2022

Bottleneck Management through Strategic Sequencing in Smart Manufacturing Systems

Sayantee Roy
West Virginia University, sr00031@mix.wvu.edu

Bottleneck Management through Strategic Sequencing in Smart Manufacturing Systems

Sayantee Roy

Thesis submitted

to the Benjamin M. Statler College of Engineering and Mineral Resources

at West Virginia University

in partial fulfilment of the requirements for the degree of

Master of Science in

Department of Industrial and Management Systems Engineering

Thorsten Wuest, Ph.D., Co-Chair

Imtiaz Ahmed, Ph.D., Co-Chair

John Saldanha, Ph.D.

Department of Industrial and Management Systems Engineering

Morgantown, West Virginia

2022

Keywords: Sequencing, Bottleneck, Makespan, Idle Time, Simulation, Processing Time

ABSTRACT

Bottleneck Management through Strategic Sequencing in Smart Manufacturing Systems

Sayantee Roy

Nowadays, industries put a significant emphasis on finding the optimum order for carrying out jobs in sequence. This is a crucial element in determining net productivity. Depending on the demand criterion, all production systems, including flexible manufacturing systems, follow a predefined sequence of job-based machine operations. The complexity of the problem increases with increasing machines and jobs to sequence, demanding the use of an appropriate sequencing technique. The major contribution of this work is to modify an existing algorithm with a very unusual machine setup and find the optimal sequence which will really minimize the makespan. This custom machine setup completes all tasks by maintaining precedence and satisfying all other constraints. This thesis concentrates on identifying the most effective technique of sequencing which will be validated in a lab environment and a simulated environment. It illustrates some of the key methods of addressing a circular non permutation flow shop sequencing problem with some additional constraints. Additionally, comparisons among the various heuristics algorithms are presented based on different sequencing criteria. The optimum sequence is provided as an input to a real-life machine set up and a simulated environment for selecting the best performing algorithm which is the basic goal of this research. To achieve this goal, at first, a code using python programming language was generated to find an optimum sequence. By analyzing the results, the makespan is increasing with the number of jobs but additional pallet constraint shows, adding more pallets will help to reduce makespan for both flow shops and job shops. Though the sequence obtained from both algorithms is different, for flow shops the makespan remains same for both cases but in the job shop scenario Nawaz, Enscore and Ham (NEH) algorithms always perform better than Campbell Dudek Smith (CDS) algorithms. For job shops with different combinations the makespan decreases mostly for maximum percentage of easy category jobs combined with equal percentage of medium and complex category jobs.

Table of Contents

List of Figures

List of Tables

List of Graphs

CHAPTER 1

INTRODUCTION

Manufacturers must fulfill orders quickly and on time in the competitive marketplaces of today. Failing to do so, at the minimum, may result in a considerable loss of goodwill. Sequencing is one of the operations that manufacturers use to complete tasks on time and produce goods within the lead time, among many other approaches that make manufacturers competent in the commercial world (Kassu & Eshetie, 2015). Customer happiness is the result of organizations. This calls for an efficient sequencing system that distributes limited resources. To meet customer needs, sequencing is the process of assigning shared resources over a period of time to competing tasks (Sumathi et al., 2015).

(Al-Harkan, 2010) categorized sequencing contexts based on the types of information:

1. The jobs and procedures to be carried out

2. The quantity and variety of machines in the shop

3. The field that places limitations on how assignments can be made

4. The criteria used to evaluate a sequence

The sequencing issue in the manufacturing sector is characterized by the placement of tasks on machines and determining the ideal order for those tasks to be performed in order to maximize a predetermined criterion. The path to be traveled in the workshop's production cycle for each job is defined by a group of tasks. In the industrial process, each machine has a running period for each work that is launched (Belabid et al., 2020a).

Several classifications are distinguished:

- **Single Machine:** In this sequencing method, each machine's jobs are first sorted in a specific order. When jobs are arranged according to process time, which means that jobs with shorter process times are more advanced than those with longer process times, the goal of lowering the makespan is best achieved (Das, 2014).

- **Flow Shop:** The following options are available for processing work in the flow shop's m machines in series:
  - ✓ <u>Permutation Flow Shop:</u> The fundamental idea behind a permutation flow shop is that each machine accepts the jobs in a specific order, requiring that every work be completed on every machine in the flow shop environment. Additionally, each machine has a unique processing time for each job. As a result, when analyzing a flow shop problem, various possible workflows are taken into account, and the best one is ultimately selected.
  - ✓ <u>Non-permutation Flow Shop:</u> Jobs are processed by a series of m machines, not always in the same order, in a non-permutational flow shop.

- **Job Shop:** The primary distinction between job shop sequencing and flow shop sequencing is that, in contrast to flow shop, here it is not required that all projects be completed on every machine available; rather, each unique job may be completed on any unique number of machines as necessary.

  The optimal sequence out of a number of potential sequences is chosen in this case based on the requirements of the problem. Jobs can be processed using one of the following methods on one of the m machines according to (Al-Harkan, 2010):
  - ✓ Assembly Job Shop: Assembly job shop is one that does jobs that require at least two component items and one assembly step.
  - ✓ Hybrid Job Shop: Some jobs' operations are prioritized in the same sequence.
  - ✓ Hybrid Assembly Job Shop: The characteristics of both an assembly and a hybrid workshop are combined in a hybrid assembly job shop.

- **General shop:** A general shop is one in which all jobs can be processed in any order. There are m machines, and there are no restrictions on how each job can be routed through them. For any job, there is no predetermined flow pattern.

The performance of heuristics in sequencing is directly impacted by job distinction and sequencing. Based on job sequencing, numerous effective sequencing rules and algorithms have been created, including the Shortest Processing Time rule (SPT), the Longest Processing Time rule (LPT), the First Come First Serve rule (FCFS), the Page algorithm (Page, 1961), the NEH heuristic (Nawaz et al., 1983), the Palmer algorithm (Drezner, 1987), etc. Prioritizing jobs still remains to be challenging in flow shop sequencing, particularly for mass customized production of today (W. Liu et al., 2017).

The bottleneck is one of the highly debated topics in industrial systems. The production line's bottlenecks and waiting periods are being addressed by managers and engineers. Additionally, industrial firms are working to maintain their competitiveness by reducing bottlenecks, raising productivity, and lowering overall costs. One of the helpful techniques for assessing various manufacturing implementation scenarios to increase productivity and reduce bottlenecks is computer simulation. Additionally, there are numerous benefits to using computer simulation in various contexts, such as industrial processes, building projects, etc., including reduced costs, improved throughput, increased resource utilization, and decreased cycle times (Zahraee et al., 2014). The planning and control of manufacturing is one of the components of lean manufacturing. This element focuses on methods on how to maximize the utilization of the resources, minimize waste and improve the quality of both processes and products (Salihah et al., 2018). Numerous manufacturing and service sectors employ sequencing on a regular basis as a key decision-making process. It involves allocating resources among jobs within set time constraints with the intention of maximizing one or more goals. An organization's tasks and resources might take on a variety of shapes. The resources might include tools at a workshop, airport runways, laborers working on a construction site, processing capacity in a data center, and so on. Tasks could also involve running computer programs, taking part in industrial process operations, landing and taking off at airports, and other activities. Each work may have a certain priority level, the earliest time it can begin, and a due date. The goals can also be expressed in a variety of ways. Reducing the number of assignments that are finished after their respective due dates may be one goal. Sequencing is a key decision-making process in the majority of manufacturing and production systems as well as in the majority of information processing environments. Additionally, it is crucial in the context of transportation and distribution, as well as in other service-related industries. The short-term sequences show the best order and timing for processing jobs together with timelines for all other resources needed to support the production plan, such as jobs, equipment, staff, supplies, and facilities (Das, 2014). The sequences should utilize the resources as effectively as possible to ensure low costs and high utilization. Other goals of sequencing include reducing consumer wait times for products, meeting delivery deadlines, maintaining low stock levels, providing preferred working sequences, and reducing patient wait times in hospitals for various sorts of examinations, among others (Uzorh & Innocent, 2014).

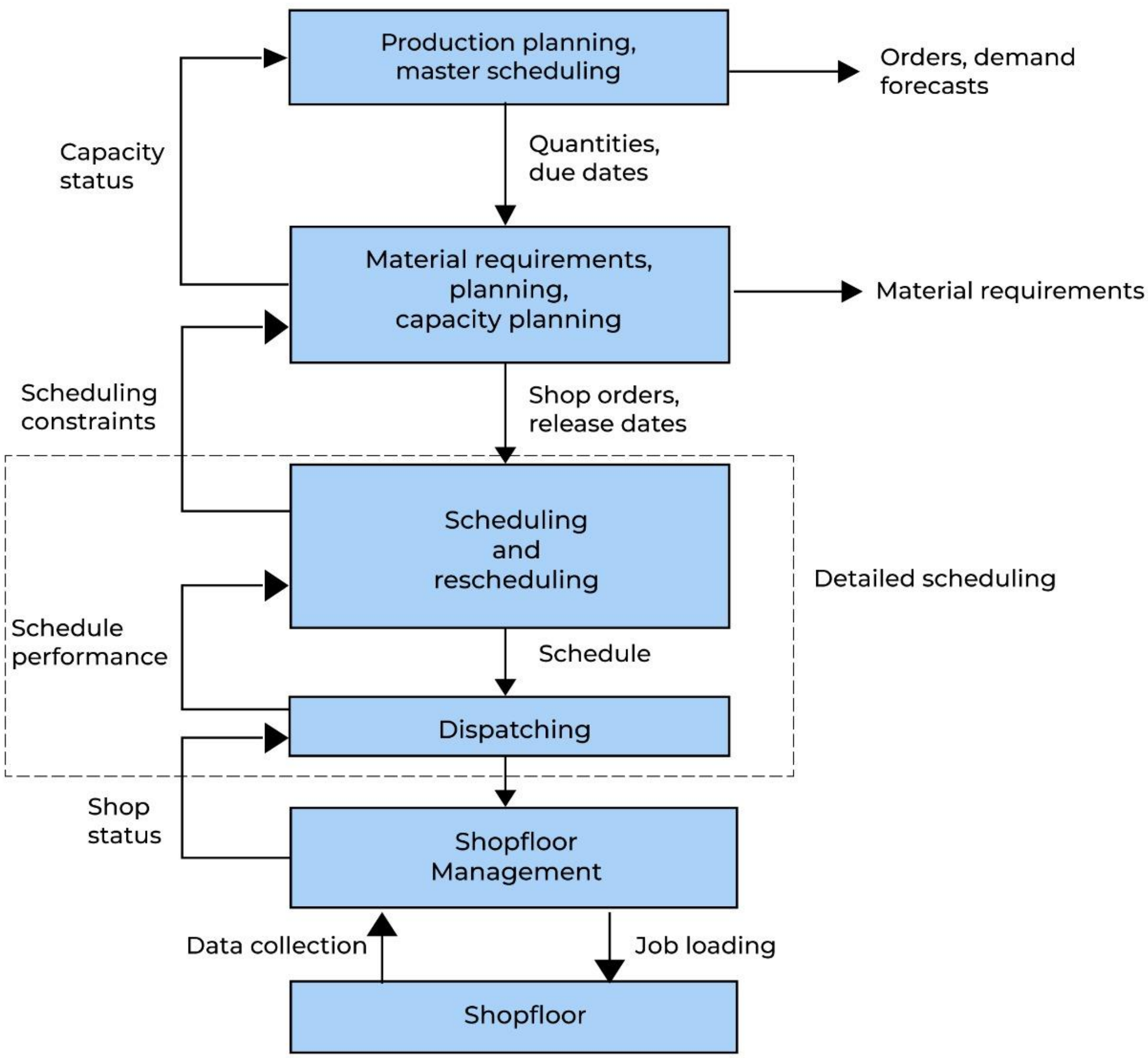


Figure 1.1 Information flow diagram in a manufacturing system (Das, 2014)

The basic elements of a system are resources and tasks. The resources (machines) are required to perform a service on the tasks and are classified as being of one or several types. Single-stage resources are those of a single type, while multiple-stage or multistage resources are those of several different types. Both types of resources can be available for parallel or serial processing of the tasks. When more than one resource is available for performing the same set of tasks, then such resources are called parallel processors. On the other hand, resources are referred to as serial processors if a task is multistage in nature and needs more than one processor for subsequent actions. If the set of tasks available for sequencing does not change over time, the system is considered static. However, the system is regarded as dynamic if the list of available tasks does change over time. Additionally, a problem is referred to as deterministic if its parameters (such as processing times, arrival timings, due dates, etc.) are known in advance. Otherwise, it is called a stochastic problem. Whichever type of system is under study, there are common measures of performance. These performance evaluations are typically one-dimensional and include data from all jobs. Some examples of measures of performance are machine utilization, maximum job completion time (makespan), waiting time (as a function of tardiness, due dates, or setup times, etc.).

The concept of assembly lines has been introduced for mass production as well as cost efficiency. The multiple steps are carried out on the products as they pass through the system on assembly lines, which are flow-oriented mass production systems. These workstations are typically arranged in a series, parallel, U-shape, and two-sided configuration. To accomplish assembly processes, a transportation system (such as conveyor belts, robotic arms, or automatic guided vehicles) is employed to carry the product from one workstation to another. Cycle time is the amount of time it takes for each workstation to complete various operations on the product. Assembly lines are categorized into three categories according to (Mönch et al., 2021):

- Single Model Assembly Lines (SMALs)
- Multi Model Assembly Lines (MuMALs)
- Mixed Model Assembly Lines (MMALs)

The environment of mass customization makes extensive use of MMALs. The need for customized products is growing rapidly, encouraging industries to adopt MMALs. This naturally resolves MuMAL's batch sizing and inventory issues. Additionally, MMALs require significantly less investment than distinct SMALs, but at the expense of added complexity. On the other hand, inefficient usage of MMAL increases cycle time and production cost per unit. In MMALs, there are two significant challenges:

- <u>Model Sequencing Problem (MSP):</u> Sequence/order of various models is determined while optimizing the makespan, flow time, cycle time, idle time, and lateness etc.
- <u>Assembly Line Balancing Problem (ALBP):</u> Various tasks/operations are assigned to different stations while optimizing cycle time, balance efficiency and smoothness index etc.

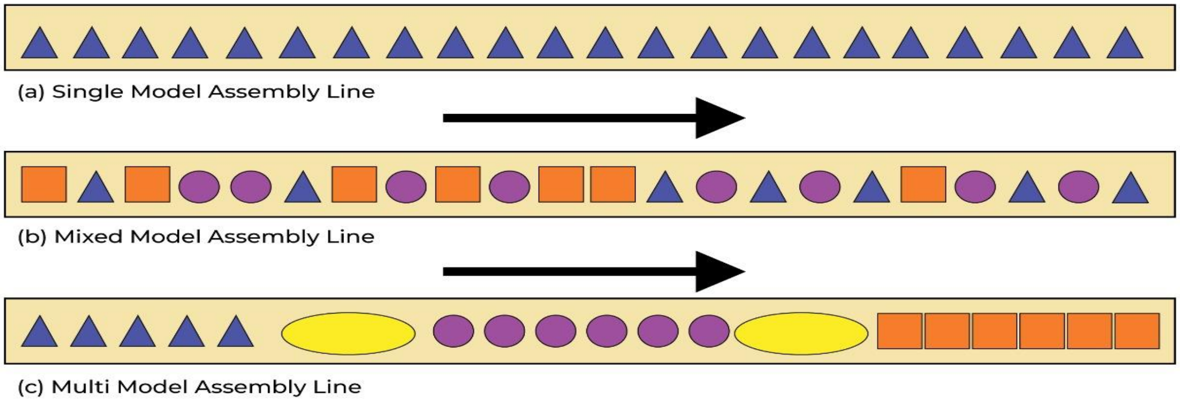


Figure 1.2: Assembly lines for single and multiple products (Haque et al., 2018)

According to (Baker, 1984) there are decision making goals in sequencing:

- Efficient utilization of resources: Arrange all activities to maintain high utilization of people, tools, and space.
- Rapid response to demands: Job processing should be facilitated by sequencing so that there is little work-in-progress inventory.
- Close conformance to meet deadlines: By utilizing shorter lead times, sequencing should guarantee that deadlines are always met.

The research addresses the problem of resource utilization when n jobs, each made up of m processes, need to be processed non-preemptively on m machines. Each job must be completed in a different order (i.e., routing does not differ from one job to another). Despite a machine only being capable of processing one job at a time, the capacity of the intermediate buffer space is infinite. (Sauvey & Sauer, 2020). The goal is to reduce the overall completion time, often known as makespan. The difficulty is determining a sequence (a throughput order of jobs in the constant suite of machines). This sequence needs to be precise (as close to the ideal makespan as possible) and easily attainable.

The structure of this article is as follows. The state of the art is discussed, and the problem definition is provided in Section 2. In Section 3, the research objective is defined and explained. In Section 4, research methodology is developed including a general theoretical framework. Section 5 deals with the development of the data collection and validation of the model. Section 6 is all about result analysis and discussions. Finally, Section 7, concludes this paper, and perspectives are given for this research work.

CHAPTER 2

STATE OF THE ART & PROBLEM DEFINITION

This chapter provides a description of the context of reference to which this research is related to. To understand it correctly, a review of the existing literature on the topic has been performed. Moreover, the state of the art for sequencing in smart manufacturing systems has been defined to find the current gap and issues that this research aims at covering by proposing a new type of solution.

2.1 Overall Throughput

The methods created using Overall Throughput Effectiveness (OTE) (Muthiah et al., 2008) to automate the factory-level performance monitoring and diagnostics process are described. The algorithms are included in a software program called SIMPRO that enables factory specialists to track performance and carry out factory-level diagnostics, locating bottlenecks and untapped potential to create a methodical improvement strategy. The impact of an improvement plan may be immediately evaluated, and the return on investment can be computed, when cost data is accessible. A case study in glass manufacturing is presented to demonstrate automated performance diagnostics and the benefits of employing the method.

The Throughput Maximization Strategy (TMS) proposed by (Adhianto et al., 2010) contains two different approaches for scheduling transaction-intensive operations at the instance and task levels. For instance, the opposite average load algorithm is used for scheduling, but the extended min-min algorithm is utilized for task scheduling. By maximizing the rate of resource usage within each local autonomous group, the second method, extended min-min, aims to further maximize total throughput at the task level. It was explained that these two algorithms are theoretically superior to their respective original algorithms. By seeking overall load balance at the instance level, the first algorithm, opposite average load, aims to maximize throughput overall. Our approach can greatly increase overall throughput when scheduling transaction-intensive workflows over existing scheduling methods, according to comparison and simulation results on Swinburne Decentralized Workflow for Grid (SwinDeW-G), a peer-to-peer-based grid workflow environment.

For factory-level performance monitoring and diagnosis, the developed Overall Throughput Effectiveness (OTE) measurements are presented in (Muthiah & Huang, 2007). Examples are used to explain the concept of OTE development and the OTE validation methodology. A sensitivity study is performed to confirm the bottleneck detection methodology utilizing OTE, and the overall factory effectiveness (OFE) computation and bottleneck detection methods are shown. Finally, two case studies highlight OTE's diagnostic ability while also explaining the OTE development process, validating the generated OTE measures, and demonstrating its development technique. It was discovered that productivity bottlenecks and improvement possibilities can be measured through the use of OTE in a case study involving a wafer fab and the production of glass.

2.2 Industry 4.0 and Bottleneck

In order to prioritize process improvement efforts, (Ongbali et al., 2021) makes an effort to discover and analyze numerous variables in order to locate the critical variables that influence and turn manufacturing elements into bottleneck problems. The respondents' scores were compiled into a (m x n) data matrix that was used as an input variable into the factor analysis model, and the data matrix was evaluated using StatistiXL software. The 95 percent confidence level was chosen to ensure adequate representation of the population and to validate the data for the study. According to the results, the "process capability index" accounts for 25% of the variables studied, making it the main bottleneck factor. Equipment failure, operations, material scarcity, and market demand, among other factors, all have an impact on the cluster. Based on the size of their respective variable factor loadings, such as random events, raw material flow, process technology, and random environmental factors, manufacturing process restraint, resources, weather, communication, logistics, and line dedication are other important aspects. The issues identified and solutions proposed in this study are generic, and improvement efforts should concentrate on fixing the primary variables while not ignoring the minor and weaker variables, even when bottleneck challenges vary from one manufacturing system to another.

(Prasetyo & Veroya, 2020) offer a conceptual framework for implementing OEE using the DMAIC method of lean six sigma in a bottleneck process of a multinational semiconductor company in the Philippines. The researcher hypothesizes that the application of DMAIC approach integrated with OEE will increase the bottleneck process' productivity indices. Both classical and modern approaches have been taken into consideration and synthesized to create the suggested framework, which uses lean six sigma tools like FMEA, DOE, SMED, Visual Management, and Mistake Proofing.

To assist developing nations in improving their management practices, (Cezarino et al., 2019) seek to look into how the ideas of Industry 4.0 and Circular Economy (CE) relate to one another. By researching scientific production patterns to understand the interface of both constructions, attempts are made to identify obstacles to industry 4.0 and CE adoption in Brazil. They also make unique framework and tactical suggestions to get around obstacles for developing nations.

(Lanke et al., 2016) introduced the Mine Production Index, MPI, which is an operational measure that helps to accurately spot bottlenecks in mine production processes with an extension of the OEE idea that can be applied more clearly in the field. Assigning different weights to the OEE components aids engineers in detecting the bottleneck and determining the main cause in a timely and accurate manner. From the perspective of MPI evaluation, the crusher is the production bottleneck machine in a Swedish open pit mine. With the mine production department and management, the validity of the crusher being a bottleneck for the study period was established. The performance of the crusher can be significantly related with the reason for it being a bottleneck, according to the weights assigned and MPI value calculated. The performance of a crusher appears to be linked to its design parameter, as well as scheduling and planning. An algorithm that combines MPI to find bottlenecks in continuous mining operations has been proposed as a methodical technique.

Based on analytically discovered bottlenecks, (Wedel et al., 2016) propose and test effective fault repair prioritizing approaches. The three novel approaches are aimed at detecting short-term and real-time bottlenecks, as well as near-future constraints, and their effectiveness is demonstrated using a simulation model of a real automobile machining line.

Based on the novel SCORE (Simulation-based COnstraint REmoval)-method, which treats bottleneck identification and improvement as a multi-objective optimization problem for identifying the optimal (minimal) number of changes to maximize throughput, (Bernedixen et al., 2016) shows how a generic way of defining improvements of the decision variables, in terms of processing times, availabilities, and repair times, can largely automate the analysis process. It is important to recognize the value of such automated assistance for users, who are typically simulation or production engineers, as manually defining variables is not only time consuming but also prone to error. The effectiveness of the automated SCORE-analysis process is demonstrated by small academic research and its application to a challenging real-world industrial improvement project. Furthermore, when compared to other bottleneck detection techniques like machine utilization monitoring and shifting bottleneck detection, the results of both trials vividly illustrated the advantages of SCORE.

(Roser et al., 2001) offer a unique method for determining the bottleneck in a discrete event system by looking at the average duration of a machine's active time for all machines. The bottleneck is defined as the machine with the longest average unbroken active period, and the method is generally applicable and capable of assessing complicated and sophisticated systems. The results are extremely accurate, with a high level of confidence in discriminating between bottleneck and non-bottleneck machines. This method is simple to use and can be integrated into existing simulation tools with minimal effort, requiring only the analysis of a log file, which practically all simulation systems have readily available. This approach not only meets academic requirements for correctness, but also meets industry requirements for use.

2.3 Bottleneck in Different Systems

(Subramaniyan et al., 2018) provide a data-driven approach to throughput bottleneck analysis based on the active period theory that integrates the machine data from the manufacturing execution systems (MES) and evaluates the statistical significance of any bottlenecks found. Throughput can be increased by automating the algorithm to enable data-driven decision making on the work floor. Using an interdisciplinary strategy focused on production and data sciences, the algorithm was created and evaluated using real-world MES datasets, generating research outputs helpful to manufacturing businesses and expanding throughput bottleneck analysis standards.

(Li et al., 2009) present a new data-driven technique for both short and long-term throughput bottleneck diagnosis. The method employs production line blockage and starvation probability, as well as buffer content statistics, to pinpoint production bottlenecks without constructing an analytical or simulation model.

(Out, 2010) contrast their developed shifting bottleneck detection method with the two most popular bottleneck identification techniques for AGV systems, which are based on utilization and waiting time. The two standard approaches are significantly flawed when compared to the shifting bottleneck detection method because the latter not only discovers bottlenecks but also calculates their magnitude. The two conventional methods either fail to detect bottlenecks at all or detect them inaccurately. It was found that none of the bottleneck identification techniques—utilization, waiting time, and switching in all respects. While the utilization and waiting time approaches have certain restrictions in terms of usage and accuracy, the shifting bottleneck detection method has the drawback of being slightly more challenging to execute than the other two methods. The shifting bottleneck identification method performs better than the others overall, and it has been incorporated into a program called GAROPS analyzer. This program analyzes data from the GAROPS simulation and identifies bottlenecks, which are then displayed in an understandable MS Excel spreadsheet.

A brand-new technique for identifying bottlenecks in manufacturing systems and relocating them is put forth by (Roser et al., 2002). Every industrial system has one or more bottlenecks and eliminating them would improve the system. Finding the bottleneck is a challenging task, though, as the system may change over time or due to unexpected events, moving the bottleneck from one machine to another. The shifting bottleneck detection method, which is extremely reliable, simple to implement, and capable of identifying primary and secondary bottlenecks in a variety of production systems, as well as enabling simulation to anticipate bottlenecks for both steady state and variable systems, establishes the bottleneck based on the length of time a machine is running uninterruptedly. Making choices about how to distribute the available resources can be improved by considering the likelihood that a machine will be the bottleneck.

2.4 Flow Production and Bottleneck

(Urban & Rogowska, 2020) solves a critical problem in the manufacturing sector: the location of a bottleneck. It aims to provide a detailed analysis of bottleneck identification methods based on a thorough literature review, as well as the design of a generalized methodology for bottleneck identification in the production system using a combination of a narrative and scoping literature review, as well as logical design. Several approaches to finding bottlenecks are compared and studied, with some producing the same results while others offering distinctive insights into the production system under examination. Various processes arranged in logical steps are included in a technique for bottleneck identification that should be used when seeking to locate the bottleneck in a production system. The suggested method is likely to be helpful in implementing the Theory of Constraints (TOC) for locating bottlenecks in a production system and will be a helpful resource for managers and TOC experts. The proposed bottleneck detection methodology is an original proposal based on recent literature output that advances production management theory as a useful managerial tool, however it is still a theoretical concept that must be practically proven.

The new methodology was developed by observing processes and inventories (Roser et al., 2015). Blocked processes and overstocked inventories are signs of a downstream bottleneck. Empty stockpiles and starving processes are signs of an upstream bottleneck. By watching various process phases and inventory levels inside a system, it is possible to identify the direction of the bottleneck at any given time and identify the system's current bottleneck. Work sampling approaches, which can be utilized to generate a long-term image of the dynamically shifting bottleneck, can directly see bottleneck shifting. The new methodology is suitable for use by shop floor supervisors and clerks because it does not involve any computations, statistics, or time measurements. Direct observation of the bottleneck also provides additional information about the underlying causes of bottlenecks, making system capacity enhancement and optimization easier. Extensive field testing of the concept garnered excellent feedback from both management and shop floor operators, and it is now in use at Robert Bosch GmbH, where it is referred to as the bottleneck walk.

The simulation program WITNESS and a discrete event simulation approach are used by (Noguera et al., 2015) to examine several production line bottleneck analysis methodologies. An experimental framework has been developed for processing and analyzing the results received from the WITNESS simulation experiments. The benefits and limitations of the stated approaches are examined.

Using a discrete event simulation approach and the simulation program WITNESS, (Králová & Leporis, 2010) compares various techniques for production line bottleneck analysis. By setting up an experimental framework for processing and comparing the findings from the WITNESS simulation experiments, the model's benefits and limitations are examined.

The implementation of a method for detecting bottlenecks in discrete event models created by Toyota motor company is described in (Faget et al., 2005). The goal in this situation is to automate the bottleneck analysis, making simulation easier to understand and adopt for decision makers who are unfamiliar with simulation. The validation of the bottleneck detection approach and its connection with MS Excel spreadsheets are the key findings, and design of experiments is used to give system improvement options.

Process networks with bottlenecks are investigated and shown to be simple multi-source maximal flow linear programming problems (Troutt et al., 2001). According to a review of more than 30 production/operations management and management science/OR textbooks, only iterative trial-and-error procedures are currently recommended for this type of analysis. The maximal flow network approach is simpler for complex problems and offers several advantages not available with trial-and-error approaches. The modeling approach, which describes the use of a fundamental linear programming sensitivity result known as radial change, might offer fresh insights for enhancing system capacity after the implementation of theory of constraints techniques.

2.5 Sequencing

(Bean, 1994) presents a general genetic algorithm that can be used to solve a wide range of sequencing and optimization problems, such as multiple machine scheduling, resource allocation, and the quadratic assignment problem, all of which have a problem controlling feasibility from parent to offspring. This is overcome using a robust representation technique known as random keys, and the computational results are shown for multiple machine scheduling problems, resource allocation problems, and quadratic assignment problems, all of which effectively address a wide range of sequencing and optimization problems.

(Storer et al., 1992) propose a straightforward and flexible framework for creating search spaces and simple local search algorithms for a range of sequencing problems. The inclusion of problem-specific information into the definition of the search area with known heuristics and the heuristic problem solution encoding is the method's key feature. As a result, good solutions are clustered together, making local search more convenient. Simulated annealing and genetic algorithms are two probabilistic search methods that can be applied directly to the search spaces. The method's utility was proved by the development of heuristics for workshop scheduling. The strategy can be easily applied to any scheduling problem with any purpose in this arena. On test issues with the makespan objective, the provided heuristics produced high-quality answers. When compared to the shifting bottleneck approach, these findings are highly promising.

(Baker, 1984) examines the interaction between sequencing priorities and the method of identifying assignment due dates, focusing primarily on average tardiness as a measure of scheduling effectiveness, which highlights several factors that can affect the performance of dispatching rules, including average flow allowance, due date assignment method, and the use of progress milestones. A series of simulation tests give insight on how these variables interact with the dispatching rule, and the results show which combinations are most effective in a scheduling system.

(CAMPBELL HG et al., 1970) describe a simple and direct algorithm for solving very large sequence problems without the use of computers. The algorithm produces approximations to solutions to n job, m machine sequencing problems where no passing is taken into account and the criterion is minimum total elapsed time. The solutions are optimal or nearly optimal and can be produced quickly and easily.

Theorems that establish the relative order in which pairs of jobs are processed in an optimal schedule are proven in (EMMONS H, 1969) which frequently enables the jobs to be completely ordered, thereby solving the problem without the need for searching. The problem is to sequence n jobs on one machine to minimize total tardiness. Sequencing in order of non-decreasing processing times and sequencing in order of non-decreasing due dates is best under more extensive conditions than are currently recognized by corollaries. In general, even large problems can be ordered to the point where only a few schedules need to be searched. These findings are then partly extended to the more general criterion of minimizing a sum of identical, convex, nondecreasing functions of job tardiness, and an efficient algorithm is proposed.

In order to decrease the amount of late jobs, (Moore, 1968) came up with a computationally practical plan for scheduling n jobs through a single facility. The problem of each work having a continuous, monotone, non-diminishing deferral cost function is then addressed using this strategy, with the aim of creating a timetable that incurs the fewest deferral costs possible.

By providing a sequencing approach to determine the order in which models should flow down the line, (Thomopoulos, 1967) proposes a way for converting single-model line balancing procedures to mixed-model schedules. Although the results are not completely optimal, analysis indicates that they are close to it. The technique has already proven to be valuable as an assembly line simulator for examining the consequences of changing line characteristics, and it may be used for both existing lines and as a prediction of the efficiency of future lines.

2.6 Job Shop Sequencing

(Mellor, 1965) discusses the more recent literature on job shop sequencing problems classified as due date or minimum makespan and among the solutions described particular focus is on the increasing adaptation of heuristic devices.

The basic theoretical method does not accurately capture the reality of open job shop scheduling, and its applicability is restricted to those circumstances that are essentially static and behave like the models, as demonstrated by an early field research by (Mckay et al., 1988).

(Cheng et al., 1996) provide a tutorial survey of current efforts on applying evolutionary algorithms to solve classical job shop problems. The representation strategies for the job shop problem that have been presented as well as several hybrid genetic algorithms and traditional heuristic approaches are explored. Other scheduling issues in contemporary flexible production systems as well as other combinatorial optimization issues may benefit from the methodologies developed for limited combinatorial optimization challenges.

(Brucker & Schlie, 1990) showed a well-known graphical method for solving the job shop scheduling problem with two jobs generalizing to job shops with multipurpose machines. To find a schedule which minimizes the makespan in an efficient way they reduced the multipurpose machines job shop problem with two jobs to a shortest path problem. The method also works for arbitrary objective functions which depend monotonically on the finish times of the two jobs.

The well-known 10*10 problem can be solved using an optimization method that combines the heuristic method and the combinatorial branch and bound algorithm (Applegate & Cook, 1991).

Simulated annealing is a generalization of the well-known iterative improvement approach to the combinatorial optimization issue of determining the minimal makespan in a job shop, as described in (Laarhoven et al., 1992). Even though the Markov chains it generates are frequently not irreducible and that it requires accepting cost-increasing transitions with a non-zero probability to avoid becoming stuck in local minima, the technique asymptotically converges in the possibility to a globally minimum solution. The algorithm can produce shorter makespan in computational experiments than two recent approximation algorithms that are more suited to the job shop scheduling challenge.

(Jain & Meeran, n.d.) attempts to analyze a subclass of this problem where the goal is minimizing makespan by giving an overview of the history, the techniques utilized, and the direction of the work is reviewed by evaluating the reported results of the strategies on the available benchmark problems.

(A. S. Manne, 1960) suggests using discrete linear programming to solve the typical job shop scheduling problem, which includes noninterference constraints for individual pieces of equipment as well as sequencing restrictions.

A case study for job scheduling when components are handled on available machines was presented by (Abbas et al., 2016b). Setup and operation times are calculated as the overall processing time for a variety of items using various manufacturing processes through time and motion studies. Different levels of priority are assigned to the tasks based on the due dates, and the tasks are scheduled according to priority. The processing times for certain new workloads are predicted, and an algorithm is suggested and tested to make the best use of the machines available.

(Kassu & Eshetie, 2015) uses a shifting bottleneck algorithm to focus on reducing the makespan of the job shop production system of the Dejena Aviation Industry (DAVI) production system. Five machines were taken into consideration throughout the manufacturing of five jobs, and secondary data was taken from the production logbook.

2.7 Flow Shop Sequencing

Using makespan criterion, (Su & Yi, 2017) provides a two-phase heuristic solution to solve the no wait flow shop scheduling problem. The original and reverse problems were solved using the Nawaz-Enscore-Ham (NEH) method to discover a better schedule as the initial answer. The solution was quickly improved using a swap-based local search. The experimental results of benchmark examples show that the suggested algorithm works well.

According to the multi-machine and multi-product situation, (Rahman et al., 2018) expected mixed integer linear-programming approach for machine scheduling in flow shop environments. A local production system was visited multiple times to collect real data from the industry. The model was then examined using What's Best Excel Solver. The results show that by using the right sequence, it is feasible to complete tasks in the shortest amount of time when compared to other alternative sequence combinations of products. In addition, using the right sequence would reduce idle times for some machines while increasing their utilization.

In case of equivalency between two job orders or partial sequences, (Sauvey & Sauer, 2020) offered two approaches to improve the original Nawaz-Encore-Ham (NEH), based on the two points in the process where decisions must be taken. Two results are equivalent but may produce different results when an equality arises in a sorting technique. The factorial basis decomposition approach, which transforms a number computationally into a permutation, is proposed as the first enhancement to NEH. This approach is highly beneficial for the initial improvement and enables testing of all sequencing options for issues involving up to 50 jobs. Where NEH maintains the best partial sequence is where the second improvement is found.

In order to reduce the maximum job completion time, (Belabid et al., 2020b) investigate the solution of a permutation flow shop problem with sequence-independent setup time. A Mixed-Integer Linear Programming (MILP) model, Johnson's rule and NEH-based heuristics, and metaheuristics based on the iterative local search method and other metaheuristics of the iterated greedy algorithm are all included in this contribution. To verify the efficiency of solution strategies, a collection of test problems is numerically simulated. It has been discovered that the Johnson-based heuristic performs worse than the adapted NEH heuristic for problems of a reasonable size.

The Nawaz-Enscore-Ham (NEH) heuristic is used by (W. Liu et al., 2017) to create a new priority rule that is then utilized to resolve scheduling issues in permutation flow shops. By reducing partial system idle time without increasing computing complexity, a unique tie-breaking rule is also created, outperforming the best NEH-based heuristics previously mentioned in the literature.

2.8 Missing Areas in Existing Research:

The method (Brucker & Schlie, 1990) deals with is an inefficient algorithm for problems with three or more jobs. The research does not concentrate on heuristics or branch and bound methods. (Su & Yi, 2017) employing the same heuristic technique in real-world scenarios can be used to solve no wait flow shop sequencing problems with various goal functions, and swap-based local search and the reversibility property can be used to create a metaheuristic solution. (Sauvey & Sauer, 2020) takes long computation time for the improvement method. The problem can be considered for big problems to reduce the number of tests and time taken to identify the best solutions. Global algorithms can be applied for more effective results.

(Rahman et al., 2018) considered a single machine per stage of the production. Multi stages of parallel machines of multi products production can be considered. Other strategies can be used to improvise the makespan of the production system with more advanced methods. (Belabid et al., 2020b) can consider no idle machines or unavailability of machines as other constraints.

(A. S. Manne, 1960) appears to include significantly less variables than other ideas and makes no attempt to determine the computing feasibility of the strategy in the case of large-scale realistic problems. Similarly, (K. Liu et al., 2008) also does not try to apply the throughput maximization strategy to real world applications.

Analyzing the previous literature, one exciting concept, which has not yet been researched, involves the non-permutational circular flow shop with different category of multiple products following specific work orders on multiple machines. The special structure provides formulations using different algorithms that has the potential to open a new horizon for looking at a sequencing problem. Comparing the result with a simulated environment and for more realistic approach the case study can be performed for precision analysis.

2.9 Problem Definition

Consider a typical circular flow shop with m machines and n jobs. There is no requirement that all jobs be processed through all machines; instead, each job will only be processed through one machine once. Every job will be processed in a specific order. Each job's processing time is predetermined and known. The machine performs the jobs in the order they are received (Abbas et al., 2016a).

The following criteria must be determined before sequencing jobs:

- The facilities on the shop floor.
- Number of jobs in the system.
- The amount of time needed to set up and run each job on each machine.

Following parameter identification, the jobs are sequenced based on the following assumptions.

1. At the start of the planning period, all n jobs are available at once (a case of static problem).
2. No more than one machine can work on the same job at once.
3. Each job's processing time on each machine is known and deterministic.
4. Setup and transportation times are not adjusted during the computation and are not dependent on the sequence.
5. At the start of the planning period, all m machines are available and prepared to start working on any of the n projects.
6. No machine is capable of handling more than one job at once.
7. Each type of machine in the shop has just one unit.
8. Pre-emption is not permitted, i.e., once a job has started its processing on any machine, it must be carried through to completion on that machine.
9. Job has a specific order of operation to follow.
10. Jobs may travel in a circular path to complete all processes.
11. Job may skip one or more process temporarily to maintain the order
12. At a time, the maximum number of jobs in circulation must be equal to the number of pallets.
13. There are various types of machines that can be used. That is, no two machines can perform similar operations.
14. Buffer capacity between machines is infinite.
15. Anticipatory sequencing is responsible for the issue. To put it another way, machine setup for processing a group can begin before any jobs associated with the group physically arrive at that stage.
16. Every job is equally important (weight).
17. Jobs may wait in between two machines.
18. The feeds are assumed to be available from time zero onwards.
19. All model parameters are deterministic.
20. Machines are available without breakdown.

Machines i=1, 2...., m processed job j, j=1, 2......, n, in with a nonnegative processing time p. (i, j). Job j's completion time on machine i's is indicated by c (i, j). The objective is to identify a job sequence that reduces the makespan, or Min $\{max_{1 \le j \le n} C(m, j)\}$

When the machine is processing, each operation may have to wait. The goal is to reduce the makespan, or the amount of time it takes from the start of the first job's execution on the first machine to the end of the final job's execution on the last machine.

The main challenge is to create a sequence that meets the requirements listed below.

- All tasks are completed
- All constraints are satisfied, and
- All selected criteria are optimized.

The necessary notations are presented in the following to introduce the model.

i	Machine index, $i \in \{1, \ldots\ldots, m\}$
j	Job index, $j \in \{1, \ldots., n\}$
k	Position index, k $n \in \{1\ldots., n\}$
π	$\pi = \{\pi_1, \pi_2 \ldots. \pi_n\}$ Is a feasible sequence
P_{ij}	Processing time of job j on machine i
P_{ik}	Processing time of job π_k on machine i

$$x_{jk} = \begin{cases} 1, \text{If job j occupies position k} \\ \quad 0 \; Otherwise \end{cases}$$

C_{ik}	Completion time of job π_k on machine i
C_{max}	Makespan
I_{ij}	Idle time on machine j from the end of job (j - 1) to the start of job i.

The important Terminologies are described here:

- Number of Machines: The term "number of machines" refers to the total number of service locations that a job must pass through to be considered finished.
- Processing Time: This is how long a job takes to complete on each machine.
- Processing Order: This term describes the order (or sequence) in which machines are needed to complete a job.
- Idle Time on a Machine: This is the period during which a machine is not processing a job, i.e., the period between the conclusion of job (i-1) and the beginning of job i.
- No Passing Rule: This rule specifies that jobs must continue to be processed on specific machines in the same order. For instance, each work should go to machine M1 first, followed by M2, if n jobs are to be processed on machines M1 and M2 in the order M1 M2.(Su & Yi, 2017)
- Makespan: This term is used to describe the amount of time that passes between the beginning of the first job in the sequence on the first machine and the end of the final job in the sequence on the last machine.

CHAPTER 3

RESEARCH OBJECTIVE

Manufacturing systems design is shifting away from the job shop perspective and toward a flow shop or hybrid job shop orientation because of the current expanding interest in Group Technology (GT), Computer-Integrated Manufacturing (CIM), Just-In-Time manufacturing (JIT) and other concepts. Consequently, flow shop-oriented research is expected to become more significant in the future. (Stafford, 1988). As stated earlier in the previous section, productivity of the flow shop is affected directly by the sequencing decisions made. The objectives of this investigation are to examine the n-job, m-machine static sequencing problem with the objective of minimizing makespan, to review the literature, and to compare if an effective, approximate algorithm can be applied to get the optimal output. There are three main areas of concentration in this investigation.

1. Investigation of the previous approaches to understand the underlying assumptions and to study the effectiveness of those assumptions in solving the problem.
2. Analysis of some effective algorithm which is both efficient and accurate in giving optimal/near optimal solutions for the static case of the problem.
3. Evaluation of the algorithms by comparing its performance.

The next section describes the methodology adopted to cover each one of the three main areas of investigation.

3.1 Research Question:

How different techniques can apply in the SM lab system and simulated environment to compare the ability to mitigate bottlenecks?

For many years the optimal sequencing has been a problem of interest for researchers in the field of operation research. In most manufacturing and production systems, Sequencing is considered as an important tool which decides the order in which jobs pass through the machines or workstations. There are different sequencing techniques to solve the real-life job shop and flow shop problem. A common approach problem where it is impossible to find the optimal solution in feasible time is to solve them using heuristics. Computer simulation is an effective tool to analyze the validation of the approximate heuristics solution. The SM lab system is selected as a case study and the basic application of Arena software is used to analyze the production line bottleneck. A hybrid model of job shop and flow shop presented in a complex circular system is proposed to solve using different techniques and comparing the impact on real life and simulated environment. Different job categories are selected to make the scenario complicated, and the existing complex machine setup added the new era in sequencing. The methodology used to solve the problem is technically used to get the approximate optimal sequence and analysis of makespan and other times are adding values for comparison.

3.2 General Theoretical Framework

The purpose of this research is to find the near optimal solution for sequencing using heuristics algorithms. The focus is to compare the makespan of a sequence in a simulated environment and lab system. To pursue this mission and achieve the target, a general theoretical framework has been scrupulously and accurately investigated and developed.

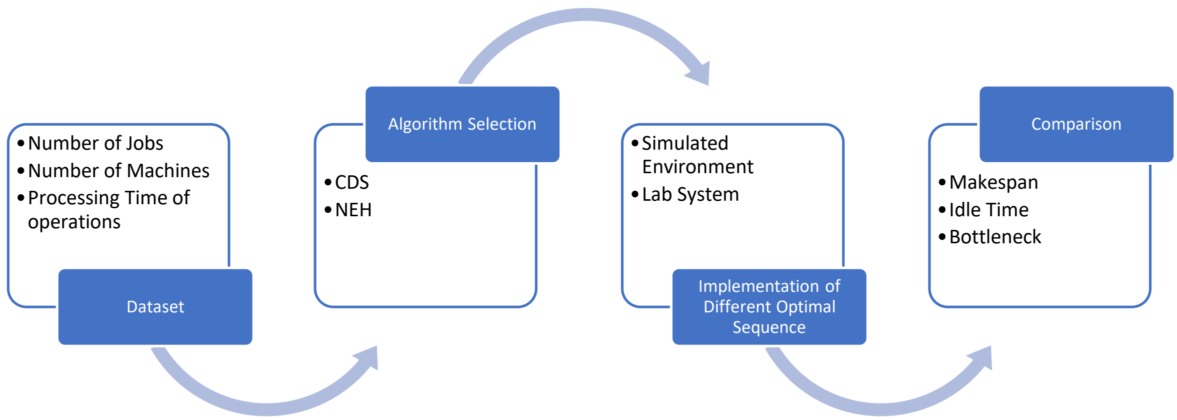


Figure 3.2.1: A block diagram to summarize the proposed framework

The framework aims to compare different heuristics solution methodology in real life as well as simulated environments. As a first step, it is necessary to understand different algorithms, the lab system, and the available dataset. The obtained data are not properly arranged to feed them in different algorithms; therefore, a virtual job concept is created, and some intermediate steps need to be performed to adjust some other real life environment constraints. A simulated environment is created using Arena software to replicate the lab environment. In comparison, the makespan and other time may vary as transportation time and other non-value-added time is not considered in this model.

The framework proposed provides an effective smart manufacturing solution that will allow sequence in a complicated circular system with different types of job category and variety of machines with specific work orders.

CHAPTER 4

RESEARCH METHODOLOGY

Scientific methodology of analysis, synthesis and evaluation is used during this investigation. During the analysis stage of this investigation, existing heuristic solutions for the flow shop problems are studied thoroughly. Although the heuristics chosen for analysis are wide-ranging in their approaches to the problem, they are representative, rather than an exhaustive collection of the sequencing heuristics. The analysis performed in this study revealed patterns and characteristics of some efficient heuristics. Some cases where these heuristics tend to perform less accurately are identified (Aalla, n.d.). Based on these observations, a set of rules for effective sequencing of n jobs on m-machine static and deterministic flow shops is developed. In the evaluation stage of this investigation, these heuristics are tested for accurate performance in terms of makespan, efficiency in terms of idle time and bottleneck and effectiveness in terms of accuracy vs computational effort. Comparisons are made with some well-known heuristics: Primarily CDS heuristic, and NEH heuristic. Various graphical, empirical, and statistical methods are planned to use for comparison. Finally, based on the results, some appropriate suggestions and conclusions will be mentioned. The next section provides the necessary data collection and validation for an investigation of this methodology. It will help to discuss the results compared with real life scenarios and simulated environments as well.

4.1 Method of Analysis

Appropriate technologies have been chosen to allow for a partial automation of tasks. The core idea is to leverage on open-source tools and software to support broad adoption and lower the entry barriers of sequencing problems. The tools and software that have been selected for each step of the framework are presented here. (*Torlino, H*, 2020)

4.1.1 Data Source and Dataset

<u>Data Source</u>

The collection of a dataset that triggers the framework has been performed through the WVU Smart Manufacturing Lab. It is endowed with a Festo Didactic Learning System (Figure 4.1.1), that is composed of eight modules that correspond to eight different steps provided by different sensors and RFIDs that simulate eight different manufacturing processes. The modules of the Cyber-Physical Lab are industry-oriented, and the hardware consists of industrial components for didactic training.

Figure 4.1.1: WVU's Festo Didactic Learning System.(Torlino, H, 2020)

The eight modules are:

1. CM-AM-MAG-FRONT (Magazine): For feeding parts. Differentiated in magazine front cover and back cover. This releases front covers.
2. CP-AM-MEAS (Measuring): For quality assurance. Processing of analog input signals.
3. CP-AM-iDRILL (iDrilling): For drilling parts. With a controller with a web interface for CPS.
4. CP-AM-MAG.BACK (Magazine): For feeding parts. Differentiated in magazine front cover and back cover. This releases back covers.
5. CP-AM-MPRESS (Muscle press): For pressing parts.
6. CP-AM-HEAT (Heat tunnel): For heating workpieces with thermal processing.
7. CP-AM-TURN (Turning): For turning workpieces.
8. CP-AM-OUT (Output): For removing workpieces from output.

In addition, eight resources to move parts and workpieces are present. They are pallet carriers for transporting the pallet. Pallets are put on the carriers, for receiving one workpiece. The workpiece location can be identified through RFID present on the pallets carriers, they automatically communicate the position to the system and the process execution is triggered.

Figure 4.1.2: A pallet carrier with pallet.(Torlino, H, 2020)

The modules are connected through a conveyor system. It is made of 8 conveyor belts, assembled in a rectangle shape connected with passive corners. The carrier can be moved by using the balls that are fitted in the corners. Sensors activate and deactivate the conveyor belts when the carrier passes through the workstations. Every workstation is equipped with a Programmable Logic Controller (PLC) to individually control and visualize the process, and to set parameters. The CP Lab conveyor is an 80 mm wide and 700 mm long pallet transfer system. At the start and the end of the CP Lab conveyor there are capacitive sensors which recognize the pallet on it.

Each of the eight modules is equipped with touch panels provided with an emergency stop mushroom actuator. The operation mode is automatic, and all the processes are executed and monitored by an MES software. The data source is an MES4 software system present in the Festo CP Lab. It is a revolutionary manufacturing execution system that has been exclusively created for industry 4.0 learning platforms. It can

- Reading orders and status updates.
- Arrange the order lines.
- Write the order's distribution of the goods carriers.
- Provide a material master that shows a graphic of the workpiece.
- Prepare machines, considering costs and energy usage.
- Establish a material and data buffer for the warehouse.
- Develop and control customer data.
- Use icons to describe system layouts.
- Automated routing based on machine capabilities and routing cards.
- The creation of reports on OEE, PLC, and malfunctions using graphics.

Considering the modules provided in the Festo Didactic Learning System, a brief description of the functioning of each module is here presented:

Front Part of Magazine:
This application module is designed to place a front cover on the carrier. The carrier is recognized and stopped. The condition of the carrier is checked by some sensors. In case of front cover magazine, if there is no workpiece on the carrier, then the front cover is released. The operation performed by this module has an average processing time of 2 seconds. The front cover piece is defined by the part number 110.

Figure 4.1.3: Magazine module: Front cover.(Torlino, H, 2020)

<u>OUTPUT:</u>

This application module is designed for using an electro-pneumatic, two-axis handling device that dispenses good and bad parts into two different ramps. The carrier is detected when fed into the module and then stopped. The application module removes the good part from the carrier and places it on one of the two sides, while it removes the bad part placing it on the other one. The definition of which ramp to use can be freely defined in MES. The operation performed by this module has an average processing time of 7 seconds. The left ramp is chosen for a good product, while the right ramp is destined for scraps.

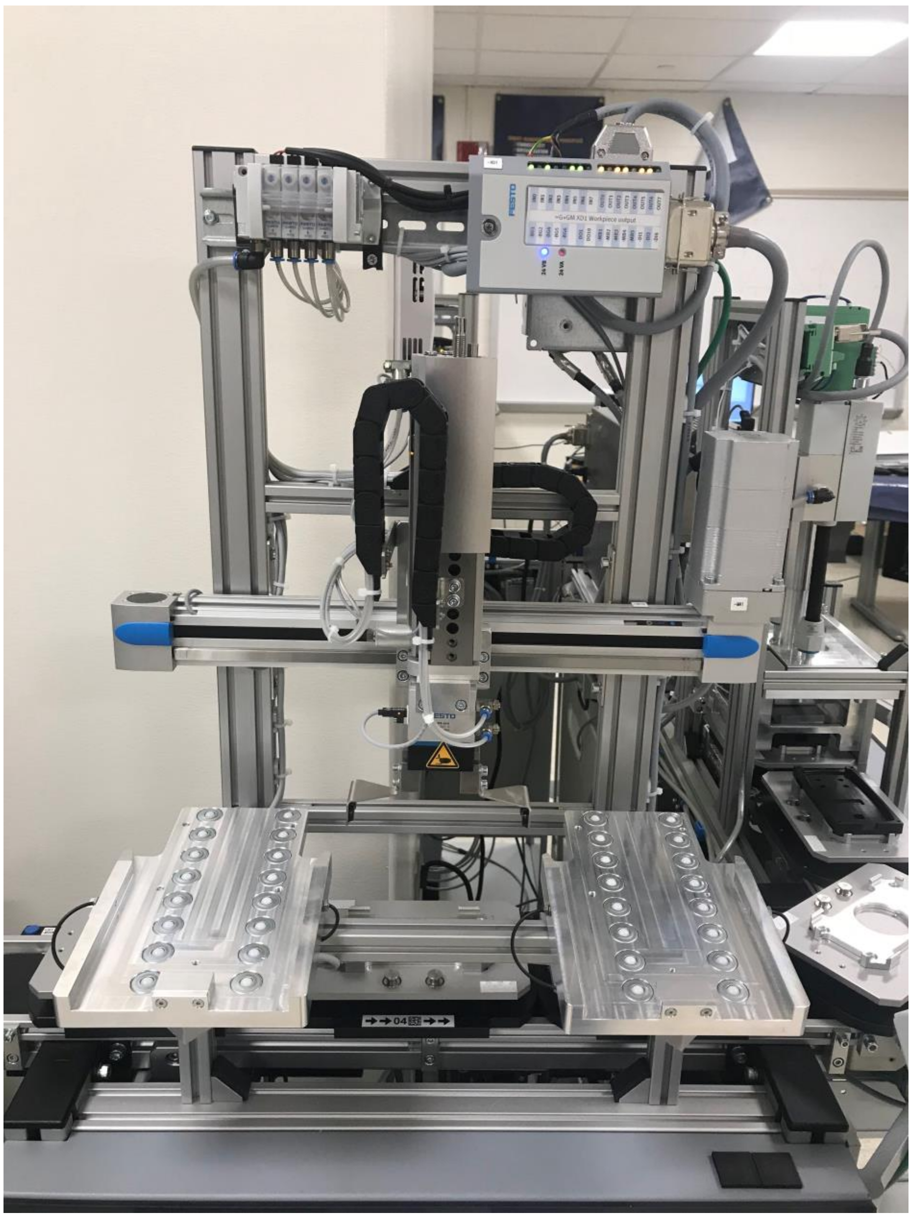


Figure 4.1.4: Output module.(Torlino, H, 2020)

MUSCLE PRESS:

This application module is designed for pressing the front cover and the back cover together. When moving into the module, the workpiece is recognized, and the carrier is stopped. The covers on the carrier are pressed together with the help of the muscle with a defined pressure (N). Then the carrier leaves the module. The operation performed by this module has an average processing time of 25 seconds. The pressing pressure is 50N for a time of 5 seconds.

Figure 4.1.5: Pressing module.(Torlino, H, 2020)

<u>HEATING TUNNEL:</u>

This application module is designed to heat workpieces to prepare them for thermal processing. The workpiece is recognized during the infeed into the module and the carrier is stopped. The temperature of the workpiece is measured, then the piece is heated until the desired temperature is reached. Then, after a waiting time, the carrier is released from the module. The operation performed by this module has an average processing time of 55 seconds. The heating temperature is set at 25 °C.

Figure 4.1.6: Heating module.(Torlino, H, 2020)

<u>TURNING:</u>

This application module is designed for turning the workpiece 180°. The carrier is recognized and stopped. The workpiece is gripped from the turn unit and turned for 180°. Then the workpiece is placed back on the carrier, which is then released. The operation performed by this module has an average processing time of 3 seconds.

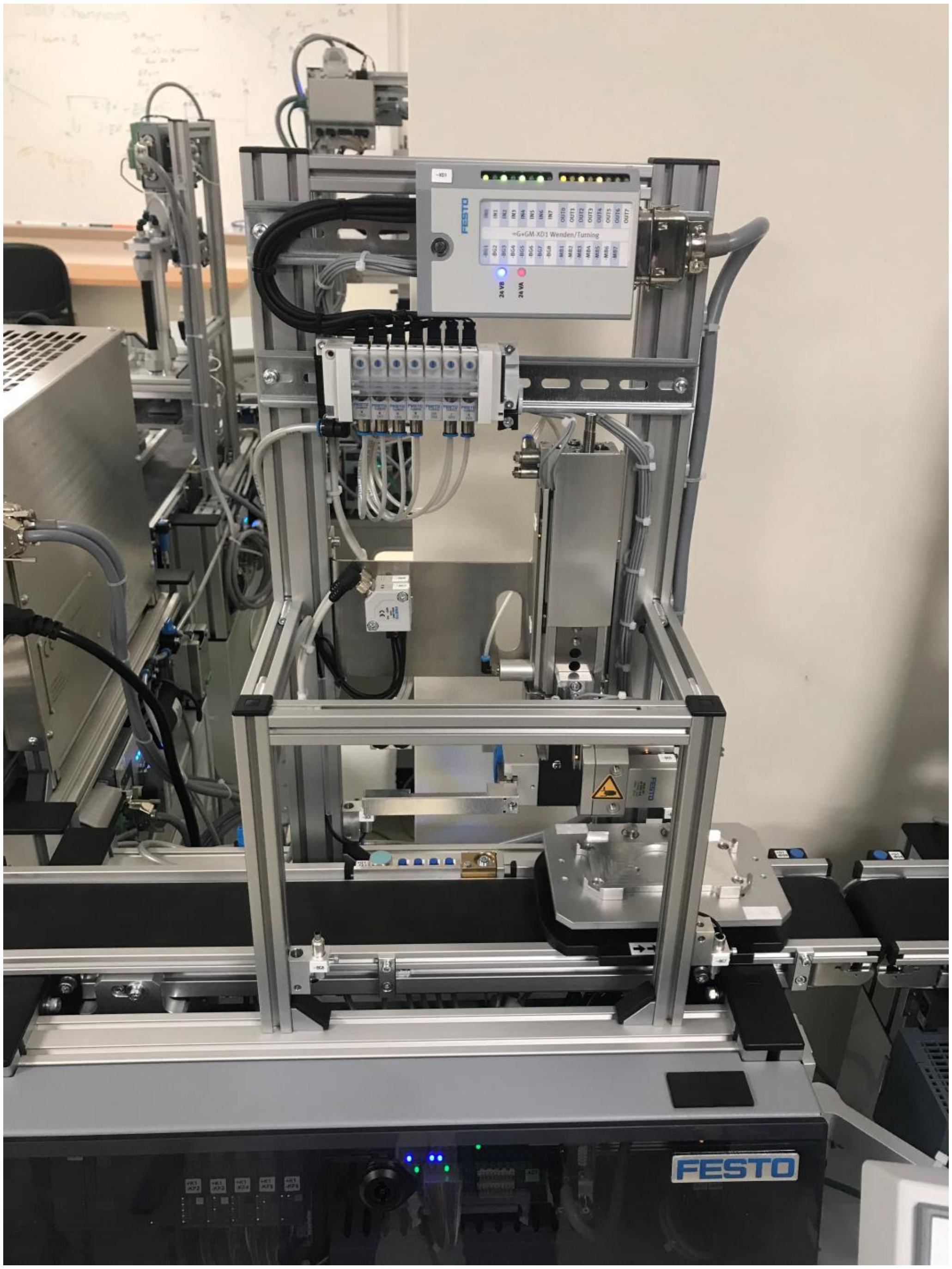


Figure 4.1.7: Turning module.(Torlino, H, 2020)

__Back Part of Magazine__

This application module is designed to place a back cover on the carrier. The carrier is recognized and stopped. The condition of the carrier is checked by some sensors. In case of front back cover magazine, if there is the front cover on the carriers, then the back cover is released. The operation performed by this module has an average processing time of 2 seconds. The back cover pieces are defined by the part number 111.

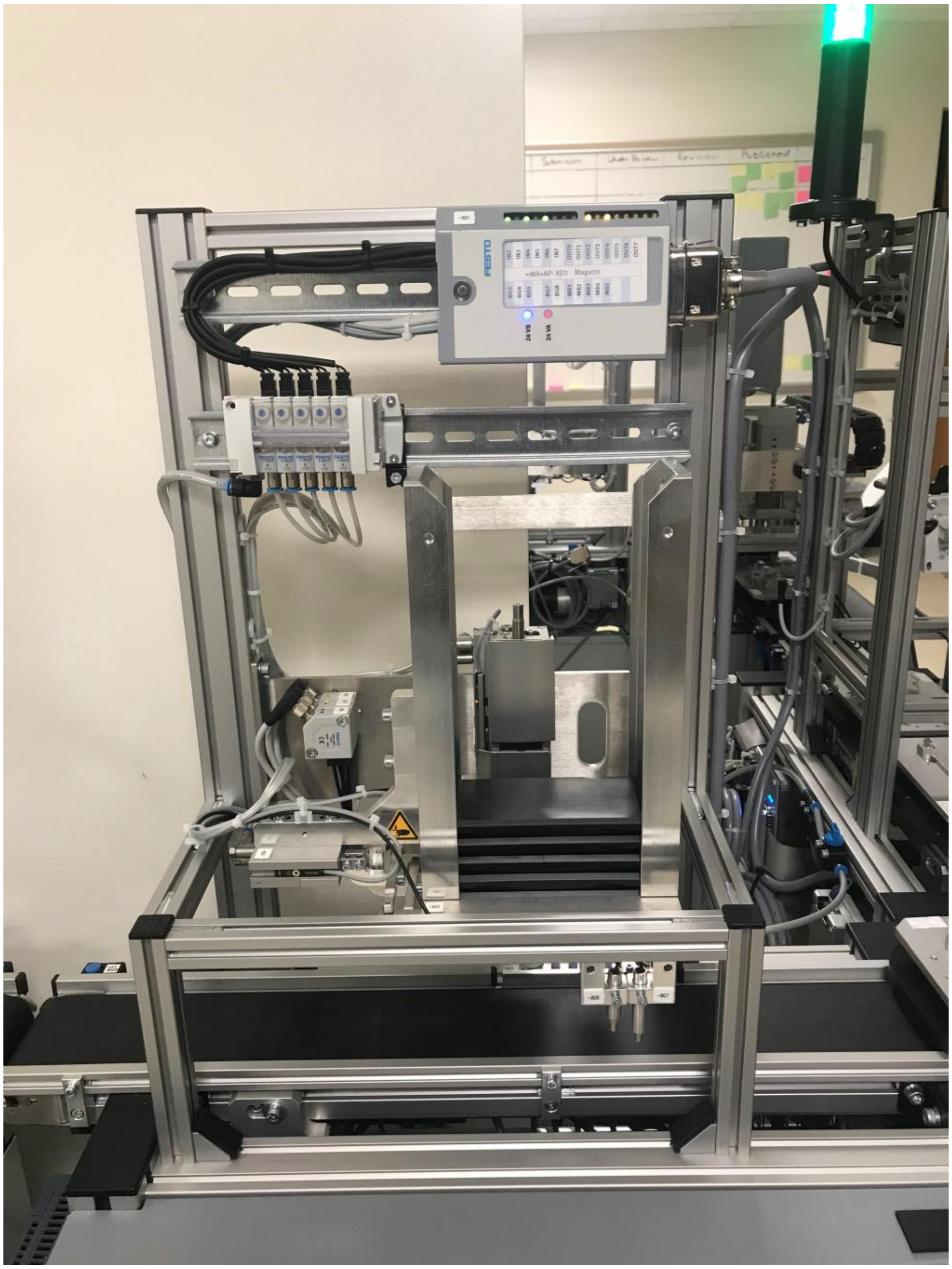


Figure 4.1.8: Magazine module: Back cover.(Torlino, H, 2020)

<u>MEASURING:</u>

This application module is designed for quality control, it checks the height of workpieces and announces bad parts. This module has two analog sensors that measure the height difference of two detected points. The workpiece is recognized, and the carrier is stopped, then the quality control takes place: if the difference is within a defined range the workpiece is good, otherwise is considered a scrap. Then the carrier leaves the module. The operation performed has an average processing time of 2 seconds. Upper Limit and lower limit and default difference can be set, and according to this setting there might be scarps during the production.

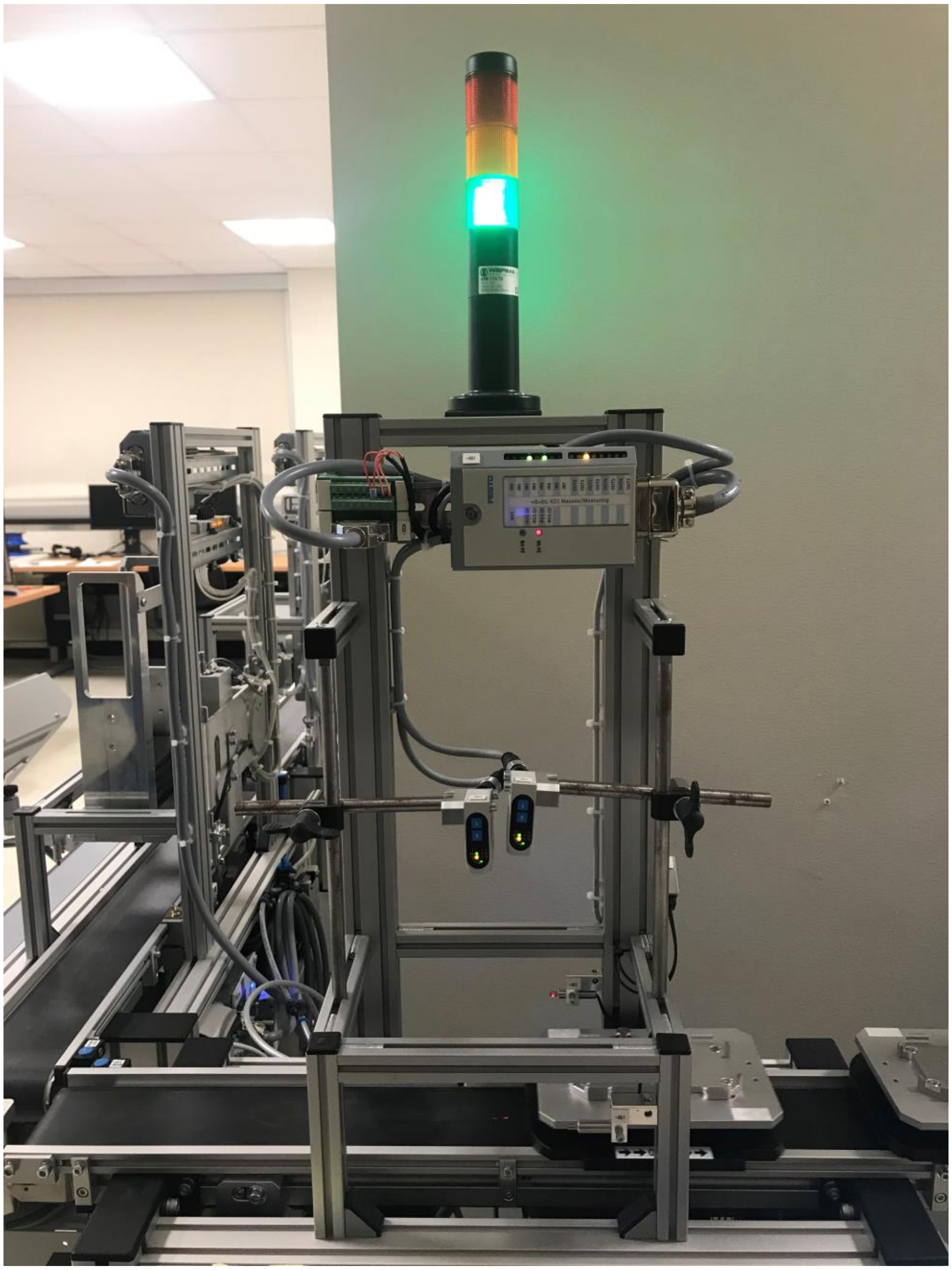


Figure 4.1.9: Measuring module.(Torlino, H, 2020)

DRILLING:

This application module is designed to drill 4 holes in the front cover. The workpiece is recognized when moving into this application module and the carrier is stopped. Then the workpiece is checked and the machine questions itself if the workpiece is correctly formed by only a front cover without back cover. Then the machine makes 2 drilled holes in the left part, and it moves the workpiece to make 2 additional holes on the right. After that the carrier leaves this application module. The operation performed by the module has an average processing time of 10 seconds.

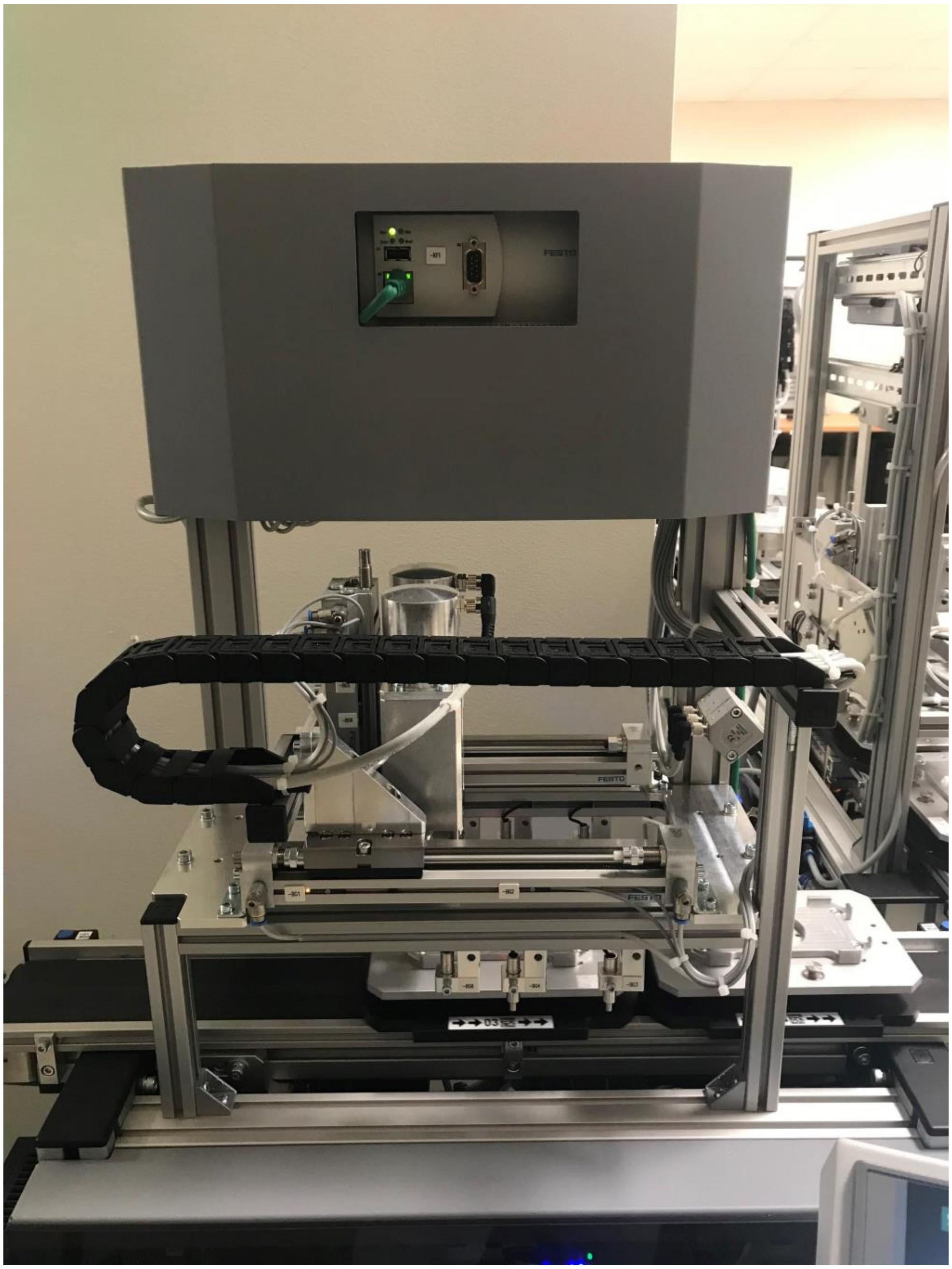


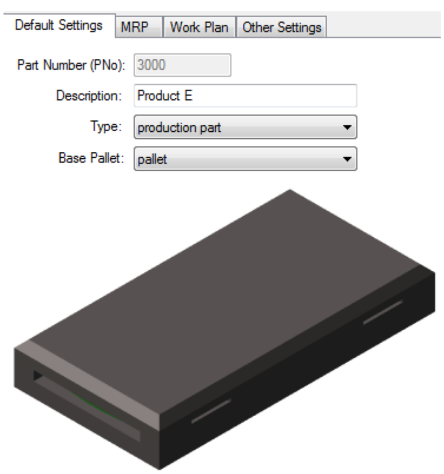
Figure 4.1.10: Drilling module.(Torlino, H, 2020)

Dataset

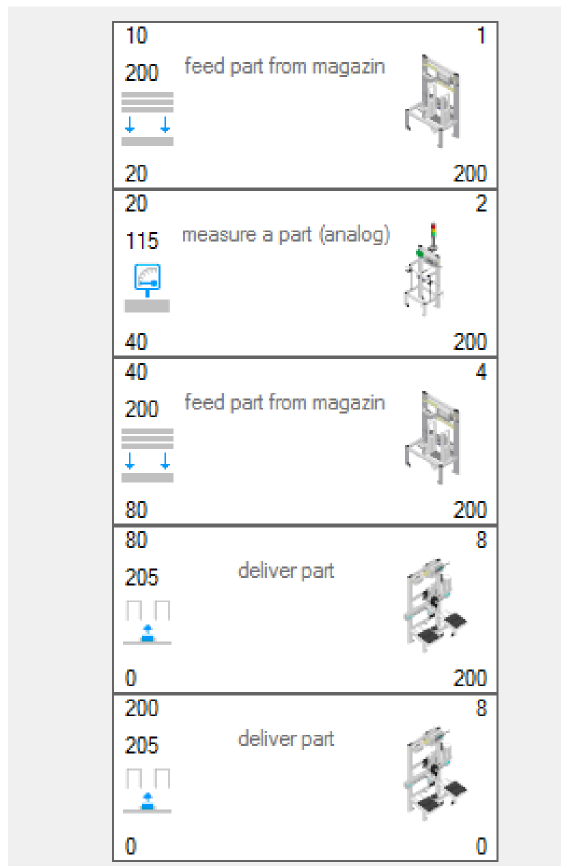
To retrieve a satisfactory sample of data it has been decided to create an optimized production sequence, launch it on the Festo System and monitor the total throughput time. To make this step as much as possible like a real manufacturing situation, three products have been defined that are characterized by three different working plans (i.e., the sequence of the modules needed to produce the specific product).

- Product E: Defined by the code 3000 (Figure 4.1.11a). Its working plan in n. 6, defined as easy in terms of complexity (Figure 4.1.11b). The path followed is:

✓ Module 1 feed the front cover from magazine;
✓ Module 2 measuring the part.
✓ Module 4 feed the back cover from magazine;
✓ Module 8 delivers the workpiece.



Product E.



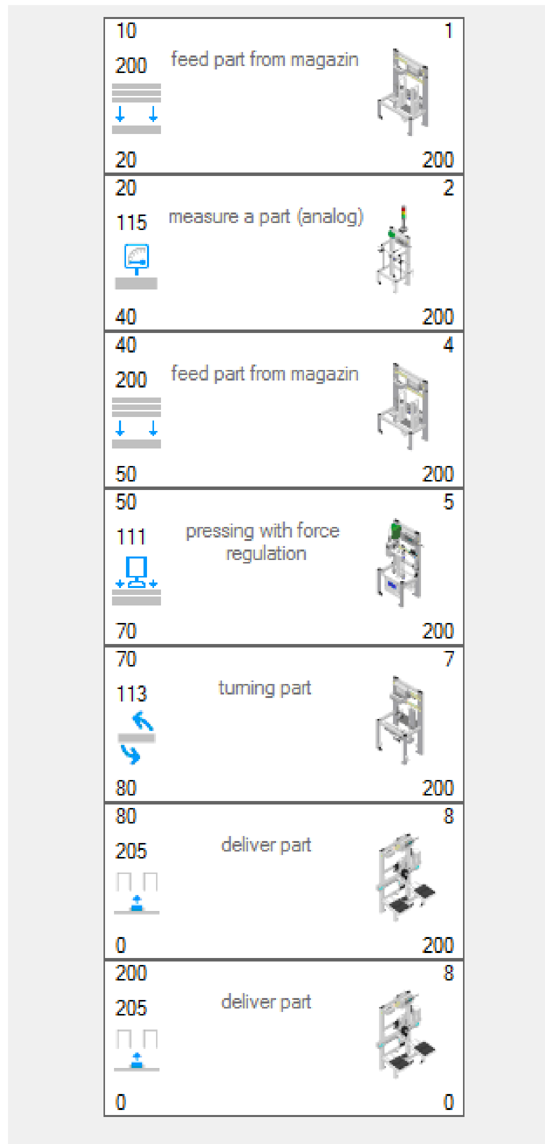
Working plan 6.

Figure 4.1.11: Product E and its relative working plan.(Torlino, H, 2020)

- Product M: Defined by the code 3001 (Figure 4.1.12a). Its working plan is n.7, defined as medium in terms of complexity (Figure 4.1.12b). The path followed is:

✓ Module 1 feed the front cover from magazine;
✓ Module 2 measuring the part.
✓ Module 4 feed the back cover from magazine;
✓ Module 5 pressing the two covers together.
✓ Module 7 turn the workpiece.
✓ Module 8 delivers the workpiece.

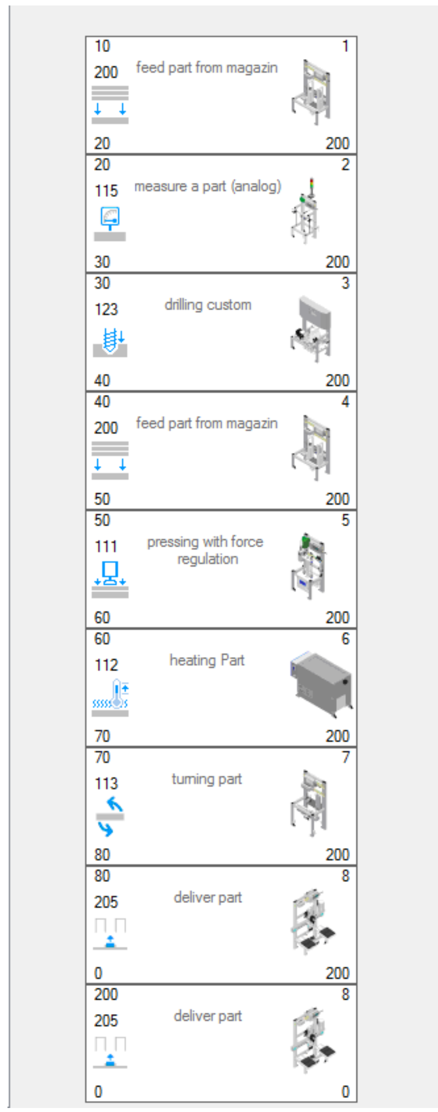
Product M.



Working plan 7.

Figure 4.1.12: Product M and its relative working plan.(Torlino, H, 2020)

- Product C: Defined by the code 3002 (Figure 4.1.13a). Its working plan is n.8, defined as complex (Figure 4.1.13b). The path followed is:
  - ✓ Module 1 feed the front cover from magazine;
  - ✓ Module 2 measuring the part.
  - ✓ Module 3 drilling the holes in the workpiece.
  - ✓ Module 4 feed the back cover from magazine;
  - ✓ Module 5 pressing the two covers together.
  - ✓ Module 6 heating the workpiece.
  - ✓ Module 7 turn the workpiece.
  - ✓ Module 8 delivers the workpiece.



Product C. Working plan 8.

Figure 4.1.13: Product C and its relative working plan.(Torlino, H, 2020)

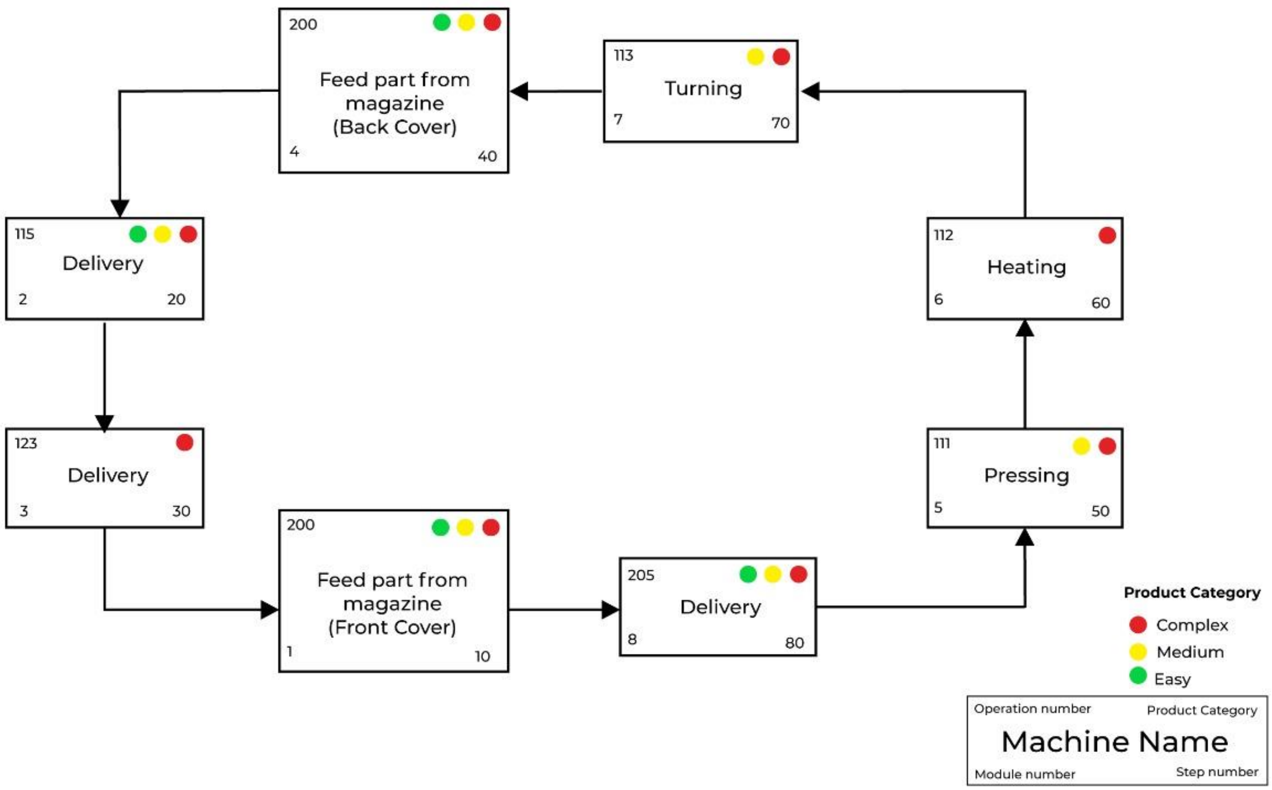


Figure 4.1.14: Schema of Festo Didactic Learning System with relative product paths.

The different machines cannot be bypassed. If a machine is busy, pallets must wait in queue, which can lead to the creation of bottlenecks. Each machine can work only one product at a time, therefore if it is busy, it cannot work another piece. Each piece can be processed at each machine only once according to the established paths. The interruption of the workpiece being processed is not allowed; each operation once started must be completed. As there are only eight pallets, up to eight products can be worked simultaneously on the line. If no more pallets are available, it is not possible to start a new production; a pallet is again available when its product reaches the eighth module.

It has been considered a production plan of different pieces in the three product types: type E, type M, and type C. Some assumptions have been made before retrieving this dataset. It has been considered that orders are coming from the same customer. Job orders are known and ready for processing. It is assumed that all jobs are available at the start of production and may be performed in any order. Moreover, each job order has a size equal to one, which means that each job order corresponds to only one production piece. Finally, no scraps have been considered, as it is not relevant for the type of analysis performed. To do so, on the MES interface, in the setting of each working plan for the measuring production step, where the quality control happens, Upper Limit and lower limit have been set in a way that no scraps could be produced, they are respectively 10mm and 0mm, with a default difference of 2mm. In addition, at the starting time of the production all the pallets are needed to be immediately available, therefore they have been put in queue right before module 1.

The initial situation of the analysis is supposed to be completely optimized using a different algorithm, to define a specific-case scenario.

4.2 Methodology of Research:

The availability of machines and the appropriate operations to be carried out on each machine are used to examine the manufacturing facility. The manufacturing facility is segmented into front machine, output machine, muscle press, heating tunnel, turning machine, back machine, measuring machine, and drilling machine centers based on similarity. The facility is analyzed before the jobs to be processed are chosen. Using a stopwatch, the operating time for each task on each machine center was measured. (Abbas et al., 2016a)

After collecting the dataset from the data source, the methodology is implemented to find the optimal sequence using different algorithms. The research problem is complicated because it is a hybrid problem of circular flow shop problem additionally constrained with job shop problem. To solve the constraint, virtual jobs are considered.

Virtual jobs

According to the model, the operational order for processing is not the same as the machine setup order. Different category jobs have different precedence to follow. To adjust this constraint, an actual cycle of process is split in some sub-cycles. The tables below will help to easy visualization of this topic:

Table 4.2.1: Operational Order of Process

Product Category	Front Machine	Measuring Machine	Drilling Machine	Back Machine	Muscle Press	Heating Tunnel	Turning Machine	Output
Easy	2	2	0	2	0	0	0	7
Medium	2	2	0	2	25	0	3	7
Complex	2	2	10	2	25	55	3	7

Table 4.2 2: Precedence for Each Job

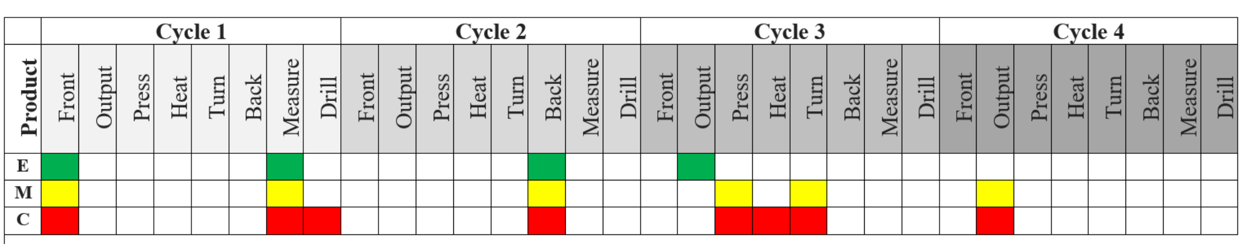


Table 4.2 3: Creating Virtual Jobs by maintaining Precedence according to their Processing Times

Product Category	Virtual Job	Front Machine	Output	Muscle Press	Heating Tunnel	Turning Machine	Back Machine	Measuring Machine	Drilling Machine
Easy	Easy 1.1	2	0	0	0	0	0	2	0
	Easy 1.2	0	0	0	0	0	2	0	0
	Easy 1.3	0	7	0	0	0	0	0	0
Medium	Medium 1.1	2	0	0	0	0	0	2	0
	Medium 1.2	0	0	0	0	0	2	0	0
	Medium 1.3	0	0	25	0	3	0	0	0
	Medium 1.4	0	7	0	0	0	0	0	0
Complex	Complex 1.1	2	0	0	0	0	0	2	10
	Complex 1.2	0	0	0	0	0	2	0	0
	Complex 1.3	0	0	25	55	3	0	0	0
	Complex 1.4	0	7	0	0	0	0	0	0

Creating Virtual jobs and maintaining precedence, the processing time is assigned. This dataset is used as the input of different algorithms. Initially some algorithms are selected to analyze, which will fit in this model. After some experimentation, heuristics works better for its approximate optimal solution. Next the optimal sequence is reorganized to keep pace with the precedence and pallet constraints are adjusted to replicate the real model. The final output of that system is considered as the final optimal sequence and then the makespan is calculated. This output is also useful for calculating the idle time and bottleneck of the system.

Algorithm for Methodology

```
z ← Input list of jobs of all types
SET virtual_job equal to EMPTY LIST
for each job in z
    for i in number off loop need for job
        create a virtual job
        add virtual job to virtual_job
    Endfor
Endfor
SET job_matrix equal to EMPTY LIST
for each job in virtual_job
    add processing time job to job_matrix
Endfor
Fit job_matrix to an algorithm
set algorithm_fit to false
Do
    if job_matrix Fit the algorithm
        Set job_seq to optimal sequence from algorithm
        set algorithm_fit to true
    Else
        Select another algorithm
While
    algorithm_fit is false
SET actual_seq equal to EMPTY LIST
for each job in job_seq
    j ← actual job name of job
    if j is not in actual_seq
        add j to actual_seq
Endfor
cart_no ← number of available cart
Set cart_list to EMPTY LIST
for number starting from 1 to cart_no
    add EMPTY LIST to cart_list
for job in actual_seq
    SET vL to EMPTY LIST
    vL ← list of all virtual jobs of job
    add vL to first available cart in cart_list
Endfor
SET final_virtual_list to EMPTY LIST
for i = 0 to Len of cart_list[0]
    for j = 0 to Len of cart_list
        add cart_list[j][i] to final_virtual_list
    Endfor
Endfor
SET final_job_matrix to EMPTY LIST
for each job in final_virtual_list
    add processing time of job to final_job_matrix
Endfor
calculate makespan of final_job_matrix
calculate idle time of final_job_matrix
calculate bottleneck of final_job_matrix
output makespan,idle time, bottleneck, job_seq
```

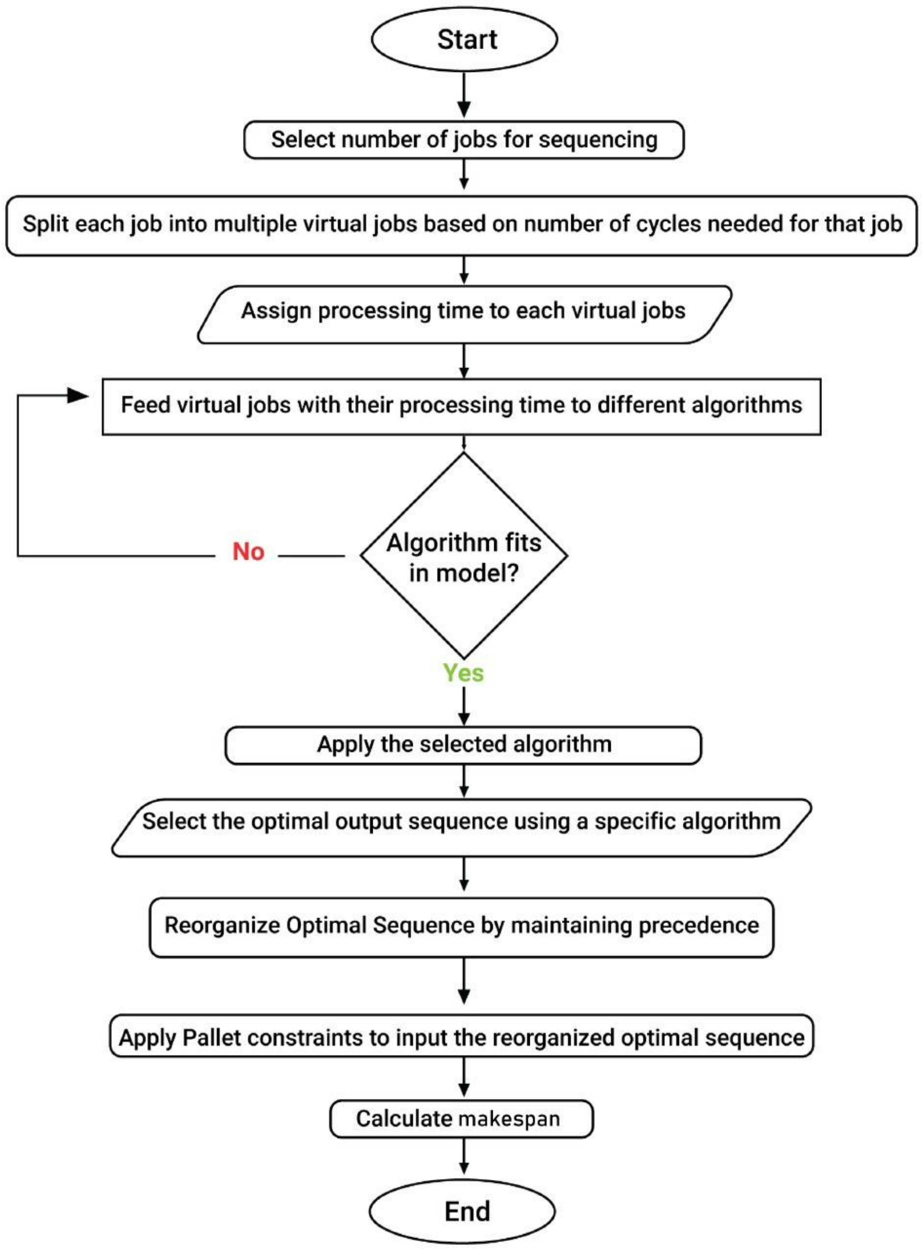


Figure 4.2.1: Flowchart of Methodology

4.3 Selection of Algorithms:

Both accurate and approximate methods of solving the problem of determining an optimal or nearly optimal sequence of jobs being sequenced in a flow shop environment have been considered. Exact procedures, which in fact require an electronic computer, have been created to reduce a problem comprising a small number of jobs and a set of well-defined criteria.

(CAMPBELL HG et al., 1970) must be concerned not only with finding the best solution but also with using the solution technique in a useful and cost-effective way. At this point in time, companies with sequencing problems involving large numbers of jobs and machines must use approximate methods while awaiting further development of exact techniques or faster or more economical computers. The fact that the procedural steps can be kept simple enough to prevent the problem solver from losing sight of the overall view of the problem is another incentive to look into approximate methods (Byung Park, 1984).

For a setup where the number of jobs and machines are not very small there is usually a tradeoff between the quality of the sequence and the computational effort involved in arriving at the sequence. When computing an optimal solution is impractical, one must accept choosing a good solution. These problem-solving techniques are known as heuristics. Heuristics frequently have an intuitive justification, but they are not always guaranteed to produce an optimum or even a good answer.

Considering all the scenario, algorithms can classify as follows:

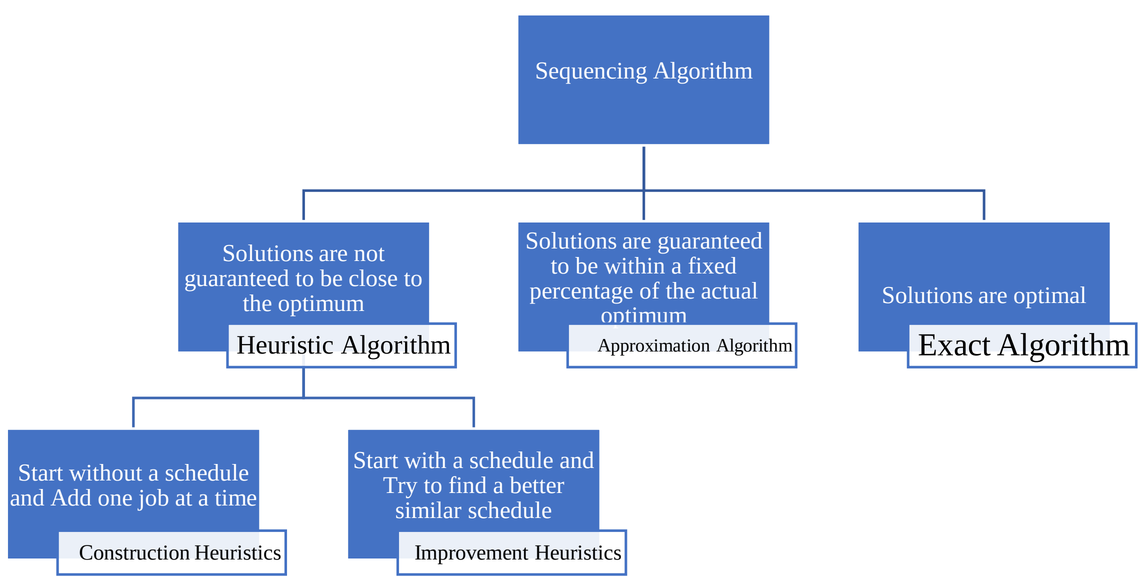


Figure 4.3.1: Classification of algorithms

This section deals with different types of algorithms which are tested to fit in the research model. The brief explanation is attached here about different algorithms to specify which algorithms fit in the model and the reasons behind rejecting some other algorithms.

4.3.1 Johnson's Algorithm**: -**

Johnson's algorithm is a method of sequencing jobs in two work centers. Its primary objective is to find an optimal sequence of jobs to reduce makespan. It also reduces the amount of idle time between the two work centers. Furthermore, the method finds the shortest makespan in the case of more than two work centers if additional constraints are met.

4.3.1.1 Processing n Jobs through Two Machines

Let there be n jobs, each of which is to be processed through two machines, M1 and M2 in the order M1 M2, i.e., each job must be passed through the same sequence of operations. In other words, a job is assigned on machine M1 first and after it has been completely processed on machine M1, it is assigned to machine M2. If the machine M2 is not free now for processing the same job, then the job must wait in the waiting line for its turn on machine M2, i.e., passing is not allowed. (Aalla, n.d.)

Since passing is not allowed, therefore, machine M1 will remain busy in processing all the n jobs one-by-one while machine M2 may remain idle for the second machine. Only by planning the order in which n jobs are to be processed on the two machines M1 and M2 would this be possible (Becker et al., 2015). The method Johnson suggests for figuring out the best order can be summed up as follows:

Step 1: List the jobs along with their processing times on each machine in a table as shown below:

Job Number	Processing Time on machine	
	M1	M2
1	T_{11}	T_{21}
2	T_{12}	T_{22}
3	T_{13}	T_{23}
…..	…..	…..
n	T_{1n}	T_{2n}

Step 2: Examine the columns for processing times on machines M1 and M2, and find the smallest processing time in each column, i.e., find out, min. (t_{1j}, t_{2j}) for all j.

Step 3: If the smallest processing time is on machine M1, then schedule the job as early as possible without moving jobs already scheduled, i.e., place the job in the first available position in the sequence. If the processing time is on machine M2, then schedule the job as late as possible without moving any jobs already scheduled, i.e., place the job in the last available position in the sequence.

There are three possibilities if choosing the shortest processing time results in a tie:

a. Minimum among all processing times is the same for the machine i.e., min (t_{1j}, t_{2j}) = t_{2k} = t_{2r}, then process the k^{th} job first and the r^{th} job last.

b. If the tie for the minimum occurs among processing times t_{1j} on machine M1 only, then select the job corresponding to the smallest job subscript first.

c. If the tie for the minimum occurs among processing times t_{2j} on machine M2, then select the job corresponding to the largest job corresponding to the largest job subscript last.

Step 4: Remove the assigned jobs from the table. If the table is empty, stop. Otherwise, go to Step 2.

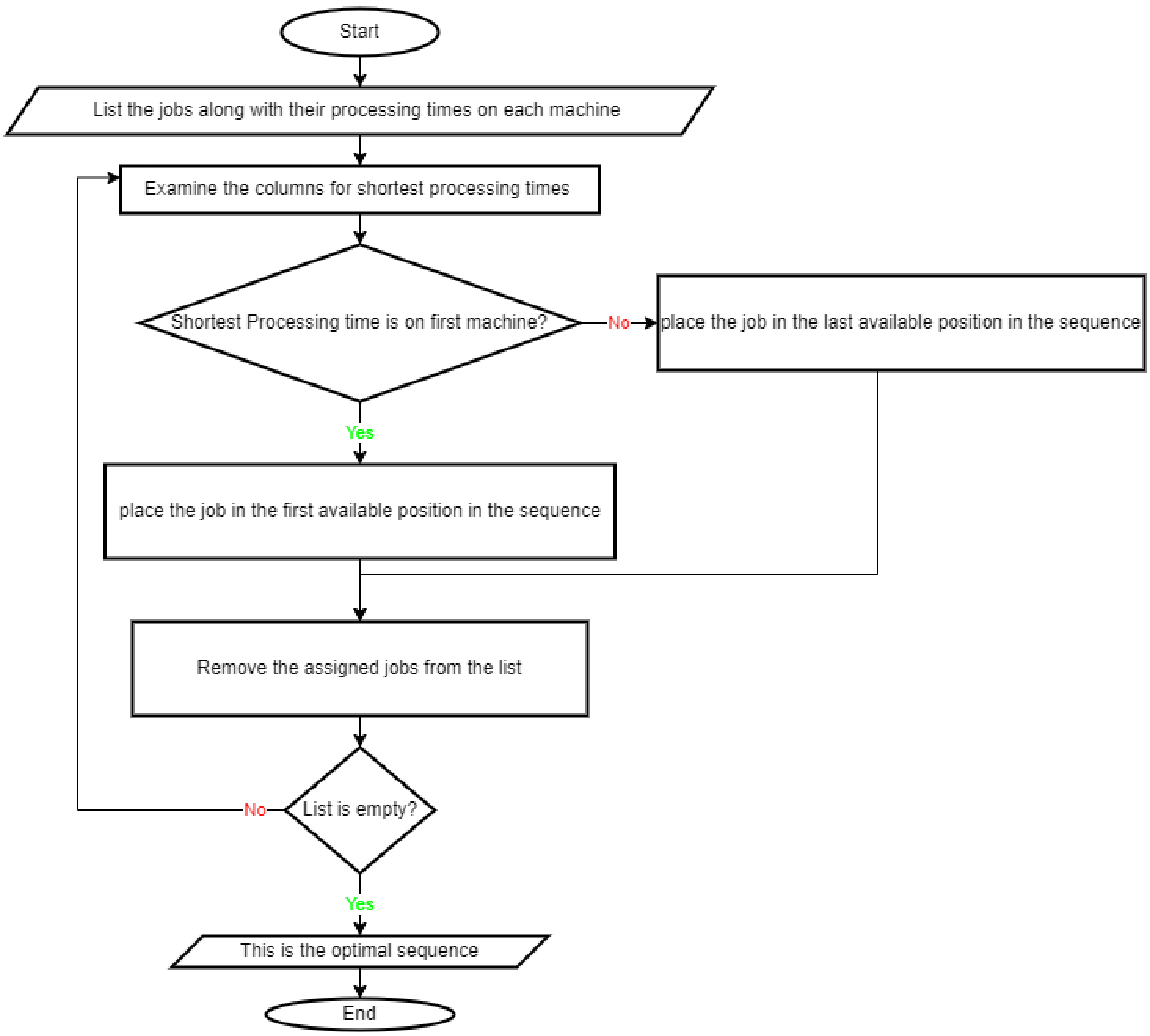


Figure 4.3.2: Flowchart of Johnson's Algorithm for N jobs through 2 machines

The model of this research problem deals with sequencing jobs in multiple work centers. So, these are the necessary steps to follow Johnson's Algorithm and the other constraints are also described here.

4.3.1.2 Processing n Jobs through m Machines

Let there be n jobs, each of which is to be processed through m machines, say M1, M2, ...Mm in the order M1, M2, ...Mm.

Job Number	Processing Time on machine					
	M1	M2	M3	…..	M(m-1)	Mm
1	T_{11}	T_{21}	T_{31}	…..	$T_{(m-1)1}$	t_{m1}
2	T_{12}	T_{22}	T_{32}	…..	$T_{(m-1)2}$	t_{m2}
3	T_{13}	T_{23}	T_{33}	…..	$T_{(m-1)3}$	t_{m3}
…..	…..	…..	…..	…..	…..	…..
n	T_{1n}	T_{2n}	T_{3n}	…..	$T_{(m-1)\,n}$	t_{mn}

Step 1: Find Min $\{t_{1j}\}$, Min $\{t_{mj}\}$ and max $\{t_{ij}\}$and verify conditions.

Condition 1: Min $\{t_{1j}\} \geq$ Max $\{t_{ij}\}$; j = 2, 3, ... m - 1

Condition 2: Min $\{t_{mj}\} \geq$ Max $\{t_{ij}\}$; j = 2, 3, ... m - 1

That is, the minimum processing time on machines M1 and Mm is as great as the maximum processing time on any of the remaining (m - 1) machines.

If either or both the conditions mentioned above hold, then go to step 2. Otherwise, the algorithm fails.

Step 2: Convert m-machine problem into 2-machine problem by introducing two fictitious machines, say

$t_{Gj} = t_{1j} + t_{2j} + t_{3j} + ... + t_{(m-1)\,j} = \sum_{i=1}^{m-1} t_{ij}$ j = 1, 2, 3, ...n.

i.e., processing time of n-jobs on machine G is the sum of the processing times on Machines

M1, M2 ......M (m – 1) j

$t_{Hj} = t_{2j} + t_{3j} + t_{4j} + ... + t_{mj} = \sum_{i=2}^{m} t_{ij}$ j = 1, 2, 3, ...n.

i.e., processing time of n-jobs on machine H is the sum of the processing times on Machines

M1, M2 ......Mmj.

Step 3: The new processing times so obtained can now be used for solving n-job, two machines equivalent sequencing problem with the prescribed ordering HG in the same way as

$t_{2j} + t_{3j} + ... + t_{(m-1)\,j}$ = constant for all j = 1, 2, 3, ... m - 1, then the optimal sequence can be obtained for n-jobs and two machines M1 and Mm in the order M1 Mm as usual.

If $t_{1j} = t_{mj}$ and $t_{Gj} = t_{Hj}$, for all j = 1, 2, 3, ... n, then total number of optimal sequences will be n and total minimum elapsed time in these cases would also be the same.

The approach described above is not the universal approach to solve the problem of sequencing n jobs among m machines. It only applies to specific situations where processing jobs through the first and/or last machine would cost more or take longer than processing jobs through the other machines.

For the data presented in table 4.2, any of the conditions mentioned before are not satisfied. So, the algorithm fails.

As Johnson’s Rule is not working for this model, the selection of an approximate solutions may help to solve this problem. For approximate solution, heuristics work better. Now, Different types of heuristics will be applied here to find out the best optimal solution for this model.

4.3.2 CDS (Campbell Dudek Smith) Algorithm**: -**

In order to find a better optimal solution, the CDS algorithm utilizes Johnson's method at each iteration.(Das, 2014)

(CAMPBELL HG et al., 1970) proposed a completely new heuristic strategy that essentially involves dividing the m machines into two groups, the first of which includes machines 1, 2,...., k, and the second of which includes machines m-k+ 1, m-k+ 2,...., m, where k = 1, 2,..., m -1(King & Spachis, 1980)

The combined times for every task i inside each of the two machine groups may be generated as follows if P_{ij} is the processing time for job i on machine j.

$$\sum_{j=1}^{k} P_{ij}$$

$$\sum_{j=m-k+1}^{m} P_{ij}$$

Johnson's technique can be used to solve this issue if we now think of the two groups as trying to make up an artificially produced two-machine problem with jobs times determined by the combined times above. Thus, the best possible sequence for this fictitious problem creates a permutation sequence for the original issue. For each of the possible m-1 artificial two-machine issues, this approach is repeated. The optimal answer to the original problem is determined by the heuristic solution that has the shortest makespan time.

Procedure for CDS Algorithm for m Machine: M1, M2, M3…... M (m) & n Jobs: -

Step 1: - Create (M-1) Sequence i.e.

S1	M1	M(m)
S2	M1+M2	M(m-1) +M(m)
S3	M1+M2+M3	M(m-2) +M(m-1) +M(m)
…..	…..	…..
S(m-1)	M1+M2+M3+…. +M(m-1)	M2+M3+…...+M(m)

Step 2: - Apply extension of Johnson' algorithm to each of the above (m-1) sequences.

Step 3: - Take the best possible makespan out of them.

Step 4: - CDS evaluates (m-1) sequences.

CDS Algorithm

```
Input job_matrix
M← number of machines
s = ∞
For (i=1; i<=M-1; i++)
    j ← sum of processing time of first i machines
    k ← sum of processing time of last i machines
    s* ← sequence obtained using Johnson's rule for n job 2 machines for j and k
    if (obj(s*) <obj(s))
        s← s*
output s
```

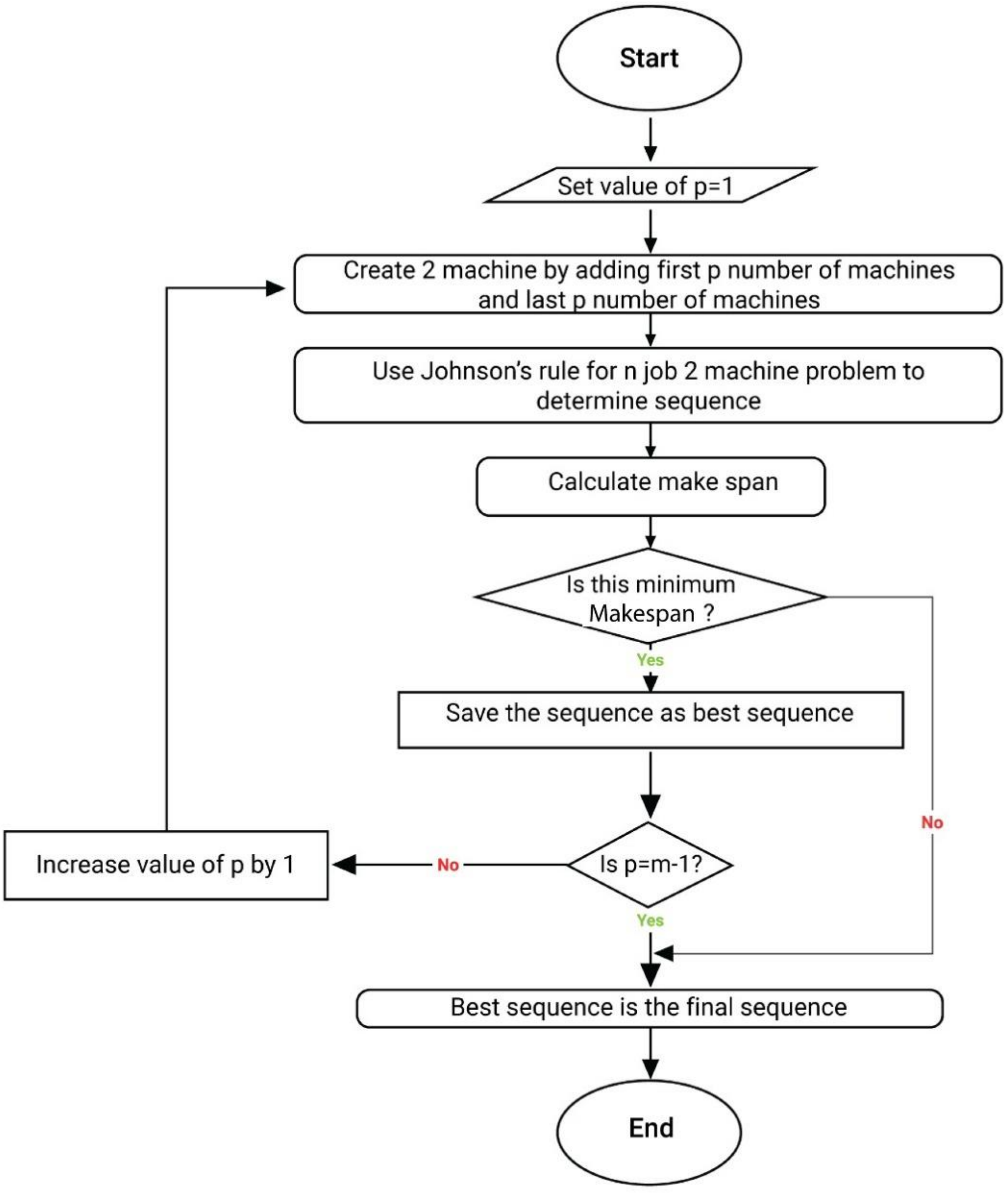


Figure 4.3.3: Flowchart of CDS algorithm

4.3.3 NEH (Nawaz, Enscore and Ham) Algorithm: -

According to Nawaz, Enscore, and Ham (1983), a task that takes longer to complete overall requires more attention than a task that takes less time (Byung Park, 1988). They proposed a new curtailed-enumeration algorithm (NEH) which finds the best partial sequence by an exhaustive search. It builds the final sequence in a constructive way, adding at each step a new job and finding the best partial solution. Each time a new job is added by fixing the relative position of the jobs already sequenced, the new job is tried at various relative positions and the best position is finalized.

Procedure for this NEH Algorithm: -

Step 1: - Calculate the total sum of processing time for each job.

Step 2: - Sort the jobs in the decreasing order of processing times.

Step 3: - Take the first to jobs in the sorted sequence and formulate different combinations.

Step 4: - Calculate the makespan for each of the combinations.

Step 5: - Select the combination with minimum makespan.

Step 6: - Insert the next job from the sequence obtained in Step 2.

Step 7: - Carry out all possible combinations of the 3 jobs now.

Step 8: - Repeat Step 4 followed by Step 5 followed by Step 6.

Step 9: - Continue the process till all jobs are completed.

NEH Algorithm

```
Input job_matrix
S ← a sequence obtained by sorting all job by summation of its processing time in descending order
S* ← the first job of S
for (t=2; t<= n; t++)
    i ← t-th job of S
    nowObj = ∞
    t* = 0
    for (t'=1; t'<=t, t'++)
        s' ← sequence by insert i into t'-th of S*
        if(obj(s') <nowObj)
            nowObj = obj(s')
            t* = t'
    S* ← sequence by insert i into t*-th of S*
output S*
```

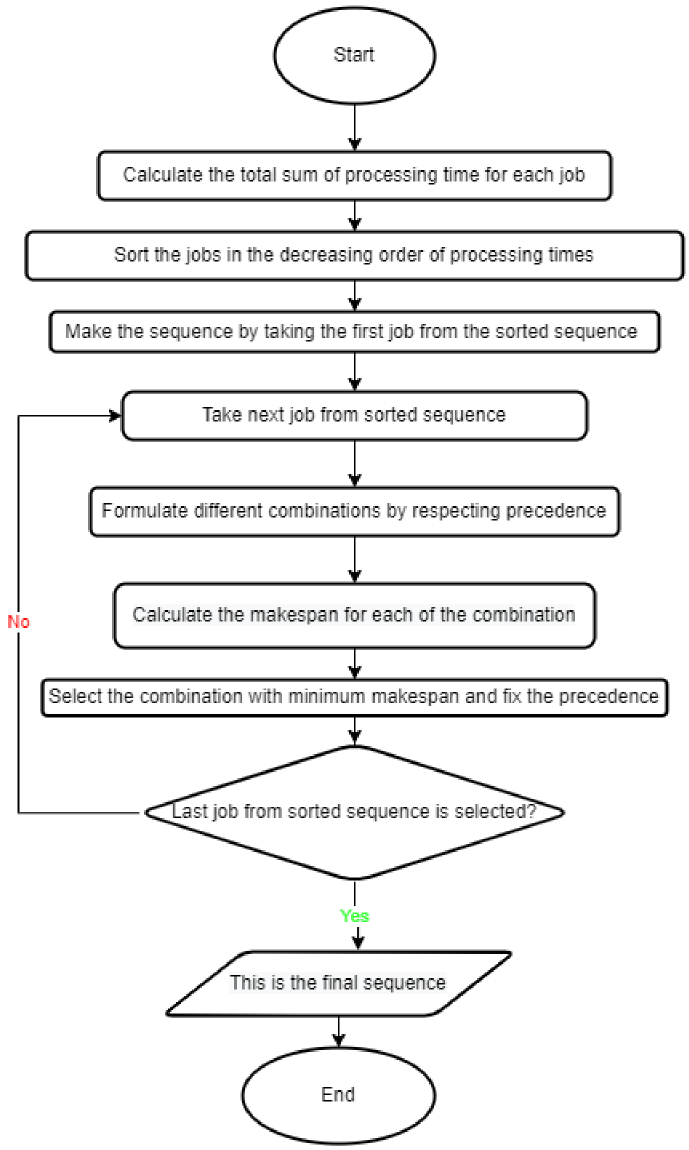


Figure 4.3.4: Flowchart of NEH Algorithm

4.3.4 Palmer’s Algorithm: -

To determine the sequence in which jobs should be processed in flow shop sequencing problems, Palmer proposed the concept of a job priority function, specifically a slope index for the job based on work processing times. The design of the priority function was chosen specifically to provide precedence to tasks that frequently progress from short to long processing periods as they move through the machines.(King & Spachis, 1980)

Procedure for this Palmer’s algorithm:

This method will try and find out a weighted sum for each of these jobs. So, assign weights to each of these machines and then try to find out the weighted sum of each job.

Step 1: - Consider a job sequencing problem for m machine and n jobs.

Step 2: - Assign some specific weights to each machine by the formula: -

S (j) = $-\sum_{j=1}^{m}[\big(m-(2*j-1)\big)*p(i,j)$

Sep 3: - Evaluate the weight of each job by a given procedure by multiplying the weights with the processing times.

Step 4: - Sort the jobs in the decreasing order of their weights.

Step 5: - Formulate a sequence based on the sorting done in Step 4.

Step 6: - Calculate the makespan for the above sequence.

According to this algorithm, machines are weighted. The research problem deals with the machines where all of them carry the same importance. So, this is not a proper fit for the research model.

4.3.5 Dispatching Rules

Examples of the construction heuristics are dispatching rules. Each job that is awaiting processing on a machine is given a priority using dispatching rules. A dispatching rule examines the waiting jobs whenever a machine has been freed and chooses the one with the highest priority. A job's priority is established based on its requirements, the requirements of the machine, or the features of the shop. Jobs are sorted after their priorities have been established, and the one with the highest priority is then chosen to be processed first (Nageswara et al., 2017).

Some of the dispatching rules that have been created, examined, and put into practice include the following:

- Shortest Processing Time (SPT): The work with the least operation processing time is processed first, and this is also known as the Shortest Expected Processing Time. There are numerous alternatives: Least Work Remaining (LWR) in terms of the number of operations, Total Work in terms of Processing Time (TWORK), Truncated Shortest Processing Time (TSPT), Weighted Shortest Processing Time (WSPT), Total Shortest Remaining Processing Time (SRPT), and Assembly Jobs First with Shortest Processing Time (AJF-SPT).

- Longest Processing Time (LPT): The job with the longest operation processing time is processed first, also known as Longest Expected Processing Time (LEPT). Other versions include: Most Work Remaining (MWR) in terms of the number of operations, Total Longest Processing Time (TLPT), and Total Longest Remaining Processing Time (LRPT).

The above-mentioned dispatching rules prioritized jobs based on processing time. In the research model, the processing time is mentioned. But there is some precedence which each category of jobs needs to be followed and this rules specially followed by single machine problem. So, these rules cannot be utilized in the research model.

- Earliest Due Date (EDD): The task that has the earliest due date gets processed first. There are some variations: Operation Due Date (ODD), Modified Due Date (MDD), Modified Operation Due Date (MODD).

- Job Slack Time (JST): The job with the least amount of slack is completed first. The time between the job due date, the amount of work still to be done, and the current time is used to calculate the job slack time. The variations are Operation Slack Time (OST or S/OPN), Allowance over remaining number of operation (A/OPN), Slack Time over Allowance (S/A), Weighted Processing Time plus Weighted Operation slack Time (WPT+WOST), Slack Over Remaining work Time (S/RPT) (Al-Harkan, 2010).

- Critical Ratio (CR): First, the job with the smallest ratio is completed. By dividing the job's allowance by the remaining work time, one can get the critical ratio. The Critical Ratio version known as Operation Critical Ration (OCR) selects which operation has the smallest ratio and gets handled first. By dividing the operation's allowance by the operation process time, the operation critical ratio is calculated.

The above-mentioned dispatching rules prioritized jobs based on due dates. For the lab environment used for this research, there is no mentioned due date for each job assignment and no previous data is found for further analysis.

- Random: A job is chosen at random from the group of jobs that are in queue at the machine. Jobs are equally likely to be chosen from the waiting list when the random rule is used. The biased-random rule, however, does not select jobs with an equally high probability. According to one or more of the other dispatching rules, the selection process is biased. First, a group of jobs that are waiting are sorted by dispatching rules before applying the biased-random. The waiting jobs are then sorted using the chosen dispatching rule. Next, selection probabilities are assigned to the jobs in the ordered list. These probabilities are typically estimated using a geometric distribution. The position with the highest selection probability will be in the first place, and the position with the lowest selection probability will be in the last place. By doing this, the jobs that are listed first in the ordered list are more likely to be chosen, but the ones that are listed last are less likely to be chosen (Al-Harkan, 2010).

- First Come, First Served (FCFS): This is also called Smallest Ready Time (SORT) which selects the job which arrives first at the machine will be served first. First Smallest Release Time (SRT) or First Served (FASFS) deals with a work arriving at the shop first being given precedence to go first in all machines.

- Last Come First Served (LCFS): The job that arrives last will be served first

- Least Flexible Job (LJF): The least flexible job is the job that requires the least amount of flexibility.

- First Off, First On (FOFO): Even if the operation is not yet in the queue, the job with the operation that might be finished the earliest will be processed first. In this scenario, the device will be inactive until the time of the operation.

- Least Anticipated Work in Next Queue (LAWINQ): A work that will face the least queue at the next machine along its journey will be chosen from the group of jobs awaiting a certain machine.

The above-mentioned dispatching rules prioritized jobs based on shop and/ or job characteristics. The rules are a bit arbitrary based on system data and there is no previous analysis of this data to go further.

- Cost OVER Time (COVERT): A composite rule known as Cost OVER Time (COVERT) places the task with the highest COVERT ratio in first place. The COVERT ratio is calculated by dividing the expected tardiness for the associated job by the processing time for that activity. Apparent Tardiness Cost (ATC) and Apparent Tardiness and Earliness Cost (ATEC) are two versions of the rule (Al-Harkan, 2010).

A ranking expression that combines a few basic dispatching rules is called a composite dispatching rule. The scaling parameters for each basic rule in the composite dispatching rule are specifically chosen to scale the basic rule's contribution to the total ranking expression.

Dispatching rules are quick, easy to use, and can provide a reasonably good solution in a short amount of time. The solution might be the best one in some unique situations. However, their usefulness in real-world situations is constrained since occasionally they can produce unexpectedly poor results (Hübl, 2018).

Among the algorithms described in this section, CDS Heuristics and NEH heuristics fit in the model. The optimal sequence received from this model will be adjusted with the precedence and pallet constraints. The final optimal solution will be implemented in a simulated environment and lab environment to compare the total makespan for each algorithm. The Arena simulation software is used to replicate the lab environment of this model.

CHAPTER 5

DATA PROCESSING AND VALIDATION

In this chapter a model for the theoretical framework above described is presented as well as an implementation of the lab environment for its validation. The framework has been conceived in order to pursue a first testing of the proposed idea for the general theoretical framework. The validation test has been performed exploiting Dr. Thorsten Wuest's Smart Manufacturing Laboratory at West Virginia University (WVU).

5.1 Preparing Input Dataset:

Using Jupyter notebook, a python code is prepared to input processing times of each category of jobs. For each different category of products, the processing time is different. The number of necessary cycles is also different for each category of jobs. By assigning machine name, number of cycles, processing time for each cycle and number of jobs on each category is forms a matrix as input dataset.

Next step is to process each job as a virtual job and assign them in a processing order considering the available number of pallets. The file is stored as CSV format for further analysis. Table 5.1 is an example of each equal category job shop.

5.2 Algorithm Inputs and Outputs Processing:

For each algorithm input, table 5.1.1 is converted to figure 5.2 which is actually the .txt version of that previous CSV format with some manual input as machine number and number of virtual jobs.

After completing calculation of specific algorithms another CSV file is produced with optimal sequence for virtual jobs (Table 5.2).

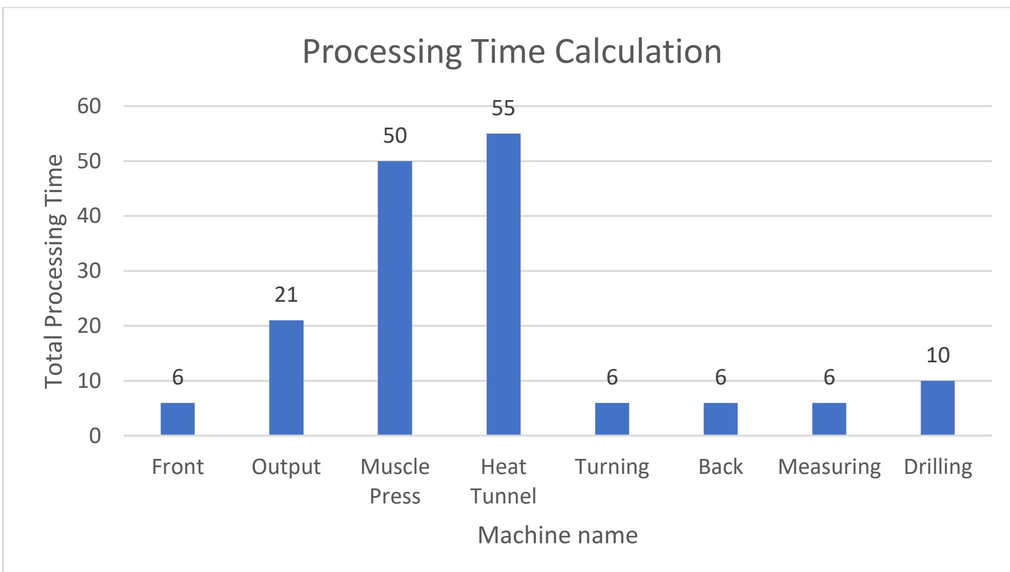
5.3 Conversion of Optimal Virtual Job to Machine Setup:

From the previous step, the output CSV file (Table 5.2) is imported here as input. After necessary calculation to adjust with machine setup it returns with reorganized optimal sequence. This reorganized optimal sequence worked as lab environment input and simulated environment input.

5.4 Mathematical Calculation:

Processing times are the primary input for all jobs. For a fixed sequence, total processing time will be the same for all scenarios (Table 5.4.1). A graph 5.4.1 is plotted based on that data. This graph represents the total processing time for each machine for a fixed sequence. Total processing time represents the sum of the processing time for a particular machine travelled by different categories of jobs in that ongoing sequence. As an example, front machine, back machine and measuring machine are travelled by all categories of jobs. So, for the considered sequence, there are three jobs from different categories. The average processing time considered for this calculation is 2 seconds. So, total processing time represents the sum of the processing time for each category of jobs which is 6 seconds.

Graph 5.4. 1: Sample Processing Time



To calculate makespan, a machine in/ out table is formed after necessary calculation (Table 5.4.2). Machine in/out table actually represents the total time a particular job takes to complete the whole process and necessary waiting time if the previous job is under work in process at that certain moment. It also includes the machine idle time. For example, medium cycle 1.4 completed processing in the front machine at 14 seconds. But, as the complex cycle 4 starts working on the output machine at that time, medium parts need to wait till 28 seconds to start its processing. This time can be considered as the bottleneck for medium cycle 1.4. Next, for all other machines, as transportation time is not adjusted, the machines behave like idle stations for all other cycles. Finally, the last value found from this table is the makespan time for the whole process.

Idle time for all machines is calculated and stored in a CSV file (Table: 5.4.3). As previously mentioned, the sum of individual idle time for each machine is presented as total idle time. The sum of idle time is huge compared with processing time, as virtual jobs are travelling in a loop to complete all the process. This large idle time represents the machine is not efficient enough to utilize all the resourses due to this complicated machine setup. Graph 5.4.3 is plotted based on that data. A Gantt chart is presented on figure 5.4.1 to show the idle time.

Graph 5.4. 2: Sample idle time

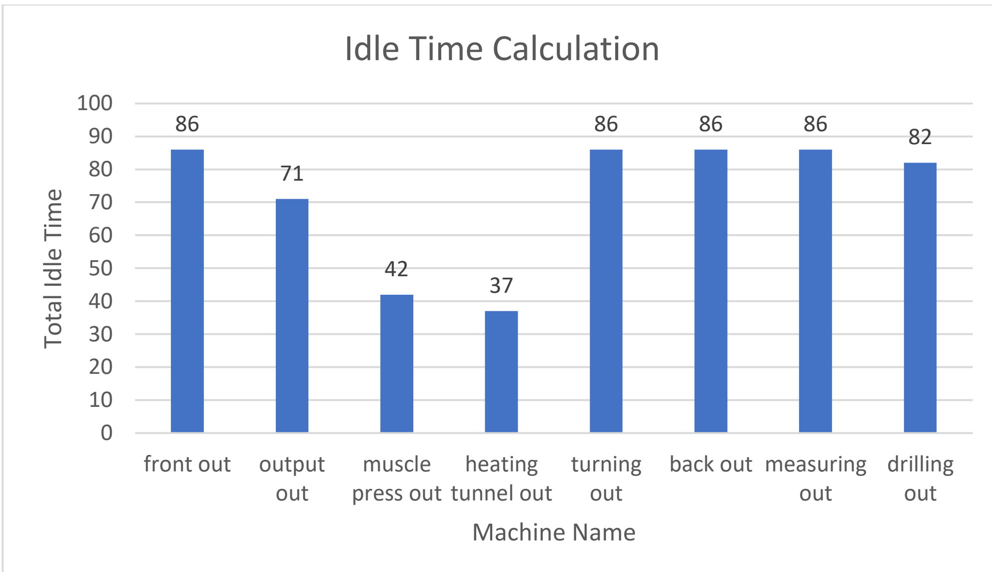


Figure 5.4. 1 Gantt chart showing idle time

Bottleneck for all jobs is calculated and stored in a CSV file (Table 5.4.4). Graph 5.4.4 showed the sub of possible bottlenecks for all jobs. Similarly, Figure 5.4.2 presents a gantt chart to show the bottleneck for all jobs.

Graph 5.4. 3 Sample bottleneck Calculation

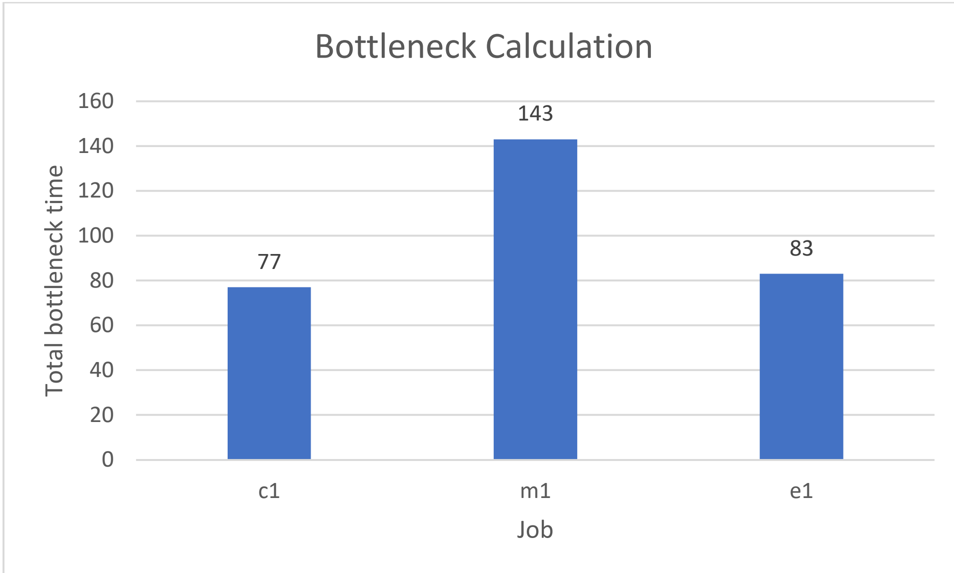
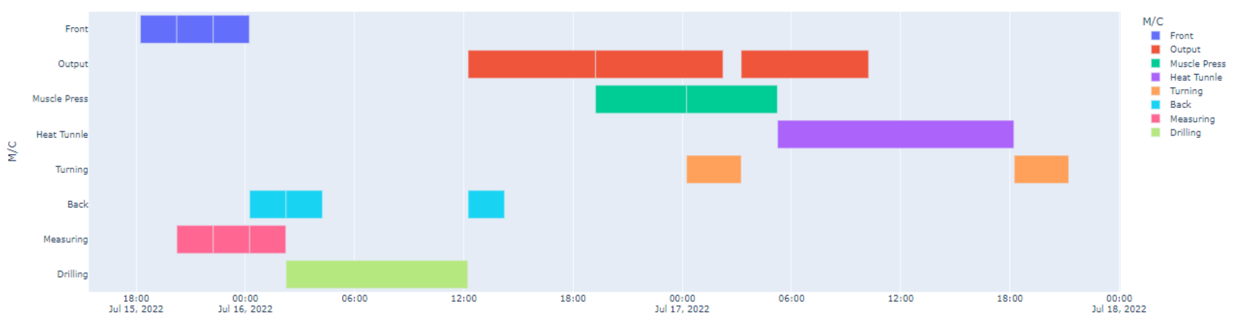


Figure 5.4.2: Gantt chart showing bottleneck



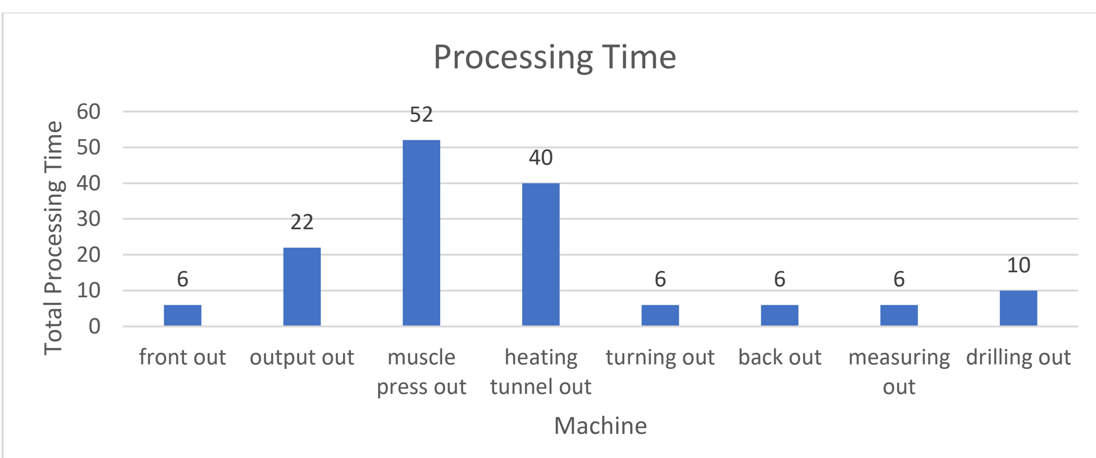
5.5 Machine Data Transformation and Pre-Processing:

This step consists in the automated transformation and pre-processing of the dataset in order to obtain an organized file. As data is coming from an excel format file, the output of this transformation is in excel format as well. The preprocessing of the data consists in an analysis of the raw data in order to extract a first level of information. This pre-processing is considered to be case specific and depends on data requirements. It is necessary to consider the type of analytics that has to be performed in the end, as well as the final purpose of the knowledge that could be extracted by the analyzed data. Based on that, a first screening of the data is performed in order to understand what subsets of data are relevant for that purpose. The transformation of the original dataset into an organized data consists in retrieving only the relevant collections of data and rearranging them in order to reach a first level of information. Automatizing this task would reduce potential errors and reduce the time to perform it, especially if it is performed repeatedly. With the purpose of automating this phase, the tools that have been considered and selected are python.

The dataset retrieved from the MES of WVU SM Lab is an excel format file with a table containing a variety of raw data about the production process of the defined production plan. In order to obtain meaningful information, it is necessary to transform and pre-process that data. Not all the raw data here presented are useful to perform an analysis. Therefore, it is necessary to understand which type of data is valuable for the specific purpose. For this specific implementation of the framework, it has been decided to perform a simulation analysis to optimize the total makespan of a defined production plan. Arena simulation has been selected as analytical tools to accomplish this study. Carefully analyzing the dataset and considering the purpose of the analysis, it was possible to determine the crucial information required as input for the analytical simulation tool. In order to carry on this type of analysis it is necessary to know the number of job orders present in the production plan, as well as the number of products required for each defined typology. To get this first level of information, it is required to filter the original set of data, understanding what the beneficial raw data are, then it is fundamental to clean those data and initiate a first transformation process. Pre-processing the obtained output is analyzed in order to get to the comparison for a simulated environment. Thoroughly studying the original dataset, it has been decided to consider as valuable the columns of product number, the code identifying the type of product ("PNo"); step number, the code identifying the module that has been working on the product ("StepNo"); the order number, the unique code identifying the job order ("Ono") where all the unique job orders are listed, specifying the product type code and whether each of the eight modules has worked the relative product or not. For Arena simulation it requires .TXT format typology to read data. Indeed, the here obtained file contains the number of total job orders present in the production plan, as well as the way they are split among the three specified product typologies.

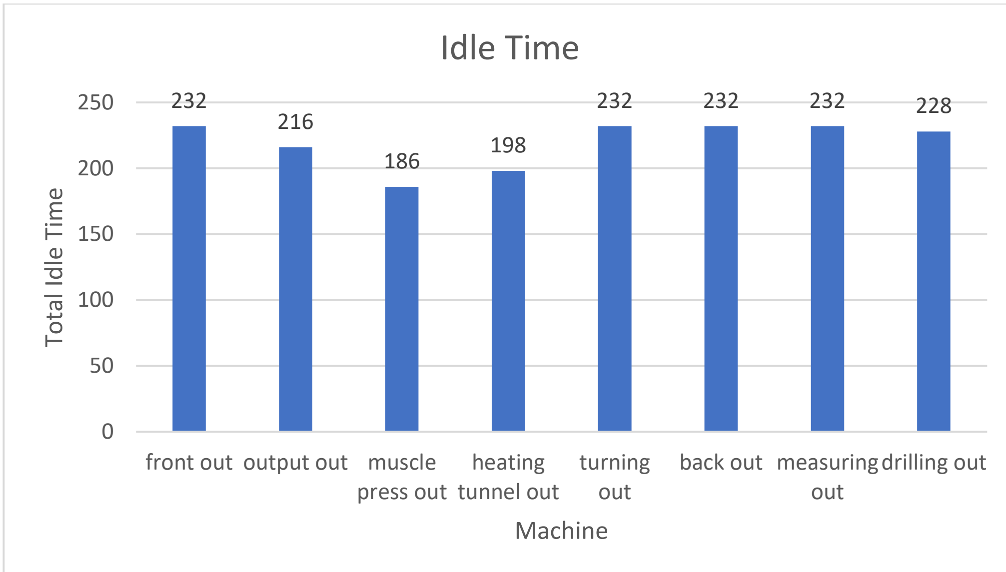
Table 5.5 represents the CSV file exported from MES software. This dataset works as input for processing machine data. By calculating makespan, processing time, idle time and bottleneck graph 5.5.1, graph 5.5.2, graph 5.5.3 is presented.

Graph 5.5. 1 Total processing time



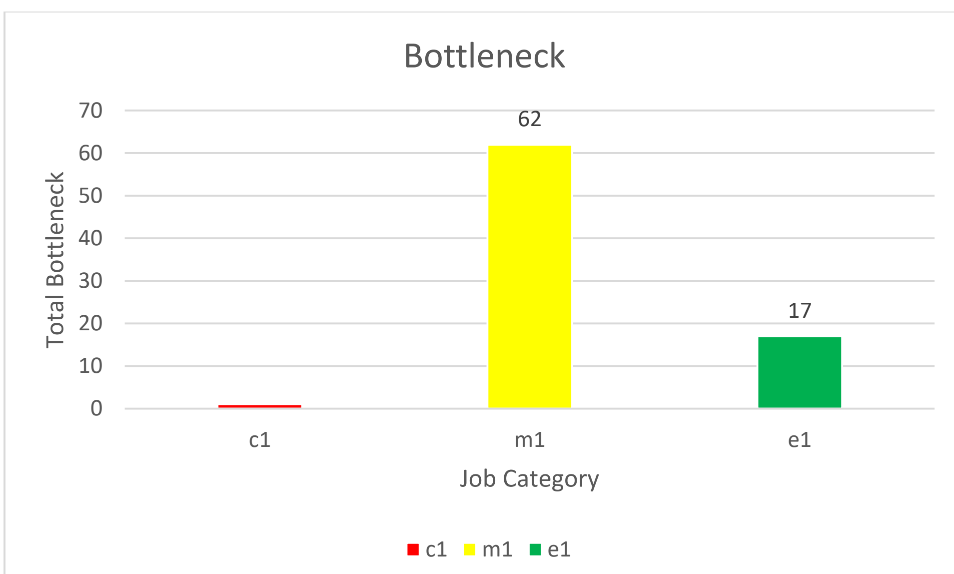
Total processing time should be the same for both input dataset and machine. There may be some variations due to some uncontrollable factors in lab machines.

Graph 5.5. 2: Total idle time



Idle time obtained from machine data is comparatively higher than simulated environment due to additional transportation time. The machine which has the longest processing time has the least idle time. Due to multiple necessary cycles to complete jobs, machine utilization is very low. The comparison between processing time and idle time figures out this issue.

Graph 5.5. 3: Total bottleneck



For bottleneck calculation, the first job which is input in the machine has no bottleneck at all. The next job is from the medium category, so it needs to travel an extra cycle rather than an easy one.

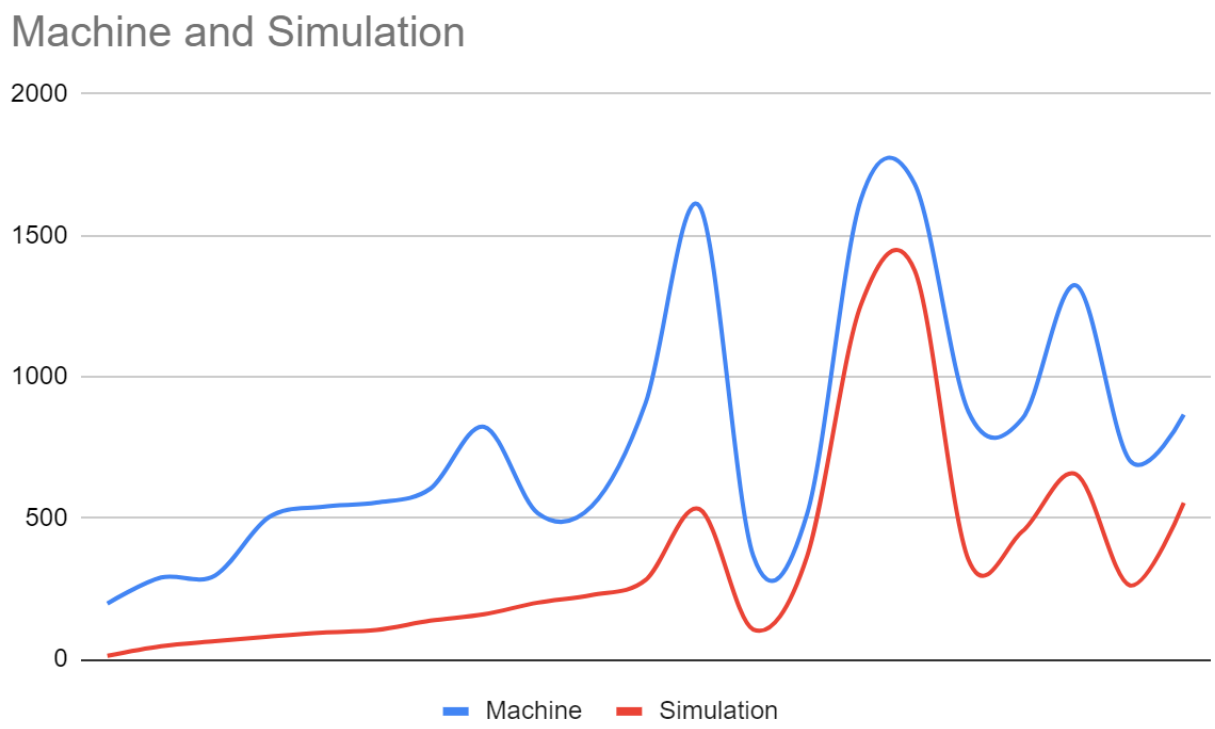
That’s why, easy job enters at last in the machine, but is disposed of earlier than a medium job. So, the bottleneck is high for medium category jobs.

For result analysis, makespan is chosen for all comparisons because it considers all of the times to complete a sequence. Processing time, idle time and bottleneck data already existed in makespan time. So, minimizing makespan will obviously help to minimize bottleneck and increase utilization for this custom machine setup.

5.6 Modeling and Validation:

The modeling and the validation phases of the framework presented are strictly connected and they have been carried out in parallel. Nevertheless, the models here presented can be easily adjusted in order to adapt to specific cases. The WVU Smart Manufacturing Lab, endowed with a Festo Didactic Learning System, has been chosen in order to practically implement the framework and validate it. The idea is to prepare a production plan of an optimized sequence of orders and launch its production on the system, monitoring the total makespan time necessary for its completion. A set of manufacturing data about the performed production can be retrieved allowing for the automatic flow and processing of these data. A first level of information is obtained and can be read by the simulation software Arena where a simulation model of the lab system is built. Through a simulation analysis, it is possible to optimize the total makespan time by finding a relative optimum one relative to a different production sequence that can be launched in production on the WVU SM Lab system. In the end, making a comparison between the total makespan time of the sequence coming from the analysis and the original makespan time relative to the optimized sequence, it should be possible to discover an improvement in the makespan time changing the job category and number of jobs. In this way, it is possible to create a loop that allows testing the functioning of the framework from the beginning to the end.

Graph 5.6. 1: Comparison of data for machine and simulated environment



Graph 5.6 represents the comparison between machine data and simulated data. Machine data collected randomly for different numbers of jobs for all environments (flow shop and job shop). As collecting machine data is time consuming and adding transportation time does not create any effect on optimal sequencing, the comparison shown in the next chapter is based on simulated data. For machine data, there is some randomness at the very beginning part of this graph but for simulated environments it is smooth enough. For the rest of the graph showing almost the same trend of change. The range between two-line charts has some variation, as the dataset is large and random.

5.7 Simulations Analysis

Once the simulation model has been completed, it is possible to run the simulation and analyze the output. It has been decided to perform single replications in one simulation in order to obtain a fairly good result. In each replication, the optimized sequence input is unique, therefore all the replications refer to different production sequences of the job orders. From the output report generated by Arena software, it was possible to find the replication with the lowest makespan time for the respective production sequence. The sequence of 9 jobs with equal percentage of product category selected correspond to a replication time equal to 262 seconds (4 minutes and 36 seconds), The sequence selected as the relative optimum one from the results of the simulation performed on Arena, has been implemented on the Festo Didactic System in order to monitor the real throughput time. This sequence was launched in production on 10.23.2022 at 16:53:00, and it ended at 17:04:43, with a total throughput time of 11 minutes and 43 seconds. It is evident how the real time differs from the replication one. The reason resides in the fact that the simulation model does not consider the transportation time and the non-value-added time that is the time that the product is stopped at each machine to allow sensor reading.

5.7.1 Arena Simulation

More than ever, manufacturing operations must increase their efficiency. They are also specializing at the same time. Understanding the interactions, variability, and resource interactions of a current manufacturing system is crucial when optimization of the system is being addressed. Manufacturing depends on these interactions, which are nearly impossible to model in a spreadsheet. A computer-based model of an actual manufacturing process that can be used to validate, test, and enhance the performance of the process can be used to address manufacturing optimization. It involves imitating reality in the digital realm, and the outcomes can be utilized to forecast and enhance actual performance rapidly, inexpensively, and with less risk than tests in the physical realm (Kelton et al., 2010). Therefore, for the implementation of the prototypical framework it has been chosen to perform analysis on the data available. In manufacturing simulation has been widely adopted and it is very flexible and useful to solve manufacturing problems rather than using complex mathematical models (Brunner & Funck, 2015). In fact, it is useful when it comes to evaluate different manufacturing scenarios with the aim of improving productivity and decreasing bottlenecks or it can help decrease the process cycle time, increase resource utilization etc. (Zahraee et al., 2014).

Simulating a process or system's behavior over time is referred to as simulation. To simulate the behavior of real systems, a wide range of techniques and tools are used, therefore simulation involves genuine systems as well as models of such systems. A real or planned facility or process, such as a manufacturing plant with machines, workers, transportation equipment, conveyor belts, and storage space, is referred to as a system. A system is frequently studied to assess its effectiveness, enhance its functionality, or create it from scratch if one doesn't already exist. Using software created to replicate the actions and characteristics of the real system over time, computer simulation enables the study of a model of a real-world system to comprehend its behavior for a given set of conditions by carrying out numerical experiments. To perform a simulation, a model needs to be created on the computer software. A model can be either static or dynamic when time is involved. Operational models are typically dynamic. Then, it may be continuous or discrete; in the former, the system's state is updated only at distinct intervals of time, whereas the last is updated constantly over time. In a manufacturing system, for example, it happens when parts arrive or leave the system and its resources. Finally, it can be deterministic if no random input is considered, otherwise it is stochastic if inputs are random and defined by probability distributions.

Arena simulation has been selected as software to perform this step in the simulated environment (figure 3.2.1). Arena models dynamic processes using a flowcharting, entity-based approach. It has a vast library of pre-built components that may be used to simulate any process. The user can build a model by combining modules present in the software that represent processes and logics that imitate the real system. Connector lines join the modules and specify how entities flow in the system. Arena is very user friendly; its modeling methodology is easy and intuitive without the need for customized code or programming. Most other commercial simulation systems are code-based and demand scripting in exclusive languages. Arena is simpler to learn than other simulation programs, easier to validate and debug, and simpler to explain the details of complex processes to others thanks to its flowchart style. Arena has been leading the simulation industry for the past 24 years. Arena has been available for close to 25 years. During this time, a lot of additional simulation companies have entered and exited the market. No other simulation provider has endured and prevailed in Arena. Additionally, arena advice and assistance are simple to locate, it is the most thoroughly documented simulation program available, and it has academic support. Finally, this software can be integrated with Microsoft technologies as reading from or sending output to MS Excel spreadsheets. Considering that, for the implementation and validation of the framework no data conversion is needed and it is possible to directly import the excel file into Arena.

5.7.2 Data Analytics

In the development of this framework, it has been decided to pursue an optimization analysis of a predefined production plan. Specifically, the aim was to improve the makespan of the implemented production plan and demonstrate this feasibility by applying a simulation-based analysis. Arena simulation software has been selected as a tool to perform this type of analysis. Given the set of job orders present in the production plan and the willingness of improving their makespan, it has been decided to create the model so as an optimized sequence of these job orders is created at one replication. The aim is to find a sequence who's associated makespan is lower compared to the monitored one on the production system. Nevertheless, it is not possible to find the very optimal time as it is not feasible to try all the possible combinations of the job orders and the algorithm applied for optimized sequence are all heuristics. Therefore, it has been decided to accept a relatively optimum solution with the aim of demonstrating that by implementing this framework and applying a simulation analysis it is possible to find a sequence of the job order whose makespan is improved compared to the original random sequence.

Using Arena simulation software and its flowcharting methodology, it has been possible to create a digital model of the CP Festo Didactic Learning Systems present in the WVU Smart Manufacturing Lab. In the construction of the model, blocks and elements have been used. The overall model can be divided into 10 groups: the reading part of the input data, the entity creation part and the 8 stations corresponding to the 8 modules in the real system. In this section they are going to be described in detail following the order according to which the machines are physically ordered. First, it is possible to define all the elements needed for the correct functioning of the model.

Elements

Files	Entities	Resources	Attributes	Sequence	Queues	Stations
Input File	Part	Front Machine	Front	Easy Job Cycle 1	Front Machine. Queue	Station Front
	Pallet	Measure Machine	Measure	Easy Job Cycle 2	Measure Machine. Queue	Station Measure
		Drill Machine	Output	Easy Job Cycle 3	Drill Machine. Queue	Station Back
		Back Machine	Drill	Medium Job Cycle 1	Back Machine. Queue	Station Press
		Press Machine	Back	Medium Job Cycle 2	Press Machine. Queue	Station Heat
		Heat Machine	Press	Medium Job Cycle 3	Heat Machine. Queue	Station Turning
		Turning Machine	Heat	Medium Job Cycle 4	Turning Machine. Queue	Station Drill
		Output machine	Turn	Complex Job Cycle 1	Output Machine. Queue	Station Output
			Parts	Complex Job Cycle 2	Seize Pallet. Queue	Easy Cycle 2
			Pallet	Complex Job Cycle 3	Match. Queue Pallet	Easy Cycle Out
				Complex Job Cycle 4	Match. Queue Product	Medium Cycle 2
						Medium Cycle 3
						Medium Cycle Out
						Complex Cycle 2
						Complex Cycle 3
						Complex Cycle Out
						Finished

Figure 5.7.2.1: Arena model elements

- **Files:** It defines the Input File (i.e., final optimal sequence) where data must be read. It is necessary to specify the file path on the computer, the access type that is text, the end of file action that is dispose as once read the file is no longer needed

- **Entities:**
  - ✓ An entity called Part, referring to a generic part.
  - ✓ An entity called Pallet, referring to the pallet carrier that moves the product around.

- **Resources:** They are called Front Machine, Measure Machine, Drill Machine, Back Machine, Press Machine, Heat Machine, Turning Machine, and Output machine representing the 8 modules/machines that work the pieces.

- **Stations:** They are called Station Front, Station Measure, Station Back, Station Press, Station Heat, Station Turning, Station Drill, and Station Output that represent the 8 different steps that each job order must go through and are necessary to model the way the station are ordered and therefore the route the pieces need to follow. There are some additional stations for specific categories of products as models need some cycles to complete each job. For easy jobs there are two additional stations named easy cycle 2 and easy cycle out. Similarly for medium and complex jobs there are some stations named medium cycle 2, medium cycle 3, medium cycle out, complex cycle 2, complex cycle 3 and complex cycle out. Finally, there is a station named Finished from where the completed jobs go to the disposal module.

- **Queues:** Front Machine. Queue, Measure Machine. Queue, Drill Machine. Queue, Back Machine. Queue, Press Machine. Queue, Heat Machine. Queue, Turning Machine. Queue, and Output Machine. Queue, referring to the possibility of product queuing in front of each machine. Similarly, there is a queue in front of the seize pallet module which is named Seize Pallet. Queue. There are additional two queues in front of the Match module to create a batch of a pallet and a product named Match. Queue Pallet and Match. Queue Product.

- **Attributes:** They are called Front, Measure, Output, Drill, Back, Press, Heat, Turn and are necessary to trace the machine for each product type. Product Type Easy, Medium and Complex is defined by other attributes named Parts. Remember that a product of type Easy is doing only 4 steps (Front Machine, Measure Machine, Back Machine and Output Machine), while a product of type Medium is doing 6 steps (Front Machine, Measure Machine, Back Machine, Press Machine, Turning Machine and Output Machine), and a product of type Complex is doing all the 8 steps (from M1 to M8). Another attribute is mentioned as a pallet.

- **Sequence:**

✓ For Sequence Easy Job cycle 1 follows:
Station front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Easy cycle 2

✓ For Sequence Easy Job cycle 2 follows:
Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Easy Cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Easy Cycle Out

✓ For Sequence Easy Job Cycle 3 follows:
Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Easy Cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Easy Cycle Out→ Station Front→ Station Output→ Station Finished

✓ For Sequence Medium Job Cycle 1 follows:
Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium Cycle 2

✓ For Sequence Medium Job Cycle 2 follows:
Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium Cycle 3

✓ For Sequence Medium Job Cycle 3 follows:
Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium Cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station

Measure→ Station Drill→ Station Medium Cycle 3→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium Cycle Out

- ✓ For Sequence Medium Job Cycle 4 follows:
  Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium Cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium Cycle 3→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Medium Cycle Out→ Station Front→ Station Output→ Station Finished

- ✓ For Sequence Complex Job Cycle 1 follows:
  Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle 2

- ✓ For Sequence Complex Job Cycle 2 follows:
  Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle 3

- ✓ For Sequence Complex Job Cycle 3 follows:
  Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle 3→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle Out

- ✓ For Sequence Complex Job Cycle 4 follows:
  Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle 2→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle 3→ Station Front→ Station Output→ Station Press→ Station Heat→ Station Turning→ Station Back→ Station Measure→ Station Drill→ Station Complex Cycle Out→ Station Front→ Station Output→ Station Finished

Read Input Data

The blocks needed to allow Arena to read the necessary input data are a create and a readwrite. The readwrite block is triggered by the flow of an entity, therefore the create block is necessary to generate an entity that has been previously defined as Part. As the only purpose of this entity is to activate the readwrite block, it can be directly disposed of and removed from the overall system. A create block generates an entity at time 0 that flows into and triggers the readwrite block which respectively reads form the Input File the value of the File elements.

Processing Time Identification

In the construction of the simulation model, for each station for each machine it is necessary to define the processing time. This data is known and does not depend on the number of products present in the production plan or their typology. Nevertheless, it has been discovered that the processing time of each module is not fixed, but it may vary by a few seconds in the processing of the different products. Machine processing times have been recorded and for each module for each single piece the processing time has been analyzed. Therefore, it has been computed the average processing time corresponding to each machine, then it has been rounded up to the closest integer. This time has been selected as processing time for each machine.

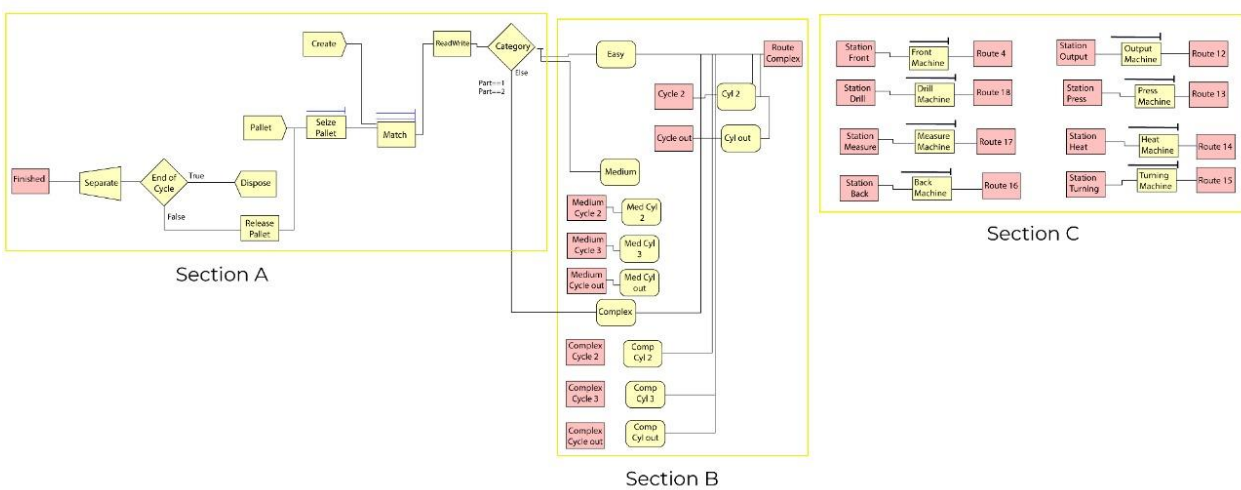


Figure 5.7.2.2: Arena Modelling

Section A Input and Output of the Model

Input:

In the group of the model, at the beginning it is necessary to consider the presence of the 8 pallet carriers that are in queue ready to receive a product. For this purpose, a Creation block is needed to create exactly 8 entities Pallet at time 0. Then there is a Seize module where the resource is PalletCar and the resource capacity is fixed with 8.

Another Create module is used for Part arrival to create exactly the number of entities Part at time 0.

A match module is used for creating a temporary batch based on entities when exactly one pallet and one part will match. The group is temporary, which means that at the end of the simulation the two entities need to be separated again, because the product must leave the system after its processing, while the pallet keeps going through the machine and accepts new products. As the number of pallets is lower than the number of parts, parts should wait till the next pallet will return after the release pallet. After receiving the released pallet, they seized the pallet again and made another temporary batch using the match module.

A readwrite module is used for reading data from csv input file considering attribute type part.

A decision module is used for categorizing the attribute part as easy, medium, and complex.

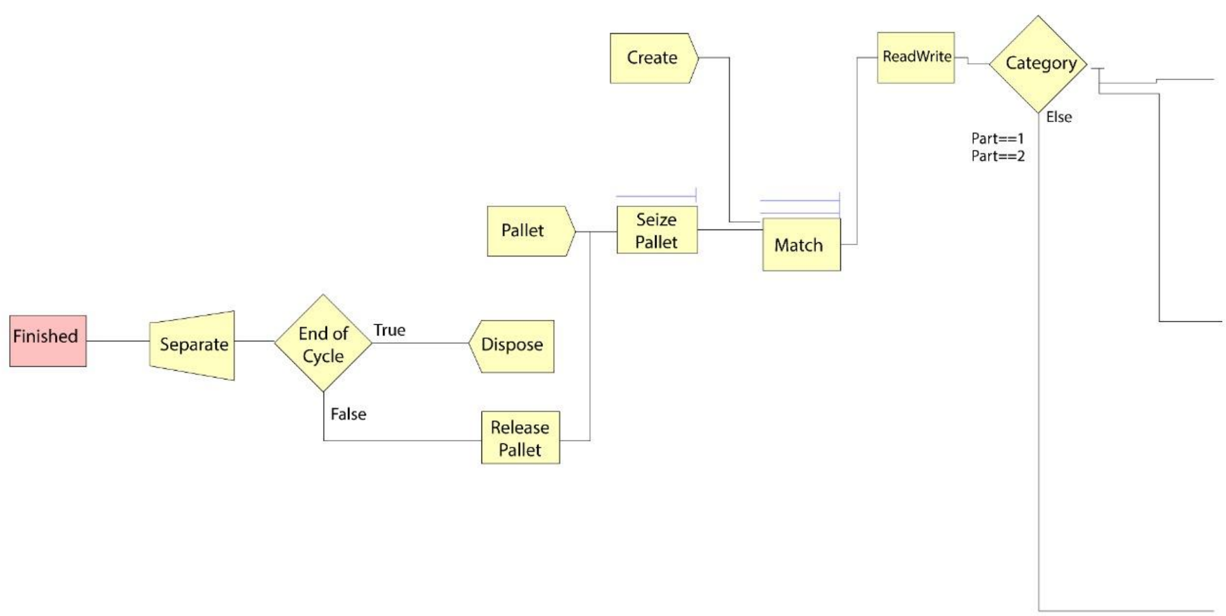


Figure 5.7.2.3: Detailed picture of section A

Output:

When the jobs completed all the cycles they reached at Finished Station. Then a Separate module is used to split existing Batch retaining original entity values by pallet and part. Another Decide module is used to identify the entity type. If the entity type is part, it will use the dispose module to complete the process and when the entity type is Pallet it will go to a release module where the resource is the same as previously named PalletCar. The released Pallet car will go to the Seize module again to create a batch with another new part.

Section B Assignment for Stations and Respective Routes

After deciding which part started travelling the model there are separate routes for each job to follow their respective paths. For easy jobs, there are 3 assigned modules and 2 stations which guide them to follow the appropriate path to complete the jobs. For medium and complex jobs, they required 4 assigned modules and 3 stations to reach their destinations.

The Assign module defines different attributes of processing time for each machine and the entity sequence which defines which sequence they will follow for their processing.

The Route modules connected all assigned stations to follow their sequence by respecting their destinations.

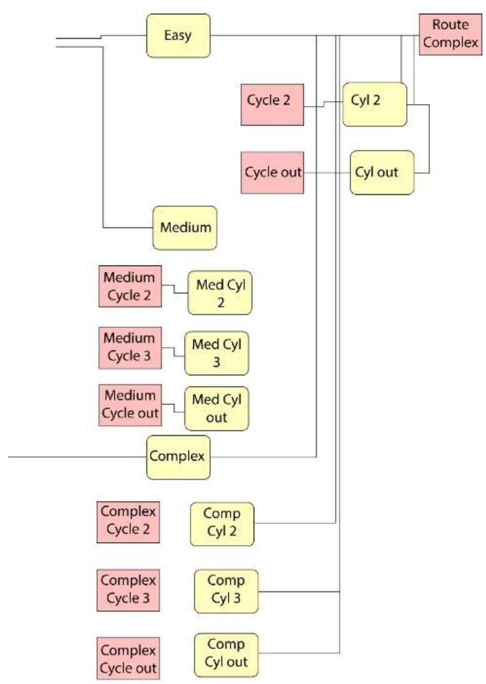


Figure 5.7.2.4: Detailed picture of Section B

Section C Stations and Route of the Model

The modeling of this section begins with a Station block referring to Station Front, followed by the Process block (Front Machine) and a Route to direct the product to the next station by sequence (Station Output). Then a Station block referring to Station Output, followed by the Process block (Output Machine) and a Route to direct the product to the next station by sequence (Station Press). Next a Station block referring to Station Press, followed by the Process block (Press Machine) and a Route to direct the product to the next station by sequence (Station Heat). After pressing the Complex job, the same way is followed by a Station block referring to Station Heat, which is also followed by the Process block (Heat Machine) and a Route to direct the product to the next station by sequence (Station Turning). The Medium and Complex job followed another Station block referring to Station turning, followed by the Process block (Turning Machine) and a Route to direct the product to the next station by sequence (Station Back). After that a Station block referring to Station Back, followed by the Process block (Back Machine) and a Route to direct the product to the next station by sequence (Station Measure). Another Station block referring to Station Measure, followed by the Process block (Measure Machine) and a Route to direct the product to the next station by sequence (Station Drill). After quality assurance of complex parts, a Station block referring to Station Drill, followed by the Process block (Drill Machine) and a Route to direct the product to the next station by sequence (Station Front). This cycle continues unless the predefined sequence elements refer them to a Station named Finished. For Processing action Seize Delay Release is chosen for all the machines with the resource of the name of that respective Machine.

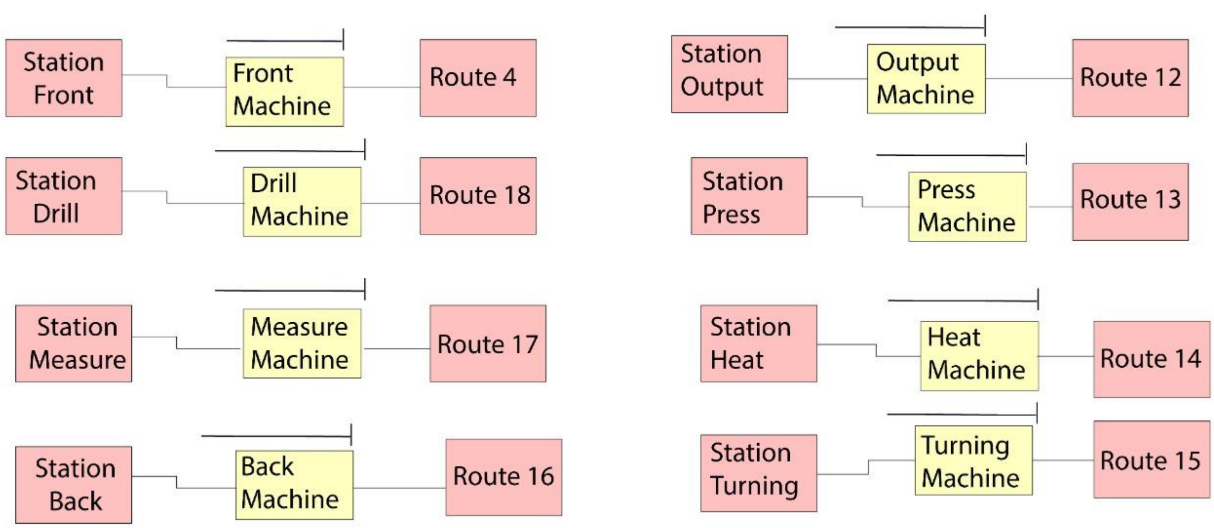


Figure 5.7.2.5: Detailed picture of section C

In this chapter details of the proposed framework, the model that have been built as well as the framework validation have been described. A list and thorough descriptions of the tools used have been presented in conjunction with the reason for these choices. It has been shown how the WVU Smart Manufacturing Lab has been useful for validating the framework. There has been a comprehensive description of the system present in the lab with all the parts involved to process a defined production plan. As well as, it has been showed that launching the production plan on the system requires a certain makespan for its completion. For each step of the framework, it has been accurately and deeply detailed the models built. First it has been made clear how the preferred data format is MS Excel as it allows to obtain only the relevant data. For the analysis step, it has been displayed the built simulation model that is the computer model of the real system present in the WVU Lab, capable of reading the obtained optimized sequence. In addition, it has been presented how the simulation analysis has been performed to compare the makespan of the selected production plan. Finally, in order to prove the flexibility and adaptability of the framework, it has been shown how the data can be converted into another format and analyzed by a different analytical tool. It has been demonstrated how this output can be read by an open-source python-based web application such as Jupyter to further visualize and analyze the related data.

CHAPTER 6

Result Analysis and Discussion

The study that has been carried out has underlined the need for developing a solution that allows sequencing effectively and efficiently. In doing so, it was necessary to consider that machine data are often stored and/or available in spreadsheet-based documents and that during their flow. Therefore, a framework was developed that enables the data flow from an MES system to an analytical software (Arena simulation), by having them in MS Excel format. Furthermore, this framework has been conceived so that tasks could be performed in an automatic way in order to avoid possible errors and reduce time for repetitive actions. To prove the effectiveness of the framework and validate it, it was evaluated in some specific user case. The aim was to improve the throughput time of a production plan by finding a relative optimum sequence to launch in production so that the makespan could be improved by comparing two scenarios of both algorithms of the defined production plan. In this way it was possible to prove that from a production process, MS Excel format data could be retrieved and make them flow, with appropriate processing, to Arena in order to perform a simulation analysis for optimization purposes. In addition, it was also possible to prove the effectiveness of the analysis by testing the new sequence on the real system and monitoring the new makespan in order to show its improvement compared to the other situation.

For different algorithms, the final optimal sequence should be different. Comparing the NEH and CDS algorithm, the makespan time is compared to analysis for better performance. Considering the system as a flow shop and job shop will give different outputs. Also considering job shop with the same ratio for all three categories and a different ratio for different categories will provide varieties of output. The figure 6.1 is visually representing the process of comparing the result section.

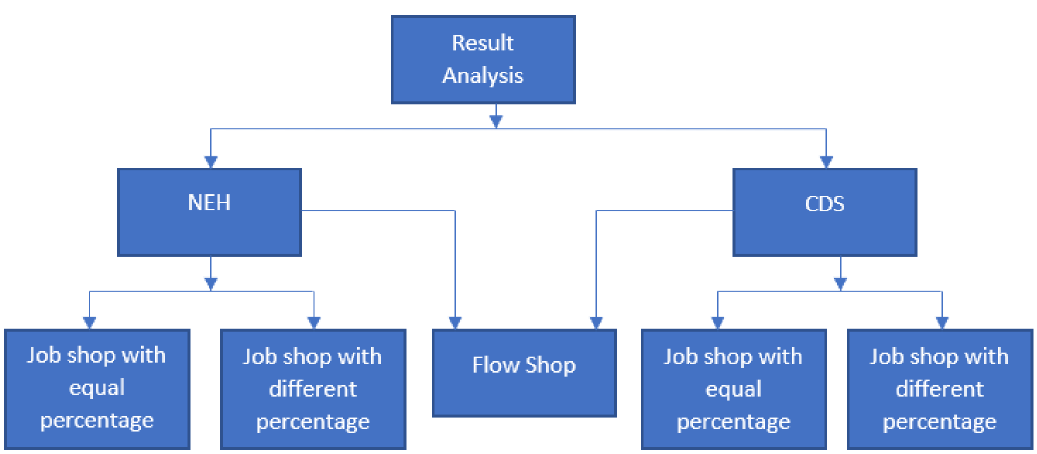


Figure 6.1: Process of comparing result analysis

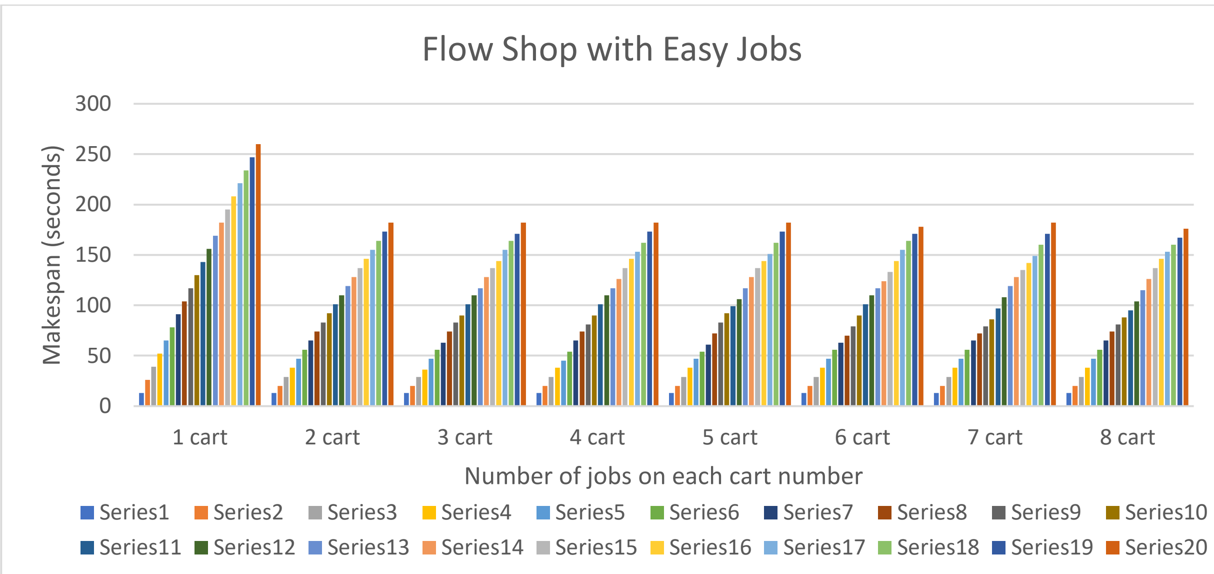
6.1 Flow Shop

In flow shop sequencing there is a strict order of all operations to be performed on all jobs. For this case, for three categories of jobs, three different types of flow shops are considered. For both algorithms, the sequence obtained is different, but the makespan is equal for all cases.

6.1.1 Job Category: Easy

Easy job processed through four machines. The processing times are lower compared to other category. It needs three cycles to process the whole job.

Graph 6.1.1. 1: Makespan time for different number of easy category jobs in flow shop



Focusing on the graph 6.1.1.1, it is clear that if the number of jobs increases, total makespan will also increase for each pallet number. This graph and from the data table also shows that for most of the cases, with the increase in pallet number results decrease in makespan. For jobs 1 through 8 can travel once to complete the jobs. But as there is a limitation of pallet number, the maximum pallet available in the lab scenario is 8. So, when the job number is more than 8, considering 9 to 16, they have to wait for completing one single job, then the new part for the next job can enter that sequence. Similar scenario happened for greater than multiple of number 8.

The table 6.1.1.2 focuses that utilizing a higher number of pallets will give the minimum output. For easy category jobs, the minimum unit makespan obtained for 11 jobs using all available pallets.

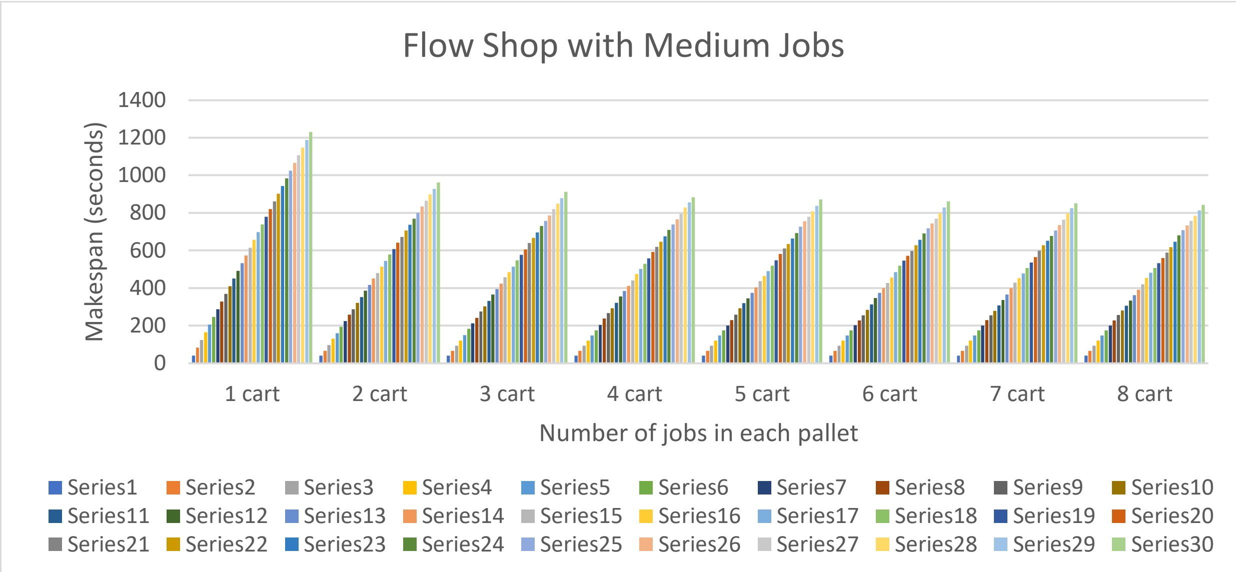
Table 6.1.1. 2 Per unit makespan time for different number of pallets for easy category jobs in flow shop

Per unit makespan time for different number of pallets								
Number of jobs	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	13.00	13.00	13.00	13.00	13.00	13.00	13.00	13.00
2	13.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
3	13.00	9.67	9.67	9.67	9.67	9.67	9.67	9.67
4	13.00	9.50	9.00	9.50	9.50	9.50	9.50	9.50
5	13.00	9.40	9.40	9.00	9.40	9.40	9.40	9.40
6	13.00	9.33	9.33	9.00	9.00	9.33	9.33	9.33
7	13.00	9.29	9.00	9.29	8.71	9.00	9.29	9.29
8	13.00	9.25	9.25	9.25	9.00	8.75	9.00	9.25
9	13.00	9.22	9.22	9.00	9.22	8.78	8.78	9.00
10	13.00	9.20	9.00	9.00	9.20	9.00	8.60	8.80
11	13.00	9.18	9.18	9.18	9.00	9.18	8.82	8.64
12	13.00	9.17	9.17	9.17	8.83	9.17	9.00	8.67
13	13.00	9.15	9.00	9.00	9.00	9.00	9.15	8.85
14	13.00	9.14	9.14	9.00	9.14	8.86	9.14	9.00
15	13.00	9.13	9.13	9.13	9.13	8.87	9.00	9.13

6.1.2 Job Category: Medium

Medium category jobs need 6 machines to complete the whole job. The longest processing time for that case needs in a muscle press to join the front and back part of the magazine. Total processing time will depend on the total cycle length which is actually 4 cycles for this category of jobs.

Graph 6.1.2. 1: Makespan time for different number of medium category jobs in flow shop



The graph 6.1.2.1 focuses that for each pallet, by increasing the number of jobs, the total makespan will always increase similarly as previous. But if the number of pallets increases, the makespan starts decreasing. Following the same trend as easy jobs, they have to complete one full job to receive the next job which is higher than the available pallet number.

Table 6.1.2.2: Per unit makespan time for different number of pallets for medium category jobs in flow shop

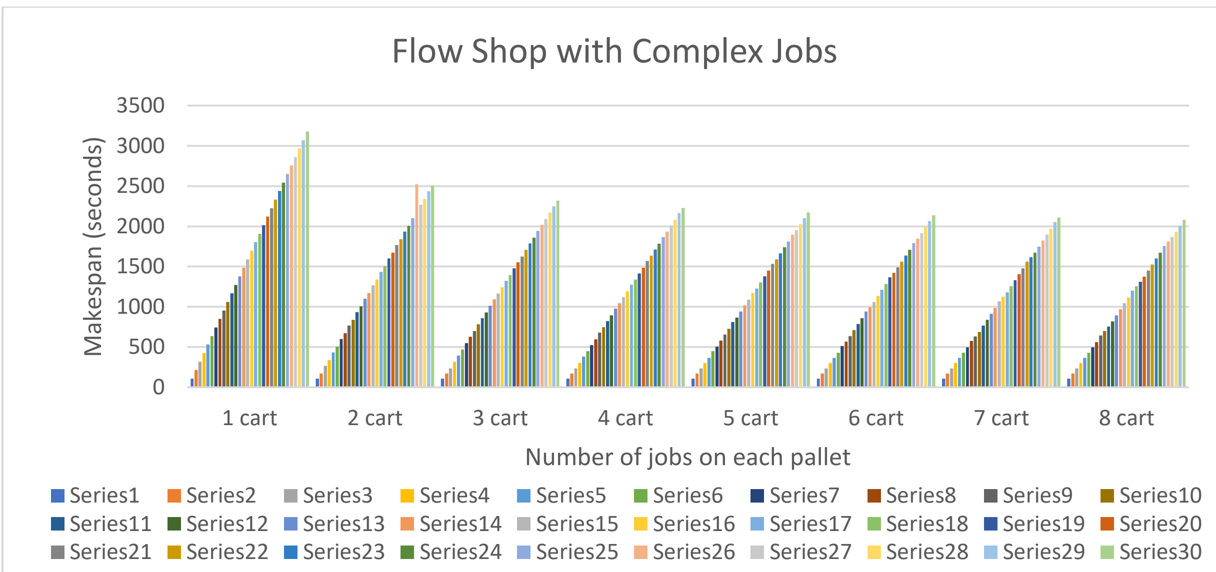
Per unit makespan time for different number of pallets								
Job Number	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	41.00	41.00	41.00	41.00	41.00	41.00	41.00	41.00
2	41.00	33.00	33.00	33.00	33.00	33.00	33.00	33.00
3	41.00	32.00	31.00	31.00	31.00	31.00	31.00	31.00
4	41.00	32.50	30.25	30.00	30.00	30.00	30.00	30.00
5	41.00	32.00	30.00	29.60	29.40	29.40	29.40	29.40
6	41.00	32.33	30.67	29.17	29.17	29.00	29.00	29.00
7	41.00	32.00	30.29	29.14	28.57	28.86	28.71	28.71
8	41.00	32.25	30.13	29.75	28.63	28.38	28.63	28.50
9	41.00	32.00	30.56	29.56	28.67	28.22	28.22	28.44
10	41.00	32.20	30.30	29.30	29.20	28.30	27.90	28.10
11	41.00	32.00	30.18	29.27	29.09	28.36	28.00	27.82
12	41.00	32.17	30.50	29.67	28.75	28.83	28.08	27.75
13	41.00	32.00	30.31	29.54	28.77	28.77	28.15	27.85
14	41.00	32.14	30.21	29.36	28.79	28.50	28.57	27.93
15	41.00	32.00	30.47	29.33	29.13	28.40	28.53	28.00

The table 6.1.2.2 focuses on the per unit makespan time for different numbers of jobs. As this is a flow shop category and all jobs are following the same processes, it is easier to calculate the per unit makespan time for comparison. Utilizing all available pallets will also give the minimum makespan for this case which is 12th number of jobs.

6.1.3 Job Category: Complex

To complete a full complex job, it will go through all the available machines. Though it's a flow shop, but due to an unusual machine setup, it needs 4 cycles to complete a full job. For better job quality, it ensures heat treatment after muscle press which actually creates a huge difference in processing time rather than other categories of jobs.

Graph 6.1.3. 1: Makespan time for different number of complex category jobs in flow shop



The graph 6.1.3.1 indicates increasing the number of jobs will increase makespan for each pallet. But the increase of pallet number creates a declining trend for makespan for most of the cases.

Table 6.1.3.2: Per unit makespan time for different number of pallets for complex category jobs in flow shop

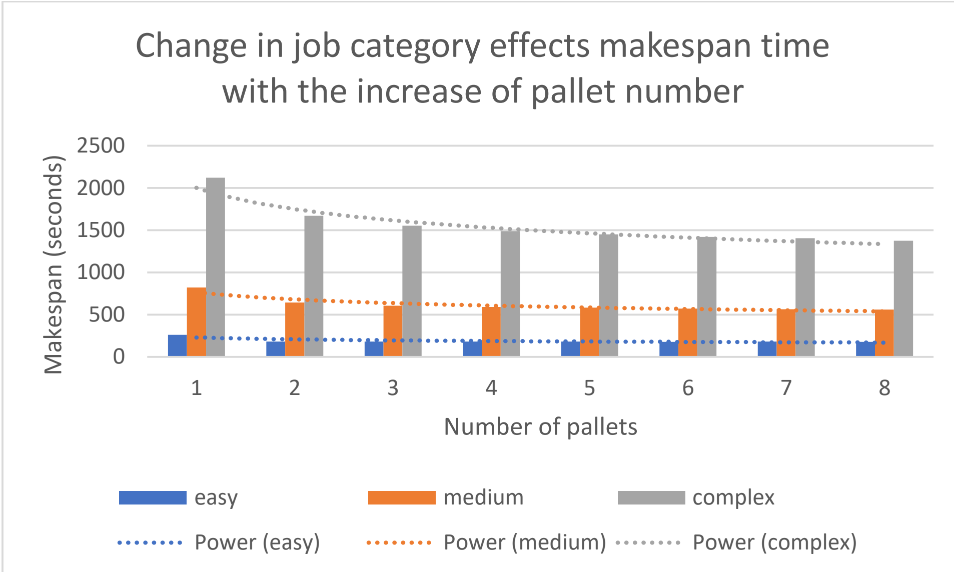
Per unit makespan time for different number of pallets								
Number of jobs	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	106.00	106.00	106.00	106.00	106.00	106.00	106.00	106.00
2	106.00	84.50	84.50	84.50	84.50	84.50	84.50	84.50
3	106.00	88.00	78.00	78.00	78.00	78.00	78.00	78.00
4	106.00	84.00	79.25	74.75	74.75	74.75	74.75	74.75
5	106.00	86.20	78.80	76.40	72.80	72.80	72.80	72.80
6	106.00	83.83	77.67	74.83	74.50	71.50	71.50	71.50
7	106.00	85.43	78.43	74.86	71.71	73.14	70.57	70.57
8	106.00	83.75	78.25	74.50	72.38	70.88	72.13	69.88
9	106.00	85.00	77.56	75.44	72.67	70.44	70.22	71.33
10	106.00	83.70	78.10	74.60	72.60	70.90	68.70	69.70
11	106.00	84.73	78.00	74.64	73.55	71.27	69.45	68.36
12	106.00	83.67	77.50	74.42	72.00	71.33	69.92	68.25
13	106.00	84.54	77.92	75.08	72.38	72.23	70.31	68.77
14	106.00	83.64	77.86	74.50	72.57	71.00	70.43	69.21
15	106.00	84.40	77.47	74.53	72.53	70.73	71.27	69.60

This table 6.1.3.2 represents the per unit makespan time for complex category jobs. Increasing pallet numbers causes downward slope for each job. For complex category job per unit makespan obtained minimum for utilizing higher number of pallets which is 12 jobs again.

6.1.4 Comparison

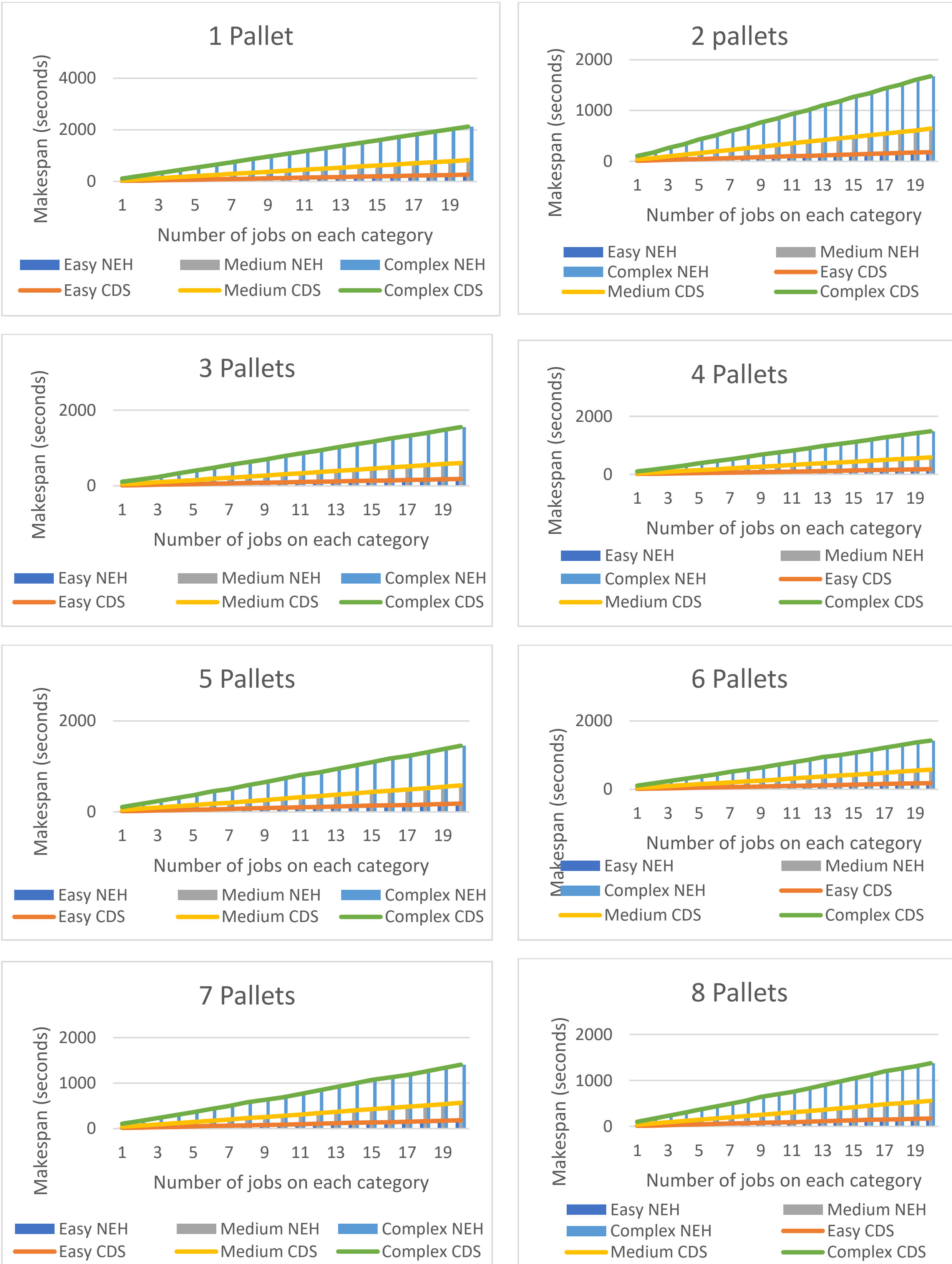
To summarize the above discussion, the graph is presented to explain that for a number of jobs in flow shops, if the pallet number increases, total makespan will follow a downward trend. But for complex category products, the trend is steeper because of its long processing time in the heating tunnel. Medium category products are also showing a declining trend which is not as steep as complex ones because there is no heating tunnel used for this category of products but the stepper trend shows the effect of muscle press on the processing times. Whereas, easy jobs are showing an almost negligible declining trend as the machines used for processing easy jobs have the lowest processing time compared with other two categories. Graph 6.1.4.1 represents the comparison among different categories of jobs and the significant variation based on available pallet numbers.

Graph 6.1.4. 1: Total makespan time decreased with number of pallets for flow shop

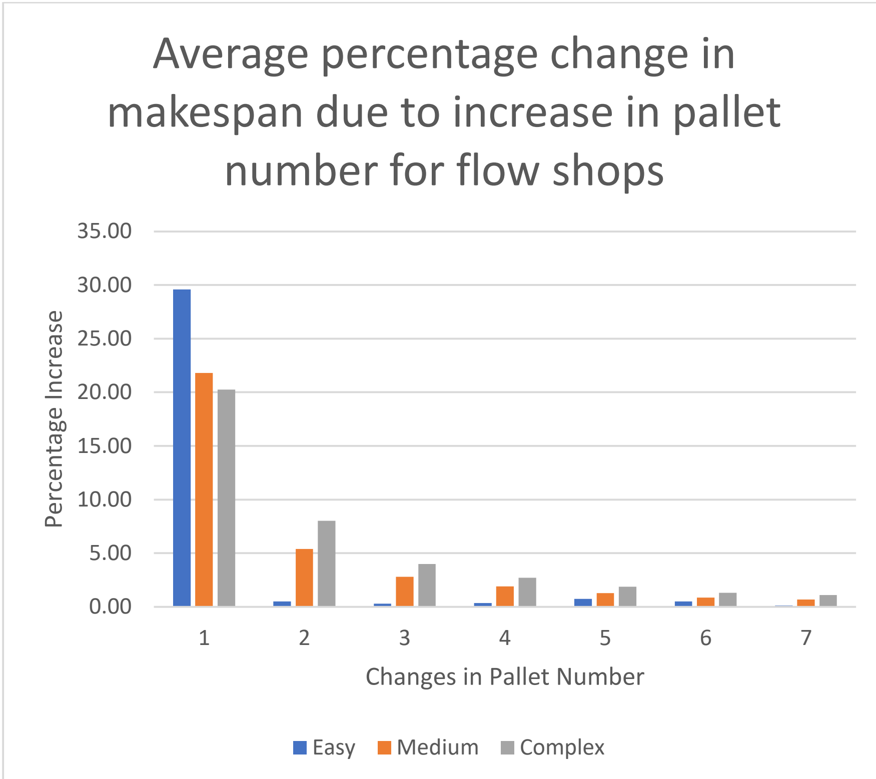


Graph 6.1.4.2 compares the makespan time for all category jobs in two different algorithms. Though the sequence obtained from those algorithms are not identical but as the product category is same, the different sequence cannot affect the makespan time for this scenario.

Graph 6.1.4. 2: Makespan time for different algorithms for different category of flow shop



Graph 6.1.4. 3: Average percentage change in makespan with increasing pallet number for flow shops



Graph 6.1.4.3 visualizes changes the average percentage increase for pallet 1 to 2 is significantly higher than any pallet. For easy category jobs, the other changes are quite negligible. For medium and complex category jobs, the average changes due to increasing numbers from pallet 1 to 2 is pretty close. Medium jobs average percentage change increasing from pallet 2 to 3 and 3 to 4 are quite significant. Same situation happens for complex category jobs increasing from pallets 2 to 3, 3 to 4 and 4 to 5. Though the complex jobs show a higher percentage change than medium jobs.

6.2 Job Shop

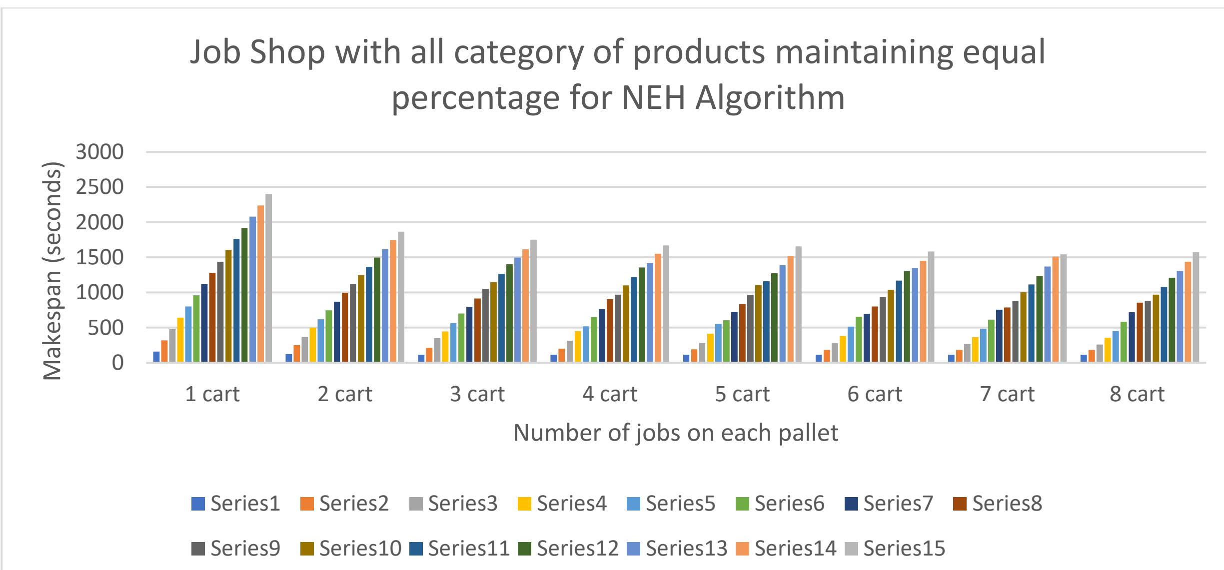
Job shops are typically considered as small manufacturing systems that handle custom job production for different batches. Job shops typically move on to different jobs when each job is completed. Job shops may contain different machines which are aggregated in shops by the nature of skills and technological process involved which gives the production system processing flexibility.

Normally in a job shop, there are a number of identical machines which allow to process jobs based on availability if there is no strict processing order. In the lab scenario, the additional constraint is there is a single machine for each category and no process can be done skipping the prerequisite process order.

6.2.1 Job Category: Mixed Category but Equal Percentage

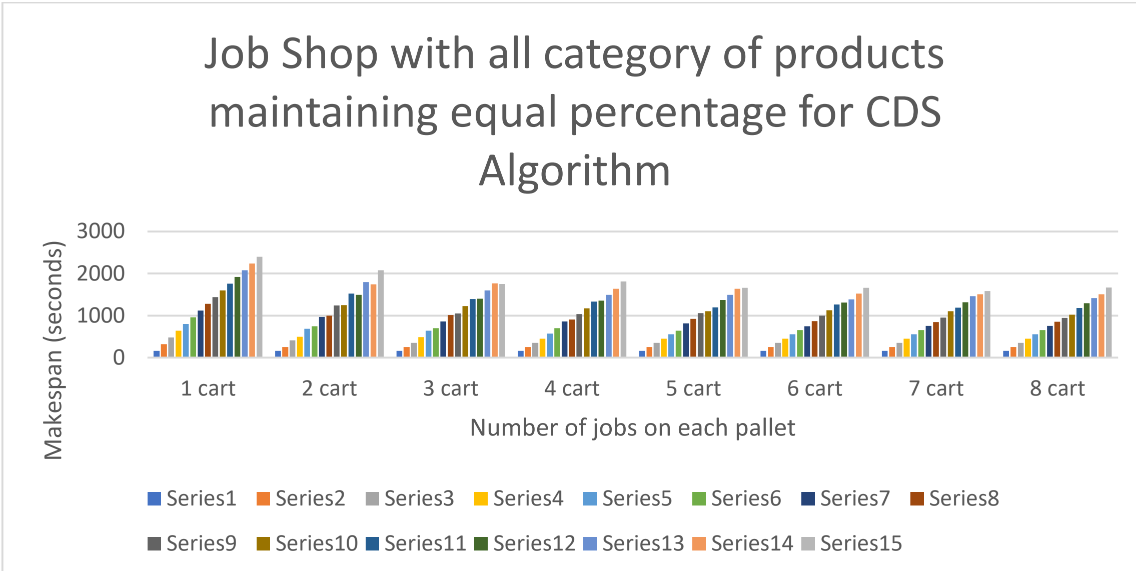
Here, all three category jobs are involved. The unique setup for this scenario is that each category of jobs should maintain 33.33% for total sequence which is actually 1/3 of total job numbers. So, they are mainly multiple of 3 jobs with different sequences from different algorithms.

Graph 6.2.1. 1: Makespan time for different number of all category jobs in job shop following NEH Algorithm



For both NEH and CDS algorithms, the sequence obtained is showing the same behavior as flow shop. When the number of jobs increases, the total makespan is increasing. But due to the increase of pallet number, creating a declining trend for most of the cases. Graph 6.2.1.1 and graph 6.2.1.2 is showing the visual representation of that scenario for both algorithms. Utilizing a higher number of pallets will give the minimum makespan for both scenarios.

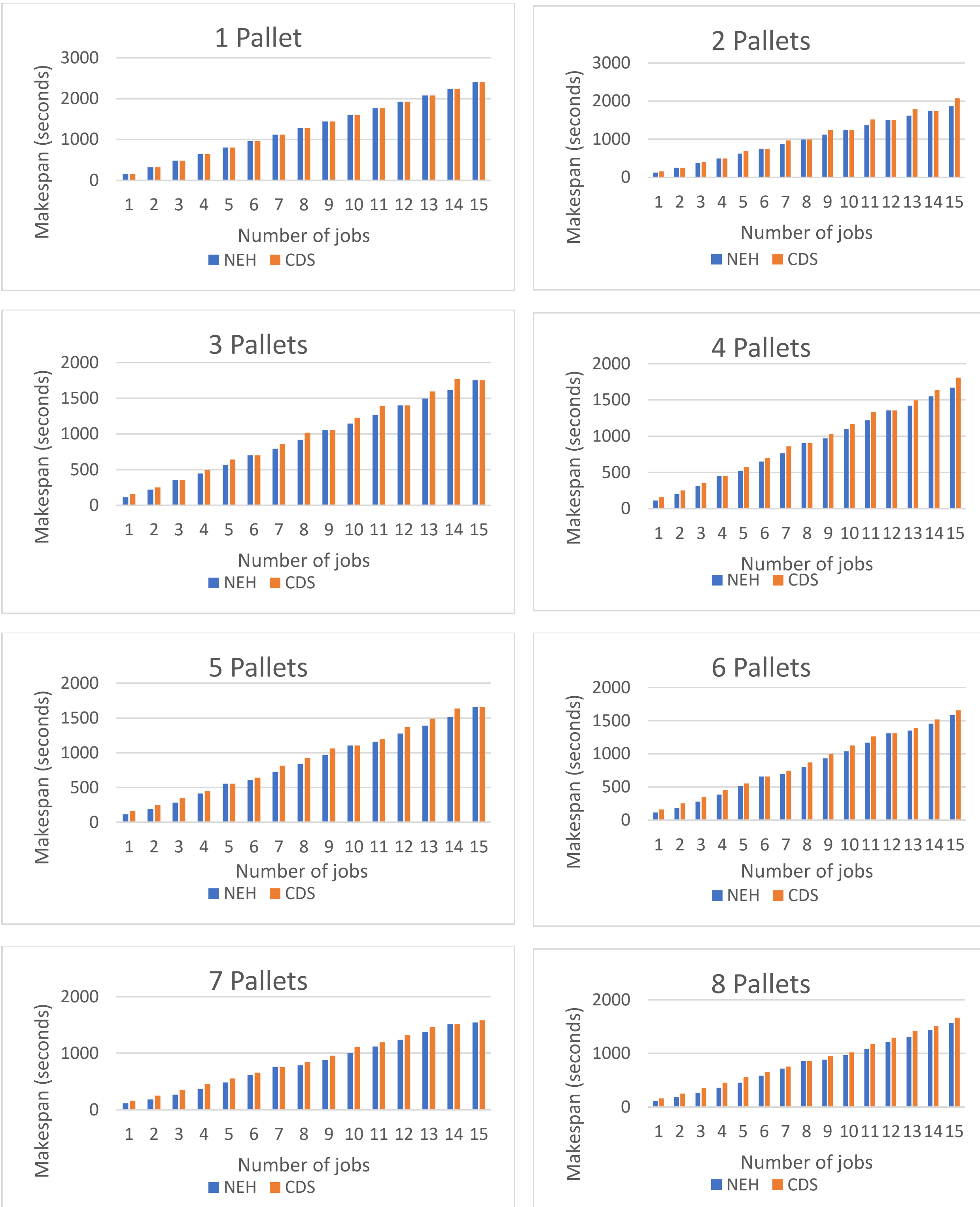
Graph 6.2.1. 2: Makespan time for different number of all category jobs in job shop following CDS Algorithm



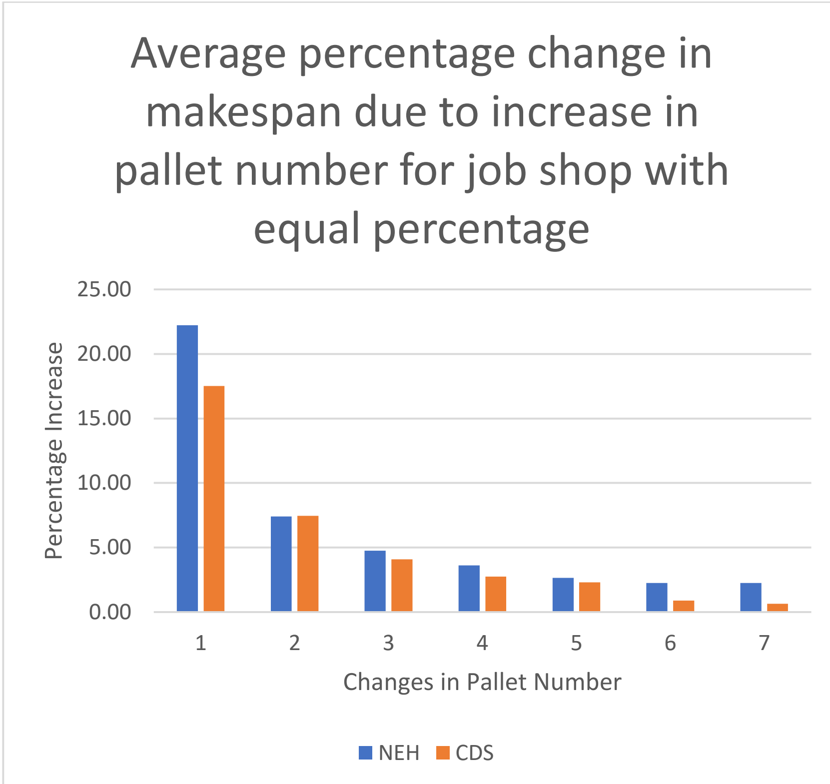
A mixed category with equal percentage of jobs also behaves the same as a flow shop for a single pallet. Though the sequence is different for different algorithms, the pallets constraint creates the scenario where the calculated makespan is equal for both cases. Focusing on the graph 6.2.2.1 it is clear that if the number of jobs increases, total makespan will also increase for each pallet number. But comparing both algorithms, the makespan obtained from the NEH algorithm is always lower than the CDS one.

Comparison

Graph 6.2.2. 1: Makespan time for different algorithms for mixed category with equal percentage of job shop



Graph 6.2.2. 2: Average percentage change in makespan with increasing pallet number for CDS algorithm with equal percentage of job shop

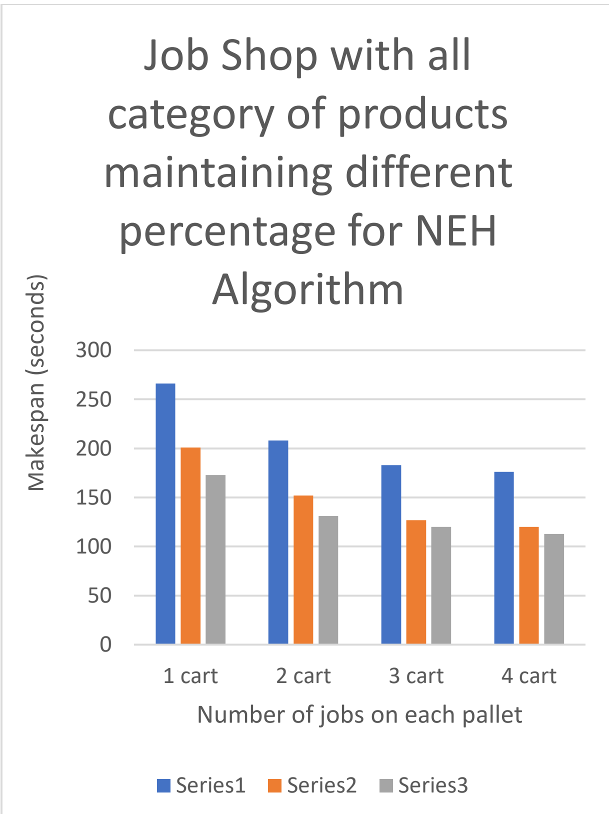


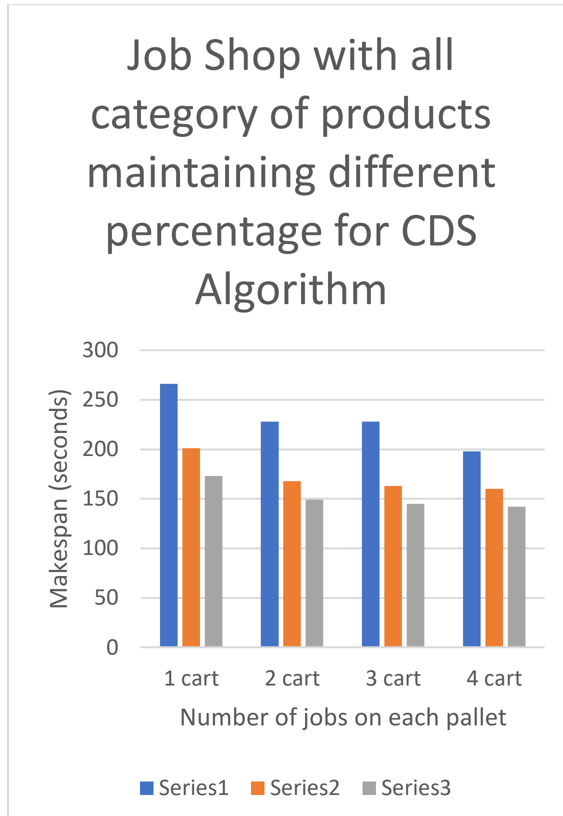
Graph 6.2.2.2 focuses on the comparison between two algorithms for percentage increase due to changes in pallet number. For change in pallet 1 to 2, NEH shows the highest increase. For change in pallet 2 to 3, they are almost the same for both algorithms. Lastly, the CDS algorithm shows almost negligible changes for pallet 6 to 7 and 7 to 8.

6.2.2 Job category: Mixed category with different percentage

For this scenario, the number of jobs is fixed but there can be different combinations possible based on job category. The minimum number of jobs for this category is 4. As there are three categories of job, it's not possible to combine less than 4 jobs.

Graph 6.2.2.1: Makespan time for different number of combinations of 4 jobs in job shop

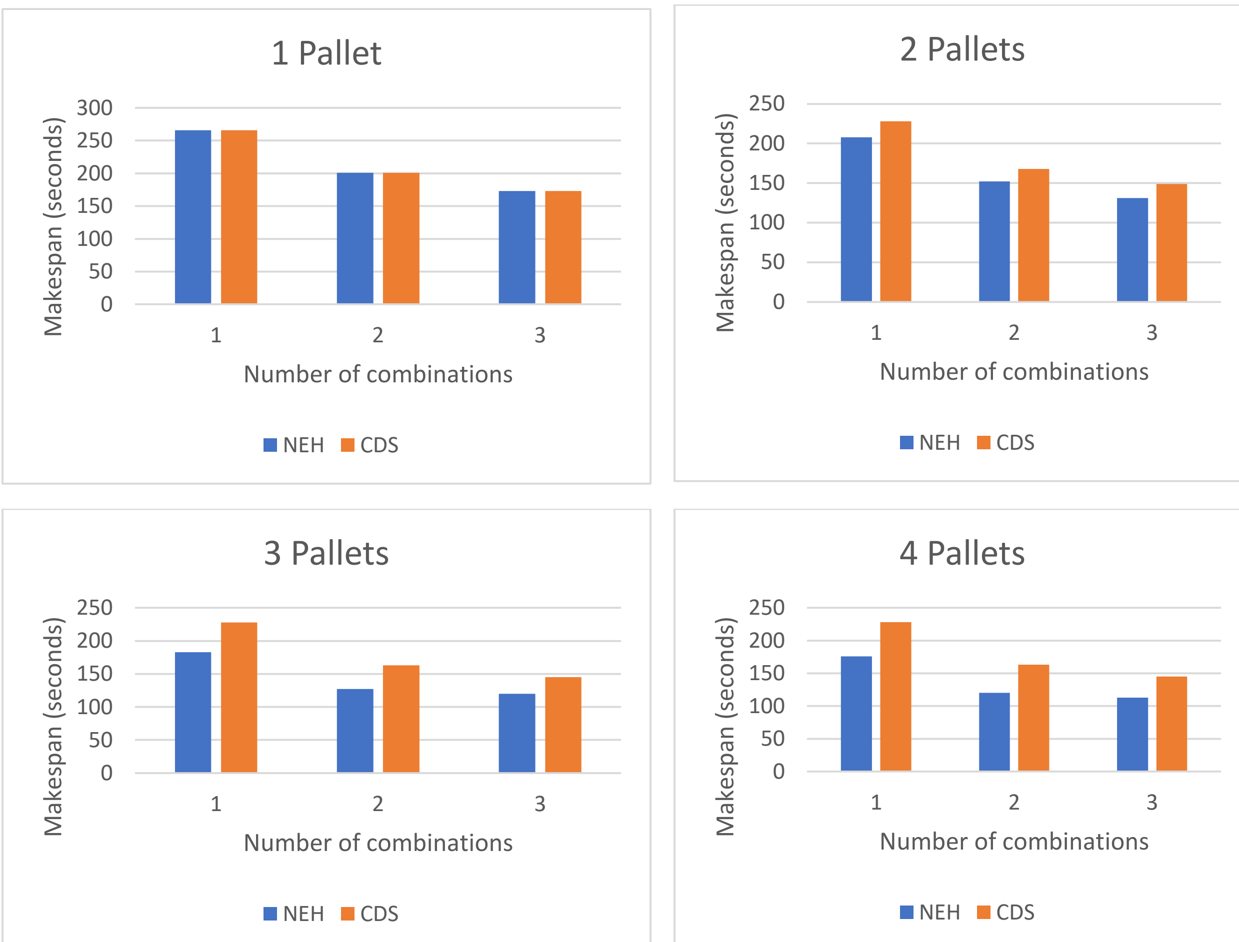




Graph 6.2.2.1 represents the scenario of 4 jobs with various number of combinations. If the number of complex jobs is higher than the easy and medium jobs, total makespan time will always be higher than all the cases. The minimum makespan found for the combination with a higher number of easy jobs than the other category.

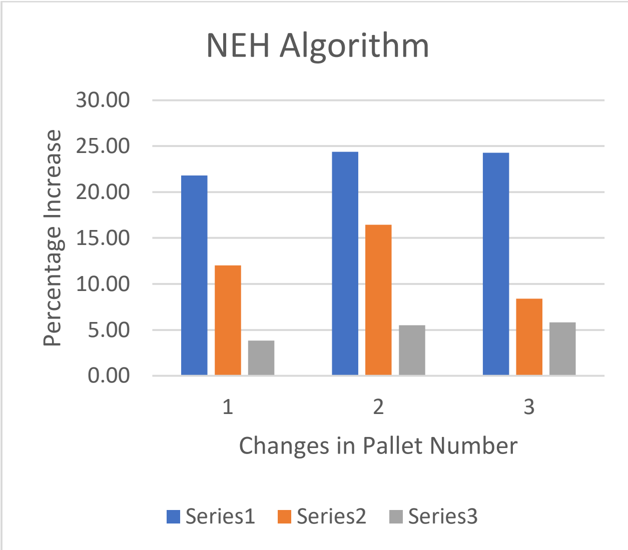
Comparison

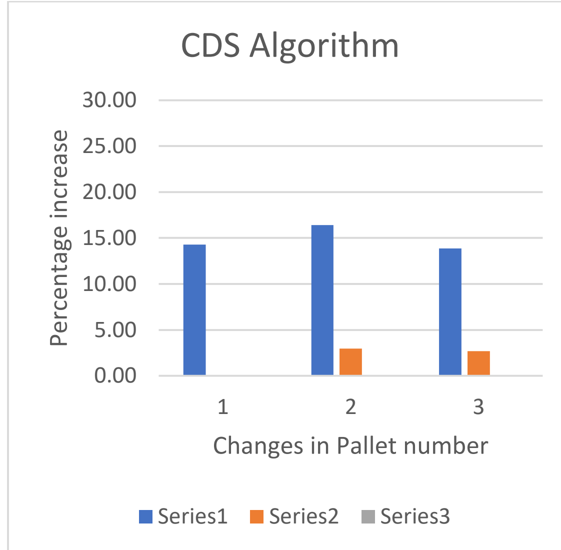
Graph 6.2.2.2: Makespan time for different algorithms for mixed category with different percentage of job shop for 4 jobs



Graph 6.2.2.2 presents the variation of makesapn using two different algorithms. For a single pallet, the scenario is as similar as the flow shop due to pallet constraint. Rather than that, all the scenarios prove that NEH performs better than CDS.

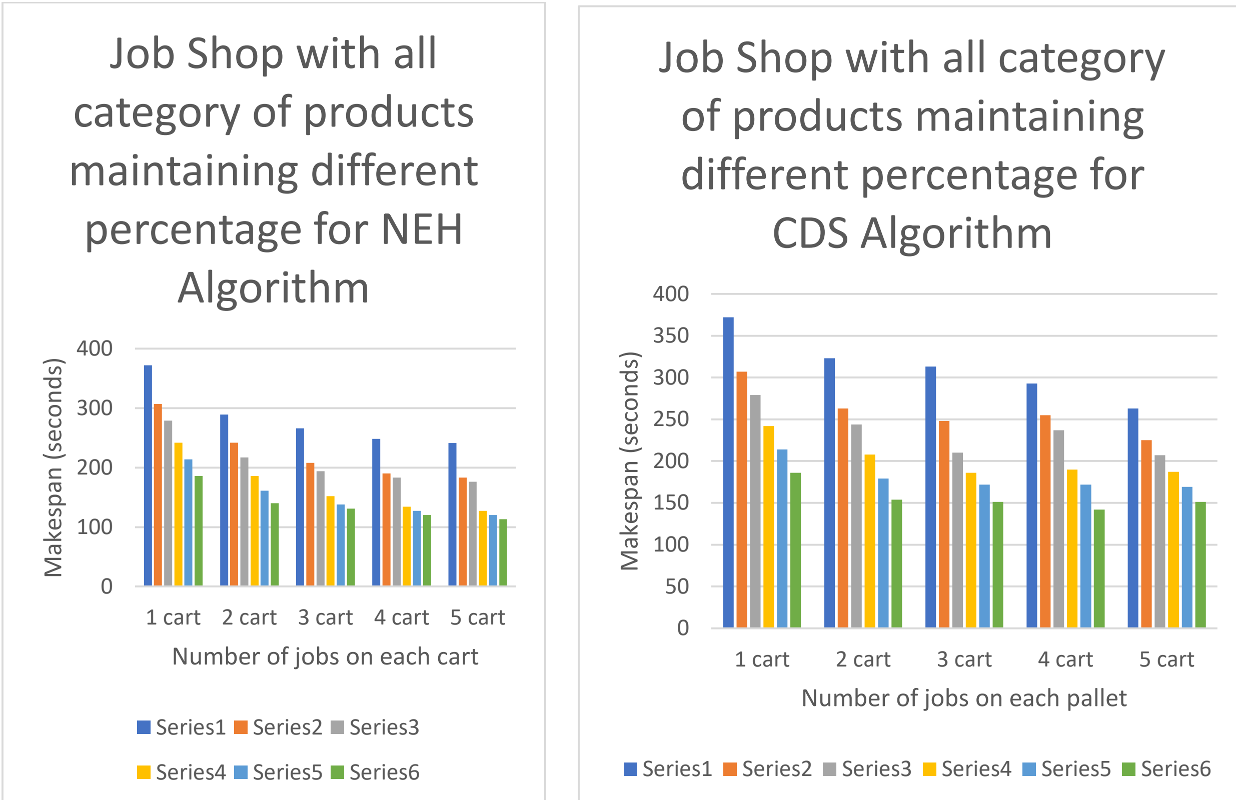
Graph 6.2.2.3: Percentage deviation in makespan with increasing pallet number for combination of 4 jobs





Graph 6.2.2.3 represents the scenario that increasing pallet number changes the makespan. Significant deviation shown for NEH algorithm. From pallet 1 to 2, the change is lower for maximum complex jobs. For maximum medium jobs, from pallet 2 to 3, the percentage increases but decreases again from pallet 3 to 4. Again, maximum percentage change for maximum easy category jobs found increasing pallet from 3 to 4.

Graph 6.2.2.4: Makespan time for different number of combinations of 5 jobs in job shop

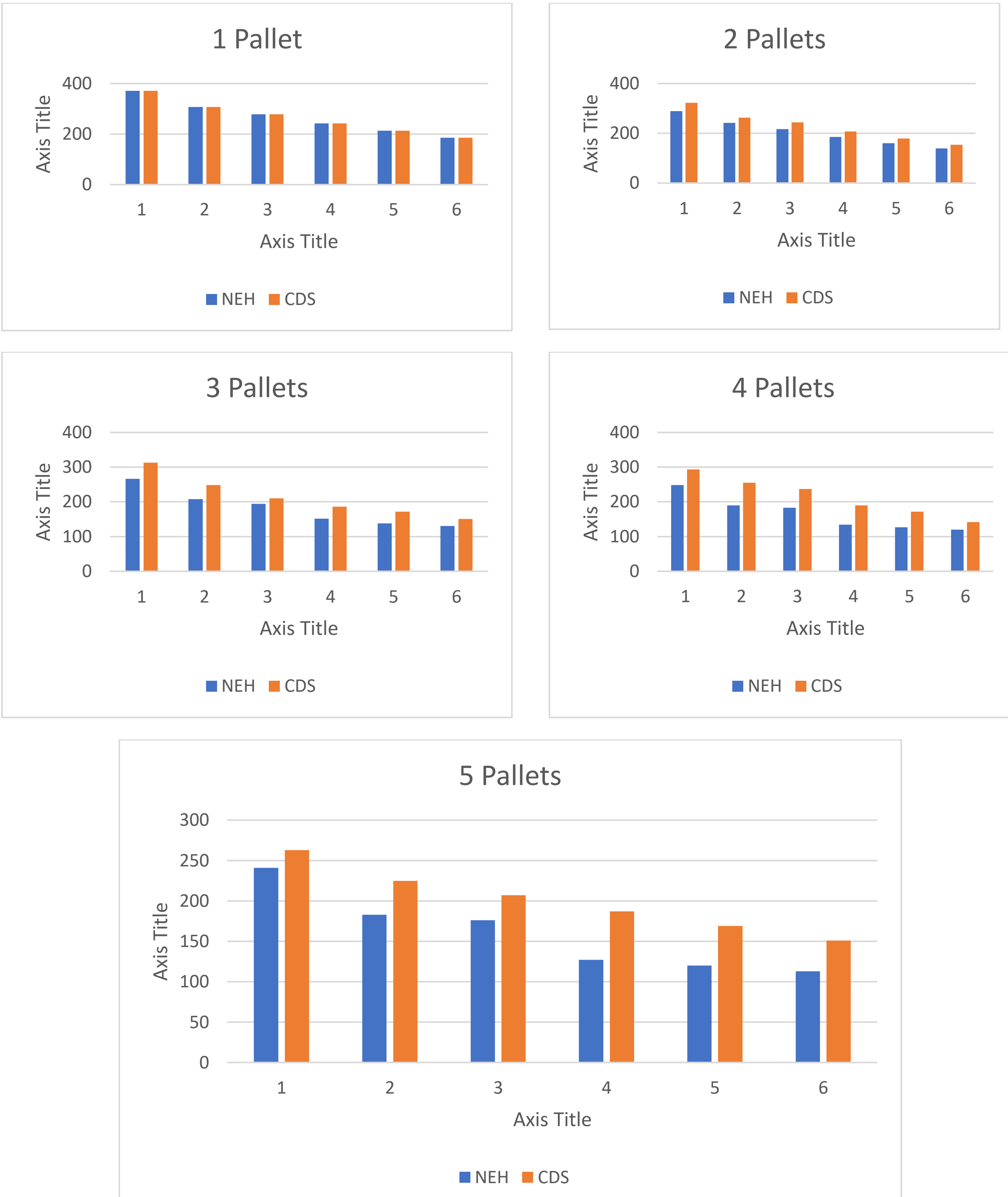


Graph 6.2.2.4 presents the case for 5 jobs with various numbers of combinations. The minimum makespan found for the combination with a higher number of easy jobs than the other category. If the number of complex jobs is higher than the easy and medium jobs, total makespan time will always be higher than all the cases. Sometimes there are some fluctuations due to the ratio of number of jobs in each category as the processing time varies for all three categories.

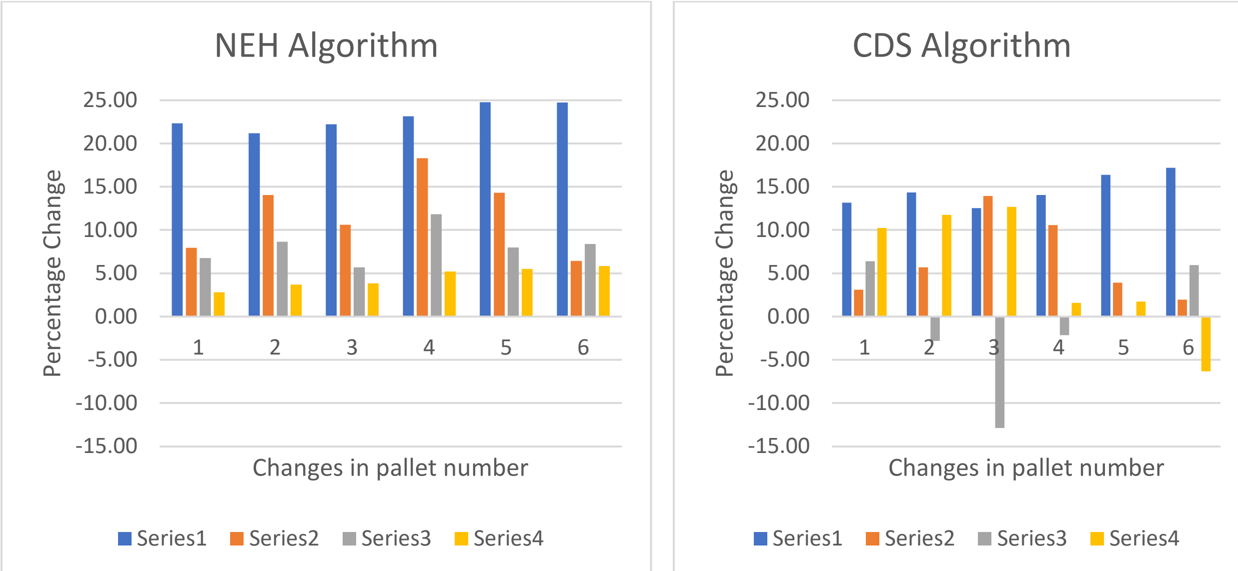
There is a comparison presented in graph 6.2.2.5. This focuses on the increase of makespan with the number of jobs. Compared with the CDS algorithm NEH always performs better except for single pallets. This different situation is the same for flow shops due to pallet constraints.

Comparison

Graph 6.2.2.5: Makespan time for different algorithms for mixed category with different percentage of job shop for 4 jobs

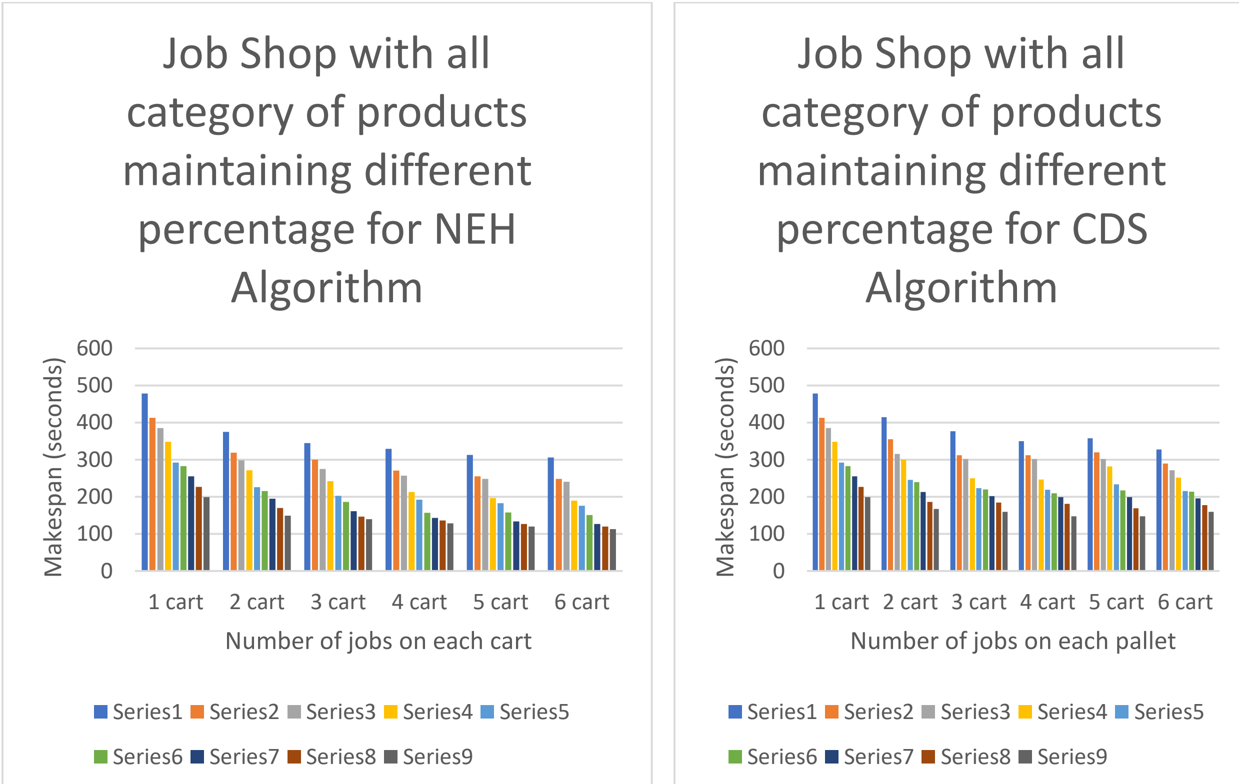


Graph 6.2.2.6: Percentage deviation in makespan with increasing pallet number for combination of 5 jobs



Similarly Graph 6.2.2.6 shows the highest percentage change for the NEH algorithm. For maximum complex category jobs, the percentage increase from pallet 1 to 2 is always higher than any other combination in the NEH algorithm. For equal complex and medium jobs, percentage change is highest, increasing pallet number from 2 to 3 for NEH algorithm but CDS algorithm shows the same for equal percentage of easy and complex jobs. Increasing the pallet number from 3 to 4 shows random behavior for both algorithms. For NEH algorithm, from pallet changes 4 to 5 shows the increasing trend. But for the CDS algorithm, it increases for some combinations and declines most for maximum percentage of easy jobs.

Graph 6.2.2.7: Makespan time for different number of combinations of 6 jobs in job shop

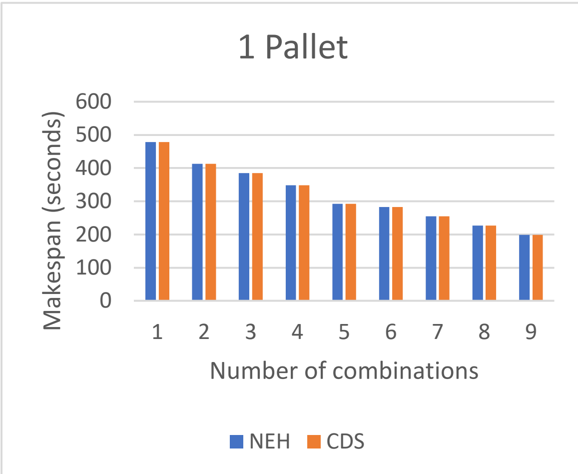


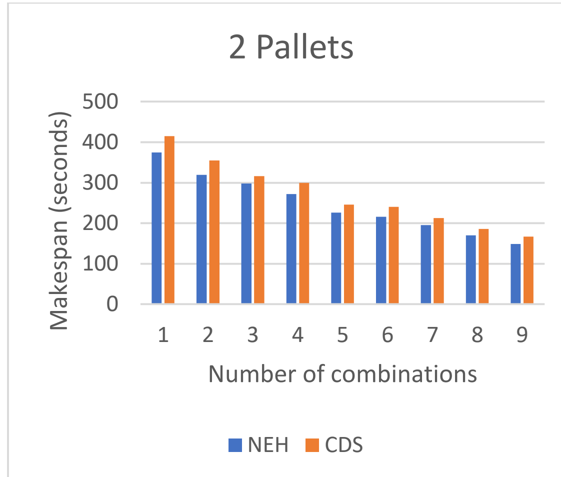
Graph 6.2.2.7 presents the 6 jobs with various combinations. Sometimes there are some fluctuations with the ratio of number of jobs in each category due to the variation of processing time for all categories. Normally, if the number of complex jobs is higher than the easy and medium jobs, total makespan time will always be higher than all the cases. On the contrary, the minimum makespan found for the combination with a higher number of easy jobs than the other category.

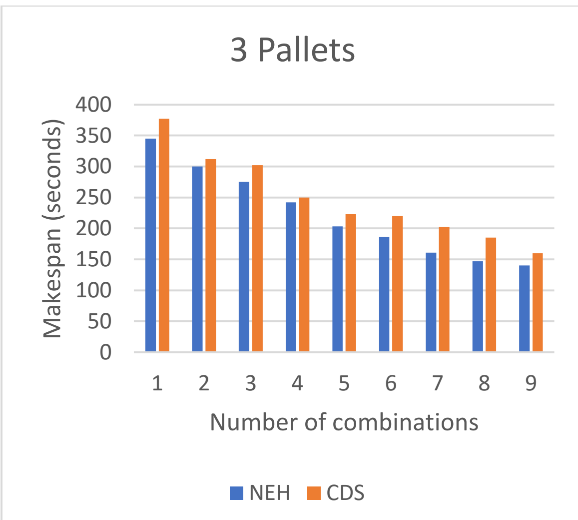
Graph 6.2.2.8 compares the makespan for two different algorithms. NEH always performs better than the CDS due to two different sequences. Single pallet scenario behaves as usual due to the pallet constraints.

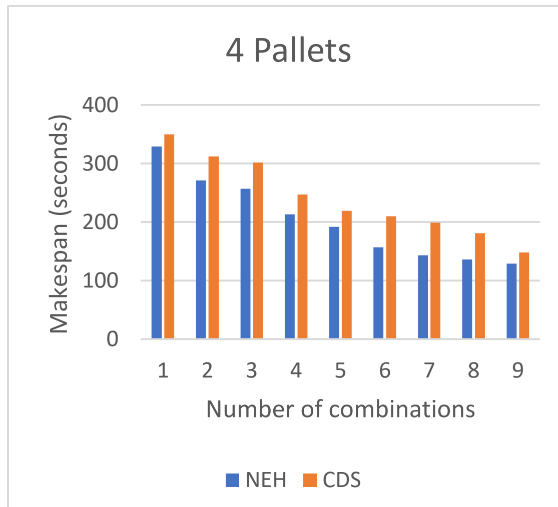
Comparison

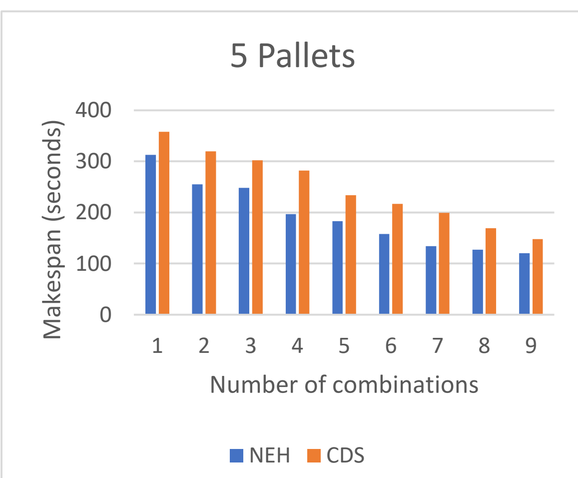
Graph 6.2.2.8: Makespan time for different algorithms for mixed category with different percentage of job shop for 6 jobs

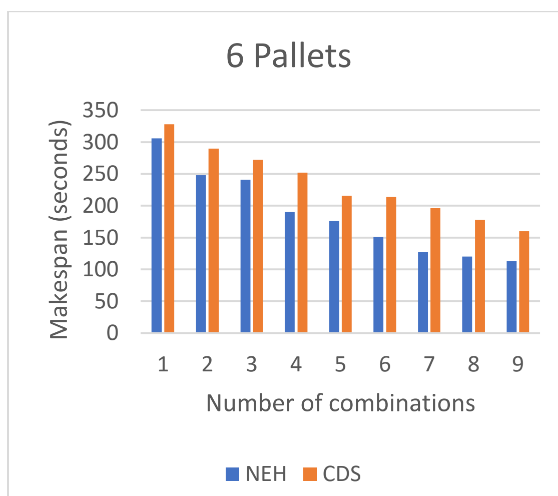




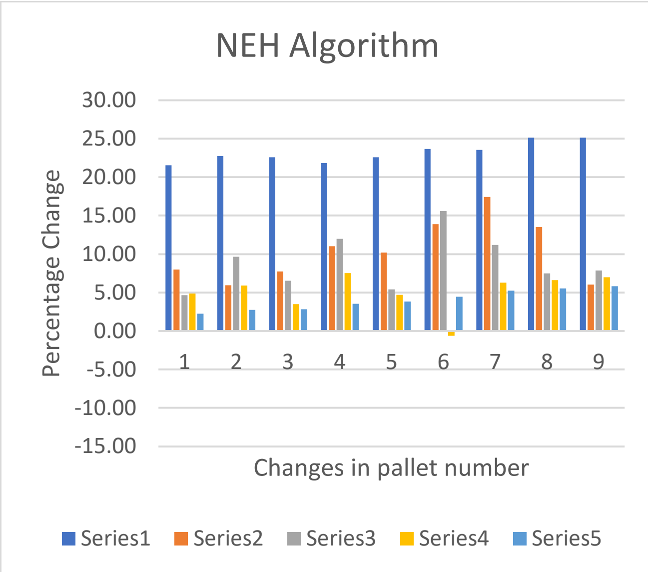


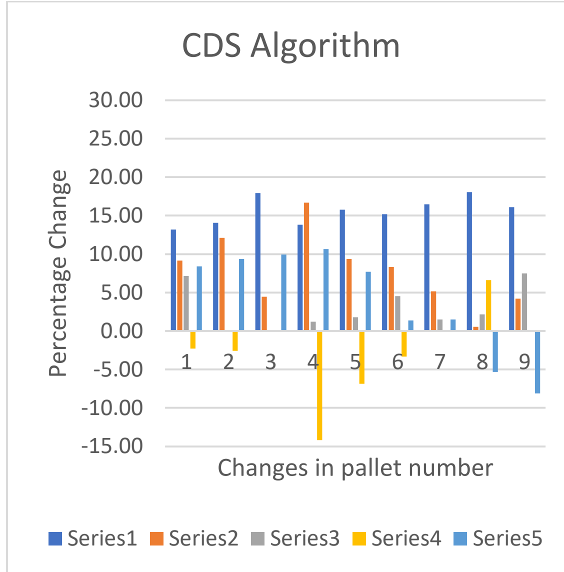






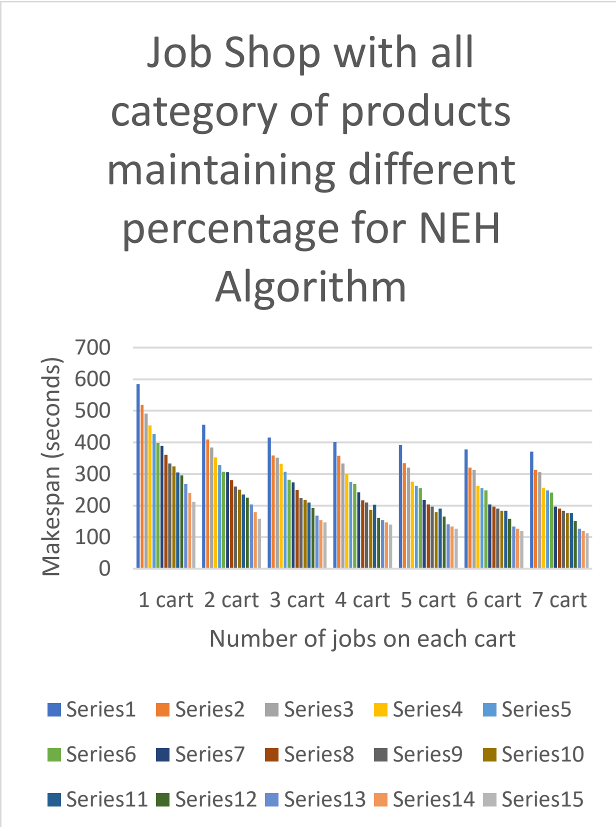
Graph 6.2.2.9: Percentage deviation in makespan with increasing pallet number for combination of 6 jobs

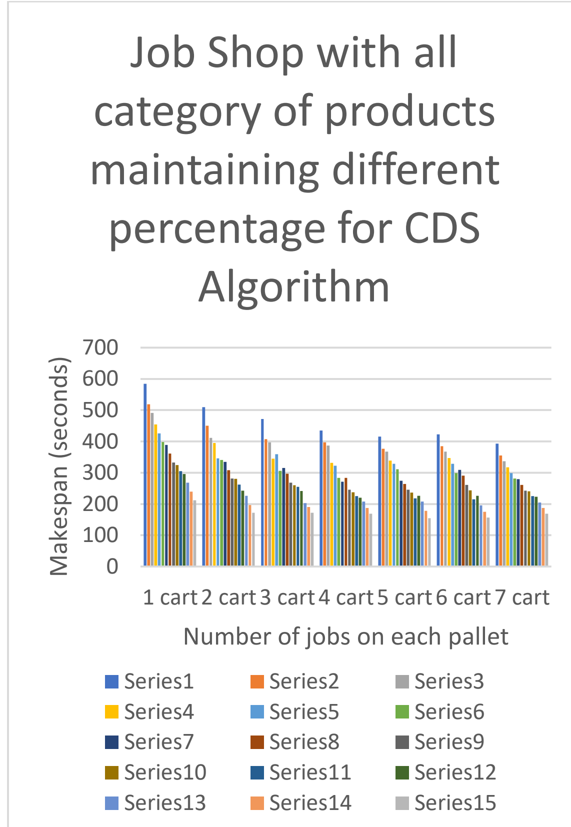




Graph 6.2.2.9 shows the same result for the NEH algorithm that, from changes in pallet 1 to 2 will always show an increasing trend. For almost all of the combinations, it is showing random changes. But for the combination with maximum easy category, showing percentage increase for NEH algorithm whereas CDS algorithm shows decrease for increasing pallet from 5 to 6.

Graph 6.2.2.10: Makespan time for different number of combinations of 7 jobs in job shop



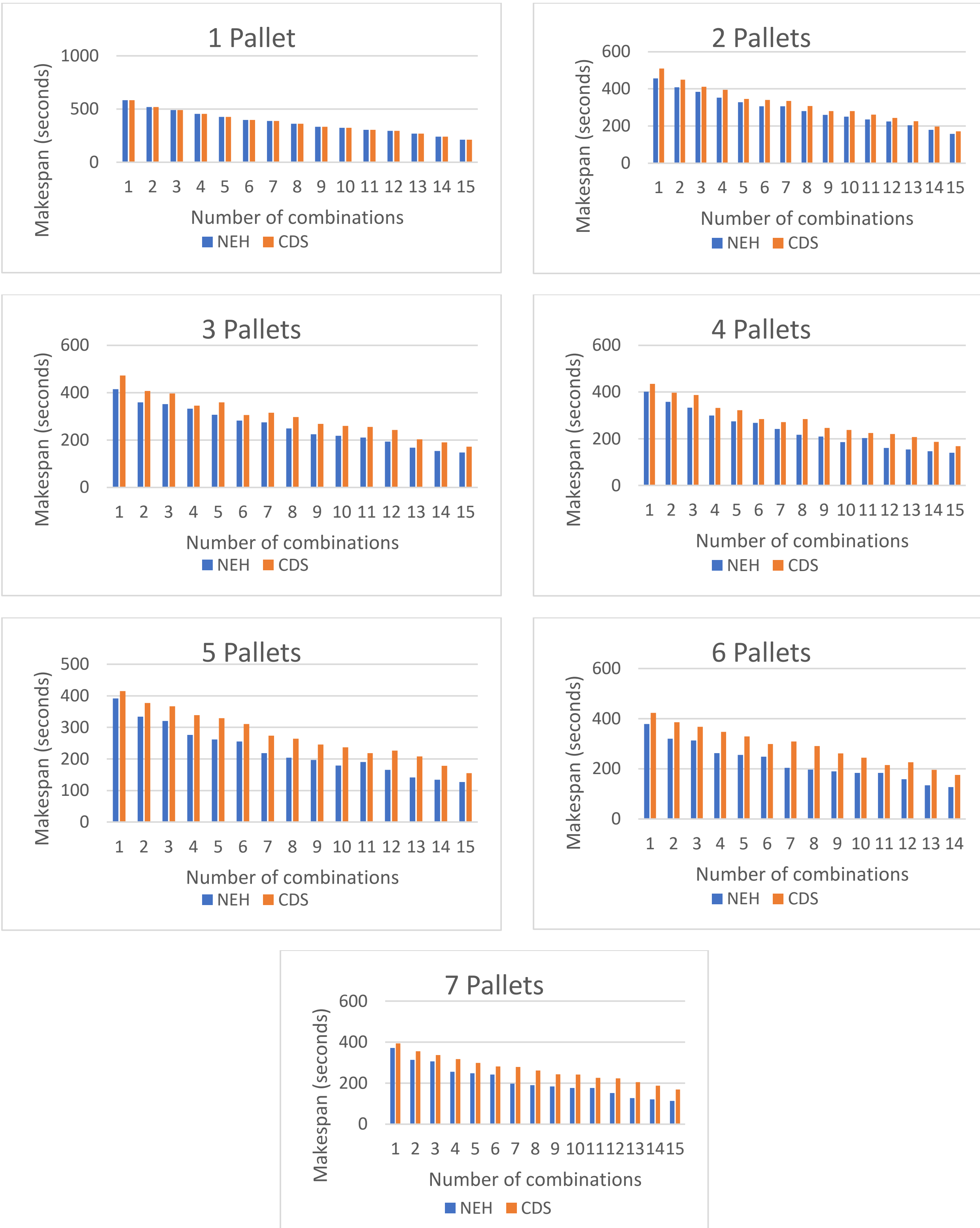


For different combinations of 7 jobs, for both algorithms, they behave almost the same for each specific combination. Graph 6.2.2.10 represents the visual representation of the declining trend for increasing pallet number in each combination.

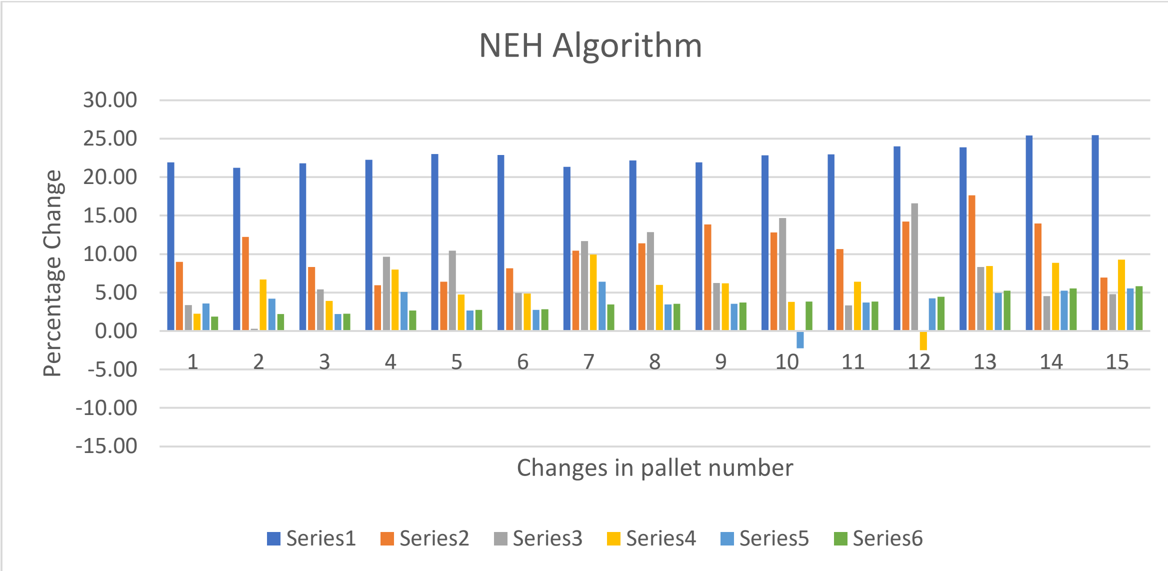
Graph 6.2.2.11 visuals the better performance of NEH algorithm except single pallet.

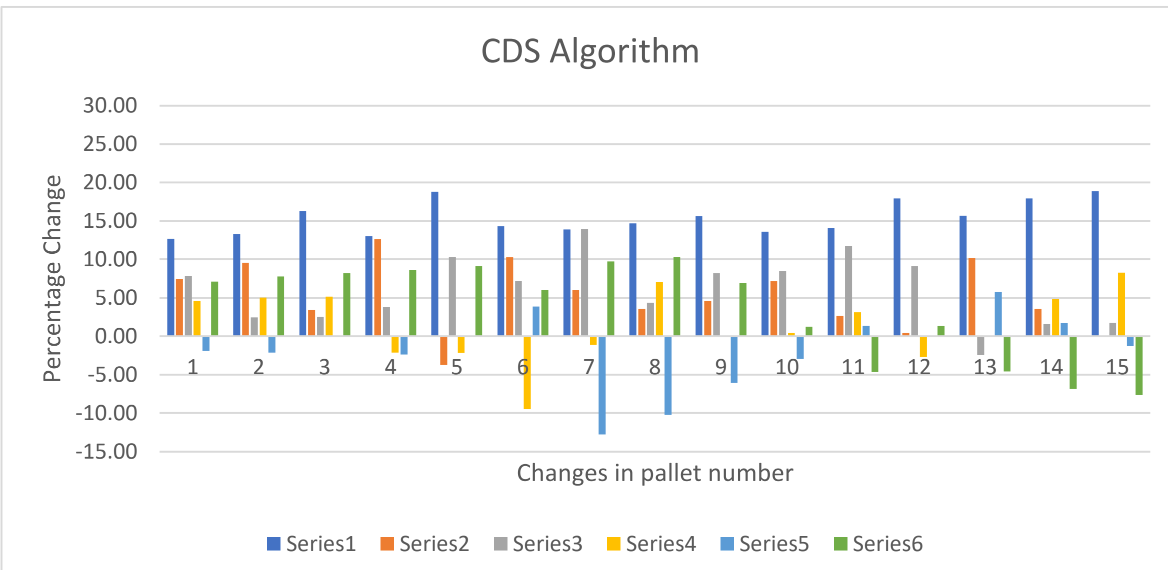
Comparison

Graph 6.2.2.11: Makespan time for different algorithms for mixed category with different percentage of job shop for 7 jobs



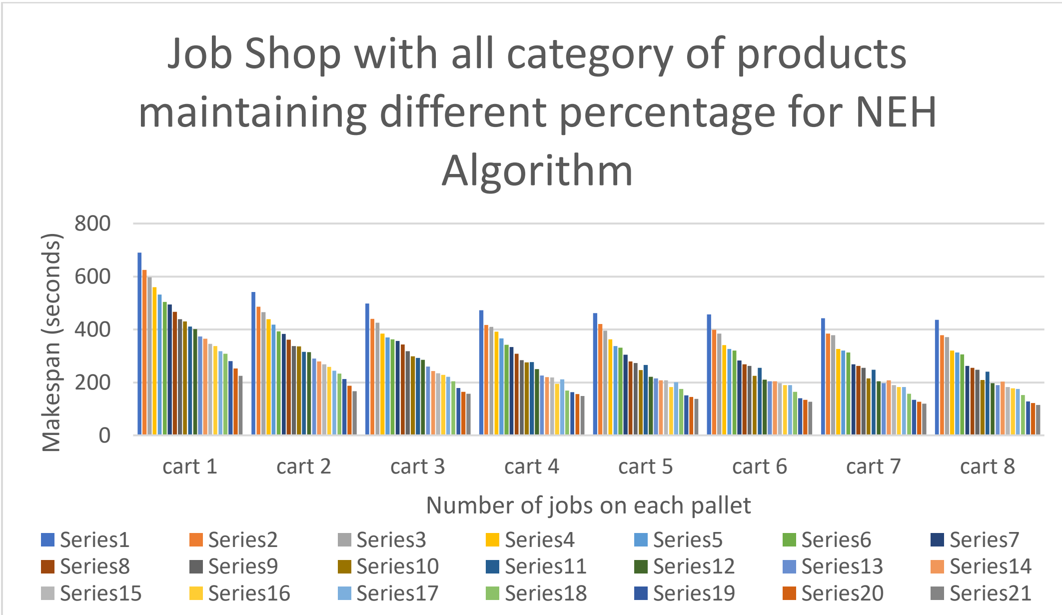
Graph 6.2.2.12: Percentage deviation in makespan with increasing pallet number for combination of 7 jobs

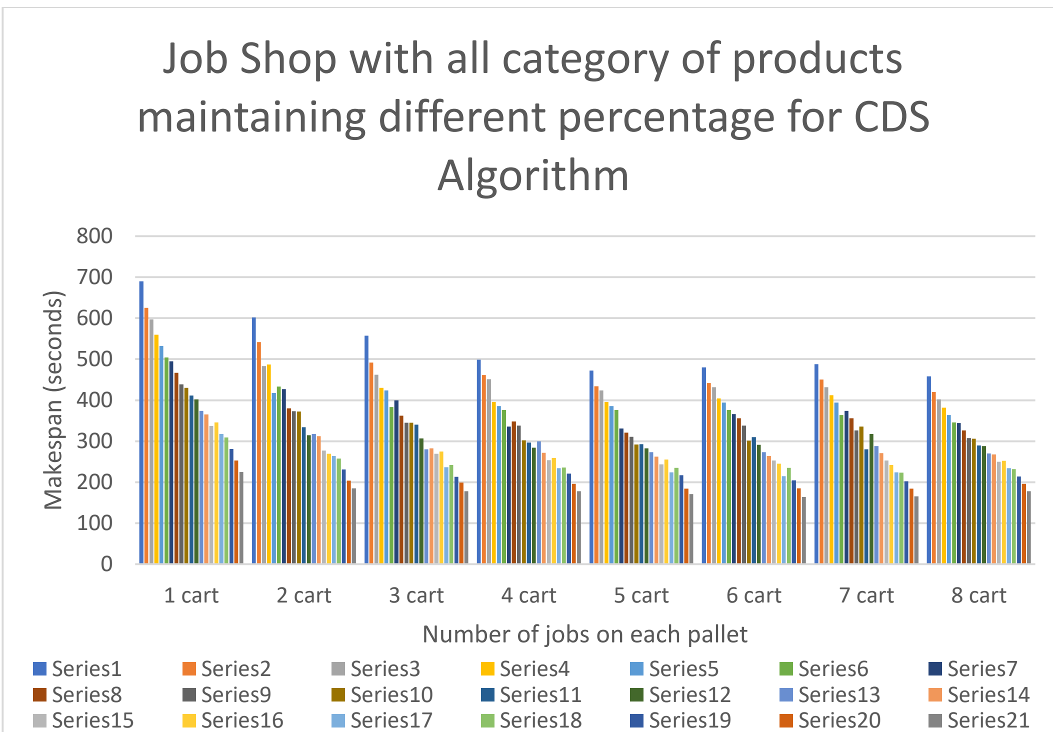




For the NEH algorithm, graph 6.2.2.12 shows the higher increasing trend for changing pallet number from 1 to 2. Similarly, CDS goes upward for that case, but there are some discrepancies in between. Mostly, for CDS algorithm shows, increasing pallet number cannot help the makespan to go down. The reverse behavior shows for maximum easy category jobs. From changing Pallet number to 6 to 7, NEH algorithm shows percentage changes are increasing which means makespan are literally decreasing with increase of pallet number. But for CDS algorithms it is a downward trend.

Graph 6.2.2.13: Makespan time for different number of combinations of 8 jobs in job shop

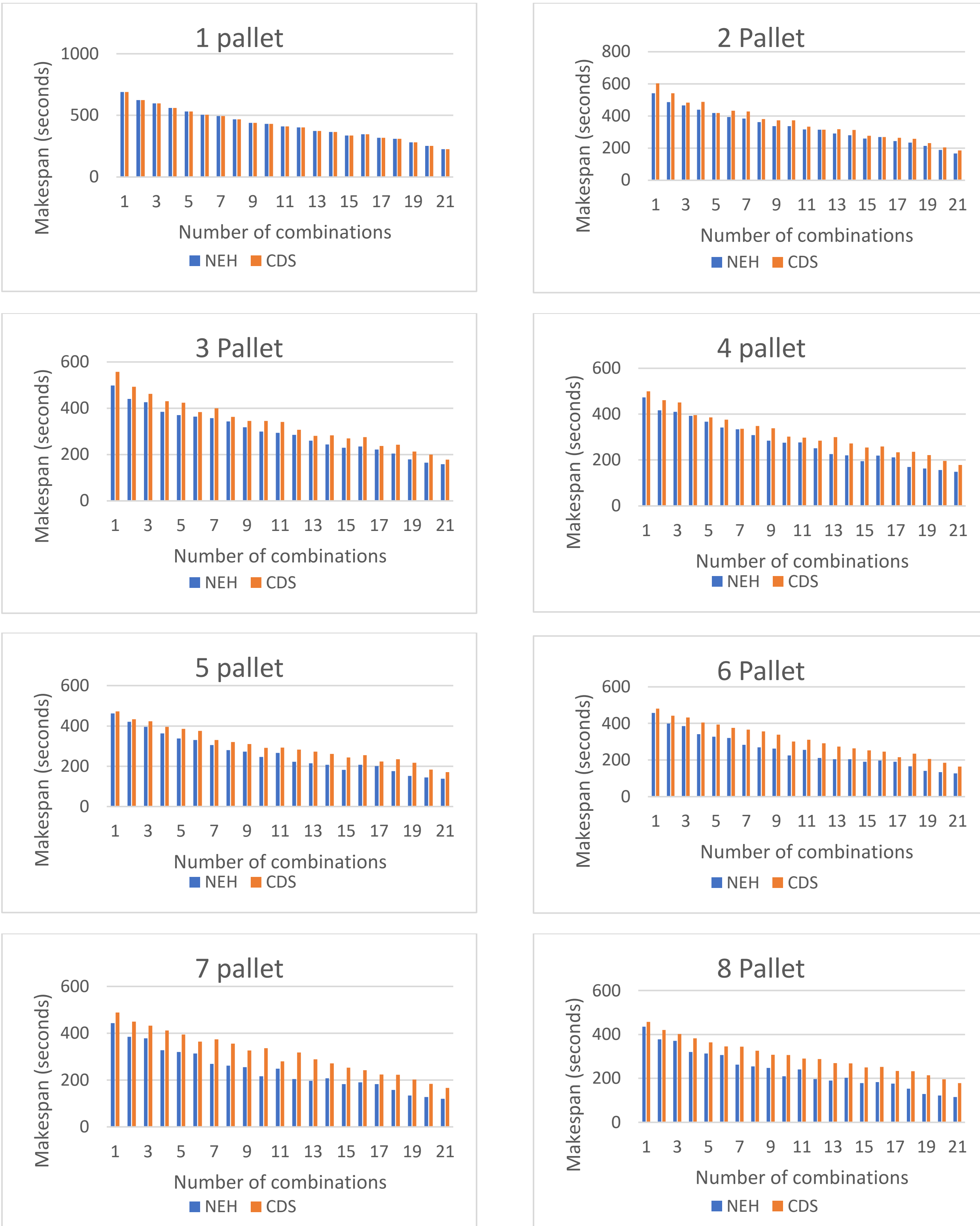




As the number of jobs increases, the possible combination is also increasing. Here, the number of jobs is equal to the all-available pallets. For specific combinations, both algorithms behave almost the same and by utilizing the highest number of pallets will give the minimum makespan. There are some variations due to some specific combination as the sequences are different for both algorithms. Though in most of the cases, makespan decreases with the increasing number of pallets. Graph 6.2.2.13 visually represents that scenario.

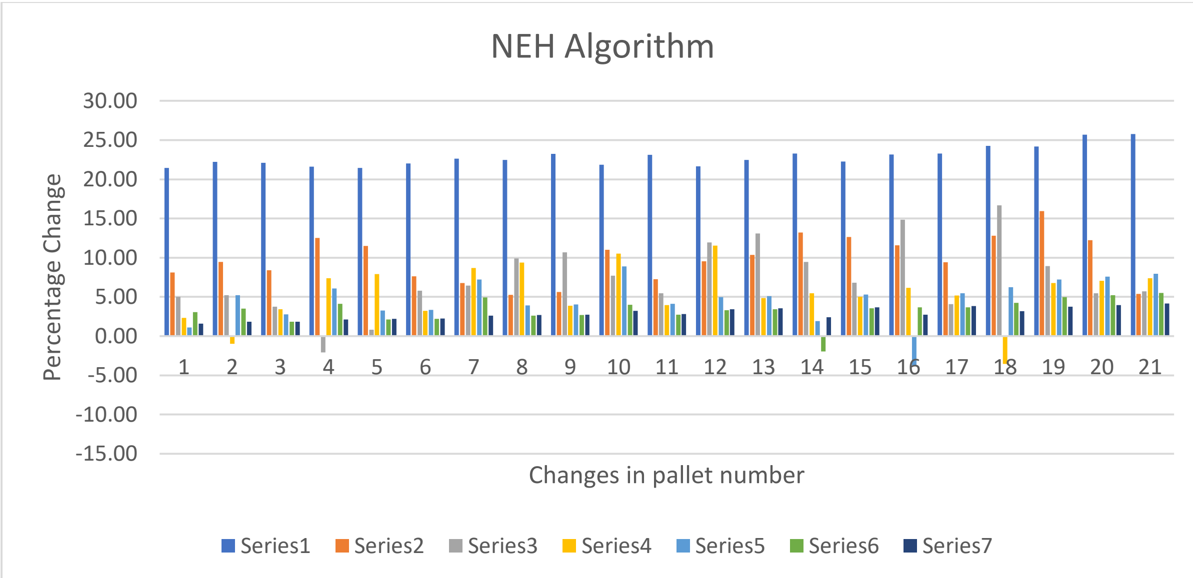
Comparison

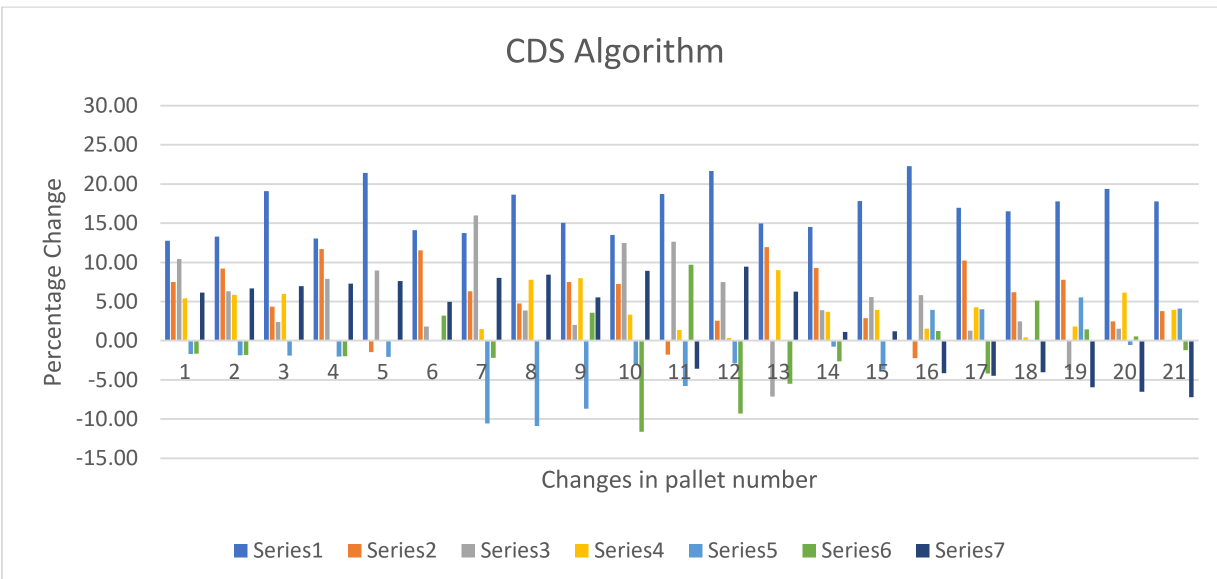
Graph 6.2.2.14: Makespan time for different algorithms for mixed category with different percentage of job shop for 8 jobs



On the other hand, graph 6.2.2.14 presents the better performance of the NEH algorithm for all cases except single pallets.

Graph 6.2.2.15: Percentage deviation in makespan with increasing pallet number for combination of 8 jobs





Finally, graph 6.2.2.15 represents, NEH algorithm has a clear effect on changes in pallet number whereas in most of the cases, the CDS algorithm shows random behavior. The highest percentage increase from pallet 1 to 2 is also for maximum easy category jobs for NEH algorithm. But the CDS algorithm shows a mixed combination for this case where medium category jobs are prioritized.

CHAPTER 7

CONCLUSION

The aim of the thesis has been to evaluate the role of an optimized sequence within a multi-stage manufacturing system. More specifically, the objective has been to model a specific machine setup and find the optimal sequencing technique to improve the efficiency of that specific system using existing algorithms. This work has allowed to achieve the contribution by modifying different heuristics algorithm accommodated with a special machine setup and the related constraints to generate the optimum sequence. The optimal sequence obtained from both algorithms is used in both real-life machine setup and simulated environment to validate the algorithm output and select the best algorithms. Based on the result, makespan showing different behavior for pallet constraints and different shop setup.

A review of the current literature on the topic has been the first task of the research in order to clearly understand the overall context of the manufacturing world and the new era it is undergoing. This analysis led to the acquisition of adequate knowledge on the topic to develop the research and determine the main gaps that have not been addressed yet. The main topic to which this study belongs is Industry 4.0 with a focus on Industrial sequencing practices. The output of this investigation led to the identification of the main gap regarding to take an existing established algorithm and show how that can be used in a very unusual setup, like in a new type of setup in the lab environment that the algorithm was previously not applied off on and was not optimized for. Despite the presence of numerous mathematically established algorithms that are advanced and provide sophisticated solutions that overcome any difficulties in large industries, they are not suitable for customized lab machine setup that have particular needs and limitations compared to big companies. Therefore, it was crucial to investigate this theme in order to support sequencing problems and provide them with something appropriate to the profile. This gap led to the formulation of the research questions with the purpose of developing an appropriate framework that could enable a simulated environment considering their special setup, as well as involving the use of tools and existing software.

The methodology section deals with different algorithms and their adjustments for customized machine setup. In real life, there are so many established algorithms to solve sequencing problems but all cannot be applied in that case. The reason behind rejecting some algorithms and other constraints is focusing on developing new methodology to solve the problem. Modifying existing algorithms helps to find the better optimal solution in that case. Comparison of heuristics algorithms helps to identify a better approach to solve this complicated circular flow shop problem.

Next, validating the framework has been the task of this research. It involved the use of the Festo Didactic CP Lab Learning Systems present in the Smart Manufacturing Lab at WVU in order to perform a production and retrieve the relative data from the MES system connected to it. Indeed, it has been created a simulation model using Arena simulation software that mirrors the existing system in laboratory. It has performed a simulation analysis reading the processed data with the aim of improving the makespan of the production plan optimally launched on the system, by identifying a better sequence using other algorithms that would have required a lower amount of time from its completion. This last task proves the effectiveness of the devised framework by demonstrating how its application could provide help to optimize an aspect of the manufacturing system. Finally, in order to prove the effectiveness of the solution, an additional specific objective has been defined: test how manufacturing data, stored in excel documents, properly processed in an automatic way, compared by a simulation software as Arena, to perform a simulation analysis in order to improve a value adding parameter as the makespan of a production plan launched on a manufacturing system. Python has been the programming language to carry out all of the activities linked to data preparation, data handling and data modeling. Different simulations of the complete sequence of the CP lab manufacturing processes have been executed to collect the needed data for the subsequent modeling and validation phases.

The result analysis focuses on different types of assembly line of production constraints with the existing constraints of the lab environment. The most important effect is created by the pallet number which is actually an important constraint for this unusual machine setup. For all cases, if a single pallet is used, job shops are behaving similarly to flow shops. Comparison of algorithms shows the better performance for NEH heuristics for all of the cases. Increasing pallet numbers will create a significant percentage change for all scenarios. The different ratio of different categories of jobs are taking the minimum time to complete a sequence, when the number of easy category jobs is highest for that specific combination.

7.1 Answer to Research Question

Research Question: How different techniques can apply in the SM lab system and simulated environment to compare the ability to mitigate bottlenecks?

This research question has been conceived with the purpose of providing proof that the methodology development and data validation helps to meet the research objective. The result analysis section answers this question by comparing two different heuristics algorithms and showing their efficiency for both flow shop and job shop scenarios. From the data validation section, it is clear that real machine setup and the simulated data follow the same trend. Comparing the simulated environment result helps to identify the factors influencing the bottleneck. Three different job categories and their processing time with additional cycle time is adding extra idle time for machines and unnecessary non value adding time including transportation time helps to create bottleneck in that scenario. Moreover, a limited number of pallets are an additional constraint to add more time to the makespan. The heuristics algorithm helps to find the best optimal sequence for this irregular machine setup which basically helps to minimize total makespan. Another important finding is that utilizing all available pallets will help to reduce the makespan too for most of the cases. Though easy jobs have lowest utilization of the machines but as the necessary cycles and processing times are minimum for this category, using more easy jobs for a specific combination can also help to minimize the makespan.

7.2 Relevant Real-World Scenarios

Nowadays manufacturing technologies are modified in a vast range after the era of additive manufacturing. CNC machine, injection molding makes the manufacturing world a new scope to develop day by day. The Festo didactic learning system is used for this study to mimic the concept of real-world scenarios.

According to (PCB SHINE International Company Limited 2005), most of the PCB manufacturing companies use drilling, punching and heat treatment to produce a qualified PCB board. Normally after drilling the inner layer, baking and deburring is performed. After platting the second panel, the image should be transferred. Next etching is performed and checked before the solder mask. Then the process goes for CIM printing, gold platting and surface finish. Next completing punching and cleaning, it goes for QC gating and electrical test. Visual inspection is performed before packing and after overall quality assurance it is shipped.

For (RAYMING TECHNOLOGY 2021), if the PCB is double sided, after solder paste printing, it waits for SMD parts placement. Then reflow the soldering it permits to flipping. Again, after solder paste printing and SMD parts placement, it allows reflow soldering. Finally, after THT component placement it goes for hand soldering.

(Song et al., 2016) proposed the engine production line, where the basic steps include installation of left and right crack part bearing. The mounting shock absorption set is pressed. Then after loading the transmission mechanism and shifting mechanism the mold closed. The other steps including fastening combined box bolt, loading neutral position switch, sealing driving shaft oil, loading shift examination, clamping engine, loading lube oil pump, loading side clutch assembly, loading piston cylinder, loading timing chains, loading chain tensioner, loading left crankcase, loading shift control system, fastening left crankcase cover, adjustment of clutch, check backflushing, loading carburetor, leak detection will help to make a qualified engine.

In case of motor production, according to (Attia et al., 2016), after die casting CNC turning is performed. Then drilling, reaming and tapping operation finally goes for motor assembly.

According to (MechGuru 2021), in the bolt fastener manufacturing process, after raw material inspection, casting or forging is performed. Then for facing and grooving, CNC machining is performed. After heat treatment grinding is done for better surface finish. Next after thread rolling, coating is completed to prevent rust. Finally, dimensional inspection happened to ensure form, fit and finish.

(MECHTECH GURU 2014) follows the following procedure for manufacturing rail. After marking, cutting is done at the right angle. Then after drilling the fish hole is chamfered. Necessary Machining is performed before bending. Then after the magnetic particle test it goes for finishing. After final assembly operation, it goes for internal inspection. Next documentation and certification are done based on quality in the final inspection section. After stamping, final painting, stenciling and packing it is gone for market.

To produce a coupler (P. Manne et al., 2016) suggested, after purchasing material it is cut into bulk metal. Then heat the metal and use a conveyor belt to send it to the impression die machine. After making the forge die for the coupler, put material on the forge die. Next force is applied to make proper shape and allow the parts to solidify. After nanoparticle coating and inspection, it will ship.

Following (CCS online clothing study 2013) in the textile industry, especially the garment finishing process works with thread cutting. After initial checking, pressing is done and go for measurement checking. After final checking, tagging, folding and packing is done. Then packing the box leads to inspection of ready to ship goods.

(Pharmaguideline 2022) provides a pathway in the pharmaceutical industry, after receiving raw material from stores and verification of material and weights, individual material shifted for tablet manufacturing. Then dry mixing and drying was performed to analyze moisture level. Next milling is performed for sizing granules and after lubricating, compression is performed. After visual inspection it goes for coating. Again, visual inspection was performed to verify the batch code. Next packing section sends it for finished good analysis.

For the automotive industry (Orbak, 2012) guided, after foam production it is cut in respect to the length of the block. Then completing the settling and drilling process it goes for lamination.

In the field of sliding aluminum windows (HONSTAR 2020) following, after designing phase aluminum profiles are cut. Then it goes for drilling, milling and punching operations. Next going through seal brush, roller and window frame and sash assembly, window lock, thrust block and cushion are installed. Finally, after glass installation it goes for finished aluminum window assembly and then tested for acceptance.

For producing tempered glass (CITEHRBLOG 2012) suggest, after receiving raw glass and inspection of glass it is marked for cutting. After completing edging, holes and cutouts another inspection happens. Then it goes for washing, toughening, pressing, silicon sealing, finishing and curing and final inspection before dispatching.

In the pipe fitting industry such as (B.M Meters Private Limited 2008), after passing the material test, cut materials go for heating. After forming, a boring operation is performed before lathe machining. Finished product inspection is performed before marking and packing to ship on market.

From (B.M Meters Private Limited 2008), flanges are actually metal parts which start working after receiving and inspection of raw materials. Heating is performed before forging and machining. Next passing a different level of inspection it goes more marking, painting and ready to ship.

In oil manufacturing company like (GOYUM GROUP 2021), shea seeds are cleaned from nuts. Clean seed goes through kernel, conditioning and press to produce shea butter. Almost the same process is followed for the sunflower oil and cocoa butter industry.

Focusing on the food industry (Nikken Foods Company Limited 2022) follows that, there are some spices in the form of powder. After collecting raw material and proper inspection they blend them all before going to heat sterilization. Next completing drying and sieve distribution, inspection happens before filling and packaging.

To produce functionally graded materials as per (Parihar et al., 2018) research, powder metallurgy is involved by mixing two types of materials. They are stacked and pressed before sintering.

In the field of paper manufacturing (ECPlaza Network Incorporate 2022) suggests that, woods are converted to wood chips and go for blow thank. After washing it forms pulp and passes through necessary steps to make different types of papers.

For magnet production (Star Trace Private Limited 2022) follows that, raw materials are melted and casted before hydrogen decrepitation. After jet milling, pressing, sintering, machining, plating is performed before inspection. Finally, it goes for magnetizing.

7.3 Limitations

As this research is still at its beginning, the results can be seen as an initial proof of concept or feasibility study for comparison between sequencing algorithms. Therefore, there are some limitations that give room for improvement and future possible developments. The obstacles encountered and the limits of this specific research are going to be listed according to the different steps of the framework.

In the methodology section, heuristics algorithms are chosen to find the optimum sequence. For customized machine setup it is not possible to select other specific algorithms to analyze the best optimal solution for those approaches.

The only data source that was possible to consider was the MES system present in the WVU SM Lab. From the system it was possible to retrieve only data about the production of the different workpieces and the output was a fixed selection of data. Besides, data needed to be exported manually from the MES and they were in csv format. Therefore, as the very first step it was necessary to import these data into an excel file for a better visualization of their content. After the dataset has been created and deeply analyzed, every following step has been conceived in order to fit with the specific structure of the file. Everything has been adapted based on its configuration and its content. For what concerns the transformation and pre-processing of this dataset, this task has been arranged based on what information needed to perform its analysis. The Arena model has been built in order to represent the Festo System present in the lab because running simulations would have been faster than testing on the real system. However, this means that all the parameters of the model are fixed and mirror the real system. In addition, the configuration of the real system could not be rearranged, and it was not the best configuration possible for production. After analyzing the specific context, it emerged that the only variables were the number of job orders for each production and how they were split according to the three different products. Therefore, the .txt file has been created in order to provide what Arena exactly required. As well as, the flow that performs the processing task is specifically created to fit the case. If a different dataset is available and a different kind of analysis wants to be performed, with a different tool, it is necessary to change the data structure accordingly, as well as the process to build it.

There is no previous record to analyze what is the optimum number of jobs for this machine or what trend they are using for increasing makespan or transportation time for different numbers and combinations of jobs. The maximum available pallet for this job is 8. So, it is not possible to check that if the pallet number is more than 8, what should be the possible results. The category of job is fixed and they are repeatedly performing the same task. The machine is not performing continually well for a long time which creates unnecessary bottlenecks in some stations. There happens some sensor issue or other mechanical problem which makes the data collection process longer than expected.

Considering the simulation analysis performed on arena, in order to input the necessary data, it has been built a part of the model that allows for reading them from an external file. Arena was able to directly read data coming from excel or text files, nevertheless in order to appropriately retrieve the data, it has been necessary to rename the single cells containing the data based on how arena was expecting to find them, which depended on how they have been defined in the model. Furthermore, in the construction of the simulation model it has not been possible to model some features and times of the systems due to lack of information. For example, it was not possible to model the transportation times of the workpieces between two consecutive workstations, as well as the time necessary to allow machines' sensors to read the workpiece code. Finally, considering that the production plan was composed of different job orders, it was not feasible to test all the possible scenarios and therefore combinations. For this reason, it was not possible to find the absolute best production sequence.

7.4 FUTURE DEVELOPMENTS

This research is still at its beginning and it is making its first steps, therefore the work developed is not intended to be an end itself. It is more a starting point for future developments, and it aims at fostering new research in this field.

Considering the limitations of this work, a first short-term development for this research could be to test the framework and optimize it for much bigger data samples since for validating this model a relatively small data sample has been considered for time frame reasons and other constraints. Under this perspective it would be valuable to verify if there is a limitation in the data sample size, and check if excel can handle a huge amount of data and what is its limit. It will also be helpful to develop any statistical approach to find the behavior of analyzed data and their relationship according to different time calculations.

Adding transportation time considering pallet numbers will also create another value to this research. Trying to apply any analytical tool for large different combinatorial problems and make it user friendly to handle large datasets. Regarding the limitations, the number of simulations carried out to develop the datasets represents an additional aspect that should be improved; indeed, a higher number of simulations would provide further samples to increase the reliability of the models.

Further specific long-term research would be to take the data and test them with different analytical suites and different algorithms to study how the connections work and if and what kind of issues might emerge. Currently, the framework is case and algorithm specific, the idea would be to make it more standard and developing mathematical models for exact algorithms will open a new era in the sequencing field.

Finally, it would be of extreme relevance to increase the level of automation of the whole workflow, allowing for a complete automatic flow of the data from the source to the analytical tool, without additional manual inputs in between for applying algorithms or other calculations. It would be worthwhile to make further research to provide with appropriate tools that would allow to create an overall automatic process and provide the end user with a best optimal sequence by entering job number and category of products.

The fourth industrial revolution is happening now, hence, the potentialities that can be triggered by new technologies have to be explored and investigated now. This work does not want to be an end in itself. On the contrary, the final aim of this work is the fostering of further research in this field to study more in depth the possibilities that different categories of products and related optimized sequence can generate, both for the efficiency of the sector and for its environmental impact.

References

Aalla, R. (n.d.). *A HEURISTIC ALGORITHM FOR FLOW SHOP SEQUENCING PROBLEMS*.

Abbas, M., Abbas, A., & Khan, W. A. (2016a). Scheduling job shop - A case study. *IOP Conference Series: Materials Science and Engineering*, *146*(1). https://doi.org/10.1088/1757-899X/146/1/012052

Abbas, M., Abbas, A., & Khan, W. A. (2016b). Scheduling job shop - A case study. *IOP Conference Series: Materials Science and Engineering*, *146*(1). https://doi.org/10.1088/1757-899X/146/1/012052

Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J., & Tallent, N. R. (2010). HPCTOOLKIT: Tools for performance analysis of optimized parallel programs. *Concurrency and Computation: Practice and Experience*, *22*(6), 685–701. https://doi.org/10.1002/cpe

Al-Harkan, I. M. (2010). Algorithms for Sequencing and Scheduling. *On Merging Sequencing and Scheduling Theory with Genetic Algorithms to Solve Stochastic Job Shops*, 438.

Applegate, D., & Cook, W. (1991). Computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, *3*(2), 149–156. https://doi.org/10.1287/ijoc.3.2.149

Attia, E.-, Megahed, A., Baioumy, S., Elbetar, A., & Duquenne, P. (2016). Aggregate Production Planning Considering Performance Evolution: a Case Study. *The International Conference on Applied Mechanics and Mechanical Engineering*, *17*(17), 1–16. https://doi.org/10.21608/amme.2016.35291

Baker, K. R. (1984). Sequencing Rules and Due-Date Assignments in a Job Shop. *Management Science*, *30*(9), 1093–1104. https://doi.org/10.1287/mnsc.30.9.1093

Bean, J. C. (1994). Genetic Algorithms and Random Keys for Sequencing and Optimization. *ORSA Journal on Computing*, *6*(2), 154–160. https://doi.org/10.1287/ijoc.6.2.154

Becker, F. G., Cleary, M., Team, R. M., Holtermann, H., The, D., Agenda, N., Science, P., Sk, S. K., Hinnebusch, R., Hinnebusch A, R., Rabinovich, I., Olmert, Y., Uld, D. Q. G. L. Q., Ri, W. K. H. U., Lq, V., Frxqwu, W. K. H., Zklfk, E., Edvhg, L. V, Wkh, R. Q., … فاطمی, ح. (2015). No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title. *Syria Studies*, *7*(1), 37–72. https://www.researchgate.net/publication/269107473_What_is_governance/link/548173090cf22525dcb61443/download%0Ahttp://www.econ.upf.edu/~reynal/Civil wars_12December2010.pdf%0Ahttps://think-asia.org/handle/11540/8282%0Ahttps://www.jstor.org/stable/41857625

Belabid, J., Aqil, S., & Allali, K. (2020a). Solving Permutation Flow Shop Scheduling Problem with Sequence-Independent Setup Time. *Journal of Applied Mathematics*, *2020*. https://doi.org/10.1155/2020/7132469

Belabid, J., Aqil, S., & Allali, K. (2020b). Solving Permutation Flow Shop Scheduling Problem with Sequence-Independent Setup Time. *Journal of Applied Mathematics*, *2020*. https://doi.org/10.1155/2020/7132469

Bernedixen, J., Pehrsson, L., Ng, A. H. C., & Antonsson, T. (2016). Simulation-based multi-objective bottleneck improvement: Towards an automated toolset for industry. *Proceedings - Winter Simulation Conference*, *2016-February*, 2183–2194. https://doi.org/10.1109/WSC.2015.7408331

*B.M Meters Private Limited, "B&M FORGED PIPE FITTINGS *MANUFACTURING PROCESS CHART*", 2008*. [Online]. Available: http://www.bmmeters.com/forged-pipe-fittings-flow-chart.html

*B.M Meters Private Limited, "B&M FORGED FLANGES *MANUFACTURING PROCESS CHART*", 2008*. [Online]. Available: http://www.bmmeters.com/forged-fittings-flanges-process-chart.html

Brucker, P., & Schlie, R. (1990). Computing Job-Shop Scheduling with Multi-Purpose Machines. In *Computing* (Vol. 45).

Brunner, C. C., & Funck, J. W. (2015). *A Discrete-Event Object-Oriented Modeling Environment For Sawmill Simulation*. 119–130.

Byung Park, Y. (1984). *The Solution of Vehicle Routing Problems in a multiple objective environment*.

Byung Park, Y. (1988). An evaluation of static flowshop scheduling heuristics in dynamic flowshop models via a computer simulation. *Computers and Industrial Engineering*, *14*(2), 103–112. https://doi.org/10.1016/0360-8352(88)90019-8

CAMPBELL HG, DUDEK RA, & SMITH ML. (1970). Heuristic Algorithm for the N Job, M Machine Sequencing Problem. *Management Science*, *16*(10), 630–637.

CCS online clothing study, "Garment Manufacturing Process Flow Chart", 2013. [Online]. Available: https://www.onlineclothingstudy.com/2013/05/garment-manufacturing-process-flow-chart.html

Cezarino, L. O., Liboni, L. B., Oliveira Stefanelli, N., Oliveira, B. G., & Stocco, L. C. (2019). Diving into emerging economies bottleneck: Industry 4.0 and implications for circular economy. *Management Decision*, *59*(8), 1841–1862. https://doi.org/10.1108/MD-10-2018-1084

Cheng, R., Gen, M., & Tsujimura, Y. (1996). A TUTORIAL SURVEY OF JOB-SHOP SCHEDULING PROBLEMS USING GENETIC ALGORITHMS-I. REPRESENTATION. In *Computers ind. Engng* (Vol. 30, Issue 4).

CITEHRBLOG, "Glass Manufacturing Process", 2012. [Online]. Available: https://citehrblog.wordpress.com/2012/07/30/glass-manufacturing-process/

Das, M. K. (2014). *Selected Heuristic Algorithms for Solving Job Shop and Flow Shop Scheduling Problems*. 1–36.

Drezner, Z. (1987). Operational Research Society is collaborating with JSTOR to digitize, preserve, and extend access to Journal of the Operational Research Society. ® www.jstor.org. *Journal of the Operational Research Society*, *38*(6), 509–514.

ECPlaza Network Incorporate, "TOILET PAPER MACHINE process flow chart", 2022. [Online]. Available: https://www.ecplaza.net/products/equipment-manufacturers-price-of-paper-mill_4203812

EMMONS H. (1969). One-Machine Sequencing To Minimize Certain Functions of Job Tardiness. *Operations Research*, *17*(4), 701–715. https://doi.org/10.1287/opre.17.4.701

Faget, P., Eriksson, U., & Herrmann, F. (2005). Applying discrete event simulation and an automated bottleneck analysis as an aid to detect running production constraints. *Proceedings - Winter Simulation Conference*, *2005*(2001), 1401–1407. https://doi.org/10.1109/WSC.2005.1574404

GOYUM GROUP, *"Sheanut Oil Manufacturing Process Flowchart", 2021*. [Online]. Available: https://www.oilexpeller.com/sheanut-oil-manufacturing-process/

Haque, T. M., Hossain, R. M., & Hasan, S. M. (2018). Bottleneck problem reduction of a garment manufacturing industry in Bangladesh by using line balancing technique. *International Journal of Research in Advanced Engineering and Technology*, *4*(2), 28–32. https://doi.org/10.13140/RG.2.2.32627.84004

HONSTAR, "Something that you may not know about Aluminium window processing flow", 2020. [Online]. Available: https://www.alu4all.com/something-that-you-may-not-know-about-aluminum-window-processing-flow/

Hübl, A. (2018). Dispatching Rules. *Stochastic Modelling in Production Planning*, 109–120. https://doi.org/10.1007/978-3-658-19120-7_6

Jain, A. S., & Meeran, S. (n.d.). *Theory and Methodology Deterministic job-shop scheduling: Past, present and future*.

Kassu, J., & Eshetie, B. (2015). Job Shop Scheduling Problem for Machine Shop with Shifting Heuristic Bottleneck. *Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc*, *15*.

Kelton, W. D., Sadowski, R. P., & Swets, N. B. (2010). *Simulation with Arena International Edition.*

King, J. R., & Spachis, A. S. (1980). Heuristics for flow-shop scheduling. *International Journal of Production Research*, *18*(3), 345–357. https://doi.org/10.1080/00207548008919673

Králová, Z., & Leporis, M. (2010). *A SIMULATION APPROACH TO PRODUCTION LINE BOTTLENECK ANALYSIS International Conference CYBERNETICS AND INFORMATICS VYŠNÁ BOCA, Slovak Republic A SIMULATION APPROACH TO PRODUCTION LINE BOTTLENECK ANALYSIS*. https://www.researchgate.net/publication/354381258

Laarhoven, P. J. M. Van, Aarts, E. H. L., & Lenstra, J. K. (1992). *Job Shop Scheduling by Simulated Annealing*. *August 2015*.

Lanke, A. A., Hoseinie, S. H., & Ghodrati, B. (2016). Mine production index (MPI)-extension of OEE for bottleneck detection in mining. *International Journal of Mining Science and Technology*, *26*(5), 753–760. https://doi.org/10.1016/j.ijmst.2016.05.050

Li, L., Chang, Q., & Ni, J. (2009). Data driven bottleneck detection of manufacturing systems. *International Journal of Production Research*, *47*(18), 5019–5036. https://doi.org/10.1080/00207540701881860

Liu, K., Chen, J., Yang, Y., & Jin, H. (2008). A throughput maximization strategy for scheduling transaction-intensive workflows on SwinDeW-G. *Concurrency and Computation: Practice and Experience*, *20*(15), 1807–1820. https://doi.org/10.1002/cpe.1316

Liu, W., Jin, Y., & Price, M. (2017). A new improved NEH heuristic for permutation flowshop scheduling problems. *International Journal of Production Economics*, *193*(June 2016), 21–30. https://doi.org/10.1016/j.ijpe.2017.06.026

Manne, A. S. (1960). *On the Job-Shop Scheduling Problem*. *February 2015*.

Manne, P., Ahmad, S., & Waterman, J. (2016). *Design and Development of Mine Railcar Components*. *4*(1), 32–41. https://doi.org/10.12691/ajme-4-1-5

Mckay, K. N., Safayeni, F. R., & Buzacott, J. A. (1988). *Job-Shop Scheduling Theory : What Is Relevant ? August 2015*.

MechGuru, "Bolt Fastener Manufacturing Process Flow Chart", 2021. [Online]. Available: https://mechguru.com/machine-design/7-steps-manufacturing-process-bolts-screws-stud-fasteners/attachment/bolt-fastener-manufacturing-process-flow-chart/

MECHTECH GURU, "PROCESS FLOW CHART: MANUFACTURING PROCESS QUALITY CONTROL ZU-1-60 THICK WEB CURVED SWITCHES RT-6155, 2014. [Online]. Available: https://www.mechtechguru.com/2021/05/process-flow-chart-manufacturing.html

Mellor, P. (1965). *A Review of Job Shop Schedulingt*. 161–171.

Mönch, T., Huchzermeier, A., & Bebersdorf, P. (2021). Variable takt times in mixed-model assembly line balancing with random customisation. *International Journal of Production Research*, *59*(15), 4670–4689. https://doi.org/10.1080/00207543.2020.1769874

Moore, J. M. (1968). An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs . *Management Science*, *15*(1), 102–109. https://doi.org/10.1287/mnsc.15.1.102

Muthiah, K. M. N., & Huang, S. H. (2007). Overall throughput effectiveness (OTE) metric for factory-level performance monitoring and bottleneck detection. *International Journal of Production Research*, *45*(20), 4753–4769. https://doi.org/10.1080/00207540600786731

Muthiah, K. M. N., Huang, S. H., & Mahadevan, S. (2008). Automating factory performance diagnostics using overall throughput effectiveness (OTE) metric. *International Journal of Advanced Manufacturing Technology*, *36*(7–8), 811–824. https://doi.org/10.1007/s00170-006-0891-x

Nageswara, M., Narayana, R., & Ranga, J. (2017). Integrated Scheduling of Machines and Agvs in Fms By Using Dispatching Rules. *Journal of Production Engineering*, *20*(1), 75–84. https://doi.org/10.24867/jpe-2017-01-075

Nawaz, M., Enscore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95. https://doi.org/10.1016/0305-0483(83)90088-9

Nikken Foods Company Limited, "Manufacturing Process", 2022. [Online]. Available: https://www.nikkenfoods.co.jp/en/overview/flow.html

Noguera, C., Fritz, B., & Clément, A. (2015). a Simulation Approach . *Journal of Colloid and Interface Science*, *February 2010*. http://dx.doi.org/10.1016/j.jcis.2015.02.050

Ongbali, S. O., Afolalu, S. A., Oyedepo, S. A., Aworinde, A. K., & Fajobi, M. A. (2021). A study on the factors causing bottleneck problems in the manufacturing industry using principal component analysis. *Heliyon*, *7*(5), e07020. https://doi.org/10.1016/j.heliyon.2021.e07020

Orbak, Â. Y. (2012). Shell scrap reduction of foam production and lamination process in automotive industry. *Total Quality Management and Business Excellence*, *23*(3–4), 325–341. https://doi.org/10.1080/14783363.2011.637813

Out, I. N. (2010). *in ןוחטיב Out קט -ייה*. 1–6.

Page, E. S. (1961). An Approach to the Scheduling of Jobs on Machines. *Journal of the Royal Statistical Society: Series B (Methodological)*, *23*(2), 484–492. https://doi.org/10.1111/j.2517-6161.1961.tb00432.x

Parihar, R. S., Setti, S. G., & Sahu, R. K. (2018). Recent advances in the manufacturing processes of functionally graded materials: A review. *IEEE Journal of Selected Topics in Quantum Electronics*, *25*(2), 309–336. https://doi.org/10.1515/secm-2015-0395

PCB SHINE International Company Limited, "Process Flow Chart", 2005. [Online]. Available: http://www.pcbshine.com/process.htm

Pharmaguideline, "Tablet Manufacturing Process Flowchart", 2022. [Online]. Available: https://www.pharmaguideline.com/2012/04/tablet-manufacturing-process-flowchart.html

Prasetyo, Y. T., & Veroya, F. C. (2020). An Application of Overall Equipment Effectiveness (OEE) for Minimizing the Bottleneck Process in Semiconductor Industry. *2020 IEEE 7th International Conference on Industrial Engineering and Applications, ICIEA 2020*, *3*, 345–349. https://doi.org/10.1109/ICIEA49774.2020.9101925

Rahman, N. S. A., Al Hazza, M. H. F., & Kays, E. (2018). Minimizing makespan of multimachine production system in flow shop environment by means of mixed integer programming model. *International Journal of Mechanical Engineering and Technology*, *9*(8), 1102–1110.

RAYMING TECHNOLOGY, "An Introduction to 2 Layer PCB", 2021. [Online]. Available: https://www.pcbcart.com/article/content/pcb-assembly-process.html

Roser, C., Lorentzen, K., & Deuse, J. (2015). Reliable shop floor bottleneck detection for flow lines through process and inventory observations: the bottleneck walk. *Logistics Research*, *8*(1), 1–9. https://doi.org/10.1007/s12159-015-0127-2

Roser, C., Nakano, M., & Tanaka, M. (2001). A practical bottleneck detection method. *Winter Simulation Conference Proceedings*, *2*, 949–953. https://doi.org/10.1109/wsc.2001.977398

Roser, C., Nakano, M., & Tanaka, M. (2002). Shifting bottleneck detection. *Winter Simulation Conference Proceedings, 2*, 1079–1086. https://doi.org/10.1109/wsc.2002.1166360

Salihah, N., Rahman, A., Al Hazza, H. F., & Kays, E. (2018). Minimizing Makespan of Multi-Machine Production System in Flow Shop Environment By Means of Mixed Integer Programming Model. *International Journal of Mechanical Engineering and Technology (IJMET*, *9*(8), 1102–1110. http://iaeme.com/Home/journal/IJMET1102editor@iaeme.comhttp://iaeme.com/Home/issue/IJMET?Volume=9&Issue=8http://iaeme.com/Home/issue/IJMET?Volume=9&Issue=8http://iaeme.com/Home/journal/IJMET1103

Sauvey, C., & Sauer, N. (2020). Two NEH heuristic improvements for flowshop scheduling problem with makespan criterion. *Algorithms*, *13*(5), 1–14. https://doi.org/10.3390/A13050112

Song, L., Jin, S., & Tang, P. (2016). Simulation and optimization of logistics distribution for an engine production line. *Journal of Industrial Engineering and Management*, *9*(1), 59–72. https://doi.org/10.3926/jiem.1555

Storer, R. H., Wu, S. D., & Vaccari, R. (1992). New search spaces for sequencing problems with application to job shop scheduling. *Management Science*, *38*(10), 1495–1509. https://doi.org/10.1287/mnsc.38.10.1495

Star Trace Private Limited, "Magnet Processing Plant", 2022. [Online]. Available: https://www.startraceltd.com/magnet-processing-plant.html

Su, Z., & Yi, J. (2017). *A Two-phase Heuristic Algorithm for No-wait Flow Shop Scheduling 2 Problem Formulation*. https://doi.org/10.1515/9783110540048-049

Subramaniyan, M., Skoogh, A., Salomonsson, H., Bangalore, P., Gopalakrishnan, M., & Sheikh Muhammad, A. (2018). Data-driven algorithm for throughput bottleneck analysis of production systems. *Production and Manufacturing Research*, *6*(1), 225–246. https://doi.org/10.1080/21693277.2018.1496491

Sumathi, S., Kumar, L., & P, S. (2015). Job Shop Scheduling Problem. *Computational Intelligence Paradigms for Optimization Problems Using MATLAB®/SIMULINK®*, *15*(1), 367–419. https://doi.org/10.1201/b19134-6

Thomopoulos, N. T. (1967). *Line Balancing-Sequencing for Mixed-Model Assembly LINE BALANCING-SEQUENCING FOR*. *August 2015*.

Torlino, H. (2020).

Troutt, M. D., White, G. P., & Tadisina, S. K. (2001). Maximal flow network modelling of production bottleneck problems. *Journal of the Operational Research Society*, *52*(2), 182–187. https://doi.org/10.1057/palgrave.jors.2601071

Urban, W., & Rogowska, P. (2020). Methodology for bottleneck identification in a production system when implementing TOC. *Engineering Management in Production and Services*, *12*(2), 74–82. https://doi.org/10.2478/emj-2020-0012

Uzorh, E. A. C., & Innocent, N. (2014). Solving Machine Shops Scheduling Problems using Priority Sequencing Rules Techniques. *The International Journal Of Engineering And Science*, *3*(6), 1813–2319.

Wedel, M., Noessler, P., & Metternich, J. (2016). Development of bottleneck detection methods allowing for an effective fault repair prioritization in machining lines of the automobile industry. *Production Engineering*, *10*(3), 329–336. https://doi.org/10.1007/s11740-016-0672-9

Zahraee, S. M., Golroudbary, R., Hashemi, A., Afshar, J., & Haghighi, M. (2014). *Simulation of Manufacturing Production Line Based on Arena*. *933*, 744–748. https://doi.org/10.4028/www.scientific.net/AMR.933.744

Appendix

A. Data Validation

Table 5.1.1: Input file preparation to feed an algorithm

Index Number	0	1	2	3	4	5	6	7	8	9	10
Machine	a1.1	a1.2	a1.3	b1.1	b1.2	b1.3	b1.4	c1.1	c1.2	c1.3	c1.4
Front	2	0	0	2	0	0	0	2	0	0	0
Output	0	0	7	0	0	0	7	0	0	0	7
Muscle Press	0	0	0	0	0	25	0	0	0	25	0
Heat Tunnel	0	0	0	0	0	0	0	0	0	55	0
Turning	0	0	0	0	0	3	0	0	0	3	0
Back	0	2	0	0	2	0	0	0	2	0	0
Measuring	2	0	0	2	0	0	0	2	0	0	0
Drilling	0	0	0	0	0	0	0	10	0	0	0

Figure 5.2: Algorithm input

File Edit View

```
8     Number of machines
11    Number of Virtual jobs
2  0  0  2  0  0   0  2   0  0   0
0  0  7  0  0  0   7  0   0  0   7
0  0  0  0  0  25  0  0   0  25  0
0  0  0  0  0  0   0  0   0  55  0
0  0  0  0  0  3   0  0   0  3   0
0  2  0  0  2  0   0  0   2  0   0
2  0  0  2  0  0   0  2   0  0   0
0  0  0  0  0  0   0  10  0  0   0
```

Table: 5.2: Algorithm outputs

8	4	1	7	9	3	0	10	6	2	5

Table 5.4.1: Total Processing time calculation

Job	Front	Output	Muscle Press	Heat Tunnel	Turning	Back	Measuring	Drilling
C1.1	2	0	0	0	0	0	2	0
M1.1	0	0	0	0	0	2	0	0
E1.1	0	7	0	0	0	0	0	0
C1.2	2	0	0	0	0	0	2	0
M1.2	0	0	0	0	0	2	0	0
E1.2	0	0	25	0	3	0	0	0
C1.3	0	7	0	0	0	0	0	0
M1.3	2	0	0	0	0	0	2	10
E1.3	0	0	0	0	0	2	0	0
C1.4	0	0	25	55	3	0	0	0
M1.4	0	7	0	0	0	0	0	0
Sum	6	21	50	55	6	6	6	10

Table: 5.4.2: Makespan calculation in/out table

JOB	Front in	Front out	Output in	Output out	Muscle press in	Muscle press out	Heating tunnel in	Heating tunnel out	Turning in	Turning out	Back in	Back out	Measuring in	Measuring out	Drilling in	Drilling out
C1.1	0	2	2	2	2	2	2	2	2	2	2	2	2	4	4	14
M1.1	2	4	4	4	4	4	4	4	4	4	4	4	4	6	14	14
E1.1	4	6	6	6	6	6	6	6	6	6	6	6	6	8	14	14
C1.2	6	6	6	6	6	6	6	6	6	6	6	8	8	8	14	14
M1.2	6	6	6	6	6	6	6	6	6	6	8	10	10	10	14	14
E1.2	6	6	6	6	6	6	6	6	6	6	10	12	12	12	14	14
C1.3	6	6	6	6	6	31	31	86	86	89	89	89	89	89	89	89
M1.3	6	6	6	6	31	56	86	86	89	92	92	92	92	92	92	92
E1.3	14	14	14	21	56	56	86	86	92	92	92	92	92	92	92	92
C1.4	14	14	21	28	56	56	86	86	92	92	92	92	92	92	92	92
M1.4	14	14	28	35	56	56	86	86	92	92	92	92	92	92	92	92

Table 5.4.3: Idle time calculation

JOB	Front out	Output out	Muscle press out	Heating tunnel out	Turning out	Back out	Measuring out	Drilling out
C1.1	0	2	2	2	2	2	2	4
M1.1	0	2	2	2	2	2	0	0
E1.1	0	2	2	2	2	2	0	0
C1.2	0	0	0	0	0	0	0	0
M1.2	0	0	0	0	0	0	2	0
E1.2	0	0	0	0	0	0	2	0
C1.3	0	0	0	25	80	77	77	75
M1.3	0	0	0	0	0	3	3	3
E1.3	8	8	0	0	0	0	0	0
C1.4	0	0	0	0	0	0	0	0
M1.4	78	57	36	6	0	0	0	0
Sum	86	71	42	37	86	86	86	82

Table 5.4.4: Bottleneck Calculation

JOB	Front out	Output out	Muscle press out	Heating tunnel out	Turning out	Back out	Measuring out	Drilling out
C1.1	0	0	0	0	0	0	0	0
M1.1	0	0	0	0	0	0	0	8
E1.1	0	0	0	0	0	0	0	6
C1.2	0	0	0	0	0	0	0	6
M1.2	0	0	0	0	0	2	0	4
E1.2	0	0	0	0	0	4	0	2
C1.3	0	0	0	0	0	0	0	0
M1.3	0	0	25	30	3	0	0	0
E1.3	0	0	35	30	6	0	0	0
C1.4	0	7	28	30	6	0	0	0
M1.4	0	14	21	30	6	0	0	0

Table 5.5: Legend for the first 15 header columns of the machine dataset CSV file

Column	Meaning	Note									
PNo	Product Number	Product Number	3000	3001		3002					
		Meaning	Easy	Medium		Complex					
WPNo	Working Plan Number	Working Plan Number	6	7		8					
		Meaning	Easy	Medium		Complex					
StepNo	Step/ Module Number	Step/ Module Number	10	20	30	40	50	60	70	80,200	
		Meaning	Front	Measure	Drill	Back	Muscle Press	Heat Tunnel	Turn	Deliver	
ONo	Order Number	Specific customer order number									
OPos	Order Position	Specific job position number according to input									
Description	Module Task	Description of the activity performed									
OpNo	Operation Number	Number of the operation performed by the relative module									
NextStepNo	Next step Number	Next step Number	0	20	30	40	50	60	70	80	
		Meaning	Deliver	Measure	Drill	Back	Muscle Press	Heat Tunnel	Turn	Deliver	
FirstStep		TRUE/FALSE depending whether the considered step is the first one or not									
ErrorStepNo		200: defines the error step for the relative module									
NewPNo	New Product Number	The product n. of the product worked on the relative module. 111 is the product n. for front cover									
PlannedStart	Planned Start	Date-time of planned production start for the step									
PlannedEnd	Planned End	Date-time of planned production end for the step									
Start		Date-time of real production start for the step									
End		Date-time of real production end for the step									

Table 5.5 continued:

PNo	WPNo	StepNo	ONo	OPos	Description	OpNo	NextStepNo	FirstStep	ErrorStepNo	NewPNo
3002	8	10	2709	1	feed part from magazin	200	20	TRUE	200	0
3002	8	20	2709	1	measure a part (analog)	115	30	FALSE	200	0
3002	8	30	2709	1	drilling custom	123	40	FALSE	200	111
3002	8	40	2709	1	feed part from magazin	200	50	FALSE	200	3002
3002	8	50	2709	1	pressing with force regulation	111	60	FALSE	200	0
3002	8	60	2709	1	heating Part	112	70	FALSE	200	0
3002	8	70	2709	1	turning part	113	80	FALSE	200	0
3002	8	80	2709	1	deliver part	205	0	FALSE	200	0
3002	8	200	2709	1	deliver part	205	0	FALSE	0	0
3001	7	10	2709	2	feed part from magazin	200	20	TRUE	200	0
3001	7	20	2709	2	measure a part (analog)	115	40	FALSE	200	111
3001	7	40	2709	2	feed part from magazin	200	50	FALSE	200	3001
3001	7	50	2709	2	pressing with force regulation	111	70	FALSE	200	0
3001	7	70	2709	2	turning part	113	80	FALSE	200	0
3001	7	80	2709	2	deliver part	205	0	FALSE	200	0
3001	7	200	2709	2	deliver part	205	0	FALSE	0	0
3000	6	10	2709	3	feed part from magazin	200	20	TRUE	200	0
3000	6	20	2709	3	measure a part (analog)	115	40	FALSE	200	111
3000	6	40	2709	3	feed part from magazin	200	80	FALSE	200	3000
3000	6	80	2709	3	deliver part	205	0	FALSE	200	0
3000	6	200	2709	3	deliver part	205	0	FALSE	0	0

Table 5.5 Continued:

PlanedStart	PlanedEnd	Start	End	OPNoType	ResourceID	TransportTime	ErrorStep
10/23/2022 16:52	10/23/2022 16:52	10/23/2022 16:53	10/23/2022 16:53	1	1	0	FALSE
10/23/2022 16:52	10/23/2022 16:52	10/23/2022 16:54	10/23/2022 16:54	1	2	8	FALSE
10/23/2022 16:52	10/23/2022 16:53	10/23/2022 16:54	10/23/2022 16:54	1	3	2	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:55	10/23/2022 16:55	1	4	9	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:56	10/23/2022 16:57	1	5	7	FALSE
10/23/2022 16:53	10/23/2022 16:54	10/23/2022 16:57	10/23/2022 16:57	1	6	2	FALSE
10/23/2022 16:54	10/23/2022 16:54	10/23/2022 16:58	10/23/2022 16:58	1	7	2	FALSE
10/23/2022 16:54	10/23/2022 16:54	10/23/2022 16:59	10/23/2022 16:59	1	8	6	FALSE
10/23/2022 16:52	10/23/2022 16:52			1	8	6	TRUE
10/23/2022 16:52	10/23/2022 16:52	10/23/2022 16:53	10/23/2022 16:53	1	1	0	FALSE
10/23/2022 16:52	10/23/2022 16:53	10/23/2022 16:54	10/23/2022 16:54	1	2	8	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:56	10/23/2022 16:56	1	4	10	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:58	10/23/2022 16:58	1	5	7	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:59	10/23/2022 16:59	1	7	4	FALSE
10/23/2022 16:53	10/23/2022 16:54	10/23/2022 17:00	10/23/2022 17:00	1	8	6	FALSE
10/23/2022 16:52	10/23/2022 16:52			1	8	6	TRUE
10/23/2022 16:52	10/23/2022 16:52	10/23/2022 16:54	10/23/2022 16:54	1	1	0	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:55	10/23/2022 16:55	1	2	8	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:56	10/23/2022 16:57	1	4	10	FALSE
10/23/2022 16:53	10/23/2022 16:53	10/23/2022 16:58	10/23/2022 16:58	1	8	5	FALSE
10/23/2022 16:52	10/23/2022 16:52			1	8	5	TRUE

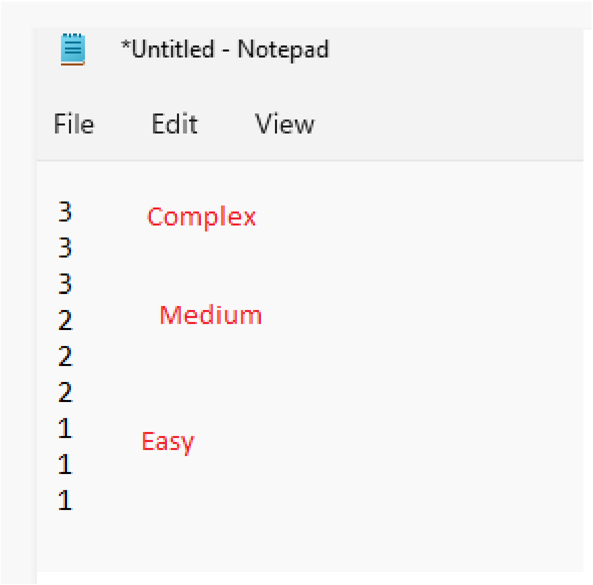
Table 5.5 continued:

ElectricEnergyCalc	ElectricEnergyReal	CompressedAirCalc	CompressedAirReal	FreeString	StaffId
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0

Table 5.6: Comparison of machine data and simulated data

EASY	MEDIUM	COMPLEX	TOTAL	SEQUENCE	MACHINE	SIMULATION
1	0	0	1	Flow	198	13
5	0	0	5	Flow	290	47
7	0	0	7	Flow	297	65
9	0	0	9	Flow	503	81
11	0	0	11	Flow	540	95
12	0	0	12	Flow	555	104
15	0	0	15	Flow	604	137
18	0	0	18	Flow	823	160
0	7	0	7	Flow	517	201
0	8	0	8	Flow	542	228
0	10	0	10	Flow	907	281
0	19	0	19	Flow	1604	532
0	0	1	1	Flow	367	106
0	0	5	5	Flow	514	364
0	0	18	18	Flow	1627	1254
0	0	20	20	Flow	1682	1376
3	3	3	9	CDS	876	352
4	4	4	12	CDS	852	453
6	6	6	18	CDS	1323	655
3	3	3	9	NEH	703	262
5	5	5	15	NEH	866	554

Figure 5.7: Arena Simulation Input



B. Flow Shop

Table 6.1.1.1: Makespan time for different number of easy category jobs in flow shop

Number of jobs	Sequence obtained from NEH algorithm	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	E1	E1	13	13	13	13	13	13	13	13
2	E2, E1	E1, E2	26	20	20	20	20	20	20	20
3	E3, E2, E1	E1, E2, E3	39	29	29	29	29	29	29	29
4	E4, E3, E2, E1	E1, E2, E3, E4	52	38	36	38	38	38	38	38
5	E5, E4, E3, E2, E1	E1, E2, E3, E4, E5	65	47	47	45	47	47	47	47
6	E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6	78	56	56	54	54	56	56	56
7	E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7	91	65	63	65	61	63	65	65
8	E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8	104	74	74	74	72	70	72	74
9	E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9	117	83	83	81	83	79	79	81
10	E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10	130	92	90	90	92	90	86	88
11	E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11	143	101	101	101	99	101	97	95
12	E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12	156	110	110	110	106	110	108	104
13	E13, E12, 11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13	169	119	117	117	117	117	119	115
14	E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14	182	128	128	126	128	124	128	126
15	E15, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15	195	137	137	137	137	133	135	137
16	E16, E15, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16	208	146	144	146	144	144	142	146

Table 6.1.1.1 Continued:

Number of jobs	Sequence obtained from NEH algorithm	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
17	E17, E16, E15, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17	221	155	155	153	151	155	149	153
18	E18, 17, 16, E15, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18	234	164	164	162	162	164	160	160
19	E19, E18, E17, E16, E15, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19	247	173	171	173	173	171	171	167
20	E20, E19, E18, E17, E16, E15, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20	260	182	182	182	182	178	182	176

Table 6.1.2.1: Makespan time for different number of medium category jobs in flow shop

Number of jobs	Sequence obtained from NEH algorithm	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	M1	M1	41	41	41	41	41	41	41	41
2	M2, M1	M1, M2	82	66	66	66	66	66	66	66
3	M3, M2, M1	M1, M2, M3	123	96	93	93	93	93	93	93
4	M4, M3, M2, M1	M1, M2, M3, M4	164	130	121	120	120	120	120	120
5	M5, M4, M3, M2, M1	M1, M2, M3, M4, M5	205	160	150	148	147	147	147	147
6	M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6	246	194	184	175	175	174	174	174
7	M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7	287	224	212	204	200	202	201	201
8	M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8	328	258	241	238	229	227	229	228
9	M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9	369	288	275	266	258	254	254	256
10	M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10	410	322	303	293	292	283	279	281
11	M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11	451	352	332	322	320	312	308	306
12	M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12,	492	386	366	356	345	346	337	333
13	M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13,	533	416	394	384	374	374	366	362
14	M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14,	574	450	423	411	403	399	400	391

Table 6.1.2.1 Continued:

Number of jobs	Sequence obtained from NEH algorithm	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
15	M15, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15,	615	480	457	440	437	426	428	420
16	M16, M15, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16,	656	514	485	474	465	455	453	454
17	M17, ,16, M15, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17	697	544	514	502	490	484	478	482
18	M18, M17, M16, M15, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17, M18	738	578	548	529	519	518	507	507
19	M19, M18, M17, M16, M15, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17, M18, M19	779	608	576	558	548	546	536	532
20	M20, M19, M18, M17, M16, M15, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1	M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17, M18, M19, M20	820	642	605	592	582	571	565	559

Table 6.1.3.1: Makespan time for different number of complex category jobs in flow shop

Number Of jobs	Sequence obtained from NEH algorithm	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	C1	C1	106	106	106	106	106	106	106	106
2	C2, C1	C1, C2	212	169	169	169	169	169	169	169
3	C3, C2, C1	C1, C2, C3	318	264	234	234	234	234	234	234
4	C4, C3, C2, C1	C1, C2, C3, C4	424	336	317	299	299	299	299	299
5	C5, C4, C3, C2, C1	C1, C2, C3, C4, C5	530	431	394	382	364	364	364	364
6	C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6	636	503	466	449	447	429	429	429
7	C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7	742	598	549	524	502	512	494	494
8	C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8	848	670	626	596	579	567	577	559
9	C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9	954	765	698	679	654	634	632	642
10	C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10	1060	837	781	746	726	709	687	697
11	C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11	1166	932	858	821	809	784	764	752
12	C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12	1272	1004	930	893	864	856	839	819
13	C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13	1378	1099	1013	976	941	939	914	894
14	C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14	1484	1171	1090	1043	1016	994	986	969
15	C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15	1590	1266	1162	1118	1088	1061	1069	1044
16	C16, C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16	1696	1338	1245	1190	1171	1136	1124	1116

Table 6.1.3.1 Continued:

Number Of jobs	Sequence obtained from NEH algorithm	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
17	C17, C16, C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17	1802	1433	1322	1273	1226	1211	1179	1199
18	C18, C17, C16, C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18	1908	1505	1394	1340	1303	1283	1256	1254
19	C19, C18, C17, C16, C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19	2014	1600	1477	1415	1378	1366	1331	1309
20	C20, C19, C18, C17, C16, C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20	2120	1672	1554	1487	1450	1421	1406	1376

Table 6.1.4.2: Makespan time for different algorithms for different category of flow shop

Number of Jobs	1 Pallet						2 Pallets					
Job Category	Easy		Medium		Complex		Easy		Medium		Complex	
Algorithms	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	13	13	41	41	106	106	13	13	41	41	106	106
2	26	26	82	82	212	212	20	20	66	66	169	169
3	39	39	123	123	318	318	29	29	96	96	264	264
4	52	52	164	164	424	424	38	38	130	130	336	336
5	65	65	205	205	530	530	47	47	160	160	431	431
6	78	78	246	246	636	636	56	56	194	194	503	503
7	91	91	287	287	742	742	65	65	224	224	598	598
8	104	104	328	328	848	848	74	74	258	258	670	670
9	117	117	369	369	954	954	83	83	288	288	765	765
10	130	130	410	410	1060	1060	92	92	322	322	837	837
11	143	143	451	451	1166	1166	101	101	352	352	932	932
12	156	156	492	492	1272	1272	110	110	386	386	1004	1004
13	169	169	533	533	1378	1378	119	119	416	416	1099	1099
14	182	182	574	574	1484	1484	128	128	450	450	1171	1171
15	195	195	615	615	1590	1590	137	137	480	480	1266	1266
16	208	208	656	656	1696	1696	146	146	514	514	1338	1338
17	221	221	697	697	1802	1802	155	155	544	544	1433	1433
18	234	234	738	738	1908	1908	164	164	578	578	1505	1505
19	247	247	779	779	2014	2014	173	173	608	608	1600	1600
20	260	260	820	820	2120	2120	182	182	642	642	1672	1672

Table 6.1.4.2 Continued:

3 Pallets						4 Pallets						5 Pallets					
Easy		Medium		Complex		Easy		Medium		Complex		Easy		Medium		Complex	
NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
13	13	41	41	106	106	13	13	41	41	106	106	13	13	41	41	106	106
20	20	66	66	169	169	20	20	66	66	169	169	20	20	66	66	169	169
29	29	93	93	234	234	29	29	93	93	234	234	29	29	93	93	234	234
36	36	121	121	317	317	38	38	120	120	299	299	38	38	120	120	299	299
47	47	150	150	394	394	45	45	148	148	382	382	47	47	147	147	364	364
56	56	184	184	466	466	54	54	175	175	449	449	54	54	175	175	447	447
63	63	212	212	549	549	65	65	204	204	524	524	61	61	200	200	502	502
74	74	241	241	626	626	74	74	238	238	596	596	72	72	229	229	579	579
83	83	275	275	698	698	81	81	266	266	679	679	83	83	258	258	654	654
90	90	303	303	781	781	90	90	293	293	746	746	92	92	292	292	726	726
101	101	332	332	858	858	101	101	322	322	821	821	99	99	320	320	809	809
110	110	366	366	930	930	110	110	356	356	893	893	106	106	345	345	864	864
117	117	394	394	1013	1013	117	117	384	384	976	976	117	117	374	374	941	941
128	128	423	423	1090	1090	126	126	411	411	1043	1043	128	128	403	403	1016	1016
137	137	457	457	1162	1162	137	137	440	440	1118	1118	137	137	437	437	1088	1088
144	144	485	485	1245	1245	146	146	474	474	1190	1190	144	144	465	465	1171	1171
155	155	514	514	1322	1322	153	153	502	502	1273	1273	151	151	490	490	1226	1226
164	164	548	548	1394	1394	162	162	529	529	1340	1340	162	162	519	519	1303	1303
171	171	576	576	1477	1477	173	173	558	558	1415	1415	173	173	548	548	1378	1378
182	182	605	605	1554	1554	182	182	592	592	1487	1487	182	182	582	582	1450	1450

Table 6.1.4.2 Continued:

6 Pallets						7 Pallets						8 Pallets					
Easy		Medium		Complex		Easy		Medium		Complex		Easy		Medium		Complex	
NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
13	13	41	41	106	106	13	13	41	41	106	106	13	13	41	41	106	106
20	20	66	66	169	169	20	20	66	66	169	169	20	20	66	66	169	169
29	29	93	93	234	234	29	29	93	93	234	234	29	29	93	93	234	234
38	38	120	120	299	299	38	38	120	120	299	299	38	38	120	120	299	299
47	47	147	147	364	364	47	47	147	147	364	364	47	47	147	147	364	364
56	56	174	174	429	429	56	56	174	174	429	429	56	56	174	174	429	429
63	63	202	202	512	512	65	65	201	201	494	494	65	65	201	201	494	494
70	70	227	227	567	567	72	72	229	229	577	577	74	74	228	228	559	559
79	79	254	254	634	634	79	79	254	254	632	632	81	81	256	256	642	642
90	90	283	283	709	709	86	86	279	279	687	687	88	88	281	281	697	697
101	101	312	312	784	784	97	97	308	308	764	764	95	95	306	306	752	752
110	110	346	346	856	856	108	108	337	337	839	839	104	104	333	333	819	819
117	117	374	374	939	939	119	119	366	366	914	914	115	115	362	362	894	894
124	124	399	399	994	994	128	128	400	400	986	986	126	126	391	391	969	969
133	133	426	426	1061	1061	135	135	428	428	1069	1069	137	137	420	420	1044	1044
144	144	455	455	1136	1136	142	142	453	453	1124	1124	146	146	454	454	1116	1116
155	155	484	484	1211	1211	149	149	478	478	1179	1179	153	153	482	482	1199	1199
164	164	518	518	1283	1283	160	160	507	507	1256	1256	160	160	507	507	1254	1254
171	171	546	546	1366	1366	171	171	536	536	1331	1331	167	167	532	532	1309	1309
178	178	571	571	1421	1421	182	182	565	565	1406	1406	176	176	559	559	1376	1376

Table 6.1.4.3.1: Average percentage change in makespan with increasing pallet number for easy jobs

Number of jobs	Percentage change from pallet 1 to 2	Percentage change from pallet 2 to 3	Percentage change from pallet 3 to 4	Percentage change from pallet 4 to 5	Percentage change from pallet 5 to 6	Percentage change from pallet 6 to 7	Percentage change from pallet 7 to 8
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	23.08	0.00	0.00	0.00	0.00	0.00	0.00
3	25.64	0.00	0.00	0.00	0.00	0.00	0.00
4	26.92	5.26	-5.56	0.00	0.00	0.00	0.00
5	27.69	0.00	4.26	-4.44	0.00	0.00	0.00
6	28.21	0.00	3.57	0.00	-3.70	0.00	0.00
7	28.57	3.08	-3.17	6.15	-3.28	-3.17	0.00
8	28.85	0.00	0.00	2.70	2.78	-2.86	-2.78
9	29.06	0.00	2.41	-2.47	4.82	0.00	-2.53
10	29.23	2.17	0.00	-2.22	2.17	4.44	-2.33
11	29.37	0.00	0.00	1.98	-2.02	3.96	2.06
12	29.49	0.00	0.00	3.64	-3.77	1.82	3.70
13	29.59	1.68	0.00	0.00	0.00	-1.71	3.36
14	29.67	0.00	1.56	-1.59	3.13	-3.23	1.56
15	29.74	0.00	0.00	0.00	2.92	-1.50	-1.48
16	29.81	1.37	-1.39	1.37	0.00	1.39	-2.82
17	29.86	0.00	1.29	1.31	-2.65	3.87	-2.68
18	29.91	0.00	1.22	0.00	-1.23	2.44	0.00
19	29.96	1.16	-1.17	0.00	1.16	0.00	2.34
20	30.00	0.00	0.00	0.00	2.20	-2.25	3.30
Average	29.58	0.49	0.30	0.36	0.73	0.49	0.13

Table 6.1.4.3.2: Average percentage change in makespan with increasing pallet number for medium jobs

Number of jobs	Percentage change from pallet 1 to 2	Percentage change from pallet 2 to 3	Percentage change from pallet 3 to 4	Percentage change from pallet 4 to 5	Percentage change from pallet 5 to 6	Percentage change from pallet 6 to 7	Percentage change from pallet 7 to 8
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	19.51	0.00	0.00	0.00	0.00	0.00	0.00
3	21.95	3.13	0.00	0.00	0.00	0.00	0.00
4	20.73	6.92	0.83	0.00	0.00	0.00	0.00
5	21.95	6.25	1.33	0.68	0.00	0.00	0.00
6	21.14	5.15	4.89	0.00	0.57	0.00	0.00
7	21.95	5.36	3.77	1.96	-1.00	0.50	0.00
8	21.34	6.59	1.24	3.78	0.87	-0.88	0.44
9	21.95	4.51	3.27	3.01	1.55	0.00	-0.79
10	21.46	5.90	3.30	0.34	3.08	1.41	-0.72
11	21.95	5.68	3.01	0.62	2.50	1.28	0.65
12	21.54	5.18	2.73	3.09	-0.29	2.60	1.19
13	21.95	5.29	2.54	2.60	0.00	2.14	1.09
14	21.60	6.00	2.84	1.95	0.99	-0.25	2.25
15	21.95	4.79	3.72	0.68	2.52	-0.47	1.87
16	21.65	5.64	2.27	1.90	2.15	0.44	-0.22
17	21.95	5.51	2.33	2.39	1.22	1.24	-0.84
18	21.68	5.19	3.47	1.89	0.19	2.12	0.00
19	21.95	5.26	3.13	1.79	0.36	1.83	0.75
20	21.71	5.76	2.15	1.69	1.89	1.05	1.06
Average	21.79	5.40	2.80	1.89	1.29	0.87	0.67

Table 6.1.4.3.3: Average percentage change in makespan with increasing pallet number for complex jobs

Number of jobs	Percentage change from pallet 1 to 2	Percentage change from pallet 2 to 3	Percentage change from pallet 3 to 4	Percentage change from pallet 4 to 5	Percentage change from pallet 5 to 6	Percentage change from pallet 6 to 7	Percentage change from pallet 7 to 8
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	20.28	0.00	0.00	0.00	0.00	0.00	0.00
3	16.98	11.36	0.00	0.00	0.00	0.00	0.00
4	20.75	5.65	5.68	0.00	0.00	0.00	0.00
5	18.68	8.58	3.05	4.71	0.00	0.00	0.00
6	20.91	7.36	3.65	0.45	4.03	0.00	0.00
7	19.41	8.19	4.55	4.20	-1.99	3.52	0.00
8	20.99	6.57	4.79	2.85	2.07	-1.76	3.12
9	19.81	8.76	2.72	3.68	3.06	0.32	-1.58
10	21.04	6.69	4.48	2.68	2.34	3.10	-1.46
11	20.07	7.94	4.31	1.46	3.09	2.55	1.57
12	21.07	7.37	3.98	3.25	0.93	1.99	2.38
13	20.25	7.83	3.65	3.59	0.21	2.66	2.19
14	21.09	6.92	4.31	2.59	2.17	0.80	1.72
15	20.38	8.21	3.79	2.68	2.48	-0.75	2.34
16	21.11	6.95	4.42	1.60	2.99	1.06	0.71
17	20.48	7.75	3.71	3.69	1.22	2.64	-1.70
18	21.12	7.38	3.87	2.76	1.53	2.10	0.16
19	20.56	7.69	4.20	2.61	0.87	2.56	1.65
20	21.13	7.06	4.31	2.49	2.00	1.06	2.13
Average	20.25	8.03	4.01	2.71	1.89	1.32	1.10

C. Job Shop with Equal Percentage

Table6.2.1.1: Makespan time for different number of all category jobs in job shop following NEH algorithm

Number Of jobs	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
3	C1, M1, E1	160	122	113	113	113	113	113	113
6	C2, C1, M2, M1, E2, E1	320	251	217	199	190	183	183	183
9	C3, C2, C1, M3, M2, M1, E3, E2, E1	480	371	352	316	284	278	269	262
12	C4, C3, C2, C1, M4, M3, M2, M1, E4, E3, E2, E1	640	500	445	453	415	385	365	357
15	C5, C4, C3, C2, C1, M5, M4, M3, M2, M1, E5, E4, E3, E2, E1	800	620	567	519	554	516	484	452
18	C6, C5, C4, C3, C2, C1, M6, M5, M4, M3, M2, M1	960	749	702	650	608	655	617	583
21	C7, C6, C5, C4, C3, C2, C1, M7, M6, M5, M4, M3, M2, M1, E7, E6, E5, E4, E3, E2, E1	1120	869	795	767	724	698	756	718
24	C8, C7, C6, C5, C4, C3, C2, C1, M8, M7, M6, M5, M4, M3, M2, M1, E8, E7, E6, E5, E4, E3, E2, E1	1280	998	917	904	836	800	788	857
27	C9, C8, C7, C6, C5, C4, C3, C2, C1, M9, M8, M7, M6, M5, M4, M3, M2, M1, E9, E8, E7, E6, E5, E4, E3, E2, E1	1440	1118	1052	970	967	931	879	882
30	C10, C9, C8, C7, C6, C5, C4, C3, C2, C1, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	1600	1247	1145	1101	1106	1038	1005	969
33	C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	1760	1367	1267	1218	1160	1169	1117	1081
36	C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	1920	1496	1402	1355	1276	1308	1238	1212

Table 6.2.1.1 Continued:

Number Of jobs	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
39	C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	2080	1616	1495	1421	1388	1351	1371	1307
42	C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	2240	1745	1617	1552	1519	1453	1510	1438
45	C15, C14, C13, C12, C11, C10, C9, C8, C7, C6, C5, C4, C3, C2, C1, M15, M14, M13, M12, M11, M10, M9, M8, M7, M6, M5, M4, M3, M2, M1, E15, E14, E13, E12, E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1	2400	1865	1752	1669	1658	1584	1542	1573

Table 6.2.1.2: Makespan time for different number of all category jobs in job shop following CDS algorithm

Number Of jobs	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
3	E1, M1, C1	160	160	160	160	160	160	160	160
6	E1, E2, M1, M2, C1, C2	320	251	251	251	251	251	251	251
9	E1, E2, E3, M1, M2, M3, C1, C2, C3	480	413	352	352	352	352	352	352
12	E1, E2, E3, E4, M1, M2, M3, M4, C1, C2, C3, C4	640	500	491	453	453	453	453	453
15	E1, E2, E3, E4, E5, M1, M2, M3, M4, M5, C1, C2, C3, C4, C5	800	690	640	575	554	554	554	554
18	E1, E2, E3, E4, E5, E6, M1, M2, M3, M4, M5, M6, C1, C2, C3, C4, C5, C6	960	749	702	704	644	655	655	655
21	E1, E2, E3, E4, E5, E6, E7, M1, M2, M3, M4, M5, M6, M7, C1, C2, C3, C4, C5, C6, C7	1120	967	859	858	813	745	756	756
24	E1, E2, E3, E4, E5, E6, E7, E8, M1, M2, M3, M4, M5, M6, M7, M8, C1, C2, C3, C4, C5, C6, C7, C8	1280	998	1016	904	925	868	846	857
27	E1, E2, E3, E4, E5, E6, E7, E8, E9, M1, M2, M3, M4, M5, M6, M7, M8, M9, C1, C2, C3, C4, C5, C6, C7, C8, C9	1440	1244	1052	1034	1061	998	954	947
30	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10	1600	1247	1227	1171	1106	1126	1108	1019
33	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11	1760	1521	1392	1333	1195	1263	1192	1181
36	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12	1920	1496	1402	1355	1372	1308	1318	1293
39	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13	2080	1798	1595	1493	1492	1389	1465	1414

Table 6.2.1.2 Continued:

Number Of jobs	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
42	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14	2240	1745	1768	1638	1636	1520	1510	1510
45	E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15 M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15	2400	2075	1752	1808	1658	1658	1583	1667

Table 6.2.2.1: Makespan time for different algorithms for mixed category with equal percentage of job shop

Number of Jobs	1 Pallet		2 Pallets		3 Pallets		4 Pallets		5 Pallets		6 Pallets		7 Pallets		8 Pallets	
Algorithms	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
3	160	160	122	160	113	160	113	160	113	160	113	160	113	160	113	160
6	320	320	251	251	217	251	199	251	190	251	183	251	183	251	183	251
9	480	480	371	413	352	352	316	352	284	352	278	352	269	352	262	352
12	640	640	500	500	445	491	453	453	415	453	385	453	365	453	357	453
15	800	800	620	690	567	640	519	575	554	554	516	554	484	554	452	554
18	960	960	749	749	702	702	650	704	608	644	655	655	617	655	583	655
21	1120	1120	869	967	795	859	767	858	724	813	698	745	756	756	718	756
24	1280	1280	998	998	917	1016	904	904	836	925	800	868	788	846	857	857
27	1440	1440	1118	1244	1052	1052	970	1034	967	1061	931	998	879	954	882	947
30	1600	1600	1247	1247	1145	1227	1101	1171	1106	1106	1038	1126	1005	1108	969	1019
33	1760	1760	1367	1521	1267	1392	1218	1333	1160	1195	1169	1263	1117	1192	1081	1181
36	1920	1920	1496	1496	1402	1402	1355	1355	1276	1372	1308	1308	1238	1318	1212	1293
39	2080	2080	1616	1798	1495	1595	1421	1493	1388	1492	1351	1389	1371	1465	1307	1414
42	2240	2240	1745	1745	1617	1768	1552	1638	1519	1636	1453	1520	1510	1510	1438	1510
45	2400	2400	1865	2075	1752	1752	1669	1808	1658	1658	1584	1658	1542	1583	1573	1667

Table6.2.2.2.2.1: Average percentage change in makespan with increasing pallet number for NEH algorithm with equal percentage of job shop

Number of jobs	Percentage change from pallet 1 to 2	Percentage change from pallet 2 to 3	Percentage change from pallet 3 to 4	Percentage change from pallet 4 to 5	Percentage change from pallet 5 to 6	Percentage change from pallet 6 to 7	Percentage change from pallet 7 to 8
9	22.71	5.12	10.23	10.13	2.11	3.24	2.60
12	21.88	11.00	-1.80	8.39	7.23	5.19	2.19
15	22.50	8.55	8.47	-6.74	6.86	6.20	6.61
18	21.98	6.28	7.41	6.46	-7.73	5.80	5.51
21	22.41	8.52	3.52	5.61	3.59	-8.31	5.03
24	22.03	8.12	1.42	7.52	4.31	1.50	-8.76
27	22.36	5.90	7.79	0.31	3.72	5.59	-0.34
30	22.06	8.18	3.84	-0.45	6.15	3.18	3.58
33	22.33	7.32	3.87	4.76	-0.78	4.45	3.22
36	22.08	6.28	3.35	5.83	-2.51	5.35	2.10
39	22.31	7.49	4.95	2.32	2.67	-1.48	4.67
42	22.10	7.34	4.02	2.13	4.34	-3.92	4.77
45	22.29	6.06	4.74	0.66	4.46	2.65	-2.01
Average	22.23	7.40	4.75	3.61	2.65	2.26	2.24

Table 6.2.2.2.2.2: Average percentage change in makespan with increasing pallet number for CDS algorithm with equal percentage of job shop

Number of jobs	Percentage change from pallet 1 to 2	Percentage change from pallet 2 to 3	Percentage change from pallet 3 to 4	Percentage change from pallet 4 to 5	Percentage change from pallet 5 to 6	Percentage change from pallet 6 to 7	Percentage change from pallet 7 to 8
9	13.96	14.77	0.00	0.00	0.00	0.00	0.00
12	21.88	1.80	7.74	0.00	0.00	0.00	0.00
15	13.75	7.25	10.16	3.65	0.00	0.00	0.00
18	21.98	6.28	-0.28	8.52	-1.71	0.00	0.00
21	13.66	11.17	0.12	5.24	8.36	-1.48	0.00
24	22.03	-1.80	11.02	-2.32	6.16	2.53	-1.30
27	13.61	15.43	1.71	-2.61	5.94	4.41	0.73
30	22.06	1.60	4.56	5.55	-1.81	1.60	8.03
33	13.58	8.48	4.24	10.35	-5.69	5.62	0.92
36	22.08	6.28	3.35	-1.25	4.66	-0.76	1.90
39	13.56	11.29	6.39	0.07	6.90	-5.47	3.48
42	22.10	-1.32	7.35	0.12	7.09	0.66	0.00
45	13.54	15.57	-3.20	8.30	0.00	4.52	-5.31
Average	17.52	7.45	4.09	2.74	2.30	0.89	0.65

D. Job Shop with Different Percentage

Table 6.2.2.1.1: Makespan time for different number of combinations of 4 jobs in job shop following NEH algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets
1	1	1	2	25	25	50	C2, C1, M1, E1	266	208	183	176
2	1	2	1	25	50	25	C1, M2, M1, E1	201	152	127	120
3	2	1	1	50	25	25	C1, M1, E2, E1	173	131	120	113

Table 6.2.2.1.2: Makespan time for different number of combinations of 4 jobs in job shop following CDS algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets
1	1	1	2	25	25	50	E1, M1, C1, C2	266	228	228	228
2	1	2	1	25	50	25	E1, M1, M2, C1	201	168	163	163
3	2	1	1	50	25	25	E1, E2, M1, C1	173	149	145	145

Table 6.2.2.2: Makespan time for different algorithms for mixed category with different percentage of job shop for 4 jobs

Series number	Number of jobs			Percentage			1 Pallet		2 Pallets		3 Pallets		4 Pallets	
	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium Percentage	Complex %	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	2	25	25	50	266	266	208	228	183	228	176	228
2	1	2	1	25	50	25	201	201	152	168	127	163	120	163
3	2	1	1	50	25	25	173	173	131	149	120	145	113	145

Table 6.2.2.3: Percentage deviation in makespan with increasing pallet number for combination of 4 jobs

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Percentage change from pallet 1 to 2		Percentage change from pallet 2 to 3		Percentage change from pallet 3 to 4	
							NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	2	25	25	50	21.80	14.29	24.38	16.42	24.28	13.87
2	1	2	1	25	50	25	12.02	0.00	16.45	2.98	8.40	2.68
3	2	1	1	50	25	25	3.83	0.00	5.51	0.00	5.83	0.00

Table 6.2.2.4.1: Makespan time for different number of combinations of 5 jobs in job shop for NEH Algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets
1	1	1	3	20	20	60	C3, C2, C1, M1, E1	372	289	266	248	241
2	1	2	2	20	40	40	C2, C1, M2, M1, E1	307	242	208	190	183
3	2	1	2	40	20	40	C2, C1, M1, E2, E1	279	217	194	183	176
4	1	3	1	20	60	20	C1, M3, M2, M1, E1	242	186	152	134	127
5	2	2	1	40	40	20	C1, M2, M1, E2, E1	214	161	138	127	120
6	3	1	1	60	20	20	C1, M1, E3, E2, E1	186	140	131	120	113

Table 6.2.2.4.2: Makespan time for different number of combinations of 5 jobs in job shop for CDS Algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets
1	1	1	3	20	20	60	E1, M1, C1, C2, C3	372	323	313	293	263
2	1	2	2	20	40	40	E1, M1, M2, C1, C2	307	263	248	255	225
3	2	1	2	40	20	40	E1, E2, M1, C1, C2	279	244	210	237	207
4	1	3	1	20	60	20	E1, M1, M2, M3, C1	242	208	186	190	187
5	2	2	1	40	40	20	E1, E2, M1, M2, C1	214	179	172	172	169
6	3	1	1	60	20	20	E1, E2, E3, M1, C1	186	154	151	142	151

Table 6.2.2.5: Makespan time for different algorithms for mixed category with different percentage of job shop for 5 jobs

Series Number	Number of Jobs			Percentage			1 Pallet		2 Pallets		3 Pallets		4 Pallets		5 Pallets	
	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium Percentage	Complex %	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	3	20	20	60	372	372	289	323	266	313	248	293	241	263
2	1	2	2	20	40	40	307	307	242	263	208	248	190	255	183	225
3	2	1	2	40	20	40	279	279	217	244	194	210	183	237	176	207
4	1	3	1	20	60	20	242	242	186	208	152	186	134	190	127	187
5	2	2	1	40	40	20	214	214	161	179	138	172	127	172	120	169
6	3	1	1	60	20	20	186	186	140	154	131	151	120	142	113	151

Table 6.2.2.6: Percentage deviation in makespan with increasing pallet number for combination of 5 jobs

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Percentage change from pallet 1 to 2		Percentage change from pallet 2 to 3		Percentage change from pallet 3 to 4		Percentage change from pallet 4 to 5	
							NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	3	20	20	60	22.31	13.17	7.96	3.10	6.77	6.39	2.82	10.24
2	1	2	2	20	40	40	21.17	14.33	14.05	5.70	8.65	-2.82	3.68	11.76
3	2	1	2	40	20	40	22.22	12.54	10.60	13.93	5.67	-12.86	3.83	12.66
4	1	3	1	20	60	20	23.14	14.05	18.28	10.58	11.84	-2.15	5.22	1.58
5	2	2	1	40	40	20	24.77	16.36	14.29	3.91	7.97	0.00	5.51	1.74
6	3	1	1	60	20	20	24.73	17.20	6.43	1.95	8.40	5.96	5.83	-6.34

Table 6.2.2.7.1: Makespan time for different number of combinations of 6 jobs in job shop following NEH Algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets
1	1	1	4	16.67	16.67	66.67	C4, C3, C2, C1, M1, E1	478	375	345	329	313	306
2	1	2	3	16.67	33.33	50	C3, C2, C1, M2, M1, E1	413	319	300	271	255	248
3	2	1	3	33.33	16.67	50	C3, C2, C1, M1, E2, E1	385	298	275	257	248	241
4	1	3	2	16.67	50	33.33	C2, C1, M3, M2, M1, E1	348	272	242	213	197	190
5	3	1	2	50	16.67	33.33	C2, C1, M1, E3, E2, E1	292	226	203	192	183	176
6	1	4	1	16.67	66.67	16.67	C1, M4, M3, M2, M1, E1	283	216	186	157	158	151
7	2	3	1	33.33	50	16.67	C1, M3, M2, M1, E2, E1	255	195	161	143	134	127
8	3	2	1	50	33.33	16.67	C1, M2, M3, E3, E2, E1	227	170	147	136	127	120
9	4	1	1	66.67	16.67	16.67	C1, M1, E4, E3, E2, E1	199	149	140	129	120	113

Table 6.2.2.7.2: Makespan time for different number of combinations of 6 jobs in job shop following CDS Algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets
1	1	1	4	16.67	16.67	66.67	E1, M1, C1, C2, C3, C4	478	415	377	350	358	328
2	1	2	3	16.67	33.33	50	E1, M1, M2, C1, C2, C3	413	355	312	312	320	290
3	2	1	3	33.33	16.67	50	E1, E2, M1, C1, C2, C3	385	316	302	302	302	272
4	1	3	2	16.67	50	33.33	E1, M1, M2, M3, C1, C2	348	300	250	247	282	252
5	3	1	2	50	16.67	33.33	E1, E2, E3, M1, C1, C2	292	246	223	219	234	216
6	1	4	1	16.67	66.67	16.67	E1, M1, M2, M3, M4, C1	283	240	220	210	217	214
7	2	3	1	33.33	50	16.67	E1, E2, M1, M2, M3, C1	255	213	202	199	199	196
8	3	2	1	50	33.33	16.67	E1, E2, E3, M1, M2, C1	227	186	185	181	169	178
9	4	1	1	66.67	16.67	16.67	E1, E2, E3, E4, M1, C1	199	167	160	148	148	160

Table 6.2.2.8: Makespan time for different algorithms for mixed category with different percentage of job shop for 6 jobs

Series number	Number of jobs			Percentage			1 Pallet		2 Pallets		3 Pallets		4 Pallets		5 Pallets		6 Pallets	
	Number of Easy jobs	Number of Medium	Number of Complex	Easy %	Medium %	Complex %	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	4	16.67	16.67	66.67	478	478	375	415	345	377	329	350	313	358	306	328
2	1	2	3	16.67	33.33	50	413	413	319	355	300	312	271	312	255	320	248	290
3	2	1	3	33.33	16.67	50	385	385	298	316	275	302	257	302	248	302	241	272
4	1	3	2	16.67	50	33.33	348	348	272	300	242	250	213	247	197	282	190	252
5	3	1	2	50	16.67	33.33	292	292	226	246	203	223	192	219	183	234	176	216
6	1	4	1	16.67	66.67	16.67	283	283	216	240	186	220	157	210	158	217	151	214
7	2	3	1	33.33	50	16.67	255	255	195	213	161	202	143	199	134	199	127	196
8	3	2	1	50	33.33	16.67	227	227	170	186	147	185	136	181	127	169	120	178
9	4	1	1	66.67	16.67	16.67	199	199	149	167	140	160	129	148	120	148	113	160

Table 6.2.2.9: Percentage deviation in makespan with increasing pallet number for combination of 6 jobs

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Percentage change from pallet 1 to 2		Percentage change from pallet 2 to 3		Percentage change from pallet 3 to 4		Percentage change from pallet 4 to 5		Percentage change from pallet 5 to 6	
							NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	4	16.67	16.67	66.67	21.55	13.18	8.00	9.16	4.64	7.16	4.86	-2.29	2.24	8.38
2	1	2	3	16.67	33.33	50	22.76	14.04	5.96	12.11	9.67	0.00	5.90	-2.56	2.75	9.38
3	2	1	3	33.33	16.67	50	22.60	17.92	7.72	4.43	6.55	0.00	3.50	0.00	2.82	9.93
4	1	3	2	16.67	50	33.33	21.84	13.79	11.03	16.67	11.98	1.20	7.51	-14.17	3.55	10.64
5	3	1	2	50	16.67	33.33	22.60	15.75	10.18	9.35	5.42	1.79	4.69	-6.85	3.83	7.69
6	1	4	1	16.67	66.67	16.67	23.67	15.19	13.89	8.33	15.59	4.55	-0.64	-3.33	4.43	1.38
7	2	3	1	33.33	50	16.67	23.53	16.47	17.44	5.16	11.18	1.49	6.29	0.00	5.22	1.51
8	3	2	1	50	33.33	16.67	25.11	18.06	13.53	0.54	7.48	2.16	6.62	6.63	5.51	-5.33
9	4	1	1	66.67	16.67	16.67	25.13	16.08	6.04	4.19	7.86	7.50	6.98	0.00	5.83	-8.11

Table 6.2.2.10.1: Makespan time for different number of combinations of 7 jobs in job shop following NEH algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets
1	1	1	5	14.29	14.29	71.43	C5, C4, C3, C2, C1, M1, E1	584	456	415	401	392	378	371
2	1	2	4	14.29	28.57	57.14	C4, C3, C2, C1, M2, M1, E1	519	409	359	358	334	320	313
3	2	1	4	28.57	14.29	57.14	C4, C3, C2, C1, M1, E2, E1	491	384	352	333	320	313	306
4	1	3	3	14.29	42.86	42.86	C3, C2, C1, M3, M2, M1, E1	454	353	332	300	276	262	255
5	2	2	3	28.57	28.57	42.86	C3, C2, C1, M2, M1, E2, E1	426	328	307	275	262	255	248
6	3	1	3	42.86	14.29	42.86	C3, C2, C1, M1, E3, E2, E1	398	307	282	268	255	248	241
7	1	4	2	14.29	57.14	28.57	C2, C1, M4, M3, M2, M1, E1	389	306	274	242	218	204	197
8	2	3	2	28.57	42.86	28.57	C2, C1, M3, M2, M1, E2, E1	361	281	249	217	204	197	190
9	3	2	2	42.86	28.57	28.57	C2, C1, M2, M1, E3, E2, E1	333	260	224	210	197	190	183
10	1	5	1	14.29	71.43	14.29	C1, M5, M4, M3, M2, M1, E1	324	250	218	186	179	183	176
11	4	1	2	57.14	14.29	28.57	C2, C1, M1, E4, E3, E2, E1	305	235	210	203	190	183	176
12	2	4	1	28.57	57.14	14.29	C1, M4, M3, M2, M1, E2, E1	296	225	193	161	165	158	151
13	3	3	1	42.86	42.86	14.29	C1, M3, M2, M1, E3, E2, E1	268	204	168	154	141	134	127
14	4	2	1	57.14	28.57	14.29	C1, M2, M1, E4, E3, E2, E1	240	179	154	147	134	127	120
15	5	1	1	71.43	14.29	14.29	C1, M1, E5, E4, E3, E2, E1	212	158	147	140	127	120	113

Table 6.2.2.10.2: Makespan time for different number of combinations of 7 jobs in job shop following CDS algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets
1	1	1	5	14.29	14.29	71.43	E1, M1, C1, C2, C3, C4, C5	584	510	472	435	415	423	393
2	1	2	4	14.29	28.57	57.14	E1, M1, M2, C1, C2, C3, C4	519	450	407	397	377	385	355
3	2	1	4	28.57	14.29	57.14	E1, E2, M1, C1, C2, C3, C4	491	411	397	387	367	367	337
4	1	3	3	14.29	42.86	42.86	E1, M1, M2, M3, C1, C2, C3	454	395	345	332	339	347	317
5	2	2	3	28.57	28.57	42.86	E1, E2, M1, M2, C1, C2, C3	426	346	359	322	329	329	299
6	3	1	3	42.86	14.29	42.86	E1, E2, E3, M1, C1, C2, C3	398	341	306	284	311	299	281
7	1	4	2	14.29	57.14	28.57	E1, M1, M2, M3, M4, C1, C2	389	335	315	271	274	309	279
8	2	3	2	28.57	42.86	28.57	E1, E2, M1, M2, M3, C1, C2	361	308	297	284	264	291	261
9	3	2	2	42.86	28.57	28.57	E1, E2, E3, M1, M2, C1, C2	333	281	268	246	246	261	243
10	1	5	1	14.29	71.43	14.29	E1, M1, M2, M3, M4, M5, C1	324	280	260	238	237	244	241
11	4	1	2	57.14	14.29	28.57	E1, E2, E3, E4, M1, C1, C2	305	262	255	225	218	215	225
12	2	4	1	28.57	57.14	14.29	E1, E2, M1, M2, M3, M4, C1	296	243	242	220	226	226	223
13	3	3	1	42.86	42.86	14.29	E1, E2, E3, M1, M2, M3, C1	268	226	203	208	208	196	205
14	4	2	1	57.14	28.57	14.29	E1, E2, E3, E4, M1, M2, C1	240	197	190	187	178	175	187
15	5	1	1	71.43	14.29	14.29	E1, E2, E3, E4, E5, M1, C1	212	172	172	169	155	157	169

Table 6.2.2.11: Makespan time for different algorithms for mixed category with different percentage of job shop for 7 jobs

Series Number	Number of jobs			Percentage			1 Pallet		2 Pallets		3 Pallets		4 Pallets		5 Pallets		6 Pallets		7 Pallets	
	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	5	14.29	14.29	71.43	584	584	456	510	415	472	401	435	392	415	378	423	371	393
2	1	2	4	14.29	28.57	57.14	519	519	409	450	359	407	358	397	334	377	320	385	313	355
3	2	1	4	28.57	14.29	57.14	491	491	384	411	352	397	333	387	320	367	313	367	306	337
4	1	3	3	14.29	42.86	42.86	454	454	353	395	332	345	300	332	276	339	262	347	255	317
5	2	2	3	28.57	28.57	42.86	426	426	328	346	307	359	275	322	262	329	255	329	248	299
6	3	1	3	42.86	14.29	42.86	398	398	307	341	282	306	268	284	255	311	248	299	241	281
7	1	4	2	14.29	57.14	28.57	389	389	306	335	274	315	242	271	218	274	204	309	197	279
8	2	3	2	28.57	42.86	28.57	361	361	281	308	249	297	217	284	204	264	197	291	190	261
9	3	2	2	42.86	28.57	28.57	333	333	260	281	224	268	210	246	197	246	190	261	183	243
10	1	5	1	14.29	71.43	14.29	324	324	250	280	218	260	186	238	179	237	183	244	176	241
11	4	1	2	57.14	14.29	28.57	305	305	235	262	210	255	203	225	190	218	183	215	176	225
12	2	4	1	28.57	57.14	14.29	296	296	225	243	193	242	161	220	165	226	158	226	151	223
13	3	3	1	42.86	42.86	14.29	268	268	204	226	168	203	154	208	141	208	134	196	127	205
14	4	2	1	57.14	28.57	14.29	240	240	179	197	154	190	147	187	134	178	127	175	120	187
15	5	1	1	71.43	14.29	14.29	212	212	158	172	147	172	140	169	127	155	120	157	113	169

Table 6.2.2.12: Percentage deviation in makespan with increasing pallet number for combination of 7 jobs

Series number	Number of Easy	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Percentage change from pallet 1 to 2		Percentage change from pallet 2 to 3		Percentage change from pallet 3 to 4		Percentage change from pallet 4 to 5		Percentage change from pallet 5 to 6		Percentage change from pallet 6 to 7	
							NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	5	14.29	14.29	71.43	21.92	12.67	8.99	7.45	3.37	7.84	2.24	4.60	3.57	-1.93	1.85	7.09
2	1	2	4	14.29	28.57	57.14	21.19	13.29	12.22	9.56	0.28	2.46	6.70	5.04	4.19	-2.12	2.19	7.79
3	2	1	4	28.57	14.29	57.14	21.79	16.29	8.33	3.41	5.40	2.52	3.90	5.17	2.19	0.00	2.24	8.17
4	1	3	3	14.29	42.86	42.86	22.25	13.00	5.95	12.66	9.64	3.77	8.00	-2.11	5.07	-2.36	2.67	8.65
5	2	2	3	28.57	28.57	42.86	23.00	18.78	6.40	-3.76	10.42	10.31	4.73	-2.17	2.67	0.00	2.75	9.12
6	3	1	3	42.86	14.29	42.86	22.86	14.32	8.14	10.26	4.96	7.19	4.85	-9.51	2.75	3.86	2.82	6.02
7	1	4	2	14.29	57.14	28.57	21.34	13.88	10.46	5.97	11.68	13.97	9.92	-1.11	6.42	-12.77	3.43	9.71
8	2	3	2	28.57	42.86	28.57	22.16	14.68	11.39	3.57	12.85	4.38	5.99	7.04	3.43	-10.23	3.55	10.31
9	3	2	2	42.86	28.57	28.57	21.92	15.62	13.85	4.63	6.25	8.21	6.19	0.00	3.55	-6.10	3.68	6.90
10	1	5	1	14.29	71.43	14.29	22.84	13.58	12.80	7.14	14.68	8.46	3.76	0.42	-2.23	-2.95	3.83	1.23
11	4	1	2	57.14	14.29	28.57	22.95	14.10	10.64	2.67	3.33	11.76	6.40	3.11	3.68	1.38	3.83	-4.65
12	2	4	1	28.57	57.14	14.29	23.99	17.91	14.22	0.41	16.58	9.09	-2.48	-2.73	4.24	0.00	4.43	1.33
13	3	3	1	42.86	42.86	14.29	23.88	15.67	17.65	10.18	8.33	-2.46	8.44	0.00	4.96	5.77	5.22	-4.59
14	4	2	1	57.14	28.57	14.29	25.42	17.92	13.97	3.55	4.55	1.58	8.84	4.81	5.22	1.69	5.51	-6.86
15	5	1	1	71.43	14.29	14.29	25.47	18.87	6.96	0.00	4.76	1.74	9.29	8.28	5.51	-1.29	5.83	-7.64

Table 6.2.2.13.1: Makespan time for different number of combinations of 8 jobs in job shop following NEH Algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	1	1	6	12.5	12.5	75	C6, C5, C4, C3, C2, C1, M1, E1	690	542	498	473	462	457	443	436
2	1	2	5	12.5	25	62.5	C5, C4, C3, C2, C1, M2, M1, E1	625	486	440	417	421	399	385	378
3	2	1	5	25	12.5	62.5	C5, C4, C3, C2, C1, M1, E2, E1	597	465	426	410	396	385	378	371
4	1	3	4	12.5	37.5	50	C4, C3, C2, C1, M3, M2, M1, E1	560	439	384	392	363	341	327	320
5	2	2	4	25	25	50	C4, C3, C2, C1, M2, M1, E2, E1	532	418	370	367	338	327	320	313
6	3	1	4	37.5	12.5	50	C4, C3, C2, C1, M1, E3, E2, E1	504	393	363	342	331	320	313	306
7	1	4	3	12.5	50	37.5	C3, C2, C1, M4, M3, M2, M1, E1	495	383	357	334	305	283	269	262
8	2	3	3	25	37.5	37.5	C3, C2, C1, M3, M2, M1, E2, E1	467	362	343	309	280	269	262	255
9	3	2	3	37.5	25	37.5	C3, C2, C1, M2, M1, E3, E2, E1	439	337	318	284	273	262	255	248

Table 6.2.2.13.1 Continued:

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
10	1	5	2	12.5	62.5	25	C2, C1, M5, M4, M3, M2, M1, E1	430	336	299	276	247	225	216	209
11	4	1	3	50	12.5	37.5	C3, C2, C1, M1, E4, E3, E2, E1	411	316	293	277	266	255	248	241
12	2	4	2	25	50	25	C2, C1, M4, M3, M2, M1, E2, E1	402	315	285	251	222	211	204	197
13	3	3	2	37.5	37.5	25	C2, C1, M3, M2, M1, E3, E2, E1	374	290	260	226	215	204	197	190
14	1	6	1	12.5	75	12.5	C1, M6, M5, M4, M3, M2, M1, E1	365	280	243	220	208	204	208	203
15	4	2	2	50	25	25	C2, C1, M2, M1, E4, E3, E2, E1	346	269	235	219	208	197	190	183
16	2	5	1	25	62.5	12.5	C1, M5, M4, M3, M2, M1	337	259	229	195	183	190	183	178
17	5	1	2	62.5	12.5	25	C2, C1, M1, E5, E4, E3, E2, E1	318	244	221	212	201	190	183	176
18	3	4	1	37.5	50	12.5	C1, M4, M3, M2, M1, E3, E2, E1	309	234	204	170	176	165	158	153

Table 6.2.2.13.1 Continued:

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from NEH algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
19	4	3	1	50	37.5	12.5	C1, M3, M2, M1, E4, E3, E2, E1	281	213	179	163	152	141	134	129
20	5	2	1	62.5	25	12.5	C1, M2, M1, E5, E4, E3, E2, E1	253	188	165	156	145	134	127	122
21	6	1	1	75	12.5	12.5	C1, M1, E6, E5, E4, E3, E2, E1	225	167	158	149	138	127	120	115

Table 6.2.2.13.2: Makespan time for different number of combinations of 8 jobs in job shop following CDS Algorithm

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
1	1	1	6	12.5	12.5	75	E1, M1, C1, C2, C3, C4, C5, C6	690	602	557	499	472	480	488	458
2	1	2	5	12.5	25	62.5	E1, M1, M2, C1, C2, C3, C4, C5	625	542	492	461	434	442	450	420
3	2	1	5	25	12.5	62.5	E1, E2, M1, C1, C2, C3, C4, C5	597	483	462	451	424	432	432	402
4	1	3	4	12.5	37.5	50	E1, M1, M2, M3, C1, C2, C3, C4	560	487	430	396	396	404	412	382
5	2	2	4	25	25	50	E1, E2, M1, M2, C1, C2, C3, C4	532	418	424	386	386	394	394	364
6	3	1	4	37.5	12.5	50	E1, E2, E3, M1, C1, C2, C3, C4	504	433	383	376	376	376	364	346
7	1	4	3	12.5	50	37.5	E1, M1, M2, M3, M4, C1, C2, C3	495	427	400	336	331	366	374	344
8	2	3	3	25	37.5	37.5	E1, E2, M1, M2, M3, C1, C2, C3	467	380	362	348	321	356	356	326

Table 6.2.2.13.2 Continued:

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
9	3	2	3	37.5	25	37.5	E1, E2, E3, M1, M1, C1, C2, C3	439	373	345	338	311	338	326	308
10	1	5	2	12.5	62.5	25	E1, M1, M2, M3, M4, M5, C1, C2	430	372	345	302	292	301	336	306
11	4	1	3	50	12.5	37.5	E1, E2, E3, E4, M1, C1, C2, C3	411	334	340	297	293	310	280	290
12	2	4	2	25	50	25	E1, E2, M1, M2, M3, M4, C1, C2	402	315	307	284	283	291	318	288
13	3	3	2	37.5	37.5	25	E1, E2, E3, M1, M2, M3, C1, C2	374	318	280	300	273	273	288	270
14	1	6	1	12.5	75	12.5	E1, M1, M2, M3, M4, M5, M6, C1	365	312	283	272	262	264	271	268
15	4	2	2	50	25	25	E1, E2, E3, E4, M1, M2, C1, C2	337	277	269	254	244	253	253	250
16	2	5	1	25	62.5	12.5	E1, E2, M1, M2, M3, M4, M5, C1	346	269	275	259	255	245	242	252
17	5	1	2	62.5	12.5	25	E1, E2, E3, E4, E5, M1, C1, C2	318	264	237	234	224	215	224	234
18	3	4	1	37.5	50	12.5	E1, E2, E3, M1, M2, M3, M4, C1	309	258	242	236	235	235	223	232

Table 6.2.2.13.2 Continued:

Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Sequence obtained from CDS algorithm	1 Pallet	2 Pallets	3 Pallets	4 Pallets	5 Pallets	6 Pallets	7 Pallets	8 Pallets
19	4	3	1	50	37.5	12.5	E1, E2, E3, E4, M1, M2, M3, C1	281	231	213	221	217	205	202	214
20	5	2	1	62.5	25	12.5	E1, E2, E3, E4, E5, M1, M2, C1	253	204	199	196	184	185	184	196
21	6	1	1	75	12.5	12.5	E1, E2, E3, E4, E5, E6, M1, C1	225	185	178	178	171	164	166	178

Table 4.2.2.14: Makespan time for different algorithms for mixed category with different percentage of job shop for 8 jobs

Series number	Number of Jobs			Percentage			1 Pallet		2 Pallets		3 Pallets		4 Pallets		5 Pallets		6 Pallets		7 Pallets		8 Pallets	
	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	6	12.5	12.5	75	690	690	542	602	498	557	473	499	462	472	457	480	443	488	436	458
2	1	2	5	12.5	25	62.5	625	625	486	542	440	492	417	461	421	434	399	442	385	450	378	420
3	2	1	5	25	12.5	62.5	597	597	465	483	426	462	410	451	396	424	385	432	378	432	371	402
4	1	3	4	12.5	37.5	50	560	560	439	487	384	430	392	396	363	396	341	404	327	412	320	382
5	2	2	4	25	25	50	532	532	418	418	370	424	367	386	338	386	327	394	320	394	313	364
6	3	1	4	37.5	12.5	50	504	504	393	433	363	383	342	376	331	376	320	376	313	364	306	346
7	1	4	3	12.5	50	37.5	495	495	383	427	357	400	334	336	305	331	283	366	269	374	262	344
8	2	3	3	25	37.5	37.5	467	467	362	380	343	362	309	348	280	321	269	356	262	356	255	326
9	3	2	3	37.5	25	37.5	439	439	337	373	318	345	284	338	273	311	262	338	255	326	248	308
10	1	5	2	12.5	62.5	25	430	430	336	372	299	345	276	302	247	292	225	301	216	336	209	306
11	4	1	3	50	12.5	37.5	411	411	316	334	293	340	277	297	266	293	255	310	248	280	241	290
12	2	4	2	25	50	25	402	402	315	315	285	307	251	284	222	283	211	291	204	318	197	288

Table 5.2.2.14 Continued:

	Number of Jobs			Percentage			1 Pallet		2 Pallets		3 Pallets		4 Pallets		5 Pallets		6 Pallets		7 Pallets		8 Pallets	
Series number	Number of Easy jobs	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
13	3	3	2	37.5	37.5	25	374	374	290	318	260	280	226	300	215	273	204	273	197	288	190	270
14	1	6	1	12.5	75	12.5	365	365	280	312	243	283	220	272	208	262	204	264	208	271	203	268
15	2	5	1	25	62.5	12.5	337	337	259	277	229	269	195	254	183	244	190	253	183	253	178	250
16	4	2	2	50	25	25	346	346	269	269	235	275	219	259	208	255	197	245	190	242	183	252
17	5	1	2	62.5	12.5	25	318	318	244	264	221	237	212	234	201	224	190	215	183	224	176	234
18	3	4	1	37.5	50	12.5	309	309	234	258	204	242	170	236	176	235	165	235	158	223	153	232
19	4	3	1	50	37.5	12.5	281	281	213	231	179	213	163	221	152	217	141	205	134	202	129	214
20	5	2	1	62.5	25	12.5	253	253	188	204	165	199	156	196	145	184	134	185	127	184	122	196
21	6	1	1	75	12.5	12.5	225	225	167	185	158	178	149	178	138	171	127	164	120	166	115	178

Table 6.2.2.15: Percentage deviation in makespan with increasing pallet number for combination of 8 jobs

Series number	Number of Easy	Number of Medium jobs	Number of Complex jobs	Easy %	Medium %	Complex %	Percentage change from pallet 1 to 2		Percentage change from pallet 2 to 3		Percentage change from pallet 3 to 4		Percentage change from pallet 4 to 5		Percentage change from pallet 5 to 6		Percentage change from pallet 6 to 7		Percentage change from pallet 7 to 8	
							NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS	NEH	CDS
1	1	1	6	12.5	12.5	75	21.45	12.75	8.12	7.48	5.02	10.41	2.33	5.41	1.08	-1.69	3.06	-1.67	1.58	6.15
2	1	2	5	12.5	25	62.5	22.24	13.28	9.47	9.23	5.23	6.30	-0.96	5.86	5.23	-1.84	3.51	-1.81	1.82	6.67
3	2	1	5	25	12.5	62.5	22.11	19.10	8.39	4.35	3.76	2.38	3.41	5.99	2.78	-1.89	1.82	0.00	1.85	6.94
4	1	3	4	12.5	37.5	50	21.61	13.04	12.53	11.70	-2.08	7.91	7.40	0.00	6.06	-2.02	4.11	-1.98	2.14	7.28
5	2	2	4	25	25	50	21.43	21.43	11.48	-1.44	0.81	8.96	7.90	0.00	3.25	-2.07	2.14	0.00	2.19	7.61
6	3	1	4	37.5	12.5	50	22.02	14.09	7.63	11.55	5.79	1.83	3.22	0.00	3.32	0.00	2.19	3.19	2.24	4.95
7	1	4	3	12.5	50	37.5	22.63	13.74	6.79	6.32	6.44	16.00	8.68	1.49	7.21	-10.57	4.95	-2.19	2.60	8.02
8	2	3	3	25	37.5	37.5	22.48	18.63	5.25	4.74	9.91	3.87	9.39	7.76	3.93	-10.90	2.60	0.00	2.67	8.43
9	3	2	3	37.5	25	37.5	23.23	15.03	5.64	7.51	10.69	2.03	3.87	7.99	4.03	-8.68	2.67	3.55	2.75	5.52
10	1	5	2	12.5	62.5	25	21.86	13.49	11.01	7.26	7.69	12.46	10.51	3.31	8.91	-3.08	4.00	-11.63	3.24	8.93
11	4	1	3	50	12.5	37.5	23.11	18.73	7.28	-1.80	5.46	12.65	3.97	1.35	4.14	-5.80	2.75	9.68	2.82	-3.57
12	2	4	2	25	50	25	21.64	21.64	9.52	2.54	11.93	7.49	11.55	0.35	4.95	-2.83	3.32	-9.28	3.43	9.43
13	3	3	2	37.5	37.5	25	22.46	14.97	10.34	11.95	13.08	-7.14	4.87	9.00	5.12	0.00	3.43	-5.49	3.55	6.25
14	1	6	1	12.5	75	12.5	23.29	14.52	13.21	9.29	9.47	3.89	5.45	3.68	1.92	-0.76	-1.96	-2.65	2.40	1.11
15	4	2	2	50	25	25	22.25	17.80	12.64	2.89	6.81	5.58	5.02	3.94	5.29	-3.69	3.55	0.00	3.68	1.19
16	2	5	1	25	62.5	12.5	23.15	22.25	11.58	-2.23	14.85	5.82	6.15	1.54	-3.83	3.92	3.68	1.22	2.73	-4.13
17	5	1	2	62.5	12.5	25	23.27	16.98	9.43	10.23	4.07	1.27	5.19	4.27	5.47	4.02	3.68	-4.19	3.83	-4.46
18	3	4	1	37.5	50	12.5	24.27	16.50	12.82	6.20	16.67	2.48	-3.53	0.42	6.25	0.00	4.24	5.11	3.16	-4.04
19	4	3	1	50	37.5	12.5	24.20	17.79	15.96	7.79	8.94	-3.76	6.75	1.81	7.24	5.53	4.96	1.46	3.73	-5.94
20	5	2	1	62.5	25	12.5	25.69	19.37	12.23	2.45	5.45	1.51	7.05	6.12	7.59	-0.54	5.22	0.54	3.94	-6.52
21	6	1	1	75	12.5	12.5	25.78	17.78	5.39	3.78	5.70	0.00	7.38	3.93	7.97	4.09	5.51	-1.22	4.17	-7.23