View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Texas A&M University

A SIMULATION ANALYSIS OF CONSTRAINED RATE AND LINE ASSEMBLY PROCESSES

A Thesis

by

RYAN COOLEY HILLEGASS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Lewis Ntaimo
Co-Chair of Committee,	Amarnath Banerjee
Committee Member,	Sivakumar Rathinam
Head of Department,	Mark Lawley

May 2018

Major Subject: Industrial Engineering

Copyright 2018 Ryan Cooley Hillegass

ABSTRACT

Simulation presents a way to analyze the performance of a system with zone capacity constraints, operator constraints, and precedence constraints in an assembly line using takt analysis. A small-scale model of an aircraft assembly line is built in Simio and precedence constraints are modified in independent simulations. The primary performance metric is traveled work, for which a definition is given. A method of calculating traveled work is presented, as well as an interpretation that states the effect on throughput. These results show that, *ceteris paribus*, traveled work increases flowtime, which decreases throughput. Modifications to the system are suggested that can reduce traveled work.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors Lewis Ntaimo and Amarnath Banerjee of Department of Industrial and Systems Engineering and Sivakumar Rathinam of the Department of Mechanical Engineering.

All work for the thesis was completed independently by the student.

Funding Sources

Graduate study was supported by an assistantship though Texas A&M University, paid for by Lockheed Martin.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

1. INTRODUCTION

In large-scale assembly operations with high variability and many constraints, a system's performance is sometimes difficult to measure. It can be even more difficult to attribute performance to one cause or another. A good example of such a situation is one of Lockheed Martin's assembly operations. Since its inception in 2006, one of their projects has consistently missed deadlines and costs have been higher than budgeted (Capaccia 2017). As of 2018, the assembly floor is still fairly new and there is much to take into account when considering why the project has underperformed. While part shortages have plagued assembly, the associated tasks for those parts cannot be entirely to blame for derailment. This illustrates some of the problems associated with complex assembly processes.

Furthermore, this particular assembly floor is still expanding as production is ramped up. It uses "takt time" to both set the tempo for "pulsing". A takt is a fixed amount of time (hours, days, etc.) during which work occurs for all entities in the system. Each time the takt time has elapsed, all entities pulse—or move from their current station to the next station in the assembly line. By definition, an entity leaves the system at the end of each takt period, and another enters the system simultaneously. Pulsing differs from traditional assembly lines because it is a discrete movement of each entity to its next station in the sequence during which the entire line temporarily stops work. Once all entities (generic widget) have arrived at their new station, work continues. At peak production, the line will consist of 16 parallel "rate stations," that is, stations where an entity remains in one location for an integer multiple of the takt time. This is followed by five parallel, rate, "soft stations". A soft station works like a buffer, but also has a small number of tasks that must be performed at the location. These feed into a sequence of 10 pulsing stations (processing entities for a single takt before progressing in the sequence) and then two parallel pulsing lines each with seven sequential positions. The line concludes with two parallel, rate, buffer-stations.

Parallel stations allow entities to work simultaneously on the same set of tasks while occupying a separate physical location. Figure 1 gives an example illustration of parallel lines, each containing three stations. Entities would enter from the left of Server 6 and would exit the lines to the right of Server 8.

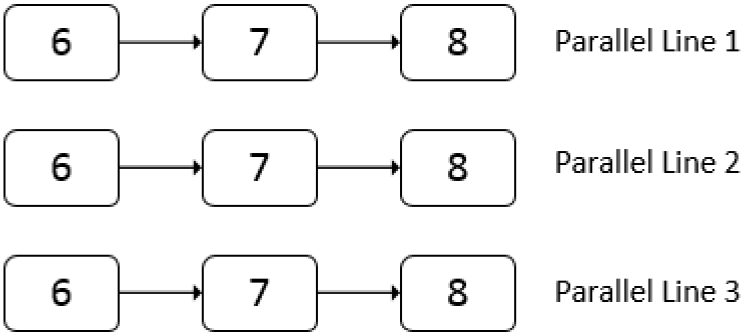


Figure 1: Lines in Parallel

The performance measures important to aircraft assembly are those that ensure on-time delivery of planes. Ideally with takt analysis, throughput would be consistent and derivative of the takt rate. This relationship would be consistent in the absence of “traveled work,” or scheduled in-station work that is not completed when the entity is pulsed. Traveled work is equivalent to the percent of unfinished work times the takt rate and number of takts at the position. The sum of each position’s traveled work gives the total amount of unfinished work on the entity and the total delay that the entity inherits beyond the theoretical makespan. This traveled work is processed either at buffer stations or downstream from the intended station. The former approach is a solution that requires more buffers and takes up precious floor space, while the latter approach compromises a line's balance and can exacerbate downstream problems.

For complex assemblies with lengthy makespan, operator utilization is important and is related to node (representation of a zone, or work location) utilization. Utilization of an operator is the percent of time that an operator spends performing tasks. For any given task, there will be X-number of operators required to perform the task that requires T-hours. Utilization of a node is the average percent of a node's capacity that is used. Node capacity is determined by the number of workers that can work in one area simultaneously.

Because the motivation of this thesis is a real-life problem, the focus is on solutions that are implementable. For example, adding more stations to the assembly line may improve throughput and decrease costs, but it may not be feasible given a company's floor space constraint. While a true precedence is strict and cannot be changed, some precedence constraints are softer and designed by engineers as the ideal order for tasks to be done. With that knowledge, precedence will be treated as variable. To contrast, part shortages can only be indirectly controlled by changing part providers, finding alternative parts, etc. Therefore, the total number of operators, node capacity (to some degree) and order of tasks can be modified, and a simulation model can be used to observe the results. It is also uncertain if changes in these constraints will affect rate stations differently than sequential pulsing positions.

There are three research questions that will be answered within this thesis:

RQ1. How do variability and changes in constraints affect the performance of a system that uses takt-time assembly?

RQ2. How do alterations to precedence affect performance of the system?

RQ3. What is traveled work and how does it factor into throughput?

The research questions will be answered by modifying input constraints in the simulation software Simio and analyzing the results from experiments.

2. LITERATURE REVIEW

The literature on this topic is most often divided between four areas: precedence constraints, takt analysis, optimization, and simulation. While some of these are often found paired together—like simulation and takt analysis in Duanmu & Taaffe (2007 and 2012), Millstein & Martinich (2014), Sandanayake et al. (2008), Seppanen (2014), and Wu & Hui (2008)—they are not often accompanied by precedence constraints. In fact, other than the survey study by Battia & Dolgui (2013), the systematic literature review performed for this thesis did not find any papers written considering all three aspects (simulation, takt analysis, precedence constraints).

2.1 Precedence Constraints

Precedence constraints are a central focus of this thesis and have been well-studied, with substantial contributions coming in the 60’s and 70’s. These early works are mostly concerned with optimizing sequences of precedence constraints (Garey 1973, Macaskill 1972) or line balancing (Hoffmann 1963). More recently the focus has shifted to simulation with precedence constraints.

2.2 Takt Analysis

Takt systems are used in a variety of assembly and manufacturing processes. Millstein & Martinich (2014) demonstrate how increasing or decreasing takt time and WIP in a CONWIP system (even to the extremes) affect the performance of an assembly line using throughput, WIP inventory, and flowtime as system performance metrics. This thesis uses a fixed takt rate because it is assumed to be necessary to meet the demand schedule. Similarly in a takt analysis environment, Sandanvake et al. (2008) were concerned with the same performance metrics when they used total quality control (quality checks at each station to increase chance of catching defects early), set-up-time elimination plans (modified by reducing set-up-time), and line-balancing (changing the number of stations in this case) in a simulation environment to examine the effects of each factor. Wu and Hui (2008) theorized that when utilization is 1.0 and in the presence of an infinite queuing WIP, mean service time is equal to effective processing time. Although an interesting result, it is not applicable to a system that possesses neither of those traits.

2.2.1 Example Characteristics of a Practical Setting

Some aircraft production lines assemble several types of aircraft, but there may not be additional set-up or change-over time when an aircraft of a different variant arrives at a station. Rather, due to learning limitations, there is likely some additional processing time at stations when less common variants arrive. Seppänen (2004) considers the phenomena of learning and pits a takt-time system with low degree learning (because operators are working on many different tasks) against the Location-Based Management System (LBMS) that sees operators move locations and work the same set of tasks repeatedly. This method gets the most benefit from learning but is not preferred in systems that do not have a sizeable, workable backlog from which operators can begin new tasks as soon as the previous task is completed. Operators may be less familiar with the specific tasks, and the *complexity* of tasks decreases performance (Zhu et al. 2008). Seppänen (2004) does not apply simulation to the assembly line, however. The application chosen instead is the assembly of a building. For the assembly line in question, workers can receive the learning benefits of LBMS while utilizing the large workable backlog seen in a takt system that may encounter traveled work.

As mentioned in the introduction, another common characteristic of airplane assembly line production is the inclusion of several types of stations in various configurations—rate stations, pulsing lines, and buffer stations. Some of these are in parallel and others are stations that each aircraft must pass through in its sequence. The effect of combinations of different station configurations can be studied with relative ease mathematically when stations are perfectly balanced and work is consistently completed in the allotted time (Battaia & Dolgui, 2013). Because observed assembly line performance is low, one or both of these may not be the case and mathematical analysis alone is likely not an appropriate way to study the system.

2.3 Simulation

The use of simulation is important, as it is the only feasible way to examine the effects of constraint changes on complex, nonlinear systems. Slight changes to the real system could have far-reaching consequences, but simulation is a great tool to test those changes in a low-stakes environment. A number of papers in archived journals and conference proceedings examine real-life assembly systems via modeling-and-simulation. Most often they are comparing types of production control systems and their effect on some set of performance metrics, (Hopp & Roof 1998, Millstein & Martinich 2014, Seppanen 2014) or demonstrating how modifications to certain variables affect these performance metrics (Duanmu & Taaffe 2007, Duanmu & Taaffe 2012, Macaskill 1972, McMullen & Frazier 1998, Mendes et al. 2005, Sandanayake et al. 2008, Wu & Hui 2008).

2.4 Optimization

Simulated annealing has been used in assembly process optimization, where small changes in constraints within experiments approximate optimums. In this way, it “attempts to avoid being trapped at local optima in its search for the global optima” (McMullen & Frazier 1998).

Duanmu and Taaffe (2007) used variable buffer sizes and small modifications to processing time to optimize throughput using Arena simulation software. While reducing processing time in practice is not always feasible, buffers can sometimes be used to complete traveled work and potentially perform the remaining tasks in a more efficient way with a workable backlog. To illustrate this, say the operators are unable to complete a task at two different stations. Rather than interrupting the flow of the system and allowing more time for the work to be completed in-station, a buffer station may allow the two tasks to be performed simultaneously, effectively turning two separate delays into one stop at a buffer station for a single takt. This keeps the line's throughput consistent.

With the way buffers are used to catch traveled work and the way new entities enter the system, aircraft manufacturers use a type of CONstant Work-In-Process production control system (CONWIP). CONWIP systems have a fixed number of entities at any one time, but production control systems are designed to increase or decrease WIP (number of entities in the system) when certain criteria are met. There has been a lot of work on statistical control of WIP in CONWIP systems—Framinan, Gonzalez, and Ruiz-Usano (2003) assembled a meta-analysis of CONWIP systems that use various methods of statistical control. The observed system does not possess the floor-space to change WIP, so this is not a viable way to decrease throughput or traveled work.

While optimization is beyond the scope of this paper, the conclusion will feature recommended ways to improve performance of the described system and discuss the feasibility of each.

Table 1 is a matrix of each of the studies found for this paper, with their contents classified by the four areas discussed.

Table 1: Literature Topic Matrix

	Precedence Constraints	**Takt Analysis**	**Simulation**	**Optimization**
Battaia, Dolgui 2013[*]	X	X	X	X
Duanmu, Taaffe 2007		X	X	X
Duanmu, Taaffe 2012		X	X	
Framinan et al. 2003				
Garey 1973	X			X
Hoffmann 1963	X			X
Hopp, Roof 1998			X	X
Macaskill 1972	X		X	X
McMullen, Frazier 1998	X		X	X
Mendes et al. 2005	X		X	X
Millstein, Martinich 2014		X	X	X
Rios et al. 2012	X			X
Sandanayake et al. 2008		X	X	
Seppanen 2014		X	X	
Wu, Hui 2008		X	X	
Zhu et al. 2008	X			

* Battaia & Dolgui 2013 is a review of existing literature

3. MODEL

3.1 Baseline Model

Through the literature review for this thesis paper, there have not been any pieces written taking into account precedence constraints and takt analysis while measuring these effects via simulation. The motivation therefore is to create a model that mimics the real assembly line that inspired this research question originally. This generalization is a smaller-scale model with different tasks. The baseline model is built in Simio, an object-oriented discrete event simulation software. It consists of a source that creates entities according to the takt rate, three parallel rate stations that hold an entity for three takts each, followed by a sequence of three pulsing positions and finally a sink. Pulsing is a process that fires at the takt time moves an entity from one location to the next in the sequence. It pulls from the end of the line first, and vacancies are filled with entities from the previous station in the sequence such that once the system has been warm-started it has a constant WIP. Entities are created with inter-arrival time equal to takt rate and are pulled through the assembly line. The parallel rate stations are entered cyclically so that each entity remains there for 3*TaktTime and at any point in time each parallel rate station is occupied by an entity. There are no buffers in this system, but the number of buffers needed should be similar to the total traveled work divided by the takt time. This will be calculated deductively as part of the analysis section. Operators are also present in the model, with each station having its own group of operators that travel to one of three finite-space nodes where they are required to perform a task.

A small-scale assembly line will be modeled with three parallel rate stations followed by three sequential pulsing stations that process entities for a single takt. See Figure 2 for representation.

In addition, precedence constraints will be implemented alongside minimum operator constraints and zone capacity constraints.

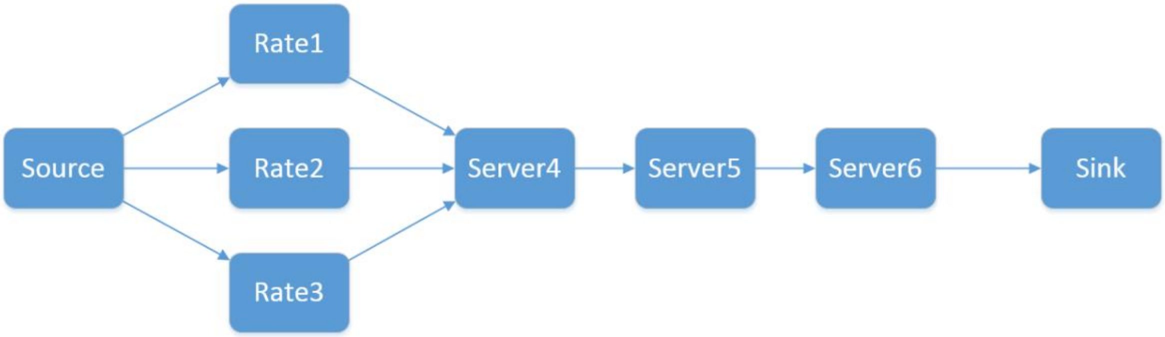


Figure 2: Small-Scale Representation of an Aircraft Assembly Line

Tasks with deterministic durations were inserted at each position with a default precedence (Figure 3) and a required number of operators to complete the task. This is the baseline model from which experiments will be run and variables/constraints can easily be changed. The precedence is read in Simio in the form of a data table, with the task precedence method defined by immediate successors. In this way, Simio determines which task to perform next by tracing back through the successors to find which ones are eligible to begin. Table 2 is a table representation of precedence that is inserted into Simio. Not shown in the table are the distributions for durations, operator requirements for each task, and the node where the work from the task is to be performed.

Table 3 states the independent and dependent variables in the model for each stage of simulation.

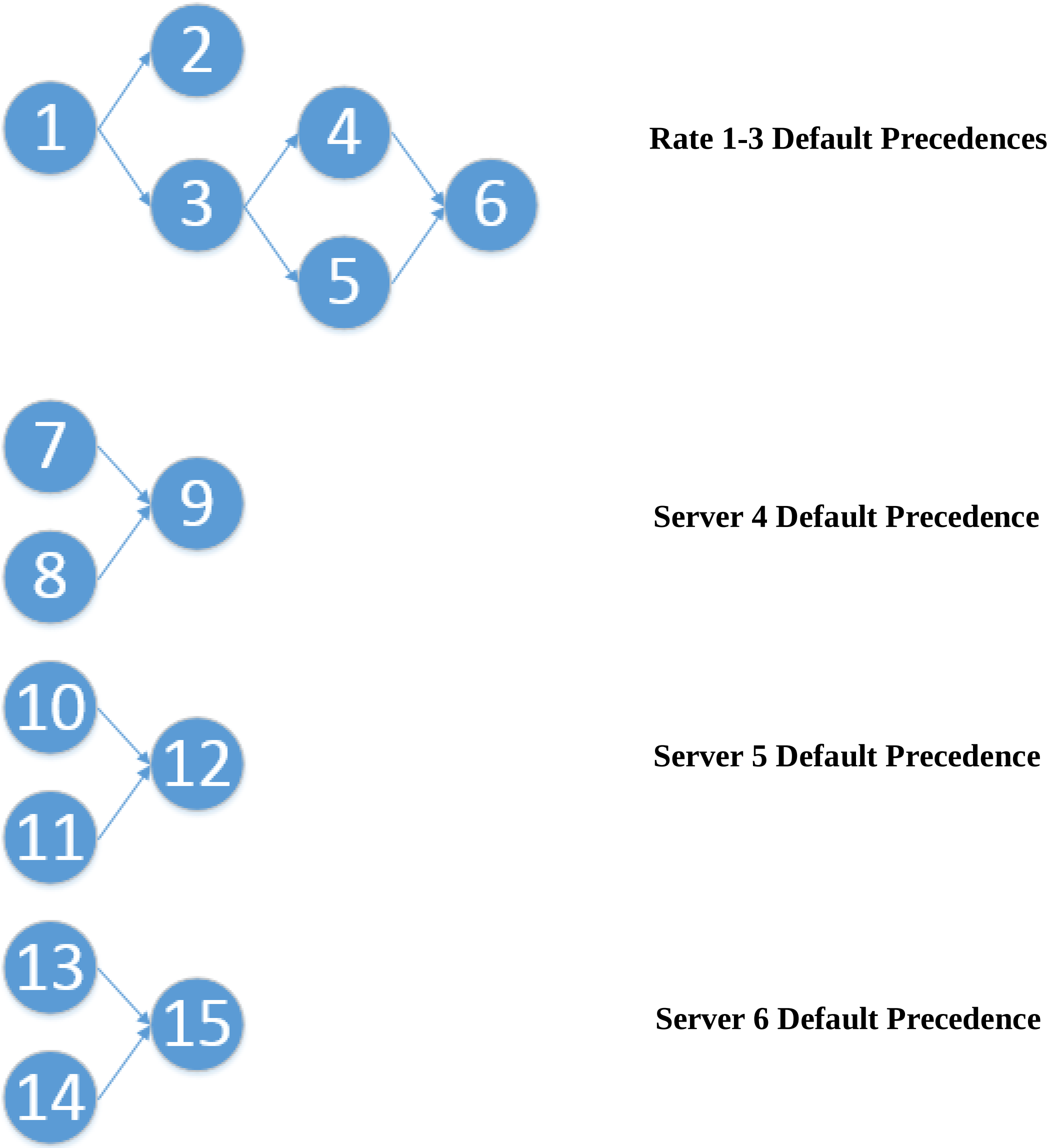


Figure 3: Default Precedence Diagram

Table 2: Simio Precedence--Table Representation of Default Precedence

Station	Task ID	Immediate Successor
Rate 1-3	1	2,3
	2	
	3	4,5
	4	6
	5	6
	6	
Server4	7	9
	8	9
	9	
Server5	10	12
	11	12
	12	
Server6	13	15
	14	15
	15	

Table 3: Variable Type by Stage

	Stage 1	Stage 2
Precedence	Independent (Fixed)	Independent (Variable)
Takt time	Independent (Fixed)	Independent (Fixed)
# Operators per station	Independent (Variable)	Independent (Fixed)
Node Capacity	Independent (Variable)	Independent (Fixed)
Operator/Node Utilization	Dependent	Dependent
Traveled Work	Dependent	Dependent

3.2 Verification

To verify the system works as it should, stochastic task durations are substituted for the deterministic ones. These durations are drawn from distributions that have the same theoretical means as the deterministic ones they replace. The distributions are randomly assigned from a list of common manufacturing/assembly task duration distributions (Normal, Uniform, Pert, Triangular). The Normal distribution is truncated below to prevent non-positive durations and above to ensure the mean is the same as the deterministic case. Similarly, Uniform, Pert, and Triangular distributions are all strictly positive and symmetric about the deterministic mean. Verification is accomplished by simulating multiple replications of experiments and comparing the individual task times as well as the total time in system for the average entity.

A screenshot of the static-state of the model is shown in Figure 4. This shows how the entities are created and enter the decision node before selecting the open Rate server. When they leave the rate server after three takts, they travel to the buffer node for an instant before moving into the sequence of three servers for one takt each. After an entity's sixth takt in the system, it moves from Server 6 to Sink 1 where it is destroyed, which corresponds to the completion of production for that entity.

This model will allow for the modification of number of operators, zone capacity, and precedence constraints. The output from simulation will provide answers to each of the research questions.

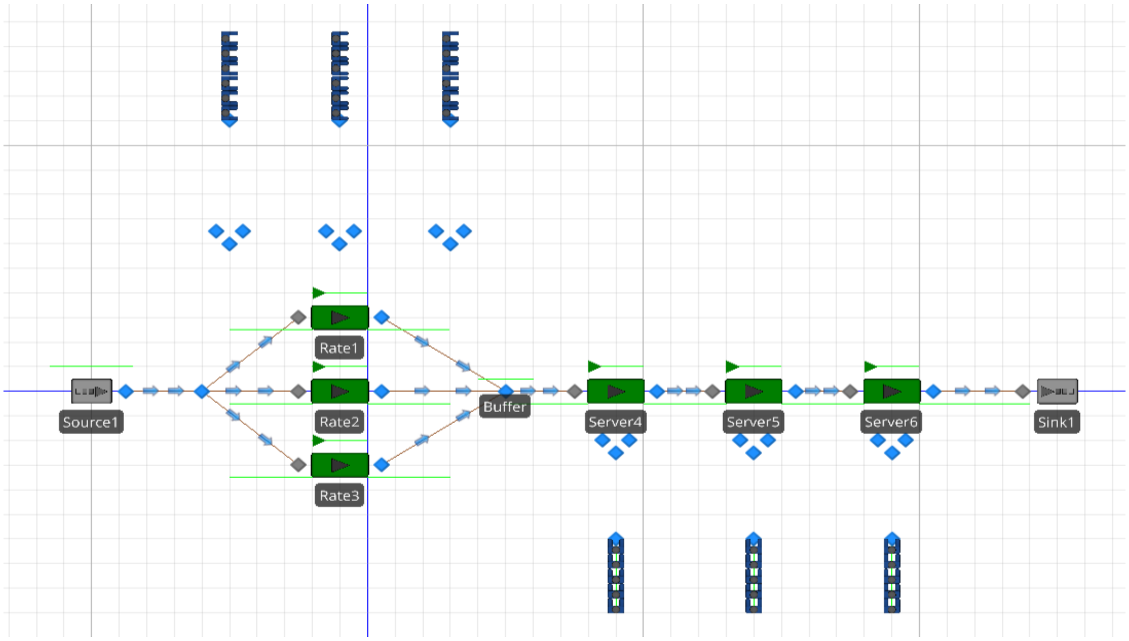


Figure 4: Simio Screenshot of Static State, Arrows Represent Direction of Flow

4. DESIGN OF EXPERIMENTS

Experiments were designed to test the model's performance subject to changes in constraints. Due to limitations in Simio, a two-stage approach was utilized. The first stage was designed to provide the configuration of number of operators and node capacity constraints. Then these configurations were implemented in the interactive view of the model where task logs were populated and a more precise calculation of traveled work could be captured.

4.1 Stage 1

Initially, the Simio add-in *OptQuest* was used to determine an appropriate number of operators per station and node capacities for each station. OptQuest takes a combinatorial approach to finding the best configuration for the model subject to some initial responses. These responses were worker utilization and hours of interrupted work. When OptQuest finished, the best configuration of number of operators and capacity for each node can be determined by examining the response variables' percentiles with equal weight and selecting the highest sum. Stage 1 results will answer RQ1's question about constraint changes and effect on system performance.

4.2 Stage 2

Next, the Stage 1 configurations were implemented in the interactive mode of the model. Each of the three different precedence types are simulated in experiments. The experiments here were run using 10 independent trials, running until 100 entities were destroyed for each configuration/precedence pair (three pairs). The logs and outputs from each trial were aggregated and conclusions drawn from its content.

Most notably, traveled work was calculated from two sources. The first was the captured interrupted task time. The tasks that were in process at the end of a takt give the remaining delay for the task's stochastic duration. The second is tasks that never started before the takt expired. For these tasks, we sample from their distributions to determine how long their delay would be in traveled work. These were each captured for every station a given entity passes through and multiplied by the number of operators at the station and the average utilization of the station's associated workers. This relationship is summarized below, where *i* is a particular station in the set *I* with workers for an entity *j* passing through, and *U* is the set of tasks at *i* that are not finished in the takt time. *Hours* is the number of hours remaining for the triple *i,j,k.* that is, (station, entity #, task #).

$$WorkerHrsRemaining_{i,j} = \sum_{k \in U} (Hours_{i,j,k} * Req.Workers_{i,j,k})$$

$I = \{Rate1, Rate2, Rate3, Server4, Server5, Server6\}$

$U \subseteq \{1,2,3, \dots ,14,15\}$

$j \in \{1,2,3, \dots\}$

To estimate the amount of *time* it will take to complete this amount of work out-of-station, the WorkerHrsRemaining is divided by the worker utilization percentage at station i and the number of workers at station i (this is simply *four* for all stations).

$$EstTraveledWorkHrs_{i,j} = \frac{WorkerHrsRemainaing_{i,j}}{WorkerUtil_i * \#Workers_i}$$

Again, traveled work is comprised of both interrupted tasks and tasks that never started. This is one reason that the traveled work time is an estimate: tasks that never began do not draw a duration from their stochastic distribution during the simulation run, rather we take an independent sample of that distribution in post-processing and add it to the traveled work calculation. The second reason that the traveled work time is an estimate is because the precedence is no longer intact once an entity leaves the station. The utilizations from each station were used to give an approximate measure of the effect that a precedence and node constraints had on processing time in-station. These constraints are responsible for less-than-full utilization of workers. The utilization of workers is a direct result of precedence and node constraints, so dividing by the observed utilization rate imitates their effect for the purposes of finding how long the traveled work should take to perform. The total traveled work for an entity *j* that enters the sink at the end of the line is the sum of each station's traveled work for the entity,

$$TotalEstTraveledWork_j = \sum_{i \in I} EstTraveledWorkHrs_{i,j}$$

The theoretical makespan in a takt system is the number of stations times the takt time. Since $TotalEstTraveledWork_j$ estimates how long all of the traveled work should take to be completed, it also represents the delay that should be expected for entity *j* beyond the theoretical makespan. Stage 2 results will answer RQ2, RQ3 after post-processing to include the traveled work calculations as defined above.

5. RESULTS

5.1 Stage 1 Results

The results from Stage 1 are the configurations that are to be used in the interactive mode during Stage 2. The sum of the highest percentile performance between the two responses Worker Utilization (higher utilization means higher percentile) and Interrupted Work (lower sums mean higher percentile) is from a configuration of 4 operators per station with each node at every station having capacity 3. This comes from running the experiment combinatorically and testing each configuration for the two responses. Each node capacity was given values from 3 (minimum needed for some tasks) to 6, which was thought to be a reasonable upper bound. The number of operators per station ranged from 3 to 7. Due to the huge number of combinations, some values were thrown out early because the responses were very bad—for example, 3 operators per station was not at all competitive, while 6, 7 or 8 operators per station did not improve responses at all. This saved time in completing the exhaustive run.

Changing node capacities and number of operators had an effect on utilization and interrupted time of tasks. There was a limit to their benefits, and after a certain point increasing both did not improve performance. Only a 3-operator capacity per node was necessary, but the increase from 3 operators to 4 per station had a large impact on performance. This answers RQ1 regarding the effects of constraint changes on performance, even though performance is not specifically traveled work.

5.2 Stage 2 Results

Stage 1 configuration was implemented into the interactive mode where logs can be generated. This provided the task log from which traveled work times can be calculated and a comparison of precedences is possible. Data from task logs is gathered and the formulas in the Design of Experiments section are used to calculate total expected traveled work for the following results:

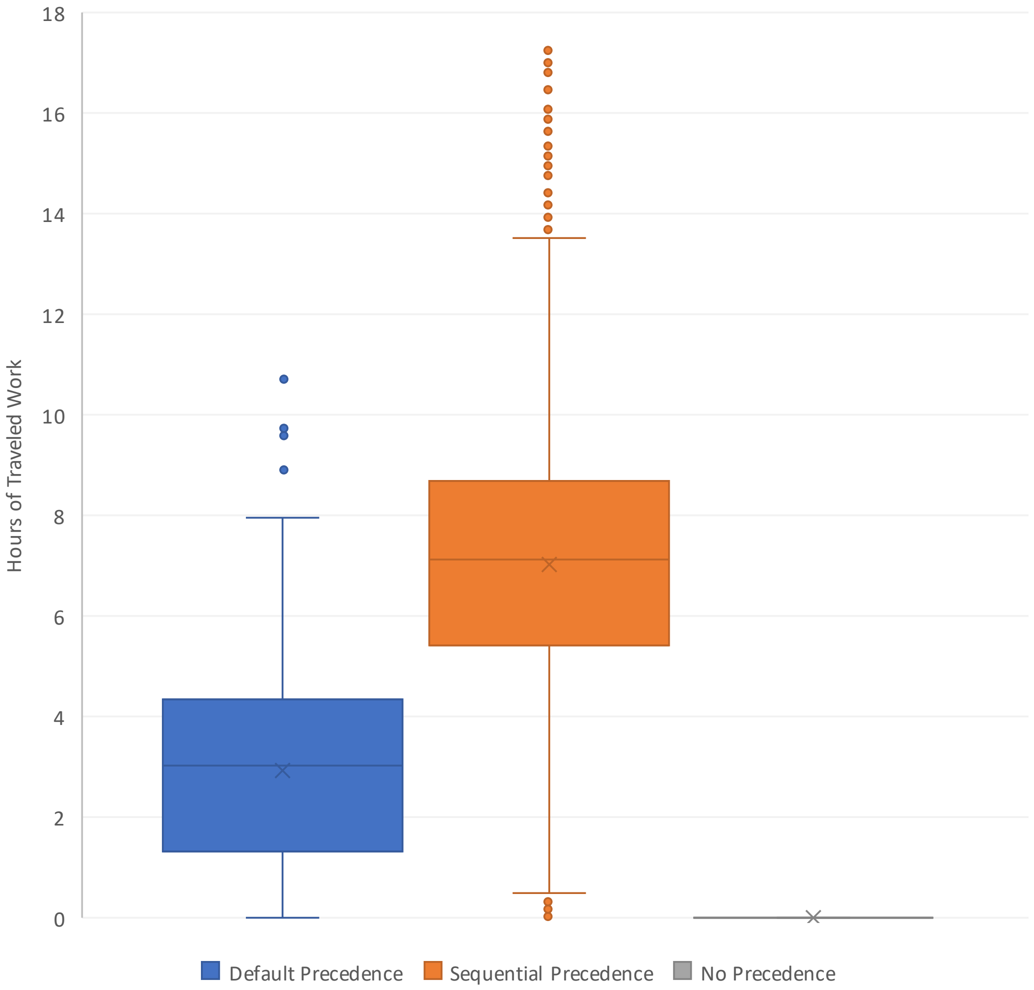


Figure 5: Total Estimated Traveled Work by Precedence

A zero traveled work time means that an entity has finished all required tasks in the time allotted for the model with the specific precedence. The results from Stage 2 answer RQ2, which is concerned with the effect of changes in precedence on performance. Here we use the traveled work calculation to better estimate performance. Where traveled work becomes problematic is in the larger sums, and Table 4 gives a closer look at the upper ends of the distribution. In parentheses is the decrease in average throughput if the system waited for all work to be completed in-station for each entity (no traveled work). This partially answers RQ3, where traveled work directly decreases throughput.

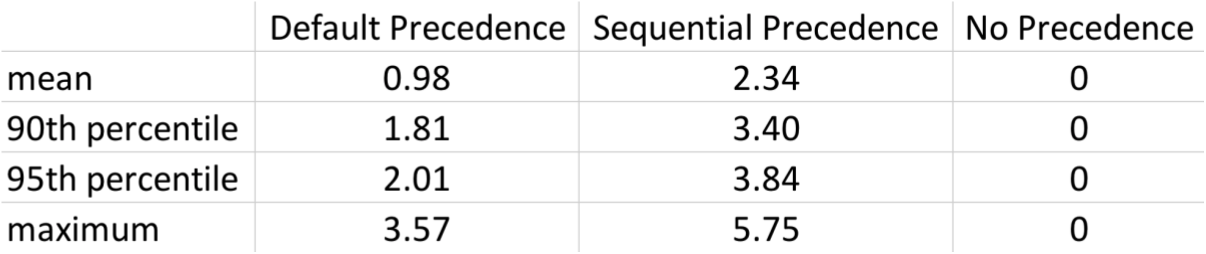
Table 4: Traveled Work by Precedence Type

	Default Precedence	Sequential Precedence	No Precedence
mean	2.94 (-14%)	7.02 (-28%)	0 (0%)
90th percentile	5.43 (-23%)	10.21 (-36%)	0 (0%)
95th percentile	6.04 (-25%)	11.52 (-39%)	0 (0%)
maximum	10.72 (-37%)	17.25 (-49%)	0 (0%)

Simply by dividing the results of Table 4 by the takt time (in this case three hours), we can give the approximate the number of buffer stations needed to ensure the completion of entities in the allotted time (Table 5). Table 5 equivalently shows the increase in makespan—measured in number of additional stations needed. Increasing makespan along with the number of buffers maintains throughput of the system because one entity is still completed each takt period. This completes the answer to RQ3, analyzing the effect traveled work has on throughput by adding necessary buffers.

The benefit of looking at traveled work this way is that one can see how often traveled work would be problematic if some number of buffers were added to the station. For example, using the default precedence and adding two buffers to the system, one can reasonably expect to only encounter end-of-line traveled work once in about every 20 entities.

Table 5: Number of Buffer Stations Needed by Precedence Type



	Default Precedence	Sequential Precedence	No Precedence
mean	0.98	2.34	0
90th percentile	1.81	3.40	0
95th percentile	2.01	3.84	0
maximum	3.57	5.75	0

Table 4 and Table 5 give two different ways to determine the effect of precedence changes on throughput.

5.3 Discussion

As mentioned in the introduction, not all precedences are equivalent. A hard precedence is what most strive for, but often, some of “what should be done” creeps into “what must be done”. Intuitively, the less constrained the problem, the better the system should perform. In highly nonlinear and complex systems, further constraining a problem (operator constraints, node constraints, strict precedence constraints) has the potential to exacerbate performance problems disproportionately. A set of precedence constraints cannot be analyzed in a vacuum, but simulation does a good job highlighting some of the issues that can arise from implementing one version or another of precedence in the presence of other constraints.

For the assembly line in question, there are three options to complete traveled work:

1. Introduce buffer stations to complete traveled work
2. Temporarily suspend the pulse to allow entities to complete tasks at each station
3. Increase takt time

Each have their concerns. Depending on a decision maker’s tolerance for risk and the costs associated with adding buffers/missing delivery deadlines, one may be preferred over the others.

Additionally, a closer look at the precedence could uncover instances of soft constraints where removing the constraint from the task sequence improves performance. If this is the case, perhaps the system could even be condensed to fewer stations!

Option 2 is different from Option 3 because it disregards the tempo aspect of takt time. In some situations this could be beneficial (a rare entity that takes longer to produce can be accommodated once), but if entities of this variety are more common, it might be better to increase takt time and allow entities more time to complete tasks. This decision is dependent on task variation and frequency of large sums of traveled work.

6. CONCLUSION

6.1 Summary

Although assembly processes with precedence constraints or takt analysis are not new to the field, literature containing the trio is scarce. The real-life application of this problem makes for an interesting study on the effects of precedence constraints within a takt-system. The small-scale model built for analysis in this paper captures some of those intricacies with a focus on feasibility for businesses—minimizing number of workers while meeting production demand and constrained by finite floor-space and node capacities.

Traveled work is defined and a method of estimating total traveled work time is given. The total traveled work time is then transformed to demonstrate impact on throughput, and ways to reduce traveled work are proposed.

6.2 Conclusions

- Changes to zone capacity constraints and operator constraints influence performance, but their effect has an upper bound and is limited below by the minimum operator requirements of tasks
- Simulation as a tool to analyze sensitivity to several constraints in a complex system
- Method of estimating traveled work from interrupted tasks and tasks not begun
- Traveled work as a result of changes in precedence
- Demonstration of high sensitivity to precedence constraints
- Traveled work is a part of throughput; in a takt-system traveled work is the difference between theoretical throughput and observed throughput

- Suggestions to common problems involving precedence constraints in assembly lines

Each of the three research questions has been answered as well,

1. Changes to operator and node constraints affect system performance measured by operator utilization and interrupted work. This was demonstrated in Stage 1.
2. Changes in precedence were shown to affect performance of the system, as defined by amount of traveled work. This was demonstrated in Stage 2.
3. Traveled work is scheduled in-station work that is not completed when an entity is pulsed. Traveled work must be completed for an entity to leave the system. All else equal, traveled work extends the makespan of entities, thus decreasing throughput. However, throughput can be maintained if buffers are added to the system, and the makespan still increases as all traveled work is completed.

6.3 Limitations

During validation it became evident that the system could benefit from line-balancing. This is made less significant, however, because a perfectly balanced system still has to anticipate traveled work and account for traveled work in its balancing. The chosen simulation software, Simio, only generates logs in the interactive mode. This makes both experiments and the interactive mode less useful and is the reason for a two-phase approach. Limitations in time to completion made scaling-up the model infeasible. This would have given a more well-rounded analysis with less variance.

6.4 Future Work

Expanding the simulation to contain more stations and tasks and a more intricate precedence could give a lower variance than the configurations of stations/tasks/precedence chosen by the author. Where this thesis sampled from the distributions for tasks that never began in post-processing, generating stochastic durations for each such task is possible within Simio and is a more stable approach.

REFERENCES

Battaïa, O. and A. Dolgui (2013). "A taxonomy of line balancing problems and their solution approaches." International Journal of Production Economics 142(2): 259-277.

Capaccio, A. (2017) "F-35 Program Costs Jump to $406.5 Billion in Latest Estimate."

Duanmu, J. and K. Taaffe (2007). “Measuring manufacturing throughput using takt time analysis and simulation.” 2007 Winter Simulation Conference, Piscataway, NJ, USA, IEEE.

Duanmu, J. and K. Taaffe (2012). "Production capabilities using takt times, requirements analysis and simulation." International Journal of Industrial and Systems Engineering 10(2): 197-216.

Framinan, J. M., P. L. Gonzalez and R. Ruiz-Usano (2003). "The CONWIP production control system: review and research issues." Production Planning & Control 14(3): 255-265.

Garey, M. R. (1973). "Optimal task sequencing with precedence constraints." Discrete Mathematics 4(1): 37-56.

Hoffmann, T. R. (1963). "Assembly line balancing with a precedence matrix." Management Science 9(4): 551-562.

Hopp, W. J. and M. Roof (1998). "Setting WIP levels with statistical throughput control (STC) in CONWIP production lines." International Journal of Production Research 36(4): 867-882.

Macaskill, J. (1972). "Production-line balances for mixed-model lines." Management Science 19(4-part-1): 423-434.

McMullen, P. R. and G. Frazier (1998). "Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations." International Journal of Production Research 36(10): 2717-2741.

Mendes, A. R., A. L. Ramos, A. S. Simaria and P. M. Vilarinho (2005). "Combining heuristic procedures and simulation models for balancing a PC camera assembly line." Computers & Industrial Engineering 49(3): 413-431.

Millstein, M. A. and J. S. Martinich (2014). "Takt Time Grouping: implementing kanban-flow manufacturing in an unbalanced, high variation cycle-time process with moving constraints." International Journal of Production Research 52(23): 6863-6877.

Ríos, J., F. Mas and J. Menéndez (2012). “Aircraft final assembly line balancing and workload smoothing: a methodological analysis.” Key Engineering Materials, Trans Tech Publ.

Sandanayake, Y. G., C. F. Oduoza and D. G. Proverbs (2008). "A systematic modeling and simulation approach for JIT performance optimisation." Robotics and Computer-Integrated Manufacturing 24(6): 735-743.

Seppänen, O. (2014). “A comparison of takt time and lbms planning methods.” 22nd Annual Conference of the International Group for Lean Construction: Understanding and Improving Project Based Production, IGLC 2014, June 25, 2014 - June 27, Oslo, Norway, The International Group for Lean Construction.

Wu, K. and K. Hui (2008). "The Determination and Indetermination of Service Times in Manufacturing Systems." IEEE Transactions on Semiconductor Manufacturing 21(1): 72-82.

Zhu, X., S. J. Hu, Y. Koren and S. P. Marin (2008). "Modeling of Manufacturing Complexity in Mixed-Model Assembly Lines." Journal of Manufacturing Science and Engineering 130(5): 051013-051013-051010.