12-2013

A Novel Work-Sharing Protocol for U-Shaped Assembly Lines

Srinath Sriram

ROCHESTER INSTITUTE OF TECHNOLOGY

A Novel Work-Sharing Protocol for U-Shaped Assembly Lines

A Thesis submitted in partial fulfillment of the requirement for the degree of Master of Science in Industrial Engineering in the Kate Gleason College of Engineering

Industrial and Systems Engineering Department

SRINATH SRIRAM (RIT Student)

December 2013

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Srinath Sriram has been examined and approved by the committee as satisfactory for the thesis requirement for the Master of Science degree

Approved by:

____

Dr. Andres L. Carrano, Thesis Advisor,

Associate Professor, Department of Industrial & Systems Engineering, RIT

____

Dr. Michael E. Kuhl, Committee Member,

Professor, Department of Industrial & Systems Engineering, RIT

____

Dr. Brian Thorn, Committee Member,

Associate Professor, Department of Industrial & Systems Engineering, RIT

Acknowledgements

This thesis work was by no means the work of an individual. I would like to thank everyone involved - directly, or indirectly.

Dr. Andres Carrano, without whose support and guidance, this work would not have seen the light of day. His adherence to the scientific method in all aspects - including critique and advice - made me realize what a master's thesis entails. Dr. Kuhl and Dr. Thorn, both experts in their own fields, for taking the time out of their busy schedules to answer my questions. Dr. Kuhl's simulation class was one of the inspirational forces behind my choice of methodology for this thesis. My parents and brother, without whose efforts, well wishes, nourishment, and love, I wouldn't be here today. Prof. John Kaemmerlen, whose inspiration, guidance, and teachings gave legs to my career. My friends, roommates, and colleagues who were there with me and for me throughout this journey.

All the Toyota lab assistants and volunteers who participated in the experiments for their help and support

Last, but not the least, Manuel Tolentino, and the anonymous Mr.E, for their help in simulation, when I needed it the most.

Abstract

Companies worldwide try to employ contemporary manufacturing systems that can cope with changes in external competitive environments and internal process variability. Just In Time (JIT) philosophy helps achieve the required resilience by its policy of having people, machines, and material just-in-time for any given process. U-shaped assembly lines (U-lines) are used to implement JIT principles. Another principle that helps achieve competitive advantage by developing a flexible workforce that responds efficiently to change is that of work-sharing. Operators share work and help each other in a dynamic and floating way, requiring little management effort to distribute workload amongst operators, or balance the assembly line.

The aim of this thesis work is to develop an effective work-sharing protocol for U-shaped assembly lines that will provide the combined advantages of U-lines and work-sharing principles. The new protocol is based on two ideas from literature - the Cellular Bucket Brigade (CBB) system, and the Modified Work-Sharing (MWS) system. To keep the focus on developing the protocol, the scope of this work was limited to two worker systems. The methodology used is to model the protocol and U-line system as a discrete event simulation model, and then use an optimization model to maximize throughput and find optimal buffer locations and levels. A physical simulation experiment was conducted in the Toyota Production Systems lab at RIT to validate the model. Once validated, computer simulation experiments were run with industry data, and results obtained were compared with existing protocols from literature.

It was found that the new protocol performed at least as well as the CBB protocol, improving the output by an average of 1%, for the scenarios tested. Increase in processing speed variability as well as larger variation among workers were found to negatively impact the performance of the protocol. The results were analyzed further to understand why these factors are significant, and

why there are anomalies and patterns, or lack thereof. Finally, limitations of the protocol, and opportunities for future research in the field are presented. Major limitations of the protocol are that it is difficult to comprehend, and the assumption of an assembly line divided into equal tasks is not practical in the industry.

Contents

List of Figures

List of Tables

1. Introduction

In today's competitive world, manufacturing companies are constantly trying to increase their productivity with the same amount of resources. They also find the need to make their systems flexible to counter external demand variability, while dealing with internal process variability. Reducing idle time of a limiting resource is the key to increased productivity (McClain, Schultz & Thomas, 2000). When labor is the limiting resource in a production facility, work-sharing is one way to reduce idle time (McClain et al., 2000). U-shaped assembly lines (U-lines) are another option that companies turn to when they try to increase their productivity without increasing their resources. After reviewing existing literature, a research gap was identified in the area of developing a work-sharing methodology or protocol for U-shaped assembly lines. This work proposes a new protocol for two worker U-lines in which buffers (temporary storage locations within the assembly line) are employed in an existing protocol.

One of the ways that companies try to reduce idle time is by following the Just In Time (JIT) philosophy. The JIT philosophy is a policy of having men, machine and material arrive and leave just-in-time for any given process. JIT philosophy can be considered a subset of the 'Lean Principles', also known as the Toyota Production System (Hopp & Spearman, 2008). U-lines are used to implement these JIT principles (Hopp & Spearman, 2008). Miltenburg (2001) claims that the productivity improved in companies by an average of 76%, WIP dropped by 86%, lead time shrunk by 75% and defective rates dropped by 83% when U-lines replaced linear assembly lines without increasing the workforce. Another principle that helps companies achieve competitive advantage (by developing a flexible workforce that can respond to changes quickly) is that of work-sharing. As it focuses on utilizing workers more efficiently and nurturing team work, assembly lines that employ work-sharing require little effort from the management to distribute the

work amongst workers (Ghiram, 2012). In these systems, workers share work and help each other in a dynamic and floating way.

The bucket brigade system is based on this idea of sharing work. This system lists a set of rules that these workers should follow while working on the assembly line, proposed by Bartholdi & Eisenstein (1996). The modified work-sharing (MWS) system (introduced in Montano et.al. 2007) builds on the Bucket Brigade system and explores the option of having inventories between workstations, so that the probability of a worker being blocked by his/her downstream neighbor is reduced. Cellular Bucket Brigades (CBB), proposed by Lim (2011) seeks to reduce the unproductive walking time of workers in a bucket brigade by considering a linear assembly line bent in half, so that workers can exchange work across the aisle based on a set of rules. This system, in effect, is a U-shaped assembly line using a work-sharing protocol. But, this system too has its flaws: it requires the aisle width to be less than 4% the entire length of the line (which may not be practical) and assumes a continuous assembly line. In addition to these flaws, there is a gap in literature in the area of a self-balancing protocol for U-lines that accommodates discrete tasks and stochastic processing times. This thesis aims to address this gap by developing a new work-sharing protocol for U-shaped assembly lines by creating a simulation model in ARENA (a discrete event simulation software) and then comparing the performance of this new protocol with existing protocols. The new protocol will utilize the cellular bucket brigade system and also capture the effectiveness of using buffers (as in the MWS system). Section 2 provides a background of the topics, section 3 is a detailed review of existing literature, section 4 outlines the problem statement, section 5 describes the methodology used to attack this problem, section 6 explains the results and conclusions that were drawn, and section 7 speaks to the scope of future work in this area.

2. Background

The sections in this chapter introduce the background concepts such as the Just In Time (JIT) production system, traditional assembly lines, U-shaped assembly lines, contemporary solutions, and relevant work-sharing protocols and principles.

2.1 Just In Time (JIT) Production System

The JIT production system is a set of principles, rules and ideas which are essentially a subset of the principles that form the lean manufacturing system. The main idea being that people, machines and materials should ‘arrive’ and ‘leave’ just-in-time for any given process. This tends to eliminate waste – in terms of time, material and resources. Companies worldwide try to achieve this by employing a flexible manufacturing system (and hence a flexible assembly line) that can cope with changes in the external competitive environments. Research suggests that there are several reasons why JIT manufacturing systems are able to cope with these external changes, including cross-trained multi-functional workers and efficient facility layouts (Miltenburg, 2001). Due to its wide range of advantages, JIT has been increasingly gaining interest by many manufacturing companies. Some benefits of JIT production systems (Hopp, 2008) are:

1. WIP reduction

2. Increased Quality

3. Increased productivity

4. Reduced space requirements

5. Increased flexibility

6. Lower overheads

7. Increased employee motivation.

2.2 Traditional Assembly Lines

An assembly line typically consists of a sequence of workstations through which the product is processed from Raw Materials (RM) to Finished Good (FG). The product remains for a certain time at each station, during which a group of tasks are performed; and workers are assigned to different stations. Traditionally, as designed by Henry Ford in 1915, assembly lines are arranged in a straight line (Miltenburg, 2001), i.e. raw materials arrive at one side of the line and finished goods are produced at other end. In this case, each worker may be assigned to one workstation or multiple workstations.

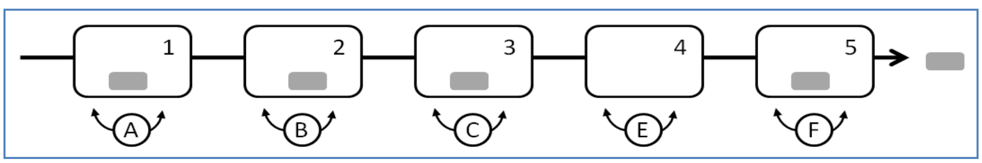


Figure 1: A traditional assembly line with five workstations and fiveoperators.

2.3 U-Shaped Assembly Lines

A layout or design that is widely used in the JIT system is the U-shaped assembly line U-shaped cells are more flexible to changes in demand and production in comparison to traditional assembly lines. Moreover, the number of workers can be lower in JIT systems, since multifunctional operators can be assigned to multiple stations (Miltenburg, 1994).

On the other hand, in U-lines, the material enters and exits on each end of the U. Hence, raw materials arrive into the line and finished goods depart from the line on the same side of the assembly line. As in traditional assembly lines, workers may be assigned to one more workstations, but, the U-shape allows for flexible worker-workstation assignments (Miltenburg, 1994). Workers may cross-over from one leg to the other leg of the U-line. Workers are multi-skilled and cross trained to work in different 'zones' of the line. Figure 3 is an example of a U-line. In this case, there are three workers (A, B, and C) and five workstations (M represents the number of workstations and N the number of workers). Worker A is assigned the first and fifth workstations and hence requires a cross-over.

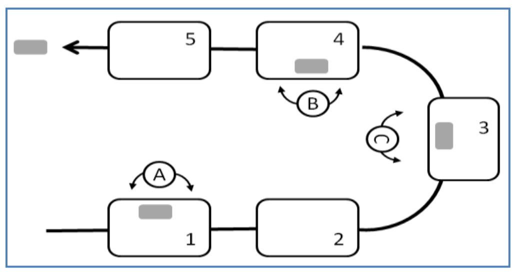


Figure 2: A U-line with three workers and five workstations (M = 5, N = 3)

Miltenburg (2001) notes from surveying U-lines in 114 companies in the US and Japan that the average U-line has 10.2 machines and 3.4 operators. The paper also states that when U-lines were employed in a company that used traditional lines previously, productivity improved by an average of 76%, WIP dropped by 86%, lead time shrunk by 75% and defective rates dropped by 83%. The following are some of the potential advantages of U-lines over traditional lines and the reasons for their popularity (Miltenburg, 1994 and Miltenburg, 2001):

- The input and output of the line are both close to shipping – reduces transportation of goods within the facility

- Lower inventories between stations
- Simpler material handling processes.
- Easier production planning and control
- Opportunities for problem solving
- Visibility and Communications are improved because of the close proximity of operators to each other
- Multi-skilled operators make the workers to stations assignments more flexible and hence, assignments can be changed more easily when production rate needs to be altered or if a bottleneck is encountered.
- The number of workers required on a U-line is usually not more than that required on a standard/traditional line.

2.4 Work Sharing and Self-Balancing Lines

In a traditional assembly line, each station (the term "station" is used synonymously with "workstation") is manned by one worker or operator and that worker is specifically assigned to that station. In a work sharing environment, the number of operators is less than or equal to the number of work stations ($N \leq M$). More importantly, each worker is not restricted to one station – he or she is cross trained to work in multiple stations ahead and/ or behind his station. This helps the worker share work with co-workers, thereby reducing the waiting time and number of jobs in process by taking advantage of extra capacity of faster workers. These types of assembly lines are often called 'self-balancing' lines. These lines are balanced automatically as the line runs - when the operators follow a set of rules, known as a protocol. The most researched work sharing systems in literature that provide such rules or protocols are the bucket brigade system, the modified work-sharing system (MWS) and chaining.

2.4.1 Bucket Brigades

The first practical implementation of the bucket brigades was made in the textile industry, in the 1970s, under the name of "Toyota Sewn Products Management System" –TSS for short (Bratcu & Dolgui, 2005). In bucket brigades, each worker continues to process his job (while walking downstream) until he or she is stopped by his or her successor who then takes the job from him and continues downstream. This worker then moves upstream until he/she in turn pulls from his/her predecessor, and so on. This process of pulling or handing over an incomplete job is known as preempting. There are certain rules that the workers follow to reduce waiting time. Bratcu & Dolgui (2005) present a survey of bucket brigades covering its history in detail.

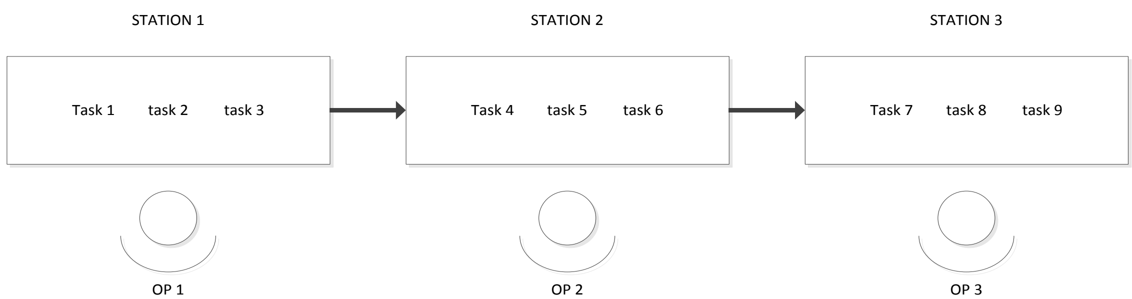


Figure 3: A traditional assembly line with three stations containing three tasks each, and three operators

The rules that workers follow in this system are presented in Bartholdi (1996) as follows**:**

Forward Rule: Continue processing the job in your hand on successive workstations, moving downstream until your successor takes over the work from you or you (in the case of the last worker) reach the end of the line. Then give up the job and follow the backward rule. Never skip your successor.

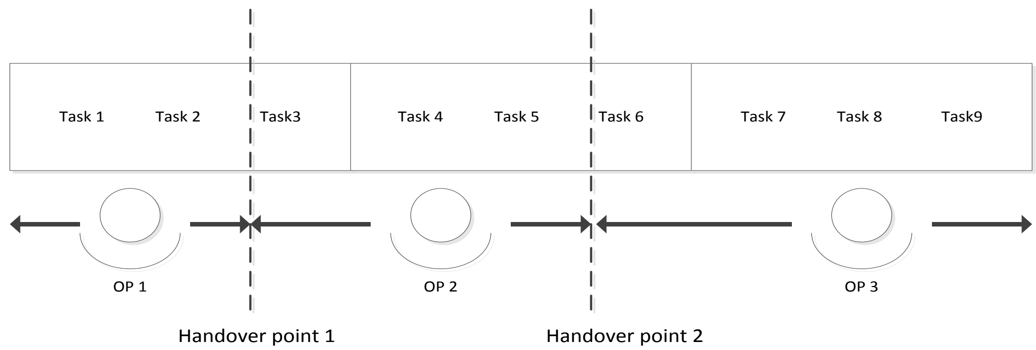


Figure 4: A self-balanced bucket brigade line reaching balance

Backward Rule: Walk back and take over the item of your immediate predecessor and follow the forward rule. If you are the first worker, start a new job and move downstream.

These rules also form the backbone of the Modified Work Sharing (MWS) system (Montano et al., 2007), and the cellular bucket brigade protocol (Lim, 2011). Bartholdi & Eisenstein (1996) proved mathematically that when the TSS (or the bucket brigade) line reaches a state of self-balance when these rules were followed and provided the workers are sequenced from slowest to fastest, the following results hold true:

There exists a fixed point such that if the workers start at positions x*, then they will always reset to position x*, i.e. balance is always possible (the two lines in Figure 4 are examples of these points).

- If the workers are sequenced from the slowest to the fastest, then there exists a unique fixed position and any cycle of worker positions converges to this point.

- If worker velocities (velocities at which workers process the job while moving downstream in the assembly line) are constant and this order (from slowest to fastest worker) is maintained, the production rate is the largest possible.
- If the workers are not arranged in this fashion, the fixed point could act as a 'repeller', so that if the system ever deviates, however slightly from that point, then the system inexorably diverges from it.

2.4.2 Modified Work Sharing System (MWS system)

The MWS system (described in Montano, Villalobos, Gutierrez, & Mar (2007)) is a slight modification of the bucket brigade system. Montano et al. (2007) suggest placing 'buffers' or inventory locations between 'zones'. Each worker primarily works in his or her zone – a series of workstations for which the worker is trained. Instead of being preempted by the worker's successor, the worker drops off the job in a buffer located at the beginning and end of each zone. Each buffer has a pre-assigned control limit for each worker. That is, each worker can only deposit up to a fixed number of parts in each buffer.

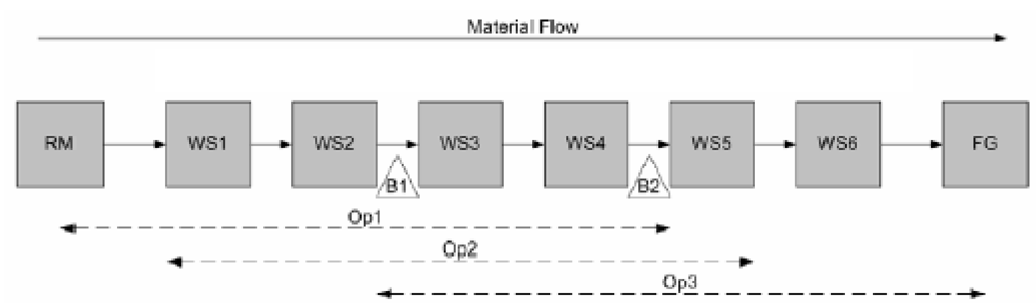


Figure 5: 3 Operators working in their 'zones' in a 6 station line with 2 buffers

When the number of parts in the buffer is equal to the control limit, the worker picks a job from that buffer and starts processing it downstream from his zone. And when the control level of the

buffer at the beginning of his zone becomes zero, he moves upstream to pull from his predecessor. The general rules that the workers are required to follow in this system are:

While moving downstream:

- Advance processing the part downstream until encountering a control buffer or being preempted by a downstream operator.
- If a buffer is encountered and the number of parts in that buffer is equal to or greater than the operator's control number for that buffer, continue processing the current part until encountering the next buffer or being preempted, otherwise deposit the part in the buffer and return upstream.
- If the control number for a buffer is zero leave the part in the buffer and start the upstream motion

While moving upstream:

- When advancing upstream if an operator is encountered, preempt that operator by taking the part being processed by this operator and continue processing downstream. Otherwise, continue advancing upstream until a buffer is encountered.
- If the buffer contents are equal to the operator's control level or less, continue advancing upstream.
- If the buffer contents are greater than the operator's control level, take a part and start the downstream cycle.

- If the operator reaches the upstream limit of his or her work zone and there are no parts in the control buffer stay idle until a part becomes available.

2.4.3 Chaining

Chaining, as explained by Gong, Wang, & Zhang (2011), is similar to the MWS system in the sense that workers work in zones. But these zones overlap each other and there are no buffers. The main advantage of such overlapping zones over the traditional bucket brigades is that it significantly reduces the cross training of workers by limiting how far they can overlap with each other. While in Bucket Brigades all the workers have to be fully cross trained, chaining just requires each worker to be trained in one station downstream and one station upstream of his zone. Practical applications of the chaining protocol have been reported to reduce the WIP in the line that tends to accumulate in the MWS system (Gong, 2011).

2.4.4 Advantages of self-balancing lines over traditional lines

Some of the benefits of the self-balancing systems such as the bucket brigades and MWS over traditional assembly lines, as detailed in archival literature (Bartholdi & Eisenstein (1996), Bratcu & Dolgui (2005) and Ghiram (2012)) can be summarized as follows:

- The need for planning and management, and balancing the line over and over again is reduced considerably in self-balancing lines. Traditional lines often require re-balancing when the demand rate changes, when there is a different mix of products, when there is considerable and frequent worker turnover etc. (Bratcu & Dolgui, 2005).
- Self-balancing lines are more flexible and agile. The throughput of the system can be changed by simply adding or removing an operator in the line. The line is also more ready to respond to takt time changes and when the product mix is high (Bratcu & Dolgui, 2005).

- The throughput is increased in most cases, because self-balancing lines spontaneously generate the optimal division of work, taking advantage of the faster workers - who tend to take up more share of the work (Ghiram, 2012).
- Though training of workers in traditional lines is easier, coordination of the workers is made easy in self-balancing lines because it is easy for workers to know what to do next. The workers also benefit from each-others' knowledge and experience, in a way receiving more training (Ghiram, 2012).
- In traditional lines, each operator is focused only on the worker's unique standard amount of work content and is hence restricted to the worker's workstation. In self-balancing processes, the workers are made to work together as a team – which boosts worker morale, improves communication between them, and allows them to learn from each-others' experience (Ghiram, 2012).

In Ghiram (2012), self-balancing lines are claimed experientially to be better over traditional lines from a management perspective. The classic approach works well if the company has (or has committed to developing) a complete lean management system that responds to problems every takt time because it seeks to force leveling and takt-pacing. However, since most organizations do not have the management capacity to respond every cycle time, Self-Balancing proves an extremely efficient and effective delivery method while they work to develop the management support systems. Ghiram (2012) considers self-balancing lines as the means to achieve true lean continuous flow, as the focus of trouble shooting is more on the stoppage of material flow and waiting of workers than being worker-process centric. Ghiram (2012) also highlights the benefits of self-balancing and claims that some benefits will be realized over time as the people involved gain a deeper understanding of the process.

3. Literature review

The literature that was reviewed for this thesis is detailed in this section under three topics: work-sharing methods in linear assembly lines, cellular bucket brigades (linear assembly lines bent in half to resemble U-lines) and the U-line balancing problem. These three topics are discussed in sections 3.1,3.2 and 3.3 respectively. In section 3.4, the gap that was identified in this section is explained.

3.1 Work-Sharing methods in linear assembly lines

In this section, two important work-sharing methods in literature that are relevant to this thesis topic are reviewed: the Bucket Brigades system (section 3.1.1) and the Modified Work-sharing System (section 3.1.2). In each section, the protocols that the workers follow in these systems are explained, and their benefits and drawbacks are analyzed. Other work-sharing systems that have surfaced during this literature review are mentioned in section 3.1.3.

3.1.1 Bucket Brigades

One of the first papers in the field of the bucket-brigade system of work-sharing was that of Bartholdi & Eisenstein (1996). This paper explains succinctly the forward and backward rules that every worker is required to follow independently for a bucket brigade system to work in a serial assembly line. Although these rules and postulates hold true for the model, the real world scenario is a contrast to the assumptions made in this paper. The authors assume that all the work is done in deterministic times– providing no room for variations in the process which occur often in a real manufacturing environment – and assume instantaneous walk-back, i.e. it takes the worker no time to execute the backward rule.

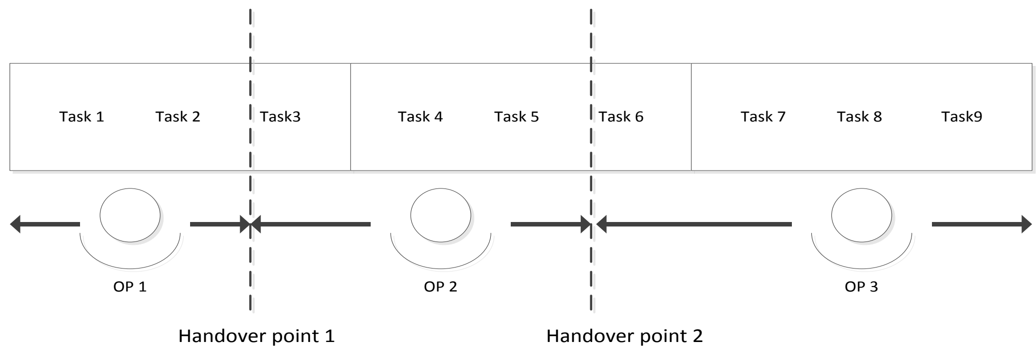


Figure 6: A balanced bucket brigade system of nine tasks, three workstations, and three workers

The authors also consider workers with a large variation in their work-velocities. While this may hold true for the apparel industry that the authors have collected data from, it may not hold true for other industries where the workforce is mostly at the same skill level. They also assume continuous work handover i.e. the jobs should be immediately preempted and transferred (when preemption is invoked according to the rules). This has been a cause of a variety of quality issues (Armbruster & Gel, 2006).

The deterministic assumptions in Bartholdi & Eisenstein (1996) were relaxed to accommodate stochastic processing times in other extension papers. Bartholdi, Eisenstein, & Foley (2001) have proved by convergence analysis in topographic spaces that the more the number of work stations in a stochastic system, the more its behavior will tend to the behavior of a deterministic one (assuming the order of workers does not change). An exception to this rule was observed when the number of workstations is almost equal to the number of workers. It is to be noted that the optimality of the ordering of workers from slowest to fastest has not yet been proved for the stochastic case. Zavadlav, McClain, & Thomas (1996), while considering systems in which the

number of workers are more than the number of stations, also considers identical worker velocities.

Another cause for concern is when the a worker placed upstream ‘catches up’ to a worker placed more downstream. This leads to certain amount of waiting experienced by the upstream worker. This scenario is commonly referred to as "blocking". Lim & Yang (2009), attempted to find policies that maximize the throughput of the line for a given work distribution on stations to avoid the blocking scenario. They found that the sequencing of workers from slowest to fastest outperforms other policies for most work distributions except for some cases (such as when the work content for the 1st station is much larger than the work content for the 2nd station). In these cases, limiting workers to zones provided higher throughput.

To counter variability in the processing times in a bucket brigade, temporary buffers may be used in the system. Although this could increase the WIP in the system, it may considerably reduce the cost caused by other factors such as waiting and blocking. McClain, Schultz, & Thomas (2012), observe that when worker velocities are nearly equal, even when the machine-to-worker ratios are high, bucket brigades perform poorly (lower throughput, more waiting etc.) ; the ‘drop off’ rule performs better in this case. This paper mainly focuses on environments in which processing time variability challenges the flexibility of the system and low machine-to-worker ratio limits the amount of sharing that is possible.

3.1.2 Modified Work-sharing System (MWS)

The MWS system is a slight modification of the bucket brigade system (Montano, Villalobos, Gutierrez, & Mar (2007)). Montano et al. (2007) suggest placing ‘buffers’ or inventory locations between ‘zones’. Each worker primarily works in his or her zone – a series of workstations for

which the worker is trained. Instead of being preempted by the worker's successor, the worker drops off the job in a buffer located at the beginning and end of each zone. Each buffer has a pre-assigned control limit for each worker. Figure 7 illustrates these zones. Buffers B1 and B2 will each have 3 separate buffers – one for each operator. Thus, each worker can only deposit up to a certain fixed number of parts in each buffer. When the number of parts in the buffer is equal to the control limit, the worker picks a job from that buffer and starts processing it downstream from his zone. And when the control level of the buffer at the beginning of his zone becomes zero, the worker moves upstream to pull from his/her predecessor.

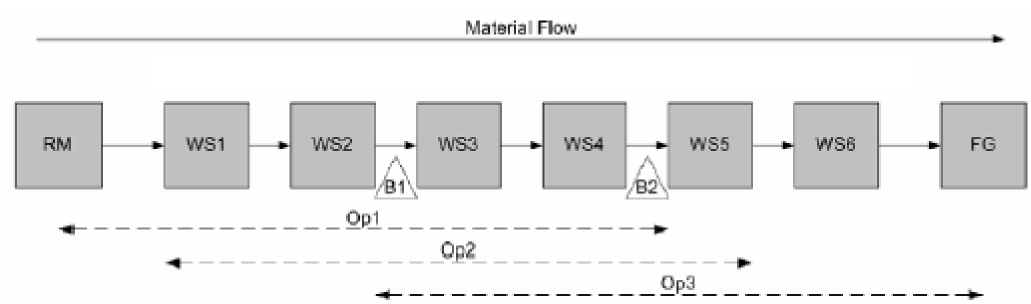


Figure 7: The MWS system for a three worker - six station line with two buffers (B1 and B2)

The control level is represented as C_{ij}, where *i* denotes the buffer location and *j* denotes the operator for whom the control level applies. For example, $C_{11} = 3$ means that the control level for operator 1 in buffer 1 would be 3. In Figure 7 (from Montano et al., 2007), the dashed lines represent the zones for each operator. The authors suggest that these zones can be accomplished with the following values of control levels:

$$C_{11} = a \qquad C_{21} = \infty$$

$$C_{12} = 0 \qquad C_{22} = b$$

$$C_{13} = \infty \qquad C_{23} = 0$$

It can be noted that a worker in MWS will move forward processing the part as much as possible, but the amount a worker can advance is limited by both, the potential preemption by the downstream worker, and by the control buffer levels encountered as the worker moves downstream.

It can also be observed that this system would behave in one extreme as a bucket brigade system if all the control levels are 0, and would behave as a traditional system (with almost constant WIP, or a balanced line) in the other extreme. The flexibility offered by this buffer system changes the pure pull system of a bucket brigade to a combination of push and pull systems. Montano et al. (2007) model this MWS system in a similar fashion to the bucket brigades, and draw comparisons (Figure 8) between the them.

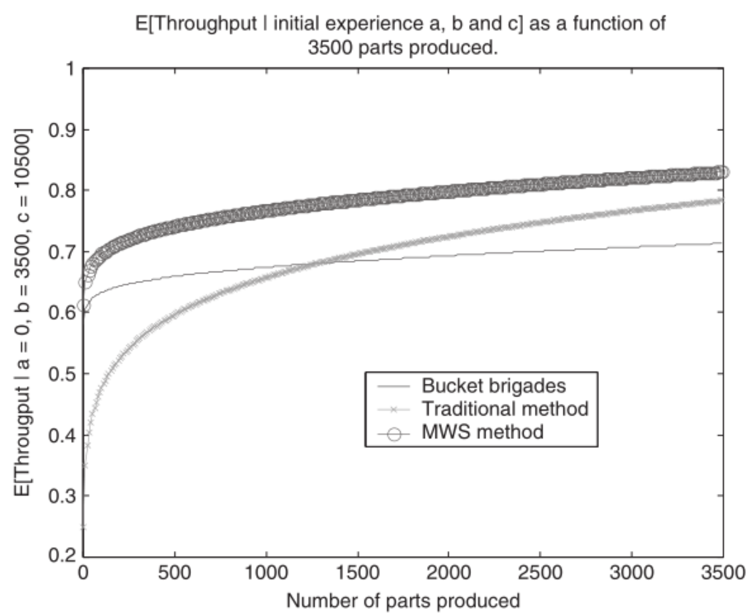


Figure 8: Expected throughput as a function of learning process (source: Montano et al.2007)

They conclude that the MWS method is a good alternative for use in high labor turnover environments where fully cross trained operators are often found and tool replication does not

represent a major investment. However, both the latter conditions of fully cross trained operators and cheap replicate tooling are not often found in real-world manufacturing.

3.1.3 Other work-sharing systems in literature

Jordan et.al (2004) analyzes the traditional cross-training policy of chaining, in which the worker is trained in the tasks of the next station apart from his own. This helps when the worker next to this worker lags behind, and also when the throughput has to be sustained in cases of worker absenteism. Gong et al.(2011) introduces a new workforce cross-training policy for U-shaped assembly lines by improving the traditional chaining policy. The paper suggests that rather than training the workers to perform tasks only next in sequence to his task, they should be trained to perform tasks before and after their own task. This will make the system much more flexible and balanced when absenteeism occurs. Experiments were conducted to compare the traditional training policy with the new training policy and it was found that the new policy produced better throughput and more flexibility. This work does not present a comparison with the bucket brigade or any other work sharing system. The workers could still be starved when the previous worker lags behind significantly. Unlike in the MWS system, the preemption rules are not used in chaining and hence, the workers may have more idle time, leading to less throughput and lower efficiencies.

3.2 Cellular Bucket Brigades

Lim (2011) suggests a new design for bucket brigades that reduces unproductive walk back time, claiming up to 30% improvement in throughput when compared to a bucket brigade system. The design (as shown in Figure 9) consists of a serial line folded in half. In the first half of the line, workers process the work forward, then at the end of the line carry the work over to the second half of the line where the work is processed backwards. Consistent with the characteristics of a

traditional U-line, the beginning and end of the assembly line are right next to each other. The two halves of the line are also called as the two ‘legs’ of the U-line.

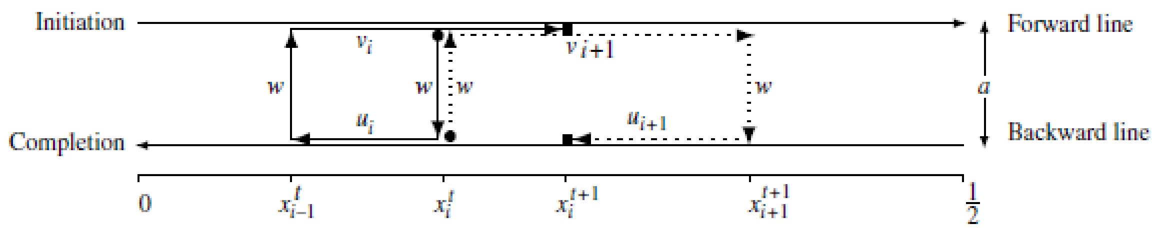


Figure 9: A cellular bucket brigade. The paths of workers i and i+1 between two successive exchanges (at x_i^t and x_i^{t+1}) are shown. Source: Lim (2011).

The author models this design and proposes new rules for the workers to follow:

Work Forward (on first-half of the line): Continue to process the job forward until one of the following happens:

1. You are preempted by your successor, then exchange jobs with him and then work backward on the second-half of the line.
2. You reach the end of the forward line if you are the last worker, then carry the job to the second-half of the line and continue to work backward on the same job.
3. You catch up to your successor working forward, then slow down and continue to process job at his pace.
4. You catch up with your successor who is crossing the aisle (from forward to backward line), then wait.

Work Backward (on the second-half of the line): Continue to process your work until one of the following happens:

1. You preempt your predecessor who is working on the forward line and exchange work with him/her and then continue to work forward
2. You complete your job at the end of the backward line (if you are the first worker), cross-over and initiate a new item
3. You catch up with your predecessor while crossing the aisle, then wait.
4. If you catch up with him on the same aisle, slow down and continue working the job at the worker's velocity.

Waiting rules: Continue carrying your item

1. If you are on the forward line, remain idle until your successor has finished crossing the aisle, then work forward
2. If you are on the backward line, remain idle until your predecessor has finished crossing the aisle, then work backward.

Analysis of the protocol using the model presented in the paper shows that the waiting rule will never be invoked under normal operation of a properly configured line. Let 'a' be the aisle width for this line and 'w' be the velocity with which workers cross this aisle. Similar to the bucket brigades (refer to Figure 9), there exists a fixed point x* (x is a real number between 0 and 1, representing the position on the line. x will be between 0 and ½ for the forward line and between ½ and 1 for the backward line). If v_i and u_i are the forward and backward velocities of worker i, then θ_i is defined as: $\theta_i = (1/v_i + 1/u_i)^{-1}$. For this fixed point x*, the following holds true:

$$x_i^* = \frac{\sum_{j=1}^{i} \theta_i}{\sum_{j=1}^{n} \theta_{ji}}, \qquad i = 1, \ldots, n-1$$

This point acts as the point of self-balance (i.e. x* is an attractor), provided the following condition is satisfied:

$$\frac{1}{v_1} - \frac{1}{u_1} > \frac{1}{v_2} - \frac{1}{u_2} > \cdots > \frac{1}{v_n} - \frac{1}{u_n}$$

The throughput increases with the number of workers ‘n’, but the increase in throughput is less than linear because of the corresponding increase in unproductive travel. On comparing serial bucket brigades with this cellular system, the author finds that the cellular bucket brigade is more productive than serial lines when the width of the aisle and/or number of workers are sufficiently small (Figure 10).

From Figure 10, it is evident that the cellular bucket brigades perform better than serial lines when $a \leq 0.04$ (i.e. the aisle is less than 4% of the total length of the assembly line). This seems like an unlikely situation, because the aisle has to be wide enough to accommodate workers while moving.

Similar to Bartholdi & Eisenstein (1996) , this paper has considered a real line model of an assembly line. Although this is convenient for modeling and drawing results, it does not represent the real world situation. In the real world, tasks are not usually continuously laid out on the assembly line – a set of tasks is associated with each workstation. If cellular bucket brigades were to be adapted for U-lines, the discrete nature of the line will have to be considered – each assembly line consists of a certain number of fixed stations. Each of these stations (or locations of tasks) are generally preferred to be completed by the same operator. Also, the complex nature of the rules might confuse workers, who not only have to progress with their work, but are now also expected to look back frequently to decide whether to cross-over and exchange the work or not. Another

drawback of this model is the assumption of deterministic processing times without taking into consideration process variability.

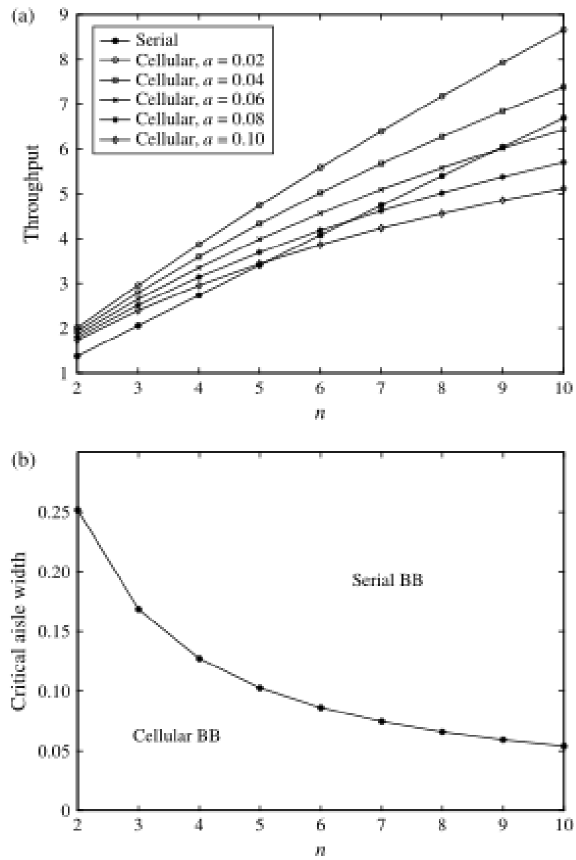


Figure 10: (a) Comparison of serial BB with cellular BB for different aisle widths. (b) Aisle width vs number of workers plot. Source: Lim (2011).

3.3 The U-line balancing problem and worker allocations

The U-line balancing problem can be defined as the assignment of approximately same amount of workload to each workstation or worker in a U-shaped assembly line, as well as assigning workers to a set of workstations (as generally n<m). U-shaped line balancing problems have been investigated since 1994 with the evolution of JIT.

Miltenburg (1994) investigated the line balancing problem for U-shaped lines. Purpose of this study was two folded. One was to offer a new line balancing algorithm for U-shaped lines, and

second is to prove that traditional line algorithms can be successfully adapted to new problems. Other papers (Urban, 1998; Scholl & Klein, 1999; Gökçen & Agpak, 2006; and Erel, Sabuncuoglu, & Aksu, 2001) have improved on Miltenburg's balancing problem and have come up with different approaches or have suggested improvements in the problem to solve the U-line balancing problem. Most of the archival research investigated aims to minimize number of stations (workers). Since they have adapted the linear assembly line approach and as each worker can work on multiple stations in a U-line, they use the term 'station' to represent the set of work-stations that each worker works in. Thus, making the number of stations equal to the number of workers. They minimize the number of workers instead of physical work station, since a worker can work more than one station in u-line.

All line balancing research has been done for stable environments. Difference between stable and dynamic environment is the frequency of rebalancing. If rebalancing is required often due to variations in production or demand, it is considered as a dynamic environment. Rebalancing technique should be easy and less costly. Proposed algorithms from previous research require the redesigning of work stations and task assignments at each station. Redesigning the cell from beginning is still costly and time-consuming. Erin (2007) provides a more detailed research on the U-line balancing problem literature.

Research in worker allocation aims to prove that optimizing workforce assignment is as essential as line balancing itself. The majority of the research in line balancing focuses on the minimization of the number of workers (stations) without considering walking times and waiting times. However, it is not a practical approach in real life. Nakade and Ohno (1997) criticized Miltenberg and Wijingard (1994) approach to U-line balancing problem - it only takes into account the minimization of number of stations and ignores the crossovers in walk paths. Ohno & Nakade

(1997) proposed Petri Net and GSMP theories to prove that reduction in variances of operation and walking times of workers increases the throughput. In addition, throughput is same for reverse system, as well. Nakade & Ohno (1999) proposed a model for deterministic walking and process time cases. Erin (2007) also considers worker walking time. In this paper, first minimum number of workers is determined under given cycle time, and then an optimal worker allocation with minimum number of operators is proposed.

Nakade & Ohno (2003) worked on separate, and carousel type of allocations of workers using both deterministic and stochastic times. Each worker was responsible for specific machine groups in the separate system, while every worker was allocated to all machines in the carousel system. This study showed that if workload between operators are same in separate allocation, system cycle time is smaller than carousel allocation. However, if the difference is big between operators in deterministic time, carousel allocation is better when time is stochastic. Erin (2007) provides a table outlining the details of the literature in this field.

3.4 Research Gap

Very limited archival literature is currently available on bucket brigades in U-lines. Even fewer have considered worker crossovers, stochastic processing times, and discrete tasks. Crossing over of workers is one of the main advantages of the U-line and makes the U-line a more flexible assembly line balancing problem as a larger set of worker to station assignments are possible (Miltenburg, 2001). Considering stochastic processing times are important to take into account the variability in the time it takes the worker to complete a task. Also, most of the literature available in bucket brigades (specifically Bartholdi et al., 1996; Bartholdi et al.,1999; and Lim, 2011) assumes a continuous assembly line in which workers continuously advance as they process the

job. Few papers (Bartholdi et al., 2001; Montano et al., 2007) consider the discrete breakdown of a process. This is nothing but breaking down the assembly line into a finite number of relatively smaller tasks (or steps), as opposed to a continuous system, which can be conceived as breaking down the entire assembly line into infinitesimally small tasks.. Although cellular bucket brigades (proposed in Lim, 2011) comes close to addressing this gap in literature, it does not consider process variability or the importance of preempting or exchanging only after a task is completed – two important factors that contribute to the practicality of a protocol (outlined in Bratcu et al.,2005; and Montano et al., 2007). On surveying three local manufacturing facilities, the importance of these two issues was further strengthened. They considered it to be a quality related best-practice to not divide a task between two workers, and practically imperative to consider variations in task-times as inherent to the assembly line. Cellular bucket brigades also assume a continuous assembly line and very small aisle widths (less than 10% only, with 4% being the critical width), limiting its practicality.

Hence, a literature gap was identified in the area of a work-sharing protocol for U-shaped assembly lines that considers discrete tasks and worker crossovers using stochastic processing times.

4. Problem Statement

The literature gap outlined at the end of the previous section specifically outlines problems in contemporary work-sharing protocols that can be addressed. The objective of this thesis is to develop a novel self-balancing work sharing protocol for two worker U-shaped assembly lines that will accommodate:

1. Worker crossovers
2. Discrete tasks (i.e. no exchange/ preemption during a task)
3. Stochastic processing times

Specific sets of research questions that this thesis aims to answer are:

1. Can an effective work sharing protocol (that approaches self-balancing) be developed for U-shaped assembly lines? Can it provide advantages of work-sharing similar to those obtained when the MWS and the CBB protocols are employed?

2. What are the factors that affect this protocol? How does the performance of these U-lines (when this protocol is employed) compare with the performance of traditionally balanced U-lines and cellular bucket brigades (in terms of metrics such as utilization, throughput, WIP, cycle time and number of preemptions)?

3. Is this protocol generalizable (with respect to number of stations, workers, various processing times, types of processes, aisle width to line length ratio, etc.)? What are the capabilities and limitations?

5. Methodology

In this section, the methodology adopted to answer the research questions posed in the previous section is detailed. The methodology primarily consisted of two stages. In the first stage, a new protocol along with a working model was developed. In the second, the performance of this protocol was analyzed, evaluated and contrasted against the performance of existing protocol in literature. Figure 11 is a schematic on the steps involved in this methodology.

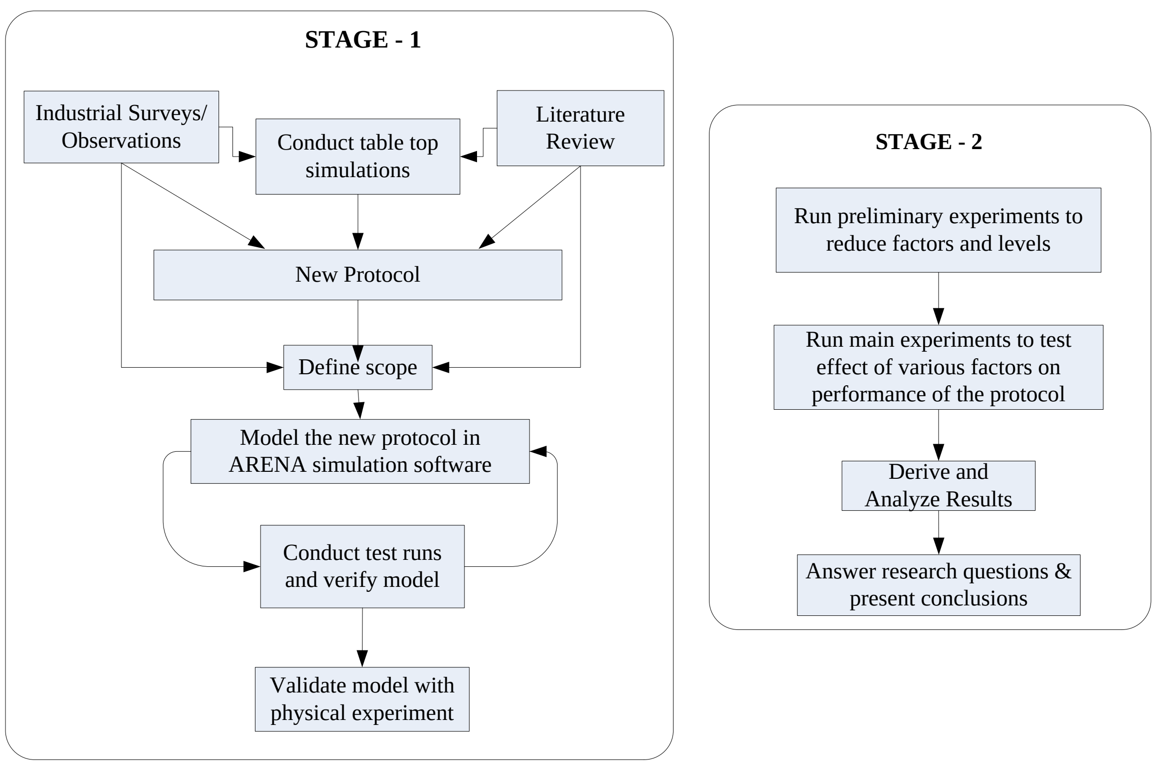


Figure 11: Methodology flowchart

In the first stage, an iterative process using simple table-top simulations to gain a preliminary understanding of how work-sharing systems behaved in U-lines, resulted in a

hypothesis for a new protocol. This hypothesis suggested that it was possible to combine the benefits of both the CBB system and the MWS protocol by including buffers with control levels in a U- line that followed the cellular bucket brigade protocol (research question 1). Observations from the literature review and industrial surveys helped strengthen the protocol, arrive at a generalized protocol, and narrow down its scope to a feasible size (detailed in sections 5.2.1, 5.2.2, and 5.2.3). To test the new protocol, simulation models in ARENA, a discrete event simulation software, were developed (section 5.2.4). Then, an iterative process to modify the model to more accurately represent the protocol was undertaken. This involved conducting test runs and verifying the model. The simulation model was then validated by conducting a physical experiment. This involved staging and physically simulating the protocol with volunteers in a U-shaped assembly line set-up in the Toyota Production Systems Lab at RIT (section 5.2.5). Once validated, the simulation model was used to run a preliminary round of experiments to screen factors and to check if exploring more levels of that factor would be fruitful (section 6.1). After narrowing down factors and levels, a final round of experiments was conducted (section 6.2). The results of these experiments were then analyzed, and attempts were made to answer concerns that arose during various stages of the thesis. Finally, conclusions were drawn from the analyses, and research questions were answered (section 7).

5.1 Definitions and Assumptions

Some of the terms that will be used in this work are defined as follows (refer Figure 12 and Figure 13):

- **Worker (Operator)**: A person who performs the tasks assigned to him/her in an assembly line. It is assumed that the tasks are sequenced and allocated along the length of the line and that the operator moves down the line as he/she completes each task. The index *i* is used to refer to a worker and the total number of workers in the assembly line is represented by N.
- **Stations and tasks:** The entire work content of the assembly line can be divided into a number of smaller divisions of equal work content, called stations. The total number of stations is represented by M. Every station in-turn consists of smaller sub-divisions of equal work content, called tasks. The number of tasks for each workstation is denoted by T_j, where the index j = 1,2,.., M. Figure 12 shows a U-line divided into 6 stations and 24 tasks.

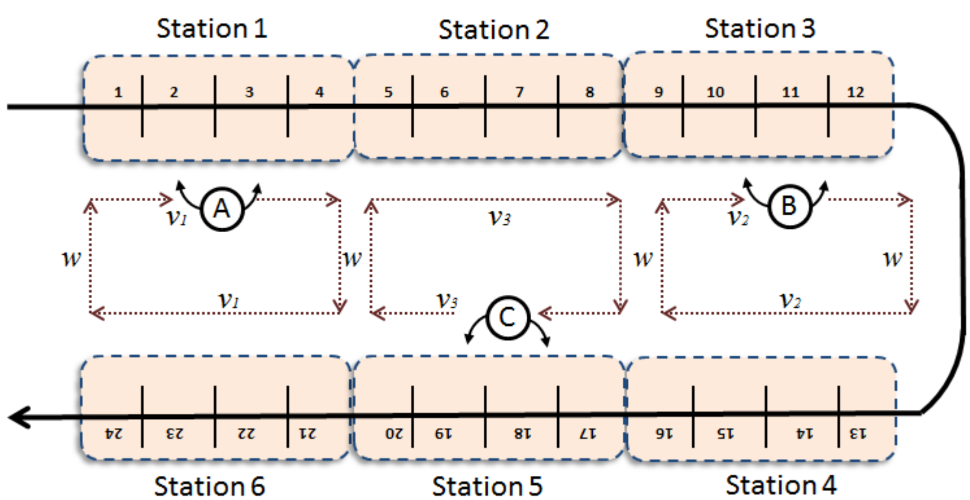


Figure 12: A 6 station 3 worker U-line in which the entire process is divided into 24 with workers having equal work velocities $v_1 = v_2 = v_3$ and aisle walking velocity w

- **Buffers:** Buffers are physical locations used to temporarily store WIP in the assembly line. There is one buffer between every pair of tasks. In literature

(Montano et.al, 2007), buffers are located between pairs of consecutive workstations, as well as one before the first workstation, and one after the last one (Figure 13 illustrates this). In this case, for M workstations, there will be a total of (M + 1) buffers, represented as: B_1, B_2, B_3,…,B_{M+1}. The buffer preceding station 1 is referred to as B_1, the buffer preceding the second workstation as B_2, and so on until the last buffer. The last buffer succeeds the M^{th} workstation and is referred to as B_{M+1}. If buffers were instead located between tasks, for a T tasks line, there will be a total of (T + 1) buffers, represented as: B_1, B_2, B_3,…,B_{T+1}.

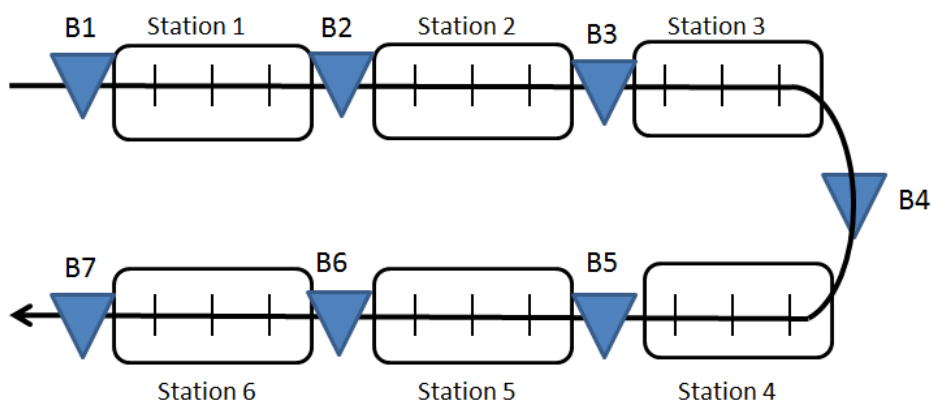


Figure 13: An example of the location of buffers introduced in the 6 station U-line (M=6) shown in Figure 12

- **Work-zone** (or zone): An area consisting of a combination of stations or tasks within which the worker works. For example, in Figure 12, worker A's zone consists of stations 1 and 6.
- **Worker velocity** ,v_i: Each worker (i = 1...N) is assumed to have a finite work velocity v_i, which can be defined as the assembly speed (or job processing speed) of the i^{th} worker, given in parts per unit time (Montano et al.,2007). It is to be noted that this velocity includes walking while performing the task.

- **Aisle width**, *a*: The ratio of the width of the aisle to the total length of the line. It can take any value from 0 to 1,i.e. $a \in [0,1]$ (Lim, 2011).
- **Walking velocity** (or Walk velocity, as used in Lim, 2011) *w*: All workers walk with a finite walk velocity, *w* units per unit time. It is important to distinguish walking velocity from worker velocity. When a worker is not assembling (or processing) a job, but moves from one location to another in the U-line, he/she does so with a walking velocity, *w*. Hence, for a line of length L and aisle width ratio *a*, the time taken by each worker to cross the aisle will be (aL/w) units. If all T tasks in a line of length L are equally spaced, then distance between two consecutive tasks on the same side of the aisle will be (L/T), and the time taken to walk back from one of task to the previous task on the same side of the aisle will be $L/(Tw)$.
- **WIP**: Work in Process - the total number of jobs present in the entire system

The following assumptions are made in this work:

1. Each workstation consists of a number of discrete tasks T_j, j = 1..M (where M is the total number of workstations). Figure 12 shows an example of this.
2. Workers can be preempted only after a task is completed (i.e. no preemption in the middle of a task)
3. The average work content is the same for all tasks i.e. a worker would take the same amount of time for every task, if there were no variability.
4. Task times are distributed based on a Gamma distribution
5. Walking velocity is a normal distribution with a mean of 4.39 ft/sec, and a variance of 1.21 ft/sec.

Assumption 1 is to address the mismatch between continuous models used by most of the previous authors (such as Bartholdi et al., 1996, 1999; Lim, 2011) and discrete stations used in the industry (outlined in Bratcu et al. (2005) and used in Bartholdi (2001), and from a local industry survey). This also allows for the lean practice of 'standard work' - where each station is described as a list of tasks to be performed by every worker in order to minimize defects (Ghiram, 2012). The second assumption follows the first assumption. Besides being impractical, if a worker were to be preempted during a task, unforeseen quality issues could occur due to a possible loss of information and more than one person performing the same task (which increases the variability in the quality of the product). Assumption 3 is from previous literature (Bartholdi et al. (2001), Montano et al. (2007)) and is in place to keep the focus on balancing the workload between the workers rather than focusing on assigning work to workstations.

Assumption 4 is to consider the variability in processing times. Across the manufacturing industry, processing times vary greatly depending on the product being manufactured, tools and machinery used, training imparted to the operators, operators' experience, number of stations in the assembly line, number of workers, etc. Within each assembly line task itself, there may be some sources of variability due to human errors and other internal process variability. Bartholdi et.al (1999), Miltenburg et.al (2007), and several other works in literature that consider stochastic processing times chose an exponential distribution. Their primary argument for doing so is that an exponential distribution is a conservative assumption. This means that there will be greater variance at each work station than one would expect to find in practice, and this unrealistically large variance reduces the throughput of bucket brigades because it increases the chances of blocking

(Bartholdi et.al, 1999). Although this is a reasonable assumption to make, it compromises on the opportunity to analyze whether the system behaves differently for small and large variances.

On the other hand, normal distributions are commonly used to represent quantities that are expected to be the sum of several independent processes (including human errors). Das, Garcia-Diaz, MacDonald & Ghoshal(2010), Scholl (1999), and Smunt & Perkins (1985) among others, consider normally distributed task times in their approach to stochastic assembly line balancing problems.

One of the assumptions made in this thesis work is that the average work content is the same for all workstations, i.e. a worker will take the same amount of time to finish the work in every station if there is no variability. This work also does not consider any scheduled or unscheduled stoppages to the processes either due to equipment failures, time breaks, maintenance requirements, workers' learning curves, or process failures. The sources of variability are thus limited to human errors while processing the job(captured in this work by different worker velocities) and walking (captured by walking velocities). Mason et al. (2005) considered the impact of Human Performance Variance (HPV), and analyzed the fit of various empirical distributions such as Pearson IV, Weibull, Gamma, and Normal. The conclusion was that Pearson IV ($r^2 = 0.997$) and Gamma ($r^2 = 0.975$ to 0.991) represent the manufacturing operation process the closest (with coefficients of variances, CV, varying from 0.3 to 0.6). Pearson IV has only slightly a higher correlation coefficient r^2 than Gamma. While Gamma distribution is commonly used included in other literature such as MacDonald & Ghoshal(2010), Scholl (1999), Person IV is mentioned only in Mason et al. (2005). Hence, considering all the factors discussed

above, for this work, Gamma distribution is used as the empirical distribution for task times.

While Bartholdi & Eisenstein (1999) and Miltenburg et al. (2007) consider walking velocities as a fraction of the working velocities, this work uses empirical data for the walking velocities. Assumption 5 is made based on data on human walking speeds from Daamen & Hoogendoom (2006), Mohler et al. (2007), and Levine & Norenzayan (1999). The rationale behind this choice was that walking that occurs when the protocol is followed is no different from walking in normal everyday life. The walking velocities of individuals appeared to follow a normal distribution, with an estimated mean of 1.34 m/s (4.39 ft/s) and a standard deviation of 0.37 m/s (1.21 ft/s). The data on which the distributions were estimated came from large-scale laboratory walking experiments in more than twenty countries.

5.2 Stage -1

To develop a new protocol, review of existing protocols in literature was essential. Simple table top simulations were used for this purpose. Big sheets of paper, markers, and objects such as paper clips and pins were used to simulate U-shaped assembly lines. A simple U-line of six stations and two workers was used as an example. Role-playing was used to simulate time. Worker positions, buffer levels, and cycle times were annotated. First, a simple CBB protocol was simulated, and then a simple linear MWS line with buffers was simulated. When MWS buffers were introduced into the CBB line, the simulation quickly grew in complexity, and it became difficult to keep track of all the metrics. Also, while simulating deterministic cases on a table-top was feasible, simulating

stochastic cases was not. At this point, a hypothesis was developed based on observations made during these manual table-top simulations: CBB protocol with MWS-like buffers will produce a higher throughput than the traditional CBB system under similar conditions. The rationale behind this hypothesis was that buffers would reduce the waiting time that existed in the CBB protocol when workers exchanged jobs across the aisle. Although this introduced new walking times, the trade-off between increasing walking times, and decreasing waiting times seemed to result in higher efficiency, and hence higher throughput.

5.2.1 Developing the New Protocol

The new protocol can be viewed as a simple combination of the Cellular Bucket Brigade system introduced by Lim (2011), and the Modified Work-Sharing protocol introduced by Montano et al. (2007). However, there were some connecting rules and overlapping rules that were appended and deleted from these protocols. The approach to developing the new protocol was using simple, specific case studies to simulate various conditions and situations, and then generalizing them based on lessons learnt from the simulation model in ARENA. Due to their simplicity, the most common examples found in literature are the 6station - 2 worker, and 6 station - 3 worker assembly lines (Bartholdi & Eisenstein 1996, 2001, Montano et.al, 2007, Lim, 2011). Hence, these also used as the starting point to study various permutations and combinations of the rules. Here, first the protocol is explained through an example, and then the generalized rules are presented. For this example, a 6 station - 2 worker U-line is considered, with each station having 4 tasks, and the workers following both the MWS, and the CBB protocols (Figure 14). This U-line will contain 7 buffers (B1 through B7), and 24 tasks. Control levels, C_{ij} , that were

considered for each worker *i* - buffer *j* combination are mentioned in Figure 14. Worker B with velocity v_2, is assumed to have twice the work velocity of A with velocity v_1 ($v_2 = 2v_1$). In the beginning, worker A and worker B start processing from tasks 1 and 13 respectively. What will happen from this time forward is explained below under three scenarios or situations. The first two scenarios are observed when the process times are deterministic, while the third scenario deals with what would happen in the first two conditions with stochastic processing times.

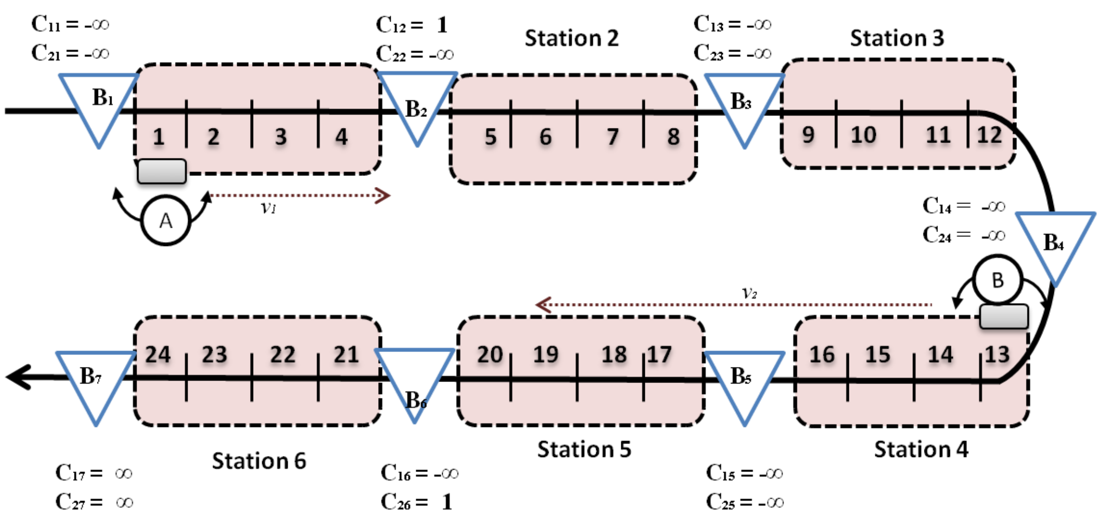


Figure 14: Schematic of a six station, two workers U-line, with seven buffers and control levels.

Note that in this case, all control levels at B1, B3, B4, and B5 are -∞. This means that when workers A and B encounter these buffers, they behave as if there was no buffer (even zero is greater than -∞). Both the control levels at B1 are -∞, as the assumption is that there is infinite raw material; the control levels at B7 are ∞, as this is represents finished goods storage. B2 has a control level of 1 for worker A, while B6 has a control level of 1 for worker B.

Scenario 1 - Building WIP and exchanging: As worker A approaches B2 after completing task 4 and as B approaches B6 after completing task 20 (B will cover twice as many stations as A does as $v_2 = 2v_1$), they will first reach their non $-\infty$ buffers. At this point, they will follow the MWS protocol, i.e. they check the buffer to see if current Buffer Level (BL) is greater than Control Level (CL, C_{ij}). Note: current buffer level (BL) includes the part that the worker carries to the buffer. As this condition is not true when they approach B1 and B6 for the first time, they will both drop off their jobs at B2 and B6 respectively, and walk back till they either reach a buffer that satisfies BL > CL condition, or they encounter a fellow worker.

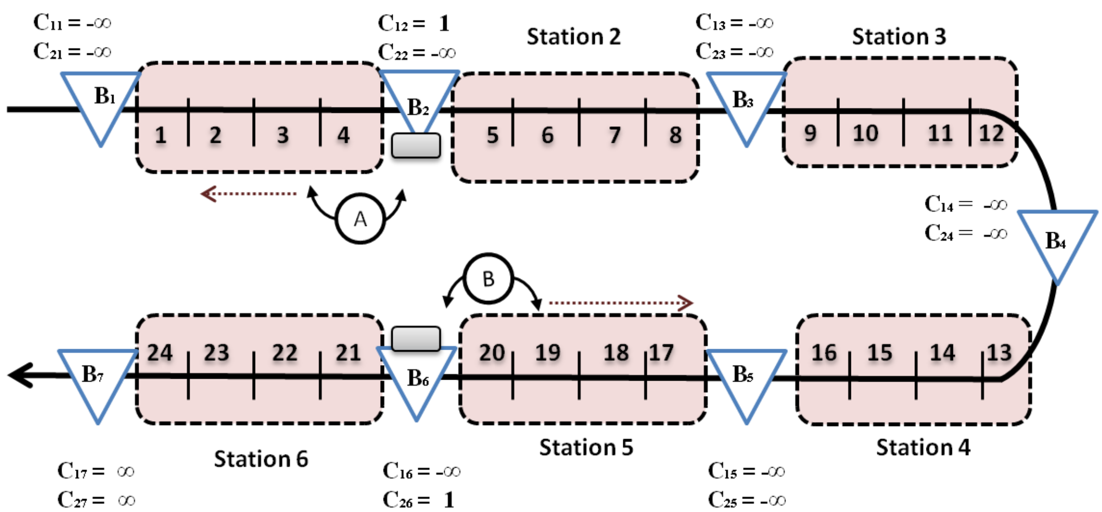


Figure 15: Example case: When A and B drop their jobs at buffers 2 and 6 respectively by following the MWS protocol

In this case, as worker B walks back till B2, A walks back to B1. Both will pick a job from the respective buffers, start processing the job forward. A reaches B1 before B reaches B2 as $2v_1 = v_2$. A and B will both pick up their next jobs from buffers B1 and B2, and start processing their jobs forward. This time when A completes task 4, B will be at task 12. A checks B2 and finds that BL = 1 is not greater than CL = 1 (as B has already

picked up the job from B2), and hence drops the part again at B2 and begins walking back to B1. At this point, the line will look like it is shown in Figure 16.

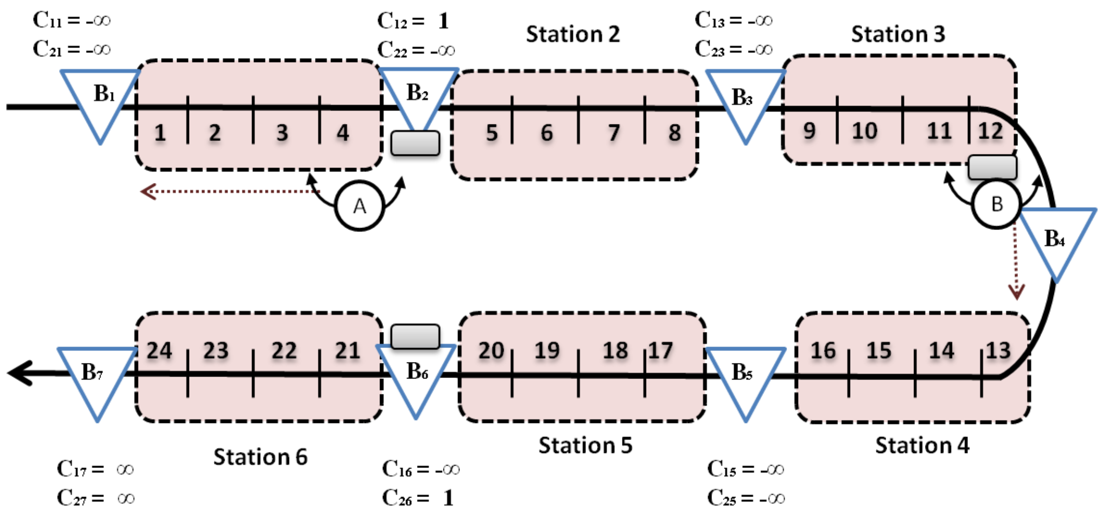


Figure 16: Example case: When A drops job at B2, B is at task 12.

When B completes the 20th task, A will be at task 4. Now, there are two possibilities depending on who finishes the task first:

1. If A completes task 4 before B completes task 20, A will proceed to buffer B2. This time for A, BL = 2 is greater than CL = 1. Hence, A will continue to task 5. Meanwhile, worker B is ready to drop the part in buffer B6, but as A will now be approaching task 5 (past the buffer), they are at the same point in the line (at diagonally opposite tasks). Hence, the CBB protocol will take effect, and A and B will exchange jobs just before task 5
2. If B completes task 20 before A completes task 4, then B will proceed to buffer B6. This time for B, BL = 2 is still greater than CL = 1. Hence, B will continue to task 21. Meanwhile, worker A is ready to drop the part in buffer B2, but as B will now be approaching 21 (past the buffer), they are at the same point in the U-line

(at diagonally opposite tasks). Hence, the CBB protocol will take effect, and A and B will exchange jobs just before task 2.

After both cases, the workers will now be in the positions shown in Figure 17.

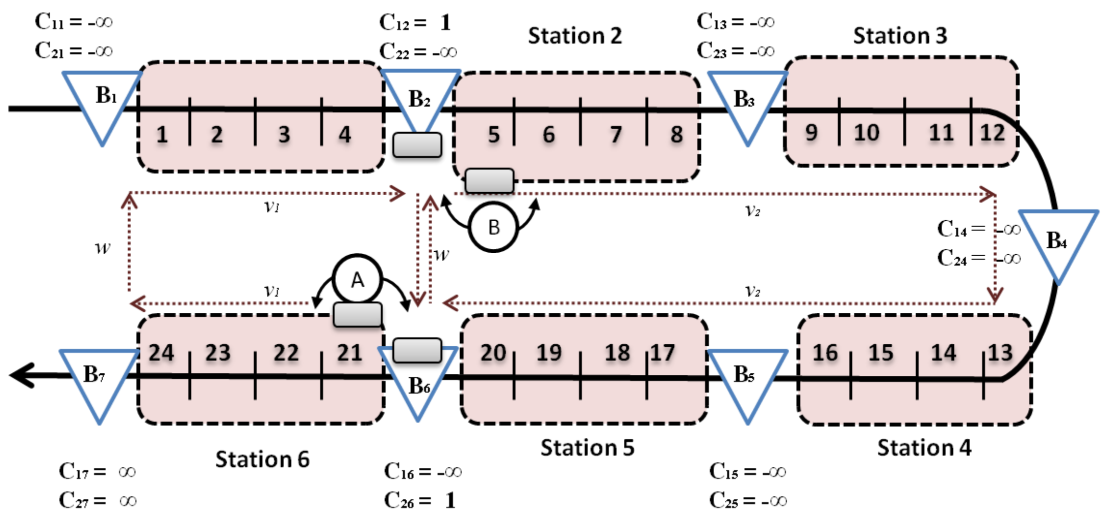


Figure 17: Example Case: When steady state is reached

Scenario 2 - Steady State: Moving forward, if A and B process their jobs deterministically from their current positions, then it can be simulated that they will exchange again at the same point - after tasks 4 and 20. The U-line is said to reach a steady state at this point, as A and B will continue to exchange at the same point in every work-cycle (Figure 17). During a cycle in this state, worker A would process a job through tasks 1,2,3,and 4 sequentially, while during the same time, B would process another job through tasks 13 through 20. At this point, A and B would be at the same point in the line. After completing their tasks, following the CBB rule, they exchange jobs as there is another worker at the diagonally opposite task. Worker A will now start processing the job from tasks 21 through 24, while B will start processing the from task 5 onwards till he encounters buffer 6 the next time, and so on. Worker B works in stations

2, 3, 4 and 5 in his work zone while worker A only works in stations 1 and 6. At this stage, the line is said to be balanced. It is noted that while the workers operate in their respective zones, the job always flows linearly from raw material to finished goods.

Scenario 3: Stochastic processing times: During steady state cycles, when the process times are deterministic, the control levels used in this example will ensure that these workers would work in tandem only in their zones (i.e. the line is balanced). However, if there is variability in process times, at least one of these workers may not reach the end of their zones in time. This would lead to the CBB exchange occurring either before or after the steady state point of exchange at B2 and B6, depending on which worker is lagging. Figure 18 shows the line when the exchange occurs after task 6, when worker B was lagging behind.

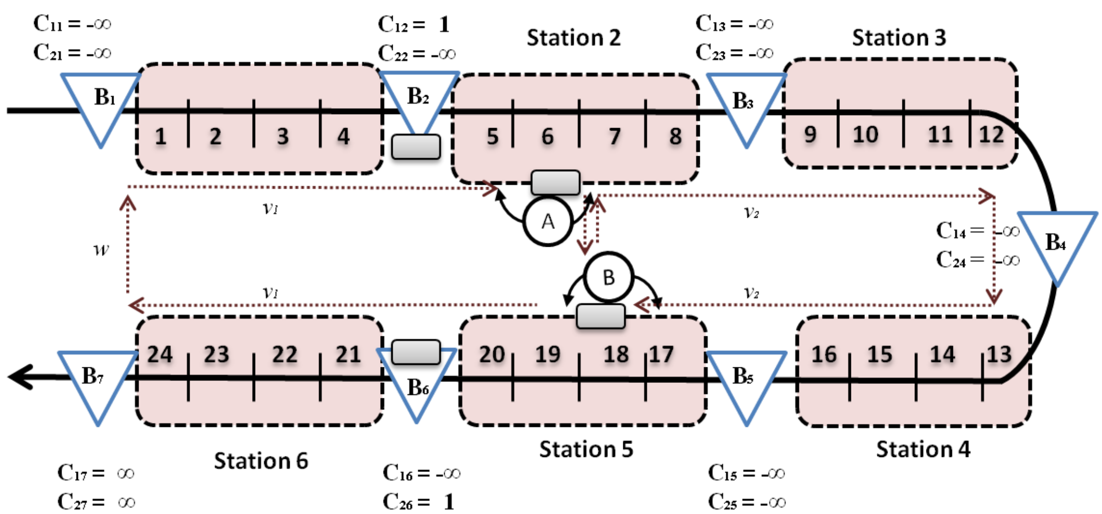


Figure 18: Example case: Exchange at tasks 6 and 18 when worker B lags behind due to variable processing times

Similarly, during the initial stages when WIP is being built-up, the protocol may require the workers to use the buffers to reduce waiting times. For instance, in this case, referring

back to Figure 18, if there was a buffer after task 6, then worker A may not have to wait till worker B completes task 18 before exchanging. This leads to the idea that the protocol can accommodate buffers between each pair of adjacent tasks. But, what should the control levels be? How do we decide where the buffers are to be located? These are questions that arise primarily for a stochastic system, as it is difficult to imagine, comprehend, and extrapolate. These questions are internal to this research work, and will be answered in subsequent sections.

Another observation that arises from this exercise is that the U-line itself seems symmetric i.e., the legs of the U-line are similar. When a worker drops off the job after the final task (task 24) in the finished goods buffer (in this case B7, and generally $B_{j=T+1}$) and moves to task 1 after picking up a new job from B1 on the first leg of the line, it is the same as a worker processing task T/2 (task 12 in this case, at the end of the first leg of the line) and moving to process task T/2 + 1 (task 13 in this case) on the second leg of the line.

5.2.2 Generalized New Protocol

This sub-section is devoted to describing the generalized new protocol. The example case discussed above was just one of many simple simulations used to gain an understanding of how the MWS protocol can be integrated into the CBB protocol. Through an iterative process of simulating the protocols and revisiting the protocols to make changes based on observations, a set of rules for the new protocol emerged. These rules were a combination of MWS and CBB protocols, but were interconnected in certain unique ways. In sections 2.4.2 and 3.2, the MWS and CBB protocols from literature were

presented. Montano et.al (2007), and Lim (2011) listed a set of forward-processing and backward-walking rules that workers follow for their protocols. However, the new protocol will be difficult to comprehend when the rules are listed out in that fashion. Instead, a flowchart approach (Figure 19) is used to describe the protocol. This flowchart consists of the decisions to be made and paths to be taken by worker *i*, in an assembly line with 'T' tasks. In the above section, the idea of expanding the total number of buffer locations from one between consecutive stations to one between consecutive tasks was discussed. In the flowchart, this idea is captured by using buffer notations B*j*, representing the location of the buffer between tasks *j*-1 and *j*. There will be a total of T +1 buffers.

Lighter color-crisscrossed items in the flowchart represent the pieces of rules borrowed from the CBB protocol, and the darker shaded-checked items represent the rules borrowed from the MWS protocol. The two items in white are common to both. The inter-connected nature of the two protocols is visible here.

The flowchart starts from worker *i* completing task *j*. After finishing task *j*, the worker checks task T-*j* (which is diagonally opposite to task *j*) for the presence of another worker. If there is another worker waiting at task $T - j$, worker *i* exchanges jobs (CBB protocol type exchange) with that worker. Referring back to the example case in Figure 18, at the point of exchange, $j = 6$, and $T = 24$; worker A at $j = 6$ will check if there is another worker at task $T - j = 24 - 6 = 18^{th}$ task. Worker B at $j = 18$ would have, in turn, checked at task $T - j = T - 18 = 6^{th}$ task. This symmetric nature of the protocol is to be noted. If there was no other worker at task T-*j* , or if the other worker was still processing the job at task T-*j*, then worker *i* will check if there was another worker

waiting at task $j + 1$ to MWS preempt worker i. If not, worker i will then check the buffer level, similar to the MWS protocol. The CBB protocol takes precedence over the MWS protocol here. The premise behind this precedence is that a MWS preemption causes walking back, whereas a CBB exchange does not, which could increase efficiency and throughput. This premise would be tested at a later stage when the simulation model produces results.

In addition to the flowchart, a tree-chart form of the protocol covering all possible scenarios, based on where the worker is located at that point of time, is provided in Figure 20. These flowcharts were also used as the basic logic around which the simulation model was built, tested, and verified.

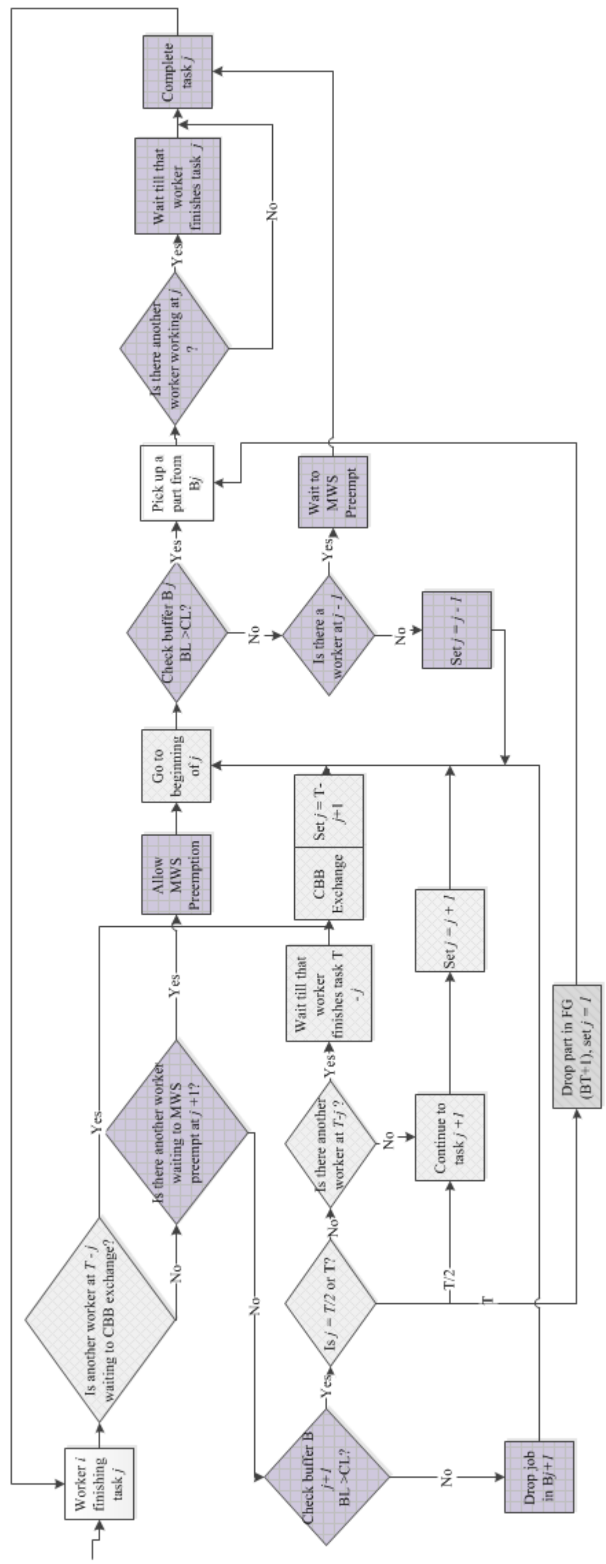


Figure 19: Flowchart showing the new protocol for a two worker U-line system to be followed by worker *i*, starting from task *j*

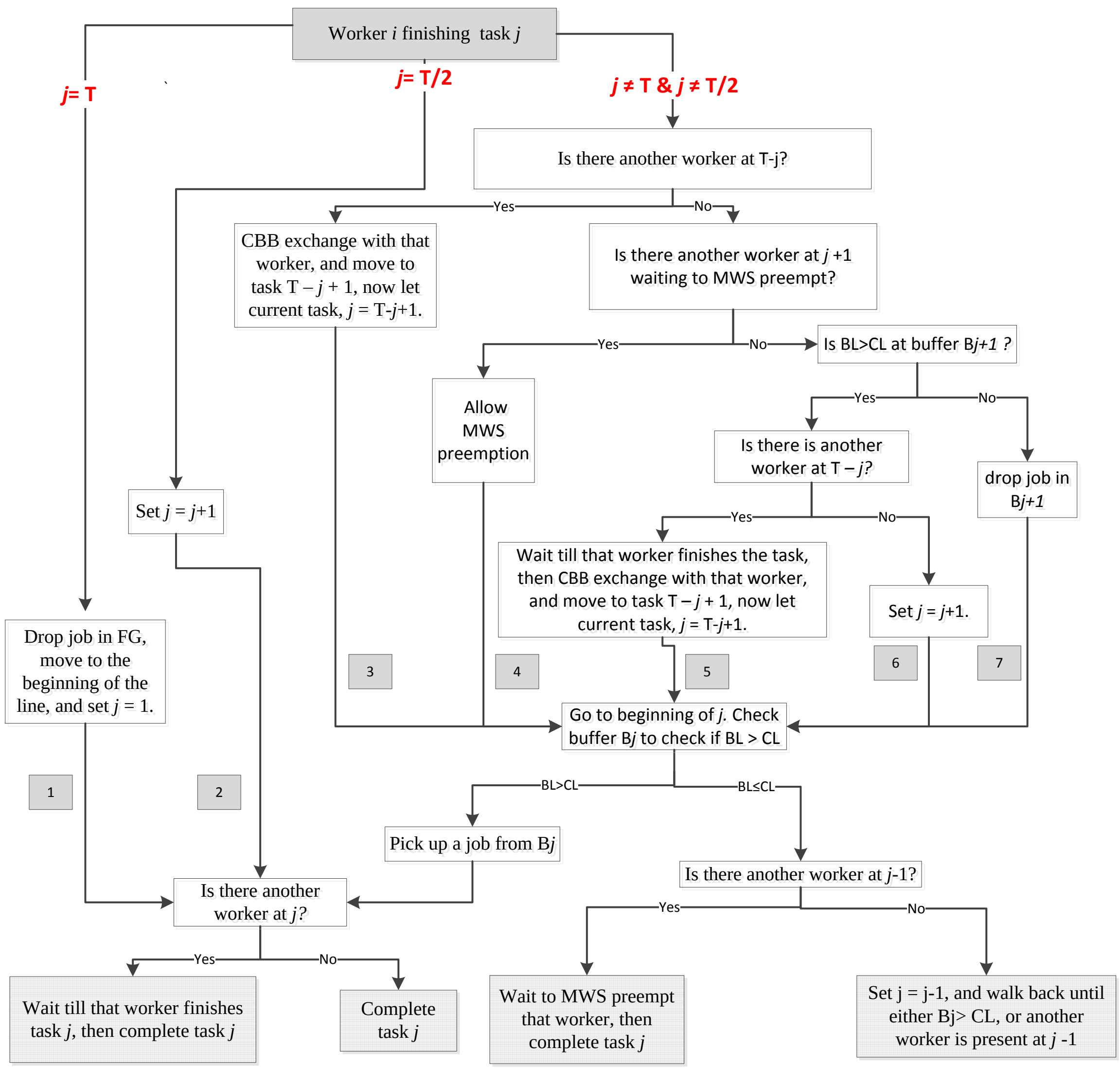


Figure 20: Protocol flowchart starting from the location of the worker. Seven different paths or branches (possible scenarios) of the protocol are shown.

While following this new protocol, a few patterns were observed. When two workers process jobs on opposite legs of the U-line, they do not proceed to the subsequent task on their side of the line, if there is another worker present at the diagonally opposite task. i.e., they never pass each other when they are moving forward processing a job. This requires that they constantly check the diagonally opposite station for another worker before and after a task. Also, CBB exchanges and MWS preemptions happen only before or after a task is completed. When a worker is ready to exchange jobs with another worker or preempt him/her, the worker will have to wait till the other worker completes the task before the exchange/ preemption occurs. Hence, the workers will not only have to look out for these rules, MWS preemptions and CBB exchanges, but also have to adhere to their own control levels in each buffer. These buffers can be located before or after every task. This drastically increases the number of possibilities and rules that the worker has to remember. Taking a step back, the questions that arise at this point are:

- Does the protocol require every buffer at every location, or are some strategic locations better than others? If so, where are these locations, and what are the optimal control levels (C_{ij}) for each worker at every buffer?
- Should these control levels be the same for every worker?
- Will these control levels vary with factors such as worker velocities, aisle width, line length, walking speeds, and task times?

To answer these questions, the approach taken was to use ARENA's built-in optimization tool, OptQuest. Once the model is built in ARENA, optimization models were setup in OptQuest using variables and constraints derived from the simulation model. Once the scope of the model is discussed in section 5.2.3, the modeling approach itself is detailed in 5.2.4.

5.2.3 Thesis Scope

As seen in the previous section, various factors such as varying worker velocities, stochastic task times, aisle widths, length of the line, number of tasks, number of buffers, control levels etc., increases the permutations that are possible, and the scope grows exponentially. To keep the focus on developing the protocol and nuances associated with it, and to keep the size of the problem at a level where it could be addressed effectively, the scope was defined. To help define this scope, a survey of three local manufacturing companies was conducted. These companies provide industrial technology solutions, manufacturing medium to high variability products, to serve the power and medical electronics industries. The survey included a total of thirty two U-lines located in five different facilities in the state of New York. The data collected is presented in Table 1. For worker velocities and worker velocity ratios, actual data was not available, and hence, expert opinion was collected from company representatives. Worker velocities varied greatly - cycle times varied from 2 minutes to 3.5 hours.

Table 1: Data of 32 U-lines compiled from three local manufacturing companies

Characteristic	Range			
	Minimum	Maximum	Average	Median
Number of discrete tasks per line	7	25	15	21
Number of workers per line	2	4	2.7	3
Length (L)	20ft	40ft	2	40ft
Worker velocities (v_i)	*Varied depending on product, training, tools			
Worker velocity ratios (fastest: slowest)	*1:1	*1.5:1	-	*1.2:1
Aisle width (a)	4ft	6ft	5ft	5ft

* expert opinion, actual data not available

From the data presented in Table 1 and survey of existing literature presented in chapter 2, and to effectively answer the research questions that are posed in this work, the range of some of these factors were restricted. The rationale is presented below.

Number of tasks and workers. Montano et.al (2007) worked on 3 and 6 station line with 2 and 3 operators. Their work considered three or four work-elements (or tasks) per station, totaling 18 to 24 tasks per assembly line. Miltenburg (2001) notes from surveying U-lines in 114 companies in the US and Japan that the average U-line has 10.2 machines and 3.4 operators. Considering these evidences from literature, and to probe how the number of tasks affected the effectiveness of the protocol, three task levels of 8, 16 and 24 are selected. Bartholdi & Eisenstein (1996), Bartholdi & Eisenstein (2001), Montano et. al (2007), Miltenburg et al. (2005), have all considered a two worker line before considering three workers. Given the complexity of this new protocol and to limit the size of the experiment, the scope is restricted to two workers. Hence, assembly lines consisting of 8, 16, and 24 tasks, with 2 workers (T = 8, 16, 24, N = 2).

Length of the U-line. The range for this factor is directly derived from industry data presented in Table 1. U-line lengths of 20ft and 40 ft have been considered.

Worker Velocities. Assumption 4 in section 5.1 describes the choice of Gamma distribution and the CV values (between 0.3 and 1). The average task times vary greatly across the industry, and most literature does not deal with specific values for task times. Hence, an average task time of 60 seconds per task was chosen based on the input from one of the companies in the survey, consisting of 30 U-lines.

Worker Velocity Ratios. This is directly derived from industry data. Three ratios are to check if the velocity ratios have a linear effect on the performance of the protocol. Thus, average velocity ratios of 1: 1, 1:1.25, 1:1.5, were chosen.

Aisle width ratios. Lim (2011) states that at a = 0.04 (or 4% of the length of the assembly line), CBB outperforms the traditional BB for serial lines. 10% aisle width is a predominant configuration in U-shaped cells, from Table 1. To include both the industry data and data from literature, aisle width levels of 4% and 10% were considered.

Control Levels. Montano et al. (2007) proposes using four different control buffer levels: -∞, 0, ∞ and *b;* where *b* is a fixed value between 0 and ∞. The authors use *b* = 3 (which is the maximum number of workers that they experiment with), to keep the complexity of the resulting system at a manageable level. This thesis work will expand this assumption to consider *b* = 1, 2 or 3.

Based on the discussion presented above, the scope of the thesis model is presented in Table 2.

Table 2: Scope of the thesis

Factors	Units	Levels
No.of tasks (T)	Count	8,16, and 24
No.of workers (N)	Count	2
Lengths of the line(L)	ft	20, and 40
Worker velocities (v_i)	seconds	mean 60s, CV values between 0.3 and 1
Worker velocity raitos	Ratio	1:1, 1:1.25, and 1:1.5
Aisle widths (a)	%	4, and 10
Control Levels (C_{ij})	Count	0, 1, 2, and 3

5.2.4 Modeling the new protocol in ARENA

The approach to modeling the protocol in ARENA was to model each task as a block of logic, such that every task contained all the elements of the protocol. Each task was modeled as a Submodel in ARENA. Workers and jobs were modeled as "entities". The logic of entities entering and exiting each Submodel was handled through "station" and "route" modules. The processing time within each task was modeled using the "process" basic module.

Three different models were built – one for 8 tasks, one for 16 tasks, and one for 24 tasks. In Appendix II, screenshots of all the models in ARENA are shown. The workers were modeled as entities so that attributes such as worker velocity and worker number could be assigned to them; and it was convenient to collect statistics of entities. Each task consisted of a number of decision modules, process modules, pickup, delay blocks, drop-off, and assign modules, that came together to model the logic of the protocol (Figure 21). To consolidate all the tasks into one screen, each task was modeled into a sub-model. The logic described in Figure 19 and Figure 20 was used to build each task. Appendix V shows a screenshot of the first task submodel. Figure 24 is a snapshot of a 16 task line, each sub-model represents a task.

All stations were modeled the same, except for the first task, T/2 task, (T/2 + 1) task, and the last T task. These tasks that are located at the ends of each leg of the U-line, will be referred to as terminal tasks or terminal submodels henceforth. The create module used to create the workers fed into the first and (T/2 + 1) task.

Jobs are created only at the beginning (before the first task), except in the beginning - when the second worker is created at task (T/2 + 1), one job is created at that time for that worker. Tasks T/2 and T, will not have workers performing CBB exchanges from them. Each non-terminal submodel was connected from two station modules, and lead out to five route modules. While

workers and jobs were allowed to enter one station (named Station "X”), the second station was only for jobs that were dropped off at buffers (named Station "X" buffer). The five routes are used to execute entities in that come out from each task in different conditions.

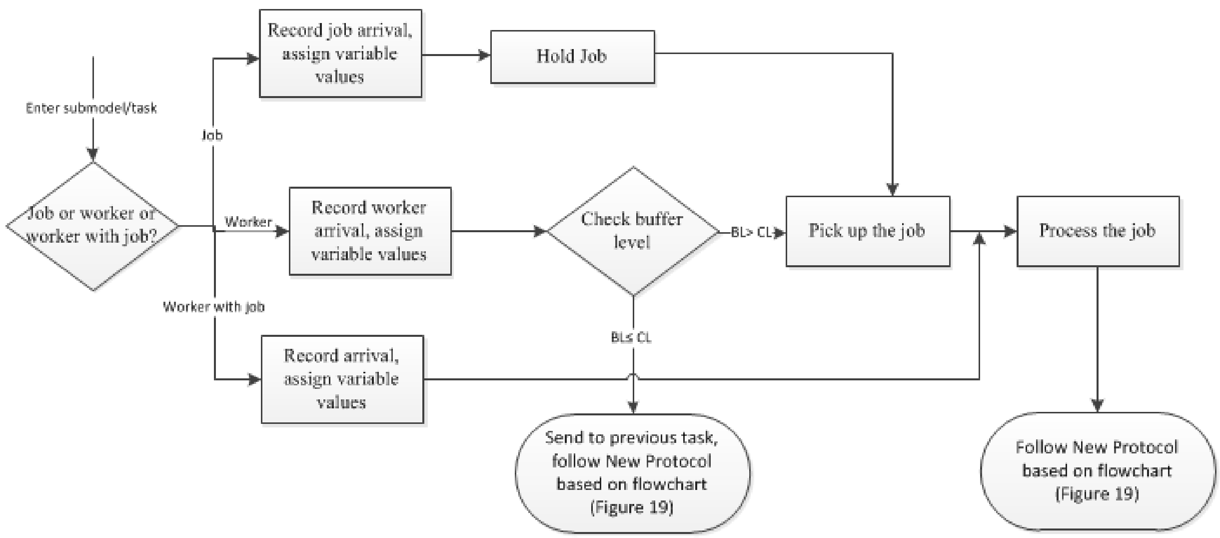


Figure 21: Flowchart showing logical flow of entities inside a task submodel

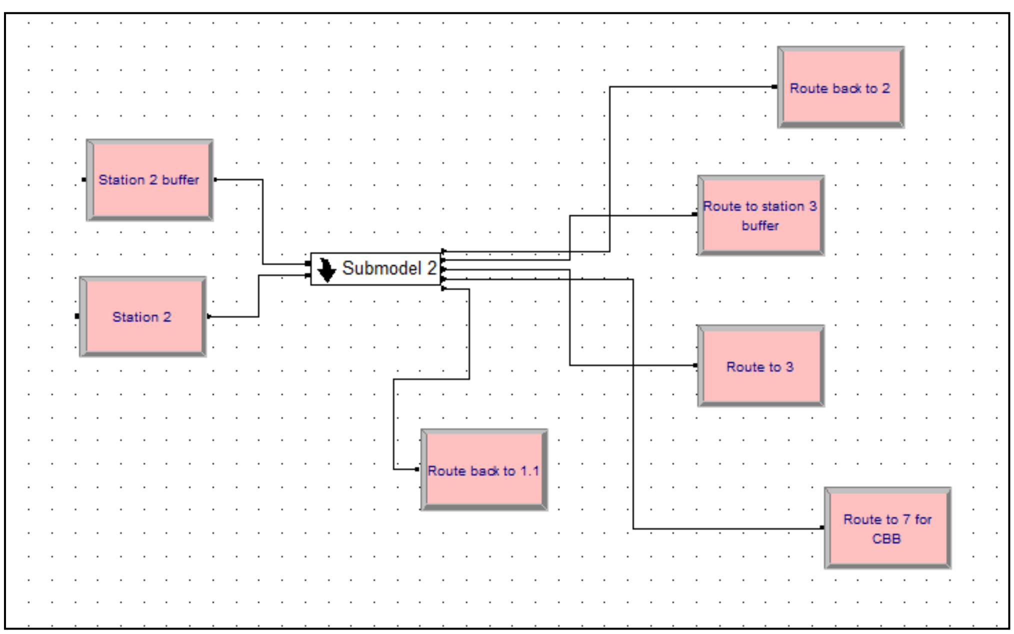


Figure 22: Submodel for the task 2 in the 8 tasks model, showing station and route modules

Figure 22 shows them for submodel and task 2. They include routing the worker back to the beginning of the station, routing the worker to the next station, routing the worker to station T – $j + 1$ (the station to which the worker would go if a CBB exchange occurred at station j), and routing back to the previous station. The terminal stations will have only four outputs, but may have an additional input where the workers or jobs are fed into the system.

Variable - Basic Process

	Name	Rows	Columns	Data Type	Clear Option	File Name	Initial Values	Report Statistics
1	Watstation	24		Real	System		0 rows	
2	Velocity variable	2		Real	System		0 rows	
3	Control Level	2	25	Real	System		49 rows	
4	walk back	2		Real	System		0 rows	
5	workers at task	24		Real	System		0 rows	
6	MWS trigger	24		Real	System		0 rows	
7	T			Real	System		1 rows	
8	CBB trigger	24		Real	System		0 rows	
9	L			Real	System		1 rows	
10	aisle width			Real	System		1 rows	
11	No.of times CBB Holds	24	2	Real	System		0 rows	
12	CBB Waiting Times	24	2	Real	System		0 rows	
13	CBB Waiting Times Timein	24		Real	System		0 rows	
14	Blocking Times Timein	24		Real	System		0 rows	
15	Blocking Times	24	2	Real	System		0 rows	
16	Processing Times Timein	24		Real	System		0 rows	
17	Processing Times	24	2	Real	System		0 rows	
18	MWS Waiting Times Timein	24		Real	System		0 rows	
19	MWS Waiting Times	24	2	Real	System		0 rows	
20	Throughput			Real	System		0 rows	
21	SIGNAL FOR CREATE			Real	System		0 rows	
22	Total Number of CBBs			Real	System		0 rows	
23	Total WIP			Real	System		0 rows	

Figure 23: Snapshot of the variables created in the model.

Many variables were used to facilitate the logic of the protocol. Figure 23 provides a snapshot of those variables, and Table 3 provides descriptions to the important (non-data collecting) ones. Although an attribute, Control Levels are entered as variables, so that they can be used as variables for the optimization model in OptQuest. Variables "Watstation", and "workers at task" were used to track the count and position of workers in the system. "MWS Trigger" and "CBB

Trigger" variables were used to indicate to the workers if they need to exchange or preempt another worker. Variable "walk back" was used to track of the worker(s) who were walking back in the U-line.

Table 3: Description of important variables used in the simulation model

Variable	Notation used	Description	No. of Rows	No. of Columns
Watstation	-	stores number of workers present in the task module	T	-
velocity variable	v_i	Velocity of worker *i*	2	-
Control Level	C_{ij}	Control levels for worker *i* buffer *j*	2	T+1
walk back	-	1 if worker *i* is walking back, 0 otherwise	2	-
workers at task	-	1 if worker *i* is present in the j^{th} task module, 0 otherwise	2	T
MWS Trigger	-	1 when worker is starting to wait to MWS preempt the other worker, 0 otherwise	T	-
T	T	Total number of tasks	-	-
CBB Trigger		1 when worker is starting to wait to CBB exchange with the other worker at T - *j*, 0 otherwise	T	-
L	L	Length of the line	-	-
aisle width	a	Aisle width ratio	-	-

“ReadWrite” advanced process modules were used to collect data during simulation runs, and “File” advanced process tab was used to record the data collected into excel sheets. The data that was collected from the simulation runs included throughput, average cycle time, total number of MWS preemptions at each task, total number of CBB exchanges at each diagonally

opposite pairs of tasks, processing time for each worker, waiting time for each worker, and blocking time for each worker. These outputs from each run were stored in excel files and can be accessed at a later time. An important point to be noted is that, because the new protocol combines both the CBB and MWS systems, either of those features could be turned "off" or "on". That is, the model behaves as a simple CBB system when all the control levels are made zero, and the model behaves as a simple MWS system in a linear assembly line when the CBB exchanges are turned "off".

In the previous section, there were questions raised about how to find optimal control values and buffer locations. Once the simulation model was developed, ARENA's inbuilt optimization tool called OptQuest could be used. In OptQuest, the user can select variables from the list of variables in the simulation model, specify limits to them, create constraint equations, and provide an objective function. The optimization problem used to find the optimal control values can be defined as follows:

Variables:

C_{ij} (Integer) = Control level for worker *i* (total N workers) in buffer *j* (total T+1 buffers)

Th = Throughput: the number of finished parts produced in the U-line

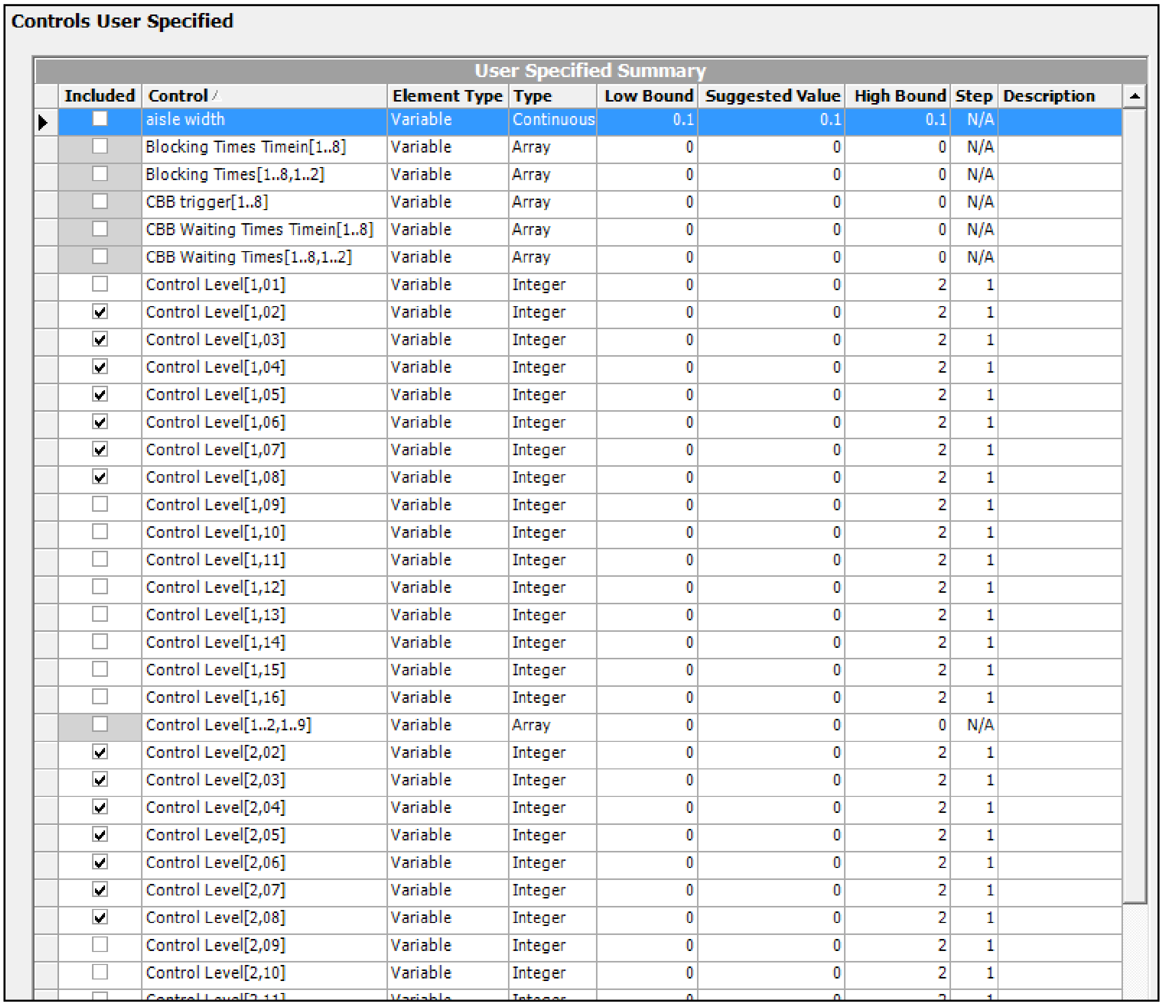
Objective:

Maximize (Th)

Subject to constraints:

$0 \le C_{ij} \le 2 \qquad \forall\, i \in [1,2]$ and $\forall\, j \in [2, T]$

The objective was to Maximize Throughput. The constraints used in this case were the limitations on the values of the control level variables (0,1, and 2). This constraint was applied to every worker-control buffer combination in the U-line, excluding the first buffer (buffer 1, which is the raw material stock) and the last buffer (buffer T+1, which is the finished goods inventory).


Controls User Specified

User Specified Summary

Included	Control	Element Type	Type	Low Bound	Suggested Value	High Bound	Step	Description
☐	aisle width	Variable	Continuous	0.1	0.1	0.1	N/A	
☐	Blocking Times Timein[1..8]	Variable	Array	0	0	0	N/A	
☐	Blocking Times[1..8,1..2]	Variable	Array	0	0	0	N/A	
☐	CBB trigger[1..8]	Variable	Array	0	0	0	N/A	
☐	CBB Waiting Times Timein[1..8]	Variable	Array	0	0	0	N/A	
☐	CBB Waiting Times[1..8,1..2]	Variable	Array	0	0	0	N/A	
☐	Control Level[1,01]	Variable	Integer	0	0	2	1	
☑	Control Level[1,02]	Variable	Integer	0	0	2	1	
☑	Control Level[1,03]	Variable	Integer	0	0	2	1	
☑	Control Level[1,04]	Variable	Integer	0	0	2	1	
☑	Control Level[1,05]	Variable	Integer	0	0	2	1	
☑	Control Level[1,06]	Variable	Integer	0	0	2	1	
☑	Control Level[1,07]	Variable	Integer	0	0	2	1	
☑	Control Level[1,08]	Variable	Integer	0	0	2	1	
☐	Control Level[1,09]	Variable	Integer	0	0	2	1	
☐	Control Level[1,10]	Variable	Integer	0	0	2	1	
☐	Control Level[1,11]	Variable	Integer	0	0	2	1	
☐	Control Level[1,12]	Variable	Integer	0	0	2	1	
☐	Control Level[1,13]	Variable	Integer	0	0	2	1	
☐	Control Level[1,14]	Variable	Integer	0	0	2	1	
☐	Control Level[1,15]	Variable	Integer	0	0	2	1	
☐	Control Level[1,16]	Variable	Integer	0	0	2	1	
☐	Control Level[1..2,1..9]	Variable	Array	0	0	0	N/A	
☑	Control Level[2,02]	Variable	Integer	0	0	2	1	
☑	Control Level[2,03]	Variable	Integer	0	0	2	1	
☑	Control Level[2,04]	Variable	Integer	0	0	2	1	
☑	Control Level[2,05]	Variable	Integer	0	0	2	1	
☑	Control Level[2,06]	Variable	Integer	0	0	2	1	
☑	Control Level[2,07]	Variable	Integer	0	0	2	1	
☑	Control Level[2,08]	Variable	Integer	0	0	2	1	
☐	Control Level[2,09]	Variable	Integer	0	0	2	1	
☐	Control Level[2,10]	Variable	Integer	0	0	2	1	

Figure 24: Screenshot of the controls screen in OptQuest. The selected variables are included in the optimization model for a 8 task system. The bounds for these controls can be seen here.

A screenshot of the selection window for choosing boundaries in OptQuest for the 8 task model is shown in Figure 24. Once these constraints are entered, the final screen in OptQuest (Figure

25) can be used to set number of replications, tolerances, and automatic stop options. When the "Optimize" tab is clicked, OptQuest starts running the simulations in batch mode. The program will automatically stop when it finds the optimal solution, or multiple optimal solutions. These solutions were the optimal control levels at which the system achieved maximum throughput. The results were saved in an excel file for future review. This optimization model was run for each treatment level combination in the main experiments. Results and analysis of these experiments are detailed in sections 6.1 and 6.2.

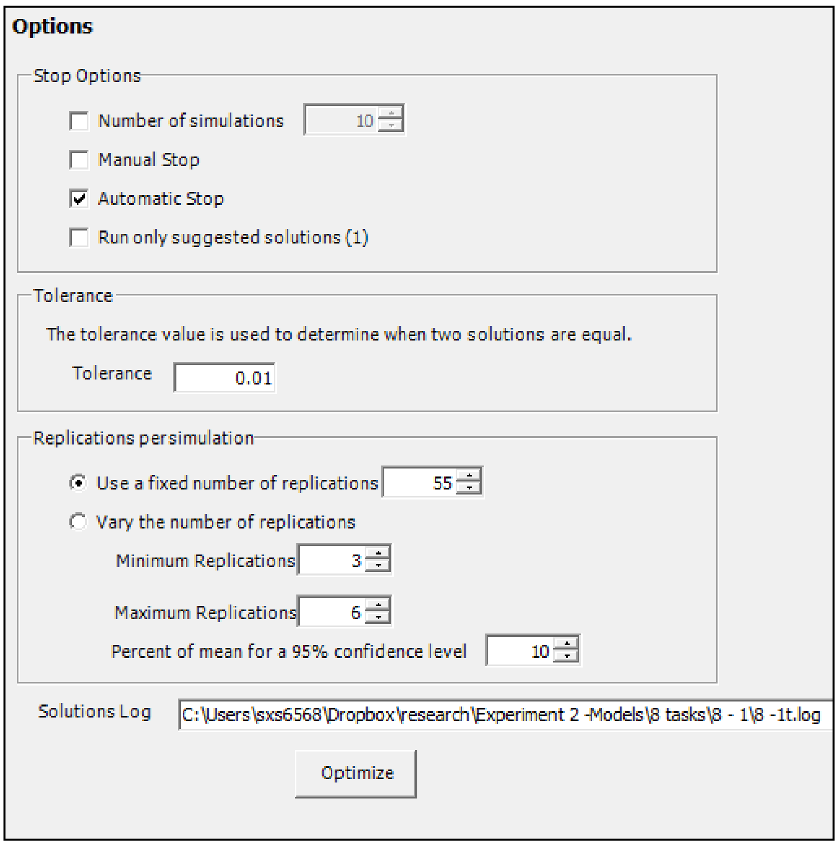


Figure 25: Final settings screen in OptQuest

The time taken for each simulation depends on the size of the model, run time, and number of replications. When the model was initially run, the simple 8 task model took more than 2 minutes to run 55 replications. Since OptQuest ran around 300 runs for these models, that was

around 600 minutes, or 10 hours per model, per treatment combination. To reduce the run time of the model, the number of entities in the model were reduced. This was done in cases where there was potential to "split" the entity into two entities, allowing one of those entities to proceed in the system, and the other to loop back into the system after a delay period. This delay period mimicked the time before which another entity was created.

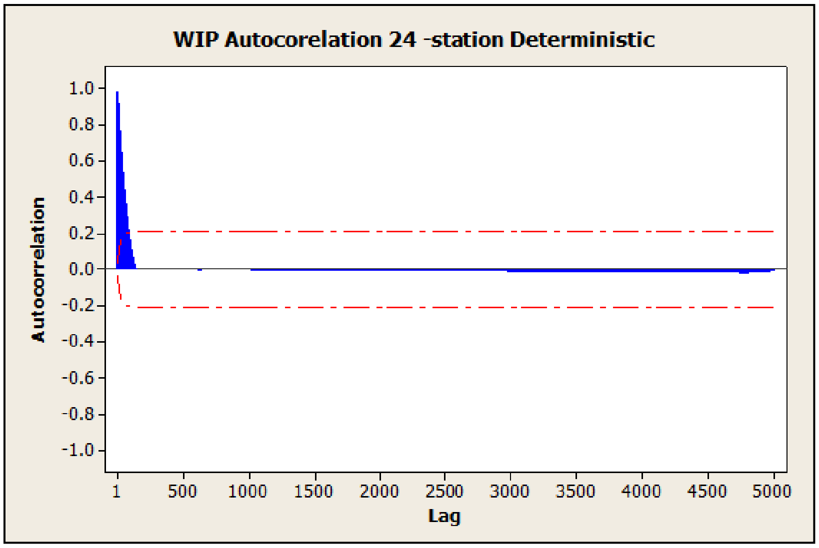
Run Time Calculations. Since the model is a non-terminating system (assembly lines are generally considered non-terminating), the simulation needs to run for a certain amount of time , called warm-up time, before it can reach steady state. To calculate the warm-up time required for this model, the model that is expected to take the longest time to reach steady state (in this case, 24 tasks model) is chosen. The system is said to reach steady state when the system has a constant WIP over a period of time. To identify the warm-up period, the 24 tasks line was run for 5000 min (an arbitrarily large period of time). The cycle time in the deterministic case, can be calculated as (24tasks X 1min/task) / 2 workers = 12 minutes. The WIP data was set-up to be collected every 2 minutes, about five times every cycle. As the WIP level varied with different control levels and deterministic and stochastic processing times, two extreme cases were chosen. For these two cases, the autocorrelation data was plotted in Minitab statistical software; warm-up times and the corresponding run times were calculated for different cases:

1. Deterministic task times, with control levels set at 0

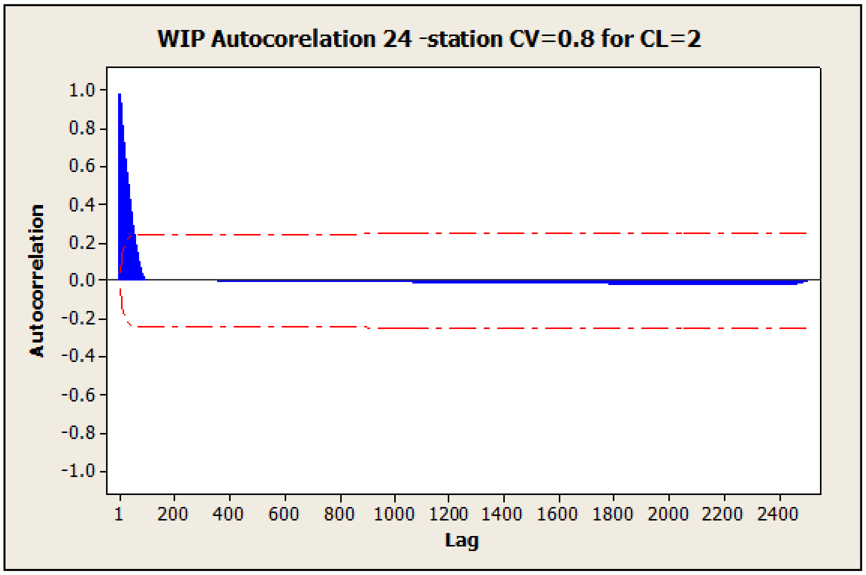
   Lag from the graph below = 70, Each lag = 2 min

   Warm-up time = 70 X 2 = 140min

   Corresponding run time $\geq$ 6 X warm-up period = 840 min



2. Stochastic processing time - Gamma distribution with α = 1.5625, 1/β = 38.4



Lag from above graph = 87, Each lag = 2 min

Warm-up time = 87 X 2 = 174min

Corresponding run time ≥ 6 X warm-up period ≥ 1044 min

Hence, a run time of 1200 min was used to run the simulation model

5.2.5 Physical Experiment to validate model

Once the simulation model was verified with all the scenarios presented in Table 20, it could be said with confidence that the model included all the factors and performed as intended. However, to test the model against a real-world environment to ensure that all the factors that affect the protocol were considered, the working model had to be compared to a physical assembly line. The facility and resources for this physical assembly line were available at the Toyota Production Systems Lab at RIT. Hence, a physical experiment was setup in the Toyota Lab to test the validity of the model.

Figure 26: The experimental setup in the Toyota Lab at RIT with two volunteers assembling Legos on either legs of the assembly line

A U-shaped assembly line was setup using roller conveyor lines (Figure 26). Flow racks on one side and tool stands on the other side were used to supply parts into the assembly line. Simple Lego blocks were chosen as parts that would be assembled together to form a Lego "building" (Figure 27).

Figure 27: Completed Lego assembly block at the finished goods location, on a plastic pallet used as a carrier.

Volunteers were called for to perform the building tasks and help with time studies.

Data collected from this experiment were used to calculate output variables such as number of preemptions and exchanges, throughput, and time when these preemptions and exchanges occurred. The values of factors such as worker velocities, aisle width, length of the line, worker walk velocities, and waiting times were physically recorded and measured for this experiment were fed into the simulation model to obtain values of outputs such as throughput and utilization. These output values were then compared with the output obtained experimentally to ascertain the validity of the simulation model.

A few industrial options were available for the assembly process, but Legos were chosen because of their simplicity. Using simple Lego blocks as assembly material provided a number of distinct advantages over industrial assemblies. They provided the flexibility to allocate work to each task and hence it was possible to re-balance the assembly line easily. More importantly, they demanded a very small learning curve (if at all any) from the volunteer-workers. This was important, because the focus of this experiment was to have volunteers execute the protocol, and any time saved in training them to assemble would be useful. Also, using Legos ensured that the focus of volunteer-workers would be on executing the protocol rather than the assembly process itself. This ensured that the assumption of having the same average work content for all

tasks would more or less hold true. Legos were also easy to pull apart once put together. This was necessary as there was a limited supply of assemblies.

This was an activity that involved no greater than minimum risk. An application to conduct this experiment on human subject volunteers sent to the RIT Institutional Review Board was approved (approval form attached in APPENDIX I). Student volunteers were sought for and a total of 11 students volunteered for this experiment.

Each volunteer was briefed on the intention and contents of the research work, and the protocols were explained to them. Volunteers who would represent workers were then allowed to assemble the Legos by themselves. They were asked to perform each task in sequence – starting from placing each Lego block one by one on the board in task 1 (4 blocks), then dragging the pallet with the base plate and Lego assembly to the next task, assembling 4 blocks at task 2, and so on around the U-line. They were asked to perform this exercise twice, so that they learnt the process. Timing each volunteer-worker at each task provided worker velocities for each worker. Two of these volunteer-workers, whose velocities were closest together, were then chosen. First, they were trained on executing simple CBB system. After three runs (of 15 minutes each), they gained confidence in the protocol. Then, buffers were introduced in the system, and the new protocol was explained. It was noted that the time taken to learn the new protocol was significant. Three trial runs of 15 minutes each consisted of multiple stoppages, and lessons were learnt. It was only in the final 30 minutes of this entire exercise that usable data could be obtained. This part of the experiment was recorded.

Given the size of the available conveyors (approximately 50 ft), material, and space, the values of factors that were set-up are listed in table 3. The space on the line between each pair of

stations represented buffers. The 40 ft line was divided into 24 equally spaced tasks. Each task consisted of assembling 4 Lego blocks. This system worked better, and 2 trial runs were conducted before executing the protocol. However, the volunteers who assembled this time were different from the ones who assembled the previous time. This again caused confusion, resulting in multiple stoppages, and skewed data.

Table 4: Values of factors (measured and calculated) that were used for the validation experiment

Factors	Measured / Calculated Values
No.of workers (N)	2
No.of tasks (T)	16
Length of the line (L)	40 ft
Aisle width	10% (4 ft)

The experiment was conducted in one - two hour blocks over a period of three weeks to accommodate volunteers' schedules. A total of 12 hours were spent for this experiment. A significant amount of time was spent in training, re-arranging, and repeating experiments. Initially, the 40 ft line was divided into 24 tasks, each task consisting of putting together 4 blocks of Lego. After four trials of 15 minutes each, there were multiple inconsistencies with the data collected. The data was skewed due to stoppages in the experiment.

The Lego blocks were of short supply and the line had to be stopped many times to replenish the parts, and the volunteers were often confused as to when they should exchange the job as the tasks were placed too close to each other. Hence, the tasks were re-distributed along the 40 ft line. The line now consisted of 8 tasks on each line. Each task was to assemble four blocks of Legos onto the board. Visual signs were explicitly put up to indicate which task they were at j, and which task to check for the CBB exchange ($T – j$). Figure 28 shows a schematic

representation of the physical validation experiment. Figure 29 is a picture of two volunteers executing the protocol.

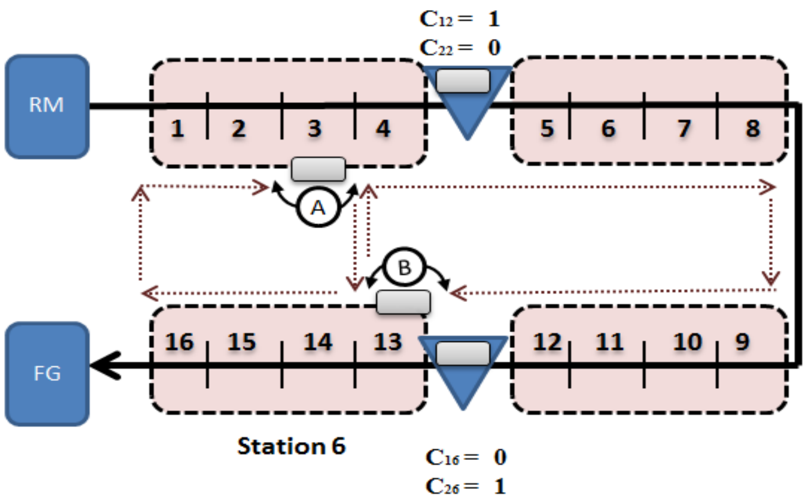


Figure 28: Schematic of the experiment, showing one exchange position

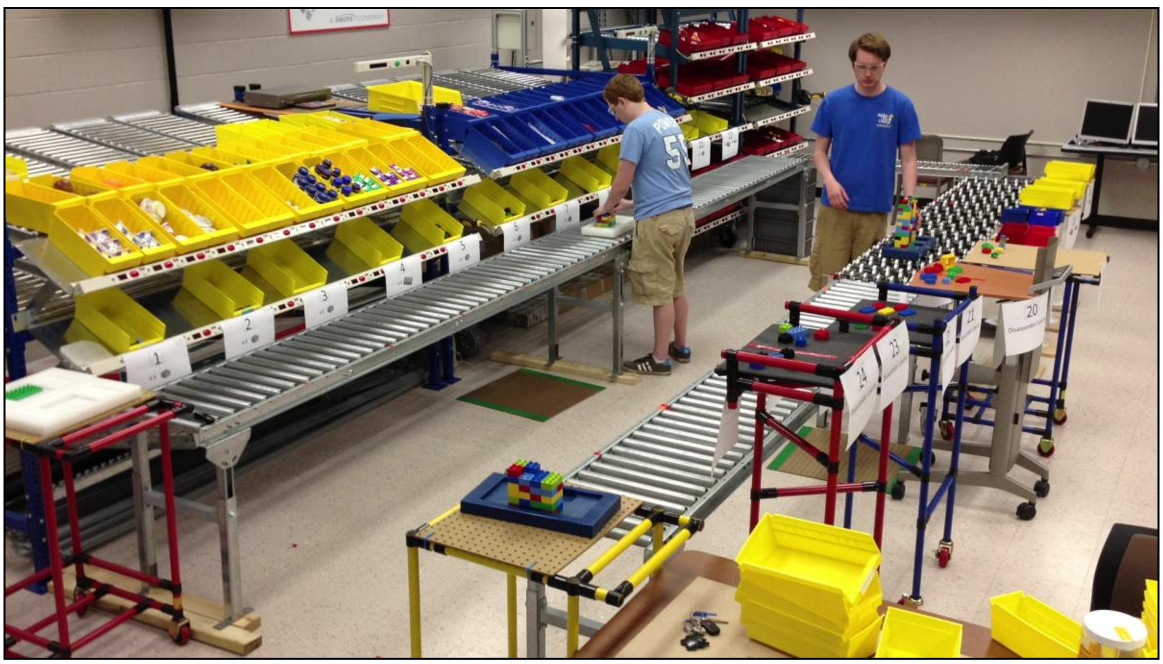


Figure 29: Volunteer in the second leg of the assembly line waiting to perform CBB exchange after task 16

Three 12 minute runs were conducted during the course of the final leg of the experiment. The data collected from the first two runs was impaired by obstructions due to disassembly, or the

workers making mistakes in executing the protocol. The third run produced better data, and hence, this data was used to validate the simulation model.

Inputs from the experiment that went into the simulation model:

1. Worker A and B have work velocity distributions of NORM(8.51, 0.7), and NORM(8.37, 0.52) respectively. These were derived by measuring the time that each worker took to complete specific individual tasks, and then fitting this data to a distribution in Minitab. Considering the fact that workers may not perform every task in every cycle, task 3 and task 10 were chosen for this purpose to collect more data from the run.

2. Walking across from task 8 to 9 while dragging the pallet along increases task times for task 8. Hence, the time taken for this was noted during the experiment, and its average value was 2.6 s. It also took the workers an extra 2 s (similarly observed average value) to drop the part off after task 16 onto the finished goods table and walk to task 1. These times were also used as input for the simulation model.

3. The initial positions of the workers were tasks 1 and 9.

4. Due to a shortage of parts, and as the finished part had to be disassembled and supplied back to the line, the control levels used for the experiment were limited to 1. The control levels that were used for the experiment, that were also used for the simulation model were: $C_{1,5} = 1$, and $C_{2,13} = 1$. All the other buffer levels were zero.

Table 4 contains data to show that the simulation model is validated by the physical experiment conducted at the Toyota Lab. From this data, it can be observed that the time at which

exchanges occur, time between exchanges, and throughput for the experiment and the simulation model are very similar.

	Experiment			Simulation (including specific walking time data)		
Cycle	CBB Exchange Positions (tasks)	Time (min)	Time since last exchange (min)	CBB Exchange Positions (tasks)	Simulation Time	Time since last exchange (min)
	MWS exchange at the end of 4	0:36	-	MWS exchange at the end of 4	0:33	-
1	3,13	1:47	1:11	3,13	1:47	1:14
2	5,11	3:04	1:17	5,11	2:59	1:12
3	2,14*	4:17	1:13	3,13	4:10	1:11
4	5,11	5:35	1:18	5,11	5:21	1:11
5	3,13	6:49	1:14	3,13	6:32	1:11
6	5,11	8:05	1:16	5,11	7:43	1:11
7	2,14**	9:18	1:13	3,13	8:54	1:11
8	5,11	10:31	1:13	5,11	10:04	1:10
9	3,13	11:44	1:13	3,13	11:17	1:13
Output	9			10****		

Table 4: Experimental data and data from ARENA simulation

However, there were some deviations found from the Simulation model. Reasons for deviation from the Simulation model (refer above table) are:

- At 4:08, A fumbled at task 2 with a lego block, which delayed task 2. This caused B to proceed to task 14, and hence, A had to wait after task 2 to complete a CBB exchange with B

- At 8:46, A fumbled at task 16 with a lego block, which slightly delayed that task. This delay resulted in B once again proceeding to 14, as A was still at 2 when B completed 13. Hence, the CBB exchange happened at (2,14) instead of the usual(3,13).

- There is a difference of 28s that can be noted between the times at which the final exchange (cycle 9) took place. This can be attributed to the following reasons: 1. Tasks 4 and 12 take slightly longer than other tasks, as there is a buffer after them. 2. There

were a few instances when parts where dropped, or there were stumbles by the workers and other human errors. However, as similar incidents occurred almost every cycle and this 28 seconds is a result of a few seconds being added every cycle (as can be noted from the table), it is reasonable to assume that this time difference can be neglected in the larger scheme of validating the model.

- One extra part was produced in the simulation model, as there were enough seconds available after the last exchange for worker B to finish task 16 from task 13.

By comparing the data, and understanding the anomalies, it can be said that the model was validated. Figure 30 is a screenshot from ARENA showing the MWS preemption taking place after task 4. The simulation model was animated to visually show what was going on.

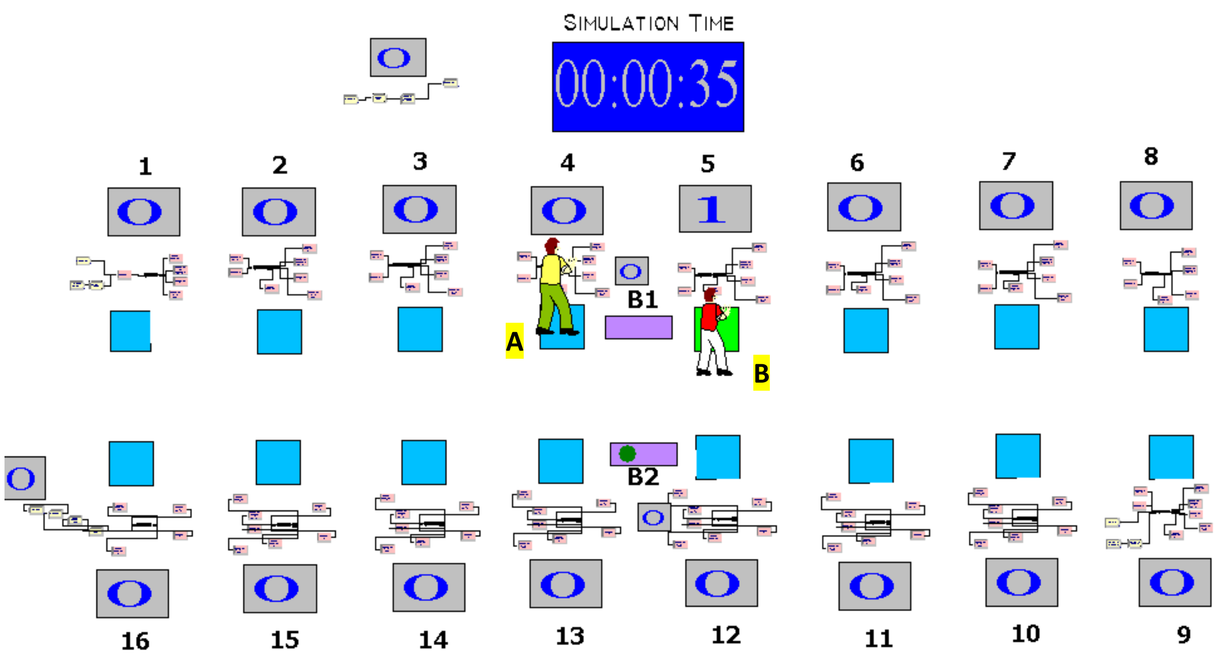


Figure 30: After A drops the part in buffer 1 and B drops the part in buffer 2 and B MWS preempts A at buffer 1.

Figure 31 is a screenshot from ARENA showing worker B crossing task 12 after checking the buffer there. When B approached the buffer in this case, the buffer already contained a job in it. Thus his/her control level was met, and he proceeded to the next station.

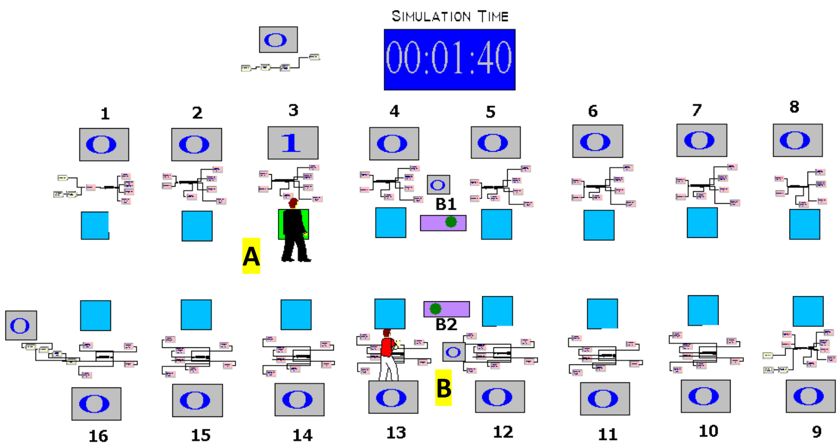


Figure 31: When B crosses the buffer

Figure 32 is a screenshot from ARENA showing workers A and B after exchanging at 5,11 position. A is now at task 12, and B is at task 6.

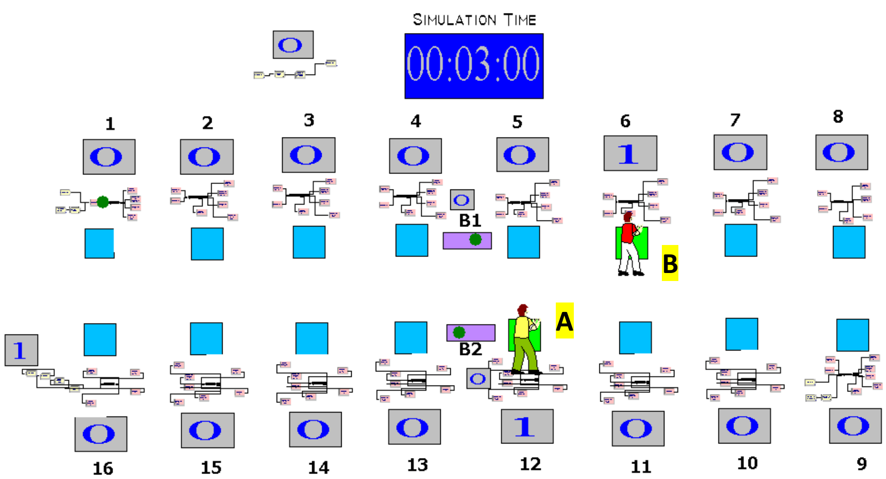


Figure 32: After CBB exchange at 5, 11

5.3 Stage 2

Once the simulation model was developed and validated, a series of larger experiments were conducted to compare the performance of the new protocol with the traditionally balanced U-line and the Cellular Bucket brigade system in terms of some performance metrics. Important performance metrics that can be used to compare and analyze the effectiveness of the new protocols are: Throughput (TH, average output of a production process per unit time), Number and Position of preemptions and exchanges, and Worker Efficiencies. In literature, throughput, cycle time and WIP have been considered as the prime factors to determine which protocol works better. High throughput, low cycle time and low WIP are preferred (Hopp & Spearman, 2008). But, it is known that WIP and cycle time are going to vary as the control levels and number of tasks are varied. Hence, the focus was on throughput, efficiencies, and number of exchanges. Other metrics that will be tracked include processing times, waiting times, walking times, CBB waiting times, MWS waiting times, and cycle times.

Table 2 shows that there can be a total of 6 factors with 2, 3, or 4 levels each. Hence, attempting to run one experiment in which all these factors and levels are tested at the same time was inefficient. First, a screening experiment was setup with two levels each of CVs for Gamma distribution, length, aisle width, and velocity ratio. When some of these factors did not show a significant effect, they were discarded, and similarly, if a factor showed promising results, a center point was added in the subsequent experiment.

6. Results, Analyses and Discussion

This section details the analysis of results obtained during the experimentation on the three simulation models to test the effect of various factors on the new protocol, and provides a comparison of the new protocol against the CBB protocol.

As mentioned earlier, the role of OptQuest was critical to these experiments. For each model and for every treatment level combination, variables (control levels C_{ij}, for every *i* and *j*), constraints (C_{ij} = 0,1,2 for every *i* and *j*; fixed treatment level combinations), and an objective function (Maximize Throughput) were provided to OptQuest. OptQuest would then run 50 replications of one set of Cij values by following a branch and bound method, to obtain an average throughput of these 50 replications, and move on to the next set of Cij values until an optimal set of Cij values are reached for which the average throughput is maximum. During the course of this work, the number of iterations taken by OptQuest to reach an optimal solution varied between 121 and 387. Each iteration took between eight seconds and one minute, when performed on an 8GB RAM windows desktop computer with a 3GHz processor. The average throughput of 50 replications obtained using these optimal control levels will be referred to as "New Protocol Th", henceforth in this thesis.

6.1 Screening Experiments

A screening experiment was conducted to potentially reduce the number of factors or levels. The various treatment combinations considered in this experiment are listed in Table 5 and Table 6. For the screening experiment, extreme levels of factors, discussed in section 5.1 and summarized in Table 2, were considered.

Table 5: Screening Experiment Levels

	Factors	Levels		No. of Levels
		-1	***1***	
A	Gamma distribution CV (constant mean 60s)	0.3	1	2
B	Length	20	40	2
C	Aisle width	0.04	0.10	2
D	$v_1:v_2$	1:1	1:1.5	2
	Total Number of experimental combinations			**16**

Table 6: Full Factorial treatment combinations of the screening experiment

	A	B	C	D
1	-1	-1	-1	-1
2	1	-1	-1	-1
3	-1	1	-1	-1
4	1	1	-1	-1
5	-1	-1	1	-1
6	1	-1	1	-1
7	-1	1	1	-1
8	1	1	1	-1
9	-1	-1	-1	1
10	1	-1	-1	1
11	-1	1	-1	1
12	1	1	-1	1
13	-1	-1	1	1
14	1	-1	1	1
15	-1	1	1	1
16	1	1	1	1

For Gamma distribution of task times, a low variability CV of 0.3 was contrasted with a high CV of 1. Both the levels of length and aisle width were included. In the case of velocity ratios as well, the extreme cases of 1.5:1 and 1:1 were chosen. The rationale behind choosing extremes is that if a certain factor affects a process, it would show a more pronounced difference between the extreme cases.

Each of the 16 treatment combinations in table 6 was run with in both 8 task and 16 task models. The optimization problem was setup in OptQuest with a warm-up period of 100 min and a run-time of 800 min, with 50 replications. These time intervals calculated using similar autocorrelation calculations are different from the times calculated in section 5.2.5. This is because the extreme case here was a 16 task line with CV = 1. At these settings, and with an objective function of maximizing throughput, OptQuest ran ARENA simulations for multiple combinations of control levels within the given constraints, arriving at an optimal control level combination. In some cases, there were multiple optimal control levels producing the same maximum throughput. Results from these experiments are discussed individually for 8 tasks and 16 tasks in the following sections. Although analysis could have been done with both combined, a separate analysis was preferred as the effect of increasing the number of tasks was known (increasing number of tasks increases cycle time, and hence decreases throughput).

6.1.1 Screening Experiment Results - Eight tasks model

Table 7 summarizes the data obtained from the screening experiment for the eight tasks model. The main observation is that the difference between throughputs of maximum throughput (Max Th) case for the new protocol and the throughput for the CBB system was always positive. i.e. the new protocol always outperformed the CBB protocol. The percentage improvement was an average of 0.6%, minimum of 0.2%, and a maximum of 1.3%.

The minimum increase of 0.2% in throughput was found in treatment combination 5 – CV of 0.3, length 20ft, aisle width 10%, and velocity ratio 1:1. The maximum of 1.3% was found in treatment combination 11 - CV of 0.3, length 40ft, aisle width 4%, and velocity ratio 1.5:1. The

reasons behind these specific treatment combinations producing minimum and maximum percentage increase is probed in detail in the main experiments section.

Table 7: Results of Screening Experiment: 8 tasks

Treatment combinations					CBB Th.	New Protocol Th	Difference	% Difference
	A	B	C	D				
1	-1	-1	-1	-1	163.64	164.22	0.6	0.4
2	1	-1	-1	-1	146.73	147.36	0.6	0.4
3	-1	1	-1	-1	163.58	164.02	0.4	0.3
4	1	1	-1	-1	146.69	147.31	0.6	0.4
5	-1	-1	1	-1	163.42	163.76	0.3	0.2
6	1	-1	1	-1	146.44	147.18	0.7	0.5
7	-1	1	1	-1	162.69	163.20	0.5	0.3
8	1	1	1	-1	146.09	147.09	1.0	0.7
9	-1	-1	-1	1	127.13	128.66	1.5	1.2
10	1	-1	-1	1	116.82	117.15	0.3	0.3
11	-1	1	-1	1	126.87	128.49	1.6	1.3
12	1	1	-1	1	116.80	117.31	0.5	0.4
13	-1	-1	1	1	127.05	128.50	1.5	1.1
14	1	-1	1	1	116.78	117.12	0.3	0.3
15	-1	1	1	1	126.95	128.00	1.1	0.8
16	1	1	1	1	115.96	116.63	0.7	0.6

A 2^4 full factorial experiment was setup and run on the new protocol throughput data obtained in Table 7, in Minitab statistical analysis software. Figure 33 shows the half normal plot of the effects from Minitab. Figure 34 shows the main effects plot of all the four factors. It can be seen that factors A and D seem significant. Interaction effect between A and D also seems significant in Figure 33, however, in Figure 35, shows that there might not be much interaction . Factors B, C, and their interaction effects do not seem to be significant.

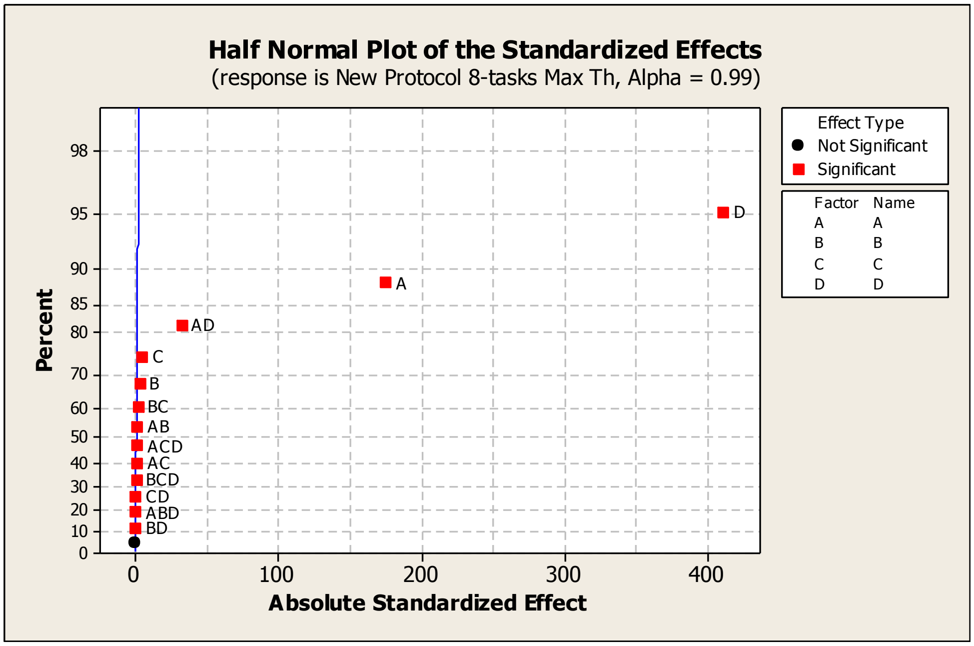


Figure 33: Half normal plot of standardized effects (α = 0.99) for New Protocol Th data for 8 tasks model.

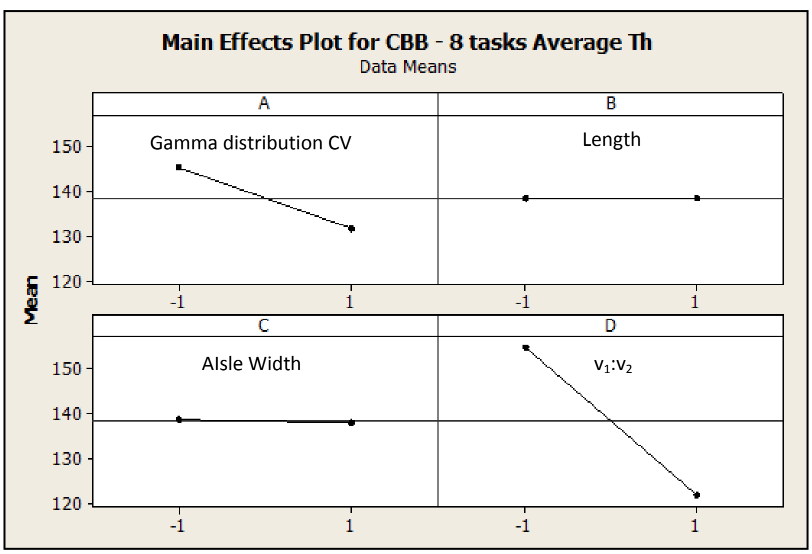


Figure 34: Main Effects plot for New Protocol Th for the 8 tasks model

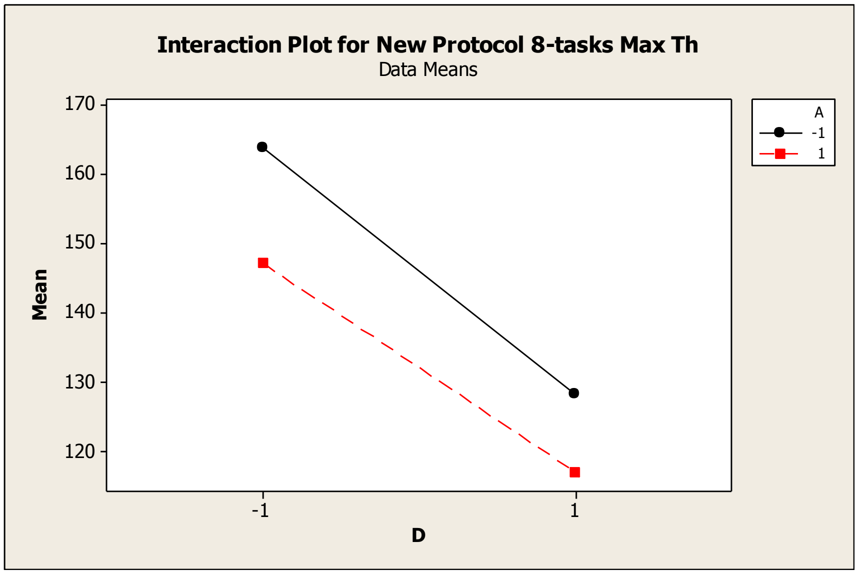


Figure 35: Interactions plot for factors A and D for New Protocol Th data for the 8 tasks model

Based on the previous results, it was possible to reduce the experiment. The data was re-analyzed using Minitab by only considering effects of A, D, and AD, for $\alpha = 0.05$. The results are presented in Table 8. Factors A, and D are statistically significant, as they have large F values and their corresponding P values are close to zero ($P < P_{\alpha=0.05}$). Although the P value for the interaction effect AD is zero, the corresponding SS and MS values are very small compared to the main effects of AD.

Table 8: ANOVA table from Minitab for 8 tasks screening experiment

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	2	5079.14	5079.14	2539.57	26855.43	0
A	1	779.78	779.78	779.78	8245.99	0
D	1	4299.36	4299.36	4299.36	45464.87	0
2-Way Interactions	1	27.09	27.09	27.09	286.44	0
A*D	1	27.09	27.09	27.09	286.44	0
Residual Error	12	1.13	1.13	0.09		
Pure Error	12	1.13	1.13	0.09		
Total	15	5107.36				

Observations from the above data analysis can be summarized as follows:

1. Length (B) and aisle width (C) were both found to be statistically insignificant
2. Gamma distribution CV (A) was found to be statistically significant. Throughput was found to be higher for a CV of 0.3 than for a CV of 1
3. Velocity ratio (D) was also found to be statistically significant. Throughput was significantly higher for $v_1:v_2$ = 1:1, than for $v_1:v_2$ = 1:1.5
4. Interaction effect between Gamma distribution CV (factor A) and velocity ratio (factor D) was found to be statistically significant, but not entirely relevant

Observations 1 and 3 were expected. Longer and wider assembly lines only contribute towards increasing worker walking time. Worker walking times are only a small fraction of the total processing times. Hence, any advantage gained by bringing tasks closer together is negligible in comparison to the total time spent in the assembly line. Observation 3 is likely because a velocity ratio of 1:1.5 means worker 1 now takes 90 seconds on an average to complete a task, whereas worker 2 only takes 60s. Observation 2 and observation 4 are more interesting. At this stage, these observations are difficult to explain, thus are analyzed more in detail in the second round of experiments.

16.1.2 Screening Experiment Results - Sixteen tasks model

Table 9 shows the results of the screening experiment for the 16 task line. Similar to the previous case, the new protocol performed at least as well as the CBB protocol, in all instances. However, the percentage differences in throughput in some conditions were found to be larger than those in the 8 tasks case. The average percentage difference between the New Protocol

throughput and CBB throughput was 1.3%, maximum percentage difference was 3.5%, while the minimum difference was found to be 0 %.

Table 9: Results of Screening Experiment: 16 tasks

Treatment combinations					CBB Th	New Protocol Th	Difference	% Difference
	A	**B**	**C**	**D**				
1	-1	-1	-1	-1	84.49	84.69	0.2	0.2
2	1	-1	-1	-1	79.24	79.71	0.5	0.6
3	-1	1	-1	-1	84.49	84.53	0.0	0.0
4	1	1	-1	-1	79.35	79.45	0.1	0.1
5	-1	-1	1	-1	81.67	84.54	2.9	3.5
6	1	-1	1	-1	79.36	79.58	0.2	0.3
7	-1	1	1	-1	81.65	84.49	2.8	3.5
8	1	1	1	-1	78.98	79.24	0.3	0.3
9	-1	-1	-1	1	67.11	69.29	2.2	3.2
10	1	-1	-1	1	63.76	63.78	0.0	0.0
11	-1	1	-1	1	67.05	68.96	1.9	2.8
12	1	1	-1	1	63.78	63.78	0.0	0.0
13	-1	-1	1	1	67.05	68.53	1.5	2.2
14	1	-1	1	1	63.73	63.75	0.0	0.0
15	-1	1	1	1	66.87	68.98	2.1	3.2
16	1	1	1	1	63.44	64.00	0.6	0.9

The minimum increase of 0 % (or no increase) in throughput was found in treatment combinations 3, 10 and 14. The maximum of 3.5% was found in treatment combination 5 and 7. The reasons behind these specific treatment combinations producing minimum and maximum percentage increase is probed in detail in the main experiments section.

A 2^4 full factorial experiment was setup and run on the New Protocol Th data obtained in Table 9, in Minitab statistical analysis software. Figure 36 shows the half normal plot of the effects from Minitab. Figure 37 shows the main effects plot of all the four factors. It can be seen that factors A and D seem significant. Unlike the eight tasks case, the interaction effect between A

and D did not seem statistically significant. Factors B, C, and all their interaction effects do not seem to be statistically significant as well.

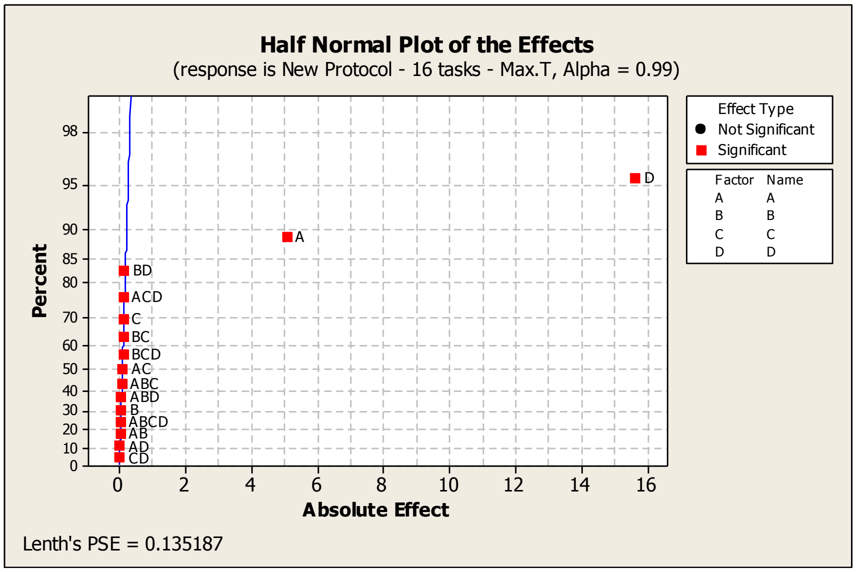


Figure 36: Half normal plot of standardized effects (α = 0.99) for the 16 tasks model

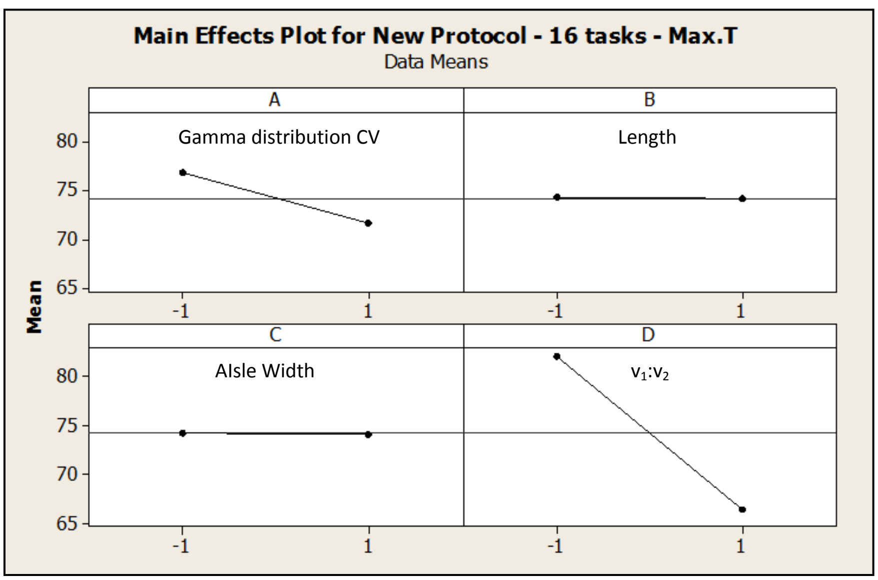


Figure 37: Main Effects plot for the 16 tasks model

Based on the previous results, it was possible to reduce the experiment. The data was re-analyzed using Minitab by only considering effects of A and D, for $\alpha = 0.05$. The results are presented in Table 10. Factors A, and D were both found to be statistically significant, as they have large F values and their corresponding P values are close to zero ($P < P_{\alpha=0.05}$).

Table 10: ANOVA table from Minitab for 16 tasks screening experiment

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	2	1082.69	1082.69	541.343	14722	0
A	1	103.64	103.64	103.637	2818.46	0
D	1	979.05	979.05	979.048	26625.54	0
Residual Error	13	0.48	0.48	0.037		
Pure Error	12	0.48	0.48	0.04		
Total	15	1083.16				

Observations from the above data analysis and results from Table 10 and Figure 37, can be summarized as follows:

1. Length (B) and aisle width (C) were both found to be statistically insignificant
2. Gamma distribution CV (A) was found to be statistically significant. Throughput was found to be higher for a CV of 0.3 than for a CV of 1
3. Velocity ratio (D) was also found to be statistically significant. Throughput was significantly higher for $v_1:v_2 = 1:1$, than for $v_1:v_2 = 1:1.5$
4. No interaction effects were found to be statistically significant

All these observations were similar to those made in the eight tasks case, except observation 4. Observations 1 and 3 were expected. Longer and wider assembly lines only contribute towards increasing worker walking time. Worker walking times are only a small fraction of the total processing times. Hence, any advantage gained by bringing tasks closer together is negligible in

comparison to the total time spent in the assembly line. Observation 3 is likely because a velocity ratio of 1:1.5 means worker 1 now takes 90 seconds on an average to complete a task, whereas worker 2 only takes 60s. Observation 2 and observation 4 are again, interesting. However, at this stage, these observations are difficult to explain, thus are analyzed more in detail in the second round of experiments.

Hence, from the screening experiments, it can be concluded that the main effects of Aisle width and length of the line are insignificant, and they can be eliminated. For further experiments, the length of the line was set to 40ft and the aisle width to 0.1, as these were the observed industry standards (Table 1).

6.2 Main Experiments

Since two factors were eliminated in the first round, there was potential for increasing the number of levels for the second round and exploring more levels within the scope. The previous experiment had considered CVs of 0.3 and 1. For the main experiment, additional CV levels of 0, 0.4 and 0.8 were considered to check if there was linearity in its effect. CV = 0 represents the deterministic case, CV = 0.4 represents a medium variability level, and CV = 0.8 is high variability. Although a minimum CV of 0.3 was stated in the scope (Table 2, section 5. 2.3), the deterministic case was included in the experiment to analyze the effect that the protocol has on variability. The extreme values of the velocity (task time) ratios were kept the same, and a center point of 1.25: 1 was added (table 11 and table 12). In addition, the scope was expanded to consider 24 tasks and a 24 tasks model was developed. Hence, a total of 27 experiments were conducted in this round – nine treatment level combinations for each of the three models.

Table 11: Factors and Levels for the secondary experiment

	Factors	Levels			No. of Levels
		1	***2***	***3***	
A	Gamma distribution CV (constant mean 60s)	0	0.4	0.8	3
B	v_1:v_2	1.:1	1:1.25	1:1.5	3
	Total Number of experimental combinations				**9**

Table 12: Treatment Combinations for the secondary experiment

Treatment Level Combinations	A	B
1	1	1
2	1	2
3	1	3
4	2	1
5	2	2
6	2	3
7	3	1
8	3	2
9	3	3

Each model is discussed in greater detail individually in the following sections. It is to be noted that because the 24 task model is also analyzed here, the waiting time required for its autocorrelation data to stabilize will be the same as calculated in section 5.1.5. Hence, for this round of experiments, a warm-up time of 500 minutes and total run time of 3500 minutes was used.

6.2.1 Main Experiments Results - Eight Tasks Model

For the eight tasks model, the new protocol performed at least as well as the CBB protocol under all conditions. Table 13 presents the results and comparison between the CBB protocol throughputs, and New Protocol Ths. The average percentage difference in the throughput values

between the CBB protocol and the New Protocol Th was 0.2%, while the minimum percentage difference was 0.1%, and the maximum difference was 0.4 %.

Table 13: Results of Main Experiment: 8 tasks

Treatment combinations	A	B	CBB Th	New Protocol Th	Difference	% Difference
1	1	1	743.00	743.93	0.9	0.1
2	1	2	596.00	598.56	2.6	0.4
3	1	3	596.00	596.62	0.6	0.1
4	2	1	688.29	689.25	1.0	0.1
5	2	2	608.84	609.58	0.7	0.1
6	2	3	539.65	541.24	1.6	0.3
7	3	1	648.15	649.00	0.9	0.1
8	3	2	573.16	574.44	1.3	0.2
9	3	3	511.69	512.67	1.0	0.2

Treatment combinations 1, 4, and 7. Treatment combinations 1,3,4,5 and 7 produced a minimum throughput increase of 0.1 %Amongst them, 1,4, and 7 have factor B at level 1: i.e., v_1:v_2 =1:1, the deterministic case. To probe deeper into why these three treatment combinations showed a difference in throughput, the simulation runs for 1, 4, and 7 were observed in ARENA. It was observed in all three combinations that, after the warm-up period, the new protocol behaved like the CBB protocol. That is, no MWS exchanges, or usage of the buffers were observed, only CBB exchanges took place. This is backed-up by the data presented in Table 14. The number of CBB exchanges is almost equal to the throughput, indicating that there was one CBB exchange per unit completed. The 0.1% average throughput increase in these combinations corresponds to a throughput difference of 1 unit on average. This one unit increase in throughput was traced to the warm up stages of the simulation runs. In the CBB protocol, the warm-up stage was exactly the same as the steady state – when another worker was present in the diagonally opposite task, the worker waited till that worker completed the task to CBB exchange with him/her. This waiting was reduced in the warm-up stages of the new

protocol. During this period, worker 2 was found to walk back and MWS preempt worker 1 at least once. This decreased the waiting time of worker 2 during the warm-up period, and hence increased the throughput. But, it was only enough to increase the throughput by one unit. This was because, in combinations 1, 4, and 7, both the workers have equal work velocities. And as there were only 4 tasks on either side of the line, the opportunities for worker 2 to walk back and MWS preempt worker 1 were limited.

Table 14: Results from the secondary experiment - 8 tasks model

Treatment Combination	New Protocol Th	Total No.of CBBs	Worker 1 % Efficiency	Worker 2 % Efficiency
1	743.93	742.88	99.03	99.01
2	598.56	0.00	99.59	79.80
3	596.62	595.90	89.43	99.26
4	689.25	611.66	91.86	91.89
5	609.58	512.68	95.17	86.23
6	541.24	404.36	96.76	79.77
7	649.00	513.80	86.43	86.32
8	574.44	445.98	90.89	80.44
9	512.67	376.60	93.77	74.09

Treatment combination 3,5,6,8 and 9. For treatment combination 3, a similar trend was observed – there was a CBB exchange for every job completed – indicating that the system behaved as a pure CBB system after the warm up period. And a similar advantage of getting a higher throughput was observed. However, as worker 2 was 1.5 times as fast as worker one, worker 2 completed 3 tasks for every 2 tasks that worker 1 completed. This resulted in alternate CBB exchanges at task pairs 1, 7 and 2, 6 (Table 15). This in turn led to worker 2's comparative inefficiency – 89% vs. worker 1's 99% (Table 14), which limited the throughput. The system behaved in a similar manner for treatment combination 5 as well. The percentage difference in

throughput was lesser for treatment combination 5 because of the worker velocity ratio value of 1:1.25 (reflected in the difference in worker efficiencies in Table 14). On observing the simulation run, this difference in worker velocities led to worker 2 following worker 1 for almost three cycles during the warm up period, blocking the second worker, hence causing more waiting, and in turn, lesser throughput. For treatment combination 6, it can be seen from Table 14 that the number of CBB exchanges is significantly lower than the throughput. Again, on observing the simulation run, worker 2 was found to be blocked by worker 1. The number of cycles that this blocking continued was more than for treatment combination 6, as the velocity of worker 2 was 1.5 times that of worker 1, compared to 1.25 times in treatment combination 6. This effect was found between treatment combinations 8 and 9 as well.

Table 15: Average number of CBB exchanges that occur between task pairs: 8 tasks case

	No. of CBB exchanges between tasks		
Treatment Combination	**1,7**	**2,6**	**3,5**
1	368.48	6.92	367.48
2	0.00	0.00	0.00
3	297.96	297.92	0.02
4	199.64	208.78	203.24
5	161.52	193.54	157.62
6	118.60	163.58	122.18
7	166.36	178.36	169.07
8	145.72	154.32	145.94
9	123.87	128.96	123.76

Treatment combination 5, saw a similar effect to treatment combination 3, except that the both the workers were equally efficient. The marginal gain during the warm-up period was also similar, but this time, the exchange positions were different, as the velocity ratio was different.

Treatment combination 2. Treatment combination 2 produced the maximum difference in throughput of 0.4%. For this treatment combination, the total number of CBB exchanges that occurred is zero (from table 15,deterministic case, velocity ratio 1.25:1). This was an anomaly. Further analysis on the simulation and observing the workers' performance revealed why this was an anomaly. By the end of 500 min (warm-up period), worker 2 ends up behind worker 1, following him/her. Since worker 2 is 1.25 times faster than worker 1, he/she is constantly blocked by worker 1, which considerably reduces worker 2's efficiency. To support this, Table 14 shows that worker 2 is at 79.8% efficiency, whereas worker 1 is close to 100% efficiency. Also, worker 2 records 605.38 min blocking time (almost 20% of total time). It can be seen that from Table 15, that for treatment combination 2, there were no CBB exchanges, whereas there were CBB exchanges for every other treatment combination. But, how did this treatment combination show the maximum increase in throughput? On analyzing the results of the CBB simulation for the same treatment combination in ARENA, it is seen that because of the deterministic case and worker 2 is 1.25 times as fast as worker 1, most exchanges occurred at positions 1,7 and 2,6, and there was a long waiting period for both these workers. The workers' efficiencies were 83.4% and 79.7% respectively, causing an overall decrease in throughput for the CBB system. Complete data for blocking times, processing times, number and position of exchanges, etc. has been presented in Appendix III.

ANOVA Experiment. A 3^2 full factorial experiment was setup and run on the New Protocol Th data obtained in Table 13, in Minitab. Each treatment combination consisted of 50 runs. Table 16 shows the results of the ANOVA analysis conducted for $\alpha = 0.05$ in Minitab.

Table 16: ANOVA table from Minitab - 8 tasks model

Source	DF	Seq SS	Adj SS	Adj MS	F	P
A	2	344291	344291	172146	6292.25	0
B	2	1628566	1628566	814283	29763.58	0
A*B	4	98482	98482	24620	899.92	0
Error	441	12065	12065	27		
Total	449	2083404				

From Table 16, factors A and B both are statistically significant as they have large F values and their corresponding P values are close to zero ($P < P_{\alpha=0.05}$). The interaction effect between A and B is also statistically significant. The main effects are plotted in Figure 38. This confirms the data presented in Table 16 – both factors are significant. While factor A seems to have a linear effect on the system, factor B effects a sharp decrease from level 1 to level 2, and then a relatively gradual decrease from level 2 to level 3, thus showing a non-linear trend.

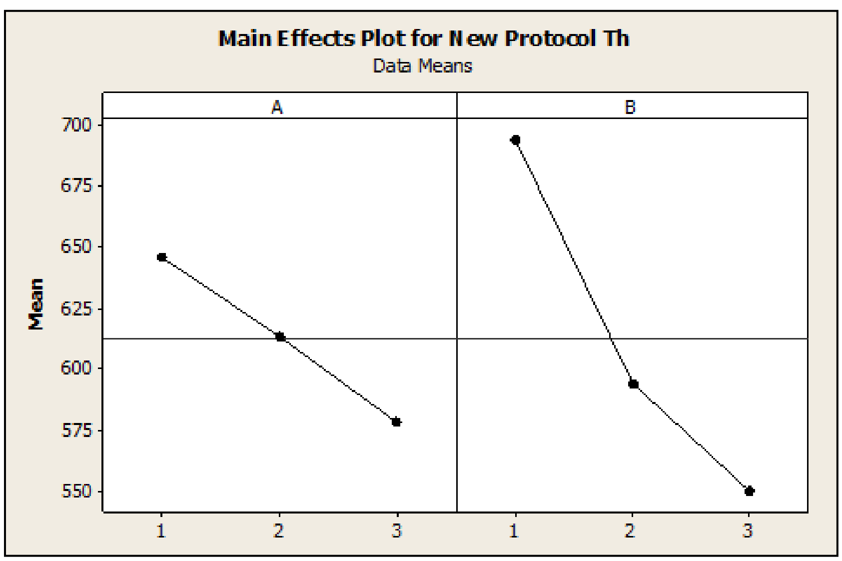


Figure 38: Main Effects plot for 8 tasks model

The interaction effects are plotted in Figure 39. For factor A level 1 and factor B level 2 (treatment combination 2), the maximum throughput is lower than for factor A level 2 and

factor B level 2 (treatment combination 5). This highlights the anomaly of treatment combination 2 discussed previously.

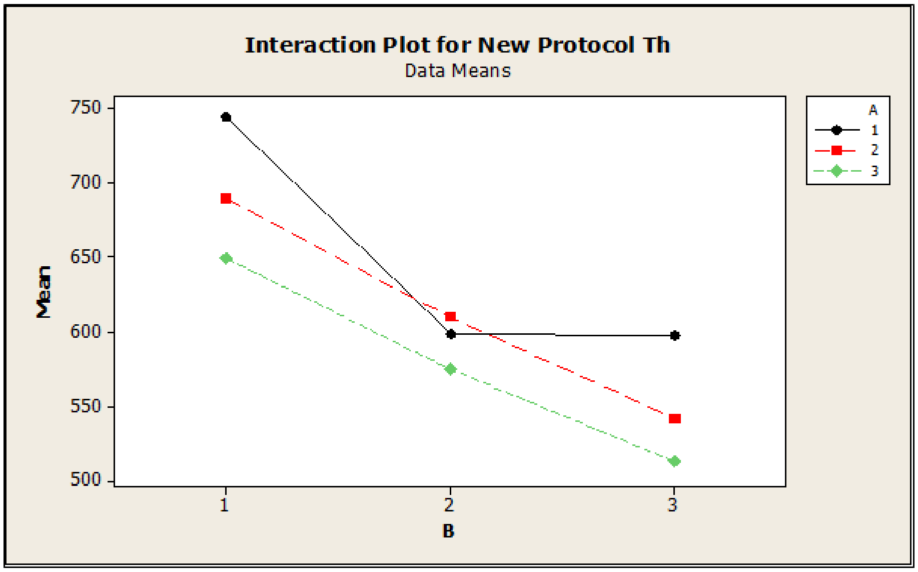


Figure 39: Interaction Effects plot for 8 tasks model

Location of CBB Exchanges. For an eight task model, there were three possible task combinations for CBB exchanges to occur - tasks 1, 7, tasks 2, 6, and tasks 3, 5. Except for treatment combination 2, the symmetric nature of these exchanges can be observed. Worker 1 starts from task 1 and worker 2 starts from task 5. So when they are working at equal velocities (treatment combinations 1, 4, and 7), they tend to alternately exchange almost equally at all task pairs. In other cases, the on-sided nature of the number of exchanges is visible.

Optimal Control Levels. Each New Protocol Th value is associated with set(s) of optimal control level values (C_{ij}). In many cases, OptQuest produced multiple optimal solutions. In all deterministic cases, there were more than two optimal solutions.

Table 17: Number of Optimal Solutions for the 8 tasks case screening experiment

	Number of best solutions
1	many
2	many
3	many
4	One
5	One
6	Two
7	One
8	One
9	One

On analyzing the various control levels for each treatment combination, there was no pattern found amongst the control levels which would aid in identifying them easily. Complete optimal control levels data is presented in Appendix IV.

6.2.2 Main Experiments Results - Sixteen Tasks Model

The results from the second round of experiments for the 16 tasks model are presented in Table 18. The average percentage difference in the throughput values between the CBB protocol and the New Protocol Th was 2%, the minimum difference was 0.1%, while the maximum difference was 14.9 %. Overall, the differences in throughput shown in Table 18 were caused by worker blocking related inefficiencies, similar to those discussed earlier in section 6.2.1 for the eight tasks model. However, there were some exceptions, which are discussed below.

Treatment combinations 5, 6, and 7. The minimum increase of 0.1% in throughput was found in treatment combinations 5, 6, and 7. Table 19 presents the number of CBB exchanges, and the efficiencies of both the workers. For treatment combination 5, during the steady state itself, there were 3 instances of MWS exchange observed - twice after task 1, and once after task 4.

Other than these three instances, there were CBB exchanges in 91% of the times a job was completed (from Table 19). Hence, there was a menial benefit of 0.6% from these MWS exchanges over the CBB case. Similarly, for treatment combinations 6 and 7 as well, similar effects were observed. Although these three treatment combinations produced the least difference in throughput, all other treatment combinations (except for 3) also produced such small differences. The differences between these treatment combinations will be captured in the main effects plot, Figure 41.

Table 18: Results from the secondary experiment - CBB comparison, 16 tasks model

Treatment combinations	A	B	CBB Th	New Protocol Th	Difference	% Difference
1	1	1	373.00	374.00	1.0	0.3
2	1	2	332.00	333.00	1.0	0.3
3	1	3	251.02	288.44	37.4	14.9
4	2	1	357.52	358.10	0.6	0.2
5	2	2	318.24	318.68	0.4	0.1
6	2	3	286.44	286.86	0.4	0.1
7	3	1	344.78	345.26	0.5	0.1
8	3	2	306.73	307.66	0.9	0.3
9	3	3	276.22	277.58	1.4	0.5

Table 19: CBB Exchanges and Efficiency Data for 16 tasks model

Treatment Combination	New Protocol Th	Total No.of CBB Exchanges	Worker 1 % Efficiency	Worker 2 % Efficiency
1	374.00	373.00	99.48	99.38
2	333.00	332.00	96.87	99.40
3	288.44	285.56	95.26	92.00
4	358.10	337.98	95.45	95.42
5	318.68	290.96	97.14	92.30
6	286.86	241.90	97.91	88.30
7	345.26	311.68	92.09	92.12
8	307.66	272.40	94.59	88.37
9	277.58	236.48	96.37	83.91

Treatment combination 3. The maximum increase of 14.9% was found in treatment combination 3. This difference stands out amongst all the data collected during the experiment. This was a deterministic case with worker velocity ratios of 1:1.5. On further analyzing the system in this treatment combination in ARENA, there was no difference between how this system behaved and how other treatment combinations behaved. This is reflected in the worker efficiencies data in Table 19 as well. However, when the CBB protocol was observed, it was noted that worker 2 almost immediately starts chasing worker 1. This is because, as worker 2 is 1.5 times as fast as worker 1 for this treatment combination. This pattern occurred for most part of the run. To check this, an interaction effects plot was created for the CBB throughput data (Figure 40).

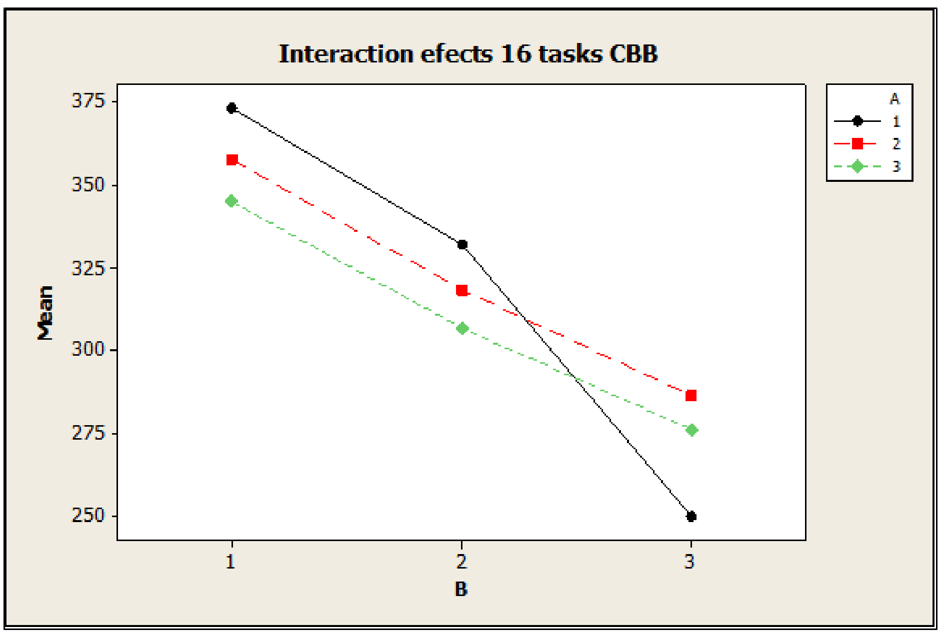


Figure 40: The interaction effects plot for the CBB system in the 16 tasks model with data from Table 16

From this plot, it can be observed that for treatment combination 3 (factor A at level 1 and factor B at level 3) there is a significant decrease in the throughput of the system when

compared to other treatment combinations. Hence, in this case, it is more appropriate to state that the CBB protocol performs much worse than the new protocol, rather than stating that the new protocol outperforms the CBB protocol.

Location of CBB Exchanges. Table 20 presents the data for the average number of CBB exchanges that occur in the new protocol system for this case. As there are 16 tasks, there will now be 7 opportunities to exchange or 7 task pairs at which exchange occurs. The symmetric nature of these exchanges when the workers possess equal work velocities can be seen in treatment combinations 1, 4, and 7. As one worker gets faster than the other worker, it can be seen that the number of CBB exchanges shifts towards the side of the slower worker. The slower worker completes lesser number of tasks in a certain period of time, and within the same time, the faster worker catches up to the slower worker on the other side of the line, prompting an exchange. Also, based on comparing the worker efficiencies presented in Table 19, the faster worker waits more for the slower worker. Complete data is presented in Appendix III. Table 21 shows the number of optimal solutions obtained for each treatment combination from Minitab.

Table 20: Average number of CBB exchanges that occur between task pairs: 16 tasks case

Treatment Combination	No. of CBB exchanges between tasks						
	1,15	**2,14**	**3,13**	**4,12**	**5,11**	**6,10**	**7,9**
1	56.1	130.9	0	0	0	130.2	55.8
2	0	0	166	0	0	166	0
3	0	13.36	42.8	72.3	72.7	57.08	27.32
4	45.72	48.9	49.96	49.68	49.8	48.98	44.94
5	35.46	41.46	45.3	45.2	43.7	44.06	35.78
6	27.6	30.8	39.74	42.32	40.4	33.08	27.96
7	43.62	44.58	44.16	45.36	45.96	45.26	42.74
8	38.06	36.34	40.46	41.24	38.86	39.88	37.56
9	33.12	32.6	35.22	36.82	34.26	31.18	33.28

Table 21: Table showing number of Optimal Solutions for the 16 tasks case screening experiment

	Best solution #
1	many
2	many
3	many
4	One
5	One
6	One
7	many
8	one
9	one

Optimal Control Levels. On analyzing the various control levels for each treatment combination, there was no pattern found amongst the control levels which would aid in identifying them easily. In all deterministic cases, there were more than two optimal solutions. In the 7th treatment combination, there were three optimal solutions. Complete data for optimal control levels is presented in Appendix IV.

Table 22: ANOVA table from Minitab - 16 tasks model

Source	DF	Seq SS	Adj SS	Adj MS	F	P
A	2	34550	34550	17275	1642.68	0
B	2	422361	422361	211181	20081.21	0
A*B	4	5542	5542	1386	131.75	0
Error	441	4638	4638	11		
Total	449	467091				

ANOVA Experiment. A 3^2 full factorial experiment was setup and run on the New Protocol Th data obtained in Table 18, in Minitab statistical analysis software. Each treatment combination

consisted of 50 runs. Table 22 shows the results of the ANOVA analysis conducted for $\alpha = 0.05$, from Minitab.

From Table 22, it can be seen that factors A and B are both statistically significant as they have large F values and their corresponding P values are almost zero ($P < P_{\alpha=0.05}$). The interaction effect between A and B is also significant.

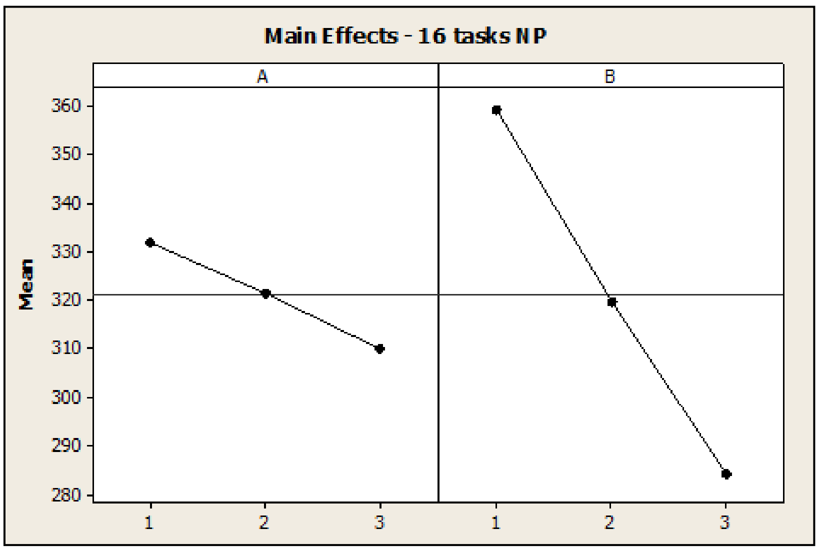


Figure 41: Main Effects plot for 16 tasks model

The main effects are plotted in Figure 41. This confirms the data presented in Table 22, i.e. both factors A and B are significant. Unlike the 8 tasks case, both factors A and B have an almost linear effect on the throughput of the system. Factor B seems to have a larger effect than Factor A. The interaction effects are plotted in Figure 42. All three lines that represent factor A seem parallel to each other for all the levels of B, except for level 3 (treatment 9). Hence, treatment 9 seems to contribute most to the AB interaction effect observed in Table 22.

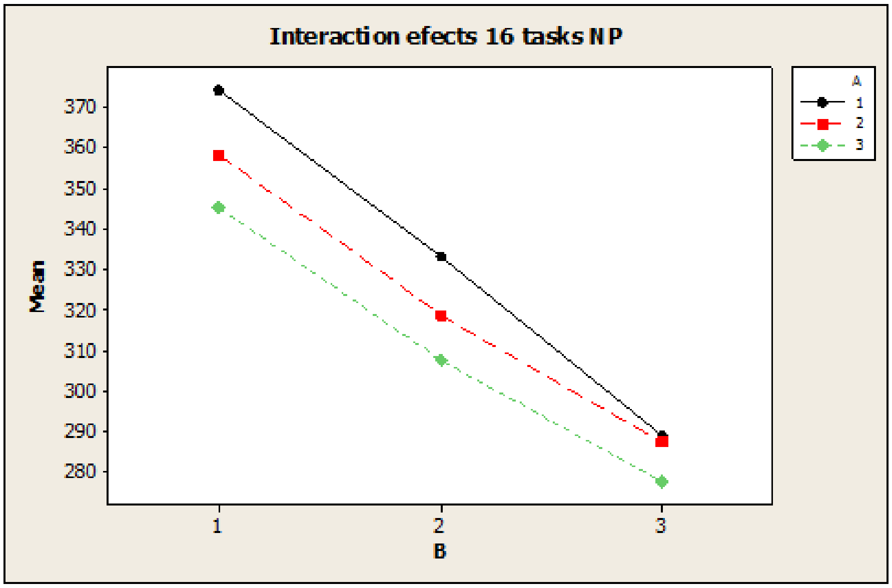


Figure 42: Interaction Effects plot for 16 tasks model

6.2.3 Main Experiments Results - Twenty Four Tasks Model

The results from the main round of experiments for the 24 tasks model are presented in Table 23. The average percentage difference in the throughput values between the CBB protocol and the New Protocol was 0.4%, minimum was 0%, while the maximum difference was 2 % .

Treatment combinations 1,5,7,8 and 9. The 0% difference in throughput was found in treatment combinations 1,7, 8 and 9. Treatment combination 5 also showed a close to 0% increase in throughput. This did not necessarily mean that the CBB protocol was better in these treatment combinations; the CBB protocol was just one of the optimal solutions. In all these cases, there were optimal control levels other than just all zeros (CBB case).

Table 24 presents the number of CBB exchanges, and the efficiencies of both the workers. For the first treatment combination, it can be seen that the total number of CBBs are equal to the

total throughput (New Protocol Th). As discussed in the previous two sections, this meant that there was one exchange per completed job in the line.

Table 23: Results from the secondary experiment - CBB comparison, 24 tasks model

Treatment combinations	A	B	CBB Th	New Protocol Th	Difference	% Difference
1	1	1	249.00	249.04	0.0	0.0
2	1	2	217.00	218.00	1.0	0.5
3	1	3	199.00	199.86	0.9	0.4
4	2	1	241.89	242.44	0.6	0.2
5	2	2	216.00	216.32	0.3	0.1
6	2	3	194.71	198.55	3.8	2.0
7	3	1	235.55	235.55	0.0	0.0
8	3	2	210.05	210.16	0.1	0.0
9	3	3	189.60	189.60	0.0	0.0

Table 24: Results from the secondary experiment - CBB comparison, 24 tasks model

Treatment Combination	New Protocol Th	Total No.of CBBs	Worker 1 Efficiency %	Worker 2 Efficiency %
1	249.04	249.00	99.60	99.61
2	218.00	217.18	99.65	94.30
3	199.86	199.30	99.62	93.24
4	242.44	236.78	96.97	97.02
5	216.32	205.62	97.89	94.87
6	198.55	193.45	98.24	95.01
7	235.55	219.58	94.41	94.21
8	210.16	192.94	96.18	91.70
9	189.60	166.24	97.23	87.13

This could also be observed when analyzing the simulation model during steady state. In fact, as the CBB protocol is as good as the new protocol in these cases, the same throughput, number of CBBs, and worker efficiencies were observed when data was collected for CBB runs. The interesting phenomenon here was that for all the three treatment combinations (7,8 and 9) where CV = 0.8, the CBB protocol did as well as the new protocol. This meant that for the highest

variability condition, the new protocol with all its flexibilities - MWS exchanges and control buffers, did not offer any new advantages. The first treatment combination is a deterministic case and worker velocities are equal. From observing the simulation runs, based on the exchange positions from Table 25, and based on the worker efficiency data in Table 24, it can be said that this was no different from the CBB protocol either.

Treatment combination 6. The highest throughput of 2% was found in treatment combination 6. Table 25 presents the average number of CBB exchanges, and the efficiencies of both the workers. In this case, there were instances of MWS exchanges for the maximum throughput case for treatment combinations 5,6, and 9 - though no recurring pattern could be found.

Table 25:Average number of CBB exchanges that occur between task pairs: 24 tasks case

		No. of CBB exchanges between tasks										
Treatment Combination	**Output**	**1,23**	**2,22**	**3,21**	**4,20**	**5,19**	**6,18**	**7,17**	**8,16**	**9,15**	**10,14**	**11,13**
1	249.04	0	0	0	5	235	20	233	5	0	0	0
2	218.00	0	0	31	0	183	181	4	35	0	0	0
3	199.86	0	0	12	156	32	159	12	4	24	0	0
4	242.44	29	34	41	51	53	56	55	52	45	31	26
5	216.32	24	26	35	41	46	43	44	44	44	38	27
6	198.55	16	30	53	68	70	67	49	20	8	3	3
7	235.48	39	38	42	41	41	38	41	41	40	39	40
8	210.16	34	32	33	36	39	34	36	37	35	33	35
9	189.22	32	26	28	31	33	34	34	32	26	25	31

Location of CBB Exchanges. As this was a 24 task line, there were a total of 11 task pairs as potential locations for CBB exchanges to take place. As was observed in section 6.2.2, the symmetric nature of these exchanges when the workers possess equal work velocities can be seen in treatment combinations 1,4, and 7. As one worker gets faster than the other worker, it

can be seen that the number of CBB exchanges shifts towards the side of the slower worker. To follow this phenomenon, the deterministic cases are ideal (treatment combinations 1,2, and 3). The slower worker completes lesser number of tasks in a certain period of time, and within the same time, the faster worker catches up to the slower worker on the other side of the line, prompting an exchange. Also, based on comparing the worker efficiencies presented in table 24, the faster worker waits more for the slower worker.

Table 26: Table showing number of Optimal Solutions for the 16 tasks case screening experiment

	Best solution #
1	many
2	many
3	many
4	One
5	One
6	Two
7	One
8	one
9	one

Optimal Control Levels. Table 26 presents the number of optimal solutions for each treatment level for this case. In all deterministic cases, there were more than two optimal solutions. In the 6th treatment combination, there were two optimal solutions.On analyzing the various optimal control levels for each treatment combination, there was no pattern found amongst the control levels which would aid in identifying them easily.

ANOVA Experiment. A 3^2 full factorial experiment was setup and run on the New Protocol Th data obtained in Table 23, in Minitab statistical analysis software. Each treatment combination consisted of 50 runs. Table 24 shows the results of the ANOVA analysis conducted for α =

0.05, from Minitab. Similar to the previous two cases, Factors A and B are both statistically significant as they have large F values and their corresponding P values are close to zero ($P < P_{\alpha=0.05}$). The interaction effect between A and B was also found to be significant.

Table 27: ANOVA table from Minitab - 24 tasks model

Source	DF	Seq SS	Adj SS	Adj MS	F	P
A	2	7774	7774	1733	433.25	0
B	2	95031	95031	52801	13200.25	0
A*B	4	1247	1247	84	21	0
Error	441	1044	1044	4		
Total	449	105095				

The main effects are plotted in Figure 43. This confirms the data presented in Table 25, i.e. both factors A and B are significant. Both the factors show some deviation from linearity, but unlike the 8 tasks case, there is no glaringly sharp jump or leveling for either factor.

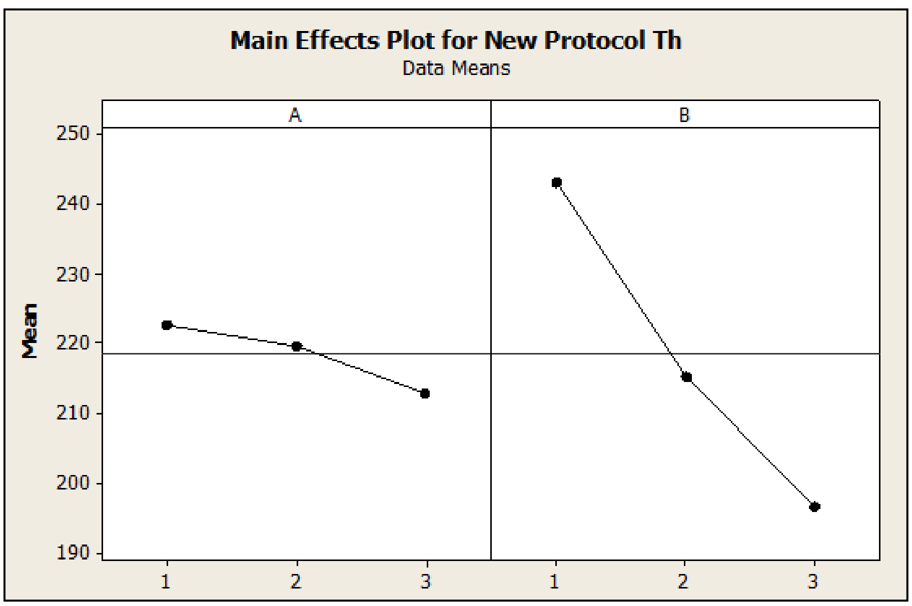


Figure 43: Main Effects plot for 24 tasks model

The interaction effects are plotted in Figure 44. All three lines that represent level A seem parallel to each other for all the levels of B, except for level 2 (treatment combination 2). Hence, treatment 2 seems to contribute most to the AB interaction effect observed in Table 24 and Figure 44.

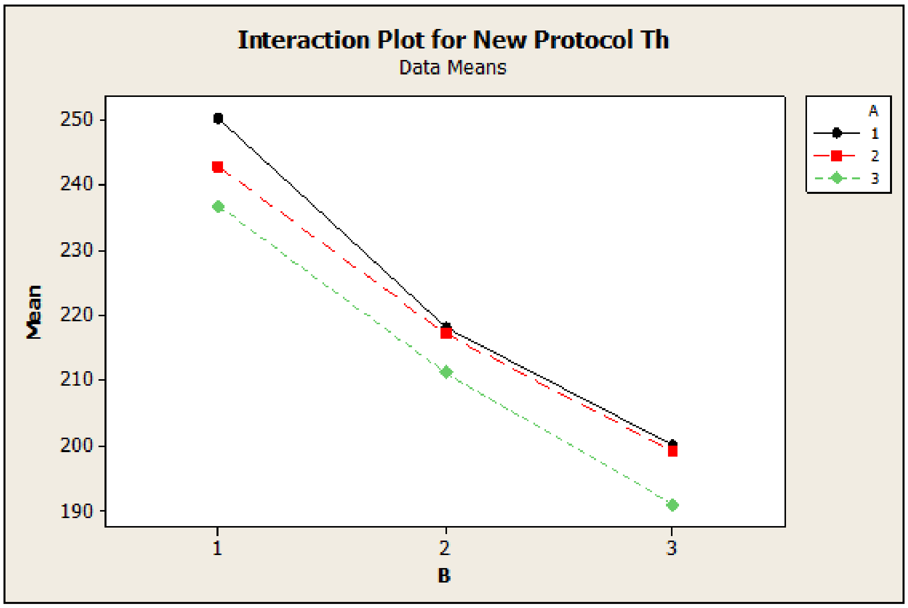


Figure 44: Interaction Effects plot for 24 tasks model

It can also be noted that the level 1 line and level 2 line for factor A are very close to each other for level 2 and level 3 of factor B; whereas, effect for level 3 of factor A seems more pronounced.

6. 3Generalized Results and Observations

In every case and for every treatment combination, it was shown that the new protocol produces at least as much throughput as the CBB protocol does, if not marginally better (average improvement around 1%). For all cases and across all treatment combinations, the new protocol system produced equivalent or higher throughput than the CBB system. There was one stand-

out case where the new protocol significantly outperformed (by 14.9%) the CBB protocol. This was the third treatment combination in the 16 tasks model - when the worker velocity ratio was 1:1.5.

There were some common trends that could be noted amongst the effects of various factors on the throughput of the system that followed the new protocol:

1. Increase in variability (in terms of CV of Gamma distribution) always decreased the throughput of the system. One exception to this rule was the second treatment combination in the eight task model - when the worker velocities were deterministic and their ratio was 1.25:1 (discussed in section 6.2.1). Figure 45 shows the effects of this factor. Based on the slope of the trend line, it can be said that factor A affected 8 tasks case more than 16 tasks case, and 16 tasks case more than 24 tasks case. But, the effect on the 8 tasks case was more profound. The linear tendency of the effect can also be noted.

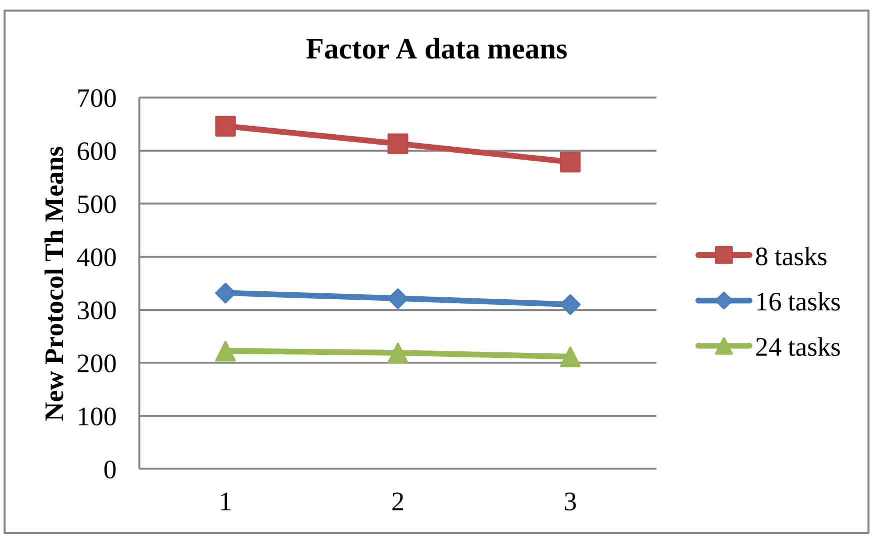


Figure 45: Main effects of Factor A, Gamma distribution CV

2. Change in the aisle width had no statistically significant impact on the throughput of the system
3. Change in the length of the U-line had no statistically significant impact on the throughput of the system
4. Increase in the difference between worker velocities (from 1:1 to 1:1.25 to 1:1.5) always decreased the throughput of the system. This was an expected result, as throughput will naturally decrease when one worker becomes slower than before. One exception to this rule was the second treatment combination in the eight task model - when the worker velocities were deterministic and their ratio was 1:1.25 (discussed in section 6.2.1). Figure 46 shows the effects of this factor. Based on the slope of the trend line, it can be said that factor B affected 8 tasks case more than 16 tasks case. But, the effect on the 8 tasks case was more profound. The linear tendency of the effect can also be noted.

Analysis of the position of CBB exchanges revealed a pattern: these exchanges were symmetric when the workers possess equal work velocities.

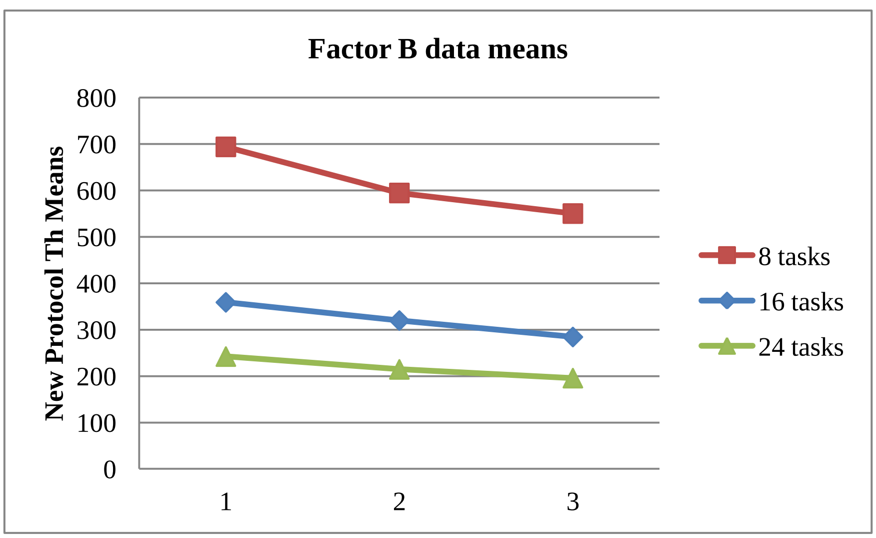


Figure 46: Main effects of Factor B, worker velocity ratio

And as one worker got faster than the other worker, it was observed that the number of CBB exchanges shifts towards the side of the slower worker. This phenomenon could be appreciated more in the 16 and 24 task cases, as there were more opportunities for exchanges to occur. For example, in the case of the 24 tasks model, 48% of all CBB exchanges occurred between tasks 1 and 5 (or 23 and 19), 17% occurred at task pair 6,18, and 35% occurred between tasks 7 and 11 (or 17 and 13). The general rationale behind this phenomenon being that slower worker completes lesser number of tasks in a certain period of time, and within the same time, the faster worker catches up to the slower worker on the other side of the line, prompting an exchange.

In cases where the number of CBB exchanges was close to the throughput of the system, it is safe to say that the system was self-balancing. In fact, in all cases, a certain pattern was repeatable - evident from the position of CBB exchanges across all treatment combinations. During the initial period (warm-up time) the buffers are filled to their control levels, leading to a fixed pattern of job exchange and preemption. Hence, the new protocol, like its predecessors, can be called as self-balancing in nature.

7. Conclusion

The main aim of this research work was to develop a new work-sharing protocol that combined benefits of both, the Cellular Bucket Brigade protocol, and the Modified Work-Sharing system, into one novel protocol. The approach taken was to simulate the protocol and U-line system into a discrete event simulation software (ARENA), and then using a built-in optimization tool (OptQuest), to maximize throughput and find optimal control buffers. Experiments were then run to determine various factors that affected this protocol, and to compare the performance of this protocol with the CBB protocol. Results and observations from these experiments were then analyzed for each individual case, and then generalized

In terms of answering the research questions posed earlier, this work was successful in answering them.

1. An effective work sharing protocol (that approaches self-balancing) was developed for U-shaped assembly lines, and it provided advantages of work-sharing similar to those obtained when the MWS and the CBB protocols are employed independently

2. The factors that primarily affect the performance of this protocol are increasing worker velocities variability, and increasing worker velocity ratios. Both these factors negatively impacted the throughput of the system. This protocol performed at least as well as the cellular bucket brigade protocol, improving the throughput by an average of 1%, and a maximum of 14%. However, it generates more WIP than the CBB protocol.

3. This protocol is generalizable with respect to the number of stations, processing times, types of processes, worker velocities, and choice of empirical distribution.

The experiments conducted and analyzed were for the industrial data collected specifically for this work. The observations made from the analysis could change significantly if different worker velocities are chosen, and if the application required a different empirical distribution of velocities. However, the protocol and model presented by themselves, are robust to handle these changes.

Now that the protocol has been shown to be effective in combining the advantages of both the MWS and the CBB protocols for a U-line with discrete tasks, the protocol may also be applied to a U-line with continuous tasks, without much loss of generality.

Despite performing at least as well as the CBB system, if not better, the new protocol in its current state has a few limitations. The main limitation of this protocol is that it is difficult to comprehend a system that has more than two workers following the protocol. This is because the decision making system will have to be expanded to include the possibility of all the three workers meeting at one point in the line. Also, there was no fixed pattern of optimal control levels. This would only add to the complexity of the protocol when three workers have their own control levels, requiring a significant amount of training compared to the bucket brigade or the MWS systems, as it combines rules from both and assigns priorities. But, the tradeoff between the complexity in executing the protocol in the industry and increased throughput is not straightforward. This will depend a lot on the number of operators, cost of labor, cost of machines, cross-training amongst workers, management buy-in, etc.

8. Future Work

This thesis work developed a framework to include buffers and discrete tasks in existing literature of protocols for U-lines. The limitations described in the previous section allow much scope for future work in this area. During the warm-up time, the buffers get filled to optimal control levels by the workers, and after steady state is reached, the new protocol performs similar to the CBB protocol, barring a few exceptions. To capitalize on these filled buffers, the protocol could be changed slightly. The rule that the worker completes the last task and moves to the first task could be reconsidered. Instead, the finished goods buffer after the last task could be assigned a finite control level. This would prompt the worker to drop the job in the FG and walk back to the previous buffer. This could reduce the probability of workers chasing each other, which would in turn increase efficiency and throughput. Another direction that this work can be expanded is in reducing the complexity of the protocol. What if the control levels were the same for all workers? This could be an interesting question to probe.

The new protocol could be expanded to include more than two workers in the decision making process. The protocol could also be tested with different worker velocities, different velocity ratios, different empirical distributions for these velocities, a larger number of tasks, different aisle widths and line lengths, etc. Another dimension that needs more probing is whether the tradeoff between increased throughput and model complexity is a fair one to make.

There are multiple avenues for improvement and expansion, as discussed above. Overall, this field of work-sharing protocols that account for variable processing times, discrete tasks, and consider including buffers to counter this variability, is fertile for conducting future research.

References

Armbruster, D., & Gel, E. S. (2006). Bucket Brigades Revisited: Are They Always Effective? *European Journal of Operational Research, 172*(1), 213-229.

Bartholdi Iii, J. J., Bunimovich, L. A., & Eisenstein, D. D. (1999). Dynamics of Two- and Three-worker `Bucket Brigade' Production Lines. *Operations Research, 47*(3), 488-491.

Bartholdi Iii, J. J., Eisenstein, D. D., & Foley, R. D. (2001). Performance of Bucket Brigades when Work is Stochastic. *Operations Research, 49*(5), 710-719. doi: 10.1287/opre.49.5.710.10609

Bartholdi, J. J., & Eisenstein, D. D. (1996). A Production Line That Balances Itself, 44, 21-34. Operations Res Soc Amer.

Bratcu, A. I., & Dolgui, A. (2005). A Survey of the Self-balancing Production Lines. *Journal of Intelligent Manufacturing, 16*(2), 139-158.

Daamen, W., Hoogendoorn, S.P. (2006). Free Speed Distributions for pedestrian traffic. Paper presented at the *85th Annual Meeting of the Transportation Research Board*.

Das, B., Garcia-Diaz, A., MacDonald, C.A., & Ghoshal, K.K.. (2010). A computer simulation approach to evaluate bowl versus inverted bowl assembly line arrangement with variable operation times. *International Journal of Advance Manufacturing Technology*, 51, 15-24.

Erel, E., Sabuncuoglu, I., & Aksu, B. A. (2001). Balancing of U-type Assembly Systems Using Simulated Annealing. *International Journal of Production Research, 39*(13), 3003-3015.

Erin, R. (2007). *Mixed-Integer Linear Programming Approach to U-Line Balancing with Objective of Achieving Proportional Throughput per Worker in a Dynamic Environment.* (M.S in Industrial Engineering), Rochester Institute of Technology, Rochester.

Ghiram, G. (2012). *Self-Balancing Processes: True Lean Continuous Flow*: Taylor & Francis Group.

Gong, J., Wang, L., & Zhang, S. (2011). A New Workforce Cross-Training Policy for a U-shaped Assembly Line. *Operations Research*, 235, 1865-0937.

Gökçen, H., & Agpak, K. (2006). A Goal Programming Approach to Simple U-line Balancing Problem. *European Journal of Operational Research, 171*(2), 577-585.

Hopp, W.J., (2008). Factory Physics: McGraw Hill International. Third Edition.

Jordan, W.C., Inman, R.R., Blumenfeld, D.E.: Chained cross-training of workers for robust performance. IIE Transactions, 953–967 (2004)

Kelton,D.W., Sadowski, R.P., Swets, N.P. Simulation with Arena, 5th edition. *McGraw Hill Publications.* Rockwell Automation. (2009)

Levine, R. V. and Norenzayan, A. (1999). The Pace of Life in 31 Countries. *Journal of Cross-Cultural Psychology*, 30 (2), 178–205.

Lim, Y. F. (2011). Cellular Bucket Brigades. *Operations Research, 59*(6), 1539-1545.

Lim, Y. F., & Yang, K. K. (2009). Maximizing Throughput of Bucket Brigades on Discrete Work Stations. *Production and Operations Management, 18*(1), 48-59.

Mason, S., Baines, T., Kay, J.M. & Ladbrook, J. Improving the design process of factories: Modeling the Human Performance Variation. *Journal of Manufacturing Systems;* 2005; *24*(1), 47.

McClain, J. O., Schultz, K. L., & Thomas, L. J. (2000). Management of Worksharing Systems. *Manufacturing & Service Operations Management, 2*(1), 49-67.

Miltenburg, G. J. (1994). The U-line balancing problem. *The Institute of Management Sciences*. 67(2), 1123-1139

Miltenburg, G. J. (2001). Taylor & Francis Online :: One-piece flow manufacturing on U-shaped production lines: a tutorial - IIE Transactions - 33(4). *IIE Transactions*(33), 303-321.

Miltenburg, G.J. (2001). U-shaped Production Lines: A Review of Theory and Practice. *International Journal of Production Economics, 70*(3), 201-214.

Mohler, B. J., Thompson, W. B., Creem-Regehr, S. H., Pick, H. L., Jr, Warren, W. H., Jr. (2007). Visual flow influences gait transition speed and preferred walking speed. *Experimental Brain Research*, 181 (2), 221–228.

Montano, A., Villalobos, J. R., Gutierrez, M. A., & Mar, L. (2007). Performance of Serial Assembly Line Designs Under Unequal Operator Speeds and Learning. *International Journal of Production Research, 45*(22), 5355-5381.

Nakade, K., & Ohno, K., (1997). Analysis and Optimization of a U-shaped Production Line. *Journal of the Operations Research Society of Japan, 40*(1), 90-104.

Nakade, K., & Ohno, K. (1999). An Optimal Worker Allocation Problem for a U-shaped Production Line. *International Journal of Production Economics,* 60(1), 353-358.

Nakade, K., & Ohno, K. (2003). Separate and Carousel Type Allocations of Workers in a U-shaped Production Line. *European Journal of Operational Research, 145*(2), 403-424.

Scholl, A. Homepage for assembly line optimization research. *Online database* http://alb.mansci.de/index.php?content=start&content2=start

Scholl, A., & Klein, R. (1999). ULINO: Optimally Balancing U-shaped JIT Assembly Lines. *International Journal of Production Research, 37*(4), 721-736.

Smunt, T.L. & Perkins, W.C. (1985). Stochastic Unpaced Line Design: Review and Further Experimental Results. *Journal of Operations Management*. *5*(3), 351-373.

Urban, L., Timothy. (1998). Note. Optimal Balancing of U-Shaped Assembly Lines. Tulsa: INFORMS, *Management Science*. 44, 738-741.

Zavadlav, E., McClain, J. O., & Thomas, L. J. (1996). Self-buffering, Self-balancing, Self-flushing Production Lines. *Management Science, 42*(8), 1151.

APPENDIX

I. IRB Approval Form to conduct Lego Experiment in the Toyota Lab

R·I·T

Rochester Institute of Technology

RIT Institutional Review Board for the
Protection of Human Subjects in Research
141 Lomb Memorial Drive
Rochester, New York 14623-5604
Phone: 585-475-7673
Fax: 585-475-7990
Email: hmfsrs@rit.edu

Form C
IRB Decision Form

TO: Srinath Sriram; Andres Carrano

FROM: RIT Institutional Review Board

DATE: **May 30, 2013**

RE: **Decision of the RIT Institutional Review Board**

Project Title – An Innovative Work-Sharing Protocol for U-shaped Assembly Lines

The Institutional Review Board (IRB) has taken the following action on your project named above.

☒ Approved, no greater than minimal risk

Now that your project is approved, you may proceed as you described in the Form A. **Note that this approval is only for a maximum of 12 months; you may conduct research on human subjects only between the date of this letter and May 30, 2014.**

You are required to submit to the IRB any:

- **Proposed** modifications and wait for approval before implementing them,
- Unanticipated risks, and
- Actual injury to human subjects.

Return the Form F, at the end of your human research project or 12 months from the above date. If your project will extend more than 12 months, your project must receive continuing review by the IRB.

Continuing review of research and approval of research studies is required so long as the research study is ongoing, that is, until research-related interactions and interventions with human subjects or the obtaining and analysis of identifiable private information described in the IRB-approved research plan have been completed.

Investigators are responsible for submitting sufficient materials and information for the IRB to meet its regulatory obligations, and should follow the institutional policies and procedures for continuing IRB review of research that are required by HHS regulations at (45 CFR 46.103(b)(4), 45 CFR 46.109(e), 45 CFR 46.115(a)(1)) as appropriate to the research activity.

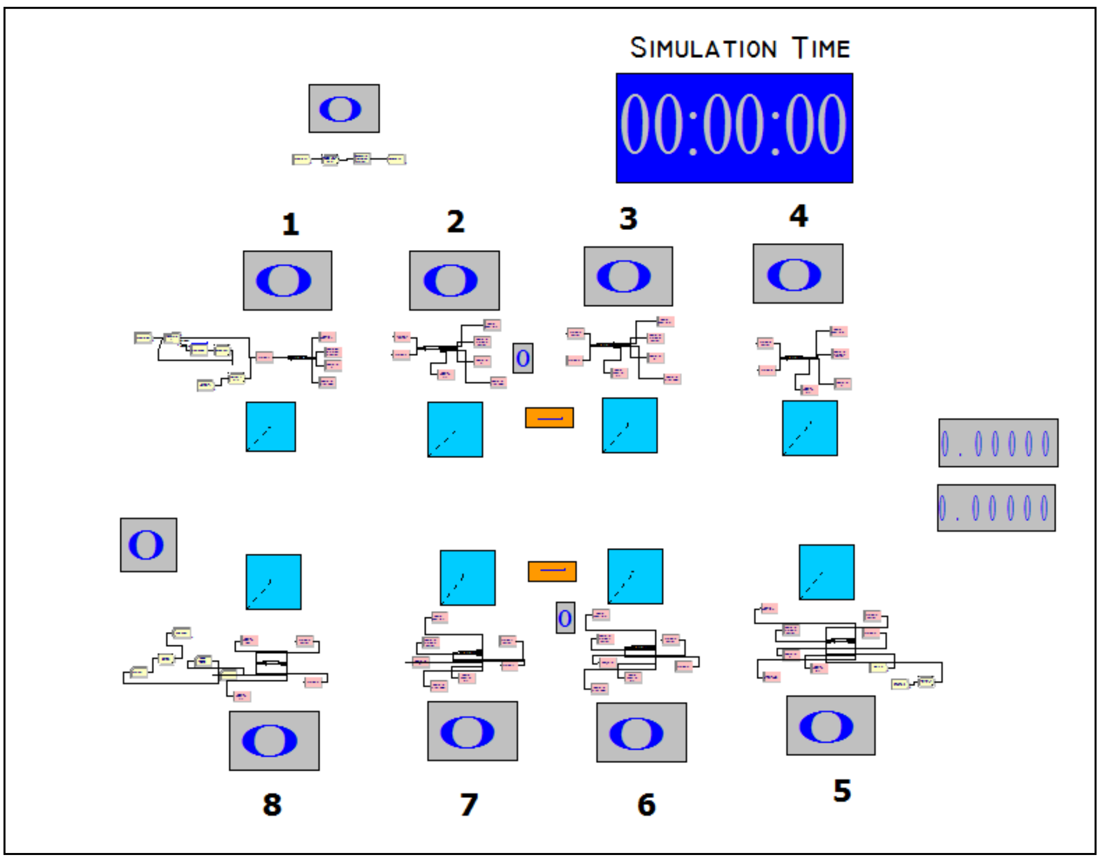
Heather Foti, MPH
Associate Director
Office of Human Subjects Research

Revised 02.09.2011

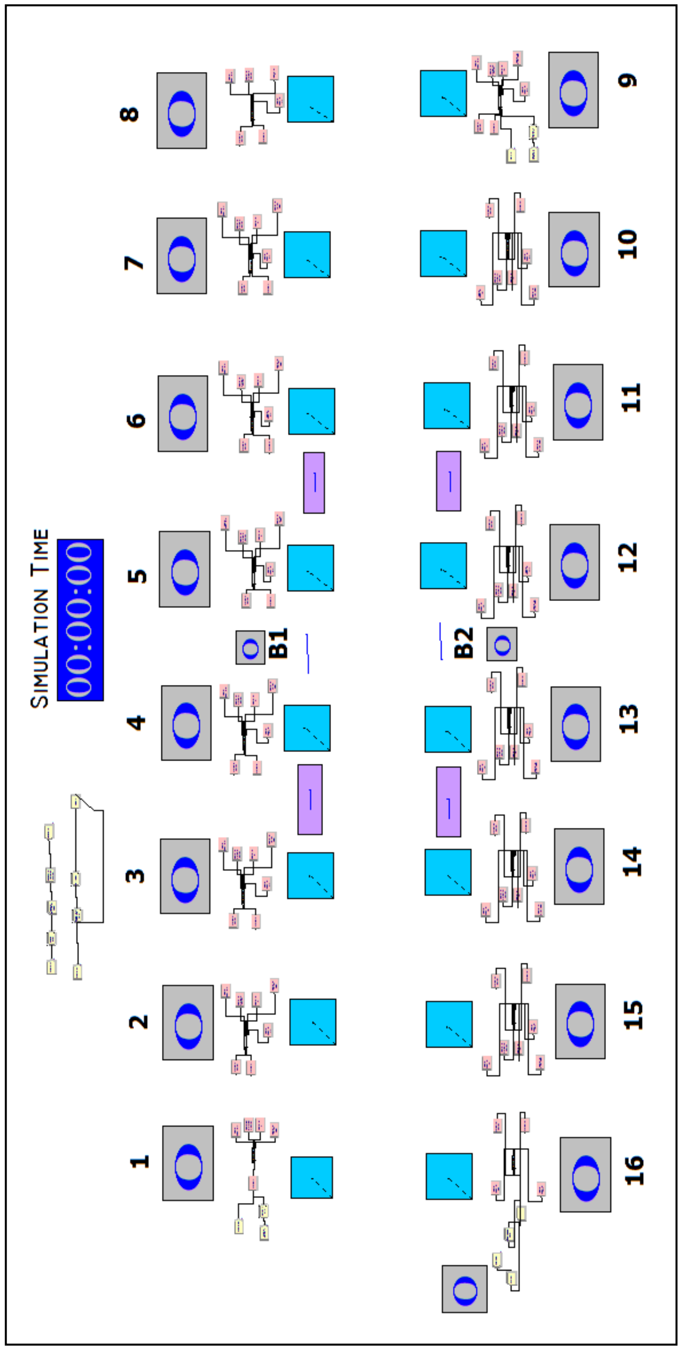
II. Snapshots of ARENA models

The zeros in grey boxes are to show the number of workers in that station. The blue boxes are to animate the workers when they move from one station (task) to another. The pink rectangles show 4 buffers.

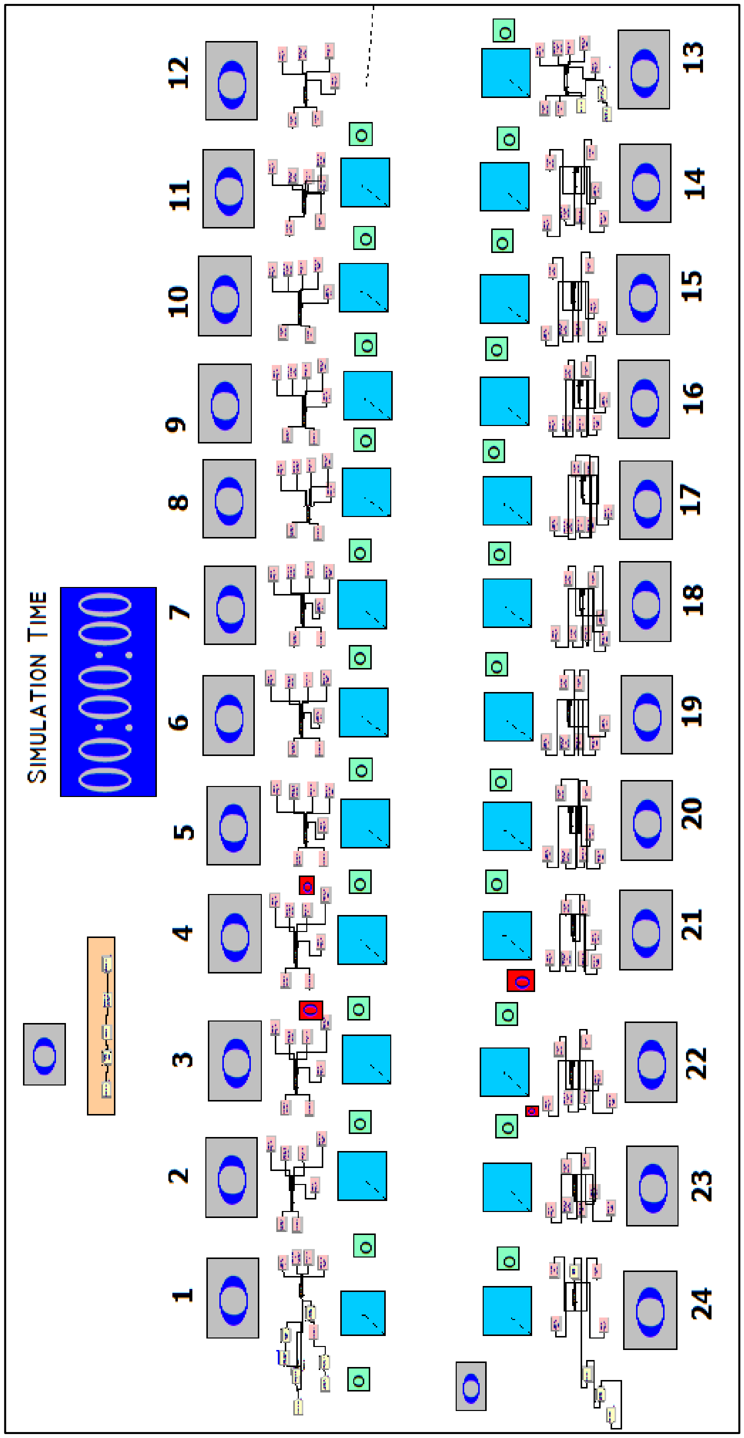
8 Tasks Model:



16 Tasks Model:



24 Taks Model:



III. Data from Secondary Experiments

The data collected from the main experiments discussed in section 6.2 is presented here. For each treatment combination, the throughput (Output), and the number of CBB exchanges (CBBs) are given at the optimal control levels. Also, for both the workers, the total time spent in processing a job moving forward (Proc. Time), waiting times for MWS and CBB exchanges, the amount of time that the worker was blocked by the other worker (Blocking Wait Time), the amount of time spent in moving without carrying a job, and worker efficiencies (Worker Eff %) are also presented here. Efficiency was calculated by using the formula:

$$ff\% = \frac{Proc.Time}{Total\ Simulation\ Time}\ .$$

8- tasks data (all times in minutes)

Treatment Combination	Output	CBBs	Worker 1 Proc. Time	Worker 2 Proc.Time	Worker 1 MWS Wait Time	Worker 2 MWS Wait Time	Worker 1 CBB Wait Time	Worker 2 CBB Wait Time	Worker 1 Blocking Wait Time	Worker 2 Blocking Wait Time	Move Time Worker 1	Move Time Worker 2	Worker 1 Eff %	Worker 2 Eff %
1	744	743	3466	3465	1	1	4	4	0	0	29	30	99	99
2	599	0	3486	2793	1	2	0	1	0	605	12	98	100	80
3	597	596	3130	3474	0	2	346	0	0	0	24	24	89	99
4	689	612	3215	3216	2	1	204	205	48	47	31	31	92	92
5	610	513	3331	3018	2	2	134	266	9	171	23	42	95	86
6	541	404	3387	2792	0	4	92	281	2	356	20	66	97	80
7	649	514	3025	3021	2	2	242	245	190	193	41	40	86	86
8	575	446	3181	2816	1	3	188	301	102	326	27	54	91	80
9	513	377	3282	2593	3	3	136	333	57	496	22	75	94	74

16- tasks data (all times in minutes)

Treatment Combination	Output	Average Cycle Time	Worker 1 Proc. Time	Worker 2 Proc.Time	Worker 1 MWS Wait Time	Worker 2 MWS Wait Time	Worker 1 CBB Wait Time	Worker 2 CBB Wait Time	Worker 1 Blocking Wait Time	Worker 2 Blocking Wait Time	Move Time 1	Move Time 2	Worker 1 Eff %	Worker 2 Eff %
1	374	48	3482	3478	0	3	3	4	0	0	15	16	99	99
2	333	81	3390	3479	1	6	95	1	0	0	14	14	97	99
3	288	209	3334	3220	0	10	154	248	0	9	12	13	95	92
4	358	167	3341	3340	1	4	112	112	31	28	16	16	95	95
5	319	205	3400	3230	0	7	82	143	5	100	13	20	97	92
6	287	72	3427	3091	0	0	61	162	1	223	11	25	98	88
7	345	121	3223	3224	1	3	148	146	110	108	18	19	92	92
8	308	58	3311	3093	1	3	116	180	59	202	14	21	95	88
9	278	162	3373	2937	0	5	85	208	30	321	12	29	96	84

24- tasks data (all times in minutes)

Treatment Combination	Output	Average Cycle Time	Worker 1 Proc. Time	Worker 2 Proc.Time	Worker 1 MWS Wait Time	Worker 2 MWS Wait Time	Worker 1 CBB Wait Time	Worker 2 CBB Wait Time	Worker 1 Blocking Wait Time	Worker 2 Blocking Wait Time	Move Time 1	Move Time 2	Worker 1 Eff %	Worker 2 Eff %
1	249	84	3486	3486	1	1	2	2	0	0	11	11	100	100
2	218	99	3488	3300	0	5	2	183	0	1	10	11	100	94
3	200	95	3487	3263	2	6	3	221	0	0	9	9	100	93
4	242	385	3394	3396	3	3	78	77	14	12	11	12	97	97
5	216	445	3426	3320	1	12	58	99	5	55	10	14	98	95
6	199	426	3438	3325	1	5	50	119	0	39	10	12	98	95
7	235	377	3304	3297	7	5	101	105	75	80	13	13	94	94
8	210	482	3366	3210	6	7	81	124	36	145	11	14	96	92
9	189	496	3403	3050	7	9	61	146	19	277	10	18	97	87

IV. Optimal Control Levels Data from Secondary Experiments

Optimal Control Levels for 8 tasks case

		Treatment Level Combinations												
	Buffer	**1**	**2**		**3**		**4**	**5**	**6**		**7**	**8**		**9**
	2	0	1	2	0	0	1	0	2	2	0	2	2	0
	3	0	1	1	1	1	2	1	1	1	2	2	2	0
	4	0	2	0	2	2	1	1	2	0	2	2	2	0
Worker 1	5	1	0	1	1	1	0	1	2	0	0	0	1	1
	6	0	2	0	1	1	2	0	1	1	2	2	2	0
	7	1	1	0	2	2	0	0	0	2	0	2	2	0
	8	1	1	2	1	1	1	1	0	2	0	0	0	0
	2	2	0	2	2	2	2	0	2	0	1	1	1	0
	3	0	2	0	0	0	0	1	1	1	0	0	0	0
	4	1	0	2	1	1	1	0	0	1	2	1	1	0
Worker 2	5	1	0	1	1	1	0	1	1	1	0	1	1	0
	6	1	2	1	2	2	1	1	0	2	0	0	1	0
	7	1	1	2	2	2	1	0	2	0	0	2	1	0
	8	1	1	0	0	1	1	1	1	1	0	1	1	0

Optimal Control Levels for 16 tasks case

		Treatment no								
worker	**Buffer**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
	2	0	0	0	0	0	0	0	0	1
	3	0	0	0	2	0	0	0	0	1
	4	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0
	6	1	0	0	0	0	0	0	0	0
	7	0	0	0	2	0	0	0	1	0
	8	0	0	0	1	0	0	1	0	1
1	9	0	1	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	1	2
	11	0	0	0	0	0	0	0	0	1
	12	0	0	0	0	0	0	0	0	1
	13	0	0	1	0	0	0	0	0	1
	14	0	0	0	0	0	0	0	1	0
	15	0	0	0	0	0	0	0	0	1
	16	0	0	0	0	0	0	0	0	1

	2	0	2	0	0	0	0	0	0	2
	3	0	0	0	0	0	0	0	0	1
	4	0	2	1	1	0	0	0	1	1
	5	0	0	0	0	0	0	0	0	0
	6	0	1	0	1	0	0	0	1	2
	7	0	0	0	0	0	0	0	1	0
	8	1	0	0	0	0	2	1	0	2
2	9	0	1	1	0	0	0	0	0	2
	10	0	1	0	0	0	0	0	0	1
	11	0	0	0	0	0	0	0	0	0
	12	1	0	1	0	0	0	1	0	0
	13	1	0	1	0	0	0	1	0	2
	14	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0

Optimal Control Levels for 24 tasks case

		Treatment no								
worker	**Buffer**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
	2	0	0	0	0	0	0	0	0	1
	3	0	0	0	2	0	0	0	0	1
	4	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0
	6	1	0	0	0	0	0	0	0	0
	7	0	0	0	2	0	0	0	1	0
	8	0	0	0	1	0	0	1	0	1
	9	0	1	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	1	2
	11	0	0	0	0	0	0	0	0	1
1	12	0	0	0	0	0	0	0	0	1
	13	0	0	1	0	0	0	0	0	1
	14	0	0	0	0	0	0	0	1	0
	15	0	0	0	0	0	0	0	0	1
	16	0	0	0	0	0	0	0	0	1
	17	0	0	0	2	1	0	0	1	0
	18	0	0	0	0	0	0	0	0	0
	19	0	1	0	0	0	0	0	0	0
	20	1	0	1	0	0	0	0	0	0
	21	1	2	0	0	1	0	0	0	1

	22	0	0	0	0	1	0	0	0	0
	23	0	0	0	0	0	1	1	1	2
	24	0	1	0	0	0	0	0	0	1
2	2	0	2	0	0	0	0	0	0	2
	3	0	0	0	0	0	0	0	0	1
	4	0	2	1	1	0	0	0	1	1
	5	0	0	0	0	0	0	0	0	0
	6	0	1	0	1	0	0	0	1	2
	7	0	0	0	0	0	0	0	1	0
	8	1	0	0	0	0	2	1	0	2
	9	0	1	1	0	0	0	0	0	2
	10	0	1	0	0	0	0	0	0	1
	11	0	0	0	0	0	0	0	0	0
	12	1	0	1	0	0	0	1	0	0
	13	1	0	1	0	0	0	1	0	2
	14	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0
	17	1	1	1	0	1	0	0	0	0
	18	1	0	1	0	0	0	1	1	2
	19	0	1	2	0	2	0	1	0	0
	20	0	0	0	0	0	0	1	0	0
	21	1	0	0	1	0	0	0	0	1
	22	2	0	0	2	0	1	1	0	1
	23	1	2	0	0	1	1	0	1	0
	24	0	0	1	0	2	0	0	1	0

V. Snapshot of a task submodel in ARENA

Task 1 modeled in ARENA. This is a suubmodel with one entry node and 4 exit nodes