1,n}$	1	0	1	0	1
		$g_{1,n}$	142.7	138.4	134.2	130.2	125
		$d_{1,n}$	0	4.3	0	4.0	0
2	146.9	$y_{2,n}$	0	0	1	1	0
		$g_{2,n}$	141	138.2	135.4	130	128.7
		$d_{2,n}$	5.9	8.7	0	0	1.3
3	114.1	$y_{3,n}$	0	0	1	0	1
		$g_{3,n}$	113.0	108.4	105.2	103.1	100
		$d_{3,n}$	1.1	5.7	0	2.1	0
4	162.7	$y_{4,n}$	1	0	1	0	0
		$g_{4,n}$	156.2	151.5	150	148.5	147.0
		$d_{4,n}$	0	4.7	0	1.5	3.0

Table 5.9: Additional solution information-2.

Stage	$j1$			$j2$			$j3$			$j4$		
n	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$
1	1	142.7	0.0	0	141.0	5.9	0	113.0	1.1	1	156.2	0.0
2	0	138.4	4.3	0	138.2	8.7	0	108.5	5.7	0	151.5	4.7
3	1	134.2	0.0	1	135.4	0.0	1	105.2	0.0	1	150.0	0.0
4	0	130.2	4.0	1	130.0	0.0	0	103.1	2.1	0	148.5	1.5
5	1	125.0	0.0	0	128.7	1.3	1	100.0	0.0	0	147.0	3.0
b_j	148.6			146.9			114.1			162.7		

Table 5.10: Additional solution information-3.

j	b_j		n = 1	2	3	4	5
1	148.6	$y_{1,n}$	1	0	1	0	1
		$g_{1,n}$	142.7	138.4	134.2	130.2	125
		$d_{1,n}$	0	4.3	0	4.0	0
2	146.9	$y_{2,n}$	0	0	1	1	0
		$g_{2,n}$	141	138.2	135.4	130	128.7
		$d_{2,n}$	5.9	8.7	0	0	1.3
3	114.1	$y_{3,n}$	0	0	1	0	1
		$g_{3,n}$	113.0	108.4	105.2	103.1	100
		$d_{3,n}$	1.1	5.7	0	2.1	0
4	162.7	$y_{4,n}$	1	0	1	0	0
		$g_{4,n}$	156.2	151.5	150	148.5	147.0
		$d_{4,n}$	0	4.7	0	1.5	3.0

Table 5.11: Additional solution information-4.

Stage	$j1$			$j2$			$j3$			$j4$		
n	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$	$y_{j,n}$	$g_{j,n}$	$d_{j,n}$
1	1	142.7	0.0	0	141.0	5.9	0	113.0	1.1	1	156.2	0.0
2	0	138.4	4.3	0	138.2	8.7	0	108.5	5.7	0	151.5	4.7
3	1	134.2	0.0	1	135.4	0.0	1	105.2	0.0	1	150.0	0.0
4	0	130.2	4.0	1	130.0	0.0	0	103.1	2.1	0	148.5	1.5
5	1	125.0	0.0	0	128.7	1.3	1	100.0	0.0	0	147.0	3.0
b_j	148.6			146.9			114.1			162.7		

Table 5.12: Details of Problem-1, 2, 3, and 4

Problem Number	Inspection Stations	Number of Jobs	Number of Stages
1	1	1	10
2	1	1	15
3	2	4	5
4	3	5	12

Table 5.13: Solution Information of Problem-1, 2, 3, and 4

Problem Number	Method	time to Determine the Solution	Inspection Cost (I)	penalty Cost (P)	penalty Cost (C)	Makespan of the Schedule
1	BB	00:06:17	408	227.359	0	893.867
	GA	00:00:08	408	227.359	0	893.867
2	BB	02:01:06	712	134.071	0	1816.45
	GA	00:00:08	712	134.071	0	1816.63
3	BB	05:26:37	164	474.933	332.773	2112.71
	GA	00:00:22	179	465.099	169.794	2084.92
4	BB	30:22:21	2232	1833.81	178.5	2712.06
	GA	00:00:42	2838	737.56	178.5	2434.84

GA = Genetic Algorithm, BB = Branch and Bound Algorithm. P = Penalty Cost of Processing Defective Items, C = Penalty cost of per defective item that can reach to the customer.

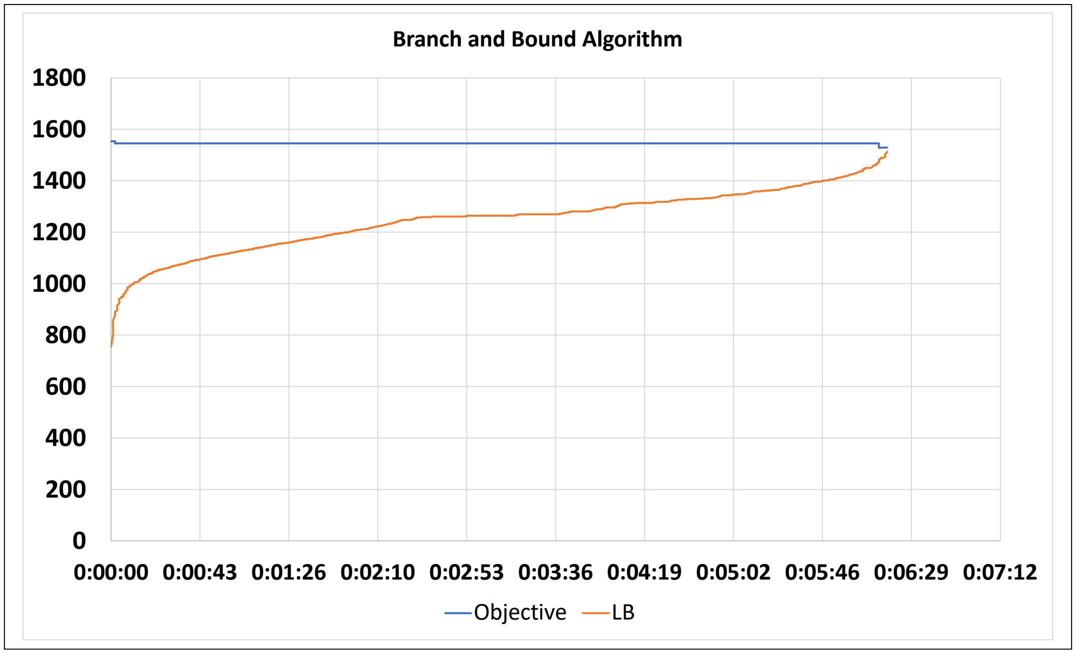

Figure 5.4: Graph-2 for Problem-1

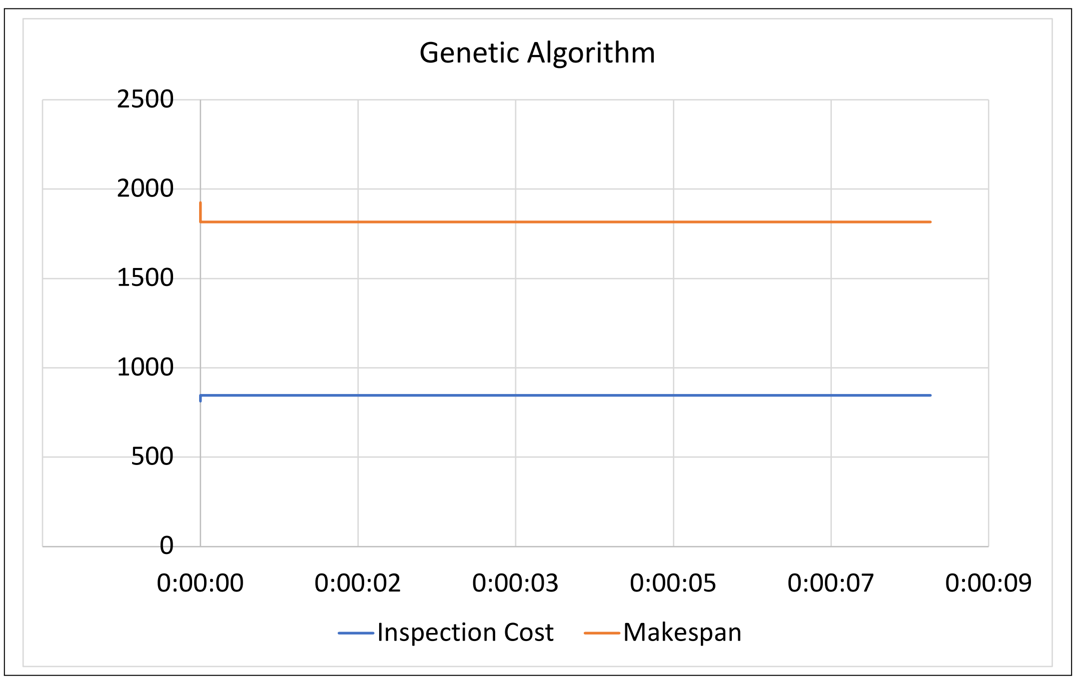

Figure 5.5: Graph-3 for Problem-1

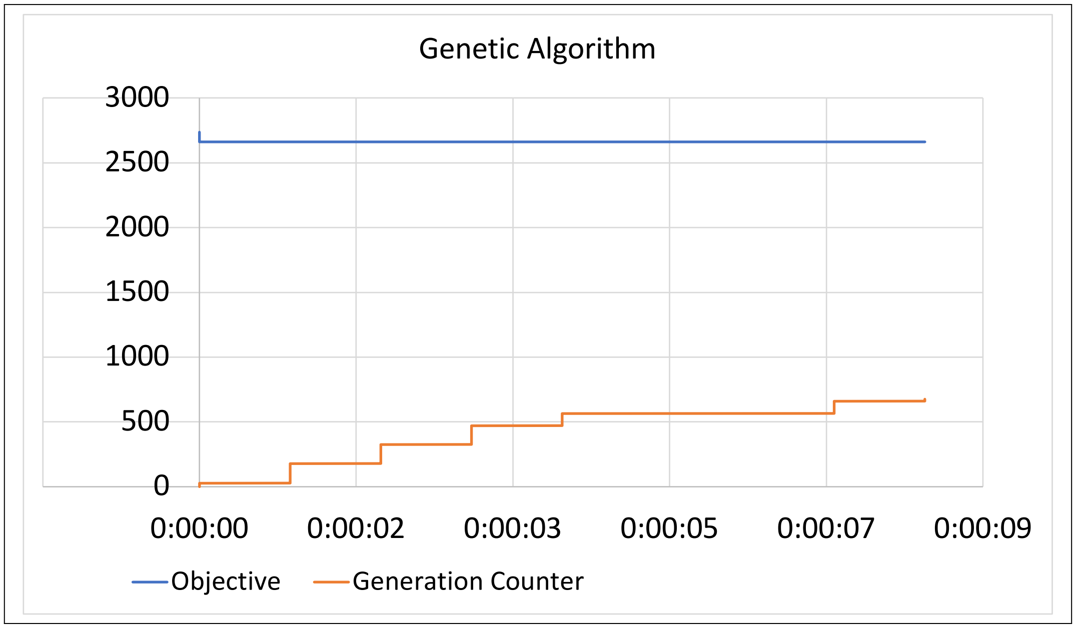

Figure 5.6: Graph-4 for Problem-1

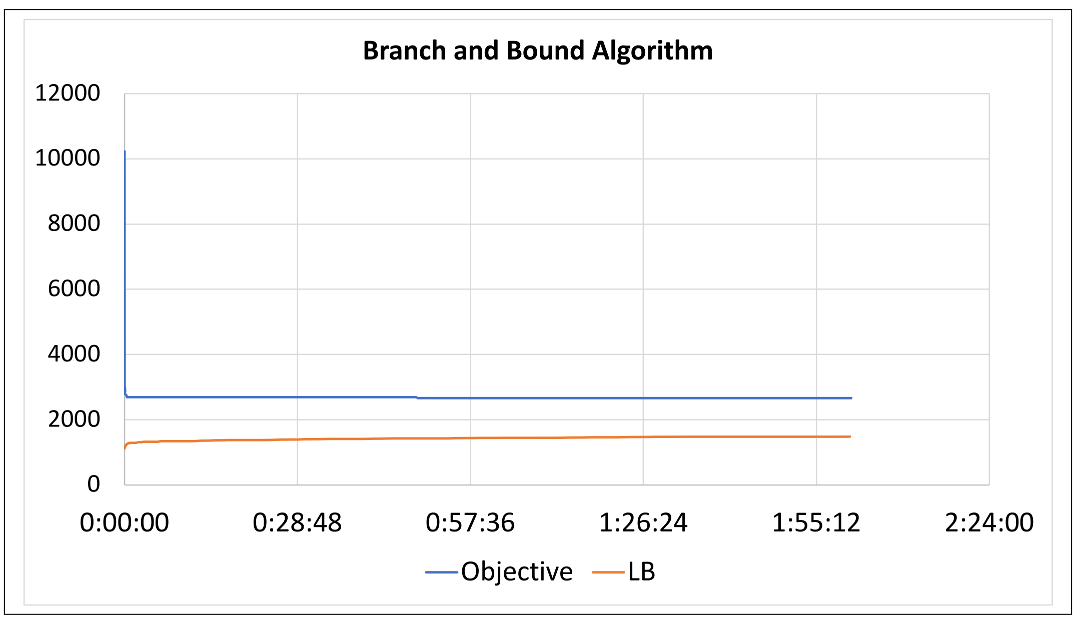

Figure 5.7: Graph-1 for Problem-2

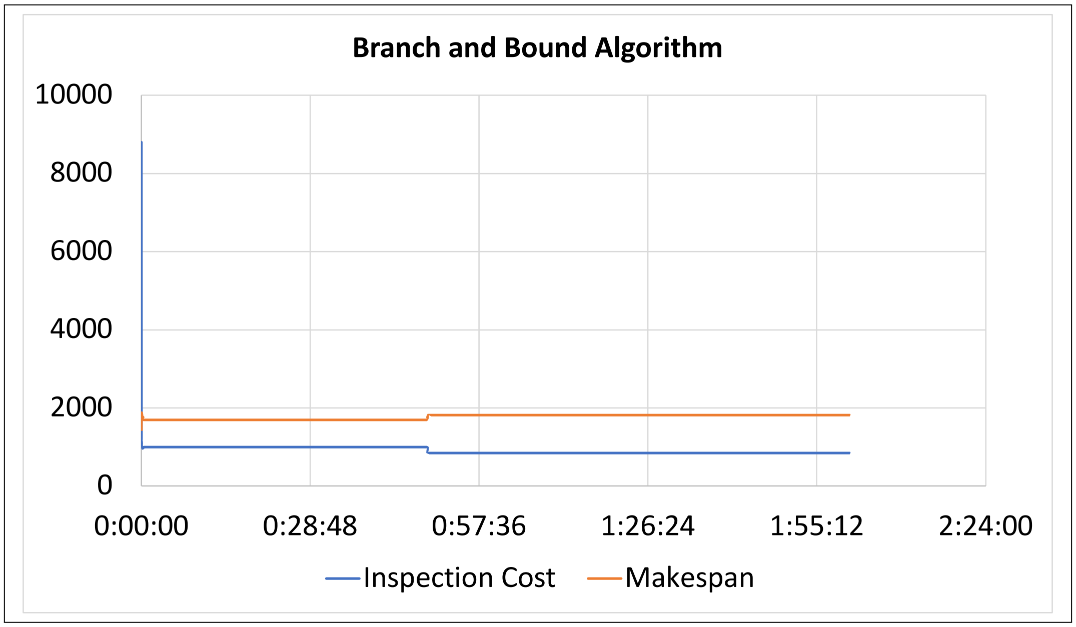

Figure 5.8: Graph-2 for Problem-2

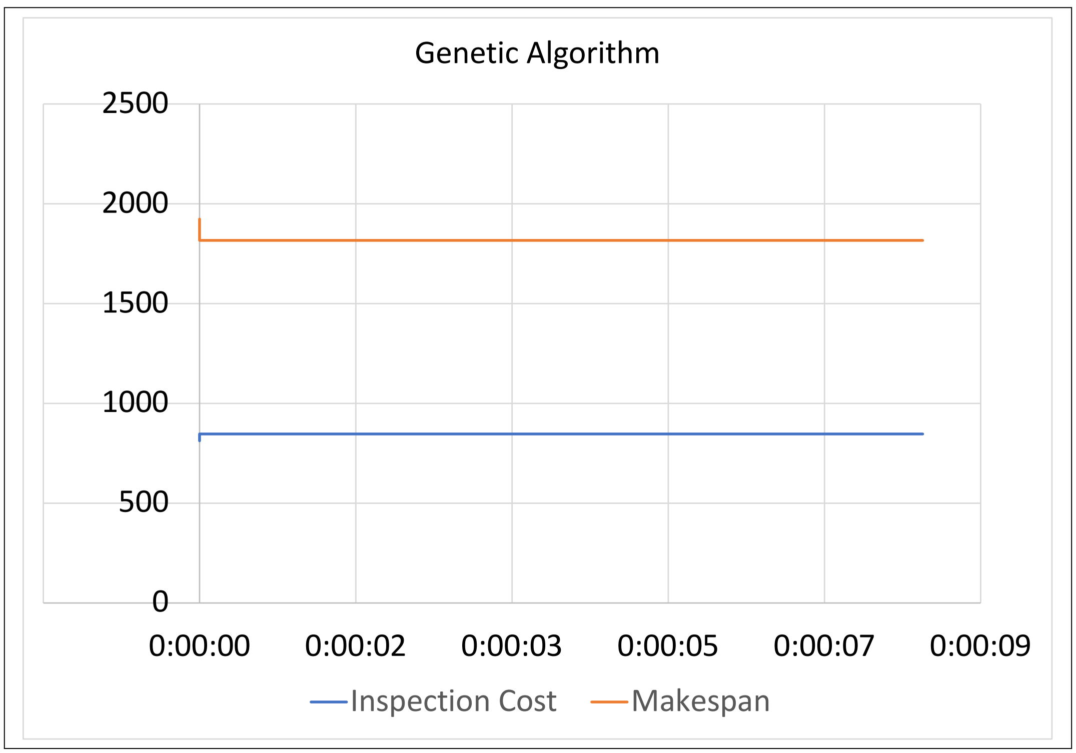

Figure 5.9: Graph-3 for Problem-2

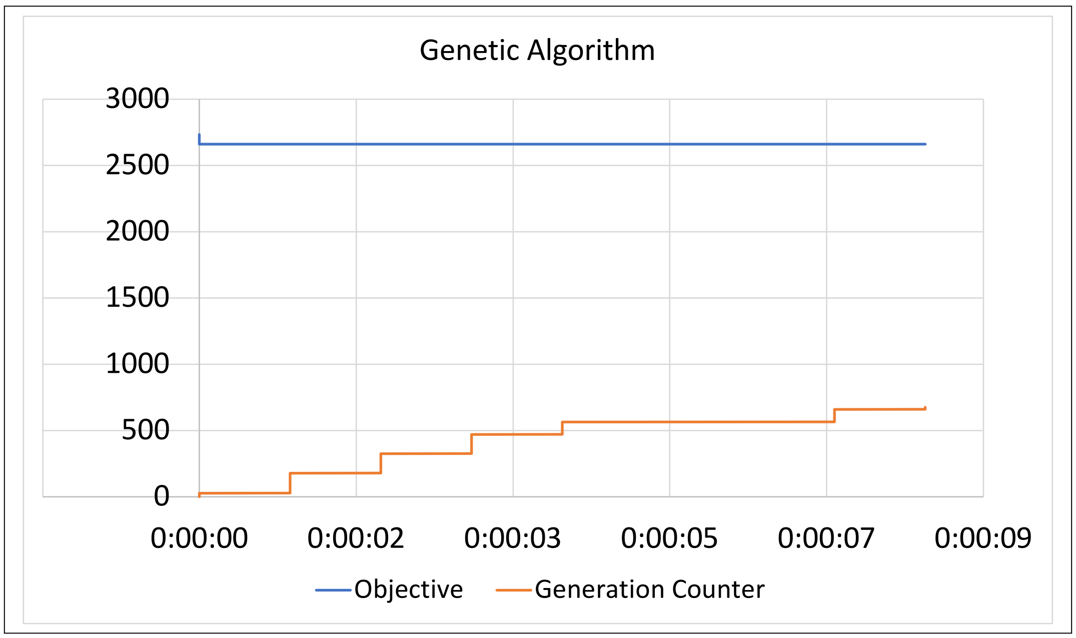

Figure 5.10: Graph-4 for Problem-2

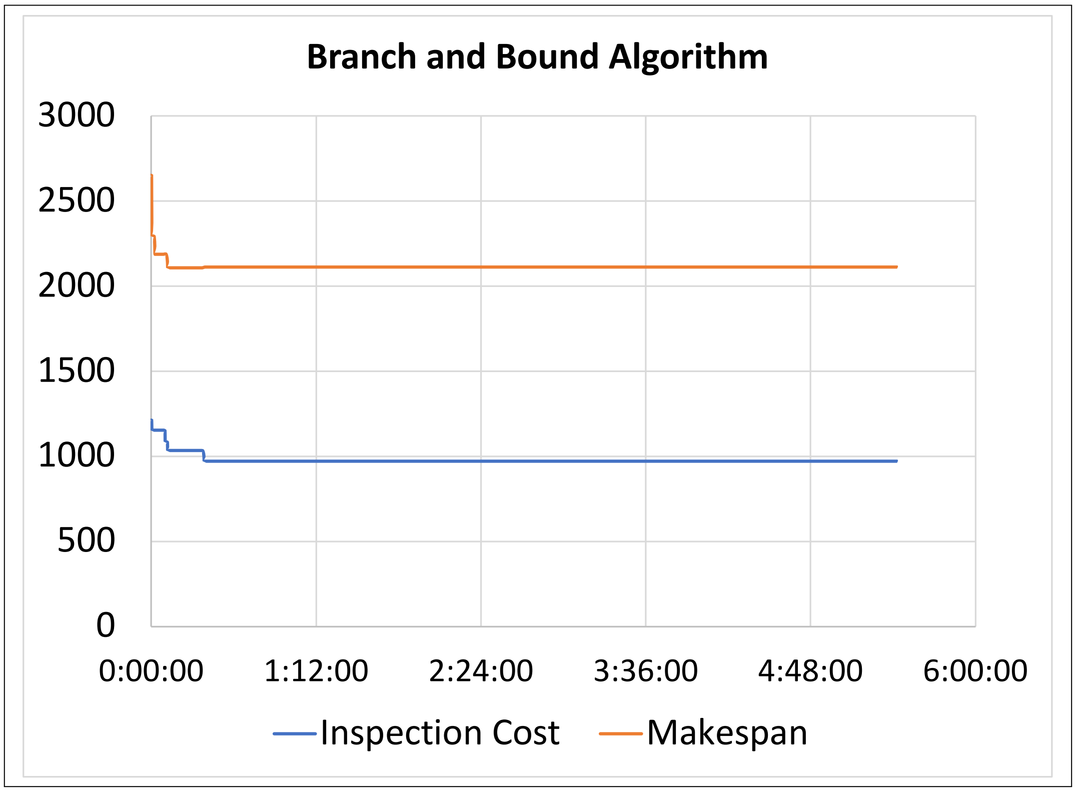

Figure 5.11: Graph-1 for Problem-3

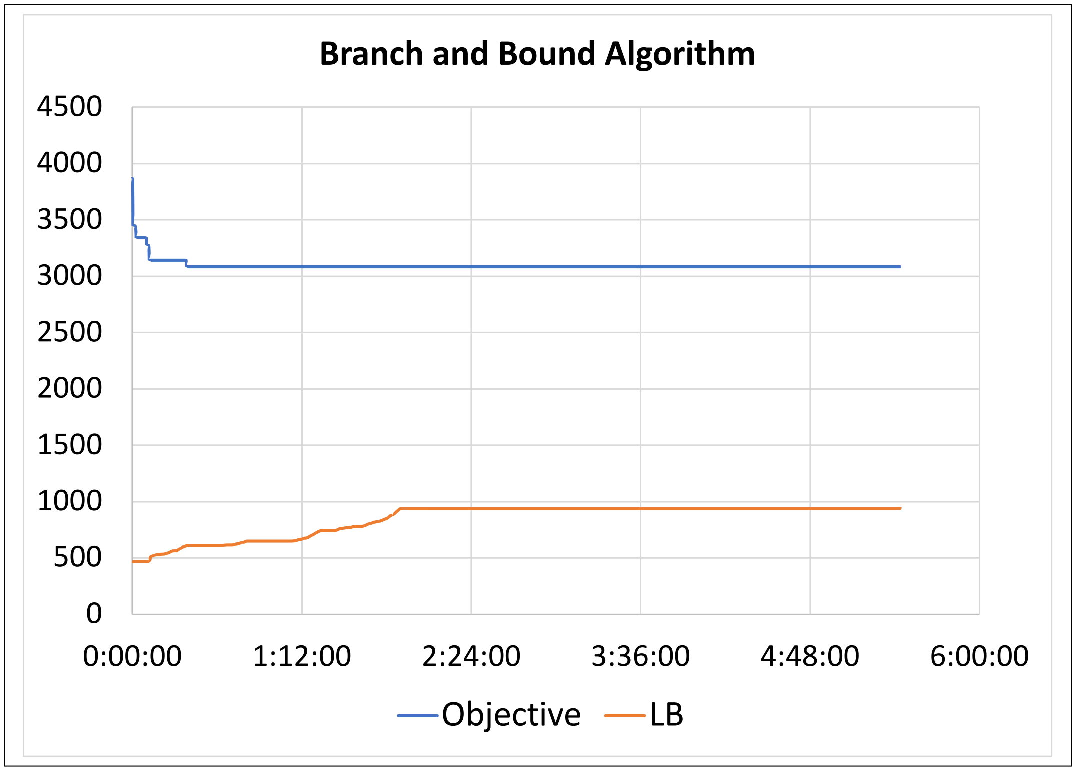

Figure 5.12: Graph-2 for Problem-3

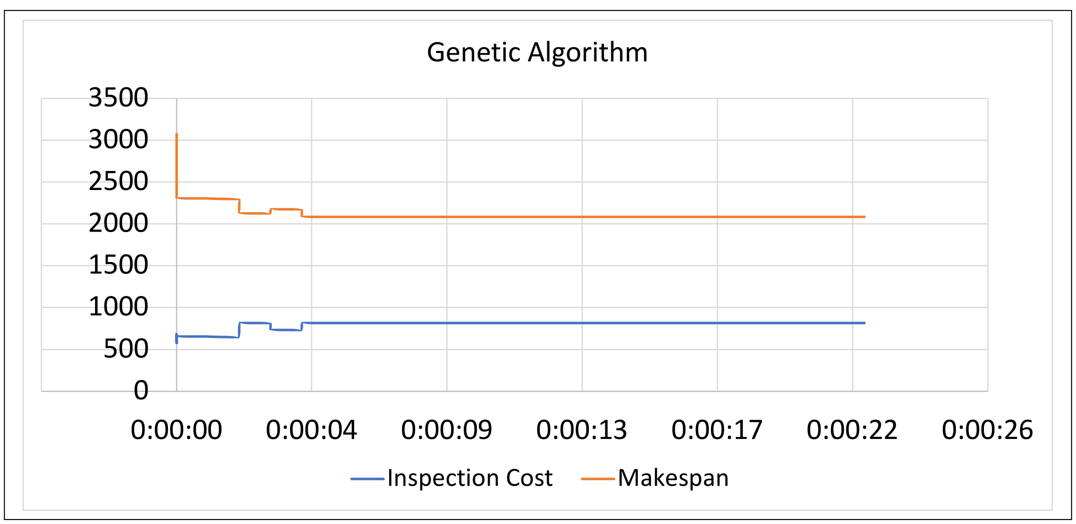

Figure 5.13: Graph-3 for Problem-3

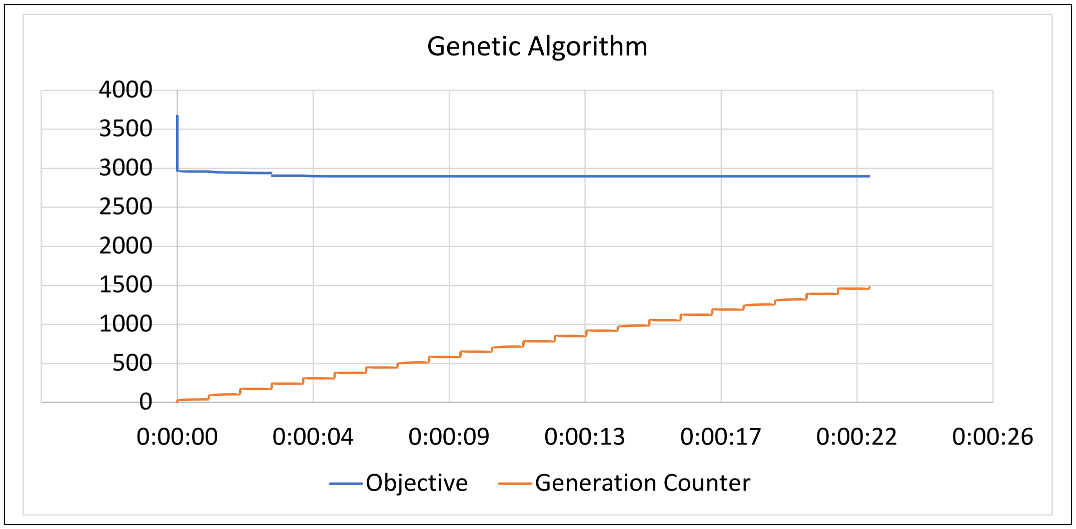

Figure 5.14: Graph-4 for Problem-3

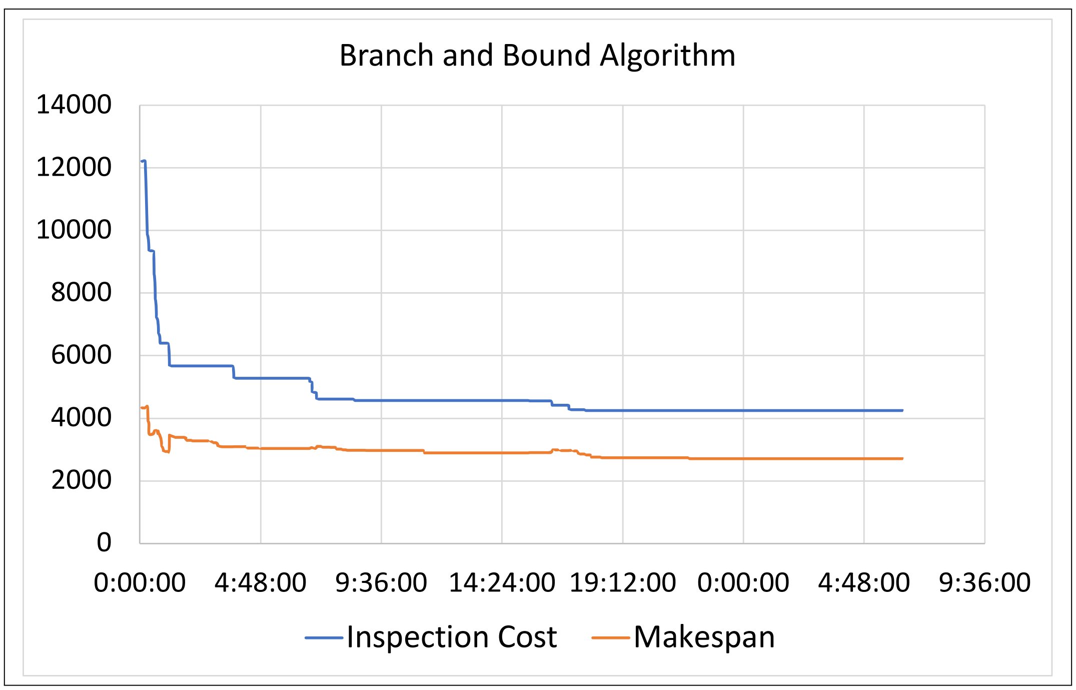

Figure 5.15: Graph-1 for Problem-4

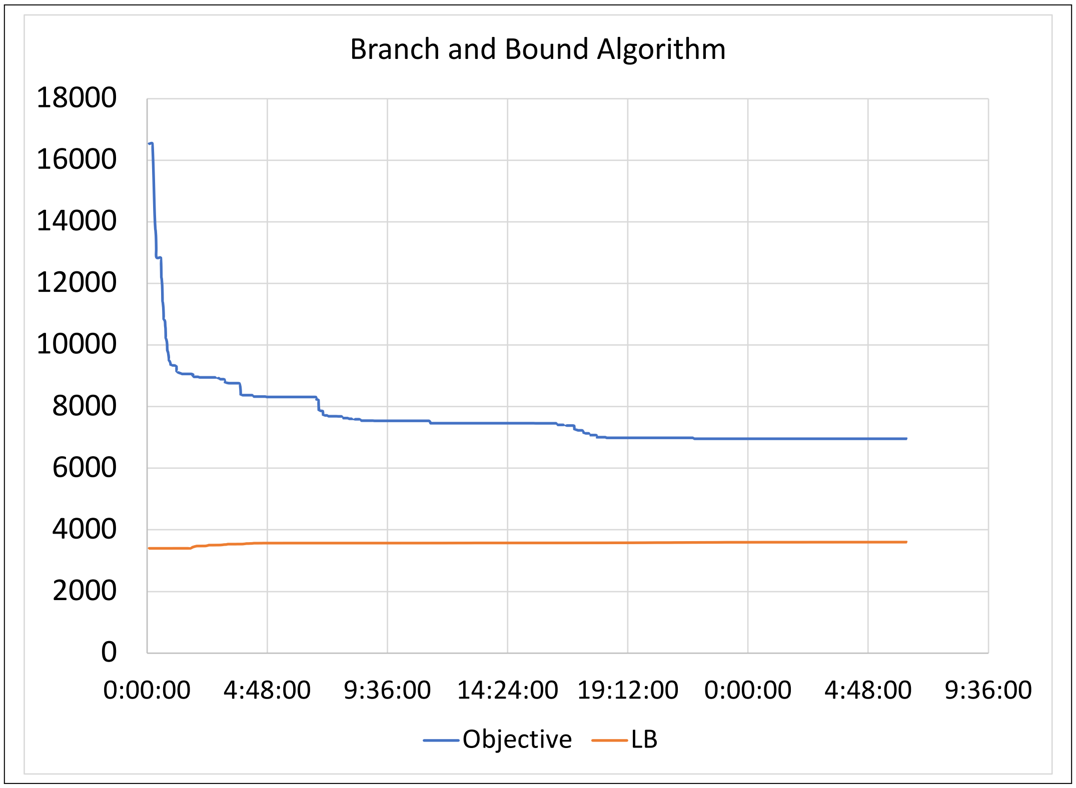

Figure 5.16: Graph-2 for Problem-4

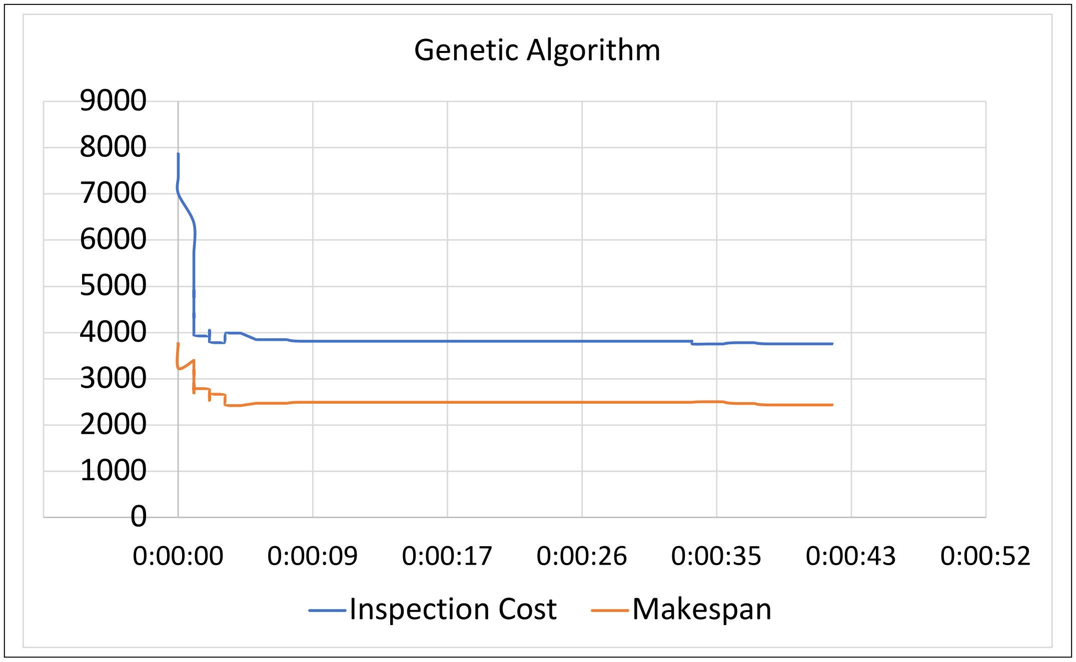

Figure 5.17: Graph-3 for Problem-4

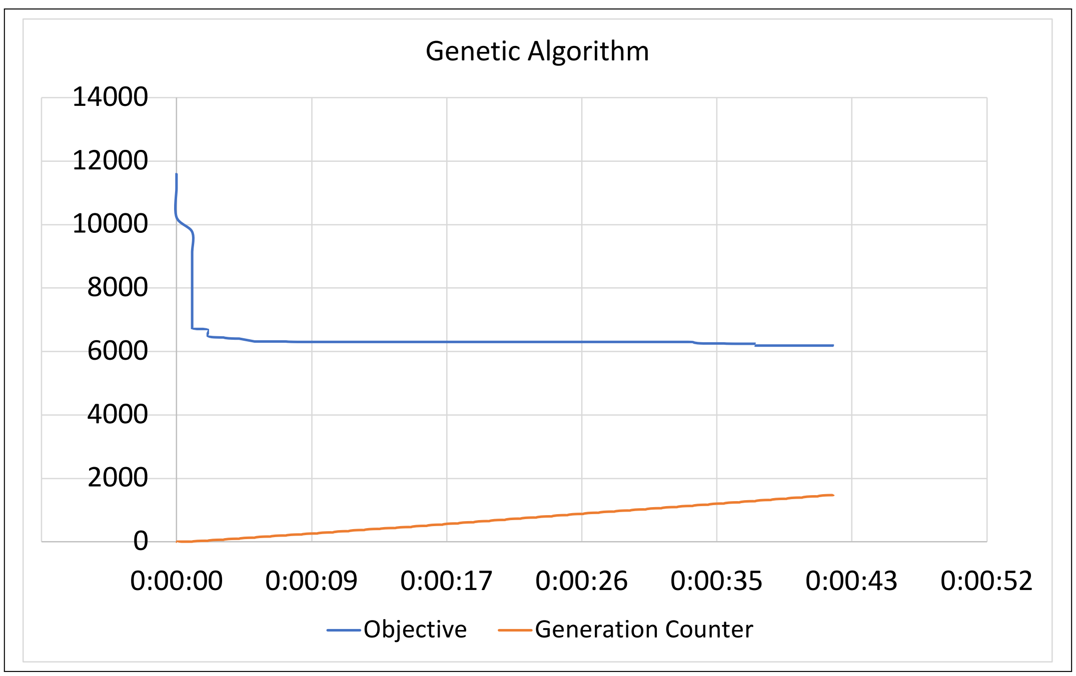

Figure 5.18: Graph-4 for Problem-4

Chapter 6

Concluding Remarks

6.1. Conclusion

A manufacturing system contains many elements depending on its complexity and the final product, including the number of machines, machines' environment, automation level, and other resources. In most cases, scheduling problems need to be solved quickly to maintain work requirements. hence, providing a quick and efficient solution is critical in today's competitive environment. As such, we have developed a mathematical model to solve flow shop Scheduling problem integrated with inspection allocation. Speeding up the process by avoiding unnecessary inspection operation is one of our primary objectives in this thesis. According to lean management, unnecessary activities during the operation are considered as non-value added activities. Removing this kind of activities is important in order to minimize operation cost. Optimizing inspection allocation can impact the defect rate, penalty cost, and customer satisfaction. Therefore, implementing an optimal inspection strategy is the best way to achieve high-quality standards with lowest cost in multistage production systems. Providing excellent quality with low cost are significant factors to enhance end-user satisfaction. Accordingly GA has been used to solve this problem.

6.2. Recommendations and Suggestions

Some of suggestions ideas for future research are summarized in the following points:

- Instead of integrating pure flow shop and inspection allocation, we can enhance the difficulty of the problem by integrating inspection allocation with flexible flow shop scheduling problems.
- The outputs from any inspection operation are rejected or passed products as our assumption. However, future research can involve rework and scrap items.
- We solve the FSSP integrated with inspection operations by using GA;however, it is possible to solve the same problem by combining genetic algorithm and simulated annealing.
- In this thesis, we assume that the inspection operation can be done after each workstation only. Instead of that assumption, inspection operation can be done before and after any workstation.
- Adding buffer allocation can be a value added in future studies.

Bibliography

Y. Adler, V. Shper, and O. Maksimova. Assignable causes of variation and statistical models: Another approach to an old topic. *Quality and Reliability Engineering International*, 27(5):623–628, jul 2011. ISSN 07488017. doi: 10.1002/qre.1207. URL https://onlinelibrary.wiley.com/doi/full/10.1002/qre.1207https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.1207https://onlinelibrary.wiley.com/doi/10.1002/qre.1207.

Edem O.P. Akpan. Job-shop sequencing problems via network scheduling technique. *International Journal of Operations and Production Management*, 16 (3):76–86, 1996. ISSN 01443577. doi: 10.1108/01443579610110503.

Basim Al-Najjar. Total quality maintenance: An approach for continuous reduction in costs of quality products. *Journal of Quality in Maintenance Engineering*, 2(3):4–20, sep 1996. ISSN 13552511. doi: 10.1108/13552519610130413.

Adel Al-Shayea, Emad Fararah, Emad Abouel Nasr, and Haitham A. Mahmoud. Model for Integrating Production Scheduling and Maintenance Planning of Flow Shop Production System. *IEEE Access*, 8:208826–208835, 2020. ISSN 21693536. doi: 10.1109/ACCESS.2020.3038719.

Ali Allahverdi and Fawaz S. Al-Anzi. The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *International*

Journal of Advanced Manufacturing Technology, 37(1-2):166–177, 2008. ISSN 02683768. doi: 10.1007/s00170-007-0950-y.

Ali Allahverdi, C. T. Ng, T. C.E. Cheng, and Mikhail Y. Kovalyov. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032, jun 2008. ISSN 03772217. doi: 10.1016/j.ejor.2006.06.060. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221706008174.

Davide Anghinolfi and Massimo Paolucci. A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times. *European Journal of Operational Research*, 193(1):73–85, 2009. ISSN 03772217. doi: 10.1016/j.ejor.2007.10.044.

Donald P. Ballou and Harold L. Pazer. Impact of Inspector Fallibility on the Inspection Policy in Serial Production Systems. *Management Science*, 28(4): 387–399, apr 1982. ISSN 00251909. doi: 10.1287/mnsc.28.4.387. URL https://pubsonline.informs.org/doi/abs/10.1287/mnsc.28.4.387.

C. G.E. Boender, A. H.G. Rinnooy Kan, G. T. Timmer, and L. Stougie. A stochastic method for global optimization. *Mathematical Programming*, 22(1): 125–140, dec 1982. ISSN 00255610. doi: 10.1007/BF01581033. URL https://link.springer.com/article/10.1007/BF01581033.

Shaukat A. Brah and John L. Hunsucker. Branch and bound algorithm for the flow shop with multiple processors. *European Journal of Operational Research*, 51(1):88–99, 1991. ISSN 03772217. doi: 10.1016/0377-2217(91)90148-O. URL https://www.researchgate.net/publication/249012477.

Chin Chih Chang, Tai Hsi Wu, and Chien Wei Wu. An efficient approach to determine cell formation, cell layout and intracellular machine sequence in cellular manufacturing systems. *Computers and Industrial Engineering*, 66 (2):438–450, 2013. ISSN 03608352. doi: 10.1016/j.cie.2013.07.009. URL http://dx.doi.org/10.1016/j.cie.2013.07.009.

Yan Chun Chen. An optimal production and inspection strategy with preventive maintenance error and rework. *Journal of Manufacturing Systems*, 32(1):99, jan 2013. ISSN 02786125. doi: 10.1016/j.jmsy.2012.07.010.

Shu Hsing Chung, Tai Hsi Wu, and Chin Chih Chang. An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations. *Computers and Industrial Engineering*, 60(1):7–15, 2011. ISSN 03608352. doi: 10.1016/j.cie.2010.08.016. URL http://dx.doi.org/10.1016/j.cie.2010.08.016.

Fantahun M Defersha, Mingyuan Chen, F M Defersha, and · M Chen. A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups. *The International Journal of Advanced Manufacturing Technology 2009 49:1*, 49(1):263–279, dec 2009. ISSN 1433-3015. doi: 10.1007/S00170-009-2388-X. URL https://link.springer.com/article/10.1007/s00170-009-2388-x.

Glenn R. Drake, Jeffrey S. Smith, and Brett A. Peters. Simulation as a planning and scheduling tool for flexible manufacturing systems. *Winter Simulation Conference Proceedings*, pages 805–812, 1995. ISSN 02750708. doi: 10.1145/224401.224735.

M Dudek-Burlikowska and D Szewieczek. The Poka-Yoke Method as an Improving Quality Tool of Operations in the Process. *Journal of Achievements in Materials and Manufacturing Engineering*, 36(1):95–102, 2009. ISSN 1734-8412. URL https://www.researchgate.net/publication/44385664.

Gary D. Eppen and E. Gerald Hurst. Optimal Location of Inspection Stations in a Multistage Production Process. *Management Science*, 20(8):1194–1200, apr 1974. ISSN 00251909. doi: 10.1287/mnsc.20.8.1194. URL https://pubsonline.informs.org/doi/abs/10.1287/mnsc.20.8.1194.

Tamer Eren. A bicriteria parallel machine scheduling with a learning effect of setup and removal times. *Applied Mathematical Modelling*, 33(2):1141–1150, feb 2009. ISSN 0307904X. doi: 10.1016/j.apm.2008.01.010. URL www.elsevier.com/locate/apm.

Stephen E. Fienberg and Stephen M. Stigler. W. Edwards Deming. *Statisticians of the Centuries*, pages 485–489, 2001. doi: 10.1007/978-1-4613-0179-0_104.

L. R. Foulds. The heuristic problem-solving approach. *Journal of the Operational Research Society*, 34(10):927–934, 1983. ISSN 14769360. doi: 10.1057/jors.1983.205. URL https://www.tandfonline.com/action/journalInformation?journalCode=tjor20.

Jose M. Framinan, Rainer Leisten, and Rubén Ruiz García. Manufacturing scheduling systems: An integrated view on models, methods and tools. *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*, 9781447162:1–400, 2014. doi: 10.1007/978-1-4471-6272-8.

Helio Yochihiro Fuchigami, Ruhul Sarker, and Socorro Rangel. Near-optimal heuristics for just-in-time jobs maximization in flow shop scheduling. *Algorithms*, 11(4), 2018. ISSN 19994893. doi: 10.3390/a11040043. URL https://www.mdpi.com/279814.

G. Galante and G. Passannanti. Integrated approach to part scheduling and inspection policies for a job shop manufacturing system. *International Journal of Production Research*, 45(22):5177–5198, nov 2007. ISSN 00207543. doi:

10.1080/00207540600788976. URL http://www.tandfonline.com/doi/abs/10.1080/00207540600788976.

T. N. Goh. Essential principles for decisionmaicing in the application of statistical quality control. *Quality Engineering*, 1(3):247–263, 1989. ISSN 15324222. doi: 10.1080/08982118908962658. URL https://www.tandfonline.com/action/journalInformation?journalCode=lqen20.

Teofilo Gonzalez and Sartaj Sahni. Open Shop Scheduling to Minimize Finish Time. *Journal of the ACM (JACM)*, 23(4):665–679, 1976. ISSN 1557735X. doi: 10.1145/321978.321985.

Jatinder N.D. Gupta. Flowshop Schedules With Sequence Dependent Setup Times. *Journal of the Operations Research Society of Japan*, 29(3):206–219, 1986. ISSN 0453-4514. doi: 10.15807/jorsj.29.206.

Jatinder N.D. Gupta and Edward F. Stafford. Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3):699–711, 2006. ISSN 03772217. doi: 10.1016/j.ejor.2005.02.001. URL www.elsevier.com/locate/ejor.

Harrington. Poor-Quality Cost. *Poor-Quality Cost*, feb 1987. doi: 10.1201/9780429259128. URL https://www.taylorfrancis.com/books/mono/10.1201/9780429259128/poor-quality-cost-james-harrington.

Don Herzog and Ronald Dworkin. *How to Think about Equality*, volume 100. 2002. ISBN 007034003X. doi: 10.2307/1290459. URL https://d1wqtxts1xzle7.cloudfront.net/38161019/juran-with-cover-page-v2.pdf?Expires=1651292829&Signature=NDEtWv4h1fEQvAz8irBsFTJOpb70XK8QKBJrRJgA6Jl-c3Ds2tJLoF1OEVY5vy9AG2hZV4FAomO4t

Wilfried Hinrichs. Product-specific adaption of conformity assessment criteria and their financial consequences. *Production Engineering*, 5(5):549–556, 2011. ISSN 09446524. doi: 10.1007/s11740-011-0329-7.

Muhammad Muazzem Hossain, Victor R. Prybutok, A. B.M. Abdullah, and Majharul Talukder. The development and research tradition of statistical quality control. *International Journal of Productivity and Quality Management*, 5(1):21–37, 2010. ISSN 17466482. doi: 10.1504/IJPQM.2010.029505. URL https://www.researchgate.net/publication/238341190.

H. Hwang and J. U. Sun. Production sequencing problem with re-entrant work flows and sequence dependent setup times. *International Journal of Production Research*, 36(9):2435–2450, 1998. ISSN 1366588X. doi: 10.1080/002075498192616. URL https://www.tandfonline.com/action/journalInformation?journalCode=tprs20.

Yoshihiro Ito. Strategic goals of quality costing in Japanese companies. *Management Accounting Research*, 6(4):383–397, dec 1995. ISSN 10445005. doi: 10.1006/mare.1995.1027.

Pempera Jarosław, Smutnicki Czesław, and Zelazny Dominik. Optimizing bicriteria flow shop scheduling problem by simulated annealing algorithm. *Procedia Computer Science*, 18:936–945, 2013. ISSN 18770509. doi: 10.1016/j.procs.2013.05.259. URL www.sciencedirect.com.

Elizabeth M. Jewkes. Optimal inspection effort and scheduling for a manufacturing process with repair. *European Journal of Operational Research*, 85(2): 340–351, sep 1995. ISSN 03772217. doi: 10.1016/0377-2217(94)00046-F.

Vimal Kumar, Pratima Verma, and Vignaesh Muthukumaar. The performances of process capability indices in the Six-Sigma competitiveness levels. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2018-March:1945–1951, 2018. ISSN 21698767. URL https://www.researchgate.net/publication/325959776.

Nani Kurniati, Ruey Huei Yeh, and Jong Jang Lin. Quality Inspection and Maintenance: The Framework of Interaction. *Procedia Manufacturing*, 4:244–251, 2015. ISSN 23519789. doi: 10.1016/j.promfg.2015.11.038.

J. Lee and S. Unnikrishnan. Planning quality inspection operations in multistage manufacturing systems with inspection errors. *International Journal of Production Research*, 36(1):141–156, 1998. ISSN 1366588X. doi: 10.1080/002075498193976. URL https://www.tandfonline.com/action/journalInformation?journalCode=tprs20.

I. Leksic, N. Stefanic, and I. Veza. The impact of using different lean manufacturing tools on waste reduction. *Advances in Production Engineering And Management*, 15(1):81–92, 2020. ISSN 18556531. doi: 10.14743/APEM2020.1.351. URL https://doi.org/10.14743/apem2020.1.351.

Cheng Dar Liou and Yi Chih Hsieh. A hybrid algorithm for the multi-stage flow shop group scheduling with sequence-dependent setup and transportation times. *International Journal of Production Economics*, 170:258–267, 2015. ISSN 09255273. doi: 10.1016/j.ijpe.2015.10.002. URL http://dx.doi.org/10.1016/j.ijpe.2015.10.002.

Jiyin Liu and Colin R Reeves. P Constructive and composite heuristic solutions to the P// C. *European Journal Of Operational Research*, 132, 2001. URL www.elsevier.com/locate/dsw.

Peter B. Luh, Ling Gou, Yuanhui Zhang, Takaaki Nagahora, Makoto Tsuji, Kiyoshi Yoneda, Tetsuo Hasegawa, Yuji Kyoya, and Toshiyuki Kano. Job shop scheduling with group-dependent setups, finite buffers, and long time horizon. *Annals of Operations Research*, 76:233–259, 1998. ISSN 15729338. doi: 10.1023/a:1018948621875.

Sampatraj S. Mandroli, Abhishek K. Shrivastava, and Yu Ding. A survey of inspection strategy and sensor distribution studies in discrete-part manufacturing processes. *IIE Transactions (Institute of Industrial Engineers)*, 38(4):309–328, apr 2006. ISSN 0740817X. doi: 10.1080/07408170500327352. URL https://www.tandfonline.com/doi/abs/10.1080/07408170500327352?casa_token=RnkRXFn9df0AAAAA:m8QrpontkQrrerYD1PpvVG_0Kg8CsyBzrvXmDEzvn-4a4jg-hFdHmXH_M16P3u6nFP5WCfT0tmogFw.

Ehud Menipaz. A taxonomy of economically based quality control procedures. *International Journal of Production Research*, 16(2):153–167, 1978. ISSN 1366588X. doi: 10.1080/00207547808930010. URL https://doi.org/10.1080/00207547808930010.

Rym M'Hallah and R. L. Bulfin. Minimizing the weighted number of tardy jobs on parallel processors. Technical Report 2, 2005. URL www.sciencedirect.comwww.elsevier.com/locate/dsw.

Melanie Mitchell. Genetic Algorithms: An Overview 1. *Complexity*, 1(1):31–39, 1995.

Joseph A Montagna. Yale New Haven Teachers Institute The Industrial Revolution by. pages 1–12, 2015.

Tadahiko Murata, Hisao Ishibuchi, and Hideo Tanaka. Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers and Industrial Engineering*, 30(4):957–968, sep 1996. ISSN 03608352. doi: 10.1016/0360-8352(96)00045-9.

Hiroshi O. Ohta. An application of sampling inspection to corrective plan for semi-markov production process. *AIIE Transactions*, 6(2):151–158, 1974. ISSN 05695554. doi: 10.1080/05695557408974947. URL https://doi.org/10.1080/05695557408974947.

Byung Joo Park, Hyung Rim Choi, and Hyun Soo Kim. A hybrid genetic algorithm for the job shop scheduling problems. *Computers and Industrial Engineering*, 45(4):597–613, 2003. ISSN 03608352. doi: 10.1016/S0360-8352(03)00077-9. URL www.elsevier.com/locate/dsw.

Michael H. Peters and William W. Williams. Location of Quality Inspection Stations: an Experimental Assessment of Five Normative Heuristics. *Decision Sciences*, 15(3):389–408, jul 1984. ISSN 15405915. doi: 10.1111/j.1540-5915.1984.tb01225.x. URL https://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-5915.1984.tb01225.xhttps://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-5915.1984.tb01225.xhttps://onlinelibrary.wiley.com/doi/10.1111/j.1540-5915.1984.tb01225.x.

Thomas Pyzdek. The Six Sigma Handbook - A Complete Guide for Green Belts, Black Belts, and Managers at All Levels. *Search*, page 830, 2003. ISSN 0040-1706.

Gad Rabinowitz and Hamilton Emmons. Optimal and heuristic inspection schedules for multistage production systems. *IIE Transactions (Institute of Industrial Engineers)*, 29(12):1063–1071, 1997. ISSN 15458830. doi: 10.1080/07408179708966425.

Chandrasekharan Rajendran and Hans Ziegler. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155(2):426–438, 2004. ISSN 03772217. doi: 10.1016/S0377-2217(02)00908-6. URL www.elsevier.com/locate/dsw.

Tzvi Raz. A Survey of Models for Allocating Inspection Effort in Multistage Production Systems. *Journal of Quality Technology*, 18(4):239–247, oct 1986. ISSN 0022-4065. doi: 10.1080/00224065.1986.11979018. URL https://www.tandfonline.com/doi/abs/10.1080/00224065.1986.11979018.

Tzvi Raz and Moshe Kaspi. Location and sequencing of imperfect inspection operations in serial multi-stage production systems. *International Journal of Production Research*, 29(8):1645–1659, 1991. ISSN 1366588X. doi: 10.1080/00207549108948037. URL https://doi.org/10.1080/00207549108948037.

Tzvi Raz and Marlin U. Thomas. A method for sequencing inspection activities subject to errors. *IIE Transactions (Institute of Industrial Engineers)*, 15(1):12–18, 1983. ISSN 15458830. doi: 10.1080/05695558308974608. URL https://doi.org/10.1080/05695558308974608.

Mohammad Rezaei-Malek, Mehrdad Mohammadi, Jean Yves Dantan, Ali Siadat, and Reza Tavakkoli-Moghaddam. A review on optimisation of part quality inspection planning in a multi-stage manufacturing system. *International Journal of Production Research*, 57(15-16):4880–4897, 2019. ISSN 1366588X. doi: 10.1080/00207543.2018.1464231. URL https://www.tandfonline.com/doi/abs/10.1080/00207543.2018.1464231.

Frederick A. Rodammer and K. Preston White. A Recent Survey of Production Scheduling. Technical Report 6, 1988.

Kenneth A Ryan. THE RE-INNOVATION OF FORD MOTOR COMPANY TO A SUSTAINABLE LEAN ENTERPRISE By Program in Engineering Management. *Electronic Theses and Dissertations*, aug 2006. doi: 10.18297/etd/1243. URL https://ir.library.louisville.edu/etd/1243.

Arun Kumar Sangaiah, Mohsen Yaghoubi Suraki, Mehdi Sadeghilalimi, Seyed Mostafa Bozorgi, Ali Asghar Rahmani Hosseinabadi, and Jin Wang. A new meta-heuristic algorithm for solving the flexible dynamic job-shop problem with parallel machines. *Symmetry*, 11(2), 2019. ISSN 20738994. doi: 10.3390/sym11020165. URL https://www.mdpi.com/405240.

Serpil Sayin and Selçuk Karabati. A bicriteria approach to the two-machine flow shop scheduling problem. *European Journal of Operational Research*, 113(2): 435–449, mar 1999. ISSN 03772217. doi: 10.1016/S0377-2217(98)00009-5.

Wan S. Shin, Stephen M. Hartw, and Heungsoon Felix Lee. Strategic allocation of inspection stations for a flow assembly line: A hybrid procedure. *IIE Transactions (Institute of Industrial Engineers)*, 27(6):707–715, 1995. ISSN 15458830. doi: 10.1080/07408179508936787. URL https://www.tandfonline.com/doi/abs/10.1080/07408179508936787.

Wenchi Shou, Jun Wang, Peng Wu, and Xiangyu Wang. Value adding and non-value adding activities in turnaround maintenance process: classification, validation, and benefits. *Production Planning and Control*, 31(1):60–77, 2020. ISSN 13665871. doi: 10.1080/09537287.2019.1629038. URL https://www.tandfonline.com/action/journalInformation?journalCode=tppc20.

Wilfrido Quiñones Sinisterra and Cristiano Alexandre Virgínio Cavalcante. An integrated model of production scheduling and inspection planning for resumable jobs. *International Journal of Production Economics*, 227, 2020. ISSN 09255273. doi: 10.1016/j.ijpe.2020.107668. URL https://doi.org/10.1016/j.ijpe.2020.107668.

M. Solimanpur, P. Vrat, and R. Shankar. A neuro-tabu search heuristic for the flow shop scheduling problem. *Computers and Operations Research*, 31(13): 2151–2164, nov 2004. ISSN 03050548. doi: 10.1016/S0305-0548(03)00169-2.

Paul Stief, Jean-Yves Dantan, Alain Etienne, and Ali Siadat. ScienceDirect ScienceDirect A new methodology to analyze the functional and physical architecture of existing products for an assembly oriented product family identification. 2018. doi: 10.1016/j.procir.2020.04.069. URL www.sciencedirect.comAvailableonlineatwww.sciencedirect.comwww.elsevier.com/locate/procedia2212-8271.

Pratya Poeri Suryadhini, Sukoyo Sukoyo, Suprayogi Suprayogi, and Abdul Hakim Halim. A batch scheduling model for a three-stage flow shop with job and batch processors considering a sampling inspection to minimize expected total actual flow time. *Journal of Industrial Engineering and Management*, 14(3):520–537, 2021. ISSN 20130953. doi: 10.3926/JIEM.3438. URL https://doi.org/10.3926/jiem.3438.

Mukesh Taneja and N. Viswanadham. Inspection allocation in manufacturing systems: a genetic algorithm approach. *Proceedings - IEEE International Conference on Robotics and Automation*, (pt 4):3537–3542, 1994. ISSN 10504729. doi: 10.1109/robot.1994.351027.

Lixin Tang, Wenxin Liu, and Jiyin Liu. A neural network model and algorithm for the hybrid flow shop scheduling problem in a dynamic environment. *Journal of Intelligent Manufacturing*, 16(3):361–370, 2005. ISSN 09565515. doi: 10.1007/s10845-005-7029-0.

R. Tavakkoli-Moghaddam, A. R. Rahimi-Vahed, and A. H. Mirzaei. Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm. *International Journal of Advanced Manufacturing Technology*, 36(9-10):969–981, jan 2008. ISSN 02683768. doi: 10.1007/s00170-006-0906-7. URL https://link.springer.com/article/10.1007/s00170-006-0906-7.

S N Teli and Umesh Bhushi. Role of Cost of Quality in the Automotive Industry Framework for Cost of Quality View project. 2010. URL https://www.researchgate.net/publication/317428436.

Berkin Toktaş, Meral Azizoğlu, and Suna Kondakci Köksalan. Two-machine flow shop scheduling with two criteria: Maximum earliness and makespan. *European Journal of Operational Research*, 157(2):286–295, 2004. ISSN 03772217. doi: 10.1016/S0377-2217(03)00192-9. URL www.elsevier.com/locate/dsw.

E. Trovato, P. Castagliola, G. Celano, and S. Fichera. Economic design of inspection strategies to monitor dispersion in short production runs. *Computers and Industrial Engineering*, 59(4):887–897, nov 2010. ISSN 03608352. doi: 10.1016/j.cie.2010.08.019.

Ergin Uskup and Spencer B. Smith. a Branch-and-Bound Algorithm for Two-Stage Production-Sequencing Problems. *Operations Research*, 23(1):118–136, feb 1975. ISSN 0030364X. doi: 10.1287/opre.23.1.118. URL https://pubsonline.informs.org/doi/abs/10.1287/opre.23.1.118.

M. H. Veatch. Inspection strategies for multistage production systems with time-varying quality. *International Journal of Production Research*, 38(4):837–853, 2000. ISSN 00207543. doi: 10.1080/002075400189185. URL https://www.tandfonline.com/action/journalInformation?journalCode=tprs20.

Adan Verduzco, J. Rene Villalobos, and Benjamin Vega. Information-based inspection allocation for real-time inspection systems. *Journal of Manufacturing Systems*, 20(1):13–22, jan 2001. ISSN 02786125. doi: 10.1016/S0278-6125(01)80016-9. URL https://linkinghub.elsevier.com/retrieve/pii/S0278612501800169.

N. Viswanadham, Shashi M. Sharma, and Mukesh Taneja. Inspection allocation in manufacturing systems using stochastic search techniques. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 26(2):222–230, 1996. ISSN 10834427. doi: 10.1109/3468.485748. URL https://ieeexplore.ieee.org/abstract/

document/485748/?casa_token=7LTvTVQYXFkAAAAA:juo2xp5WXH3AEoh_CHbt7NicR6rCB-XZ39fEoJuF6bOOU11_TTa9GHGrbLO-lqatPkE2mF-xh5E.

Huiying Wang, Wenbin Wang, and Rui Peng. A two-phase inspection model for a single component system with three-stage degradation. *Reliability Engineering and System Safety*, 158:31–40, 2017. ISSN 09518320. doi: 10.1016/j.ress.2016.10.005. URL http://dx.doi.org/10.1016/j.ress.2016.10.005.

Ji Bo Wang, C. T. Daniel Ng, T. C.E. Cheng, and Li Li Liu. Minimizing total completion time in a two-machine flow shop with deteriorating jobs. *Applied Mathematics and Computation*, 180(1):185–193, 2006. ISSN 00963003. doi: 10.1016/j.amc.2005.11.162. URL www.elsevier.com/locate/amc.

Leon S. White. Shortest Route Models for the Allocation of Inspection Effort on a Production Line. *Management Science*, 15(5):249–259, jan 1969. ISSN 0025-1909. doi: 10.1287/mnsc.15.5.249. URL https://pubsonline.informs.org/doi/abs/10.1287/mnsc.15.5.249.

William Winchell. Inspection and Measurement in Manufacturing: Keys to Process Planning and Improvement, 1996. URL https://books.google.ca/books?hl=en&lr=&id=be9ZWBfMpTgC&oi=fnd&pg=PR13&dq=Winchell,+W.+(1996).+Inspection+and+measurement+in+manufacturing:+keys+to+process+planning+and+improvement.+Society+of+Manufacturing+Engineers&ots=dcdmy1pJBl&sig=xiHHB1IW3g5D78omRA4.

Wendong Xiao, Peifeng Hao, Sen Zhang, and Xinhe Xu. Hybrid flow shop scheduling using genetic algorithms. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 1:537–541, 2000. doi: 10.1109/wcica.2000.860026. URL https://ieeexplore.ieee.org/abstract/document/860026/?casa_token=

Cifd-Vk0TDcAAAAA:Hyy0nhwikKAP5MlSPfYQSDSzkAF3VWMfQMsmML_-ABnTyv_Ztqj8i5da8tUQnXp6UShSKbQ2UH8.

Kuo Ching Ying and Ching Jong Liao. An ant colony system for permutation flow-shop sequencing. *Computers and Operations Research*, 31(5):791–801, 2004. ISSN 03050548. doi: 10.1016/S0305-0548(03)00038-8. URL www.sciencedirect.comwww.elsevier.com/locate/dsw.

Yingchao You, Guijiang Duan, Rui Liu, Taotao Liu, and Mingcong Huang. Inspection Allocation Optimization with Resource Constraints Based on Modified NSGA-II in Flexible Manufacturing Systems. *Mathematical Problems in Engineering*, 2021, 2021. ISSN 15635147. doi: 10.1155/2021/3976576.

Yoshiki Yumbe, Misa Miyakoshi, Masaharu Kondo, Toshiyuki Arao, and Naohiro Furukawa. Evaluation of Optimization Method for Inspection Scheduling of Power Distribution Facilities Using Maintenance Data Accumulated by Power Utility. *IEEE Transactions on Power Delivery*, 32(2):696–702, apr 2017. ISSN 08858977. doi: 10.1109/TPWRD.2016.2618936.