AN ABSTRACT OF THE THESIS OF

Shardul Khandke for the degree of Master of Science in Industrial Engineering presented on September 11, 2019.

Title: Work Overloads Approximation on a Moving Assembly Line Workstation using Markov Chain.

Abstract approved:

____

David Kim

Moving mixed-model assembly lines are used by many companies to assemble multiple model types of a particular product. A moving assembly line consists of a material movement system that moves jobs at a constant velocity across a series of workstations. The variation in the work content of jobs at workstations may result in workstation inefficiencies if the jobs are processed through the line in random order. This gives rise to the job sequencing problem, which is to determine a job launching order, (i.e., the job sequence) that minimizes operational inefficiencies such as work overloads. Work overloads occur when a job cannot be processed within the limits of a workstation.

When determining the workstation parameters on a moving assembly line they should ideally be set so that the workstation is the minimum length for a given assembly line throughput, and zero operational inefficiencies occur when job sequencing is performed. Estimating the operational inefficiencies without having to solve a

sequencing problem is the topic of this research. This research establishes a discrete state Markov chain model to estimate the expected number, and the probability distribution of work overloads for a set of jobs launched in random order. The models have been tested on a variety of different job sets, and the results indicate that the model can accurately estimate the probability of work overloads as a function of workstation parameters, and whether job sequencing reduces work overloads to zero.

Work Overloads Approximation on a Moving Assembly Line Workstation using Markov Chain

by
Shardul Khandke

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 11, 2019
Commencement June 2020

Master of Science thesis of Shardul Khandke presented on September 11, 2019

APPROVED:

Major Professor, representing Industrial Engineering

Head of the School of Mechanical, Industrial and Manufacturing Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Shardul Khandke, Author

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. David S. Kim for giving me the opportunity to work with him on this research. This research would have not been possible without his persistent help and guidance throughout my time at Oregon State University. I would always be grateful to Dr. Kim for giving me the opportunity to work as a Graduate Assistant for more than two years. It was this work experience that helped me succeed during my internship. Special thanks to Dr. David Porter and Dr. Kim for nominating me for the Thesis Completion Award of the summer 2019 academic term.

I would like to thank Dr. Hector Vergara for serving as my committee member, Dr. Sharmodeep Bhattacharyya for serving as my minor professor and Dr. Bogdan Strimbu for serving as my graduate council representative. Thank you all for taking the time to read this research document.

I would like to express gratitude towards my father, Mahesh, my mother Rashmi, my sister, Apoorva and my grandparents for always believing in my abilities, blessing me and empowering me to travel to the United States for my graduate studies. I would like to thank my colleague and my friend Prashant Tiwari for being nothing less than an elder brother at Oregon State University. Also, thank you Faisal for sharing a part of your thesis work with me, it was certainly useful in this research.

Lastly, thank you to the Oregon State University Graduate School and the Mechanical, Industrial and Manufacturing Engineering department. Attending Oregon State University and working with Dr. David S. Kim has been the best decision of my life.

TABLE OF CONTENTS

Page

LIST OF FIGURES

<u>Figure</u> <u>Page</u>

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

<u>Figure</u> <u>Page</u>

LIST OF TABLES

Table Page

LIST OF TABLES (Continued)

Table Page

1 INTRODUCTION

1.1 The Mixed-Model Sequencing Problem

The job sequencing problem, which is deciding the processing order for a fixed set of jobs, can be important with respect to efficiently operating a moving assembly line. Today, there can be a wide variety of products assembled on the same assembly line. For example, in automotive assembly lines an automobile model may or may not have a sunroof, it may have automatic transmission or manual transmission, etc. depending on customer preferences. Because of vast product diversity, it becomes economically infeasible for manufacturers to assemble each model variation on a different assembly line. In addition, high investments and fixed costs have also compelled companies to use the same assembly line to assemble a variety of distinct products. Such assembly lines are known as mixed-model assembly lines (MMAL) (Sarker *et al.* 2001). In other words, a MMAL is a production line that assembles in stages, various configurations of a particular product. Along with the automobile industry, MMAL's are common in the electronics manufacturing industry, fan manufacturing industry, and are also used for component assembly on printed circuit boards

In the typical moving assembly line, the time a job spends in a workstation is fixed and depends on the speed of conveyor (the most common form of job movement in such lines). This means that each operator has a limited amount of time to complete the assigned tasks on a job before it reaches the end of the workstation. The time required to process each job varies depending on its configuration. Complicated jobs will require more processing time than simple jobs. Therefore, if complex jobs are launched on the assembly line in direct succession then eventually such a launching pattern will cause conveyor belt stoppages (if unfinished jobs reach the end of the window) provided that no additional help is provided at the workstation. This will reduce line throughput, which often is the primary performance measure for the line.

The specific details of a particular job sequencing problem on a mixed model assembly line are in part determined by line balancing, which can be considered one component of designing the assembly line. Line balancing is the assignment of tasks to the workstations and strives to maintain equal expected processing time on each workstation. These line balancing decisions also dictate the tools and/or machinery needed at the workstations (Marengo *et al*. 2010). Although the expected processing time at each workstation is designed to be approximately the same, within a workstation the individual jobs have processing times that vary around the expected time. It is this processing time variability that gives rise to the job sequencing problem when the assembly line is operational.

In the job sequencing problem, the objective is to find a specific order for launching jobs on the assembly line that results in efficient line operation, which can be measured in multiple ways. As noted earlier if the job processing times on a workstation for multiple jobs in a row are high (greater than the expected processing time) then it may lead to frequent conveyor stoppages causing throughput losses, or it may require substantial additional resources to prevent conveyor stoppages. On the other hand, if the job processing times on a workstation for multiple jobs in a row are low (less than the expected processing time), then this will result in increased operator idle time and reduce operator utilization thereby increasing the cost of assembly. The basic intuitive objective of job sequencing is to "smooth" the job processing time across all workstations for cost-efficient production (Franz *et al*. 2014).

There are two facts that make the job sequencing problem more difficult in practice. First, the processing times of tasks are seldom known precisely. Second, the assumed product mix used to determine the expected work at each workstation always differs from reality. Much of this uncertainty is handled by buffering with extra time/workstation length, but this must be minimized to reduce facility costs. The result is that in most assembly lines job sequencing is really important for a subset of the workstations on the line. For these workstations, there may be high job processing time variability (due to variable work content of jobs), and the actual processing times may be higher than estimated when line balancing was completed. Those workstations where sequencing is not needed have low variability in the job processing times, the actual processing times are less than estimated, and variations in individual jobs can be handled within the time buffer built into the workstation.

1.2 Assembly Line Description

This section provides the configuration of the assembly line that is considered in this research. First the classifications/types of assembly lines are presented, and then the types of job launching disciplines are discussed. Later, the types of workstation boundaries and the strategies used to deal with a work overload situation are presented. The assumptions are listed and lastly, an illustrative example that resembles to the system being considered in this research is presented.

1.2.1 Assembly Line Type

An assembly line can be classified into one of the three categories based on how jobs are transferred between workstations as shown in Figure 1. In a moving line, the jobs are transferred between the workstations with the help of a transport system such as a conveyor system that moves all jobs at a constant velocity. In a paced line, the transport system halts when a job enters the workstation window and starts moving after a certain amount of time. An unpaced line has temporary storage spaces known as buffers between workstations and the operator transfers a completed job to the downstream buffer as soon as it is completed and space is available in the buffer (Merengo *et al*. 2010). This research addresses the moving line assembly line workstation.

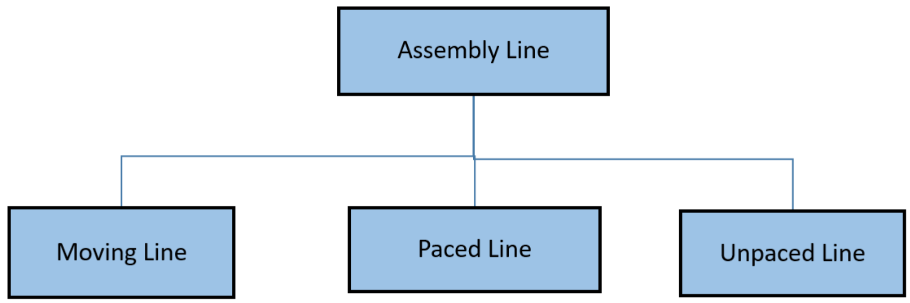


Figure 1: Classification of Assembly Line

1.2.2 Launching Discipline

Jobs are launched on a moving line type assembly line at a rate, which can be fixed or variable. In fixed rate launching, jobs are launched after equal time intervals known as the cycle time. In variable rate launching, the time interval between job launching is modified to reduce inefficiencies such as idle time, the number of incomplete jobs (Wester *et al.* 1964). For mass production, a variable rate launching system is not practical to implement (Wild, 1972). Therefore, in this research, fixed-rate launching is considered.

1.2.3 Workstation Boundaries and Work Overload Reaction

Workstations are linked serially through a conveyor system, and these workstations can have an open or closed boundary. In a closed boundary workstation, an operator is not allowed to cross the workstation boundaries at any time. This type of workstation boundary is a result of operations performed at the workstation. For example, 1) tasks to be performed in an artificially regulated environment such as heated chambers must be completed before jobs leave the chamber, 2) workstations equipped with tools connected to a power supply have limited operating range (Scholl 1999). 3) robots employed along workstations have limited reach and may collide with other robots beyond boundaries. In this research, a closed boundary workstation is considered.

Closed workstations give rise to operational inefficiencies such as operator idle time and the need for utility worker time. When a job enters the upstream boundary of the workstation, the operator starts working on it. The operator moves along with the job towards the downstream boundary of the workstation at the conveyor speed. When the tasks to be performed on the job are completed, the operator moves towards the next job in the direction opposite to conveyor movement. If a job is not available, the operator waits at the upstream boundary of the workstation for the next job to arrive. This gives rise to operator idle time. If the operator foresees that it is not possible to finish all of the tasks on the job, then they skip the job and walk back towards the next job, or waits for the next job to arrive. Such a reaction to imminent work overload is called the skip policy (Boysen, 2011). The incomplete work (which needs to be completed) is known as work overload or utility work and the time required to complete this incomplete work is known as utility time.

1.2.4 Assumptions

The following are the assumptions made in this research:

- Help is always available to perform utility work within the workstation boundaries. This ensures completion of the job by the time it leaves the workstation so that the starting times of activities in the succeeding workstations are not obstructed.
- The operator moves with an infinite velocity while returning to a succeeding job and starts working on the job as soon as they meet the job. This assumption is made because the operator movement speed is generally fast compared to the conveyor speed.
- No setup time is needed when the operator switches between different job types. This is the time required to prepare the workstation when different job types arrive at the workstation, for example, setting up fixtures and jigs, setting up a station for applying a new color, etc.

- Workstations have one dedicated operator. This operator and the utility worker can work simultaneously on two consecutive jobs in a workstation without interfering with each other’s activities.
- All job types can be completed within a workstation when worked on from the upstream boundary. This ensures that utility work is needed only because of sequencing, i.e., when multiple work intensive jobs are launched in succession and not because of a job cannot be completed in spite of being worked on from the upstream boundary.
- The workstation length is not greater than twice the cycle time.
- There are no unexpected machine breakdowns and no material shortages.
- The demand for each job type is known, and it does not change over time.
- The operator is never idle when work is available.
- The conveyor moves at a constant velocity.
- The processing times of the job types are known, and each job type has a different processing time.

Figure 2 is an illustration of the mixed model assembly line considered in this research.

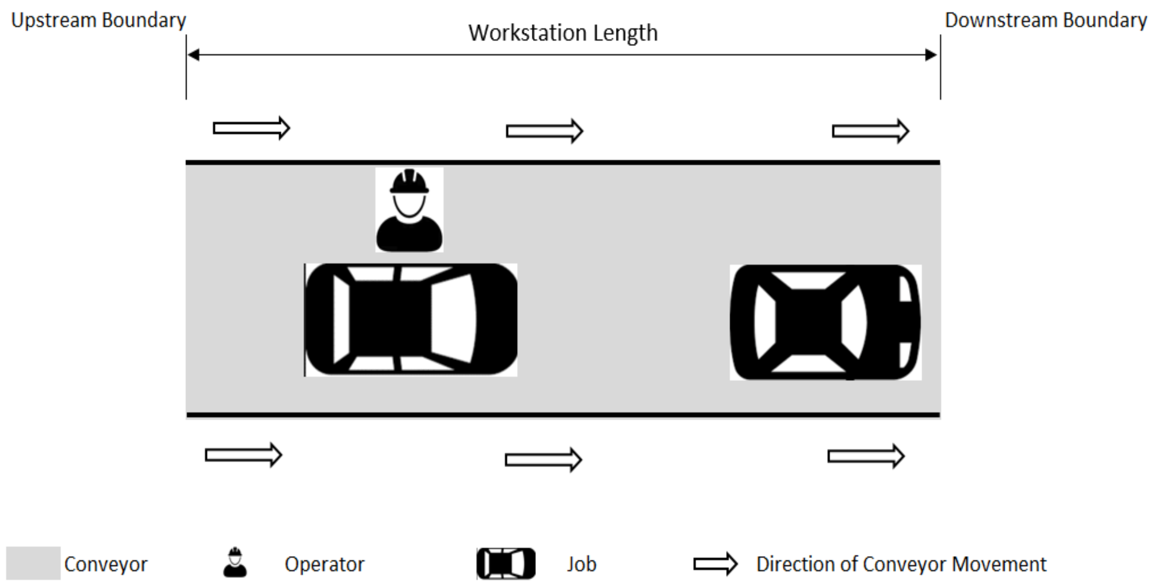


Figure 2: Mixed-Model Assembly Line

1.2.5 Illustrative Example - 1

The following example demonstrates the assembly line configuration and the operator decision-making that is addressed in this research.

Consider a mixed-model assembly line with a single workstation. Let the speed of the conveyor be one distance unit per time unit so that length units and time units can be used interchangeably. The length of the workstation is 8 distance units, i.e., a job will spend 8 time units in this workstation before it leaves the workstation. The cycle time is 4 time units which means a job will arrive at the workstation after every 4 time units. Assume three different job types *A*, *B* and *C* having processing times and demand as shown in Table 1. The sequence of launching jobs on the conveyor is *A*, *B*, *C*, *C*.

Table 1: Processing Times and Demands of Jobs for Illustrative Example -1

Job Type	Processing Time(Time Units)	Demand
A	2	1
B	5	1
C	7	2

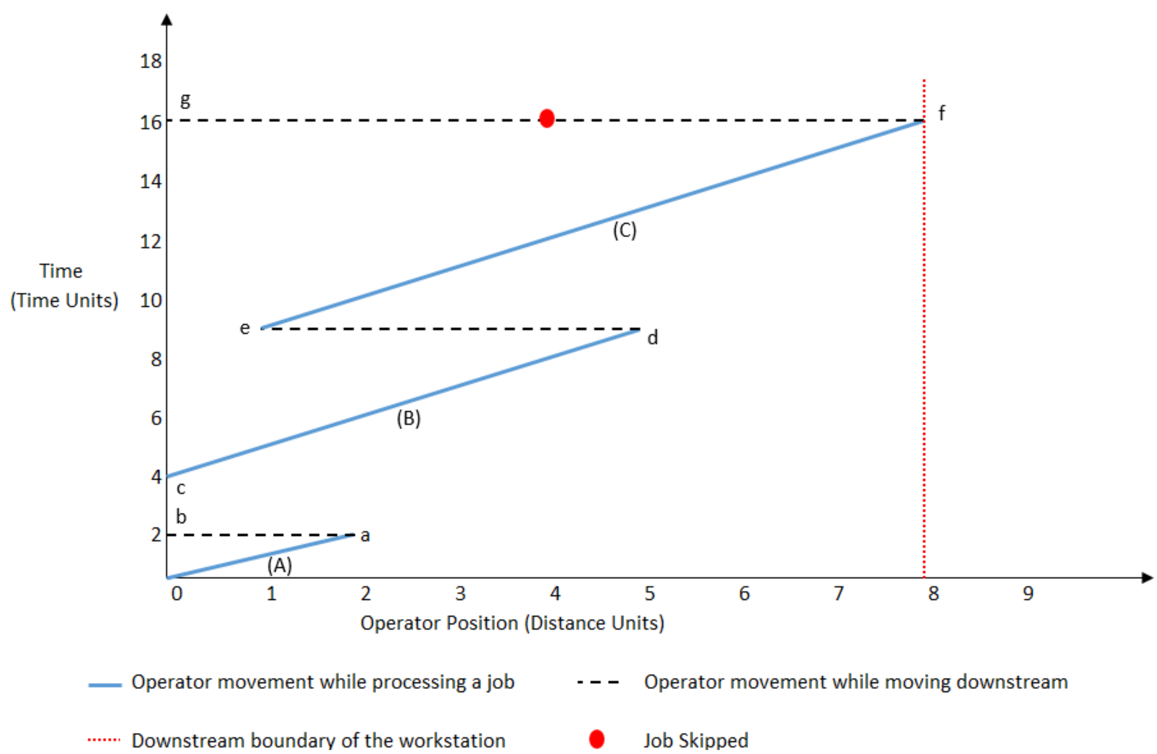


Figure 3: Operator Movement Diagram for Illustrative Example - 1

The operator movement diagram is shown in Figure 3. The diagonal lines indicate the operator is working on a job, the horizontal dashed lines indicate the upstream movement of the operator after completing a job, and the vertical dashed line indicates the downstream boundary of the workstation. Assume that the first job, i.e., job type *A* is launched at 0 time units and enters the workstation at 0 time units.

The operator starts working on job type *A* as soon as it enters the workstation and works on it for 2 time units. The operator is located 2 distance units (Figure 3, point a) away from the start of the workstation after completing job type *A*. The operator then moves towards the upstream boundary of the workstation at infinite velocity and waits for next job. Figure 3 points b-c indicates that the operator waits at the upstream boundary for the next job to arrive (idle time). At 4 time units (Figure 3 point c), the next job arrives and the operator starts working on this job type *B* and completes it at 9 time units (Figure 3 point d). Figure 3 points c-d shows the processing of the job type *B*. In the meantime, a job type *C* enters the workstation at 8 time units and is 1 distance units away from the start of the workstation at 9 time units when work begins on the job. Figure 3 points d-e shows movement of the operator towards job type *C*. The operator works on the job type *C* for 7 time units and is located at the downstream boundary after completing it (Figure 3 point f). The next job i.e. the job type *C* enters the workstation at 12 time units and is located at 4 distance units from the start of the workstation at 16 time units. The operator skips this job at 16 time units because it cannot be completed within the allowable work area (skip policy). A utility worker then processes this skipped job and the operator waits at the start of the workstation for the next job.

1.3 Research Motivation and Objective

The existing literature on the mixed-model sequencing problem primarily deals with finding an efficient sequence or launch order by applying numerous optimization search procedures. These search procedures are then validated by assuming some line and model-mix characteristics (data). Although the search procedures provide performance improvement for the respective assumed data, it is implicitly assumed that for a given line and model-mix characteristics, operational inefficiencies (work overload) always or most likely will occur and applying search procedures will reduce or eliminate such inefficiencies. Estimating operational inefficiencies for a random launch sequence before solving the sequencing problem will be beneficial because if it is unlikely to have operational inefficiencies, then the computational effort of solving sequencing problem can be avoided. This research helps in estimating such operational inefficiencies for a given workstation length and model-mix, thereby filling this gap in literature. To the best of our knowledge, this work is also the first in this area.

For demonstration purposes, consider two simple examples, Problem A and Problem B with workstation length 12. The demands and processing times of the job types are as shown in Table 2 and the operator movement diagram for Problem A and Problem B is shown in Figure 4 and Figure 5 respectively. The cycle time is 10 time units in both examples. In Problem A, there are six possible job launching sequences and sequences (*A*, *B*, *A*, *B*), (*A*, *B*, *B*, *A*) and (*B*, *A*, *B*, *A*) result in no work overload situations, whereas in Problem B, it is impossible to have a work overload situation. Therefore, it is important to solve the mixed-model sequencing problem only for Problem A because a random launching sequence has a large chance of having a work overload.

Table 2: Demand and Processing Times - 1

Problem A

Job Type	Processing Time(TU)	Demand
A	12	2
B	8	2

Problem B

Job Type	Processing Time(TU)	Demand
A	10	2
B	8	2

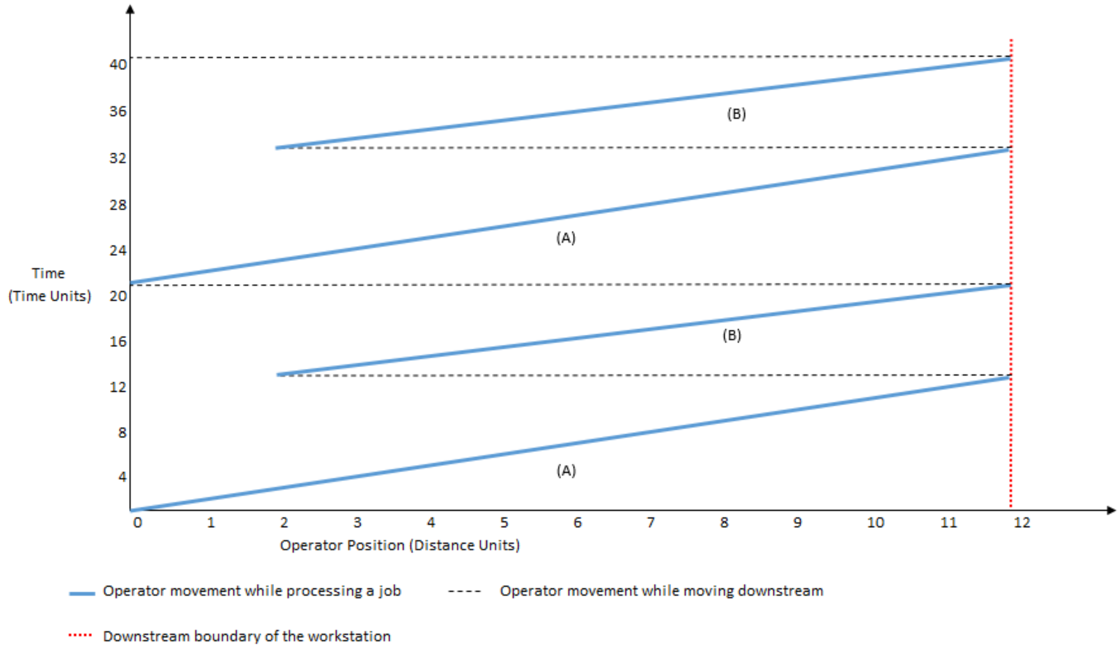


Figure 4: Operator Movement Diagram-Problem A

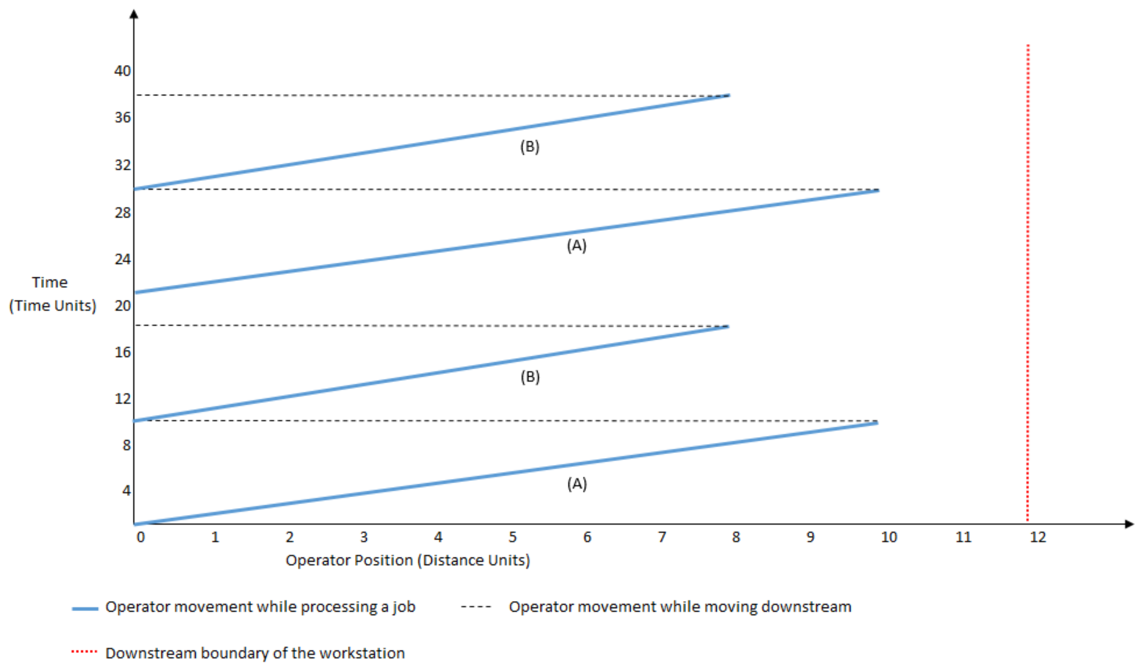


Figure 5: Operator Movement Diagram - Problem B

There are two main objectives of this research. First, to develop a model that can estimate the expected number, and the probability distribution for the number of work overload occurrences for a set of jobs launched in random order on a mixed model assembly line workstation (as a function of the workstation parameters, product mix, and characteristics of the job processing times). Second, use the model to determine whether the solution to the job sequencing problem can reduce the number of work overload occurrences to zero.

1.4 Research Outline

This research is organized as follows. In Chapter 2, the background information and the research relevant to the job sequencing problem is presented. In Chapter 3, the development of a mathematical model that can estimate work overload occurrences and the validation of the results generated by the model is presented. Chapter 4 presents the model results for multiple test cases and an application of the mathematical model. In this chapter the accuracy of the model results is also discussed. Finally, in Chapter 5 the conclusions from this research are summarized and the scope of future work is presented.

2 LITERATURE REVIEW

This chapter is organized into two parts. The first part provides relevant background information on the connection between line balancing and the short-term problem of finding a good product launching sequence. This is followed by classification of the line balancing problem. In the second part of this chapter, a classification and review of relevant mixed-model sequencing research is presented.

2.1 Background Literature

Two main decision problems arise over different planning horizons while managing a mixed-model assembly line. The first problem, which is a medium time horizon (one year) problem, is called the mixed-model assembly line balancing problem (ALBP). Before installing a mixed-model assembly line, the line length, the number of workstations and the production rate should be determined. Assignment of tasks to the workstation is another important aspect to be considered that dictates the installation of equipment and tools across workstations. All these decisions are part of the ALBP. The solution to this medium time horizon problem also determines the division labor across workstations.

The second problem, which is a short-term or operational problem, is called the model sequencing problem (MSP). As mentioned earlier, MSP deals with determining a sequence of launching jobs for a short-term planning period that optimizes some performance measure. These performance measures correspond to inefficiencies arising due to variation in workstation utilization. The MSP is often solved on a daily or weekly basis.

Inputs for the MSP include operational characteristics of the assembly line such as workstation length, cycle time, processing times of jobs (dependent on task assignment) that are determined by ALBP. Therefore, although both problems arise in different planning horizons, they are strongly related to each other. The quality of an MSP solution directly depends on the quality of ALBP solution. The precise model-mix is seldom known when the line is balanced. Therefore the ALBP solution for particular planning period may not be efficient for other periods. Both, the MSP and ALBP have to be solved separately because of their different planning horizons.

2.1.1 Assembly Line Balancing Problem

This section provides a brief classification of the ALBP and identifies the type of ALBP that needs to be solved before addressing the MSP.

Assembly line balancing is the first step in designing an assembly line. A vast literature exists on the assembly line balancing problem (ALBP) which is assigning tasks to each workstation according to the precedence relations of operations. Becker et al. (2006) and Scholl et al. (2006) provide a comprehensive survey of various forms of ALBP.

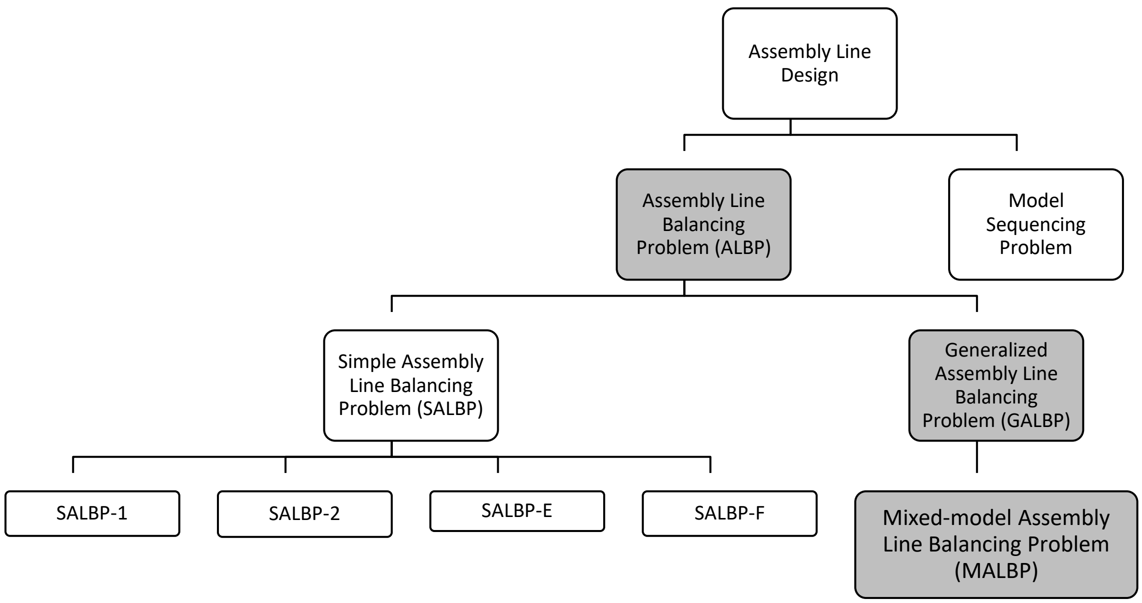


Figure 6: Classification of the Line Balancing Problem

Figure 6 shows the classification of the line balancing problem. The majority of the ALBP literature deals with modeling and solving the simple assembly line balancing problem (SALBP) which assumes a single job type (Jackson et al. 1956, Bowman et al. 1960 and Baybars et al. 1986). This SALBP can be categorized into four types based on the objective function. For SALBP-1, the goal is to minimize the number of workstations for a given production rate while SALBP-2 maximizes production rate for a given number of workstations. SALBP-E deals with minimizing the number of workstations and maximizes production rate simultaneously whereas SALBP-F evaluates the feasibility of the given assembly line.

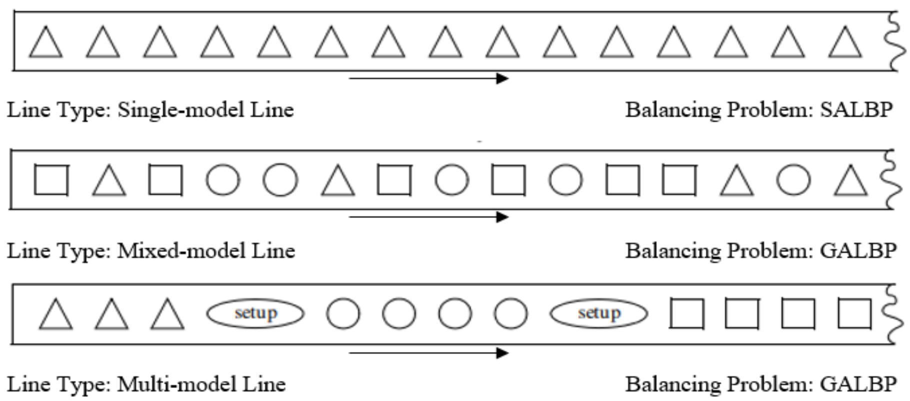


Figure 7: Assembly line type and associated balancing problem

Characteristics of the assembly line system determine the type of line balancing problem. The general assembly line balancing problem (GALBP), another version of the ALBP is obtained by relaxing one or more assumptions of the SALBP. Figure 7 shows different types of assembly lines and the associated line balancing problem. For example, an assembly line used for the production of two or more job variants (no setup times) is known as the mixed-model assembly line, and the associated balancing problem is the mixed-model assembly line balancing problem (MALBP) (Becker et al., 2006). This research requires the solution of the MALBP. Such a MALBP can be further categorised into four categories similar to that of SALBP (Scholl, 1999).

The research in this thesis assumes that the MALBP is solved and the solution is already known. In other words, it is assumed that the tasks to be performed in each workstation and the cycle time is known.

2.2 Literature Review

2.2.1 Mixed-model Sequencing Problem

This section presents a review of existing literature on the MSP. Kilbridge et al. (1963) first introduced the MSP, and since then various approaches to obtain an efficient sequence were proposed. Based on these approaches the model sequencing literature can be broadly separated into three categories as described by Boysen et al. (2011). Refer to Figure 8.

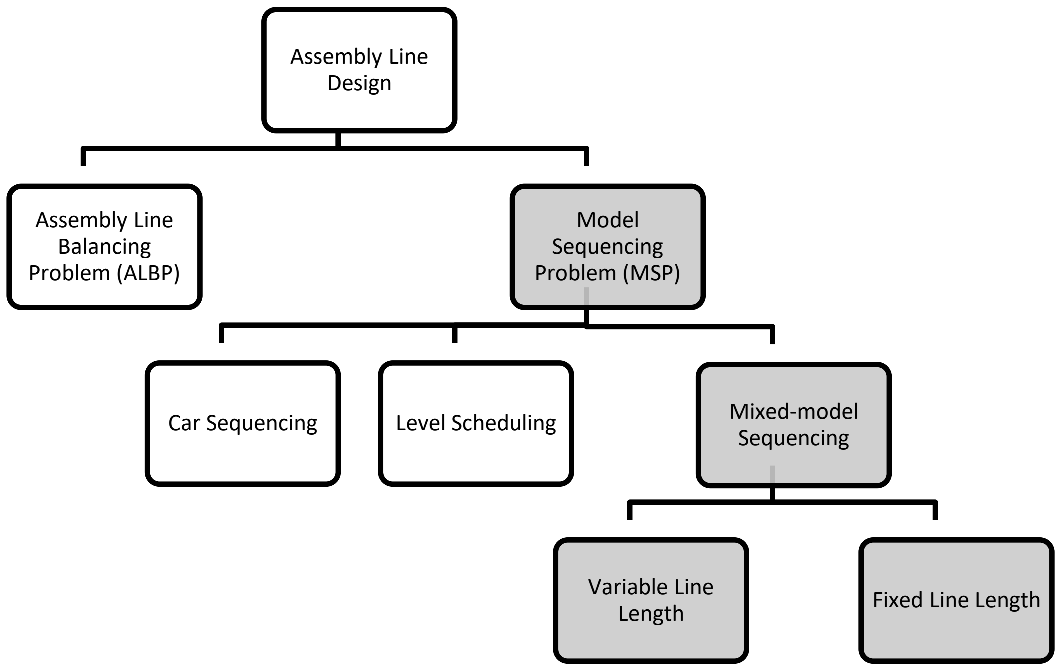


Figure 8: Classification of Model Sequencing Problem

The first approach is known as mixed-model sequencing. This approach focuses on determining the sequence of launching jobs such that some time related or cost related objective is optimized by taking into consideration various operational characteristics of assembly line, for example, workstation boundaries, job launching interval, job processing times and worker movements. The literature based on time-related objectives mainly consider minimizing total work overload, total operator idle time or setup times. Whereas, the literature on cost related objectives mainly considers minimizing total labor costs by taking into account workstation operator wages/utility worker wages and setup costs.

The second approach is known as car sequencing. The mixed-model sequencing approach requires significant effort to collect data of various operational characteristics as mentioned before. By eliminating such data collection, the car sequencing approach aims to avoid work overload by controlling the succession of labor-intensive jobs. A set of sequencing rules are formulated, and a job launching sequence adhering to these rules is determined. For example, a sequencing rule of the form $H_o{:}\,N_o$ indicates that out of N_o subsequent sequence positions, at most H_o jobs can be labor-intensive to avoid work overload occurrence.

The third approach is known as level scheduling. Raw material supplied at workstations are expensive because a large number of different parts are stored. (Scholl 1999). The first two approaches do not consider this material supply at the workstations. The level scheduling approach aims at reducing inventory levels of material stocks by implementing the just-in-time system. The main idea here is to minimize the deviation between ideal and actual production rates so that parts are supplied only when needed.

The research in this thesis deals with the first approach, therefore, the literature is mainly focused on mixed-model sequencing approach. The mixed model sequencing problem can be further separated into two main categories based on the workstation boundary assumptions. In the first category, authors search for an efficient sequence that minimizes the workstation length or throughput time. This approach determines the overall production facility dimensions. In the second approach, it is assumed that the workstation lengths (facility dimensions) are known, and the authors determine an efficient sequence that minimizes operational inefficiencies for various time-related objectives such as operator idle time, utility work, setup time and conveyor stoppage time. More detailed descriptions on the two approaches are provided in section 2.2.1.1 and 2.2.1.2.

Reaction to Imminent Work Overload Occurrence

Three policies are observed in the literature to deal with an overload occurrence as follows:

Okamura et al. (1979), Xiaobo et al. (1994), (1997), (2000) and Celano et al. (2004) address mixed-model assembly line in a Toyota production system that uses just-in-time and “Autonomation” philosophy to tackle work overload occurrence. Based on this concept, it is required that 100% defect-free jobs are moved between workstations so that operations in subsequent workstations are not disturbed. To ensure this, during a work overload occurrence, the workstation operators are given the ability to stop the conveyor until the job is completed. When a work overload situation occurs at a workstation, the whole assembly line is stopped which may give rise to operator idle time at subsequent workstations. Celano et al. (2004) describes various operator assistance policies wherein help is provided to the workstation operator stopping the conveyor system.

Another approach of addressing work overload occurrence is using utility workers as described by Yano et al. (1991) and Yano et al. (1989). Whenever a work overload occurrence threatens, the workstation operator calls a utility worker to assist him/her so that the job is completed within the workstation boundaries. When the workstation operator and utility worker work side by side, it is assumed that the processing rate is doubled, i.e. the time required to process the remaining tasks of the job is reduced to half. The utility worker is called at the point in time such that the job is completed at the downstream boundary. Boysen et al. (2011) terms this policy as the “side-by-side policy”, and describes drawbacks of this policy. The operator and utility worker may obstruct each other’s tasks when working on the same job, and if the job is small, then it cannot be processed jointly due to space constraints. It is also difficult to figure out when to call a utility worker, and it is unrealistic to assume that a utility worker arrives just when called. Tsai et al. (1995) assumes that the workstation operator leaves a job only after completing it or when he/she reaches the downstream boundary of the workstation. The unfinished job tasks are then completed by the utility worker outside the workstation boundaries.

Boysen et al. (2011) introduces the skip policy that is employed by most major European car manufacturers. A group of workers and a group leader is assigned to a section of assembly line consisting of multiple workstations. The group leader (utility worker) is cross-trained to perform all tasks within his/her section of the assembly line. If a workstation operator foresees that, the job cannot be completed within the workstation boundaries, they skip the job. The utility worker processes this skipped job as soon as it enters the workstation.

Launch Disciplines

Kilbridge et al. (1963) describes two types of launching system. Fixed rate launching (FRL) and variable rate launching (VRL). As mentioned earlier, in FRL the time interval between job launches remains constant (cycle time *c*). After every c time units, a job enters the workstation and another leaves the workstation. This time interval is equal to the weighted average of all job processing times across all workstations. The jobs are equally spaced along the assembly line. For VRL, the launch interval for the i^{th} job in the launching sequence is equal to the operating time for the $(i\text{-}1)^{th}$ job on the first workstation. VRL ensures that the operator on the first workstation starts working on i^{th} job in a sequence as soon as $(i\text{-}1)^{th}$ is completed, thus, reducing operator idle time. VRL results in different distance between jobs. Dar-El et al. (1978) mentions that for a closed workstation, though there is no evidence of VRL being better than FRL, FRL is more likely to be used because it is suitable for various conveyor system configurations such as overhead and tow-line.

2.2.1.1 Fixed Line Length Sequencing Problem

This section summarizes a line of literature that determines an efficient job launching sequence to minimize operational inefficiencies by assuming fixed workstation lengths. The most common performance measures optimized here are the risk of conveyor stoppage and work overload (total work overload measured in terms of remaining processing times of unfinished jobs or conveyor stoppage time or number of work overload occurrences).

Okamura et al. (1979) presents a formulation to solve the mixed model sequencing problem by minimizing the risk of stopping the conveyor. In this research, the conveyor stops on the occurrence of utility work, and it directly affects the plant efficiency. It is assumed that the assembly line is balanced and the workstation lengths are known. Processing times of jobs are rarely deterministic, and there is always uncertainty associated with it. To deal with the system variability and avoid conveyor stoppages, the authors suggest determining a practically acceptable sequence rather than an optimal sequence. The farther the job completion points from the downstream boundary of the workstation, the greater is the possibility of absorbing fluctuations in processing times of jobs within the workstations. Therefore, they develop a heuristic procedure to find a sequence of jobs that has the farthest work completion point (from the upstream boundary) closest to the upstream boundary. After assessing the performance of their heuristic on assumed data, they conclude that the heuristic method shall give an optimal or near-optimal sequence that minimizes the probability of conveyor stoppage.

Yano et al. (1991) et al. determines the sequence of jobs that minimizes the total work overload for a paced mixed-model assembly line having a large number of job types. In this research, it is assumed that the line is balanced. Work overload refers to the amount of work left unfinished when the job leaves the workstation. Addressing the problem of scheduling work on jobs in a workstation, the authors prove that a nonpreemptive first-come, first-served policy yields the best solution for a given sequence. This means that within a workstation, the operator should process jobs in the same sequence as they arrive and the operator should stop working on a job only after finishing it or if he/she reaches the downstream boundary of the workstation. First, formulas for estimating work overload are developed and then procedures to find optimal or near-optimal sequence are described. For a single workstation, the authors introduce a concept known as regeneration to be used to derive an optimal sequence. Regeneration means launching a certain number of labor-intensive jobs (which will move operator towards the downstream boundary of the workstation) followed by a certain number of simple jobs (which will bring operator back towards the downstream boundary of the workstation). Such a repetitive (regenerative) sequence that minimizes work overload for a single workstation can be derived by solving a nonlinear integer problem provided by the authors. For multiple workstations, a heuristic procedure is developed to find a near-optimal sequence adhering to the regeneration property. A 55% reduction (average) in work overload is reported by using this heuristic procedure for an assembly line of a major automobile company. This research can also be used for designing workstation lengths by estimating the effect of workstation length on work overload.

Okamura et al. (1979) considers the problem of mixed-model sequencing, however, they do not take into consideration the occurrence of conveyor stoppages. Xiaobo et al. (1994) develops a method to find an optimal sequence of launching jobs to minimize the conveyor stoppage time in a just-in-time assembly line. In a just-in-time assembly system, the operators have the power and responsibility to stop the conveyor when they fail to complete their operations within their workstations (Monden 1993). Such line stoppages affect operations at other workstations giving rise to operator idle time. They first define lower and upper bounds for conveyor stoppage time and operator idle time and then devise branch-and-bound method to solve the sequencing problem. Xiaobo et al. (1997) proposes a simulated annealing method to solve the same but large-scale problem that gives a good sub-optimal solution. Xiaobo et al. (2000) formulates a heuristic procedure for solving the same sequencing problem by considering the operator upstream movement times that were ignored in Xiaobo et al. (1994) and Xiaobo et al. (1997).

Boysen et al. (2011) introduces a new approach to deal with imminent work overload known as skip policy that is common in European car manufacturers. All previous research assumed side-by-side policy to tackle work overload occurrence. As mentioned before, in skip policy, before working on a new job, the workstation operator calls a utility worker if he/she predicts that the new job cannot be completed with the workstation boundaries. The utility worker then exclusively works on this job from the upstream boundary of the workstation whereas the workstation operator skips it and moves toward the succeeding job. A mathematical model is formulated to find a job launching sequence that would minimize the number of work overload occurrences instead of the amount of work overload. To find the optimal sequence, branch-and-bound and heuristic search procedures are proposed which are then tested on a newly generated data set. Using the same data, the skip policy and side-by-side policy were compared to check which policy is more economically efficient. Results indicate that if setup time (time required to interrupt ongoing work, walk towards the workstation, obtain information about the job to process) of utility workers is considered, the skip policy is superior to the side-by-side policy.

2.2.1.2 Variable Line Length Sequencing Problem

This section summarizes various time and cost related objectives considered by researchers and the methodologies adopted to solve the sequencing problem. This line of research assumes that the workstation length can be modified. The main idea in this line of literature is to determine workstation dimensions such that the workstation operator is never idle and no utility work is needed, in other words, operator interference is avoided. Concepts such as minimum part set (MPS) and start schedule are also introduced in this section.

Dar-El et al. (1975) presents the sequencing problem with an objective to minimize overall assembly line-length for zero operator idle time and utility work. In this research, it is assumed that the solution for the MALBP is known. The authors develop a heuristic algorithm to find a job sequence that minimizes the line length for both open and closed workstations. The heuristic starts with the lower bound for the length of the workstation and uses a selection heuristic to sample jobs for the sequence. If the selected job satisfies the acceptance heuristic, it is added in the sequence otherwise next highest ranking job according to the selection heuristic is tried. If no job satisfies the acceptance heuristic, the workstation length is incremented, and the same procedure is repeated. Based on the results, the authors conclude that the heuristic algorithm minimizes over all line length for no operator interference and cautions that this derived minimum line length depends upon the quality of MALBP solution. The authors also recommend the use of open boundary workstations for efficient utilization of space wherever possible.

Dar-El et al. (1977) presents a two-step algorithm to determine an optimal sequence that minimizes overall line length for zero operator idle time. This research assumes that the MALBP is solved i.e. the line is balanced and considers closed workstations. Starting from a minimum workstation length (similar to Dar-El et al. (1975)) the first step algorithm finds sequences of jobs for which the starting points and completion points coincide (cycle sequence). If no sequence exists then the workstation length is increased, and the search is repeated. Step two finds the combination of cycle sequences that satisfy the total demand by solving the integer program. If optimal solution is not found, then the workstation length is increased, and cycle sequences are regenerated. The authors conclude that the algorithm will always generate an optimal sequence only if the line is balanced. This algorithm can be used to design new mixed model assembly lines.

Bard et al. (1992) provides a mathematical framework to find sequences that minimize line length and throughput time for various MMAL configurations whereas previous literature was entirely focused on developing heuristic procedures, which applied to limited MMAL configurations. This research assumes that line balance is achieved. If the total demand of job types is represented by a vector of integers ($d_1, d_2, \dots, d_m$) where d_m represents demand for model type m and if q is the greatest common divisor of all the elements of the vector then, the vector ($\frac{d1}{q}, \frac{d2}{q}, \dots, \frac{dm}{q}$) is called minimum part set (MPS). To simplify the computations and reduce the complexity of the problem the authors obtain a sequence for the MPS and this sequence is repeated q times until demand is satisfied. The authors describe two operator working schedules. Early start schedule, wherein the operators wait for a job to arrive and start immediately when jobs enter workstations. This may increase operator idle time. Late start schedule, wherein operators start working when a sufficient amount of work is available in the workstation. This may avoid operator idle time but increase workstation length. The authors formulate six mathematical models for different combinations of four parameters as follows 1) Objective (Minimize line length or Minimize throughput time) 2) Schedule (Early start or Late start) 3) Workstation boundaries (Open or Close) 4) Launching rate (Fixed or Variable). The authors then perform various tests to study the solution of the mathematical models. For fixed rate launching, closed workstations and minimizing line length objective the results indicate a reduction in line length for an early start schedule as compared to a late start schedule however, this yielded increased throughput time and operator idle time. A similar trade-off is observed between line length and operator idle time when the job launching discipline is switched. Use of variable launching rate increased the complexity of the problem and resulted in reduced facility size but increased operator idle time for both objectives. The authors concluded that both objectives, minimize line length and minimize throughput time were always within 5% of each other i.e., both objectives generated almost the same results.

Sarker et al. (1998) developes a mathematical model to minimize total cost incurred due to operator idle time and utility time for a MMAL with optimal launch rate (FRL), workstation length and job sequence. Two separate mathematical models are developed for both open and closed workstations. The authors mention that generally, unit utility time cost is greater than unit idle time cost because utility work requires extra labor and also incur a quality cost. For assumed data, optimal line parameters are obtained by solving the mixed-integer programming model for open and closed workstations. It found that minimum total cost for the open workstation assembly line is less than closed workstation line having the same parameters. The authors also investigated the effects of various line parameters on total cost, and the results led to the following conclusions: Launch interval, workstation length, and job launching sequence affect the system throughput. When the launching interval is increased, the system throughput rate increases but the operator utilization reduces. The launching rate does not affect the optimal solution. Up to a certain assembly line length, the total cost incurred decreases and open workstations are more beneficial but, beyond it, the total cost cannot be reduced, and it is better to use closed workstations.

To the best of our knowledge, no literature exits on estimation of operational inefficiencies for a random job launch sequence and therefore it is assumed that this research is the first of its kind.

3 METHODOLOGY

The question this thesis addresses is: What is the distribution of work overload occurrences for a set of jobs launched down an assembly line in random order, and can job sequencing reduce work overload occurrences to zero? This question can be answered by estimating the number of work overload occurrences for all possible job launching sequences and then calculating the probability of getting a certain number of work overload occurrences. However, it becomes impractical to evaluate all possible job launching sequences when the number of possible sequences is large. The number of possible sequences in which jobs can be launched down an assembly line depends upon the total number of jobs to be launched in a planning period. For example, if the number of jobs to be launched are 100, then there are 100! possible sequences or in other words 100! possible ways of launching those 100 jobs which is a 158 digit number. Therefore, it becomes computationally impractical to evaluate all 100! sequences in terms of the number of work overload occurrences they can create and thus, predicting the number of work overload occurrences by brute force computation becomes impossible when the number of jobs to be launched is large.

In this section the development of a mathematical model to estimate the probability distribution of the number of work overload occurrences for a set of jobs launched down an assembly line workstation in random order is presented. This chapter consists of two main sections. Section 3.1 focuses on the formulation of the mathematical model that approximates the probability distribution of the number of work overload occurrences. Section 3.2 presents the methodology used to determine the same probabilities computationally. The computed probabilities are then used for validating the results generated using the mathematical model. Notation used in this chapter is presented in Table 3.

Table 3: Notation Summary

Notation	Description
M	Number of job types (index $m = 1, 2, \ldots, M$)
d_m	Proportion of demand that is for model m during the planning period
C	Cycle time

Notation	Description
L	Length of the workstation (distance units)
N	Number of jobs in the sequence (index n = 1, 2, …, N)
PT_m	Processing time of job type m
$PT_{(n)}$	Processing time for the job type that is n^{th} in a sequence of jobs.
$PT_{\max}$	Maximum processing time (max $\{PT_m\}$)
PT_{min}	Minimum processing time (min $\{PT_m\}$)
s_n	Starting position of the workstation operator for the n^{th} job
S_n	Completion position of the workstation operator for the n^{th} job
V	Velocity of the conveyor (distance units/time unit)

3.1 Determining Work Overload Probabilities Mathematically

In this research, the probability of a certain number of work overload occurrences is calculated using three steps. In step one, a discrete state Markov chain model is developed, in step two, the Markov chain model is transformed into a smaller Markov chain model known as macrostate Markov chain model and lastly, in step three, the probabilities are calculated using a formula that can be derived from the macrostate Markov chain model. This section has three main parts. Section 3.1.1 presents a Markov chain model applicable to any given mixed-model assembly line workstation similar to that being referred to in this research. The states of this Markov chain are then aggregated into two states, to form a macrostate Markov chain. The need for this aggregation and the aggregation procedure is elaborated in Section 3.1.2. Finally, Section 3.1.3 presents the formulas that can be used to determine the probability distribution of the number of work overload occurrences.

3.1.1 Formulating the Markov Chain Model

This section presents how the system considered is modeled as a discrete time Markov chain, and it is divided into four subsections. Section 3.1.1.1 explains the nature of the work completion points and provides justification for considering transitions of work completion points as a Markov process. The state space for the Markov chain model is defined in Section 3.1.1.2. Section 3.1.1.3 presents a formulation for the transition probability matrix of the Markov chain model, and lastly, Section 3.1.1.4 explains the existence of the limiting probabilities for the discrete state Markov chain model.

3.1.1.1 Job Completion Points

Let S_n indicate the distance the n^{th} job in the sequence is from the upstream workstation boundary when all job tasks are completed. Let s_n indicate the starting position of the operator for the same job. It is assumed that s_1 = 0. That is, the operator is waiting at the upstream workstation boundary for the first job in the sequence. After working on the n^{th} job for $PT_{(n)}$ time units, the job will be located at S_n = s_n + V*$PT_{(n)}$ distance units from the upstream boundary where the subscript *(n)* denotes the model type of the n^{th} job in the sequence. To simplify V= 1 distance unit/time unit, and the terminating position for the n^{th} job will be S_n = s_n + $PT_{(n)}$. When $S_n \leq L$ the workstation operator completes the job and then walks *C* distance units with an infinite velocity to the next job, or walks less than *C* distance units and waits at the upstream boundary for the next job so that s_{n+1} = max$\{S_n - C, 0\}$. When $S_n > L$ (work overload occurs) the workstation operator will skip job n since it cannot be completed within the workstation boundaries and s_{n+1} = 0 (assuming $2C > L$, which implies there is never more than two jobs in a workstation at any time). Figure 9 shows the transitions of work starting and completion points.

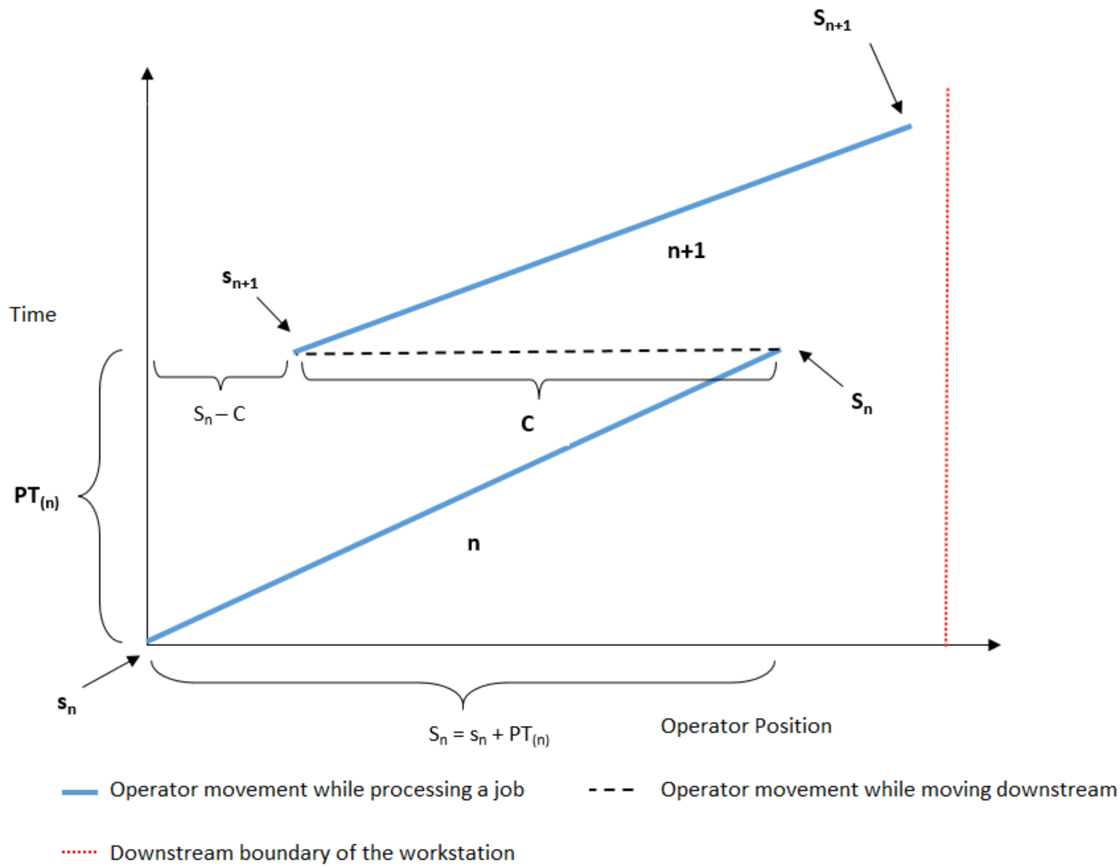


Figure 9: Job Starting and Completion Points

For purposes of exposition it will be assumed that the cycle time and process times for various job types are all integer values. This is not limiting since a single integer unit may represent different time units so that actual cycle times and process times can be closely approximated.

3.1.1.2 The Markov Chain Model

Consider an infinitely long sequence of jobs that are to be processed on a workstation with length L and cycle time C. The number of different job types in the sequence is finite (equal to M), and the proportion of each job type is fixed and known(d_m). The processing times for each job type are assumed to be constant integer values and known. Then the stochastic process $\{S_n,\ n = 1,2\ 3, \ldots\}$ is a discrete state Markov chain, where the state is the work completion point S_n for the n^{th} job in the sequence. The state space is discrete and finite since the process times and cycle time are an integer number of time units, and the number of job types is finite. The stochastic process $\{S_n,\ n = 1,2\ 3, \ldots\}$ is a Markov chain since it satisfies the Markov property: $P\{S_{n+1} = j | S_n = i, S_{n-1} = i_{n-1}, \cdots, S_1 = i_1\} \quad = P\{S_{n+1} = j | S_n = i\}$. since the completion point of the $n + 1^{th}$ job solely depends upon processing time of the $n + 1^{th}$ job and the completion point of the n^{th} job. The Markov chain is homogeneous (stationary) since the proportion of job types is constant.

The smallest value possible for a job completion point S_n occurs when the workstation operator starts working on a job type that has the smallest processing time when it enters the upstream boundary of the workstation.

The largest possible value for a job completion point (if the operator stays with the job until it completes) occurs when the operator finishes the n^{th} job at the downstream boundary of the workstation and $PT_{(n+1)} = PT_{max}$. Therefore, the state space for the Markov chain model is,

$$\{\text{State Space}\} = \{PT_{min}, PT_{min} + z, PT_{min} + 2z, \ldots, L - C + PT_{max}\}$$

Where, z is the greatest common divisor of C and all process times. For most data sets z =1.

All states having values greater than the workstation length are work overload states where a utility worker completes the job tasks. A transition of the Markov chain into any of the work overload states indicates a work overload occurrence. The following example illustrates the Markov chain state model for a given mixed-model assembly line workstation and product mix.

<u>Illustrative Example 2</u>

Consider a mixed-model assembly line workstation. Let the cycle time be 10 time units and four job types are launched down the assembly line in a random sequence with processing times as shown in Table 4. Let the length of workstation be 13 time units.

Table 4: Processing Times of Job Types

Job Type	Processing Time(time units)	Proportion in Demand
A	9	0.25
B	10	0.25
C	11	0.25
D	12	0.25

Here,

PT_{min}	= =	PT_A 9
$L - C + PT_{max}$	= =	$L - C + PT_D$ 15
z	= =	Greatest common divisor of PT_A, PT_B, PT_C, PT_D and C 1
{State Space}	= =	$\{PT_{min}, PT_{min} + z, PT_{min} + 2z, \ldots., L - C + PT_{max}\}$ {9, 10, 11, 12, 13, 14, 15}

The smallest state value possible for S_n is 9 and it occurs if the operator starts working on job type A from the upstream boundary of the workstation. The largest possible state value for $S_{n+1}$is 15 and it occurs if the n^{th} job is completed at the upstream boundary of the workstation and the $n + 1^{th}$ job is the most work-intensive job (job type D), i.e. if the work completion point of n^{th} job is 13, the operator walks 10 distance units towards the upstream boundary of the workstation (cycle time) and starts working on the $n + 1^{th}$ job from 3 so that the $n + 1^{th}$ job is completed at 15. The smallest possible distance (if not zero) between adjacent work completion points is one i.e. $z = 1$. Therefore, the state space for the given system is {9, 10, 11, 12, 13, 14, 15}.

3.1.1.3 Markov Chain Transition Probability Matrix

The next step is to build a transition probability matrix. The transition probability for moving from state *i* to state *j* in one time unit (job) is determined from the probability of observing a particular job processing time (determined by job type), and the position of the job in the workstation when processing starts (determined by *i*). For a random sequence, the probability of a specific model type *m* is d_m. Let $d_{(j)}$ be the probability of the model type whose process time results in moving to state *j* from state *i*. The transition probability matrix can be generated as follows:

$$p_{ij} = \begin{cases} d_{(j)} & for\ i \leq C, or\ i > L \quad & j = PT_1, PT_2, \dots, PT_M & (I) \\ d_{(j)} & for\ C < i \leq L & j = i + PT_1 - C, \dots, i + PT_M - C & (II) \\ 0 & otherwise & & \end{cases}$$

After completing a job, the workstation operator walks back *C* distance units towards the next job. This is because two adjacent jobs are located *C* distance units apart and it is assumed that the workstation operator walks back at infinite velocity. However, if the completion point of the n^{th} job is located at a distance less than or equal to *C* distance units, the workstation operator will have to wait at the upstream boundary for the next job to arrive. The next job begins processing as soon as it enters the workstation. If S_n is less than or equal to *C*, S_{n+1} will be at $PT_{(n+1)}$ distance units away from the start of the workstation. This is case (I). It also represents the case where the workstation operator, after completing the n^{th} job is located at a distance more than L distance units from the upstream boundary of the workstation which indicates a work overload occurrence. The workstation operator skips this job, waits for a new job to arrive, and starts operating it as soon as it arrives at the upstream boundary of the workstation. Hence, S_{n+1} will be at $PT_{(n+1)}$ distance units away from the start of the workstation.

(II) represents a case when S_n is between C distance units and L distance units away from the start of the workstation. The operator first moves C distance units towards the start of the workstation and meets the $n+1^{th}$ job at (S_n – C) distance units and then works on it for $PT_{(n+1)}$ distance units. Therefore, S_{n+1} will be located at S_n – C + $PT_{(n+1)}$ distance units away from the start of the workstation. Note that in this case, the workstation operator never reaches the upstream boundary of the workstation.

The one-step transition probability matrix has the structure as shown below:

$$\vec{P} = \begin{array}{ll|ccccc|} & & \multicolumn{5}{c}{\underline{\text{State } j}} \\ & & (PT_{min}) & (PT_{min}+z) & (PT_{min}+2z) & \ldots & (L-C+PT_{max}) \\ & (PT_{min}) & & & & & \\ & (PT_{min}+z) & & & & & \\ \underline{\text{State } i} & (PT_{min}+2z) & & & & & \\ & . & & & & & \\ & . & & & & & \\ & (L-C+PT_{max}) & & & & & \end{array}$$

The transition probability matrix for the example in section 3.1.1.2 is shown next.

Illustrative Example 2

	9	10	11	12	13	14	15
9	0.25	0.25	0.25	0.25	0	0	0
10	0.25	0.25	0.25	0.25	0	0	0
11	0	0.25	0.25	0.25	0.25	0	0
12	0	0	0.25	0.25	0.25	0.25	0
13	0	0	0	0.25	0.25	0.25	0.25
14	0.25	0.25	0.25	0.25	0	0	0
15	0.25	0.25	0.25	0.25	0	0	0

Figure 10: Transition Probability Matrix for Illustrative Example 2

The transition probability matrix for Illustrative Example 2 is shown in Figure 10. As mentioned earlier, if the completion point of n^{th} job is less than or equal to Cdistance units or greater than L distance units then the completion point of $n+1^{th}$ job will be at $PT_{(n+1)}$distance units because the workstation operator starts working on the $n+1^{th}$ job from the upstream boundary of the workstation in these cases. Therefore, when the state of the n^{th} job is 9, 10, 14 or 15, the completion point of $n+1^{th}$job is equivalent to the processing time of the $n+1^{th}$job, and the transition probability is $d_{(n+1)}$, which for this example is 0.25 for all job types. When the completion point for n^{th} job is 11, the operator starts working on the $n+1^{th}$ job at 1 distance units and depending on the type of $n+1^{th}$job the completion point will be either 10 (job type A), 11 (job type B), 12 (job type C), or 13 (job type D). Similarly, if the state of the n^{th} job is 12, the completion point for $n+1^{th}$job will be either 11,12,13, or, 14, and if the state of the n^{th} job is 13, the completion points can be either 12,13,14, or 15.

Let $p_{ij}(n)$ denote the n-step state transition probability. It is the conditional probability that the workstation operator will be located at *j* after working on exactly *n* jobs that are randomly selected from a given job set and launched down an assembly line given that he is presently located at *i*. The n-step transition probabilities can be obtained by multiplying the transition probability matrix by itself n times, where n indicates the number of jobs or Markov chain time steps. For an ergodic Markov chain, the n-step transition probability matrix will converge as *n* increases. The probabilities in each column converge to the steady state probability for the state corresponding to the column. The limiting or steady-state probabilities represent the probability of being in various states, far into the future, independent of the initial state. For an assembly line workstation, these steady-state probabilities are the probabilities of the workstation operator being located at various locations in a workstation when an infinite number of jobs is launched in a random sequence. These steady state probabilities exists only for Markov chains having a certain structure (ergodicity), which is established next.

3.1.1.4 Existence and Determination of Steady-state Probabilities

For a finite state, irreducible Markov chain with aperiodic states, the limiting probabilities π_r exist and all π_r's are greater than 0. In the assembly line workstation Markov chain model,

i. It is possible to eventually visit state j if the chain starts at state i and state i can be eventually visited from state j. For instance, in Illustrative Example 2, state 12 is visited from state 9 if the job launching sequence is A(n=1), A(n=2),D (n=3) and state 9 can be visited from state 12 if the job launching sequence is D(n=1), A(n=2). All states are communicating and therefore belong to the class. This makes the Markov chain irreducible.
ii. For every n^{th} job, having $S_n \leq C$, the $n+1^{th}$ job is worked on from the upstream boundary of the workstation. If the $n+1^{th}$ job type is the same as that of n^{th} then $S_{n+1} = S_n$. In other words, all the states having value less than or equal to C are aperiodic which makes the Markov chain aperiodic. In the Illustrative Example 2, state 9 and state 10 are aperiodic states.
iii. There are finite number of states, which implies the Markov chain is positive recurrent.

For such a finite state, aperiodic, positive recurrent (ergodic) Markov chain, steady state probabilities exists and can calculated using following steps:

Step 1: Construct the state transition matrix $\vec{P}$

Step 2: Find $\vec{P}^*$, where $\vec{P}^* = \vec{P} - \vec{I}$ and $\vec{I}$ is an identity matrix having dimensions of $\vec{P}$

Step 3: Find $\vec{P}^{**}$, by replacing the last column of $\vec{P}^*$ by 1's

Step 4: Find $(\vec{P}^{**})^{-1}$. Last row of $(\vec{P}^{**})^{-1}$ gives the steady state probabilities.

Illustrative Example 2

Using the steps mentioned above, the steady state probabilities for the Illustrative Example 2 are calculated and are shown in Table 5

Table 5: Steady State Probabilities for Illustrative Example 2

State	9	10	11	12	13	14	15
Steady State Probability	0.0961538	0.14904	0.21154	0.25	0.15385	0.10096	0.03846

As mentioned before, these steady state probabilities give us the probabilities of work completion points being at various locations along the workstation for a random job launching sequence. The number of jobs launched times the steady state probability of a state gives us an approximation of the average number of jobs that are completed at a location represented by that state. The steady state probabilities represent the portion of time the job is completed at a location for a set of infinite jobs. Since we assume finite number of jobs in a job set, the steady state probabilities are considered to be an approximation. This is discussed in detail in the next chapter. If the number of jobs launched are 1000, then the average number of jobs completed at 9 distance units away from the upstream boundary will be 1000*0.0961 which is 96.1. By similar logic, the average number of jobs causing work overload occurrence is a sum of the average number of jobs completed at 14 distance units and 15 distance units away from the upstream boundary which represents work overload states.

$$\text{Average number of work overload occurrences} = 1000 * (\pi_{14} + \pi_{15}) = 139.42$$

For a random 1000 job sequence, the approximate average number of work overload occurrences is 139.42. Although the steady state probabilities obtained from the Markov chain are used estimate the expected number of work overload occurrences in a sequence, it does not directly provide probabilities for a specific number of work overload occurrences in a sequence of jobs processed through the workstation. The probability distribution for the number of work overload occurrences is obtained by examining Markov chain transitions from any non-work overload states (non-work overload states are job completion points that are located within the work station boundaries) to any work overload states (work overload states are job completion points located beyond the downstream boundary). This was accomplished by aggregating all non-work overload states into a single state, and all work overload states of the Markov chain into a single state and examining transitions between these two states. This is presented in detail in the following section.

3.1.2 Markov Chain State Aggregation

If $\{x_i, i = 1, 2, 3, \ldots\}$ is an ergodic Markov chain, state aggregation creates another ergodic Markov chain $\{X_i, i = 1, 2, 3, \ldots\}$ by aggregating or mapping multiple states from chain $\{x_i, i = 1, 2, 3, \ldots\}$ to a single state in chain $\{X_i, i = 1, 2, 3, \ldots\}$(Kim et al. (1995)). All states in $\{x_i, i = 1, 2, 3, \ldots\}$ are mapped to some state in $\{X_i, i = 1, 2, 3, \ldots\}$. The transition probabilities in chain $\{X_i, i = 1, 2, 3, \ldots\}$ can be computed such that the limiting probability of a state in $\{X_i, i = 1, 2, 3, \ldots\}$, $\Pi_j = \sum \pi_k$, where π_k is the limiting probability of state k in Markov chain $\{x_i, i = 1, 2, 3, \ldots\}$. The summation is over all states mapped to state j in Markov chain $\{X_i, i = 1, 2, 3, \ldots\}$. The chain $\{X_i, i = 1, 2, 3, \ldots\}$ is referred to as the macrostate Markov chain. For the workstation completion time Markov chain $\{S_i, i = 1, 2, 3, \ldots\}$ the macrostate Markov chain has two states, the non-work overload state, and the work overload state. The transition probabilities for the macrostate Markov chain are calculated using the steady state probabilities and one step transition probabilities from $\{S_i, i = 1, 2, 3, \ldots\}$. Section 3.1.2.1 presents formulas for calculating the macrostate Markov chain state transition probabilities when the performance measure of interest is the number of work overload occurrences. Section 3.1.2.2 presents the same when the performance measure of interest is the severity of work overloads.

3.1.2.1 Number of Work Overload Occurrences

Let the states within the workstation boundaries, i.e. all job completion locations whose distance from the upstream boundary is less than or equal to the length of the workstation be aggregated into one state namely NWO (non-work overload) state. Let the states beyond workstation boundaries i.e. all job completion points located beyond the downstream boundary of the workstation be aggregated into one WO (work overload) state. Let A represent the total number of states within workstation boundaries (index $a = 1,2,\ldots A$) and B represent the total number of states beyond workstation boundaries (index $b = 1,2,\ldots B$).

State:	PT_{min}	$PT_{min} + z$ ….	L	|	….	$L + (PT_{max} - C)$
Steady State Probability:	π_{PTmin}	$\pi_{PTmin+z}$ ….	π_L	|	….	$\pi_{L+(PTmax-C)}$
		NWO				WO
Steady State Probability:		π_{NWO}				π_{WO}

The resulting macrostate Markov chain state transition diagram is shown in Figure 11. The state transition probabilities are labeled on the appropriate arcs.

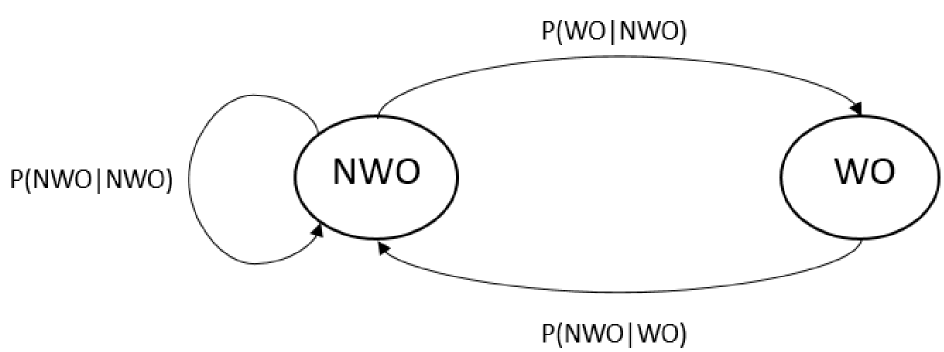


Figure 11: Macro Markov Chain State Transition Diagram

P(NWO|NWO) represents the conditional probability of transitioning to the non-work overload state given that the Markov chain is currently in the non-work overload state. This probability can be determined using (6).

$$\mathrm{P(NWO|NWO)} = \sum_{a=1}^{A} \left(\frac{\pi_a}{(\sum_{a=1}^{\mathrm{A}} \pi_a)} * (1 - \sum_{b=1}^{B} P_{ab}) \right) \quad (6)$$

Where,

P_{ab} = entry in the a^{th} row and b^{th} column of the transition probability matrix $\vec{P}$ for Markov chain $\{S_i,\ i = 1, 2, 3, \ldots\}$.

The term $\frac{\pi_a}{(\sum_{a=1}^{\mathrm{A}} \pi_a)}$ represents the weighted probability of being in a^{th} state given that it is in any of the work overload states (i.e., a =1, 2,.., A). The term $(\sum_{b=1}^{B} P_{ab})$ in indicates the total probability of transition from the a^{th} non-work overload state into some work overload state (i.e. b,=1, 2,.., B). Therefore, the term $(1 - \sum_{b=1}^{B} P_{ab})$ gives the total probability of transitioning back from the a^{th} non-work overload state into some non-work overload state (i.e. a =1, 2,.., A).

Similarly, *P(WO|NWO)*, i.e. the conditional probability of transitioning to the work overload state given that it is in the non-work overload state which can be determined using (7).

$$P(WO|NWO) = \sum_{a=1}^{A} \left(\frac{\pi_a}{(\sum_{a=1}^{A} \pi_a)} * (\sum_{b=1}^{B} P_{ab}) \right) \ldots (7)$$

P(NWO|WO) indicates the conditional probability of a transition from a work overload state to a non-work overload state. Since this research assumes a skip policy, the workstation operator will skip all work overload causing job and wait for the next job to arrive so that work on this job starts from the upstream boundary. Therefore, the job processed after encountering a work overload causing job (represented by being in *WO* state) is always completed within workstation boundaries (represented by the transition to *NWO* state from *WO* state). Hence, *P(NWO|WO)* will always be one. This also explains the absence of self-transition arc for the *WO* state in the macrostate Markov Chain state transition diagram.

Illustrative Example 2

In this example, states 9, 10, 11, 12 and 13 are aggregated into the *NWO* state and states 14 and 15 are aggregated into the *WO* state as shown in Table 6. The total number of states within workstation boundaries are A=5, and those beyond workstation boundaries are B=2. Using steady state probabilities from Table 5 and equations (6) and (7) the transition probabilities are calculated and displayed in Figure *12*. Refer to

Table **7** for calculations.

Table 6: Illustrative Example 2 State Indices for Number of Work Overload Occurrences

		a =1	a =2	a =3	a =4	a =5	b=1	b=2
		9	10	11	12	13	14	15
a =1	9	0.25	0.25	0.25	0.25	0	0	0
a =2	10	0.25	0.25	0.25	0.25	0	0	0
a =3	11	0	0.25	0.25	0.25	0.25	0	0
a =4	12	0	0	0.25	0.25	0.25	0.25	0
a =5	13	0	0	0	0.25	0.25	0.25	0.25
b=1	14	0.25	0.25	0.25	0.25	0	0	0
b=2	15	0.25	0.25	0.25	0.25	0	0	0

Table 7: Illustrative Example 2 Macrostate Markov Chain Transition Probabilities Calculations

a	$\sum_{b=1}^{B} \text{P ab}$	$\frac{\Pi a}{(\sum_{a=1}^{A} \Pi a)}$	Πa	$\sum_{a=1}^{A} \Pi a$	$\left(\frac{\Pi a}{(\sum_{a=1}^{A} \Pi a)} * \left(1 - \sum_{b=1}^{B} \text{P ab}\right)\right)$	$\left(\frac{\Pi a}{(\sum_{a=1}^{A} \Pi a)} * \left(\sum_{b=1}^{B} \text{P ab}\right)\right)$
1	0	0.112	0.096		0.112	0
2	0	0.173	0.149		0.173	0
3	0	0.246	0.212	0.8605	0.246	0
4	0	0.291	0.250		0.218	0.073
5	0.5	0.179	0.154		0.089	0.089
				Total →	0.838	0.162

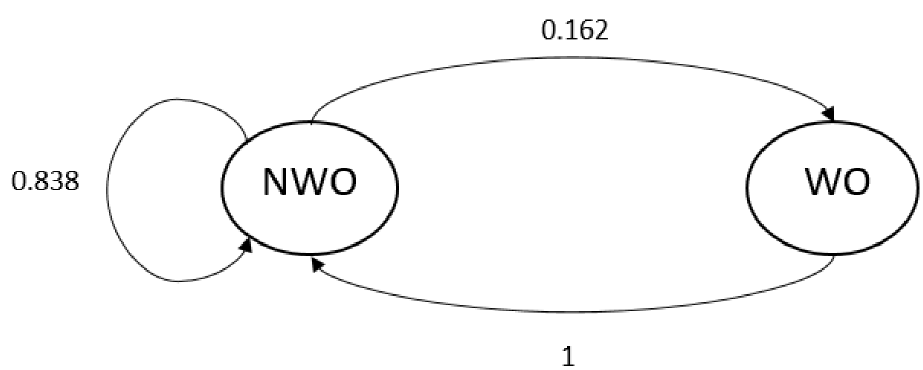


Figure 12: Illustrative Example 2 Macrostate Markov Chain State Transition Diagram

3.1.2.2 Work Overload Occurrence Severity

If the work completion location of the n^{th} job is close to the downstream boundary, then it is possible that the completion point of the $n+1^{th}$job is a work overload state. If the workstation operator completes the n^{th} job at the downstream boundary of the workstation, and the $n+1^{th}$job is the most work-intensive job then S_{n+1} will be at the farthest work completion point from the upstream boundary of the workstation. If the work overload states are numbered from 1 to B, then B is the most severe work overload state because it requires the largest amount of utility work. Similarly, b=1 is the least severe work overload state. The following establishes the formulas used to determine transition probabilities of the macrostate Markov.

Let all the states except the work overload state of interest (work overload of certain severity) say b, be aggregated into single state (say Z) and let the total number of states aggregated into Z state be A (index a = 1, .., A). The state transition diagram for such a macro Markov chain will be similar to Figure 10, and the transition probabilities can be calculated as follows:

$$P(Z|Z) = \sum_{a=1}^{A}\left(\frac{\pi_a}{(\sum_{a=1}^{A}\pi_a)} * (1 - P_{ab})\right)\dots (8)$$

Here, the term $\frac{\pi_a}{(\sum_{a=1}^{A}\pi_a)}$ gives the weighted probability of being in the state a given that it is in either of the Z states and the term P_{ab} gives the probability of transitioning from the state a of the Z state into WO state b. Similarly, the term (1 – P_{ab}) gives the probability of transitioning from state a of Z state back into one of the Z states. Because of the same reason mentioned in section 3.1.2.1 the probability of transitioning into Z state from WO state will always be one and it is not possible to reenter WO state from WO state.

<u>Illustrative Example 2</u>

In this case, state 15 is the most severe work overload state (say severity 2) as compared to state 14 (say severity 1). Let state 14 be the state of interest, i.e. we are interested in finding out the average number of jobs that cause a work overload of severity 1. Let states 9, 10,11,12,13 and 15 be aggregated into one state that is represented in the macro Markov chain as Z state. Let state 14 be indexed as b and is represented in the macro Markov chain as WO1 state. Refer to Table 8 in the appendix for state indices. Using (8) the transition probabilities into work overload state of severity 1 are calculated and displayed in Figure 13. Refer to

Table 9 for calculation details.

Table 8: Illustrative Example 2 State Indices for Examining Work Overload Severity

		a=1	a=2	a=3	a=4	a=5	b	a=6
		9	10	11	12	13	14	15
a=1	9	0.25	0.25	0.25	0.25	0	0	0
a=2	10	0.25	0.25	0.25	0.25	0	0	0
a=3	11	0	0.25	0.25	0.25	0.25	0	0
a=4	12	0	0	0.25	0.25	0.25	0.25	0
a=5	13	0	0	0	0.25	0.25	0.25	0.25
b	14	0.25	0.25	0.25	0.25	0	0	0
a=6	15	0.25	0.25	0.25	0.25	0	0	0

Table 9: Illustrative Example 2 Macrostate Markov Chain Transition Probabilities Calculations

a	Pab	$\frac{\Pi a}{(\sum_{a=1}^{A} \Pi a)}$	Πa	$\sum_{a=1}^{A} \Pi a$	$\frac{\Pi a}{(\sum_{a=1}^{A} \Pi a)} * (1 - Pab)$	$\frac{\Pi a}{(\sum_{a=1}^{A} \Pi a)} * (Pab)$
1	0	0.107	0.096	0.899	0.107	0
2	0	0.166	0.149		0.166	0
3	0	0.235	0.212		0.235	0
4	0.250	0.278	0.250		0.209	0.070
5	0.250	0.171	0.154		0.128	0.043
6	0	0.043	0.039		0.043	0
				Total →	0.888	0.112

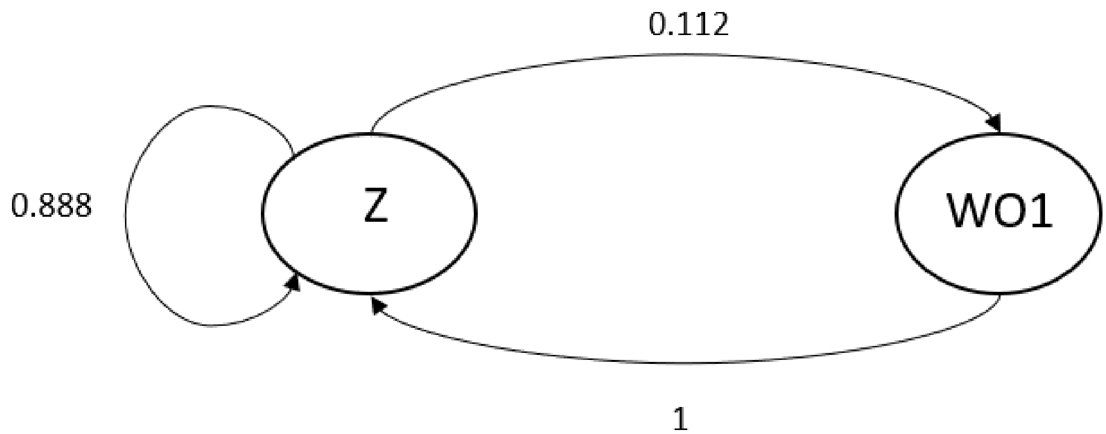


Figure 13: Illustrative Example 2 Macrostate Markov Chain State Transition Diagram

Now that we have determined the transition probabilities within the macro Markov chain, the next step is estimating the probability of the number of transitions into the work overload state i.e. number of times the WO state is visited for a fixed number of transitions. The number of transitions depends upon the number of jobs that are to be launched and each entry into WO state of the macro Markov chain resembles a work overload occurrence. This will give us an estimate of the number of times work overload situation occurs when a fixed number of jobs are launched in a random sequence.

3.1.3 Probability for the Number of Work Overload Occurrences

The macrostate Markov chain transition probabilities gives the likelihood of the Markov chain transitioning from a non-work overload state to itself and also to a work overload state. Using these probabilities, we can estimate the probability of the number of work overload occurrences for a fixed number of jobs in a random sequence. In this section, a formula for the probability distribution of the number of work overload occurrences is derived using information from the macrostate Markov chain.

Let the total number of jobs launched down an assembly line be N (index n =1,... N). There will be N state transitions (some back to the same state) in the macrostate Markov chain. The objective is to determine the probability that a transition into a work overload state occurs a specific number of times, i.e. the probability of transitioning into state WO l times when N number of jobs are processed through the assembly line workstation. This probability can be calculated using the following formula:

$$\mathrm{P(WO} = l) = \left\{\mathrm{P(WO|NWO)}^{l} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l} * \binom{\mathrm{N}-l}{l}\right\} + \left\{\mathrm{P(WO|NWO)}^{l} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l+1} * \binom{\mathrm{N}-l}{l-1}\right\} \quad (9)$$

Where l is the number of work overload occurrences, and the range of l is from 0 to $N/2$. When a work overload occurrence is encountered, the workstation operator returns to the start of the workstation and the next job starts processing from the upstream boundary. Thus the job following a work overload job will never itself cause a work overload. Hence, if N jobs are launched, the maximum number of work overload occurrences that can be encountered is $N/2$. l =0 in the above formula represents a scenario when none of the launched jobs cause work overload occurrence and $l = N/2$ represents a case when every other job in the sequence causes a work overload occurrence.

To derive formula (9) the possible outcomes are partitioned into two categories. Category 1 is when the last job (n=N) in a sequence completes in a NWO state. Category 2 is when the last job in a sequence completes in a WO state. For Category 1 outcomes and a fixed l, there is one less state NWO self-transition than for a Category 2 outcome. This is because of the skip policy, so that the transition probability into a NWO state from a WO state is equal to one. The probabilities for a specific number of transitions comes from the macrostate Markov chain transition probabilities (raised to the appropriate power).

The next step is to determine how many different ways l transitions into a WO state can occur. In Category 1 l jobs in a sequence of N jobs finish processing in a WO state, and N- l jobs finish processing in a NWO state. In Category 1 the last job finishes in a NWO state so there are N- l possible locations in a sequence where the WO states may occur (see Figure 6).

_ NWO _ NWO _ NWO ..... _ <u>NWO</u>
(n = N)

Figure 14: Possible locations for WO states in Category 1 outcomes

The *WO* states can be in any of the spaces shown in Figure 14. The l *WO* states can occupy any of the available *N*- l “spaces” preceding a *NWO* state. Therefore, the total number of possible combinations of l *WO* and *N*- l *NWO* states such that there is at least one *NWO* state between two *WO* states is $\binom{N-l}{l}$. The probability of a Category 1 outcome with l *WO* states is

$$\mathrm{P(WO|NWO)}^{l} * \mathrm{P(NWO|WO)}^{l} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l} * \binom{N-l}{l} =$$
$$\mathrm{P(NWO|WO)}^{l} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l} * \binom{N-l}{l}.$$

For Category 2 outcomes there is one additional NWO self-transition, and one less *WO* state dispersed throughput the sequence since the last job finishes in a *WO* state. For a Category 2 outcome with l work overloads, there are *N*- l possible “locations” where l -1 *WO* states can occur. This is shown in Figure 15.

_ NWO _ NWO _ NWO ..... NWO <u>WO</u>
(n = N)

Figure 15: Possible locations for WO states in Category 2 outcomes

Therefore, the total number of possible Category 2 outcomes with l *WO* states equals the number of ways l -1 *WO* locations may be selected from *N*- l choices, which equals $\binom{N-l}{l-1}$. The probability of a Category 2 outcome with l *WO* states is

$$\left\{ \mathrm{P(WO|NWO)}^{l} * \mathrm{P(NWO|WO)}^{l-1} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l+1} * \binom{N-l}{l-1} \right\}$$
$$= \mathrm{P(NWO|WO)}^{l-1} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l+1} * \binom{N-l}{l-1}$$

Combining the probability of Category 1 and Category 2 outcomes with l *WO* states gives the probability of l work overloads in a random sequence of N jobs as

$$\left\{ \mathrm{P(WO|NWO)}^{l} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l} * \binom{N-l}{l} \right\} + \left\{ \mathrm{P(WO|NWO)}^{l} * \mathrm{P(NWO|NWO)}^{\mathrm{N}-2l+1} * \binom{N-l}{l-1} \right\} \qquad (10)$$

To verify the probabilities obtained using (10), the macro Markov chain shown in Figure 12 was simulated with *Crystal Ball* software (Monte Carlo simulation add-in for Excel). The total number of transitions into a *WO* state were counted for 1000 jobs. The number of replications was set to 1,000,000. The probabilities obtained by performing the Monte Carlo simulation were then compared to those calculated using (10). Figure 16 shows the probabilities calculated using (10) and from the Monte Carlo simulation.

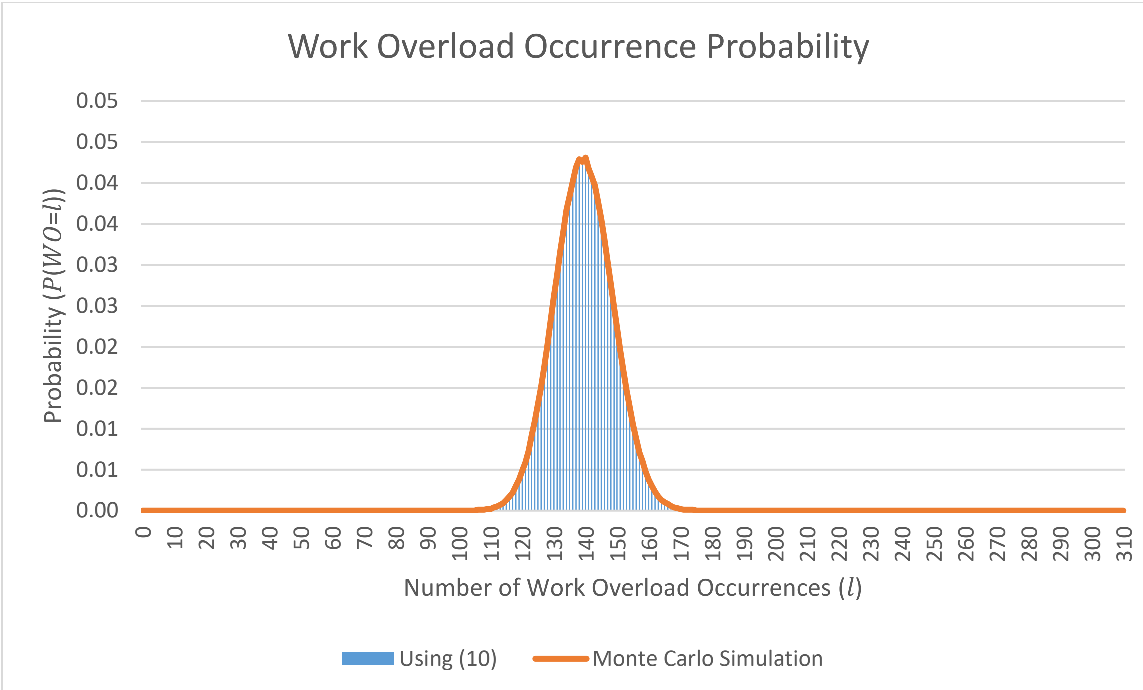


Figure 16: Probability of the Number of Work Overload Occurrences from (10) and from Monte Carlo Simulation

Appendix 1 compares the probabilities obtained using (10) and simulation for all possible values of l. Note that the probabilities in Appendix 1 are only up to four decimals. For l = 117 to 171, a small difference is observed that can be minimized by increasing the number of simulation replications. The simulation probabilities obtained using Monte Carlo simulation are similar to those obtained using (10).

<u>Illustrative Example 2</u>

In this example, the number of jobs launched down an assembly line is 1000, i.e. N=1000. The macrostate Markov chain transition probabilities P(WO|NWO) and P(NWO|NWO) are 0.162 and 0.838 respectively. The probability of getting one work overload occurrence when 1000 jobs are launched randomly can be calculated using (10) as follows

$$P(WO = 1) = \left\{0.162^1 * 0.838^{998} * \binom{999}{1}\right\} + \left\{0.162^1 * 0.838^{999} * \binom{999}{0}\right\}$$

Similarly, the probabilities for all possible values of l can be calculated using (10) and are shown in Figure 17.

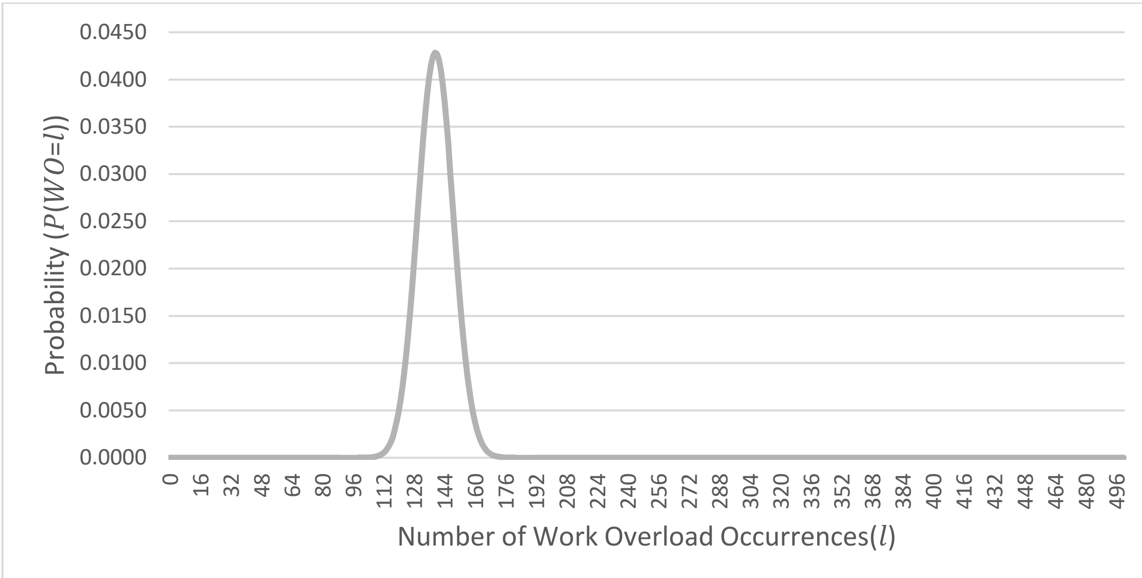


Figure 17: Probability of the Number of Work Overload Occurrences for Illustrative Example 2

The results indicate that if 1000 jobs (with known job types and proportions) are launched down an assembly line in a random sequence then the probability of getting no work overloads is extremely unlikely (P($WO = l$) = 1.75396E-77). For a random job launching sequence the most likely number of work overload occurrences is 139 (P($WO = 139$) = 0.042836).

3.2 Determining Work Overload Probabilities Computationally

This section presents the method used to computationally estimate the probability distribution for number of work overload occurrences, and the severity of work overloads for a random job launching sequence. Evaluating all possible launching sequences for a given set of jobs (if I jobs are to be launched $I!$ sequences are possible) is computationally impractical if I is large. Therefore random sequences are sampled, and the detailed method used for this sampling is presented in Figure 18 in the form of a flowchart.

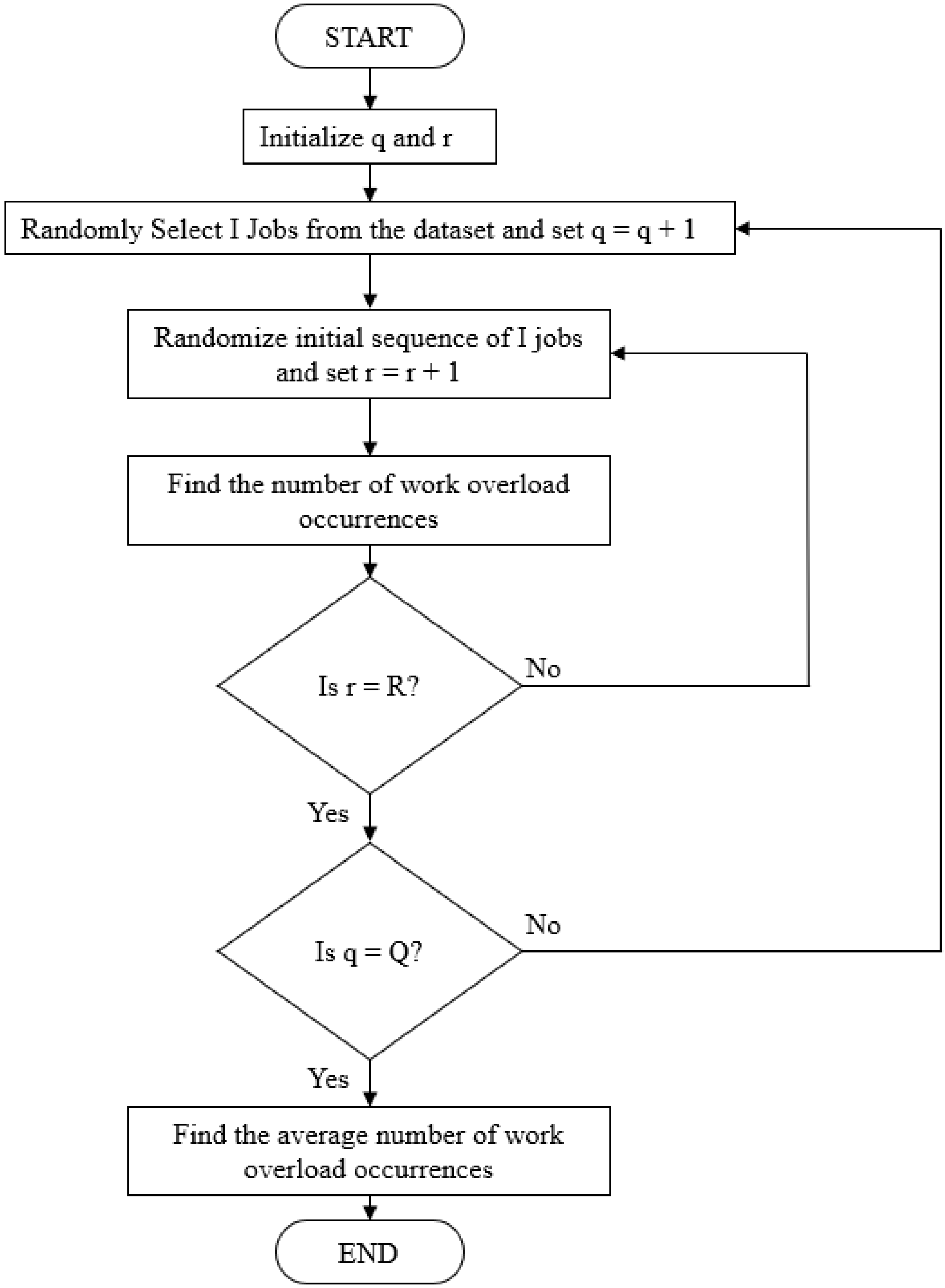


Figure 18: Flowchart of the method used to estimate the average number of work overload occurrences

In order to estimate the probability of work overload occurrences, I jobs are randomly selected from a very large population of jobs. This large population of jobs (dataset of job processing times) is generated using Crystal Ball software wherein the proportion of job type m is equal to d_m. The datasets generated for this research contains processing times of 1 million jobs. A fixed number of jobs (I) were then randomly selected from this data set. Let, Q be the number of times I jobs are randomly selected from the dataset (index q = 1, 2,.., Q)). Since these I jobs are selected randomly, the proportions of job types in each selection may not be exactly equal to d_m. However, after Q random selections of I jobs each, the average proportion of job type m over all selections will converge to d_m. The I jobs from the q^{th} selection are then randomly sequenced using the modern Fisher Yates algorithm (that was introduced by Durstenfeld, R. (1964)) and the number of work overload occurrences are determined for that sequence. The Fisher-Yates shuffle is an algorithm for generating random permutations of finite linear elements. It ensures that each random sequence generated is equally likely and therefore unbiased. The algorithm presented by Boysen et al. (2011) to evaluate the number of work overload occurrences is used. These steps of generating random sequence and determining the number of work overload occurrences are repeated R times. Once I jobs are randomly sequenced R times, a set of I jobs is again randomly selected from the dataset, and the same procedure is repeated until $q=Q$. Finally, the average number of work overload occurrences for a random launching sequence is calculated.

The results obtained using the mathematical model, and estimated computationally are discussed in detail in the following chapter.

4 RESULTS

This chapter presents the results obtained by using the Markov chain model for various combinations of job sets and workstation lengths. Different test cases are obtained by varying processing times of job types, demand fraction of job types, the workstation length and the total number of jobs in the job set. The average number of work overloads are analyzed and validated in Section 4.1 where four test cases are presented. This section also examines the discretization of continuous processing times for purposes of applying the Markov chain model. In Section 4.2 the probability distribution of the number of work overload occurrences is examined, and then the probability distribution for five test cases is presented. Section 4.3 presents the relationship between the accuracy of the Markov chain result and the numbers of jobs in a job set. Lastly, Section 4.4 presents an application of the Markov chain model. In this application it is demonstrated that the macro Markov chain transition probability can be used to identify whether solving a job sequencing problem will give a sequence with zero work overloads without solving the problem.

4.1 Average Number of Work Overload Occurrences

This section presents the average number of work overloads for four test cases Case1, Case2, Case3, and Case4. Case1 and the Case2 have different job types in same proportion and their processing times are fixed integer time units. The transition probability matrices for Case1 and Case2 are presented and explained in detail. In Case3, the demand fraction differs across job types, and the processing times follow a specific probability distribution. The last test case, Case4 consists of jobs with non-integer processing times.

<u>Case1</u>

This is a mixed model assembly line having a workstation of length 18 distance units and the cycle time is ten time units. The conveyor moves at a speed of 1 distance unit per time unit, and 100 jobs are sequenced and assembled through the line. There are eight job types with processing times and demand fractions as shown in Table 10.

Table 10: Case1 Job Set Composition

Job Type	Processing Time (Time Units)	Demand Fraction
A	9	0.125
B	10	0.125
C	11	0.125
D	12	0.125
E	13	0.125
F	14	0.125
G	15	0.125
H	16	0.125

The Markov chain state space (section 3.1.1.2) is as follows: {State Space} = {9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19(WO1), 20(WO2), 21(WO3), 22(WO4), 23(WO5), 24(WO6)}. The total number of states is 16, of which six states are work overload states. 10 other states are non-work overload states, i.e. the states that represent job completion points that are located within the workstation boundaries. State WO1 represents a work completion at 19 distance units away from the start of the workstation, and state WO6 represents a work completion at 24 distance units away from the start of the workstation.

The one-step transition probability matrix is formulated using the guidelines provided in Section 3.1.1.3 and is presented in Table 11. The rows indicate the completion point of the n^{th} job in the sequence, and the columns indicate the completion point of the $n + 1^{th}$ job in a sequence. After completing a job, the workstation operator walks ten distance units towards the upstream boundary of the workstation to catch the succeeding job in a sequence. This is because the cycle time is ten time units so that all jobs are launched ten distance units apart.

However, if the completion point of the n^{th} job is at a distance less than or equal to 10 distance units away from the upstream boundary, then the operator will have to wait at the upstream boundary for the next job to arrive. In this case, the $n + 1^{th}$ job starts processing as soon as it enters the workstation, and is completed at a distance from the upstream boundary equal to the job processing time. For example if the n^{th} job is completed at 9 distance units, the operator walks back 9 distance units and waits at the upstream boundary for 1 time unit until the next job enters the workstation. Since all job types in Case1 are in equal proportion, the probability of transitioning from state 9 to any state from 9 to 16 is equal (0.125). Similar transition probabilities are observed when the n^{th} job is completed at 10 distance units.

If the n^{th} job causes a work overload, it will be skipped by the operator. The next job starts processing at the upstream boundary, and is completed at either 9, 10, 11, 12, 13, 14, 15 distance units from the upstream boundary depending upon the processing time of the job. Since all job types occur in equal proportions, the probability of transitioning from state 19 to state 8, 11, 13 or 15 is equal to 0.125. Similarly, the probability of transitioning from state 20, 21, 22, 23, or state 24 to state 9, 10, 11, 12, 13, 14, 15, or state 16 is equal to 0.125.

If the completion point of the n^{th} job is anywhere between 11 and 18 distance units, the next job does not start processing at the upstream boundary For instance, if the completion point of n^{th} job is at 11 distance units, the operator will move back towards the next job and start processing at one distance unit past the boundary. Similarly, if the completion point of n^{th} job is at 18 distance units, the operator will move back towards the next job and start processing at 8 distance units past the boundary.

Table 11: Case1 Transition Probability Matrix

	9	10	11	12	13	14	15	16	17	18	19(WO1)	20(WO2)	21(WO3)	22(WO4)	23(WO5)	24(WO6)
9	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0
10	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0
11	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0
12	0	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0
13	0	0	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0
14	0	0	0	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0
15	0	0	0	0	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0
16	0	0	0	0	0	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0
17	0	0	0	0	0	0	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0
18	0	0	0	0	0	0	0	0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
19(WO1)	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0
20(WO2)	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0
21(WO3)	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0
22(WO4)	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0
23(WO5)	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0
24(WO6)	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	0	0	0	0	0	0	0

Table 12: Case1 Steady State Probabilities

	State															
	9	10	11	12	13	14	15	16	17	18	19(WO1)	20(WO2)	21(WO3)	22(WO4)	23(WO5)	24(WO6)
Steady State Probability	0.041	0.048	0.056	0.066	0.077	0.090	0.105	0.115	0.084	0.077	0.069	0.059	0.048	0.035	0.020	0.010

The steady-state probabilities for this Markov chain are shown in Table 12. The average number of jobs that are completed at 19 distance units is 6.9, and the average number of jobs that are completed at 24 distance units is 1. There are six possible locations from which the operator can reach WO1 location whereas the operator can reach WO6 location only if the n^{th} job is completed at 18 distance units and the next job in the sequence is of type H.

The interpretation of the steady state probabilities in Table 12 is as follows. If an infinite random sequence of the eight job types (all with equal probability) is processed on the workstation, then for a random selection of 100 jobs in sequence, on average 24.1 of those jobs will cause a work overload occurrence. Thus the Markov chain model can be considered an approximation to reality where jobs are processed in groups of 100, with the first job of each random sequence of 100 jobs starting processing at the workstation boundary.

To check the accuracy of the approximation, the average number of work overloads obtained using the Markov chain (100 jobs * steady state probabilities) are compared to the average number of work overloads obtained computationally by processing random sequences of 100 jobs. The results are shown in Figure 19. The simulation average in Figure 19 refers to the average number of work overloads for all random launching sequences. The dataset of job processing times consisted of 1 million jobs. 100 jobs were randomly sampled from this dataset 100 times. Each set of 100 jobs was then randomly sequenced 1 million times, and each sequence was processed on the workstation. In total 100 million 100 job sequences were processed on the workstation. For each sequence the number of work overloads was recorded. The simulation minimum and maximum refers to the lowest and the highest number of work overloads observed amongst all random launching sequences respectively.

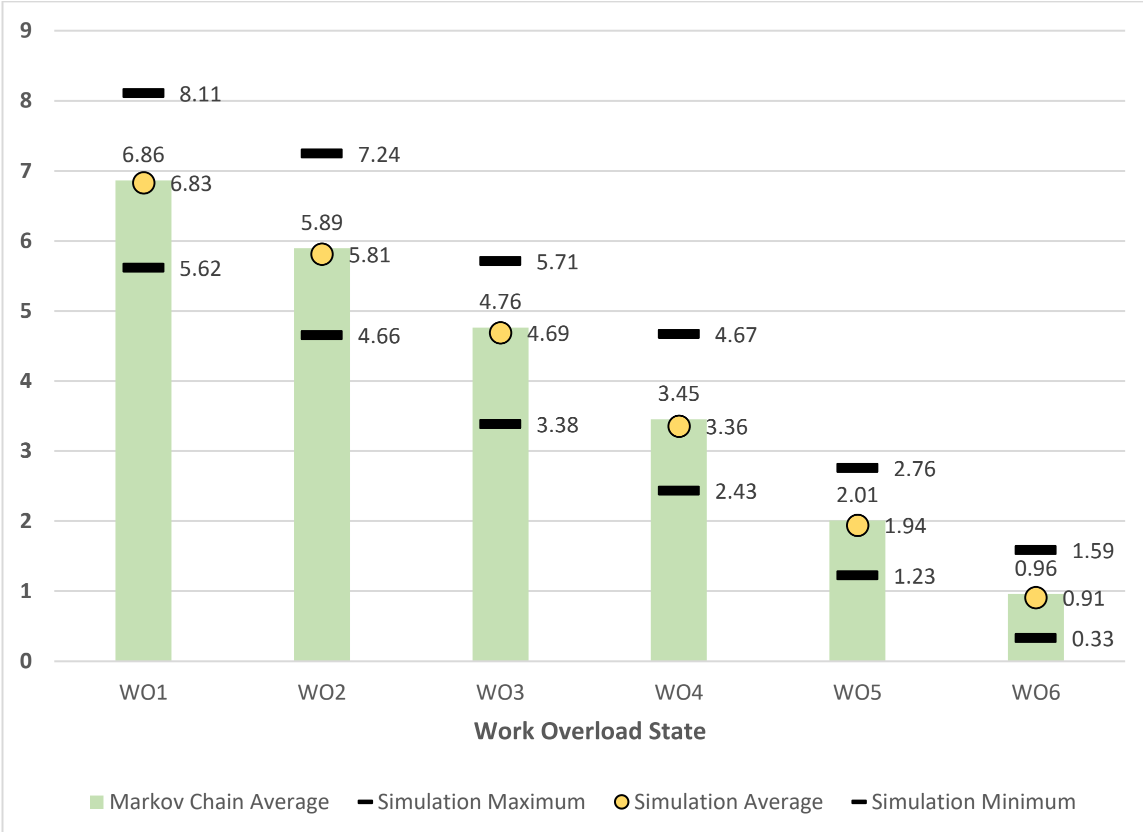


Figure 19: Case1: Severity of Work Overloads

A small difference in the Markov chain average and the simulation average is observed. It is expected that if the number of jobs in a sequence increases, the difference between the Markov chain averages and the simulation averages should decrease. This relationship is demonstrated in Figure 20. The vertical axis in Figure 20 is the "% Difference".

$$\%\ \text{Difference} = \frac{\text{Markov Chain Average} - \text{Simulation Average}}{\left(\frac{\text{Markov Chain Average} + \text{Simulation Average}}{2}\right)}$$

When the number of jobs is 10, the difference is 16.6% and as the number of jobs is increased the difference decreases and is close to zero when the number of jobs is 1 million. This relationship between the two is explained in more detail in Section 4.4.

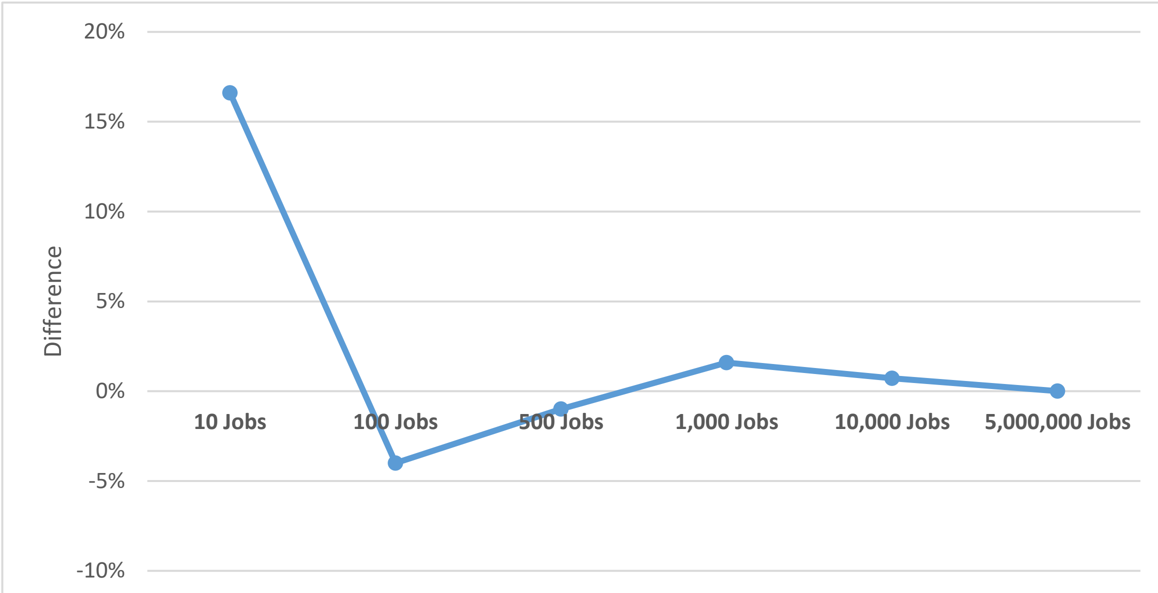


Figure 20: Case1: Effect of the Number of Jobs on Markov Chain Accuracy

<u>Case2</u>

Consider a similar mixed model assembly workstation but with different job configuration as shown in Table 13. The cycle time is ten time units and 100 jobs are to be assembled. The job set has five different job types that are in the same proportion. The discrete time Markov chain has 13 states of which three states 19(WO1), 20(WO2) and 21(WO3) are work overload states.

Table 13: Case2 Job Composition

Job Type	Processing Time (Time Units)	Demand Fraction
A	9	0.2
B	10	0.2
C	11	0.2
D	12	0.2
E	13	0.2

The Transition Probability matrix and the steady state probabilities are presented in Table 14 and Table 15, respectively. The Markov chain can transition into the least severe work overload state from states 16, 17 and 18 (distance units). The most severe work overload state can be reached only if the n^{th} job is completed at 18 distance units and the next job is of type E. Therefore, there are more jobs that are completed at 19 distance units as compared to those completed at 21 distance units.

The Markov chain results are compared with the simulation results in the Figure 21. The average number of work overload occurrences for Case2 is 10.8 with the average number of the least severe work overloads being 5.7 and the average number of most severe work overloads being 1.57.

Table 14: Case2 Transition Probability Matrix

	9	10	11	12	13	14	15	16	17	18	19(WO1)	20(WO2)	21(WO3)
9	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0	0	0
10	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0	0	0
11	0	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0	0
12	0	0	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0
13	0	0	0	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0
14	0	0	0	0	0.2	0.2	0.2	0.2	0.2	0	0	0	0
15	0	0	0	0	0	0.2	0.2	0.2	0.2	0.2	0	0	0
16	0	0	0	0	0	0	0.2	0.2	0.2	0.2	0.2	0	0
17	0	0	0	0	0	0	0	0.2	0.2	0.2	0.2	0.2	0
18	0	0	0	0	0	0	0	0	0.2	0.2	0.2	0.2	0.2
19(WO1)	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0	0	0
20(WO2)	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0	0	0
21(WO3)	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0	0	0

Table 15: Case2 Steady State Probabilities

	State												
	9	10	11	12	13	14	15	16	17	18	19(WO1)	20(WO2)	21(WO3)
Steady State Probability	0.0412	0.0566	0.0768	0.1010	0.1214	0.1018	0.1079	0.1075	0.0990	0.0786	0.0570	0.0355	0.0157

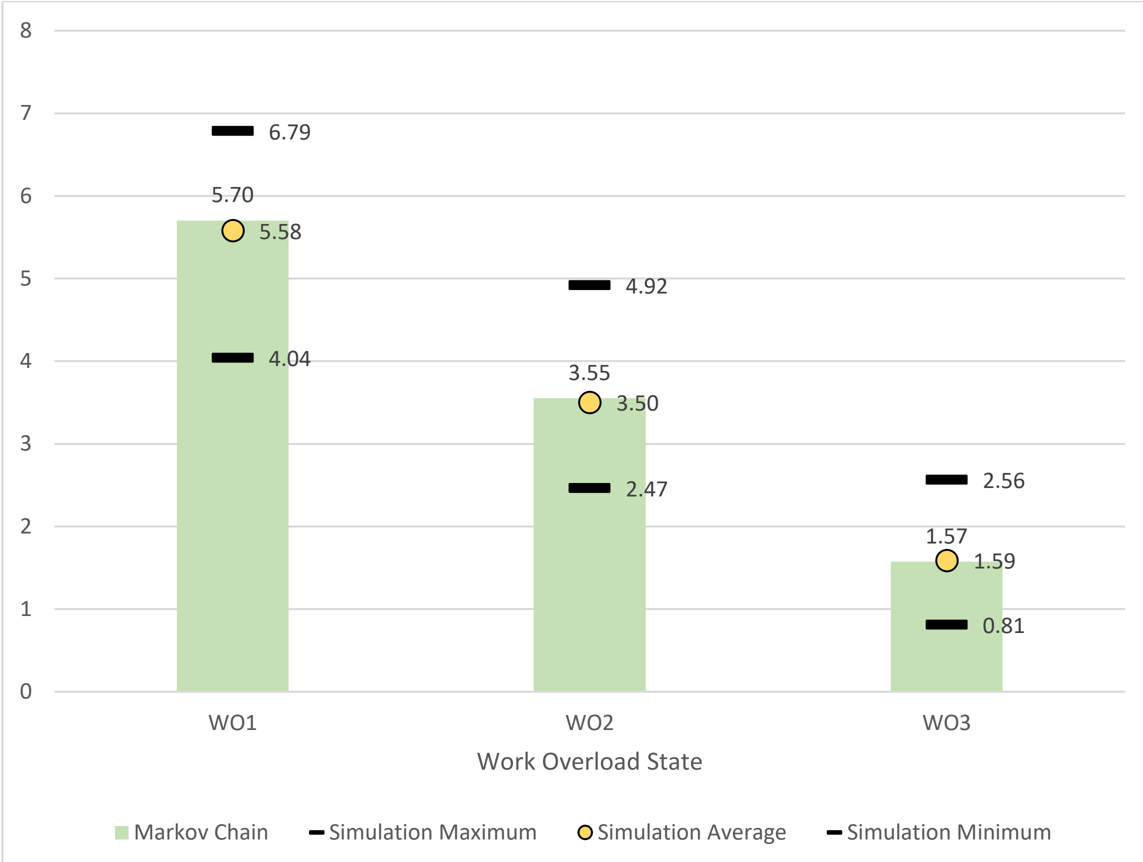


Figure 21: Case2: Severity of Work Overloads

<u>Case3</u>

The Case1 and the Case2 have multiple job types in equal proportions, and each job type has fixed processing times. In Case3 there is a similar workstation configuration, but Case3 has three job types having different demands, and each job type has uncertainty in processing times as shown in Table 16. For instance, the job type A can have any integer processing time between four and six time units. Here, the probability of a randomly selected job having processing time four time units is the probability of the job having a processing time of four time units given that the selected job is of type A (0.4*0.333 = 0.133). Similarly, the probability of a randomly selected job having a processing time of 10 time units is 0.05. The transition probability matrix for this case is presented in Table 17. The cycle time is ten time units, the number of jobs to be launched are 100 and the length of the workstation is 18 time units. The discrete time Markov chain consists of 22 states, of which, seven are work overload states.

The steady state probabilities are shown in Table 18. The average number of jobs (out of 100) with work overloads is 14.1. The Markov chain results are compared with the simulation results in the Figure 22.

Table 16: Case3 Job Composition

Job type	Processing Time (TU)	Demand Fraction (%)
A	Discrete Uniform Distribution (4,6)	0.4
B	Discrete Uniform Distribution (8,11)	0.2
C	Discrete Uniform Distribution (15,17)	0.4

Table 17: Case3 Transition Probability Matrix

	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19(WO1)	20(WO2)	21(WO3)	22(WO4)	23(WO5)	24(WO6)	25(WO7)
4	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
5	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
6	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
7	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
8	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
9	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
10	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
11	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0
12	0	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0
13	0	0	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0
14	0	0	0	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0
15	0	0	0	0	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0
16	0	0	0	0	0	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0
17	0	0	0	0	0	0	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0
18	0	0	0	0	0	0	0	0	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333
19(WO1)	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
20(WO2)	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
21(WO3)	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
22(WO4)	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
23(WO5)	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
24(WO6)	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0
25(WO7)	0.13333	0.13333	0.13333	0	0.05	0.05	0.05	0.05	0	0	0	0.13333	0.13333	0.13333	0	0	0	0	0	0	0	0

Table 18: Case3 Steady State Probabilities

	State																					
	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19(WO1)	20(WO2)	21(WO3)	22(WO4)	23(WO5)	24(WO6)	25(WO7)
Steady State Probabilities	0.0727	0.0823	0.0872	0.0177	0.0373	0.0477	0.0593	0.0717	0.0366	0.0244	0.0144	0.0876	0.0977	0.0982	0.0238	0.0112	0.0169	0.0266	0.0378	0.0293	0.0163	0.0032

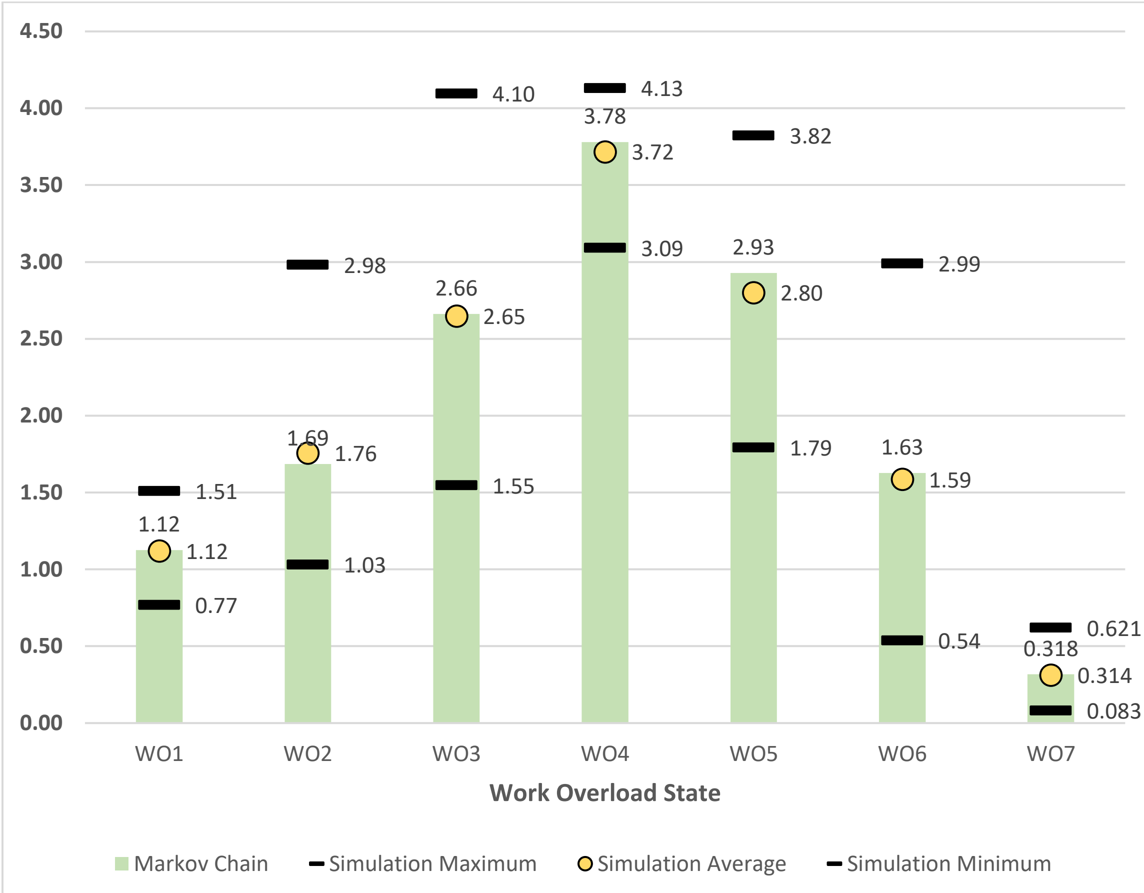


Figure 22: Case3: Severity of Work Overloads

<u>Case4</u>

Case4 consists of three sub-cases where some sub-cases have non-integer processing times. Case4A, Case4B and Case4C having processing times between 9 and 11 time units. The length of the workstation is 12 time units, the cycle time is 10 time units and the number of jobs is 100.

Case4A jobs have integer processing times. There are three job types that are equal in proportion with processing times of 9, 10 and 11 time units. The Markov chain for Case4A consists of a single work overload state (13(WO1)) and four non work overload states (9, 10, 11 and 12). Figure 23 presents the average number of jobs out of 100 that cause a work overload.

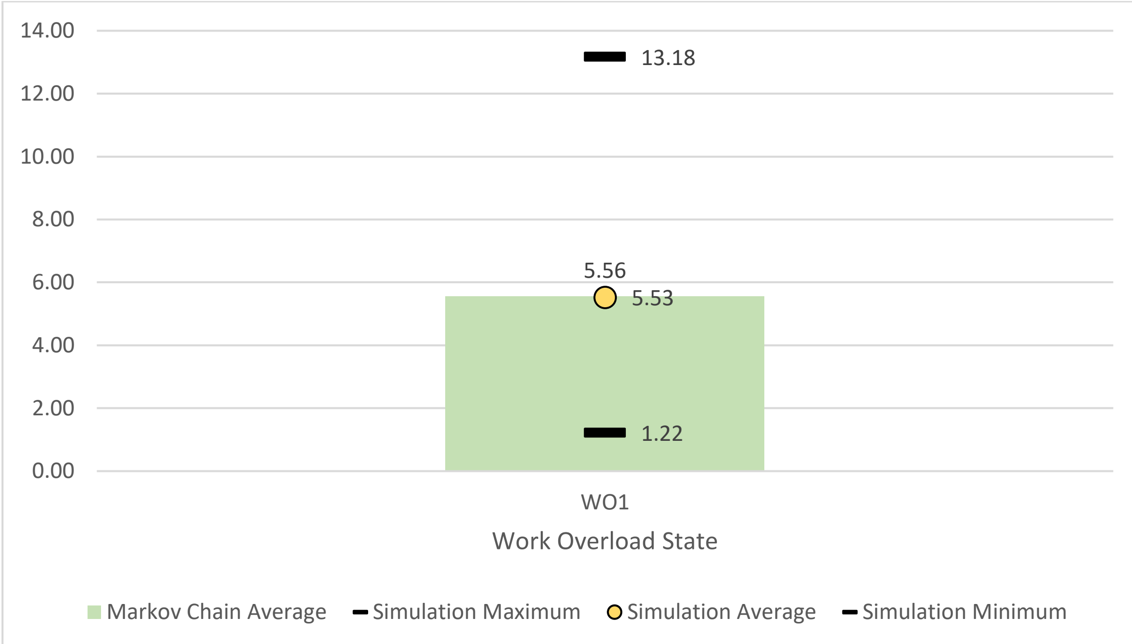


Figure 23: Case4A: Severity of Average Number of Work Overloads (100 Jobs)

Case4B jobs have processing times obtained from a uniform distribution between 9 and 11 time units. The processing times are rounded up to nearest tenth. The processing time of each of the job can be any of the following 9, 9.1, 9.2, 9.3... 10.9, 11. The discretization of the processing times to the nearest tenth results in a Markov chain with 41 states of which 10 states are work overload states. Jobs completed at 12.1 distance units are considered the least severe (WO1) work overload, and those completed at 13 distance units are considered the most severe (WO10) work overload. Figure 24 presents the average number of work overloads of all severities (for 100 jobs).

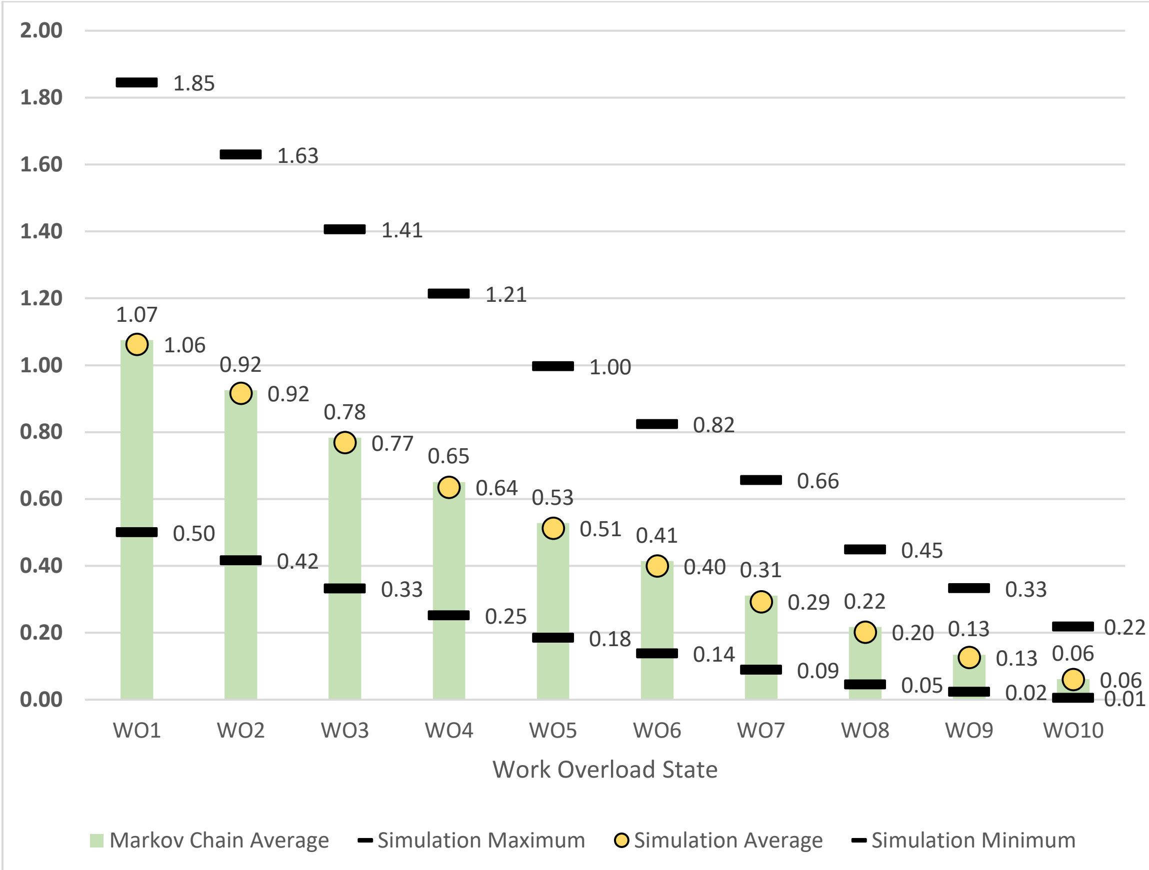


Figure 24: Case4B: Average Number of Work Overloads (100 Jobs)

Case4C jobs have processing times that are rounded up to the nearest hundredth so that the processing times could be any of the following 9, 9.01, 9.02, 9.03…, 10.99, 11. This results in a Markov chain with 401 states, of which 100 are work overload states. The average number work overload jobs out of 100 jobs obtained in all three cases are compared in Figure 25. Since the work content and the available time is the same across all three cases, the average work overloads should be similar. Note that the Case4A has only one work overload state and therefore, the average for the total number of work overloads is exactly the same as the average for the WO1 overload state.

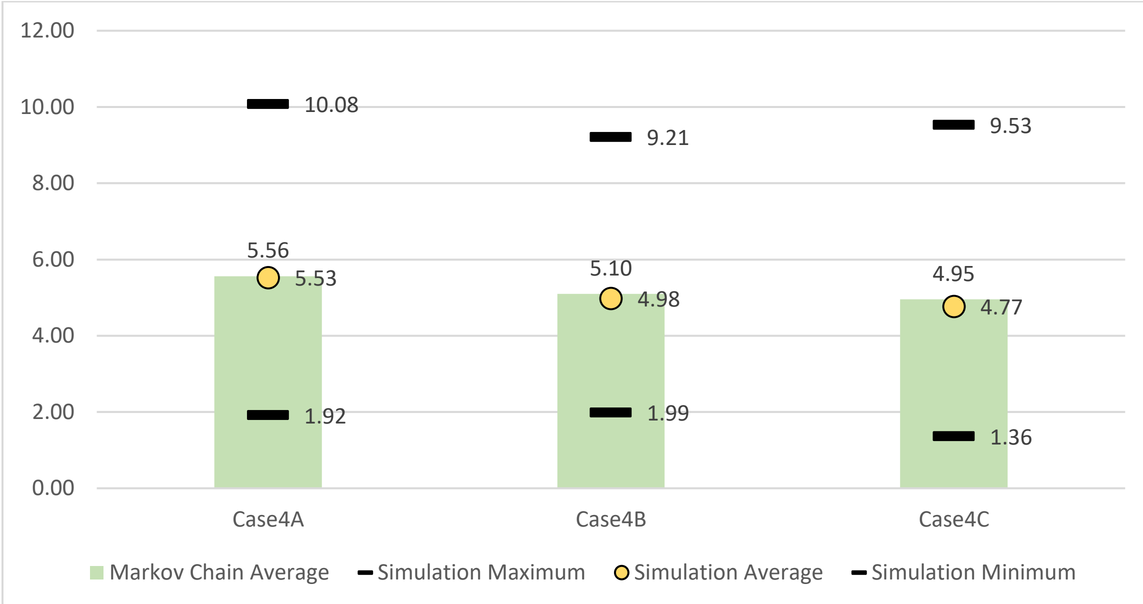


Figure 25: Comparing the Average Number of Work Overloads for 100 Jobs

Based on the simulation results observed in this section, the results produced by the Markov chain model are sufficiently accurate to predict work overload occurrences in random sequences. Additionally, as the number of jobs increases, the accuracy of the results also increases. The results also show that it is possible to model the MMAL that have continuous processing times of jobs by discretizing the process times. The next section presents an examination of the probability distribution curve of the number of work overloads.

4.2 Probability Distribution for the Number of Work Overload Occurrences

In this section the probabilities obtained for the number of workload occurrences in a finite sequence of jobs using the Markov chain model are compared with probabilities from a C++ simulation.

Consider Case1 mentioned in the previous section except now with 1000 jobs in a random sequence. The transition probability from a non-work overload state to a work overload state in the macro-state Markov chain (P(WO|NWO)) is calculated as shown in Section 3.1.2.1 and is 0.315, and the range for the total number of overload occurrences is zero to 500 (for 1000 jobs). Using the formula (10) presented in Section 3.1.3, the probabilities for the number of work overload occurrences can be calculated for a 1000 job random sequence. Figure 26 presents the probability distribution of the total number of work overload occurrences for Case1 with 1000 jobs. The most likely number of overload occurrences is 240 with a probability of occurrence being 0.0408. The probability distribution is "bell shaped" for Case1 with extremely low probabilities for a large portion of the 0 to 500 work overload occurrence range.

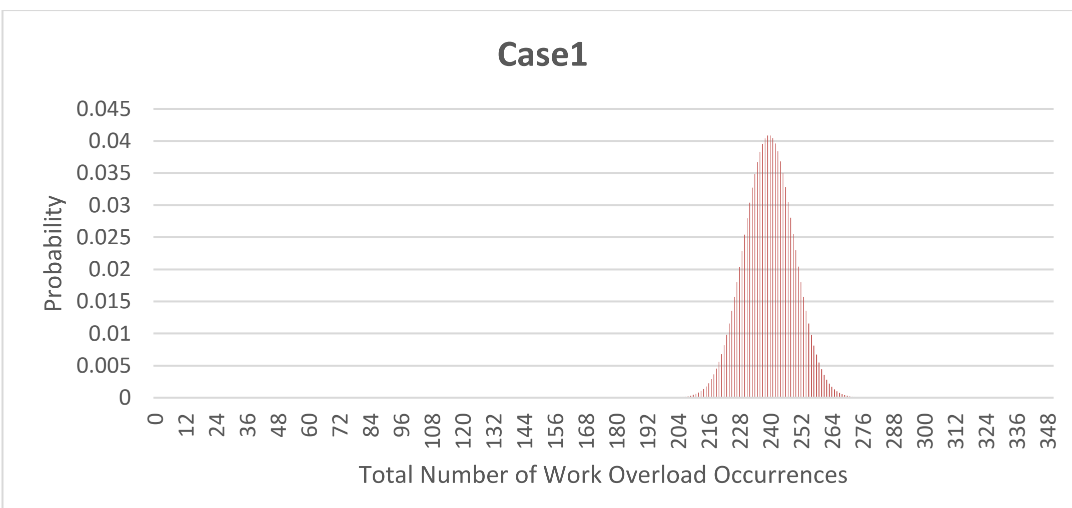


Figure 26: Case1 - Probability Distribution of the Total Number of Workload Occurrences

Figure 27 presents the probability distributions for specific work overload states that occur in Case1 using the method in Section 3.1.3, and compares them with simulation. The WO1 work overload state occurs between 48 and 92 times (in a sequence of 1000 jobs) with the most likely number being 68 that has an estimated probability of 0.0536. Similarly state WO6 occurs between 0 and 22 times with 9 being the most likely with a probability of 0.1308. The probabilities obtained using the Markov chain model are very close to the probabilities obtained using the simulation. It is also evident from Figure 26 and Figure 27 that it is unlikely for Case1 to have zero work overloads.

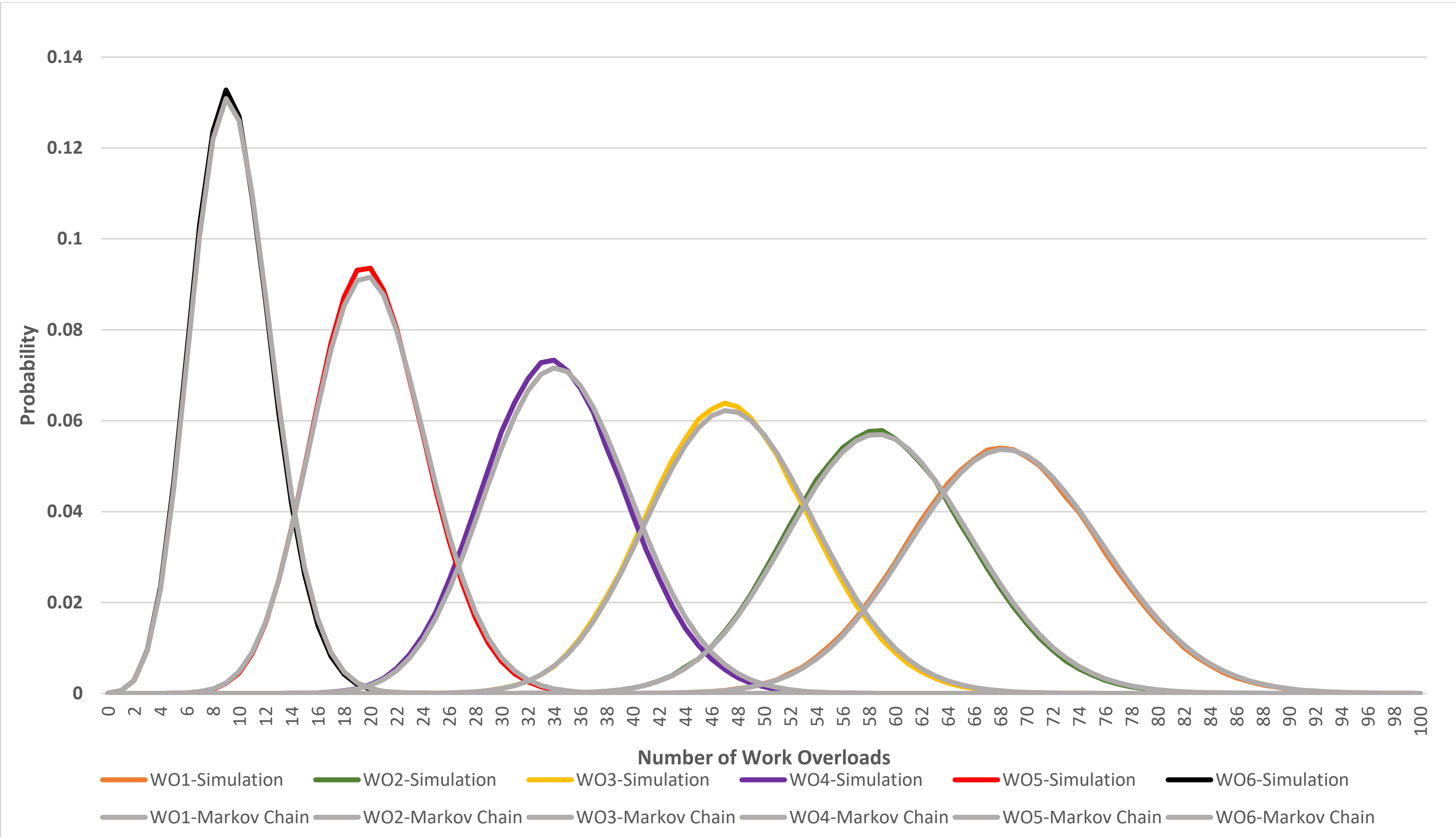


Figure 27: Case1: Probability Distribution of Specific Workload States

A new test case, Case5 is introduced that consists of five job types which are equally likely, and have processing times of 7, 8, 9, 10 and 11 time units. The length of the workstation is 18 time units, the cycle time is 10 time units and the number of jobs in a sequence is 1000. The Markov chain for this case has 13 states of which one is a work overload state. The steady state probability of the work overload state is 0.0000042. The job type with an 11 time unit processing time is the only job type that may cause a work overload, and hence it is unlikely that work overloads occur in a random job sequence.

The transition probabilities of the macro Markov chain for all the cases are shown in Table 19, and the probability distributions for the total number of work overload occurrences are shown in Figure 28**.**

Table 19: Macro Markov Chain Transition Probabilities

Case	P(WO|NWO)	P(NWO|NWO)
Case1	0.315	0.685
Case2	0.121	0.879
Case3	0.164	0.836
Case4A	0.059	0.941
Case5	0.000004	0.999996

The probability mass for Case1, Case2, Case3, and Case4 is far from zero, and therefore it is unlikely that a random launching sequence will result in zero work overloads. However, for Case5, the probability of zero work overloads is 0.9958. For such cases, job sequence optimization is not necessary since most random launching sequence will most likely yield zero overloads.

If the probability of zero work overloads is high then job sequencing optimization is not needed, however the workstation may be "overdesigned", which is a poor use of limited resource (primarily space). If the estimated chance of work overload occurrences is high, then job sequencing optimization may reduce but still result in an unacceptable number of work overload occurrences. In Section 4.4. the Markov chain model is used to predict whether sequencing optimization can give zero overloads so that workstation resources are being used efficiently. The following section explores possible Markov chain model inaccuracies due to necessary Markov chain model assumptions

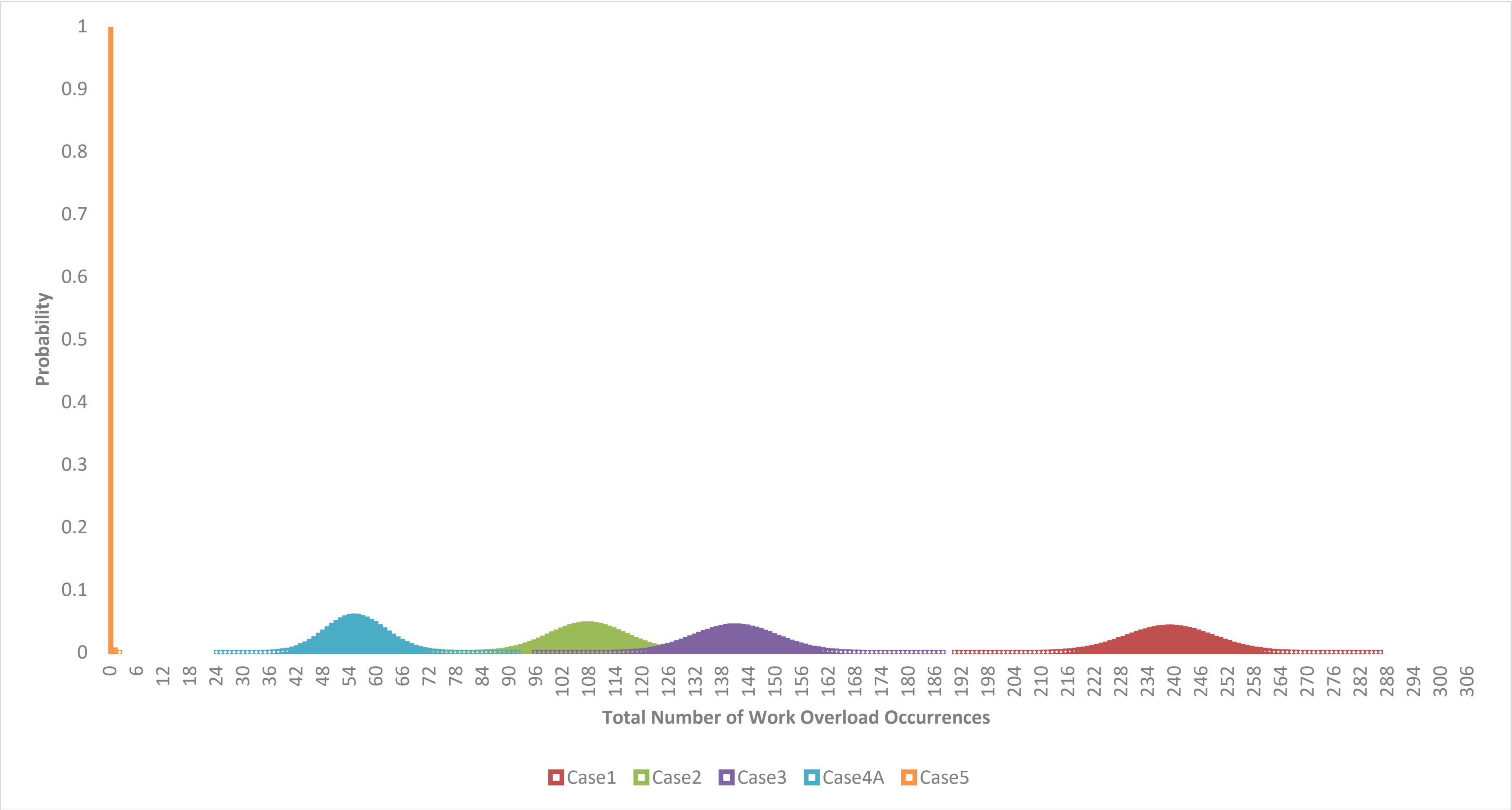


Figure 28: Multiple Cases - Probability Distribution of the Total Number of Workload Occurrences

4.3 Constant Demand Fraction Assumption

In the Markov Chain model the probability of randomly selecting job type m (d_m) remains constant but for a finite set of jobs this probability varies each time a job is launched. For instance, consider a job set with 100 jobs and two job types A and B in equal proportion. Table 20 shows the changes in job type probabilities after successive job launches. The first and the second column shows the number of type A and type B jobs in the job set. The third column shows the total number of jobs remaining. The fourth and the fifth columns show the probabilities of randomly selecting an A and B type job respectively. The last column shows the type of job selected and launched.

Table 20: Probability Variation in a Job Set

A	B	Total Number of Jobs Remaining	P_A	P_B	Job Type That is Launched
50	50	100	0.5	0.5	A
49	50	99	0.495	0.505	B
49	49	98	0.5	0.5	B
49	48	97	0.505	0.495	B
49	47	96	0.510	0.490	

As a second example, consider a set of 100 jobs with a single job having a processing time exceeding the cycle time (and the rest having processing times less than the cycle time). The number of work overload occurrences for all sequences is zero. However, the Markov chain will estimate a non-zero probability of work overloads because there will always be a 0.01 probability of a randomly selecting the job type with a processing time greater than the cycle time.

This assumption of stationary (constant) job type probabilities for the Markov chain model results in inaccuracies, and the magnitude of the inaccuracies depends on the number of jobs in a sequence. The effect of the number of jobs in a sequence on the accuracy of the probability distribution for the number of work overload occurrences (computed from the macro state Markov chain transition probabilities) is illustrated in this section.

Consider a new case, Case6 consisting of multiple job types having proportions as shown in Table 21. The cycle time is 10 time units, and the workstation length is 15 distance units.

Table 21: Case6 Job Composition

Job Type	Processing Time (Time Units)	Demand Fraction
A	9	0.1
B	10	0.1
C	11	0.2
D	12	0.2
E	13	0.2
F	14	0.2

It can be difficult to always find a provably optimal job launching sequence computationally because the job sequencing problem is a difficult combinatorial optimization problem (Xiaobo et al. (1997)). Instead, a pairwise exchange method is used to search for a sequence that minimizes the number of work overload occurrences for various test cases. The code for the pairwise exchange was written in C++. The minimum number of work overload occurrences found computationally are compared with the probability distribution for the number of work overload occurrences. The term 'Minimum' used henceforth refers to the lowest number of work overload occurrences obtained in 100 iterations.

Figure 29 shows the probability distribution for the number of work overload occurrences for Case6 and 100 jobs. The minimum number of work overload occurrences found using the pairwise exchange search algorithm is 24. According to the Markov chain result, the probability of 23 work overloads is 0.038, i.e. 0.038 * 100! launching sequences can give 23 overloads, which is not possible (assuming that the neighborhood search algorithm gives the best possible solution). Therefore, the probabilities for $l < 24$ are the imprecise results produced by the Markov Chain model if the job set is finite. The orange column in the Figure 29 shows the minimum number of work overload occurrences.

Figure 29: Case6 Probability Distribution for the Number of Work Overload Occurrences for 100 Jobs

Figure 30 to Figure 34 show the probability distribution for the number of work overload occurrences for the same case when the number of jobs in a sequence is increased in steps of 100.

Figure 30: Case6 Probability Distribution for the Number of Work Overload Occurrences for 200 Jobs

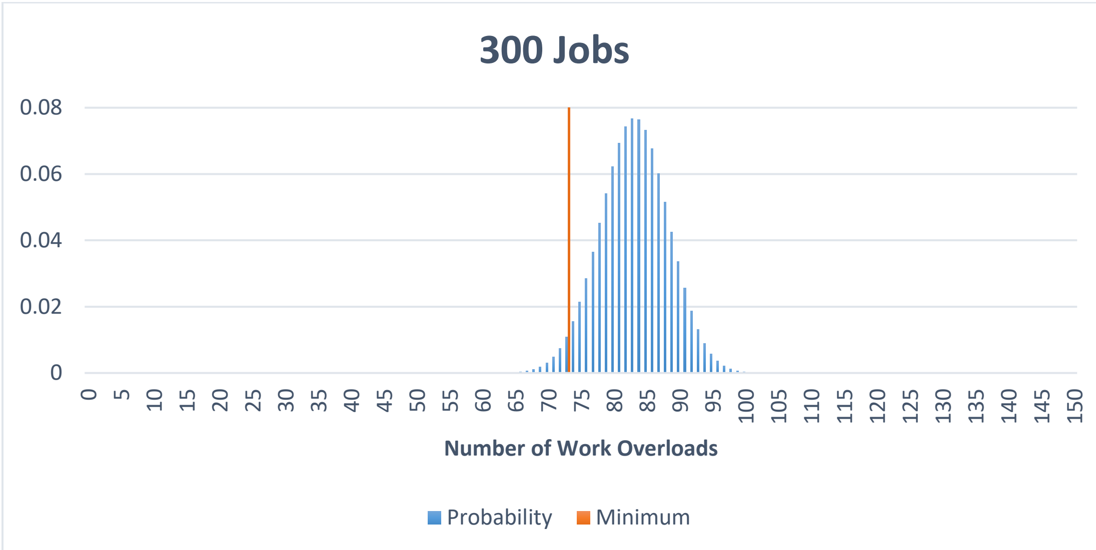


Figure 31: Case6 Probability Distribution for the Number of Work Overload Occurrences for 300 Jobs

Figure 32: Case6 Probability Distribution for the Number of Work Overload Occurrences for 400 Jobs

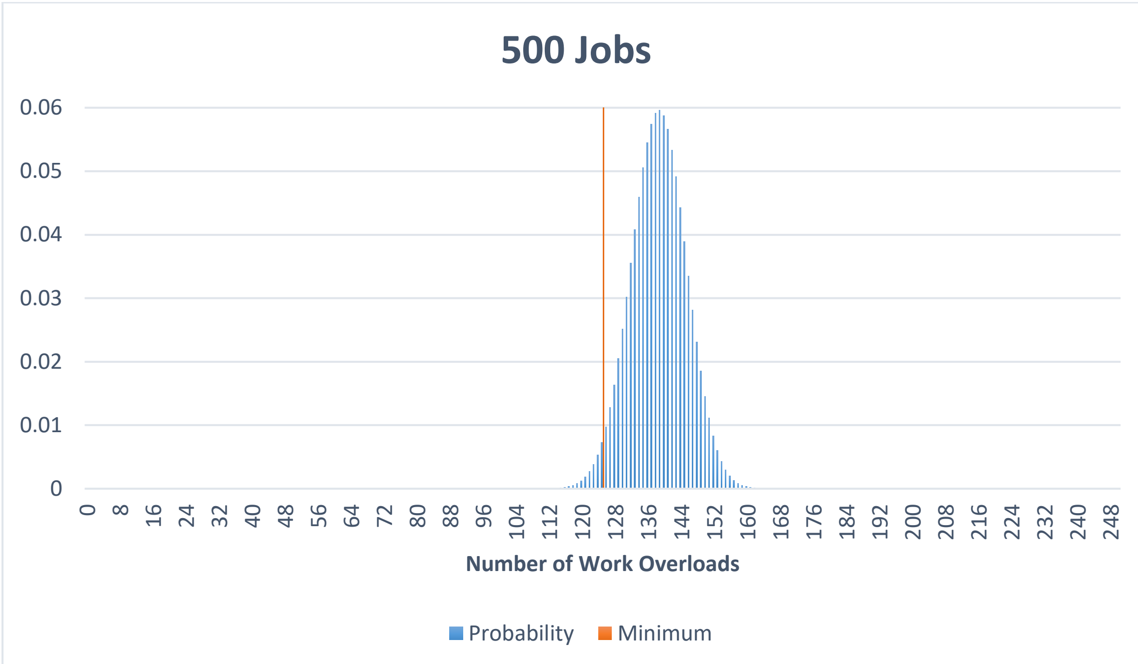


Figure 33: Case6 Probability Distribution for the Number of Work Overload Occurrences for 500 Jobs

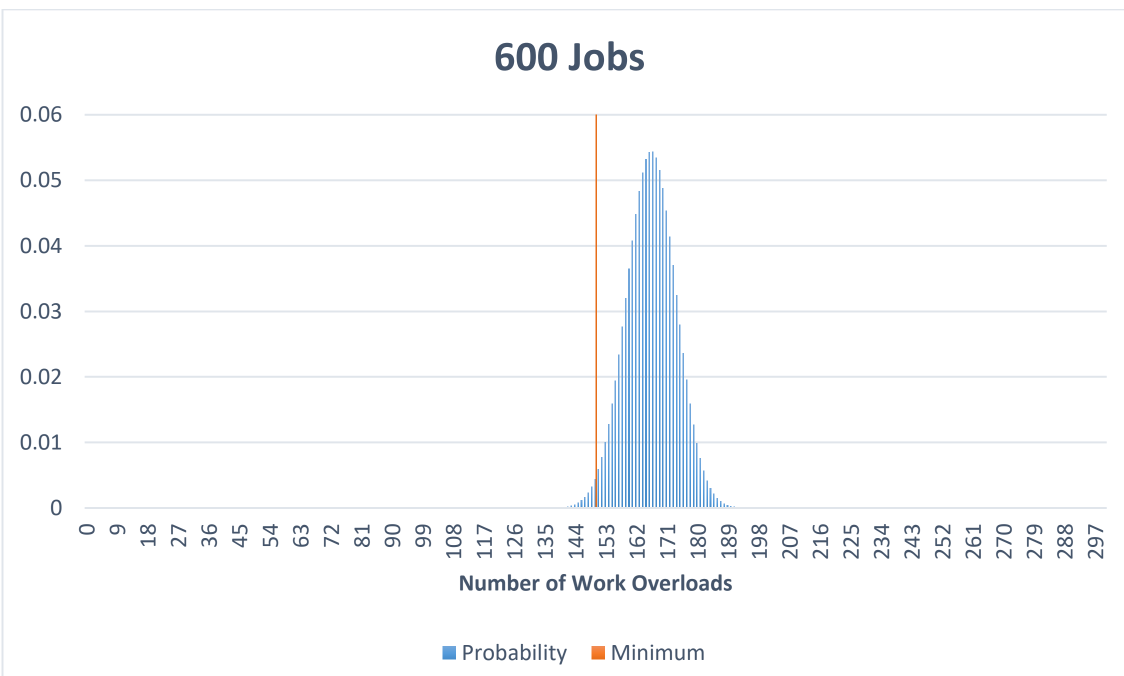


Figure 34: Case6 Probability Distribution for the Number of Work Overload Occurrences for 600 Jobs

The change in job selection probability each time a job is launched is small for large job sets. Therefore, it is observed that as the number of jobs in a sequence increases the probability mass to the left of the orange bar decreases. The minimum number of work overload occurrences obtained computationally for 100 to 600 jobs along with the cumulative probability of the total number of work overload occurrences less than the minimum number of work overload occurrences is shown in Table 22.

Table 22: Case6 Minimum Work Overload Occurrences and Cumulative Probability

Number of Jobs	Minimum Work Overload Occurrences	Cumulative Probability
100	24	0.076
200	48	0.030
300	73	0.020
400	98	0.013
500	125	0.017
600	150	0.011

Four more cases were examined. The processing times of different job types, their demand fractions and the workstation lengths of these test cases are shown in Table 23. The cycle time for all these cases is 10 time units.

Table 23: Job Compositions of Multiple Cases

Case	Processing Time (Time Units)	Demand Fraction	Workstation Length (Distance Units)
Case7	8	0.5	18
	14	0.5	
Case8	9	0.2	20
	10	0.2	
	12	0.2	
	14	0.2	
	17	0.2	
Case9	8	0.1	20
	9	0.1	
	10	0.1	

Case	Processing Time (Time Units)	Demand Fraction	Workstation Length (Distance Units)
	11	0.1	
	12	0.1	
	13	0.1	
	14	0.1	
	15	0.1	
	16	0.1	
	17	0.1	
Case10	8	25	13
	10	25	
	12	50	

The probabilities of the number of work overload occurrences are calculated, and the sum of probabilities of work overloads less than the minimum number of work overload occurrences (found computationally) for all the cases is presented in Figure 35. For instance, the area under the probability curve to the left of the orange column for the Case6 when the number of jobs is 100 is 0.076 (refer to Figure 29). This probability mass gradually reduces to 0.005 when the number of jobs is increased to 600 (refer to Figure 34). The variation in this probability mass for multiple cases is presented in Figure 35.

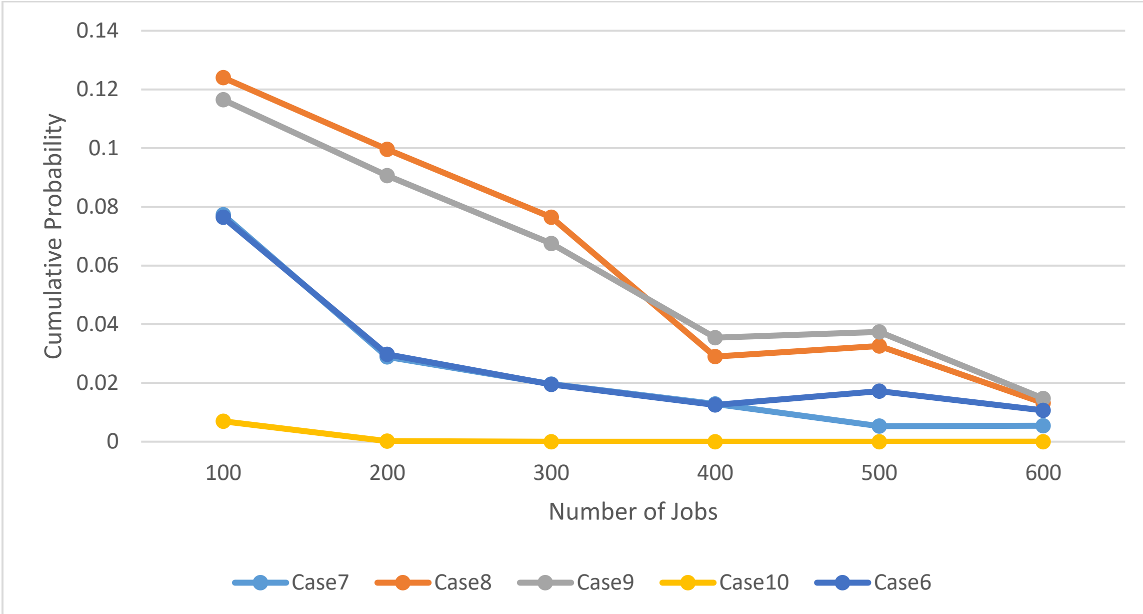


Figure 35: Multiple Cases – Cumulative Probability of the Total Number of Work Overload Occurrences

In general, a decreasing trend in the cumulative probability mass exists as the number of jobs in a sequence is increased.

4.4 Supporting Workstation Design Decisions

The models developed can be used to support initial workstation length and cycle time decisions. If the Markov chain model predicts a high probability of zero work overloads in random sequences then the workstation is likely too long for the cycle time for a predicted mix of job types. If the Markov chain predicts a very large number of work overloads then even with sequencing optimization, a large number of work overloads will occur and the workstation length should be increased. This section presents an application of the Markov chain model that helps identify when a given workstation length, cycle time, and job mix results in likely work overloads for random sequences, but zero work overloads when sequencing optimization is applied.

The available time and the required time to complete all jobs at a workstation is as follows:

$$\text{Available Time} = (\text{Cycle Time} * \text{Total Number of Jobs}) + (\text{Workstation Length} - \text{Cycle Time})$$

$$\text{Work Content} = \sum_{\text{All Jobs}} \text{Processing Time}$$

$$\text{Excess Time} = \text{Available Time} - \text{Work Content}$$

Negative Excess Time implies that the available time is insufficient to process all tasks within the provided time and therefore, a workstation and job set combination having negative Excess Time will not have any sequence with zero work overloads. For a job set having positive Excess Time there may or may not exist a job sequence that gives zero work overloads. A workstation and different types of job sets can be categorized into three as shown in Table 24, where the second column shows the Excess Time for the job set and the last column shows the minimum number of work overload occurrences obtained using sequencing optimization (the neighborhood search algorithm in this research).

Table 24: Job Set Categories

Category	Excess Time	Minimum
1	≥0	0
2	≥0	>0
3	<0	>0

The objective is to differentiate cases between the categories 1 and 2 without having to perform sequencing optimization. New test cases that fall into either Category 1 or Category 2 were generated. The composition of the job sets and the lengths of the workstations are shown in Appendix 2. All cases that are considered until this point except Case5 fall into Category 3. The parameters that are varied in the test cases are the number of job types, processing times of the jobs, demand fractions and the workstation length. The cases with the same number and different letter differ in only one parameter, e.g., workstation length. The cycle time is 10 time units, and the number of jobs is 300 for all of these cases. For visualization purpose, the test cases are represented on a three dimensional space having workstation length, number of job types and work content as the axes. Figure 36 shows a perspective view of the three-dimensional test case space and Figure 37 to Figure 39 show the orthographic projections of it. Work content is a function of two parameters, demand fraction and processing times of jobs. The relationship between these two parameters is shown in Figure 40. The test cases consist of numerous combinations of the demand fractions and processing times that are well spread out in the available space. There is no lower limit for job processing times however the maximum processing time of a job type in a job set should be such that the operator should be able to completely process it within the workstation boundaries if the job is operated from the upstream boundary. In all the test cases there is at least one job type that has a processing time more than the cycle time otherwise there would be no work overloads.

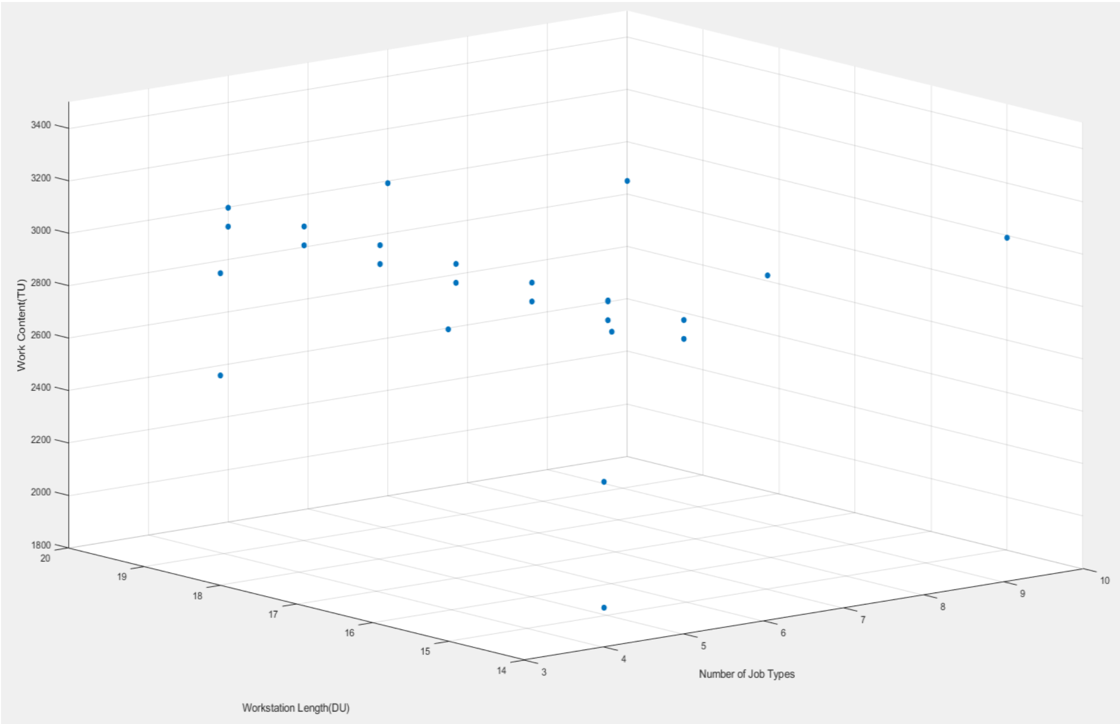


Figure 36: Three-Dimensional Visualization of Test Cases (300 Jobs)

The lower limit for the workstation length is the largest processing time amongst all job types, and the upper limit is twice the cycle time. Most of the cases have low Excess Time. If the Excess Time is large then the test case would most likely be of category 1, but it becomes difficult to differentiate between the two cases when the Excess Time is small. Therefore most of the cases have work content close to the available time.

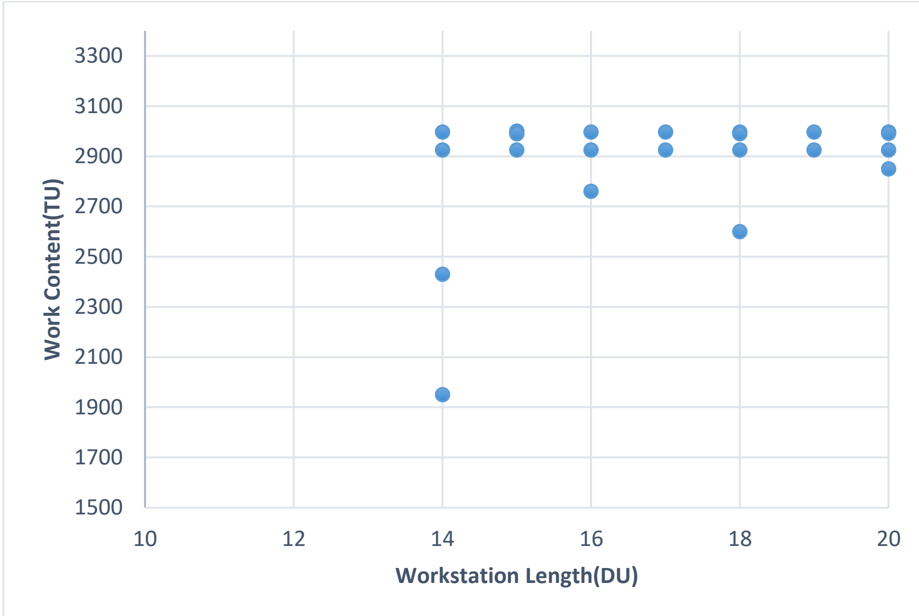


Figure 37: Orthographic Projection – I (300 Jobs)

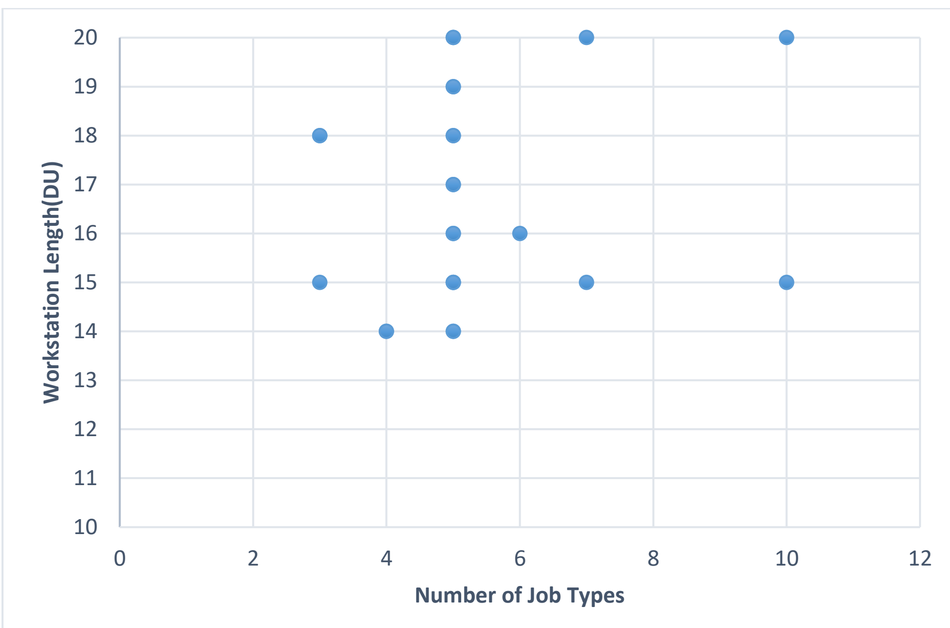


Figure 38: Orthographic Projection – II (300 Jobs)

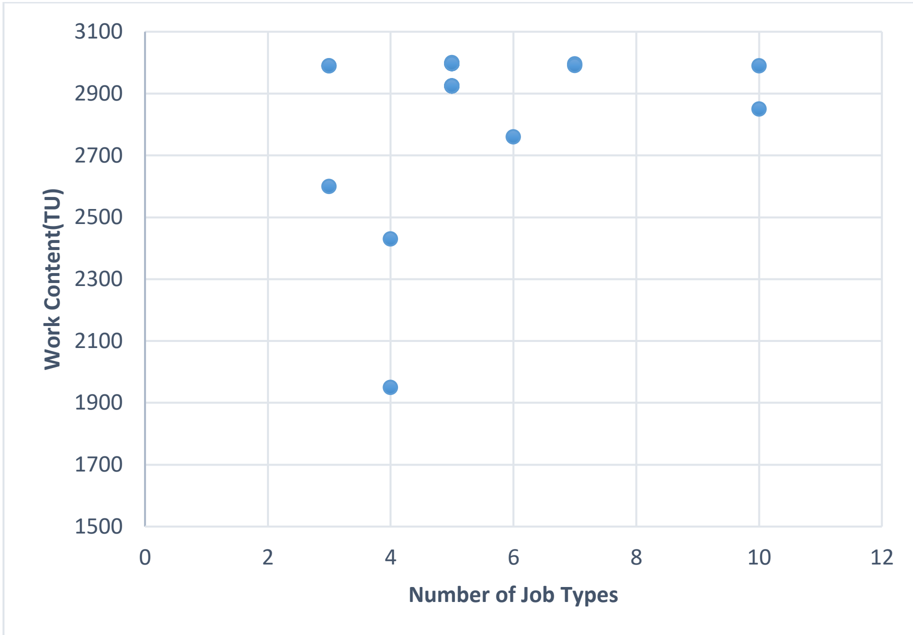


Figure 39: Orthographic Projection – III (300 Jobs)

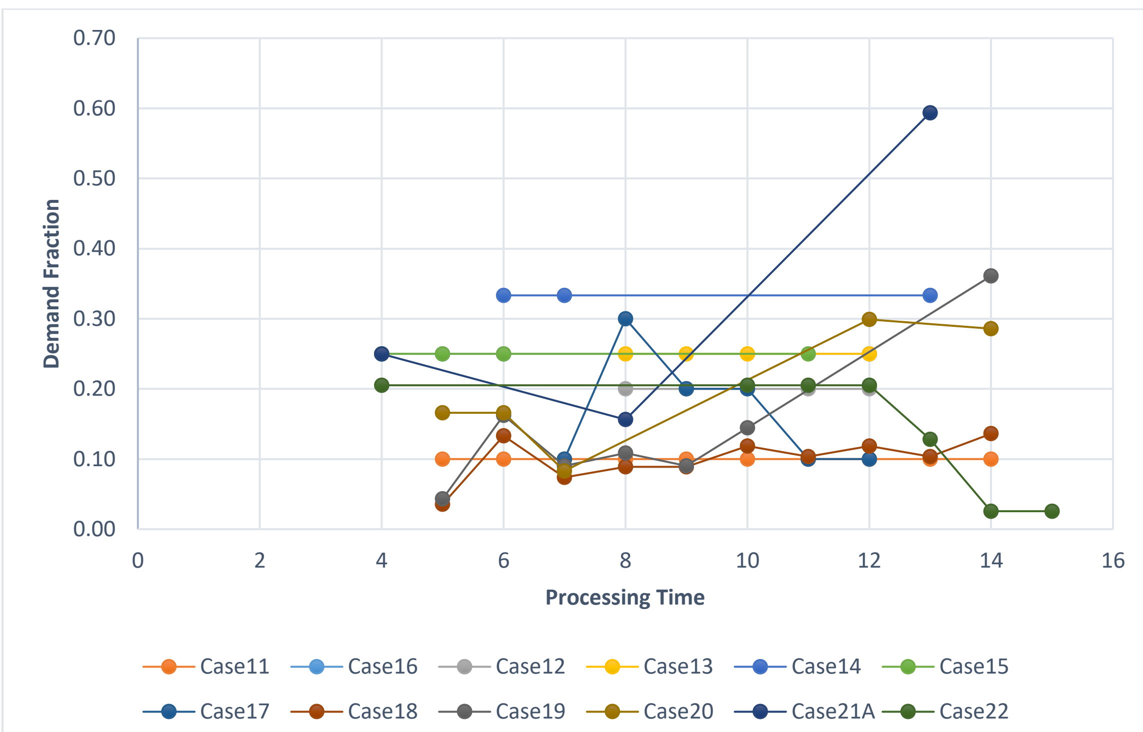


Figure 40: Demand Fractions of Job Types

The figures indicate that the test cases have variability in their parameters and are not concentrated at a particular location in the three-dimensional space.

The Excess Time, the minimum number of work overload occurrences discovered computationally, and the probability P($WO|NWO$) (from the macrostate Markov chain) is calculated for all test cases and is plotted on the graph in Figure 41. The vertical axis on the left represents Excess Time that is indicated by a green column for each test case. The same axis also represents the minimum number of work overload occurrences shown with red columns. The vertical axis on the right represents the value for P(WO|NWO) and is indicated by a blue dot, and test cases are represented on the horizontal axis. Cases can be categorized into Category 1 or Category 2 based on their P(WO|NWO) values as shown in Table 25. It is observed that Category 1 cases have low P(WO|NWO) as compared to the Category 2 cases. The maximum value of P(WO|NWO) in Category 1 cases is 0.0422, and the minimum value of P($WO|NWO$) in Category 2 cases is 0.0566. An arbitrary cutoff value of P($WO|NWO$) = 0.05 can be used to differentiate between cases.

To check if this cutoff value is applicable for a different number of jobs, similar tests are conducted by increasing the number of jobs to 500, and the results are shown in Figure 42. P($WO|NWO$) remains the same because it is independent of the number of jobs. The number of job types and workstation length remain the same but the Excess time and the minimum number of work overload occurrences change for all test cases. Refer to Appendix 3 to Appendix 6 for the three-dimensional visualization and its orthographic projections of the test cases. Figure 42 shows that even with an increased number of jobs, if P($WO|NWO$) is more than 0.05, then the minimum number of work overloads is greater than zero and otherwise it is zero.

In conclusion, based on the test cases studied, if P(WO|NWO) is large (more than 0.05), the job sequencing problem will not give a sequence with no work overloads. To best utilize available workstation space a value for P(WO|NWO) that is close to 0.05 can be used as a guideline.

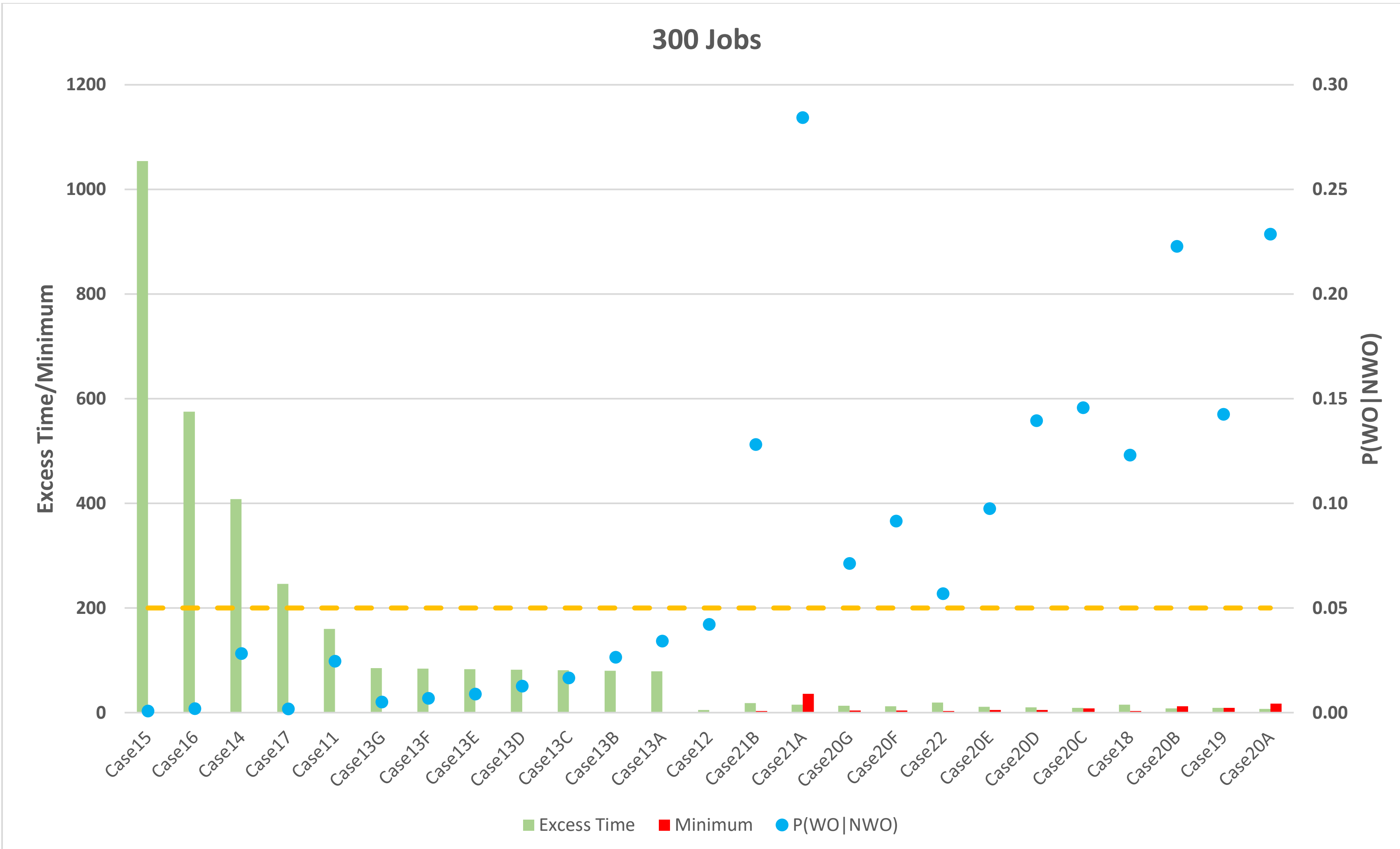


Figure 41: Excess Time, Minimum and P(WO|NWO) of Multiple Test Cases (300 Jobs)

Table 25: Multiple Category 1 and Category 2 Cases

Category	New Case Number	Excess Time	Minimum	P(WO|NWO)
1	Case11	160	0	0.0245
	Case12	5	0	0.0422
	Case13A	79	0	0.0341
	Case13B	80	0	0.0264
	Case13C	81	0	0.0166
	Case13D	82	0	0.0127
	Case13E	83	0	0.0089
	Case13F	84	0	0.0068
	Case13F	85	0	0.0050
	Case14	408	0	0.0282
	Case15	1054	0	0.0007
	Case16	575	0	0.0019
	Case17	246	0	0.0018
2	Case18	15	3	0.1230
	Case19	9	9	0.1425
	Case20A	7	17	0.2287
	Case20B	8	12	0.2228
	Case20C	9	8	0.1457
	Case20D	10	5	0.1395
	Case20E	11	5	0.0975
	Case20F	12	4	0.0915
	Case20G	13	4	0.0713
	Case21A	15	36	0.2843
	Case21B	18	3	0.1281
	Case22	19	3	0.0568

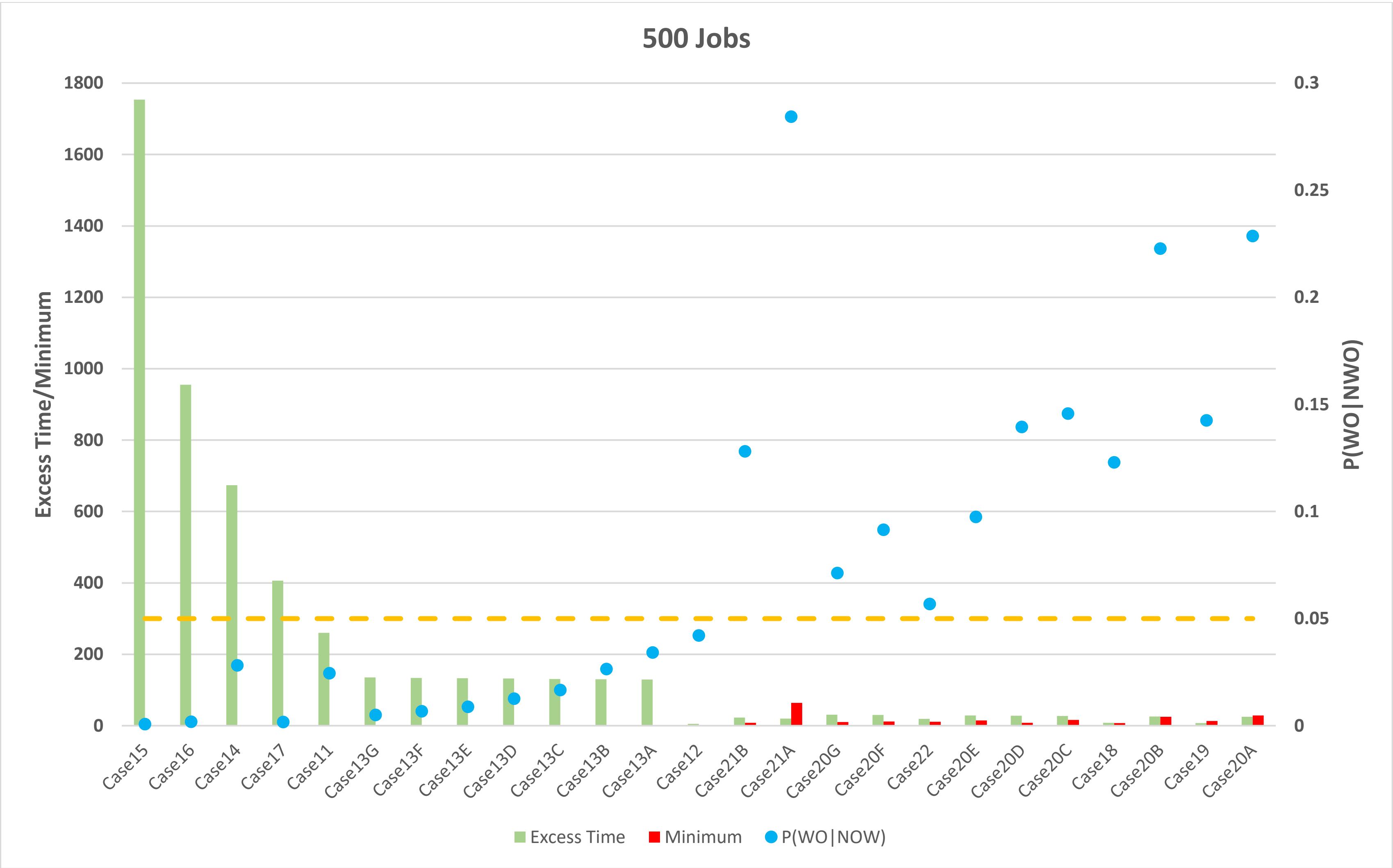


Figure 42: Excess Time, Minimum and P(WO|NWO) of Multiple Test Cases (500 Jobs)

5 CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this research, a new approach was established that approximates the probability distribution of the total number and the severity of work overload occurrences for a set of jobs that are launched down an assembly line workstation in a random order. To demonstrate the results, the model is applied to multiple test cases that are generated by varying processing times of job types, demand fraction of job types, the workstation length and the total number of jobs in the job set. By comparing the Markov chain model results with the simulation results of the assembly line workstation, it is observed that the model results are valid for the system being modeled. The model results are also compared with the minimum number of work overload occurrences obtained using a pairwise exchange search algorithm and it is observed that the accuracy of the model results is high when the number of jobs in the job set is large. This effect of the number of job types on accuracy is primarily because of the constant job type selection probability assumption in the Markov chain model. In general, the Markov chain model produces sufficiently accurate estimates of the number of work overload occurrences for a random launch sequence.

The results obtained using the model can be used to redesign the workstation dimensions and the cycle time for efficient space utilization. If the model predicts high work overload occurrences, then job sequencing optimization may reduce them, but perhaps not enough to meet requirements. The Markov chain model can also be used to predict whether applying sequencing optimization can give zero overloads and to demonstrate this, 25 test cases were generated. By examining the model results, an arbitrary cutoff value of the microstate Markov chain is established. Using this cutoff value, the test cases that can give zero work overloads after applying sequencing optimization can be identified.

5.2 Future Research Scope

The following are some recommendations that can be considered to extend this research:

1. Examine the validity of the arbitrary cutoff value of the microstate Markov chain transition probability that differentiates Category 1 and Category 2 test cases for other combinations of workstation dimension, job type demand fractions, and processing times.
2. Modify the Markov chain model for other strategies used to deal with the imminent work overload situation such as a side-by-side policy wherein the workstation operator and the utility worker work side-by-side on a job to avoid a work overload situation.
3. Investigate the use of the Markov chain model for job types that have uncertainty in processing times with times taken from distributions other than a uniform distribution.
4. Modify the Markov chain model such that it can be used to estimate other performance measures of interest such as conveyor stoppage time.
5. Develop a similar model that could estimate operational inefficiencies for a random job launching sequence for an assembly line that consists of multiple workstations.

REFERENCES

Bard, J. F., Dar-El, E., & Shtub, A. (1992). An Analytic framework for sequencing mixed model assembly lines. *International Journal of Production Research*, 30(1), 35-48.

Baybars, I. (1986). A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. *Management Science, 32*(8), 909-932.

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

Bowman, E. H. (1960). Assembly-Line Balancing by Linear Programming. *Operations Research,8*(3), 385-389.

Celano, G., Costa, A., Fichera, S., & Perrone, G. (2004). Human factor policy testing in the sequencing of manual mixed model assembly lines. *Computers & Operations Research,31*(1), 39-59.

Boysen, N., Kiel, M., & Scholl, A. (2011). Sequencing mixed-model assembly lines to minimise the number of work overload situations. International Journal of Production Research, 49(16), 4735-4760.

Dar-El, E. M., & Cother, R. F. (1975). Assembly line sequencing for model mix. *International Journal of Production Research,13*(5), 463-477.

Dar-El, E., & Cucuy, S. (1977). Optimal mixed-model sequencing for balanced assembly lines. *Omega,5*(3), 333-342.

Dar-El, E. M. (1978). Mixed-model assembly line sequencing problems. *Omega,6*(4), 313-323.

Durstenfeld, R. (1964). Algorithm 235: Random permutation. *Communications of the ACM, 7*(7), 420.

Jackson, J. R. (1956). A Computing Procedure for a Line Balancing Problem. *Management Science,2*(3), 261-271.

Kilbridge, M., Wester, L. (1963). The assembly line model-mix sequencing problem. *Proceeding of the Third International Conference on Operation Research, 247.*

Kim, D. S., & Smith, R. L. (1995). An exact aggregation/disaggregation algorithm for large scale markov chains. *Naval Research Logistics, 42*(7), 1115-1128.

Sarker, B. R., & Pan, H. (1998). Designing a mixed-model assembly line to minimize the costs of idle and utility times. *Computers & Industrial Engineering,34*(3), 609-628.

Sarker, B., & Pan, H. (2001). Designing a mixed-model, open-station assembly line using mixed-integer programming. *Journal of the Operational Research Society, 52*(5), 545-558.

Scholl, A. (1999). *Balancing and sequencing of assembly lines*. Heidelberg: Physica-Verlag.

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

Tsai, L. (1995). Mixed-Model Sequencing to Minimize Utility Work and the Risk of Conveyor Stoppage. *Management Science, 41*(3), 485-495.

Franz, C., Hällgren, E. C., & Koberstein, A. (2014). Resequencing orders on mixed-model assembly lines: Heuristic approaches to minimise the number of overload situations. *International Journal of Production Research*, *52*(19), 5823-5840.

Merengo, C.,Nava, F., & Pozzetti, A. (2010). Balancing and sequencing manual mixed-model assembly lines. *International Journal of Production Research*, 37(12), 2835-2860.

Monden, Y. (1993). Toyota Production System, Second edition, *Institute of Industrial Engineers*, pp. 224-231.

Okamura, K., & Yamashina, H. (1979). A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor. *International Journal of Production Research*, *17*(3), 233-247.

Wester, L., Kilbridge, M. (1964). The assembly line model-mix sequencing problem. *Third international conference on operation research*, Oslo 1963, 247-260.

Wild, R. (1972). *Mass-production management: the design and operation of production flow-line systems*. London: Wiley.

Xiaobo, Z., & Ohno, K. (1994). A sequencing problem for a mixed-model assembly line in a JIT production system. *Computers & Industrial Engineering*,*27*(1-4), 71-74.

Xiaobo, Z., & Ohno, K. (1997). Algorithms for sequencing mixed models on an assembly line in a JIT production system. *Computers & Industrial Engineering*,*32*(1), 47-56.

Xiaobo, Z., & Ohno, K. (2000). Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. *European Journal of Operational Research*,*124*(3), 560-570.

Yano, C.A., & Bolat, A. (1989). Survey, development, and application of algorithms for sequencing paced assembly lines. *Journal of Manufacturing and Operations Management*, 2, 172-198.

Yano, C. A., & Rachamadugu, R. (1991). Sequencing to Minimize Work Overload in Assembly Lines with Product Options. *Management Science,37*(5), 572-586.

APPENDICES

Appendix 1: Verification of (10)

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
0	0.0000	0.0000
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000
5	0.0000	0.0000
6	0.0000	0.0000
7	0.0000	0.0000
8	0.0000	0.0000
9	0.0000	0.0000
10	0.0000	0.0000
11	0.0000	0.0000
12	0.0000	0.0000
13	0.0000	0.0000
14	0.0000	0.0000
15	0.0000	0.0000
16	0.0000	0.0000
17	0.0000	0.0000
18	0.0000	0.0000
19	0.0000	0.0000
20	0.0000	0.0000
21	0.0000	0.0000
22	0.0000	0.0000
23	0.0000	0.0000
24	0.0000	0.0000
25	0.0000	0.0000
26	0.0000	0.0000
27	0.0000	0.0000
28	0.0000	0.0000
29	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
30	0.0000	0.0000
31	0.0000	0.0000
32	0.0000	0.0000
33	0.0000	0.0000
34	0.0000	0.0000
35	0.0000	0.0000
36	0.0000	0.0000
37	0.0000	0.0000
38	0.0000	0.0000
39	0.0000	0.0000
40	0.0000	0.0000
41	0.0000	0.0000
42	0.0000	0.0000
43	0.0000	0.0000
44	0.0000	0.0000
45	0.0000	0.0000
46	0.0000	0.0000
47	0.0000	0.0000
48	0.0000	0.0000
49	0.0000	0.0000
50	0.0000	0.0000
51	0.0000	0.0000
52	0.0000	0.0000
53	0.0000	0.0000
54	0.0000	0.0000
55	0.0000	0.0000
56	0.0000	0.0000
57	0.0000	0.0000
58	0.0000	0.0000
59	0.0000	0.0000
60	0.0000	0.0000
61	0.0000	0.0000
62	0.0000	0.0000
63	0.0000	0.0000
64	0.0000	0.0000
65	0.0000	0.0000
66	0.0000	0.0000
67	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
68	0.0000	0.0000
69	0.0000	0.0000
70	0.0000	0.0000
71	0.0000	0.0000
72	0.0000	0.0000
73	0.0000	0.0000
74	0.0000	0.0000
75	0.0000	0.0000
76	0.0000	0.0000
77	0.0000	0.0000
78	0.0000	0.0000
79	0.0000	0.0000
80	0.0000	0.0000
81	0.0000	0.0000
82	0.0000	0.0000
83	0.0000	0.0000
84	0.0000	0.0000
85	0.0000	0.0000
86	0.0000	0.0000
87	0.0000	0.0000
88	0.0000	0.0000
89	0.0000	0.0000
90	0.0000	0.0000
91	0.0000	0.0000
92	0.0000	0.0000
93	0.0000	0.0000
94	0.0000	0.0000
95	0.0000	0.0000
96	0.0000	0.0000
97	0.0000	0.0000
98	0.0000	0.0000
99	0.0000	0.0000
100	0.0000	0.0000
101	0.0000	0.0000
102	0.0000	0.0000
103	0.0000	0.0000
104	0.0000	0.0000
105	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
106	0.0001	0.0001
107	0.0001	0.0001
108	0.0001	0.0001
109	0.0002	0.0002
110	0.0003	0.0002
111	0.0004	0.0004
112	0.0005	0.0005
113	0.0007	0.0007
114	0.0010	0.0009
115	0.0013	0.0013
116	0.0017	0.0017
117	0.0023	0.0022
118	0.0029	0.0030
119	0.0038	0.0038
120	0.0048	0.0049
121	0.0060	0.0059
122	0.0074	0.0073
123	0.0090	0.0091
124	0.0109	0.0109
125	0.0129	0.0130
126	0.0152	0.0151
127	0.0177	0.0176
128	0.0204	0.0203
129	0.0231	0.0233
130	0.0259	0.0262
131	0.0287	0.0288
132	0.0315	0.0315
133	0.0341	0.0340
134	0.0364	0.0367
135	0.0385	0.0384
136	0.0403	0.0402
137	0.0416	0.0419
138	0.0424	0.0429
139	0.0428	0.0426
140	0.0427	0.0431
141	0.0421	0.0418
142	0.0411	0.0408
143	0.0396	0.0397

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
144	0.0377	0.0377
145	0.0356	0.0356
146	0.0331	0.0331
147	0.0305	0.0303
148	0.0278	0.0276
149	0.0250	0.0248
150	0.0223	0.0224
151	0.0196	0.0196
152	0.0171	0.0171
153	0.0147	0.0147
154	0.0125	0.0126
155	0.0106	0.0105
156	0.0088	0.0088
157	0.0073	0.0071
158	0.0059	0.0060
159	0.0048	0.0048
160	0.0038	0.0038
161	0.0030	0.0030
162	0.0023	0.0023
163	0.0018	0.0017
164	0.0014	0.0013
165	0.0010	0.0010
166	0.0008	0.0008
167	0.0006	0.0006
168	0.0004	0.0004
169	0.0003	0.0003
170	0.0002	0.0002
171	0.0002	0.0001
172	0.0001	0.0001
173	0.0001	0.0001
174	0.0001	0.0001
175	0.0000	0.0000
176	0.0000	0.0000
177	0.0000	0.0000
178	0.0000	0.0000
179	0.0000	0.0000
180	0.0000	0.0000
181	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
182	0.0000	0.0000
183	0.0000	0.0000
184	0.0000	0.0000
185	0.0000	0.0000
186	0.0000	0.0000
187	0.0000	0.0000
188	0.0000	0.0000
189	0.0000	0.0000
190	0.0000	0.0000
191	0.0000	0.0000
192	0.0000	0.0000
193	0.0000	0.0000
194	0.0000	0.0000
195	0.0000	0.0000
196	0.0000	0.0000
197	0.0000	0.0000
198	0.0000	0.0000
199	0.0000	0.0000
200	0.0000	0.0000
201	0.0000	0.0000
202	0.0000	0.0000
203	0.0000	0.0000
204	0.0000	0.0000
205	0.0000	0.0000
206	0.0000	0.0000
207	0.0000	0.0000
208	0.0000	0.0000
209	0.0000	0.0000
210	0.0000	0.0000
211	0.0000	0.0000
212	0.0000	0.0000
213	0.0000	0.0000
214	0.0000	0.0000
215	0.0000	0.0000
216	0.0000	0.0000
217	0.0000	0.0000
218	0.0000	0.0000
219	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
220	0.0000	0.0000
221	0.0000	0.0000
222	0.0000	0.0000
223	0.0000	0.0000
224	0.0000	0.0000
225	0.0000	0.0000
226	0.0000	0.0000
227	0.0000	0.0000
228	0.0000	0.0000
229	0.0000	0.0000
230	0.0000	0.0000
231	0.0000	0.0000
232	0.0000	0.0000
233	0.0000	0.0000
234	0.0000	0.0000
235	0.0000	0.0000
236	0.0000	0.0000
237	0.0000	0.0000
238	0.0000	0.0000
239	0.0000	0.0000
240	0.0000	0.0000
241	0.0000	0.0000
242	0.0000	0.0000
243	0.0000	0.0000
244	0.0000	0.0000
245	0.0000	0.0000
246	0.0000	0.0000
247	0.0000	0.0000
248	0.0000	0.0000
249	0.0000	0.0000
250	0.0000	0.0000
251	0.0000	0.0000
252	0.0000	0.0000
253	0.0000	0.0000
254	0.0000	0.0000
255	0.0000	0.0000
256	0.0000	0.0000
257	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
258	0.0000	0.0000
259	0.0000	0.0000
260	0.0000	0.0000
261	0.0000	0.0000
262	0.0000	0.0000
263	0.0000	0.0000
264	0.0000	0.0000
265	0.0000	0.0000
266	0.0000	0.0000
267	0.0000	0.0000
268	0.0000	0.0000
269	0.0000	0.0000
270	0.0000	0.0000
271	0.0000	0.0000
272	0.0000	0.0000
273	0.0000	0.0000
274	0.0000	0.0000
275	0.0000	0.0000
276	0.0000	0.0000
277	0.0000	0.0000
278	0.0000	0.0000
279	0.0000	0.0000
280	0.0000	0.0000
281	0.0000	0.0000
282	0.0000	0.0000
283	0.0000	0.0000
284	0.0000	0.0000
285	0.0000	0.0000
286	0.0000	0.0000
287	0.0000	0.0000
288	0.0000	0.0000
289	0.0000	0.0000
290	0.0000	0.0000
291	0.0000	0.0000
292	0.0000	0.0000
293	0.0000	0.0000
294	0.0000	0.0000
295	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
296	0.0000	0.0000
297	0.0000	0.0000
298	0.0000	0.0000
299	0.0000	0.0000
300	0.0000	0.0000
301	0.0000	0.0000
302	0.0000	0.0000
303	0.0000	0.0000
304	0.0000	0.0000
305	0.0000	0.0000
306	0.0000	0.0000
307	0.0000	0.0000
308	0.0000	0.0000
309	0.0000	0.0000
310	0.0000	0.0000
311	0.0000	0.0000
312	0.0000	0.0000
313	0.0000	0.0000
314	0.0000	0.0000
315	0.0000	0.0000
316	0.0000	0.0000
317	0.0000	0.0000
318	0.0000	0.0000
319	0.0000	0.0000
320	0.0000	0.0000
321	0.0000	0.0000
322	0.0000	0.0000
323	0.0000	0.0000
324	0.0000	0.0000
325	0.0000	0.0000
326	0.0000	0.0000
327	0.0000	0.0000
328	0.0000	0.0000
329	0.0000	0.0000
330	0.0000	0.0000
331	0.0000	0.0000
332	0.0000	0.0000
333	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
334	0.0000	0.0000
335	0.0000	0.0000
336	0.0000	0.0000
337	0.0000	0.0000
338	0.0000	0.0000
339	0.0000	0.0000
340	0.0000	0.0000
341	0.0000	0.0000
342	0.0000	0.0000
343	0.0000	0.0000
344	0.0000	0.0000
345	0.0000	0.0000
346	0.0000	0.0000
347	0.0000	0.0000
348	0.0000	0.0000
349	0.0000	0.0000
350	0.0000	0.0000
351	0.0000	0.0000
352	0.0000	0.0000
353	0.0000	0.0000
354	0.0000	0.0000
355	0.0000	0.0000
356	0.0000	0.0000
357	0.0000	0.0000
358	0.0000	0.0000
359	0.0000	0.0000
360	0.0000	0.0000
361	0.0000	0.0000
362	0.0000	0.0000
363	0.0000	0.0000
364	0.0000	0.0000
365	0.0000	0.0000
366	0.0000	0.0000
367	0.0000	0.0000
368	0.0000	0.0000
369	0.0000	0.0000
370	0.0000	0.0000
371	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
372	0.0000	0.0000
373	0.0000	0.0000
374	0.0000	0.0000
375	0.0000	0.0000
376	0.0000	0.0000
377	0.0000	0.0000
378	0.0000	0.0000
379	0.0000	0.0000
380	0.0000	0.0000
381	0.0000	0.0000
382	0.0000	0.0000
383	0.0000	0.0000
384	0.0000	0.0000
385	0.0000	0.0000
386	0.0000	0.0000
387	0.0000	0.0000
388	0.0000	0.0000
389	0.0000	0.0000
390	0.0000	0.0000
391	0.0000	0.0000
392	0.0000	0.0000
393	0.0000	0.0000
394	0.0000	0.0000
395	0.0000	0.0000
396	0.0000	0.0000
397	0.0000	0.0000
398	0.0000	0.0000
399	0.0000	0.0000
400	0.0000	0.0000
401	0.0000	0.0000
402	0.0000	0.0000
403	0.0000	0.0000
404	0.0000	0.0000
405	0.0000	0.0000
406	0.0000	0.0000
407	0.0000	0.0000
408	0.0000	0.0000
409	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
410	0.0000	0.0000
411	0.0000	0.0000
412	0.0000	0.0000
413	0.0000	0.0000
414	0.0000	0.0000
415	0.0000	0.0000
416	0.0000	0.0000
417	0.0000	0.0000
418	0.0000	0.0000
419	0.0000	0.0000
420	0.0000	0.0000
421	0.0000	0.0000
422	0.0000	0.0000
423	0.0000	0.0000
424	0.0000	0.0000
425	0.0000	0.0000
426	0.0000	0.0000
427	0.0000	0.0000
428	0.0000	0.0000
429	0.0000	0.0000
430	0.0000	0.0000
431	0.0000	0.0000
432	0.0000	0.0000
433	0.0000	0.0000
434	0.0000	0.0000
435	0.0000	0.0000
436	0.0000	0.0000
437	0.0000	0.0000
438	0.0000	0.0000
439	0.0000	0.0000
440	0.0000	0.0000
441	0.0000	0.0000
442	0.0000	0.0000
443	0.0000	0.0000
444	0.0000	0.0000
445	0.0000	0.0000
446	0.0000	0.0000
447	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
448	0.0000	0.0000
449	0.0000	0.0000
450	0.0000	0.0000
451	0.0000	0.0000
452	0.0000	0.0000
453	0.0000	0.0000
454	0.0000	0.0000
455	0.0000	0.0000
456	0.0000	0.0000
457	0.0000	0.0000
458	0.0000	0.0000
459	0.0000	0.0000
460	0.0000	0.0000
461	0.0000	0.0000
462	0.0000	0.0000
463	0.0000	0.0000
464	0.0000	0.0000
465	0.0000	0.0000
466	0.0000	0.0000
467	0.0000	0.0000
468	0.0000	0.0000
469	0.0000	0.0000
470	0.0000	0.0000
471	0.0000	0.0000
472	0.0000	0.0000
473	0.0000	0.0000
474	0.0000	0.0000
475	0.0000	0.0000
476	0.0000	0.0000
477	0.0000	0.0000
478	0.0000	0.0000
479	0.0000	0.0000
480	0.0000	0.0000
481	0.0000	0.0000
482	0.0000	0.0000
483	0.0000	0.0000
484	0.0000	0.0000
485	0.0000	0.0000

Number of Work Overload Occurrences	Probability using (10)	Probability using Monte Carlo Simulation
486	0.0000	0.0000
487	0.0000	0.0000
488	0.0000	0.0000
489	0.0000	0.0000
490	0.0000	0.0000
491	0.0000	0.0000
492	0.0000	0.0000
493	0.0000	0.0000
494	0.0000	0.0000
495	0.0000	0.0000
496	0.0000	0.0000
497	0.0000	0.0000
498	0.0000	0.0000
499	0.0000	0.0000
500	0.0000	0.0000

Appendix 2: Multiple Job Set and MMAL Compositions

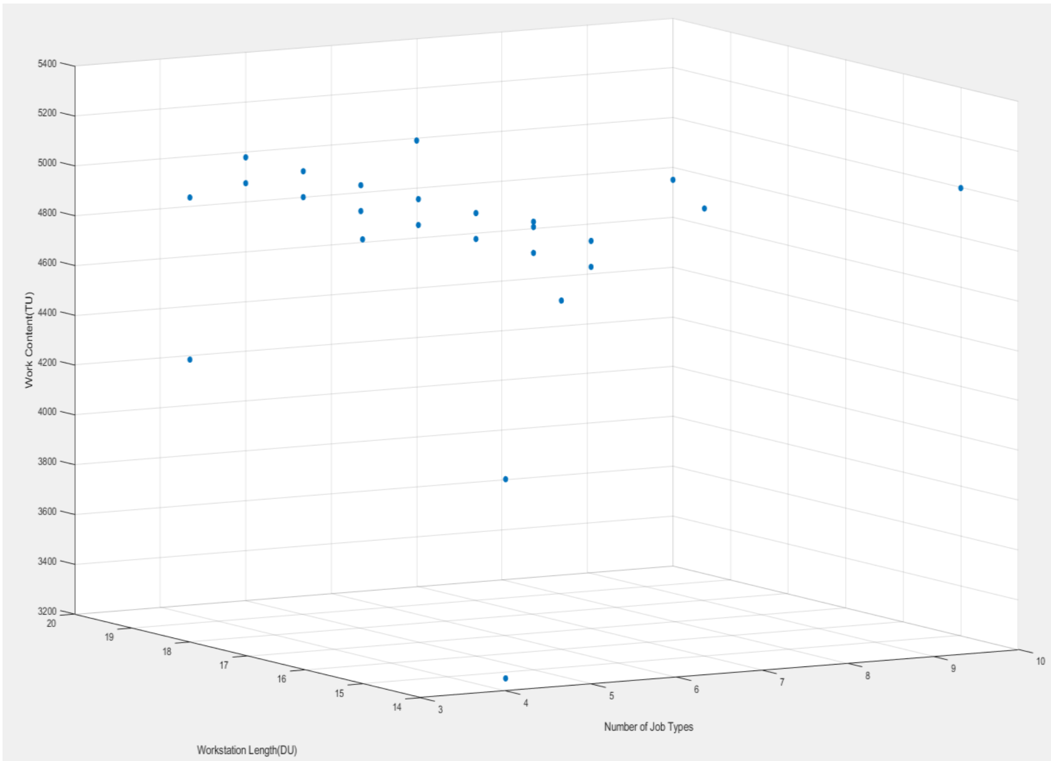
Category	Case	Processing Time (Time Units)	Demand Fraction	Workstation Length (Distance Units)
1 (Excess Time>=0, Minimum=0)	Case11	5	0.10	20
		6	0.10	
		7	0.10	
		8	0.10	
		9	0.10	
		10	0.10	
		11	0.10	
		12	0.10	
		13	0.10	
		14	0.10	
	Case16	4	0.25	14
		5	0.25	
		6	0.25	
		12	0.25	
	Case12	8	0.20	15
		9	0.20	

Category	Case	Processing Time (Time Units)	Demand Fraction	Workstation Length (Distance Units)
		10	0.20	
		11	0.20	
		12	0.20	
	Case13A	8	0.25	14
		9	0.25	
		10	0.25	
		12	0.25	
	Case13B	8	0.25	15
		9	0.25	
		10	0.25	
		12	0.25	
	Case13C	8	0.25	16
		9	0.25	
		10	0.25	
		12	0.25	
	Case13D	8	0.25	17
		9	0.25	
		10	0.25	
		12	0.25	
	Case13E	8	0.25	18
		9	0.25	
		10	0.25	
		12	0.25	
	Case13F	8	0.25	19
		9	0.25	
		10	0.25	
		12	0.25	
	Case13G	8	0.25	20
		9	0.25	
		10	0.25	
		12	0.25	
	Case14	6	0.33	18
		7	0.33	
		13	0.33	
	Case15	4	0.25	14
		5	0.25	
		6	0.25	

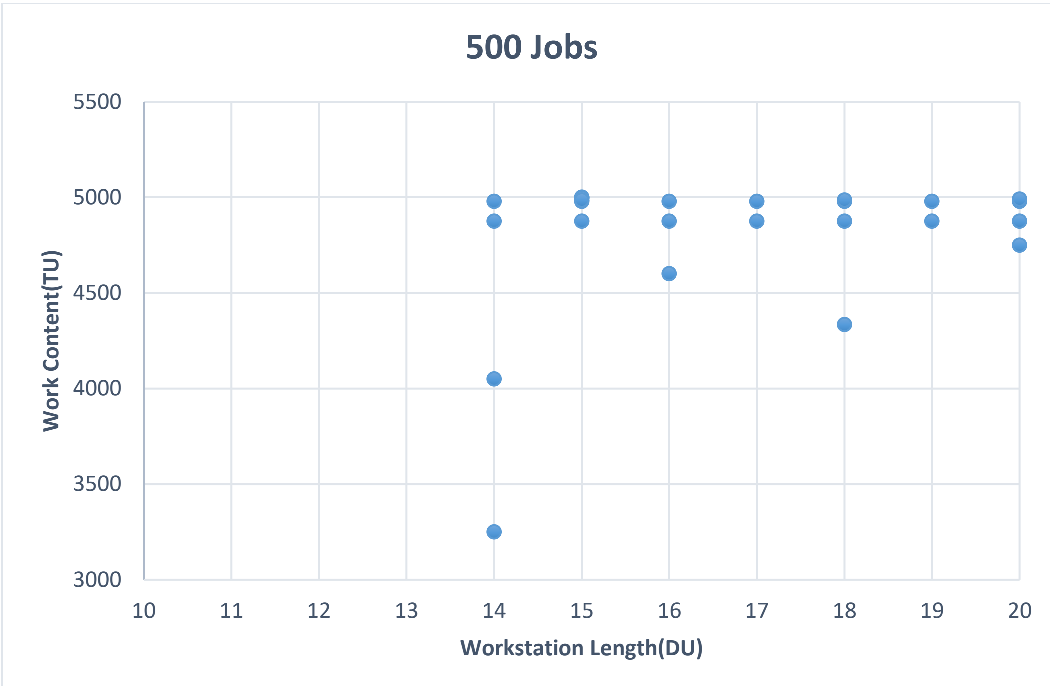
Category	Case	Processing Time (Time Units)	Demand Fraction	Workstation Length (Distance Units)
		11	0.25	
	Case17	7	0.10	16
		8	0.30	
		9	0.20	
		10	0.20	
		11	0.10	
		12	0.10	
2 (Excess Time>=0, Minimum>0)	Case18	5	0.04	15
		6	0.13	
		7	0.07	
		8	0.09	
		9	0.09	
		10	0.12	
		11	0.10	
		12	0.12	
		13	0.10	
		14	0.14	
	Case19	5	0.04	15
		6	0.16	
		7	0.09	
		8	0.11	
		9	0.09	
		10	0.14	
		14	0.36	
	Case20A	5	0.17	14
		6	0.17	
		7	0.08	
		12	0.30	
		14	0.29	
	Case20B	5	0.17	15
		6	0.17	
		7	0.08	
		12	0.30	
		14	0.29	
	Case20C	5	0.17	16
		6	0.17	
		7	0.08	

Category	Case	Processing Time (Time Units)	Demand Fraction	Workstation Length (Distance Units)
		12	0.30	
		14	0.29	
	Case20D	5	0.17	17
		6	0.17	
		7	0.08	
		12	0.30	
		14	0.29	
	Case20E	5	0.17	18
		6	0.17	
		7	0.08	
		12	0.30	
		14	0.29	
	Case20F	5	0.17	19
		6	0.17	
		7	0.08	
		12	0.30	
		14	0.29	
	Case20G	5	0.17	20
		6	0.17	
		7	0.08	
		12	0.30	
		14	0.29	
	Case21A	4	0.25	15
		8	0.16	
		13	0.59	
	Case21B	4	0.25	18
		8	0.16	
		13	0.59	
	Case22	4	0.21	20
		10	0.21	
		11	0.21	
		12	0.21	
		13	0.13	
		14	0.03	
		15	0.03	

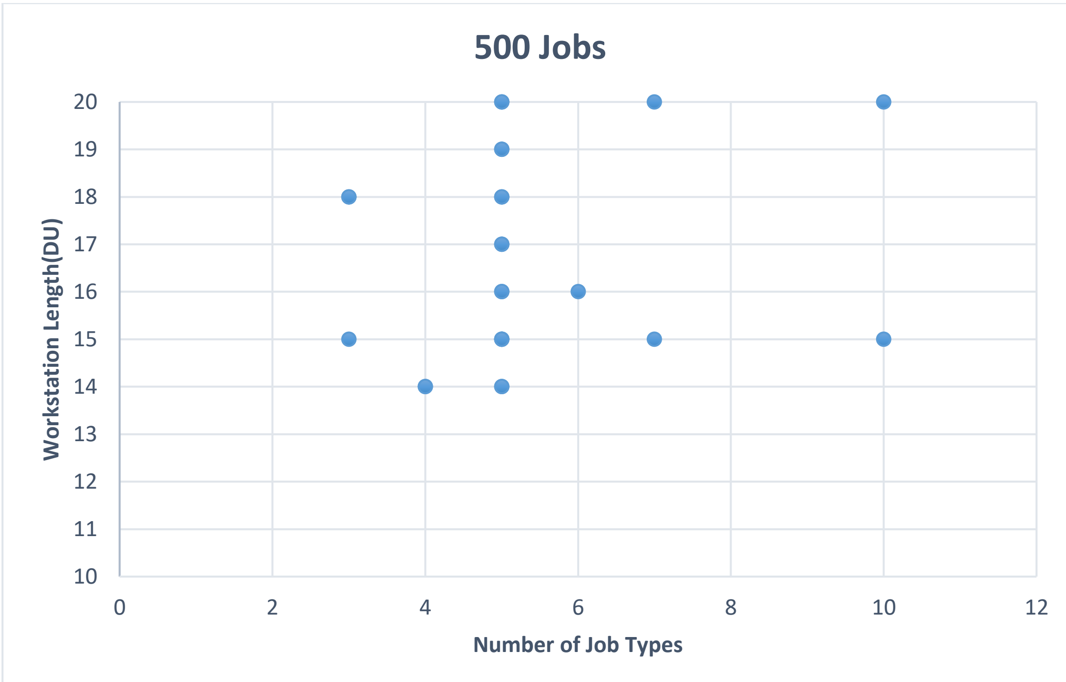
Appendix 3: Three Dimensional Visualization of Test Cases (500 Jobs)



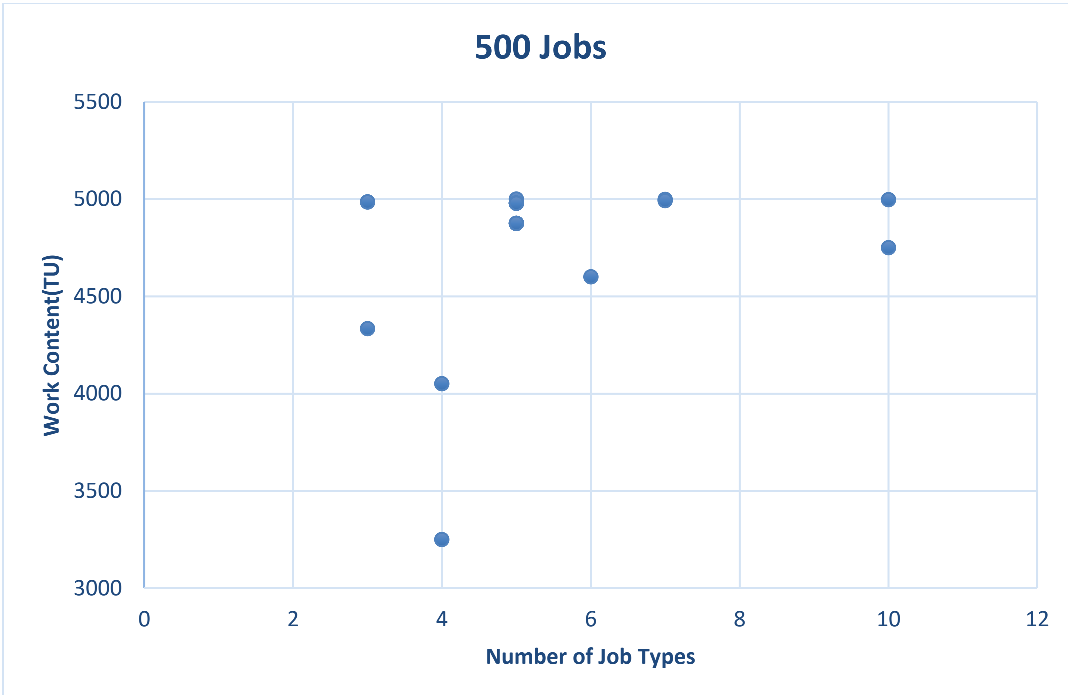
Appendix 4: Orthographic Projection - I (500 Jobs)



Appendix 5: Orthographic Projection - II (500 Jobs)



Appendix 6: Orthographic Projection - III (500 Jobs)



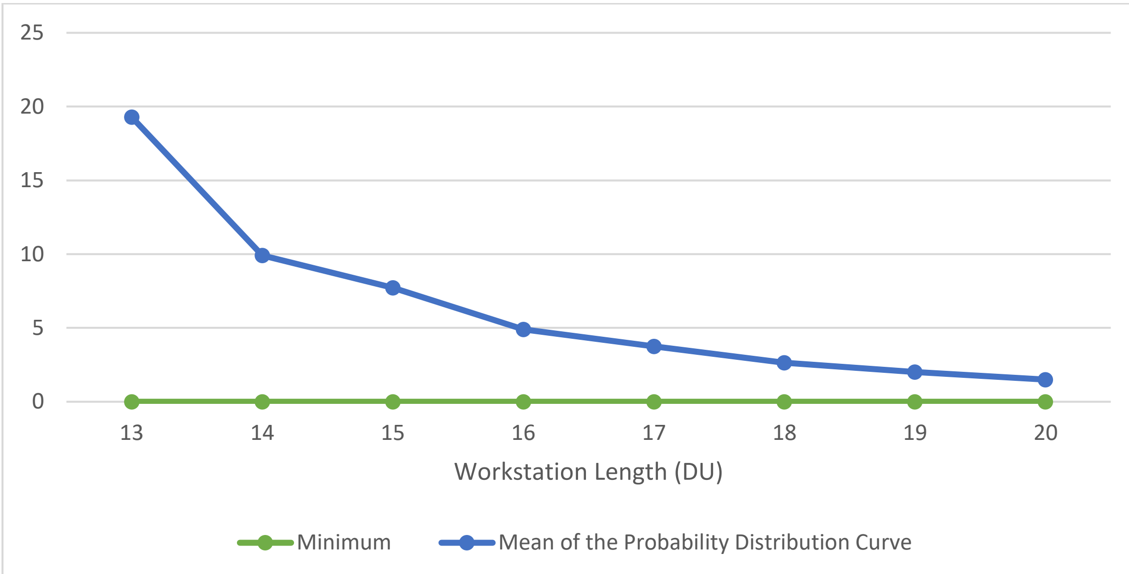
The shape parameters of the probability distribution curve are investigated to check if they can be used to differentiate between various categories of jobs. A test case belonging to each category was randomly selected and the effect of the workstation length on the shape parameters was studied. The results are presented in

Appendix **7** to Appendix 12. It is observed that the standard deviation of the probability distribution of the number of work overload occurrences of Category 1 case is lower than the Category 2 case. The mean of the probability distribution curve cannot be used for differentiating cases.

Appendix 7: Case 13: Effect of Workstation Length on the Mean and Standard Deviation of the Probability Distribution of Work Overload Occurrences

Category	Case	Number of Jobs	Workstation Length (Time Units)	Minimum	Mean of the Probability Distribution Curve	Standard Deviation of the Probability Distribution Curve
1	Case13	300	13	0	19.29	3.97
			14	0	9.91	2.99
			15	0	7.72	2.67
			16	0	4.90	2.16
			17	0	3.75	1.90
			18	0	2.64	1.60
			19	0	2.02	1.41
			20	0	1.50	1.22

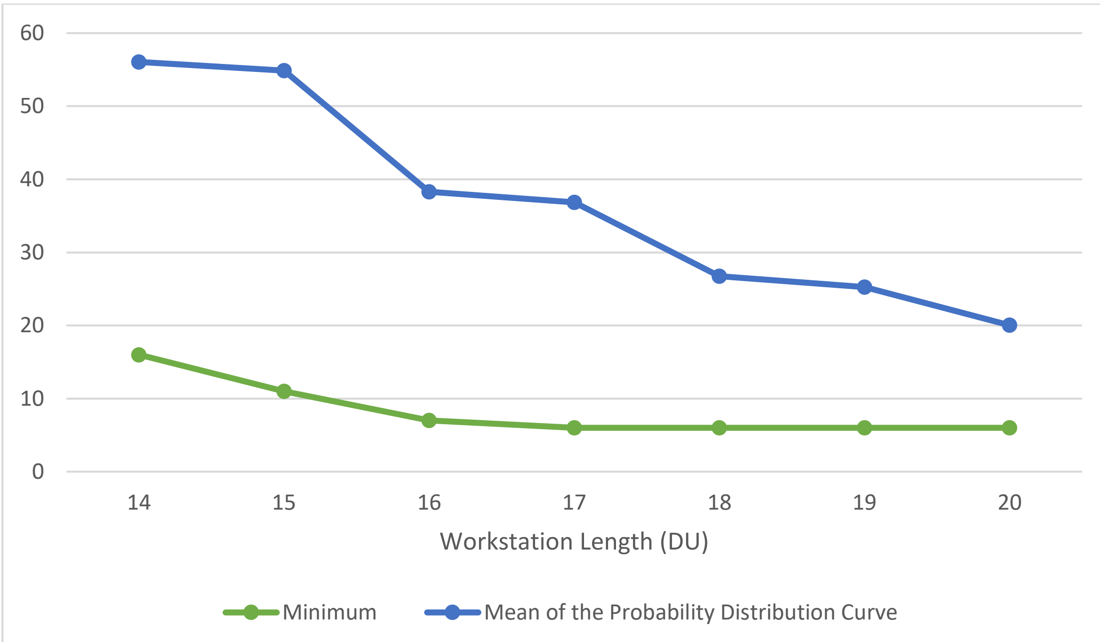
Appendix 8: Case 13: Effect of Workstation Length on the Mean of the Probability Distribution of Work Overload Occurrences



Appendix 9: Case 20: Effect of Workstation Length on the Mean and Standard Deviation of the Probability Distribution of Work Overload Occurrences

Category	Case	Number of Jobs	Workstation Length (Time Units)	Minimum	Mean of the Probability Distribution Curve	Standard Deviation of the Probability Distribution Curve
2	Case20	300	14	16	56.06	5.36
			15	11	54.87	5.34
			16	7	38.30	4.99
			17	6	36.87	4.94
			18	6	26.75	4.48
			19	6	25.25	4.39
			20	6	20.03	4.03

Appendix 10: Case 20: Effect of Workstation Length on the Mean of the Probability Distribution of Work Overload Occurrences



Appendix 11: Case 10: Effect of Workstation Length on the Mean and Standard Deviation of the Probability Distribution of Work Overload Occurrences

Category	Case	Number of Jobs	Workstation Length (Time Units)	Minimum	Mean of the Probability Distribution Curve	Standard Deviation of the Probability Distribution Curve
3	Case10	300	13	38	60.04	5.37
			14	26	35.31	4.88
			15	26	35.31	4.88
			16	19	24.50	4.34
			17	19	24.50	4.34
			18	15	18.61	3.91
			19	15	18.61	3.91
			20	12	14.96	3.58

Appendix 12: Case 10: Effect of Workstation Length on the Mean of the Probability Distribution of Work Overload Occurrences