Fall 2004

Bottleneck Analysis of CPortS

SreeKalyana Chakravarthy Kajuluri
Old Dominion University

BOTTLENECK ANALYSIS OF CPORTS

by

SreeKalyana Chakravarthy Kajuluri
Bachelor of Engineering, December 2001
Bangalore University, Bangalore, India

A Thesis submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
December 2004

Approved by:

James F. Leathrum, Jr. (Director)

Roland R. Mielke (Member)

Frederic Mckenzie (Member)

ABSTRACT

BOTTLENECK ANALYSIS OF CPORTS

SreeKalyana Chakravarthy Kajuluri
Old Dominion University, 2004
Director: Dr. James F. Leathrum

CPortS is a transportation logistics simulation that models the flow of military cargo through a seaport and the interaction of the cargo with the port resources and infrastructure. It provides information about the seaport's capabilities, how the cargo has been handled, how many days the cargo took to clear a particular port area, and the overall throughput of the seaport. The model is highly data intensive since it models the huge traffic in a real seaport.

Bottlenecks reduce system performance. Systems that are traffic intensive or simulations of systems, which are data intensive, encounter bottlenecks, which reduce their performance. In order to improve the system performance it is necessary to study the cause of the system bottlenecks and find a way to overcome them. This thesis provides support for Bottleneck analysis of CportS. The thesis stresses mainly on the "Shifting Bottleneck Detection Method" which considers system bottlenecks to be dynamic (shifting from one system component/aspect to another) rather than being static. In addition, a comparative study of the various bottleneck detection strategies will be made by applying them to the CPortS model to uphold the dominance of the shifting bottleneck detection method.

The following study deals in depth with the various bottleneck detection strategies and comes out with a suitable bottleneck detection methodology applicable to the CPortS model. It concludes with the testing results for the proposed methodology.

This thesis is dedicated to my family members.

Acknowledgements

I would like to thank my Advisor Dr James Leathrum Jr. for his guidance and support. I feel lucky to have such a helpful advisor. I would also like to thank my first advisor Dr Rasha Morsi. I thank the members of my thesis committee Dr Roland Mielke and Dr Rick McKenzie for sparing their precious time for me.

I would also like to thank Mr. Joseph Joines and Military Traffic Management Command Transportation Engineering Agency (MTMCTEA) for funding my research work. I would also like to thank my colleagues at the Virginia Modeling, Analysis, and Simulation Center (VMASC), Mr. Taylor Frith, Saurav and Reejo for helping me out when I ran into problems during my research work.

I would like to thank Dr Amin Dharamsi and Dr Sacharia Albin, my department program directors for their advice and support.

Last but not least I would like to thank my parents and my sister for their support and encouragement throughought my life.

Table of Contents

List of Figures and Tables

[1]Chapter 1

Introduction

1.1 Introduction

System bottlenecks reduce performance, either qualitatively or quantitatively. Bottleneck analysis supports attempting to reduce their impact on the overall system performance [2]. A simulation of a system can be used to analyze the potential bottlenecks in the actual system being investigated. The methodologies used to handle the bottlenecks in the simulation model can then be effectively applied to the actual system being considered. This thesis presents a methodology to analyze the bottlenecks in the CPortS model.

CPortS is a transportation logistics simulation that models the flow of military cargo through a seaport and the interaction of the cargo with the port resources and infrastructure. It provides information about the seaport's capabilities, how the cargo has been handled, how many days the cargo took to clear a particular port area, and the overall throughput of the seaport [1]. Generally, the seaport model has to handle large amounts of cargo, which requires considerable amount of resources otherwise large amount of delays will be incurred in clearing the cargo from the port areas. It is very important to detect the events that cause these delays, since the cargo is military owned. Since the simulation deals with military cargo, delays caused by bottlenecks can be very influential in determining the outcome of the military operations. This presses the need for a methodology to detect these bottlenecks and thus justifies the work done in this thesis.

[1] The reference model for this work is "A Practical Bottleneck Detection Method," *Proceedings of the 2001 Winter Simulation Conference"* pp 949-953

The objective of this thesis is to provide a methodology for determining such information from the CPortS model and to identify the bottlenecks. Once the bottlenecks are detected, it becomes easier to improve the events, which delay the cargo clearance process by pumping up the capabilities of the port areas where the delay occurs, and hasten the process of cargo clearance.

1.2 Approach

Bottlenecks in a system are caused by over utilization of resources, which can be observed by large activity times for events utilizing these resources. Finding out a prime bottleneck area and eliminating the cause does not ensure improved throughput since after improving the primary bottleneck another component of the system may become the primary bottleneck. Eliminating a bottleneck can result in the identification of a new bottleneck, i.e. the bottleneck “shifts” to another area. The final bargain from such a scenario would be to extract as much throughput as possible within the limitation of available resources and infrastructure. The shifting bottleneck methodology primarily uses the above-mentioned shifts as the basis of determining the bottlenecks. This methodology considers a bottleneck to be temporary (i.e. dominant only over a limited period of time). By considering the interval of time over which the bottlenecks are active (without shifting) the shifting bottleneck methodology ranks them as primary and secondary bottlenecks. This successfully pinpoints the primary bottleneck since longer the time for which a bottleneck is active the larger will be its influence on the overall throughput of the system. This way the shifting bottleneck detection strategy locates the major bottlenecks in a system.

1.3 Overview

Chapter 2 introduces the CportS model by discussing the various modes of operation in the model followed by the discussion of the port areas in the model. This chapter familiarizes the reader with the details of the CportS model and the associated port areas.

Chapter 3 introduces system bottlenecks and techniques to detect them. This chapter discusses in detail the shifting bottleneck methodology and compares it with other bottleneck detection strategies, identifying the advantages and disadvantages associated with each of them. It ends with a briefing of the major research carried out in detecting bottlenecks.

Chapter 4 considers an example scenario of a bicycle production system. This chapter applies the shifting bottleneck detection methodology as well as other methodologies to this example system. The results obtained thus will be used to establish the superiority of the shifting bottleneck methodology over the other bottleneck detection strategies. The chapter ends by performing sensitivity analysis to validate the results obtained.

Chapter 5 shows the results obtained by applying the bottleneck detection strategy to a model scenario provided by MTMCTEA. These results will be used to perform a sensitivity analysis to determine the accuracy and applicability of the shifting bottleneck detection strategy to the seaport simulation CPortS.

Chapter 6 ends the thesis with concluding remarks as well as with suggestions that can enhance and improvise the work done in this thesis.

Chapter 2

CportS Overview

2.1 Introduction

This section gives an introduction to CPortS, the seaport simulation model that is the main focus of this thesis. In addition to the above, this section discusses the characteristics of the simulated port areas as well as the flow of cargo through these port areas.

2.2 Port Simulation (CPortS)

CPortS is a discrete event simulation that models the military operations at a seaport [1]. It simulates the flow of military cargo and equipment through the seaport and provides several important details like the cargo arrival time, cargo departure time at a particular area as well as the quantity of cargo during a stipulated time textually as well as graphically. This data can be used to obtain such information as the cargo inflow as well as outflow from a particular port area. CPortS has two modes of operation: Port of Embarkation (POE) and Port of Debarkation (POD) [1]. The debarkation mode will be the emphasis of this thesis.

The debarkation mode handles the military cargo arriving at the port by ship. The major activities include the unloading of the ship, staging activities and clearance. The unloading activities include all processes needed to offload the vessels that arrive at the berths and transport the cargo to the staging areas. Finally, the clearance activities allow the cargo items to be moved from the port on to the highway or railway infrastructure.

This activity includes the loading of commercial highway assets such as flatbed trucks and chassis. It also includes the loading of railway assets such as trains made up of flatcars and boxcars.

2.2.1 The CPortS Process Model

The CPortS model is process oriented because all the cargo that enters into a port area has to pass through several processes associated with the port area [10]. As can be seen in the macro-level process flow in Figure 2.1, there are several port areas as well as a large number of resources utilized at these port areas. The port areas are represented as ovals and the resources utilized for accomplishing the tasks (moving cargo, loading cargo, etc) at the port area are represented as rectangles. A brief description of the various port areas involved is presented to orient the reader to the problem being addressed. The following is a screen shot of figure 2.1.

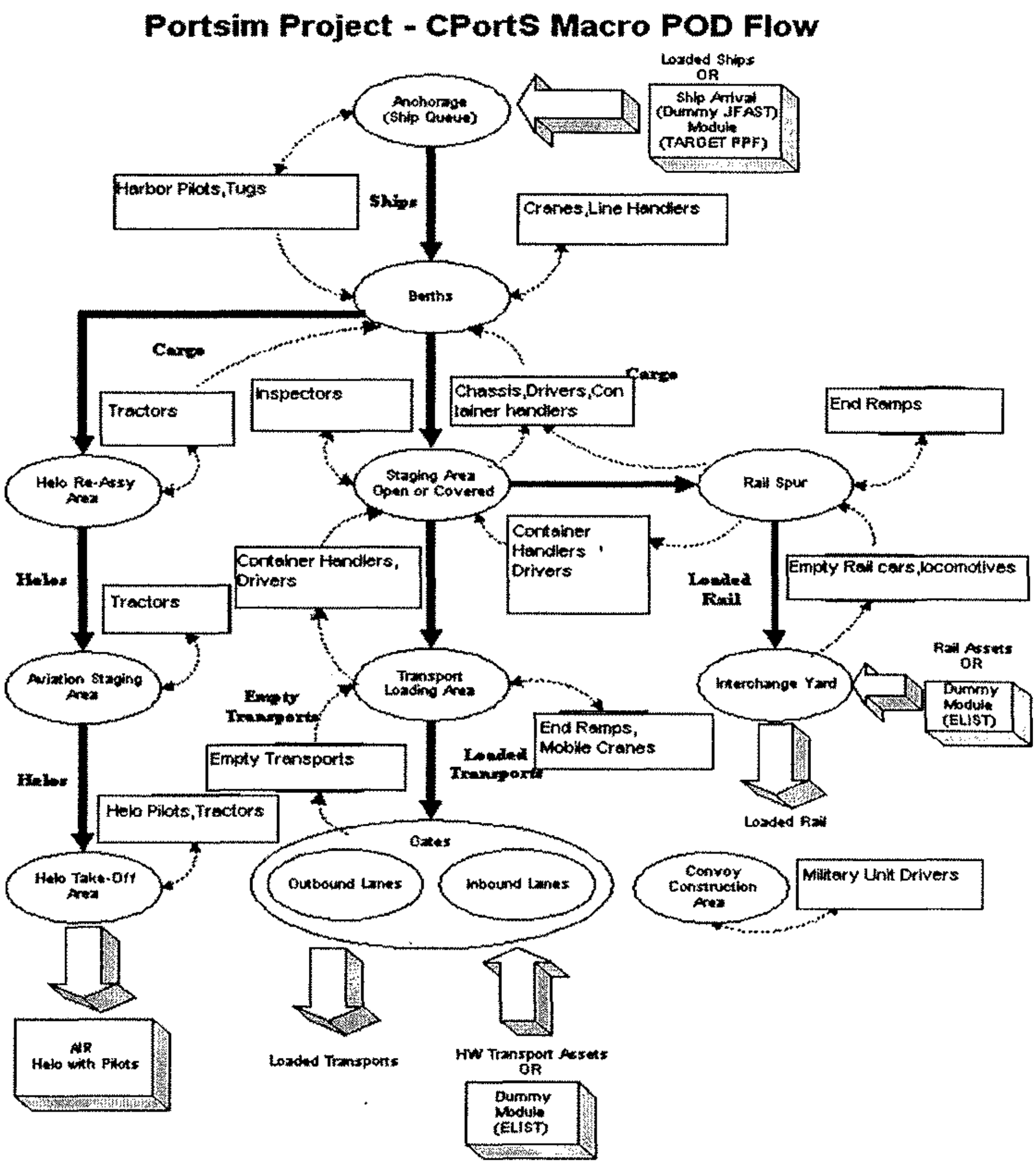

Figure 2.1: Macro flow of CPortS Model [10, 11]

(1) Anchorage Area: The portion of the seaport wherein the ships wait for berths to become available. Tugs and harbor pilots are the resources utilized in the anchorage area. They are required to move the ships to and from berthing.

(2) Berthing Areas: This is the port area where the ships are docked for offloading. The number of berths is an important parameter, since the unavailability of sufficient berths of required depth and length can be a considerable problem and may cause a large number of ships to be stranded in the anchorage area. The resources associated with the berthing areas are cranes and line handlers, which are required to lift the cargo from the loaded ships onto the berthing area. Other resources like chassis, drivers, container handlers and tractors are used to carry the cargo to their destinations (either staging area or helicopter reassembly area).

(3) Staging area: The port area wherein the cargo offloaded from the ship is stored until resources become available to move the cargo out of the port. Cargo inspection is also performed in this part of the port. The input resources associated with this area are chassis, PSA personnel and container handlers. Characteristics of staging include the number of staging areas available, the number of staging areas utilized, processing times, cargo types accepted, transport modes accepted, the total staging capacity, and the utilized capacity. These parameters can have a significant influence on the throughput.

(4) Transport Loading Area: The port area from which cargo is loaded onto highway transports. The resources associated with this area are end ramps and mobile cranes, which are used to load the transports with the cargo.

(5) Gates: The gates separate the seaport and the external world. All the cargo that leaves the port through highway must pass through the gates to exit the seaport. No specific resources are associated with the gates.

(6) Helicopter Reassembly Area: The helicopters are reassembled for take-off in this port area. The primary resource associated with all the helicopter related port areas is the tractor, which carries helicopter parts.

(7) Aviation Staging Area: The helicopters are staged for takeoff in this port area.

(8) Helicopter Take-off Area: The pilots move the helicopters to their destination area from this port area.

(9) Rail Spur: The cargo is loaded onto rail cars in this port area. The resources associated with this port area are end-ramps, rail cars and locomotives. These are used to load and move the cargo onto the rail interchange yard area.

(10) Rail Interchange Yard: The port area wherein rail cars are held, waiting to be moved out of the port.

(11) Convoy Construction Area: The port area from which the cargo scheduled to leave is assembled. Military unit drivers are utilized in this port area.

A typical cargo flow through the CPortS simulation can take the following course: a ship arriving with cargo into the seaport waits in the anchorage area for the availability of a suitable berth and the resources needed to move it to the berth area. From the berth the helicopter related cargo is routed to the helicopter areas. The remaining cargo is moved into the staging areas. From the staging areas the cargo will be routed either to the truck loading area, convoy construction area, or to the rail spur area (different modes of transportation). The cargo scheduled to leave is passed on to the convoy construction area for assembling before it leaves onto the road.

Chapter 3

Survey of Bottleneck Analysis strategies

3.1 Introduction

This chapter discusses bottlenecks, bottleneck analysis and detection strategies followed by a list of applications in which these strategies are being applied.

3.2 Bottlenecks

A system is only as strong as its weakest point [3]. For making the system foolproof it is necessary to identify the weak points in the system and find out ways to address them. For the purpose of this work a bottleneck is defined as a component in the system, which undermines the performance of the entire system not only by its inferior performance but also by influencing the other components in the system, which directly or indirectly depend upon it for their functioning.

There are two major causes, which force a component in a system to become the bottleneck. One of these causes is the lack of resources in that particular component and the other cause is the component's saturation point of performance. Finding the principal bottleneck among several bottlenecks in the system is the main objective of this thesis work. Systems can be classified either as well balanced or unbalanced [3]. These are described below.

3.2.1 Well-balanced system

A well-balanced system is characterized by components with capabilities (performance levels) that are equal/nearly equal (i.e. with slight differences). This can be visualized from figure 3.1.

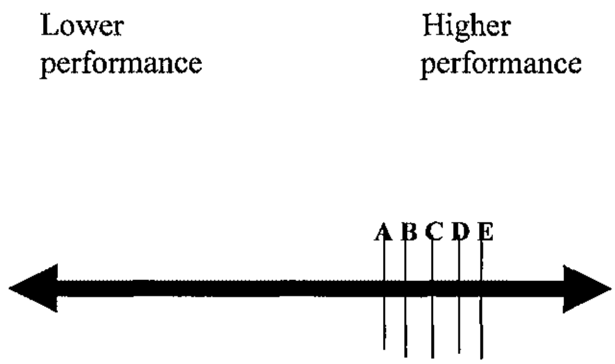

Figure 3.1: A well-balanced system.

As can be seen from the diagram above, the components A, B, C, D, E have nearly equal capability or potential. But if we rank these components according to their performance (by considering the slight differences that exist amongst their performance levels), component 'A' will be the bottleneck. If we bring component 'A' to the same performance level as component 'E', only a slight increase in the performance of the overall system can be achieved. This happens because the bottleneck now shifts to component 'B' which is now at lowest performance level. This game of shifting bottlenecks occurs until all components are brought to the same performance level. This is a well-balanced system since all components are within the same performance range.

3.2.2 Unbalanced system

In an unbalanced system the components have a wide gap in their performance levels. Bringing all the components to the same performance level can be quite costly (since it may require lots of resources for performance boost of low performance components). There is a greater disparity between the high and low performance components. The diagrammatic representation of an unbalanced system can be seen in figure 3.2.

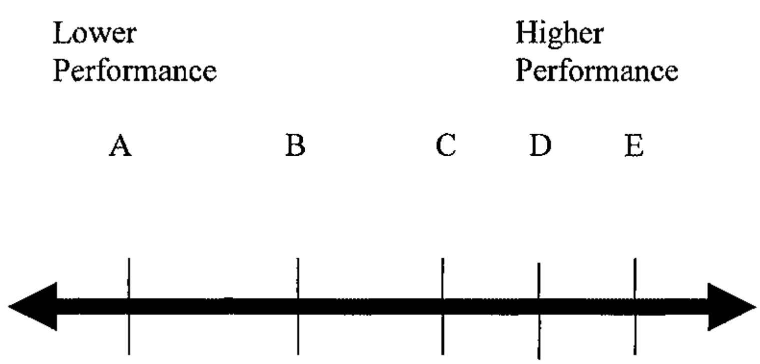

Figure 3.2: An unbalanced system

As can be seen from figure 3.2 the greater the disparity between the performances of various components the greater will be the effort required to bring all the components to the same level of performance, which can be quite a costly affair due to the amount of resources, time, and effort required.

CPortS can be categorized as an unbalanced system since there is a great disparity between the various components of the system. Components can be defined as port areas, like the anchorage, berthing, staging, and transport loading areas, do having varying capacities as well as performances, based upon their buffer sizes, processing speeds and the amount of resources (human as well as mechanical) available to them.

3.3 Bottleneck Detection Strategies

Several techniques for bottleneck detection have been proposed. Each technique has its advantages as well as disadvantages depending upon the type of system upon which the technique is applied. Some bottleneck detection techniques are preferred for networking and communication systems whereas some other techniques are apt for production lines and mechanical industries. All the bottleneck techniques are either based on "utilization time", "waiting time" or a "system theoretic approach" methodology [7]. The following gives an explanation of these methodologies as well as the successful techniques based upon these methodologies.

3.3.1 Bottleneck Detection using the utilization time

To detect the bottleneck in a system, the utilization method measures the percentage of time a component is active and then defines the component with the largest active percentage as the principal bottleneck since such a component has the least chance of being interrupted by other components in the system and also it dictates the overall throughput of the system. In the utilization based bottleneck analysis the utility of the component/machine is calculated by the following equation if the component is not functioning at its full capacity as follows: Utility =working capacity/ total capacity- Equation (3a).

If the component is functioning at its full capacity then the utility of the component is 1.

The bottleneck percentage of the component is then calculated by using the following equation: Utilization time=utility*component working time - Equation (3b).

This way the utilization method also captures the capacity of the component. The methods, which are based on the utilization method, therefore utilize even the capacity of the component also. Several bottleneck detection techniques, which are based on the utilization method, are as follows:

(a) Bottleneck Analysis of Queuing Network Models with Histogram-Based Parameters [4, 5]: This method is applicable for computer and communication systems where there are severe variations and uncertainties in the workload. In a conventional queuing network methodology, single values are accepted as model inputs and a single value is computed for each performance measure of interest. This methodology fails if uncertainties or variations in service demand exist since the single mean value calculated does not actually represent the uncertainty in the service demand. In order to overcome this disadvantage, a histogram-based methodology is used. The mean service demand (which can be considered as a parameter in the system) for each component in the system that exhibits variability or uncertainty is represented by a histogram, which consists of a number of intervals and an associated probability of occurrence. Each interval is a range of values; the parameter lies in this range with the specified probability of occurrence. In certain situations the workload may be characterized only by variability without any uncertainty. If the various mean values for a parameter take on a number of single point values, a histogram in which the intervals are of zero width is used. If the mean parameter values are close to each other and have nearly equal probability of occurrence they are clustered together to form a single interval. The number of such intervals as well as the interval durations in the histogram depends on the nature of the workload and the granularity of the measurement. In this way the variations/uncertainties are handled as an interval of values rather than a single value. The output from the histogram along with some specific mathematical proofs and lemmas can be used to generate interval matrices, which later pave the way for the bottleneck probability matrix, which helps in determining the system bottlenecks.

(b) Component cycle time divided by the total processing time

Another method for finding the bottlenecks in a system uses product of the component 'utility' and the ratio of the component cycle time divided by the total processing time. A system as mentioned earlier can be considered as a collection of components working together. Each component has some processing time or cycle time to work upon the unit to be processed. This cycle time is considered in this methodology. Here the problem is the variable cycle times. Components in a system having variable cycle times generate different performance measures, which makes it difficult to brand a particular component as the bottleneck since the component is already working at its peak efficiency and nothing can be done to improve its performance (a case of saturation) [6].

Several disadvantages are associated with the pure "utilization time" based bottleneck detection strategy. First of all, the utilization time bottleneck detection methodology fails to point out the real bottleneck when there are two or more components, which have nearly the same utilization time. The other disadvantage associated with this methodology is that it doesn't provide any insight regarding secondary bottlenecks as well as non-bottlenecks.

3.3.2 Shifting Bottleneck Detection using average duration of an active component

In this bottleneck detection strategy the component that has the maximum active period is considered to be the principal bottleneck. A set of states is initially defined for the components, which identifies whether the component is idle or active. At any given time a component can be either idle or active. Adding up all the active times for a component gives the total active time. In this manner the active times for all the components will be calculated and the component with the maximum active time is considered to be the principal bottleneck since this component is least likely to be disturbed by other components and in turn is most likely to dictate the overall system throughput. This can be analyzed mathematically as follows [2]:

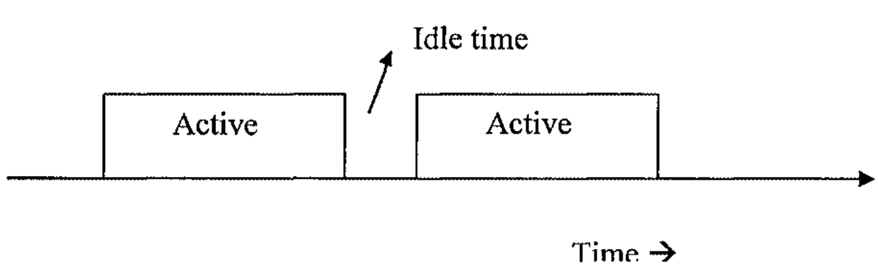

Figure 3.3: Active Inactive state diagram

The duration of active period's a_i for all components i is measured. This results in a set of durations Ai for each machine as follows:

$A_i = \{ a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5} \ldots\ldots\ldots\ldots\ldots a_{in} \}$ (equation 3.1)

The basic premise upon which the shifting bottleneck detection method works is the active duration of the components of the system. In a complex system a component will have several working states and will process multiple parts at the same time. In such cases finding the active time is not easy. The mean active time for the component will be useful in such cases. Similarly, if the active periods are estimated (inaccurate due to a highly complex system) values then the confidence intervals as well as the standard deviations of the respective data can add validity to the results obtained by such an analysis. The mean active time for a component i can be calculated as:

$$\overline{a_i} = (a_{i1} + a_{i2} + a_{i3} + \ldots + a_{in})/n \qquad \text{(equation 3.2)}$$

Simulation data cannot be assumed to be independent, and subsequently it is difficult to calculate a confidence interval of a simulation measurement. Subsequently additional techniques as for example batching have to be used to establish a valid confidence interval. The times between inactive periods are approximately independent of each other, and subsequently the average active durations are also approximately independent of each other. This allows a straightforward calculation of a standard deviation as shown in Equation (1) and a confidence interval as shown in Equation (2), estimating the accuracy of the bottleneck measurement. Therefore it is easy to determine the accuracy of the bottleneck detection.

$$\sigma_i = \sqrt{\sum_{j=1}^{n} (aij - ai)2/(n-1)} \qquad \text{(Equation 3.3)}$$

$$CI_i = t_{\alpha/2} \cdot \sigma_i / \sqrt{n} \qquad \text{(Equation 3.4)}$$

Once the bottleneck is found by using the above methodology it would be easy to improve the performance of the system by fine-tuning that particular component. This technique forms the basis of this thesis work.

3.3.3 Bottleneck Detection Using Waiting Time

Assuming a system consists of several components which process 'units' one by one. Since a component services the units one by one, queues of units will be formed before the component. The waiting time of the units in the queue is used to find out the bottleneck components in the system. But there are several disadvantages associated with such a technique. The disadvantages are as follows:

- This method is applicable for linear systems only since the methodology inherently assumes that the distribution of the units to the components, processing times of the components, as linear.
- Multiple unit types lead to occasions where a component with a few units being processed slowly constrains the system more than a component with a lot of units.
- Queues (buffers) should have infinite capacity, since different size queues can definitely be an issue.
- System capacity should exceed the supply in the long run to avoid permanently filled queues.

A unit is anything that is serviced by the components in the system and there can be different types of units that are processed by different components.

3.3.4 Bottleneck Detection Technique for Push-Pull Production Lines

Push-Pull systems are those systems that operate to maximize throughput [9]. In this case a sound theory has been developed which considers a system with buffers in between the component stages as shown in figure 3.4.

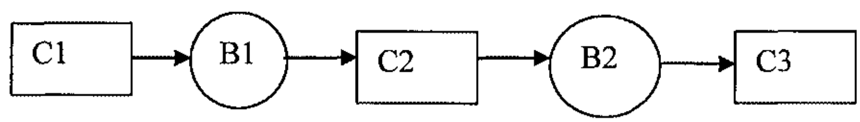

Figure 3.4: A simple push-pull production line

The following rules are used to determine the bottleneck component based on the status of the buffers.

(1) Component Ci, i=1, 2, 3…n-1, is said to be blocked during a time slot if it is up during this time slot, Bi (buffer i) is full at the beginning of this time slot, and Ci+1 fails to take the unit from Bi at the beginning of this time slot.

(2) Component Cm is blocked during a time slot if it is up during this time slot and Bm is full at the beginning of this time slot.

(3) Component Ci, i=2…n is said to be starved during a time slot if it is up during this time slot and Bi-1 is empty at the beginning of this time slot.

There are innumerous applications where bottleneck detection has been found to be very useful and the major application being in the national highway, industrial, production, computer communication and networking industries.

3.4 Latest applications of Bottleneck analysis and detection strategies

The following is the application of bottleneck detection and analysis techniques in various research activities carried out by major companies in their research centers.

(1) The Toyota Central Research and Development laboratories located in Japan has been conducting extensive research on the shifting bottleneck detection method for improving the production rate using the Automated Guided Vehicles (AGV's). The shifting bottleneck detection method proposed by them forms the basis of this thesis work [7].

(2) The Microsoft Research Advanced Technology Division has been using the decision theoretic techniques described earlier for diagnosis and treatment of bottleneck techniques in computer systems. The research is mainly concentrated on finding out the hardware bottlenecks that hinder the Windows NT operating system's performance. The outcome of the research would be helpful in determining a cost effective hardware upgrade as well as in estimating the amount of throughput increase due to the changes made [8].

(3) The Novell research group has been actively involved in isolating the real bottleneck in a system and has been working on it. Their aim is to efficiently handle the bottlenecks in the networks and improve the throughput [3].

Chapter 4

The Application of Bottleneck Analysis

4.1 Introduction

This chapter applies the *Shifting Bottleneck Analysis*, *Waiting Time Bottleneck Analysis*, and *Utilization Time Bottleneck Analysis* to an *example model* followed by a comparison of the results obtained by all the methodologies. The comparison will provide much insight as to why Shifting Bottleneck Analysis is better than the other methodologies for the problem domain that CPortS falls within.

4.2 Example Model

The example scenario defined for comparison of the various bottleneck detection techniques is an ideal and simplistic scenario with perfectly defined processing times and no failure machines and an infinite buffer supplying required bicycle parts. The example model considered is a bicycle assembly system. The system consists of 5 sequential machines (M1, M2, M3, M4, M5) with 2 buffers of size 1 between successive machines. These machines are responsible for assembling various parts of a bicycle. The final product from this system is a *ready to ride bicycle*. The system is diagrammatically represented as in figure 4.1. This model has been defined and designed for this thesis as a means of contrasting the bottleneck analysis methods.

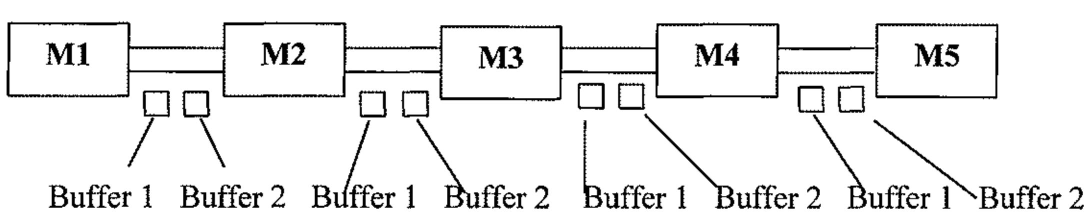

Figure 4.1 showing a system of successive machines with supporting buffers in between.

4.2.1 Machine Functionality

The machine M1 is responsible for attaching the wheels to the bicycle body, M2 for fitting the rubber tires on the wheels, M3 for attaching the pedals and cycle chain, M4 for attaching the mudguards on the front and rear wheel, and M5 for attaching the seat to the bicycle.

4.2.2 Machine States

All the 5 machines are in one of the following states at any given time: working, waiting or blocked. The next step in this system is to identify the active states and the inactive states. An active state can be defined as a state or a time interval during which the machine is doing some productive work, which contributes to the throughput. Similarly an idle state is a state or a time interval in which the machine remains idle doing nothing. These two states are mutually exclusive meaning that they can never occur at the same time. The active and inactive states are listed in Table 4.2.

State	Active	Inactive
Working	Yes	No
Waiting	No	Yes
Blocked	No	Yes

Table 4.2: Active and Inactive states

4.3 Scenario

For analysis and comparison of the Bottleneck Techniques mentioned, an ideal scenario for the bicycle assembly system is defined as follows:

There are 25 bicycle bodies to be assembled. These 25 bicycles are fed to M1. These are assumed to be readily available to M1 from a large buffer. The machine M1 takes 5 time units for processing one bicycle. Machine M2 takes 3 time units, machine M3 takes 6 time units, machine M4 takes 5 time units and machine M5 takes 2 time units to finish its task. The transmission delay between M1 and M2 is 1 time unit, between M2 and M3 is 2 time units, between M3 and M4 is 3 time units and between M4 and M5 is 1 time unit, respectively. The time schedule for assembling the 25 parts can be found in table 4.3.

In table 4.3 the following notations have been used.

M1 - Machine 1

M2 - Machine 2

M3 - Machine 3

M4 - Machine 4

M5 - Machine 5

B1 - Buffer 1

B2 - Buffer 2

Ci - Bicycle 'i'

E - Empty

Ci (a, b) - bicycle 'i' was processed by machine 'i' starting from time interval 'a' until time interval 'b'.

M1	B1	B2	M2	B1	B2	M3	M4	M5
C1(0,5)	E	E	C2(6,9)	E	E	C1(11,17)	C1(20,25)	C1(26,28)
C2(5,10)	E	E	C2(11,14)	C2(16,17)	E	C2(17,23)	C2(26,31)	C2(32,34)
C3(10,15)	E	E	C3(16,19)	C3 (21,23)	E	C3(23,29)	C3(32,37)	C3(38,40)
C4(15,20)	E	E	C4(21,24)	C4 (26,29)	E	C4(29,35)	C4(38,43)	C4(44,46)
C5(20,25)	E	E	C5(26,29)	C5 (31,35)	E	C5(35,41)	C5(44,49)	C5(50,52)
C6(25,30)	E	E	C6(31,34)	C6 (36,41)	E	C6(41,47)	C6(50,55)	C6(56,58)
C7(30,35)	E	E	C7(36,39)	C7 (41,47)	E	C7(47,53)	C7(56,61)	C7(62,64)
C8(35,40)	E	E	C8(41,44)	C7 (41,47)	C8(46,53)	C8(53,59)	C8(62,67)	C8(68,70)
C9(40,45)	E	E	C9(46,49)	C9 (51,59)	C8(46,53)	C9(59,65)	C9(68,73)	C9(74,76)
C10(45,50)	E	E	C10(51,54)	C9 (51,59)	C10(56,65)	C10(65,71)	C10(74,79)	C10(80,82)
C11(50,55)	E	E	C11(56,59)	C11(61,71)	C10(56,65)	C11(71,77)	C11(80,85)	C11(86,88)
C12(55,60)	E	E	C12(61,64)	C11 (61,71)	C12(66,77)	C12(77,83)	C12(86,91)	C12(92,94)
C13(60,65)	E	E	C13(66,69)	C13 (71,83)	C12(66,77)	C13(83,89)	C13(92,97)	C13(98,100)
C14(65,70)	C13 (71,72)	E	C14(72,75)	C13 (71,83)	C14(77,89)	C14(89,95)	C14(98,103)	C14(104,106)
C15(70,75)	C15 (76,78)	E	C15(78,81)	C15 (83,95)	C14(77,89)	C15(95,10	C15(104,109)	C15(110,112)
C16(75,80)	C15 (81,84)	E	C16(84,87)	C15 (83,95)	C16(89,101)	C16(101,107)	C16(110,115)	C16(116,118)
C17(80,85)	C17 (86,90)	E	C17(90,93)	C17(95,107)	C16(89,101)	C17(107,113)	C17(116,121)	C17(122,124)
C18(85,90)	C17 (91,96)	E	C18(96,99)	C17(95,107)	C18(101,113)	C18(113,119)	C18(122,127)	C18(128,130)
C19(90,95)	C19 (96,102)	E	C19(102,105)	C19(107,119)	C18(101,113)	C19(119,125)	C19(128,133)	C19(134,136)
C20(95,100)	C19 (96,102)	C20(101,108)	C20(108,114)	C19(107,119)	C20(113,125)	C20(125,131)	C20(134,139)	C20(140,142)
C21(100,105)	C21 (106,114)	C20(101,108)	C21(114,117)	C21(119,131)	C20(113,125)	C21(131,137)	C21(140,145)	C21(146,148)
C22(105,110)	C21 (106,114)	C22(111,120)	C22(120,123)	C21(119,131)	C22(125,137)	C22(137,143)	C22(146,151)	C22(152,154)
C23(110,115)	C23 (116,126)	C22(111,120)	C23(126,129)	C23(131,143)	C22(125,137)	C23(143,149)	C23(152,157)	C23(158,160)
C24(115,120)	C23 (116,126)	C24(121,132)	C24(132,135)	C23(131,143)	C24(137,149)	C24(149,155)	C24(158,163)	C24(164,166)
C25(120,125)	C25 (126,138)	C24(121,132)	C25(138,141)	C25(143,155)	C24(137,149)	C25(155,161)	C25(164,169)	C25(170,172)

Table 4.3: Bicycle assembly machine time table

In the construction of table 4.3 an ideal scenario has been envisioned wherein there are no machine breakdowns, no delays other than the transmission delays, and a machine is ready for processing the next part immediately after it has processed the previous part.

As can be seen from table 4.3, M1 processes bicycle 1 from 0 to 5 time units and then starts immediately at time unit 5 on the second bicycle. There is no specific unit for time, it can be hours or minutes or anything. All the times are rounded to the nearest integer (i.e. no floating point values in the above table). The above is a deterministic time process wherein the start of process on one machine depends upon the outcome on the previous machine. Since all the times are predefined the process can be considered to be purely deterministic. The above table has been constructed manually.

4.3.1 Analysis using Utilization Time Bottleneck methodology

As discussed in the previous chapter the *utilization time bottleneck analysis* methodology considers the percentage of time a machine or a component of a system is active and ranks the bottleneck machines/components according to the activity percentage as primary bottleneck, secondary bottleneck, tertiary bottleneck and so on.

Using the data in table 4.3 the following results are obtained:

Active Time for Machine M1 is: 25*5 = 125 time units.

Active Time for Machine M2 is: 25*3 = 75 time units.

Active Time for Machine M3 is: 6*25 = 150 time units.

Active Time for Machine M4 is: 25*5 = 125 time units.

Active Time for Machine M5 is: 25*2 = 50 time units.

Total assembly time= 172 time units.

The active time % of the various machines is diagrammatically shown in figure 4.4.

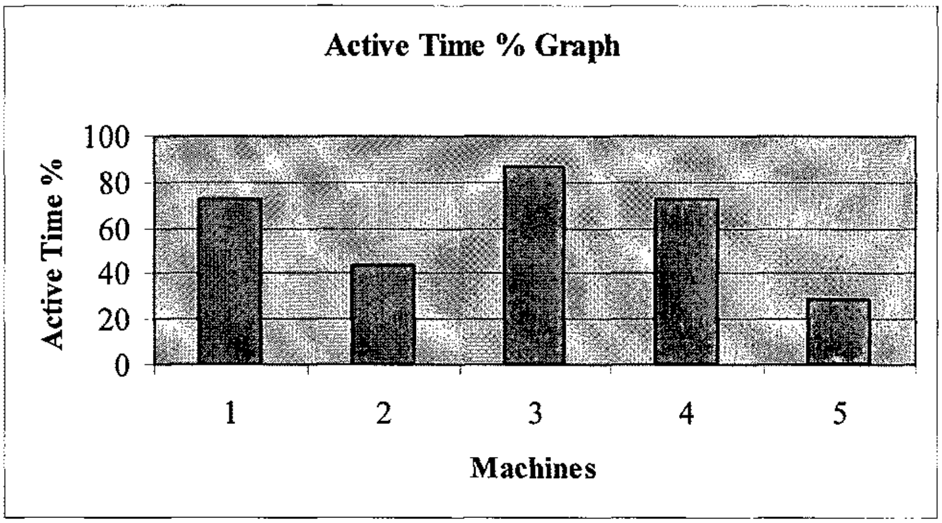

Figure 4.4: Active time % for the machines.

According to the utilization method, M3 is the bottleneck machine, since it has the highest active time %. But the utilization method cannot determine the secondary bottleneck. This is because both M1 and M4 have the same active time % (second highest active time %) and since utilization method depends on the active time %, which in this case does not give sufficient information to point the secondary bottleneck. This also points another drawback of the utilization method; it cannot distinguish a bottleneck machine from a non-bottleneck machine, since it does not specify how to distinguish a bottleneck machine from a non-bottleneck machine. Similarly if there are two machines with the highest active time %, the utilization method fails to pinpoint the principal bottleneck machine, since there is more than one machine with high active time %. In such cases the utilization method will be a complete failure, since it cannot detect the principal bottleneck, which is its main purpose.

Generally the bottleneck techniques are applied to highly parallel production and assembly lines and also to discrete systems. In such cases the utilization method will be faced with activity times, which are almost equal as well as with data which is not accurate and which has been approximated at. With approximated data the results obtained from the utilization method cannot be relied upon. This clearly indicates the limitation of the utilization method for bottleneck analysis.

4.3.2 Analysis Using Waiting Time Bottleneck Methodology

In this methodology the average waiting time to obtain the services of a machine is the deciding factor for determining the bottleneck machine. Here, the machine with the longest waiting time is considered to be the bottleneck. The accuracy of this method is compromised if the system contains buffers of limited size. If there are buffers of limited size then waiting time in front of a machine becomes dependent upon the availability of buffers. If buffers are unavailable then the system has to halt until some buffers are available to resume normal working. In turn the buffers become the cause of the bottleneck rather than the inefficient machine, which has to be sorted out. The scenario constructed above has a limited number of buffers of limited capacity, which is definitely a negative for analysis. Hence this methodology cannot be applied to the example assembly system considered.

4.3.3 Analysis Using Shifting Bottleneck Methodology

The shifting bottleneck methodology is a very simple and straightforward methodology, but has the capabilities to resolve the issues faced in the utilization time bottleneck methodology as well as the waiting time bottleneck methodology. The shifting bottleneck method classifies the bottlenecks based on the time during which the bottleneck is active. If at time t no machines are active, then there is no bottleneck. If one or more machines are active at time t the machine with the longest active period at time t is the momentary bottleneck machine, and the active period of this machine is the current bottleneck period. If the current bottleneck period ends, it is necessary to find the next bottleneck by determining the machine with longest active period after the current bottleneck period ended. The shifting of the bottleneck from the current bottleneck to the subsequent bottleneck machine happens during the overlap of the current and subsequent bottleneck periods. During the overlaps between the bottleneck periods no machine is the sole bottleneck, instead the bottleneck shifts between the two machines. If a bottleneck machine is not shifting, then this machine is the sole and only bottleneck at this time [2]. The shifting bottleneck methodology considers bottlenecks to be dynamic and associates each and every bottleneck with a time interval during which it is dominant. According to this methodology, fine-tuning a bottleneck does not guarantee immediate improvement in the system performance, since the bottleneck might now shift to another machine, which becomes the primary bottleneck to be considered.

Using this method, it can be determined at any given time if a machine is a non-bottleneck, a shifting bottleneck, or a sole bottleneck (which none of the other bottleneck detection methods could distinguish clearly). This method allows the detection of the momentary bottleneck, where and when the previous bottleneck was shifting to the current bottleneck, and where and when the current bottleneck is shifting to the next bottleneck [2]. Figure 4.5 illustrates the method for the example scenario considered.

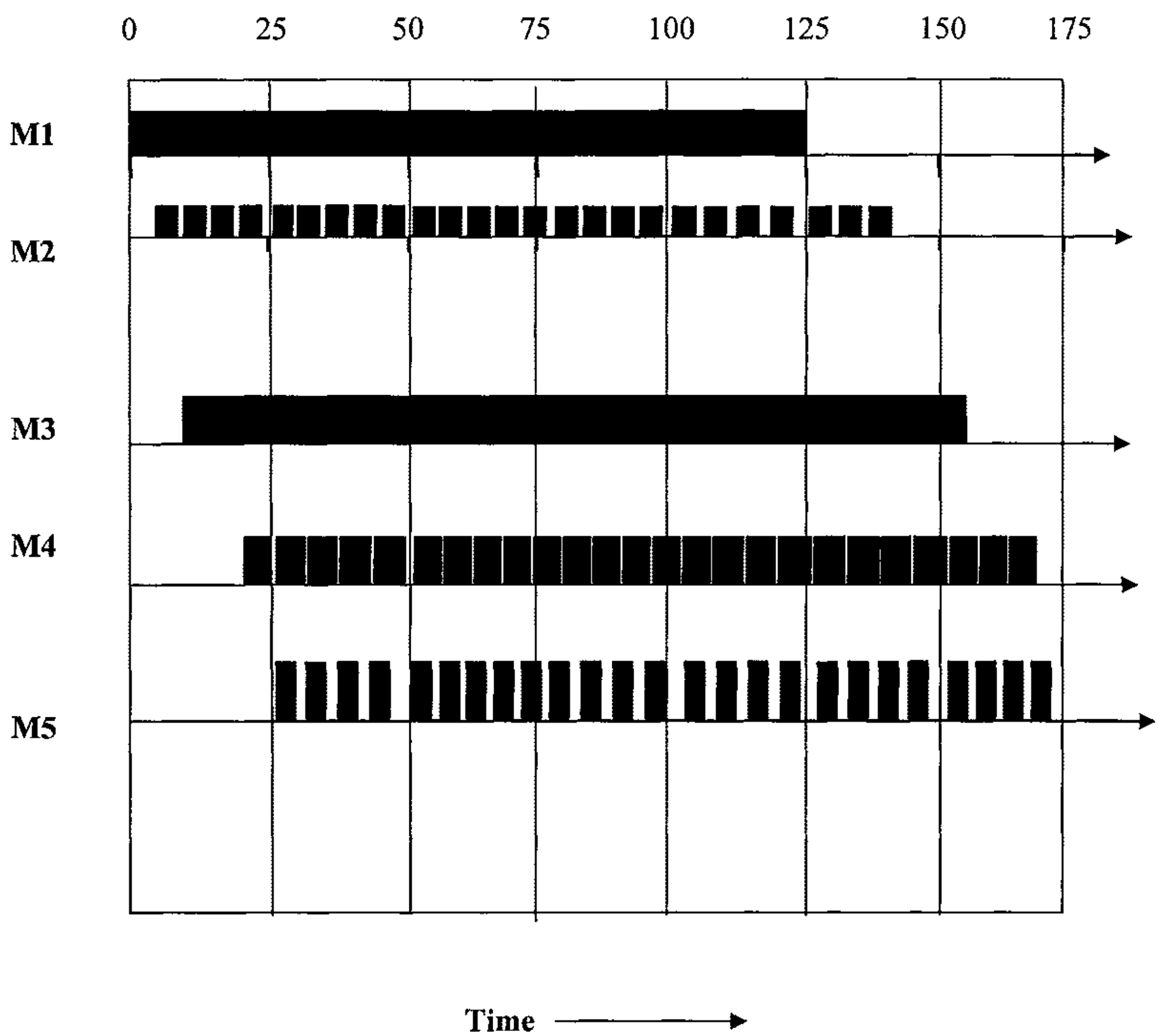

Figure 4.5: Activity and idle periods for all the machines

Figure 4.5 reflects the active times of all the machines M1 through M5. M1 is active from time unit 0 to time unit 125. Similarly M2 has several short intervals of active periods which recur after an interval of 3 time units, M3 has a continuous active period from time unit 11 to time unit 161, M4 has several short intervals of 5 units recurring after an interval of 1 time unit, and M5 has active period intervals of 2 units recurring after every 4 time units.

4.4 Detecting the Bottlenecks

During the time interval from 0 to 150, machines M1 and M3 have the largest active periods. The time interval 0-150 can be represented as in figure 4.6.

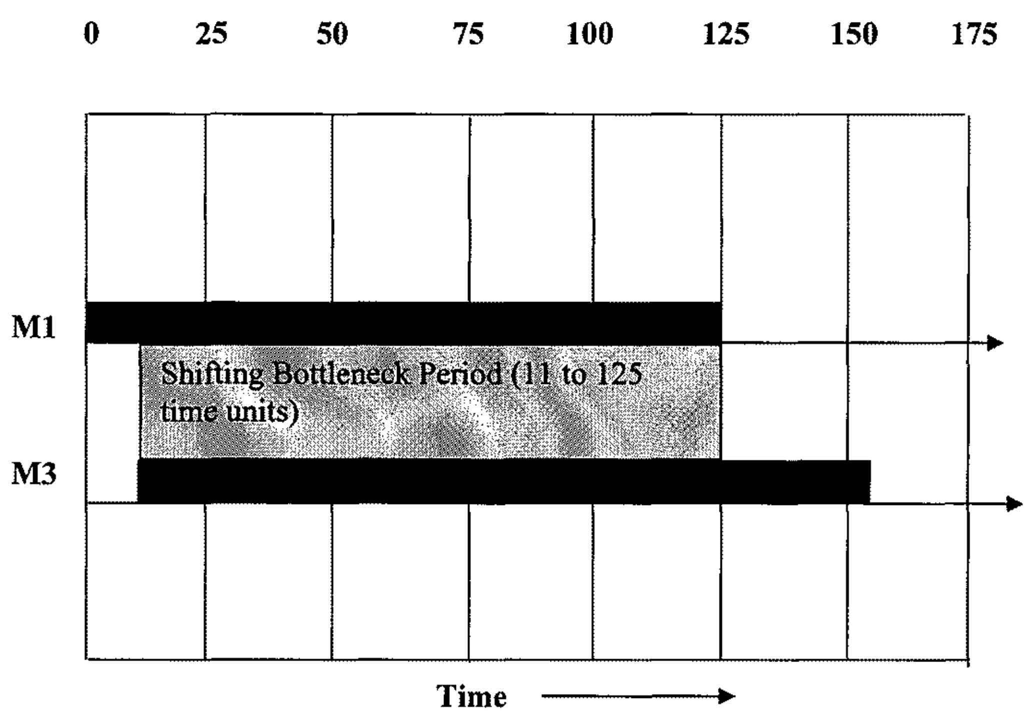

Figure 4.6: Sole and shifting bottleneck periods for machines M1 and M3

At time 10, M1 has the largest active period (10 time units), so during this period the sole bottleneck is the machine M1. The period of time for which M1 is the sole bottleneck is quite small, which indicates it might not have a great influence on the throughput rate. During the time interval from 11 to 125, M1 and M3 are the shifting bottlenecks. The machines M2 and M4 do not have a continuous active period during this interval, which indicates that they do not contribute either as a sole bottleneck or as a shifting bottleneck. M5 has chunks of active periods after 161 time units, which are not continuous, so even it can be considered as a non-bottleneck machine. At the end of time unit 125 (end of shifting bottleneck period), M3 is active and has the longest active period. Hence the subsequent bottleneck machine is M3. M3 continues to be the sole bottleneck until time unit 161. After time unit 161, no machine has a considerable, continuous or dominating active period to be labeled either as the sole bottleneck or as the shifting bottleneck.

4.5 Sensitivity Analysis

Sensitivity analysis enhances the shifting bottleneck methodology by analyzing the events of which the bottlenecks periods consist of (events are the detailed information about the active states, like fitting tires, attaching the seat, etc). The sensitivity analysis is aimed at finding the events of a machine when it is the sole bottleneck. By improvising these events (decreasing the processing time or by improving the events) the effects of the bottleneck can be alleviated.

The bottleneck periods limit the overall systems throughput, and the bottleneck periods consist of the different actions of the machines. Therefore, the actions of the bottleneck machines during the bottleneck periods determine the overall system throughput. Knowing the sole and shifting bottleneck periods and the events therein, the percentage contribution of the variables of the machines to the throughput can be calculated easily [2]. The equation (4.1) shows the calculation of the percentage effect of state j of machine i due to the sole bottleneck $P_{i,j}^{sole}$ and the equation (4.2) shows the calculation of the percentage effect of state j of machine i due to the shifting bottleneck $P_{i,j}^{shifting}$ [2].

$$P_{i,j}^{sole} = \frac{1}{T_{end}-T_{start}} * \int_{tstart}^{tend} \begin{cases} 1 & \text{If machine 'i' is in event 'j' and 'i' is the sole bottleneck.} \\ 0 & \text{Otherwise} \end{cases} \quad \text{(equation 4.1)}$$

$$P_{i,j}^{shifting} = \frac{1}{T_{end}-T_{start}} * \int_{tstart}^{tend} \begin{cases} 1 & \text{If machine 'i' is in event 'j' and 'i' is the shifting bottleneck.} \\ 0 & \text{Otherwise} \end{cases} \quad \text{(equation 4.2)}$$

By performing a sensitivity analysis on the example scenario, the following details can be observed:

- The example scenario has a single active state, which is the working state and the events corresponding to it are *attaching wheels, fitting rubber tires on the wheels, attaching pedals and cycle chain, attaching mudguards on the front and rear wheel, and attaching the seat to the bicycle.*
- The bottleneck machines as deduced earlier are M1 and M3.
- The events associated with M1 working are *attaching wheels* and the events corresponding to M3 working are *attaching pedals and cycle chain.*

By applying the equations (4.1) and (4.2) to machines M1 and M2 the following results as well as figure 4.7 are obtained.

For machine M1:

$P_{i,j}^{sole}$=(11-0)*100/(172-0)= 6.4 %.

$P_{i,j}^{shifting}$= (125-11)*100/(172-0)= 66.3 %.

For machine M3:

$P_{i,j}^{sole}$= (161-125)*100/(172-0)=20.93 %.

$P_{i,j}^{shifting}$= (125-11)*100/(172-0)= 66.3 %.

Figure 4.7 shows the sole and shifting bottleneck percentages of the dominating bottleneck machines.

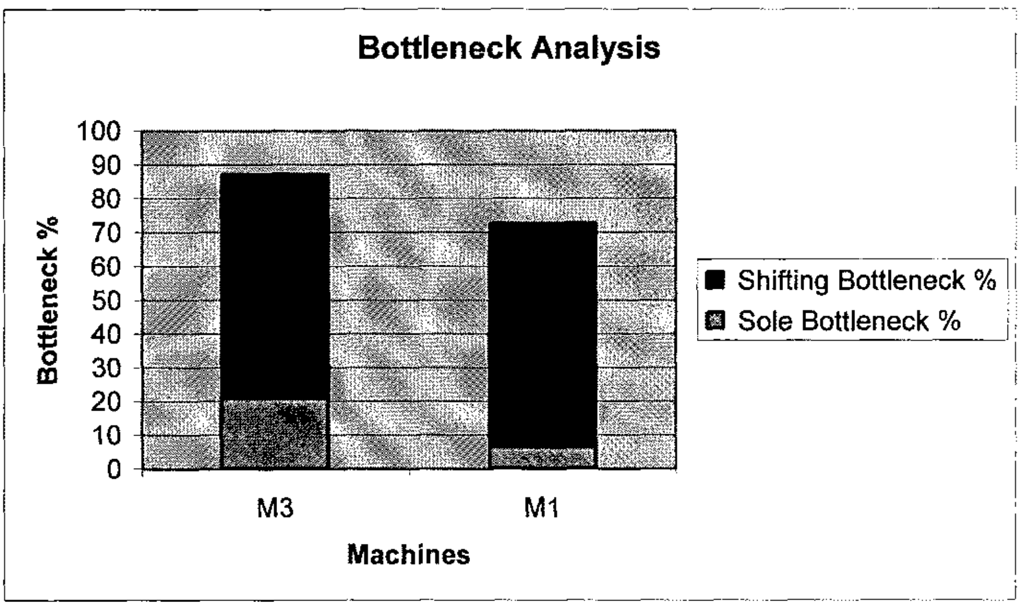

Figure 4.7: Bottleneck Machines sole and shifting bottleneck percentage

As can be seen from the outcome of the above equations and figure 4.7, M1 has a very small % of the sole bottleneck period, which indicates it may or may not influence the throughput rate considerably, but M3 has a sufficient sole bottleneck period (20.93%) and it can have some noticeable influence on the throughput rate. The important thing to be noted is the shifting bottleneck percentage for both the machines is quite high (66.3%). So during this shifting bottleneck period, improving the events of both the bottleneck machines may or may not improve the overall throughput rate since the shifting bottleneck period may be dependent equally on both the machines or might be driven by some other machine in the system.

Effectiveness of the results obtained from the various bottleneck detection methodologies

Before proceeding with the comparison the following information can be obtained from table 4.3.

- Total time taken for assembling 25 bicycles = 172 time units.
- Time taken for assembling 1 bicycle =172/25= 6.9 time units

To test whether the bottlenecks determined by the above methodologies are really the system bottlenecks, verification is required. Decreasing the processing times of the bottleneck machines does this.

Both the utilization time bottleneck methodology and the shifting bottleneck methodology consider M3 as the principal bottleneck since it has the largest utilization time. The processing time of the machine M3 is 6 time units. Assuming a technological innovation has reduced the processing time of machine M3 by half (3 time units), table 4.8 can be constructed.

M1	B 1	B2	M2	B1	B2	M3	M4	M5
C1(0,5)	E	E	C2(6,9)	E	E	C1(11,14)	C1(17,22)	C1(23,25)
C2(5,10)	E	E	C2(11,14)	E	E	C2(16,19)	C2(22,27)	C2(28,30)
C3(10,15)	E	E	C3(16,19)	E	E	C3(21,24)	C3(27,32)	C3(33,35)
C4(15,20)	E	E	C4(21,24)	E	E	C4(26,29)	C4(32,37)	C4(38,40)
C5(20,25)	E	E	C5(26,29)	E	E	C5(31,34)	C5(37,42)	C5(43,45)
C6(25,30)	E	E	C6(31,34)	E	E	C6(36,39)	C6(42,47)	C6(48,50)
C7(30,35)	E	E	C7(36,39)	E	E	C7(41,44)	C7(47,52)	C7(53,55)
C8(35,40)	E	E	C8(41,44)	E	E	C8(46,49)	C8(52,57)	C8(58,60)
C9(40,45)	E	E	C9(46,49)	E	E	C9(51,54)	C9(57,62)	C9(63,65)
C10(45,50)	E	E	C10(51,54)	E	E	C10(56,59)	C10(62,67)	C10(68,70)
C11(50,55)	E	E	C11(56,59)	E	E	C11(61,64)	C11(67,72)	C11(73,75)
C12(55,60)	E	E	C12(61,64)	E	E	C12(66,69)	C12(72,77)	C12(78,80)
C13(60,65)	E	E	C13(66,69)	E	E	C13(71,74)	C13(77,82)	C13(83,85)
C14(65,70)	E	E	C14(71,74)	E	E	C14(76,79)	C14(82,87)	C14(88,90)
C15(70,75)	E	E	C15(76,79)	E	E	C15(81,84)	C15(87,92)	C15(93,95)
C16(75,80)	E	E	C16(81,84)	E	E	C16(86,89)	C16(92,97)	C16(98,100)
C17(80,85)	E	E	C17(86,89)	E	E	C17(91,94)	C17(97,102)	C17(103,105)
C18(85,90)	E	E	C18(91,94)	E	E	C18(96,99)	C18(102,107)	C18(108,110)
C19(90,95)	E	E	C19(96,99)	E	E	C19(101,104)	C19(107,112)	C19(113,115)
C20(95,100)	E	E	C20(101,104)	E	E	C20(106,109)	C20(112,117)	C20(118,120)
C21(100,105)	E	E	C21(106,109)	E	E	C21(111,114)	C21(117,122)	C21(123,125)
C22(105,110)	E	E	C22(111,114)	E	E	C22(116,119)	C22(122,127)	C22(128,130)
C23(110,115)	E	E	C23(116,119)	E	E	C23(121,124)	C23(127,132)	C23(133,135)
C24(115,120)	E	E	C24(121,124)	E	E	C24(126,129)	C24(132,137)	C24(138,140)
C25(120,125)	E	E	C25(126,129)	E	E	C25(131,134)	C25(137,142)	C25(143,145)

Table 4.8: Table obtained by reducing M3 processing time to 3 units

Using the data in table 4.8 the following results are obtained.

- Total time taken for assembling 25 bicycles is: 145 time units.
- Time for assembling 1 bicycle is: 145/25 = 5.8 time units.

Earlier, before the bottleneck machine was improved, the time for assembling a single bicycle was 6.9 time units. Hence after improving the bottleneck machine, the time for assembling is decreased by 15.94 %.

This is a fair result as far as the utilization time bottleneck methodology is concerned. But now there is a bottleneck shift, a new bottleneck replaced M3 and as mentioned earlier utilization method cannot detect secondary bottlenecks and shifting bottlenecks. To verify whether M3 is really the principal bottleneck as suggested by the utilization time method as well as the shifting bottleneck method, lets improve the processing times of all other machines just like M3 and find out by how much % they can fasten up the assembly process.

Again consider the table 4.3 in which the processing time of M5 has been halved. Table 4.9 shows the outcome of such a modification.

M1	B1	B2	M2	B1	B2	M3	M4	M5
C1(0,5)	E	E	C2(6,9)	E	E	C1(11,17)	C1(20,25)	C1(26,27)
C2(5,10)	E	E	C2(11,14)	C2(16,17)	E	C2(17,23)	C2(26,31)	C2(32,33)
C3(10,15)	E	E	C3(16,19)	C3 (21,23)	E	C3(23,29)	C3(32,37)	C3(38,39)
C4(15,20)	E	E	C4(21,24)	C4 (26,29)	E	C4(29,35)	C4(38,43)	C4(44,45)
C5(20,25)	E	E	C5(26,29)	C5 (31,35)	E	C5(35,41)	C5(44,49)	C5(50,51)
C6(25,30)	E	E	C6(31,34)	C6 (36,41)	E	C6(41,47)	C6(50,55)	C6(56,57)
C7(30,35)	E	E	C7(36,39)	C7 (41,47)	E	C7(47,53)	C7(56,61)	C7(62,63)
C8(35,40)	E	E	C8(41,44)	C7 (41,47)	C8 (46,53)	C8(53,59)	C8(62,67)	C8(68,79)
C9(40,45)	E	E	C9(46,49)	C9 (51,59)	C8 (46,53)	C9(59,65)	C9(68,73)	C9(74,75)
(C10(45,50	E	E	C10(51,54)	C9 (51,59)	C10	C10(65,71)	C10(74,79)	C10(80,81)
C11(50,55)	E	E	C11(56,59)	C11	C10	C11(71,77)	C11(80,85)	C11(86,87)
C12(55,60)	E	E	C12(61,64)	C11	C12	C12(77,83)	C12(86,91)	C12(92,93)
C13(60,65)	E	E	C13(66,69)	C13	C12	C13(83,89)	C13(92,97)	C13(98,99)

C14(65,70)	C13 (71,72)	E	C14(72,75)	C13	C14	C14(89,95)	C14(98,103)	C14(104,105)
C15(70,75)	C15 (76,78)	E	C15(78,81)	C15	C14	C15(95,101)	C15(104,109)	C15(110,111)
C16(75,80)	C15 (81,84)	E	C16(84,87)	C15	C16	C16(101,107	C16(110,115)	C16(116,117)
C17(80,85)	C17 (86,90)	E	C17(90,93)	C17	C16	C17(107,113	C17(116,121)	C17(122,123)
C18(85,90)	C17 (91,96)	E	C18(96,99)	C17	C18	C18(113,119	C18(122,127)	C18(128,129)
C19(90,95)	C19 (96,102)	E	C19(102,105	C19	C18	C19(119,125	C19(128,133)	C19(134,135)
C20(95,10	C19 (96,102)	C20	C20(108,111	C19	C20	C20(125,131	C20(134,139)	C20(140,141)
C21(100,1	C21 (106,114)	C20	C21(114,117	C21	C20	C21(131,137	C21(140,145)	C21(146,147)
C22(105,1	C21 (106,114)	C22	C22(120,123	C21	C22	C22(137,143	C22(146,151)	C22(152,153)
C23(110,1	C23 (116,126)	C22	C23(126,129	C23	C22	C23(143,149	C23(152,157)	C23(158,159)
C24(115,1	C23 (116,126)	C24	C24(132,135	C23	C24	C24(149,155	C24(158,163)	C24(164,165)
C25(120,1	C25 (126,138)	C24	C25(138,141	C25	C24	C25(155,161	C25(164,169)	C25(170,171)

Table 4.9: Table showing the effects of halving machine M5's processing time

Here the total time taken for assembly is 171 time units, which indicates that there is no significant decrease in the assembly process. It can be concluded that M5 is a non-bottleneck machine and improving it or modifying it can at most have some negligible improvement on the overall assembly time. The above was deduced by the shifting bottleneck methodology. The utilization method did not provide any idea about that. It just ranked it as the 4/5-bottleneck machine.

Similarly decreasing the processing time of M4 will not have any significant effect on the total assembly time. Even this information was provided by the shifting bottleneck methodology where machines M2, M4 and M5 were considered as non-bottlenecks. The following table 4.10 lists the activity for M1 when its processing tme is reduced to 3 time units.

M1	B1	B2	M2	B1	B2	M3	M4	M5
C1(0,3)	E	E	C2(4,7)	E	E	C1(9,15)	C1(18,23)	C1(24,26)
C2(3,6)	E	E	C2(7,10)	C2(12,15)	E	C2(15,21)	C2(24,29)	C2(30,32)
C3(6,9)	E	E	C3(10,13)	C2(12,15)	C3(15,21)	C3(21,27)	C3(30,35)	C3(36,38)
C4(9,12)	E	E	C4(13,16)	C4(18,27)	C3(15,21)	C4(27,33)	C4(36,41)	C4(42,44)
C5(12,15)	E	E	C5(16,19)	C4(18,27)	C5(21,33)	C5(33,39)	C5(42,47)	C5(48,50)
C6(15,18)	E	E	C6(19,22)	C6 (24,39)	C5(21,33)	C6(39,45)	C6(48,53)	C6(54,56)
C7(18,21)	E	E	C7(22,25)	C6 (24,39)	C7(27,45)	C7(45,51)	C7(54,59)	C7(60,62)
C8(21,24)	C8(25,36)	E	C8(36,39)	C8(41,51)	C7(27,45)	C8(51,57)	C8(60,65)	C8(66,68)
C9(24,27)	C8(25,36)	C9(28,40)	C9(40,43)	C8(41,51)	C9(45,57)	C9(57,63)	C9(66,71)	C9(72,74)
C10(32,35)	C10(36,46)	C9(28,40)	C10(46,49)	C10(51,62)	C9(45,57)	C10(62,68)	C10(71,76)	C10(77,79)
C11(36,39)	C10(36,46)	C11(40,52)	C11(52,55)	C10(51,62)	C11(57,67)	C11(67,73)	C11(76,81)	C11(82,84)
C12(42,45)	C12(46,57)	C11(40,52)	C12(57,60)	C12(62,73)	C11(57,67)	C12(73,79)	C12(82,87)	C12(88,90)
C13(48,51)	C12(46,57)	C13(52,62)	C13(62,65)	C12(62,73)	C13(67,79)	C13(79,85)	C13(88,93)	C13(94,96)
C14(53,56)	C14 (57,68)	C13(52,62)	C14(68,71)	C14(73,85)	C13(67,79)	C14(85,91)	C14(94,99)	C14(100,102)
C15(58,61)	C14 (57,68)	C15(62,74)	C15(74,77)	C14(73,85)	C15(79,91)	C15(91,97)	C15(100,105)	C15(106,108)
C16(64,67)	C16 (68,80)	C15(62,74)	C16(80,83)	C16(85,97)	C15(79,91)	C16(97,103)	C16(106,111)	C16(112,114)
C17(70,73)	C16 (68,80)	C17(74,86)	C17(86,89)	C16(85,97)	C17(91,103)	C17(103,109)	C17(112,117)	C17(118,120)
C18(76,79)	C18(80,92)	C17(74,86)	C18(92,95)	C18(97,109)	C17(91,103)	C18(109,115)	C18(118,123)	C18(124,126)
C19(82,85)	C18(80,92)	C19(86,98)	C19(98,101)	C18(97,109)	C19(103,115)	C19(115,121)	C19(124,129)	C19(130,132)
C20(88,91)	C20(92,104)	C19(86,98)	C20(104,107)	C20(109,121)	C19(103,115)	C20(121,127)	C20(130,135)	C20(136,138)

C21(94,97)	C20(92, 104)	C21(98,11 0)	C21(110,1 13)	C20(109,121)	C21(115,1 27)	C21(127,1 33)	C21(136,141)	C21(142,144)
C22(100,1 03)	C22(10 4,116)	C21(98,11 0)	C22(116,1 19)	C22(121,133)	C21(115,1 27)	C22(133,1 39)	C22(142,147)	C22(148,150)
C23(106,1 09)	C22(10 4,116)	C23(110,1 22)	C23(122,1 25)	C22(121,133)	C23(127,1 39)	C23(139,1 45)	C23(148,153)	C23(154,156)
C24(112,1 15)	C24(11 6,128)	C23(110,1 22)	C24(128,1 31)	C24(133,145)	C23(127,1 39)	C24(145,1 51)	C24(154,163)	C24(164,166)
C25(118,1 21)	C24(11 6,128)	C25(122,1 34)	C25(134,1 37)	C24(133,145)	C25(139,1 51)	C25(151,1 57)	C25(160,165)	C25(166,168)

Table 4.10: Table showing the effect when machine M1's processing time reduced from 5 to 3 time units.

The total assembling time has decreased from 171 to 168 time units, which is not significant as predicted earlier due to the low percentage of the M1 sole bottleneck time. Since there is a significant shifting bottleneck time involving M1 and M3, by modifying the processing times of both there may or may not be an improvement in the throughput rate. This can be attributed to the fact that due to the decrease in the processing times of M1 and M3, the bottleneck might shift to other machines having a higher processing time which then start to drive the throughput rate just like the machine M3.

The essence of the shifting bottleneck analysis is that bottlenecks in a system are time dependent. There will be a sole bottleneck for a given time interval only. By improving the sole bottleneck at that time the bottleneck will shift to other machines in the system, which indicates that improving the sole bottleneck can give rise to a new bottleneck. By applying the shifting bottleneck methodology iteratively, the system can be tuned accordingly.

Chapter 5

Applying the Shifting Bottleneck Analysis to CPortS: Testing and Results

5.1 Introduction

The shifting bottleneck methodology described in Chapter 3 has been applied to the CPortS Model in this chapter by utilizing an example scenario provided by MTMCTEA. This will be followed by the verification phase that determines how effective the shifting bottleneck methodology will be in finding out the bottlenecks in CPortS.

5.2 Methodology

The foremost step in this phase is to gather the raw data by running the example scenario (henceforth known as the base scenario) using the CPortS user interface as well as the executable corresponding to the CPortS project in the MODSIM III environment. This data has been utilized for determining the bottleneck periods and the causes of those bottlenecks. The data gathered consists of details such as the arrival of cargo into a particular port area, the time at which the cargo is attended upon, the time at which the cargo leaves a particular port area, and the time duration during which the cargo remains idle. Appropriate modifications were made in the CPortS code to obtain the required details of cargo arrivals and departures and processing times as well as idle times. The raw cargo output files as well as the output analyzer from the CPortS user interface were also utilized for obtaining the cargo information along with the CPortS code.

Since there are several port areas within a port, each port area is analyzed for the above-mentioned details. So at the end of the data-gathering phase a large amount of data about the cargo events in each port area will be obtained. All this data is examined and analyzed according to the shifting bottleneck methodology to find out the principal bottleneck.

5.3 Description of the base scenario

The base scenario provided by MTMCTEA is used as input for generating the output data. It includes details such as the number of ships, the arrival time of the ships, the resources available to work on those ships in all the port areas. The base scenario can be better understood by looking at the tables that consists of the relevant information about the different port areas. Table 5.1 gives information about the physical characteristics of the port areas.

Area	Length (ft)	Breadth (ft)
Berthing Areas (7 used):		
Berth 25	787	43
Berth 26	787	43
Berth 27	640	43
Berth 28	787	43
Berth 29	787	43
Berth 30	590	43
Berth 31	590	43
Rail Spur:		
Spur1	2099.2	2099.2

Rail Interchange Yard: Unknown	15000	15000
Staging Area (1 used): Stage 4	**Capacity (SQFT)** 2980000	**Utilizable Capacity (SQFT)** 2980000
Helicopter Areas:		
Helicopter Reassembly	150000	150000
Helicopter Staging	300000	300000
Helicopter Takeoff	200000	200000

Table 5.1: Characteristics of port areas

Table 5.2 gives the details of the resources available in the different port areas.

Port Area	Resources	Number Available
Anchorage	Tugs	24
	Harbor Pilots	10
Berthing	Drivers	50
	Line Handlers	20
	Chassis	40
	Container Handlers	2
	Cranes	14

Staging	Chassis	100
	PSA Personnel	400
	Container	20
Helicopter Areas	Helicopter Tractors	2
Truck Loading Area	End Ramps	2
	Cranes	10
Interchange Yard Area	Port Locomotives	1

Table 5.2 : Resource availability in port areas

There are 30 ships in the base scenario. These ships arrive as batches (1 or 2 or 3 ships) at the port on different days. Table 5.3 gives the details about the 30 ships.

Ship Name	Length	Containers	Vehicles	Helicopters	Water crafts	Arrival Time
USNS Bob Hope	950	382	2315	0	0	0.30
Banner	493	0	439	0	0	0.30
Algol	946	171	892	78	0	0.30
Cape Decision	681	415	947	0	0	2.70
USNS Dahl	950	219	1311	0	0	2.70
Altair	946	172	759	64	0	2.70
Brinton Lykes	593	136	302	0	0	5.10
Cape Henry	750	436	837	0	0	5.10
Antares	946	107	922	0	0	5.10
Buyer	493	0	313	0	0	7.50
USNS Watkins	950	56	1727	0	0	7.50
Cape Douglas	681	99	843	0	0	7.50
Bellatrix	946	146	849	0	0	9.90
Cape Alexander	572	0	306	0	0	9.90
Cape Hudson	750	58	703	72	0	9.90
Denebola	946	69	817	0	0	12.30
Green Wave	507	97	216	0	0	12.30
Banner	493	0	239	0	0	14.70
Algol	946	50	836	0	0	14.70
USNS Bob Hope	950	141	1800	16	0	14.70

Altair	946	155	961	0	0	17.10
Cape Decision	681	90	828	0	0	17.10
USNS Dahl	950	198	1403	0	0	17.10
Brinton Lykes	593	136	397	6	0	19.50
Cape Henry	750	318	720	85	0	19.50
Antares	946	0	582	124	0	19.50
Buyer	493	0	438	0	0	21.90
Cape Douglas	681	0	1265	0	0	21.90
USNS Watkins	950	0	2283	23	0	21.90
Bellatrix	946	0	804	0	0	24.30

Table 5.3: Ship detail table

Since different batches of ships in the base scenario arrive on different days bottleneck analysis on the above scenario will be performed by grouping / batching the ships arriving at the same time. The movement of the cargo on these ships through the various port areas will be considered. If there are no ships in the port area during a particular time interval then all the port areas as well as the resources on the port areas will be idle and these idle times do not contribute to the bottleneck analysis. By using the batching methodology only the time during which the ships/cargo were in the port area will be considered. Cases wherein the processing of the cargo of a particular batch of ships, extends till the arrival of new batch of ships will be handled by considering the entire time during which the cargo was in the port areas.

A total of 11 batches of ships exist in the base scenario based on the arrival profile and it has been represented in Table 5.4.

Batch Number	Ship Name	Arrival Time
1	USNS Bob Hope Banner Algol	0.30
2	Cape Decision USNS Dahl Altair	2.70
3	Brinton Lykes Cape Henry Antares	5.10
4	Buyer USNS Watkins	7.50
5	Cape Douglas Bellatrix Cape Alexander	9.90
6	Cape Hudson Denebola Green Wave	12.30
7	Banner Algol USNS Bob Hope	14.70
8	Altair Cape Decision USNS Dahl	17.10
9	Brinton Lykes Cape Henry Antares	19.50
10	Buyer Cape Douglas USNS Watkins	21.90
11	Bellatrix	24.30

Table 5.4: Ship arrival profile table

5.4 Data Analysis Using the Shifting Bottleneck Methodology:

The bottleneck analysis will be performed on each batch of ships individually. After that, the principal bottlenecks in these 11 batches will be determined, followed by a sensitivity analysis wherein the dominating principal bottleneck components in all the eleven batches will be massaged to see if the shifting bottleneck methodology produces any positive outcome on the results.

The next section deals with the data gathered from the test runs of the CPortS model as well as the user interface. This data will be used for determining the active as well as the inactive / idle intervals in the various port areas. Details such as cargo arrival time, cargo process starts time, cargo process end time, cargo departure time and other active time intervals will be considered during this analysis. If the cargo waits on a resource, or waits in a queue, or gets stuck in a particular port area due to lack of resources such time is considered to be idle time/ inactive time and related data will not be considered in this analysis.

Considering the cargo movement across all the port areas, analysis will be performed. Each and every port area has events during which the cargo will be attended upon. Events will be classified as active or inactive based on what was being done on the cargo in that event. If during any period of time the cargo was processed and was not idling, then it is considered to be an active event. If the cargo just sits idle waiting for a resource or some handler then it is considered to be an inactive event.

After all active time intervals for a particular batch of ships have been considered, the shifting bottleneck methodology described in the Chapter 3 of this thesis will be applied to plot the bottleneck detection graphs. After all 11 batches of ships have been analyzed this way, the bottlenecks and their associated events will be improved either qualitatively or quantitatively to find the improvement in the overall throughput.

The next section shows an in-depth analysis of batch 1 ships followed by brief summaries of the other batches of ships.

5.4.1 Batch 1 analysis

Batch 1 has a total of 4277 cargo units (553 containers via road, 78 helicopters via air, 3644 vehicles via road and 2 vehicles via rail). Explaining the activity on the anchorage area and the berthing area will perform the analysis. This will show how the activity times were captured. For the remaining port areas only the activity intervals will be shown in tables, since the same methodology used for capturing the activity times in the berthing area was used for the other port areas.

(1) Anchorage Area:

The anchorage area mainly deals with the arrival of ships, and moving those ships to the berthing area. The ship departure event is resource dependent, if the resources are not available then the ship has to wait (a case of inactive time), if they are available the ship immediately moves to the berthing area, so no real active time excepting the time it takes to move the ship to the berthing area. This travel time is not considered in this thesis work since finding the bottleneck areas in the port areas is the primary goal and moreover the transit times between port areas are assumed to be a part of the port design, which cannot be changed radically unless new shorter routes can be constructed on the fly in the port. Hence throughout the thesis anchorage area events play no role as bottleneck events assuming sufficient resources i.e. tugs as well as harbor pilots are provided.

Table 5.5 shows the activity in the anchorage area for the batch 1 ships.

Ship Name	Arrival Time	Departure Time	Time in Queue	Tugs Available	Pilot Available
USNS Bob Hope	0.30	0.30	0.00	0.30	0.30
Banner	0.30	0.30	0.00	0.30	0.30
Algal	0.30	0.30	0.00	0.30	0.30

Table 5.5: Anchorage area activity table for batch 1 ships

(2) Berthing Area: The berths utilized for the batch 1 ships are berths 25, 26 and 28. So, the activity at each of these berthing areas has to be considered separately. The active time associated with the berthing events (ship arrival, cargo removal from ship, cargo movement to respective destination on the port using the movers/resources) has been gathered from the detailed cargo output as well as by running the CPortS executable with certain modifications to the modsim code in order to obtain the data not available through the output analyzer. This data consists of the start of the active time event and end of the active time event. All the unnecessary delays such as waiting for the movers, waiting for the appropriate cranes, etc. have not been considered since waiting time constitutes idle time which does not contribute to the principal bottleneck.

The active interval in a berthing area starts once the ship arrives at the berth area. The cargo on the ship will be offloaded and then it will be ready to be moved to the next port area (staging or helicopter areas). There are a large number of cargo units in this port area. Since this is the first batch of ships arriving in the port area all the resources available in the berthing area will be utilized on these three ships. Thus a continuous and busy activity can be seen on the berthing area. The helicopter parts are moved to the helicopter areas directly from the berthing area. So depending upon the rate at which these are sent into the helicopter areas, the activity take place in the helicopter areas. Table 5.6 shows the activity times on the above-mentioned berths.

	Start Activity	End Activity	Total Active Time (days)
Berth 25	0.49	1.5	1.01
	1.51	1.52	0.01
	1.53	1.56	0.03
	1.57	2.1	0.53
Berth 26	0.49	2.77	2.28
Berth 28	0.5	2.05	1.55
	2.1	2.16	0.06
	2.17	2.2	0.03
	2.21	2.28	0.07
	2.29	2.7	0.41
	2.73	2.76	0.03
	2.77	2.8	0.03
	2.81	2.84	0.03
	2.85	2.88	0.03
	2.89	3.05	0.16

Table 5.6: Berth area activity table

Staging Area: A single staging area is available for the base scenario. Thus depending upon the amount of cargo in the ships, the activity in the staging area can be busy. The cargo has to be routed to the different destination areas like the truck loading area, rail spur from the staging area. This results in a lot of activity in the staging area. Since there is a lot of cargo to be moved via road, the staging area has a continuous activity at the beginning followed by small bursts of discontinuous activity. This can be due to the wait for resources, which have to pick up the cargo from the staging area to the other port areas. This points to the high demand for the resources in the staging area. All these activity times can be found in the table 5.7.

Truck Loading Area: The truck loading area loads the cargo to be transported via road into the trucks. There is a single truck loading area. Depending upon the balance between the cargo and the resources available on the truck loading area, this area can have a busy activity. In this case, owing to sufficient resources in the truck loading area the activity occurs as quick and small activity intervals.

Helicopter Areas: The helicopter areas (i.e. reassembly, staging and takeoff) are used for assembling the helicopter parts and staging them and allowing them to takeoff. The major resources in these areas are the helicopter tractors and the pilots. Depending upon the availability of this resource the activity can be continuous or discontinuous. There are 78 helicopters in this scenario. There are two helicopter tractors and 10 pilots. The activity in this port area is very discontinuous since the parts arrive at different intervals of time from the berthing area. Thus, the activity intervals are discontinuous and distant from each other, effectively diluting the possibility of the helicopter areas being a contender for the primary bottleneck.

Rail Spur and Interchange Yard Area: These port areas are responsible for loading cargo into the rail cars and moving the rail cargo out of the port using the port locomotive. The processing time associated with the rail locomotive (its return time) is very high in the interchange yard area. So irrespective of the number of rail cargo units, the activity time will always be high. Due to this reason, the interchange yard area that is particularly influenced by the locomotive return time emerges as a strong contender for the primary bottleneck slot.

Convoy Construction Area: The port area from which the cargo scheduled to leave is assembled. Military unit drivers are utilized in this port area. The activity in this area takes place as several discontinuous intervals since because the cargo does not continuously flood this port area. It arrives at discrete intervals depending upon the rate at which the previous port areas have processed the cargo.

Table 5.7 shows the activity intervals for all the port areas in batch 1 ships

Port Area	Start Activity	End Activity	Total Active Time (days)
Staging	0.5	4.55	4.05
Truck Loading Area	3.64	3.67	0.03
	3.86	3.89	0.03
	3.90	3.94	0.04
	4.45	4.50	0.05
	4.79	4.88	0.09
Rail Spur	14.15	14.16	0.00
Interchange Yard Area	16.01	18.39	19.65
Convoy construction Area	0.54	0.95	0.41
	1.05	1.10	0.05
	1.11	1.14	0.03
	1.18	1.23	0.05
	1.27	1.29	0.02
	1.33	1.36	0.03
	2.25	2.28	0.03
	2.33	2.38	0.05
	2.60	2.63	0.03

	2.66	2.69	0.03
	2.75	2.80	0.05
	2.86	2.90	0.04
	3.00	3.03	0.03
	3.30	3.32	0.02
	3.43	3.46	0.03
	3.57	3.60	0.03
	4.17	4.20	0.03
	4.32	4.43	0.11
	4.49	4.51	0.02
	4.56	4.59	0.03
	4.75	4.78	0.03
	4.92	4.95	0.03
	5.08	5.11	0.03
	5.22	5.24	0.02
	5.61	5.65	0.04
	5.71	5.74	0.03
	6.02	6.04	0.02
	6.08	6.32	0.24
	6.42	6.44	0.02
	6.46	6.51	0.05
	6.54	6.60	0.06
	7.11	7.14	0.03
	7.21	7.23	0.02
	7.29	7.53	0.24
	7.64	7.67	0.03
Helicopter Reassembly	0.55	0.57	0.02
	0.58	0.6	0.02
	0.61	0.63	0.02
	0.65	0.66	0.01
	0.68	0.69	0.01
	0.71	0.73	0.02
	0.74	0.76	0.02
	0.77	0.79	0.02
	0.8	0.82	0.02
	0.83	0.85	0.02
	0.86	0.88	0.02
	0.89	0.91	0.02
	0.92	0.94	0.02
	0.95	0.97	0.02
	0.98	1	0.02
	1.01	1.03	0.02
	1.05	1.06	0.01
	1.08	1.09	0.01
	1.11	1.13	0.02
	1.14	1.16	0.02
	1.17	1.19	0.02
	1.2	1.22	0.02

	1.23	1.25	0.02
	1.26	1.28	0.02
	1.29	1.51	0.22
	1.53	1.55	0.02
	1.56	1.58	0.02
	1.59	1.61	0.02
	1.62	1.64	0.02
	1.65	1.67	0.02
	1.69	1.7	0.01
	1.71	1.73	0.02
	1.74	1.76	0.02
	1.77	1.79	0.02
	1.81	1.83	0.02
	1.84	1.85	0.01
	1.87	1.89	0.02
	1.9	1.91	0.01
	1.93	1.95	0.02
	1.96	1.98	0.02
	1.99	2.01	0.02
	2.03	2.04	0.01
	2.05	2.07	0.02
Helicopter Takeoff	2.49	2.72	0.23
	2.76	2.78	0.02
	2.79	2.80	0.01
	2.82	2.84	0.02
	2.85	2.87	0.02
	2.91	2.93	0.02
	2.94	2.96	0.02
	2.98	2.99	0.01
	3.00	3.02	0.02
	3.06	3.08	0.02
	3.09	3.11	0.02
	3.12	3.14	0.02
	3.15	3.17	0.02
	3.21	3.23	0.02
	3.25	3.27	0.02
	4.18	4.20	0.02
	4.25	4.26	0.01
	4.27	4.29	0.02
	4.33	4.35	0.02
	4.39	4.41	0.02
	4.42	4.44	0.02
	4.48	4.50	0.02
	4.52	4.54	0.02
	4.58	4.60	0.02
	4.61	4.63	0.02
	4.67	4.69	0.02
	4.72	4.74	0.02

	4.78	4.80	0.02
	4.81	4.83	0.02
	4.86	4.88	0.02
	5.14	5.16	0.02
	5.19	5.21	0.02
	5.20	5.22	0.02
	5.25	5.27	0.02
	5.31	5.33	0.02
	5.32	5.34	0.02
	5.37	5.39	0.02
	5.42	5.43	0.01
	5.43	5.45	0.02
	5.48	5.49	0.01
	5.51	5.52	0.01
	5.52	5.54	0.02
	5.55	5.57	0.02
	5.58	5.60	0.02
	5.60	5.61	0.01
	5.63	5.64	0.01
	5.64	5.66	0.02
	5.67	5.69	0.02
	5.70	5.72	0.02
	5.72	5.74	0.02
	5.75	5.77	0.02
	5.78	5.80	0.02
	5.79	5.81	0.02
	5.82	5.84	0.02
	5.85	5.87	0.02
	5.87	5.89	0.02
	5.90	5.91	0.01
	5.93	5.94	0.01
	5.94	5.96	0.02
	5.97	5.99	0.02
	6.00	6.02	0.02
	6.02	6.03	0.01
	6.05	6.06	0.01
	6.07	6.09	0.02
	6.09	6.32	0.23
	6.33	6.34	0.01
	6.34	6.35	0.01
	6.37	6.39	0.02
	6.40	6.42	0.02
	6.43	6.44	0.01
	6.46	6.47	0.01
	6.49	6.50	0.01
	6.52	6.53	0.01
	6.55	6.56	0.01
	6.58	6.59	0.01

	6.60	6.62	0.02
	6.63	6.65	0.02
	6.66	6.68	0.02
	0.55	0.57	0.02
	0.58	0.6	0.02
	0.61	0.63	0.02
	0.65	0.66	0.01
	0.68	0.69	0.01
	0.71	0.73	0.02
	0.74	0.76	0.02
	0.77	0.79	0.02
	0.8	0.82	0.02
	0.83	0.85	0.02
	0.86	0.88	0.02
	0.89	0.91	0.02
	0.92	0.94	0.02
	0.95	0.97	0.02
	0.98	1	0.02
	1.01	1.03	0.02
	1.05	1.06	0.01
	1.08	1.09	0.01
	1.11	1.13	0.02
	1.14	1.16	0.02
	1.17	1.19	0.02
	1.2	1.22	0.02
	1.23	1.25	0.02
	1.26	1.28	0.02
	1.29	1.51	0.22
	1.53	1.55	0.02
	1.56	1.58	0.02
	1.59	1.61	0.02
	1.62	1.64	0.02
	1.65	1.67	0.02
	1.69	1.7	0.01
	1.71	1.73	0.02
	1.74	1.76	0.02
	1.77	1.79	0.02
	1.81	1.83	0.02
	1.84	1.85	0.01
	1.87	1.89	0.02
	1.9	1.91	0.01
	1.93	1.95	0.02
	1.96	1.98	0.02
	1.99	2.01	0.02
	2.03	2.04	0.01
	2.05	2.07	0.02

Table 5.7: Activity table for all port areas

Final Analysis: By utilizing all the activity times from the above tables, the activity chart in Figure 5.8 can be plotted. It consists of all the active intervals of all the port areas associated with the ships in batch 1.

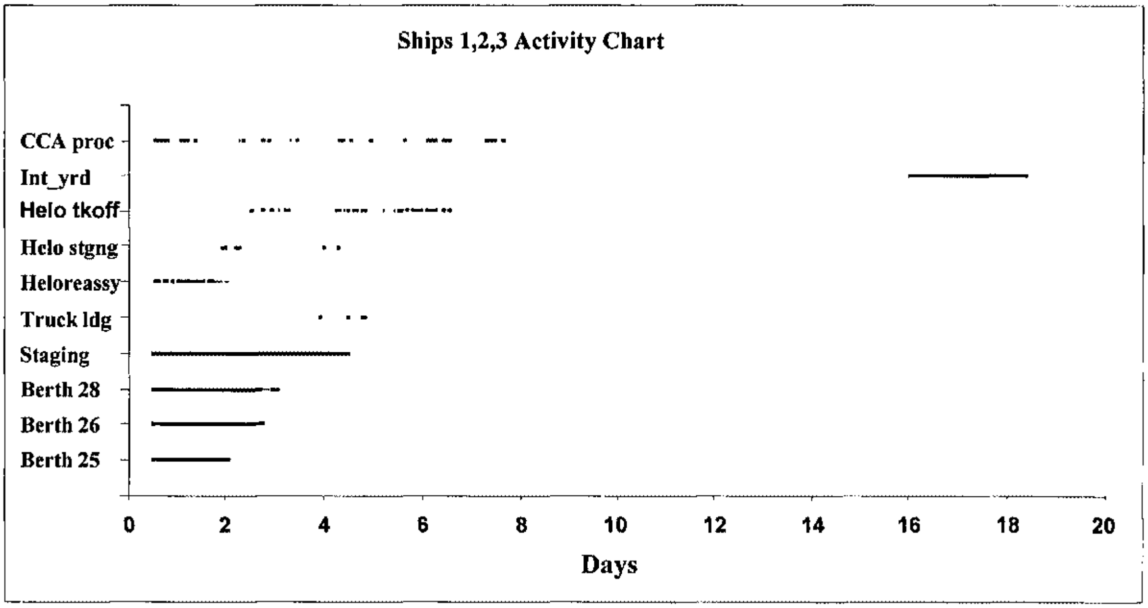

Figure 5.8: Batch 1 ships overall activity chart

The important activity times to be considered in the above Figure are that of Berth 26, Staging and the Interchange yard because these dominate all other activity intervals by being continuous over long period of time. Table 5.9 gives the information about the activity times of these port areas.

Port Area	Start Time	End Time	Total Days Active
Berth 26	0.49	2.28	1.79
Staging	0.50	4.55	4.05
Interchange Yard	16.01	18.39	2.39

Table 5.9: Major activity time in batch 1 ships

For the ships in batch 1, we have:

Start of activity = 0.30 days.

End of activity = 18.39 days.

By using equations 4.1 and 4.2

Sole bottleneck % for berth 26 = (0.50-0.49)*100/ (18.39-0.30) = 0.05 %.

Shifting bottleneck % for berth 26/staging = (2.28-0.50)*100/ (18.39-0.30) = 9.83 %.

Sole bottleneck % for staging area = (4.55-2.28)*100/ (18.39-0.30) = 12.54%.

Sole bottleneck % for interchange yard area = (18.39-16.01)*100/ (18.39-0.30) = 13.15%.

Figure 5.10 plots the graph for the above-obtained percentages.

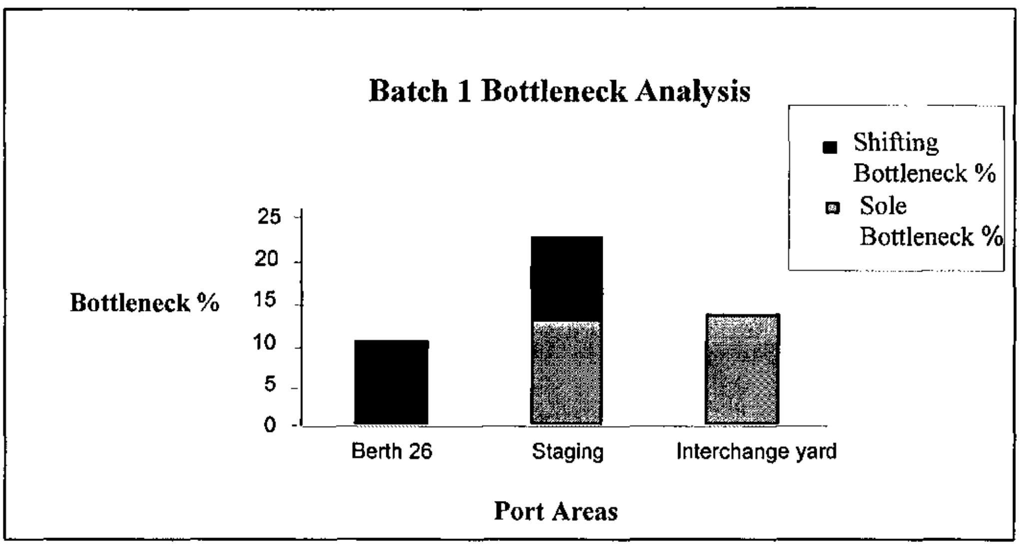

Figure 5.10: Bottleneck percentages for batch 1 ships

Even though the staging area has a total bottleneck percentage of 22.37%, the sole bottleneck percentage is less than that of the interchange yard area. Hence, the interchange yard, which has a sole bottleneck period of 2.39 days, is the primary bottleneck for batch 1 ships with the staging area being the secondary bottleneck.

5.4.2 Batch 2 Ships Analysis

The same methodology for the event classification as well as the active time determination as was described for the batch 1 ships will be followed until the batch 11 ships. So from now onwards the stress will be on the important events rather than on the data.

The activity for the batch 2 ships starts on day 2. There are 3 ships in batch 2 and they have a total of 3887 cargo units (64 helicopters via air, 806 containers via road, 3016 vehicles via road, and 1 vehicle via rail). Since there are so many cargo units, the time spent in the berthing area is high, because it requires utilizing large number of resources like the cranes, ramps and line handlers.

Berths 25 and 26 have continuous activity intervals compared to Berth 28 since the ships arrive on these berths earlier than on berth 28 and since resources like the ramps are allocated to one ship and after that ship is unloaded they will be allocated to another ship, which means a break in the activity for the ship arriving later, since it has to wait for the unavailable resources.

Even the staging area was found to be busy and has considerable overlapping activity periods with the berthing areas. There are sufficient numbers of helicopter tractors to operate upon the helicopter parts, which are operated upon in several discontinuous intervals of time owing to the small number of helicopters compared to the overall cargo count.

The processing time associated with cargo that has to be transported through rail will usually be high due to the large amount of processing times associated with the rail cargo in the interchange yard area. In this batch there are few cargo units to be transported via rail hence a considerable amount of continuous activity can be seen in the interchange yard area. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.11 can be obtained showing the sole and shifting bottleneck percentages for the batch 2 ships.

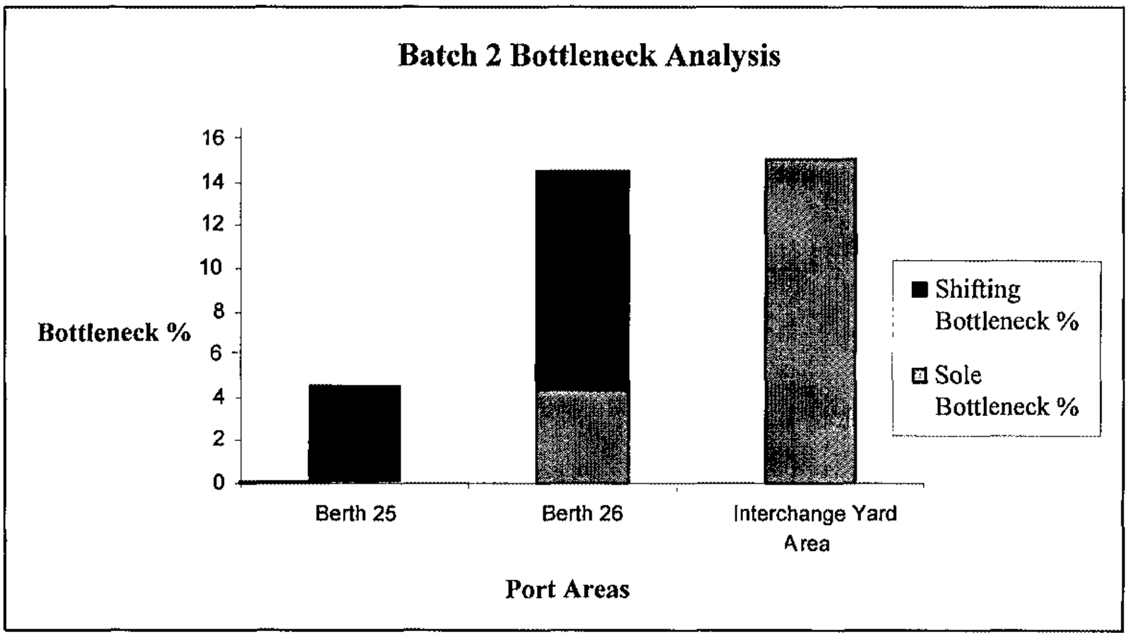

Figure 5.11: Bottleneck percentages for batch 2 ships

Clearly, the interchange yard is the primary bottleneck. Berth 26 is the secondary bottleneck and berth 25 is the tertiary bottleneck.

5.4.3 Batch 3 Ship Analysis

The activity for the batch 3 ships starts on day 5.10. There are 3 ships in batch 3 and they have a total of 2740 cargo units (679 containers via road, 2061 vehicles via road). This batch of ships has the same features as that of batch 2 ships; hence even the behavior can be seen to be similar to that of batch 2 ships. The ship on berth 25 arrives much earlier than the ships on berths 26 and 28; hence it grabs all the available resources and causes discontinuous activity on the other two berths. The staging area as was in the previous case has significant activity time, a majority of which does not overlap with other activity intervals (for batch 2 ships this was a little bit different). In this batch there are no cargo units to be transported via rail hence no activity intervals for the rail associated areas. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.12 can be obtained showing the sole and shifting bottleneck percentages for the batch 3 ships.

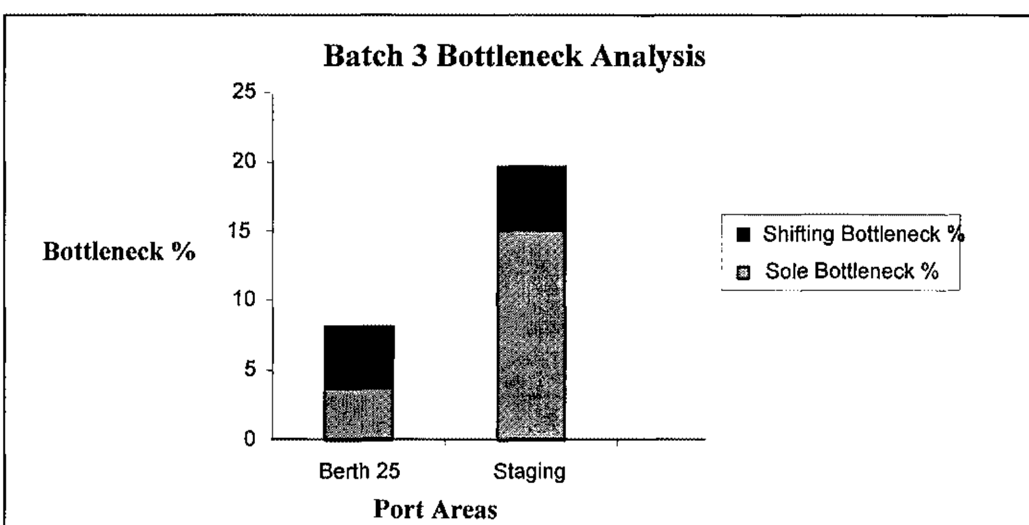

Figure 5.12: Bottleneck percentages for batch 3 ships

As can be seen the staging area becomes the primary bottleneck owing to its high sole bottleneck % followed by berth 25. All the batches of ships contribute in deciding the primary bottleneck, although some of them have a major contribution and some of them have a smaller contribution depending upon several factors like the type of cargo they carry, the tome of their arrival, the resources available to them and so on.

5.4.4 Batch 4 Ship Analysis

The batch 4 ships arrive in the anchorage area on day 7.5. There are 3 ships in batch 4 and they have a total of 3038 (0 helicopters, 155 containers, 2883 vehicles) cargo units. Due to the large number of cargo units, significant overlapping activity takes place in the berthing as well as the staging area, with the staging area having the majority of the activity. In this batch there are no cargo units to be transported via rail hence no activity intervals for the rail associated areas. Since most of the cargo has to be moved through road, there is significant activity in the convoy construction area. As a result of this the convoy construction area as well the staging area become the major contenders for the primary bottleneck slot. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.13 can be obtained showing the sole and shifting bottleneck percentages for the batch 3 ships.

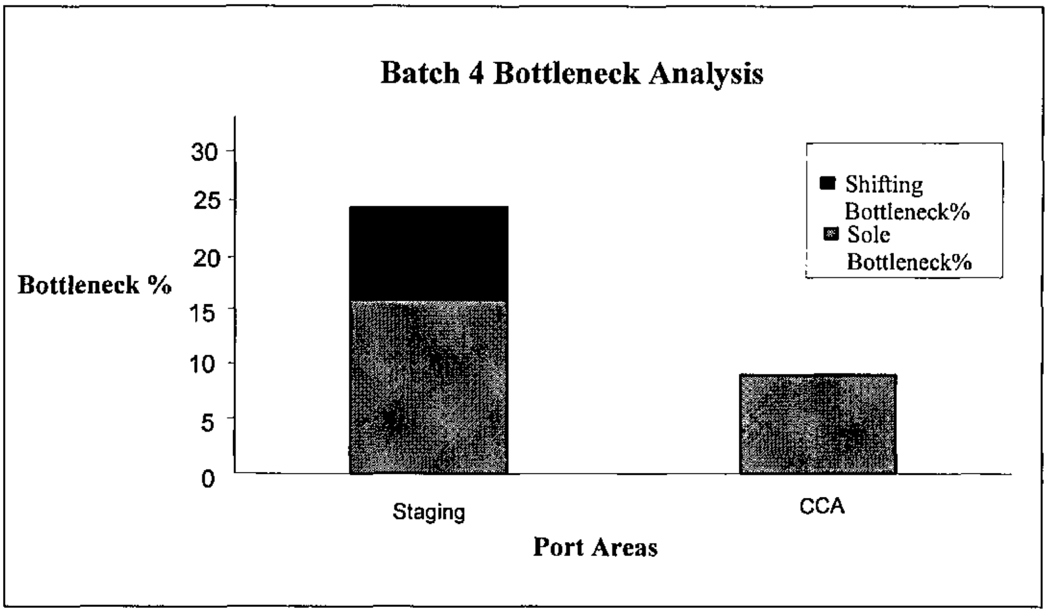

Figure 5.13: Bottleneck percentages for batch 4 ships

5.4.5 Activity chart for batch 5 ships

The activity for the batch 5 ships starts on day 10.08 when the first ship in this batch arrives on the berthing area. There are 3 ships in batch 5 and they have a total of 2134 (72 helicopters via air, 204 containers via road, and 1824 vehicles via road, 34 vehicles via rail) cargo units. This batch of ships has the same features as that of batch 2 ships; hence even the behavior can be seen to be similar to that of batch 2 ships. The ship on berth 25 arrives much earlier than the ships on berths 28 and 30; hence it grabs all the available resources and causes discontinuous activity on the other two berths. The staging area has a significant activity time due to the large number of cargo units. In this batch there are 34 cargo units that have to be transported via rail hence a significant and continuous activity interval can be found in the interchange yard area. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.14 can be obtained showing the sole and shifting bottleneck percentages for the batch 5 ships.

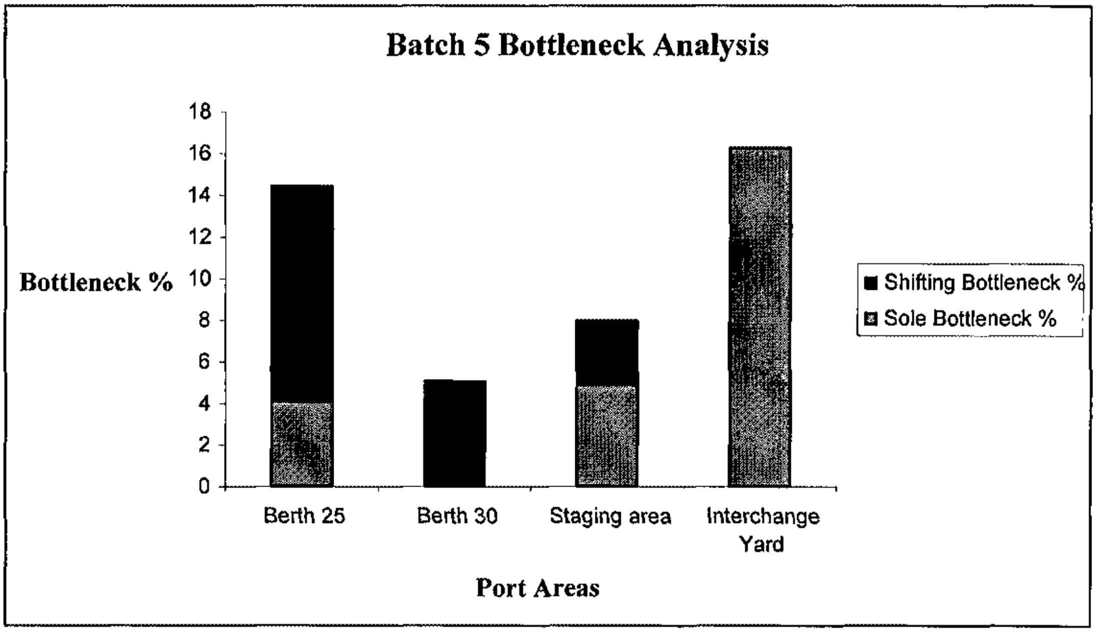

Figure 5.14: Bottleneck percentages for batch 5 ships

The interchange yard area is the primary bottleneck followed by the staging area and berth 25.

5.4.6 Activity chart for batch 6 ships

The activity for the batch 6 ships starts on day 12.48 when the first ship in this batch arrives on the berthing area. There are 2 ships in this batch and they have a total of 1199 cargo units (166 containers via road, 676 vehicles via road, and 357 vehicles via rail). There are not many cargo units as compared to the earlier batches of ships, but they do require processing on the berthing and staging area. This results in a significant activity overlap between the berthing and staging areas. The staging area has more continuous activity interval compared to the berthing areas in this case since the berthing areas have better resource utilization in this case owing to the small number of cargo units. There are a significant number of cargo units that require rail transportation; hence there are activity intervals for the rail spur area as well as the interchange yard area. These activity intervals overtake the activity interval times of the berthing areas. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.15 can be obtained showing the sole and shifting bottleneck percentages for the batch 6 ships.

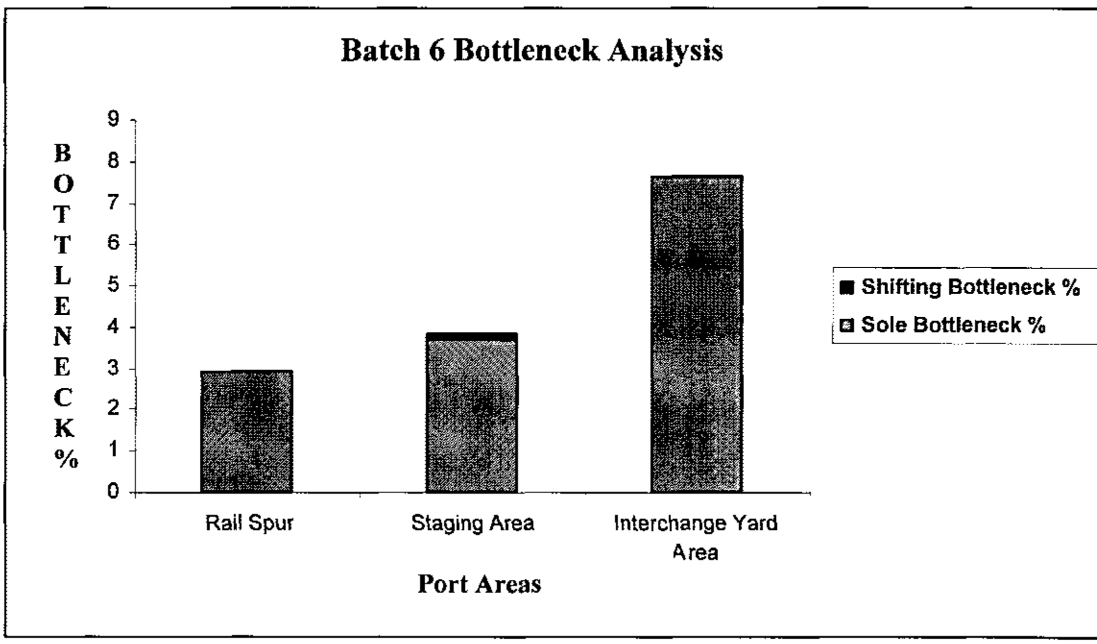

Figure 5.15: Bottleneck percentages for batch 6 ships

The interchange yard area is the primary bottleneck followed by the staging area and the rail spur area.

5.4.7 Activity chart for batch 7 ships

The activity for the batch 7 ships starts on day 12.88 when the first ship in this batch arrives on the berthing area. There are 3 ships in this batch and they have a total of 3082 cargo units (16 helicopters via air, 191 containers, and 2170 vehicles via road, 705 vehicles via rail). This batch has a large amount of cargo associated with it. But no continuous activity can be seen in these areas, since there is an overlap of activity for this batch of ships with the previous batch that results in discontinuous activity due to the unavailability of required resources. So there are several discontinuous activity intervals in the staging as well as berthing areas in this case. There are a significant number of cargo units that require rail transportation; hence there are activity intervals for the rail spur area as well as the interchange yard area. These activities do not overlap with the previous batch's activities; hence we have continuous activity intervals. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.16 can be obtained showing the sole and shifting bottleneck percentages of batch 6 ships.

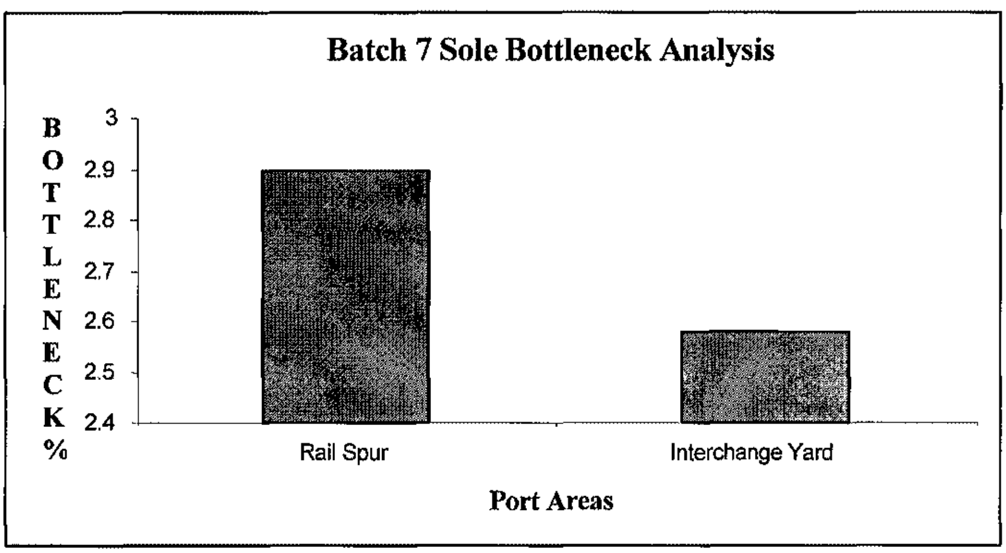

Figure 5.16: Bottleneck percentages for batch 7 ships

5.4.8 Activity chart for batch 8 ships

The activity for the batch 8 ships starts on day 17.28 when the first ship in this batch arrives on the berthing area. There are 3 ships in this batch and they have a total of 3635 cargo units (443 containers via road, 2994 vehicles via road, and 198 vehicles via rail). The activity for this batch is just similar to that of batch 6 ships; hence a similar outcome can be seen. Figure 5.17 shows the graph for batch 8 ships.

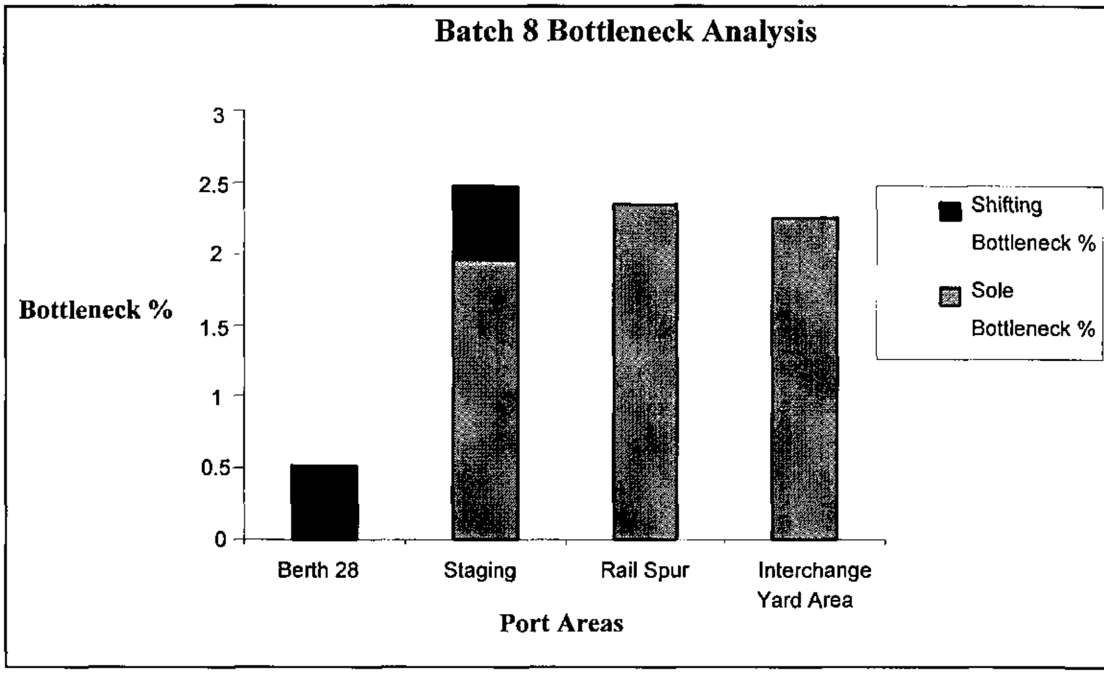

Figure 5.17: Bottleneck percentages for batch 8 ships

As can be seen the rail spur area is the primary bottleneck followed by the interchange yard area, staging area and the berth 28 area.

5.4.9 Activity chart for batch 9 ships

There are a total of 2368 cargo units (454 containers via road, 1693 vehicles via road, 6 vehicles via rail, and 23 helicopters via air) in this batch of ships. Several cargo units require rail transportation. The berth 25 has a significant activity interval starting from 19.83 to day 30.51. This long activity interval overshadows the other berthing areas as well as the staging area activity. Since there is rail cargo, the rail spur as well as the interchange yard area also have significant activity intervals. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.18 can be obtained showing the sole and shifting bottleneck percentages for the batch 9 ships. Berth 25 is the primary bottleneck followed by the interchange yard and the rail spur area.

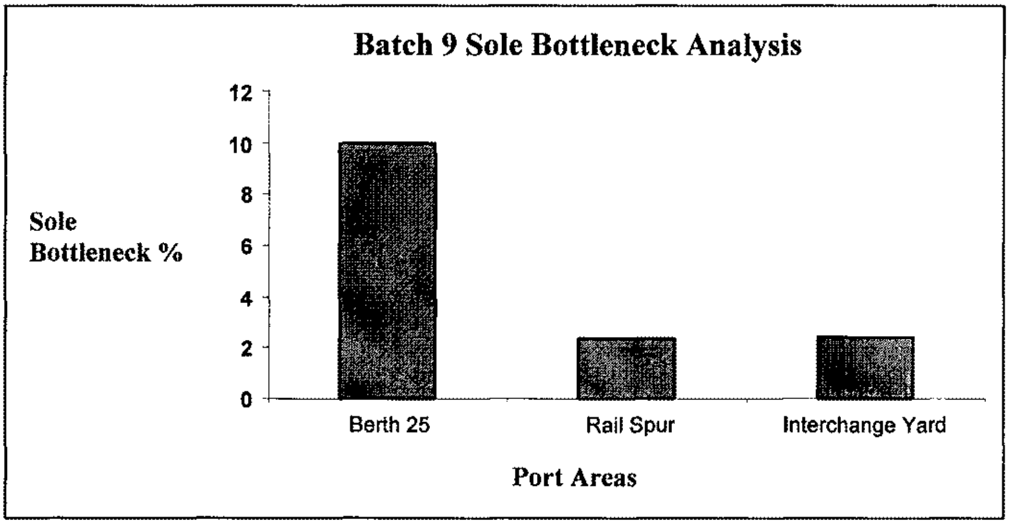

Figure 5.18: Bottleneck percentages for batch 8 ships

5.4.10 Activity chart for Batch 10 ships

There are a total of 4009 cargo units (3983 vehicles via road, 3 vehicles via rail, and 23 helicopters via air) in this batch of ships. Several cargo units require rail transportation. Berth 29 has a significant activity interval starting from day 22.08 and ending on day 32.91. This long activity interval overshadows the other berthing areas as well as the staging area activity. Since there is rail cargo, the rail spurs as well as the interchange yard area also have significant activity intervals. After analyzing the data and the outcomes using the equations (4.1) and (4.2) from Chapter 4, Figure 5.19 can be obtained showing the sole and shifting bottleneck percentages for the batch 10 ships.

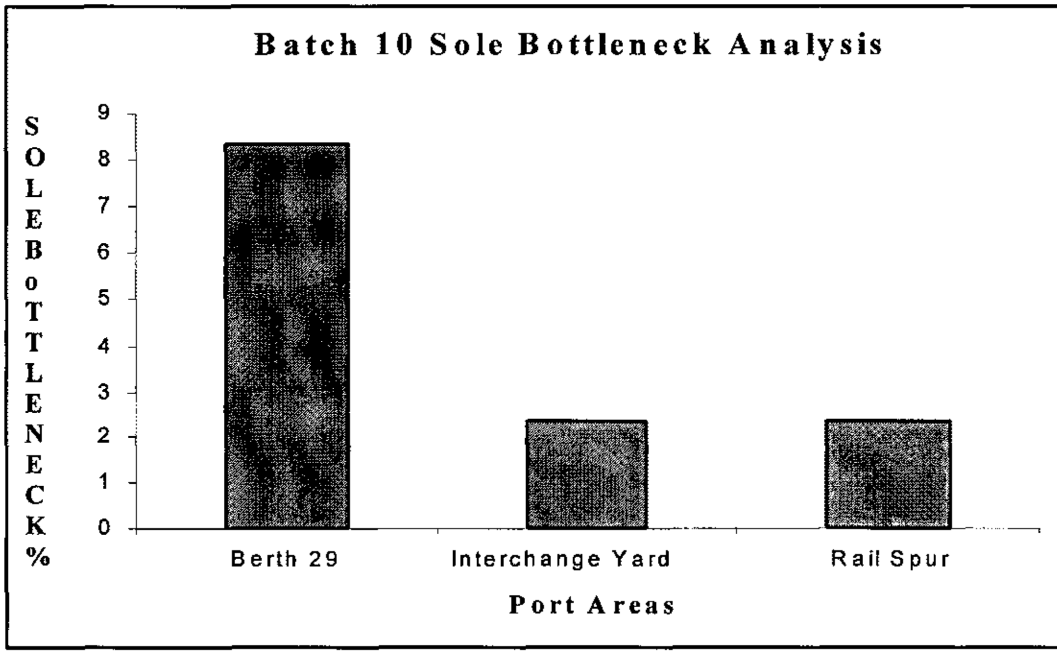

Figure 5.19: Bottleneck percentages for batch 10 ships

Berth 29 is the primary bottleneck owing to its large activity interval, followed by the interchange yard area and the rail spur area.

5.4.11 Activity chart for batch 11 ships

There is a single ship in batch 11. The ship arrives on day 24.30 into the anchorage area. There are 804 cargo units on this ship (all vehicles via road). The ship unloading starts on day 32.91 in the berthing area. The berthing area and staging area have overlapping activity intervals with the staging area having the major portion of it. There is no rail cargo in this case; hence all the cargo has to be transported through the trucks. As a result, the convoy construction area has a considerable activity interval. Figure 5.20 plots the chart for the bottleneck periods for batch 11 ships.

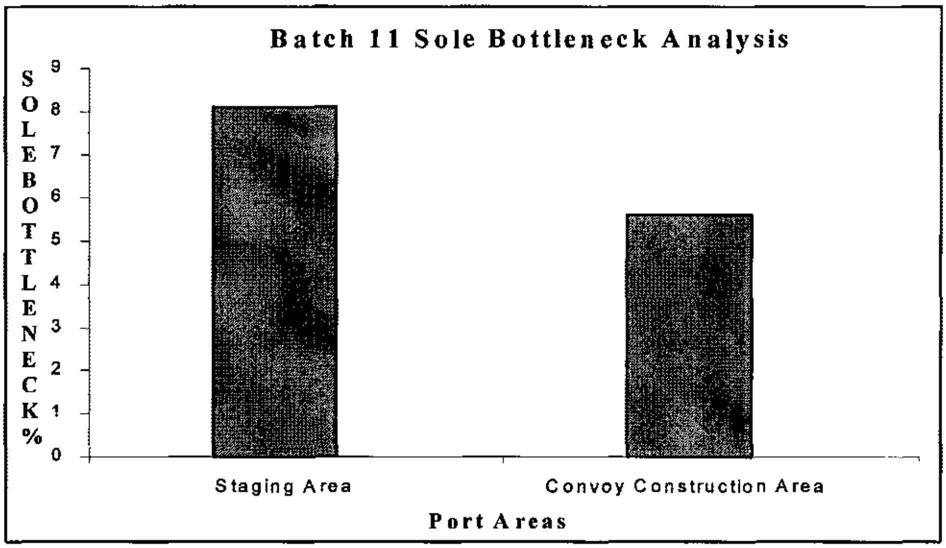

Figure 5.20: Bottleneck percentages for batch 11 ships

5.5 Verification of the effectiveness of the above results

Table 5.21 summarizes the outcome of all the 11 batches.

Batch	Primary Bottleneck	Secondary Bottleneck	Tertiary bottleneck
Batch 1	Interchange Yard	Staging Area	-
Batch 2	Interchange Yard	Berthing Area	-
Batch 3	Staging Area	Berthing Area	-
Batch 4	Staging Area	CCA	-
Batch 5	Interchange Yard	Staging Area	Berth Area
Batch 6	Interchange Yard	Staging Area	Rail Spur
Batch 7	Rail Spur	Interchange Yard	-
Batch 8	Rail Spur	Interchange Yard	Staging Area
Batch 9	Berth Area	Interchange Yard	Rail Spur
Batch 10	Berth Area	Interchange Yard	Rail Spur
Batch 11	Staging Area	CCA	-

Table 5.21: Summary table for all 11 batches

The main aim of this section is to verify if the results point to the primary bottleneck. From table 5.21 it is clear that among all the eleven batches, the dominant primary bottleneck is the interchange yard area for the batches, which have cargo-requiring transportation via rail. Similarly for batches, which do not have rail cargo, the staging area becomes the primary bottleneck. Fine-tuning the processes as well as the resources in these bottleneck areas carries out the verification process. The first step in this process is fine tuning the interchange yard area parameters and compare the result with the initial outcome of the simulation.

(a) Massaging Interchange Yard Area parameters:

Initially without fine-tuning the interchange yard area resources and process time the output from the simulation was: 31173 Cargo units cleared in 130.12665 days. After increasing the number of locomotives to 2, the output from the simulation changed to: *31173 Cargo units cleared in 73.67708 days.*

As can be seen there is a drastic change in the cargo clearance throughput. The next step is to change the port locomotive return time to half of its original time and view the output after halving the Port Locomotive return time the following throughput change was observed, (maintaining the number of locomotives at 1): *31173 cargo units cleared in a total of 73.67708 days.* In reality this time cannot be changes but for verification of the correctness of the shifting bottleneck method it is assumed that it can be changed.

So the percentage improvement in the throughput for this scenario is:

100-((73.67*100)/130.12)) = 43.38 %

Both the changes produced the same amount of improvement in the throughput. The next step in the verification process is to massage the other deduced bottleneck port area and find out what sort of improvement it produces in the throughput.

(b) Massaging Staging Area parameters:

By doubling the number of available resources on the staging area i.e. chassis, PSA personnel and container handlers the following output was: *31173 cargo units cleared in 125.18136 days.* So the percentage improvement in the throughput for this scenario is:

100-((125.18*100)/130.12)) = 3.79%.

Increasing the resources further in the staging area does not give further improvement in the throughput. As can be seen there is a great difference between the throughput improvement due to the primary and secondary bottleneck, which clearly points out that the primary bottleneck for this base scenario is the interchange yard area.

The following result was obtained when combining the resource changes made in the interchange yard area as well as the staging area: 31173 cargo units cleared in 71.18371 days. So the percentage improvement in the throughput for this scenario is:

100-((71.18*100)/130.12)) = 45.29%.

From the above results, it is clear that the shifting bottleneck analysis can successfully point out the bottleneck areas in the given scenario for CPortS simulation. By massaging the resource as well as appropriate process times in those bottleneck areas we can obtain the best possible improvement in the throughput.

Chapter 6

Conclusion

6.1 Introduction

CPortS, which is a transportation logistics simulation, simulates the movement of military cargo through various areas in a seaport. The movement of a large number of cargo units in the simulated seaport provides a good model for understanding the actual seaport traffic movement. The huge amounts of cargo require large amount of resources to move through the various port areas. The large amount of cargo traffic in the seaport may lead to congestion in several areas leading to delay in the military operations. This may be caused by the fact that one of the port areas or a component in a port area has been over utilized thus becoming the bottleneck for the seaport. The shifting bottleneck methodology described in this thesis successfully detects such bottlenecks in the system by analyzing the cargo movement in the port areas over several active durations and by ranking them (port areas) according to those activity times. Although this methodology has been applied to the CPortS model, with appropriate modifications it can be applied to other commercial applications, which have a large amount of traffic. This chapter details the achievements of this thesis work and suggests some future enhancements.

6.2 Achievements

The work described in this thesis document has achieved the following:

(1) The thesis work has successfully demonstrated the shifting bottleneck methodology's applicability to the CPortS model by considering an example scenario provided by the sponsor (MTMCTEA). The cargo movement of this

scenario has been subjected to bottleneck analysis followed by a verification phase to prove that the shifting bottleneck detection methodology can be applied to the CportS model to obtain useful results.

(2) The bottleneck detection methodology described in this thesis successfully identified the primary and secondary bottlenecks, which other bottleneck detection methodologies were unable to distinguish. The thesis work also verified the validity of this bottleneck ranking by massaging these bottlenecks and finding whether the throughput enhancements that they provided were in accordance with their ranking.

(3) The bottleneck detection methodology described in the thesis cuts through the huge output data (several hundreds of pages) to obtain the useful activity time by caul study of the cargo movement in the various port areas as well as the port areas activity during that time.

(4) This methodology can be successfully applied for any number of cargo units i.e. there is no limitation on the number of cargo units. Since the major aspect of this methodology is the activity time of the port areas and their components, large number of cargo units can in fact help in easily establishing the bottleneck, since there will be a lot of activity going on in the port area.

6.3 Future enhancements

The following enhancements can be made to the thesis work done

(1) Currently the bottleneck detection strategy has been performed in a partly automated environment using excel sheets and text files. Since the amount of data

to be processed is huge (usually thousands of cargo pieces), if a tool is designed to automate the bottleneck detection methodology the burden on the analyst will be reduced.

(2) The bottleneck detection methodology has been applied to the CportS simulation model in this thesis. Since this model is a traffic (cargo traffic) intensive application, it can be extended to other models, which have huge amounts of traffic, since the methodology has the characteristics to be applied to any traffic intensive congestion-causing environment.

(3) The bottleneck detection strategy discussed in this thesis concentrates on the activity times of the various port areas and their components; this helps in identifying the bottleneck. Similarly if a counter bottleneck detection strategy using the idling times is developed then some other important and interesting details about the traffic pattern can be deduced which can prove useful for further reducing the bottlenecks in the system.

(4) It would be helpful if some modules were embedded in the CportS code that can filter out the active and inactive times in the seaport areas, and provide graphical results to the analyst ensuring quick cargo clearance.

Overall, the methodology has successfully demonstrated its applicability in pin pointing the bottlenecks in the CPortS model. It would be useful to have an automated tool detecting the bottlenecks since it can be very useful in times of crisis.

References

[1] James F. Leathrum, Jr. and Taylor Frith, "A Reconfigurable Object Model for Port Operations," *Proceedings of the Summer Computer Simulation Conference,* SCSC2000, Vancouver, Canada, 2000, pp.603-608.

[2] Christopher Roser, Masaru Nakano, Minoru Tanaka, "A Practical Bottleneck Detection Method," *Proceedings of the 2001 Winter Simulation Conference"* pp.949-953

[3] Ron Lee," Performance Analysis, Isolating the Real Bottleneck in a System", Novell Research, Novell Application Notes, January 1996, pp 73-74.

[4] Johannes Luthi, Gabriel Kotsis, Shikharesh Majumdar, Gunter Haring," Performance Analysis using Queuing Network Models with variabilities and Uncertainties in Workload" *Parallel Computing, Vol. 22 (13) 1997,* pp. 1789-1806.

[5] Luthi, J,"Interval Matrices for the Bottleneck Analysis of Queuing Network Models with Histogram-Based Parameters" In *IEEE International Computer Performance & Dependability Symposium*, Durham, NC, USA: IEEE Computer Society Press, 1998, pp, 142-151.

[6] Delp, D., Si, J., Hwang, Y., and Pei, B." A Dynamic System Regulation Measure for Increasing Effective Capacity: the X-Factor Theory," *IEEE/SEMI Advanced semiconductor Manufacturing Conference and Workshop (ASMC)*, ed., Munich, Germany. 2003.

[7] Christopher Roser, Masaru Nakano, Minoru Tanaka, "Comparison of Bottleneck detection methods for AGV systems," *Proceedings of the 2003 Winter Simulation Conference,* Arlington, VA, pp 1192-1198.

[8] Russ Blake, John S. Breese," Automatic Bottleneck Detection" *Technical Report, MSR-TR-95-10,* Redmond, WA, October 1995,pp 1-21.

[9] Jingshan Li, Semyon M.Meerkov,"Bottlenecks with respect to the time performance in pull serial production lines" *Mathematical Problems in engineering Volume 5*,pp 479-498.

[10] T. Frith, Leathrum, J., R. Mathew et al. 2002. "CPORTS Process Model,"*http://www.vmasc.odu.edu/portsim/CPorts_Process_Model.pdf*

[11] *Microsoft Visio 2000,* Microsoft Corp., Redmond, WA.

CURRICULUM VITA

FOR

SREEKALYANA CHAKRAVARTHY KAJULURI

DEGREES:

- Master of Science (Computer Engineering), Old Dominion University, Norfolk, VA, December 2004.

- Bachelor of Engineering (Information Science and Engineering), Bangalore College of Engineering and Technology, Bangalore University, Bangalore, India, December 2001.

PART TIME EMPLOYMENT:

- Software Developer at Investors Security Inc. at Suffolk, Virginia (April 2004-Present.

- Research Assistant at the Virginia Modeling Analysis and Simulation Center (VMASC), working on *CPortS* (Sea Port Simulation) Testing and Bottleneck Analysis; projects sponsored by the Military Traffic Management Command Transportation Engineering Agency (MTCTEA) of the US Army. (August 2002 – December 2003).

- Software Developer at Howmet Castings at Hampton, Virginia (April 2003-March 2003).