Dublin City University

SCHOOL OF MECHANICAL AND MANUFACTURING ENGINEERING

Constant Flow Management

Investigating manufacturing flow variability

by

Hsiao-Hui Chung

This thesis is submitted in accordance with the requirements of Dublin City University for the degree of Doctor of Philosophy

November 2011

Supervisors: Dr. John Geraghty and Dr. Paul Young

DISCLAIMER

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy (PhD) is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

ID No.: 56102763

Date:

TABLE OF CONTENTS

Constant Flow Management

Investigating manufacturing flow variability

Hsiao-Hui Chung

ABSTRACT

This project investigates the manufacturing flow variability in order to stabilize the factory process flow. Nowadays, in manufacturing production lines and particularly in modern front end semiconductor lines, processes and equipments are very complex. Any disturbance of the process creates variability in the line, and causes substantial losses in productivity for manufacturing corporations. These disturbances are unpredictable, difficult to control and result in long recovery times.

Variability occurring in a production system disturbs the whole processing flow and results in long product cycle times. Hence, a range of sources of variability was determined from the literature and analyzed. This lead with the cooperation of factory managers to the development of four main objectives:

(1) Determine a proper metric to measure the variability in the production system.
(2) Determine the effect of batching and tool availability on the process flow.
(3) Understand the interaction between operations.
(4) Develop a release strategy in order to stabilize the production flow.

First, from the observation of real production data, a difference metric was developed and operations creating or removing variability were identified. The propagation of variability can be followed using a correlation coefficient. Nevertheless, the data were not detailed enough to explain the origin of the variability. Consequently, several simulation models were created to investigate variability.

The simulations' results show that the release strategy should be adjusted as a function of batch, tool availability and constraint parameters, in order to stabilize the flow of items in the line and control cycle time and cycle time variability. The notion of critical availability is introduced and defined. Improvement of the line performance is obtained through a tighter control of the availability of high capacity operations.

This lead to the development of a new hybrid push pull release strategy, named CONFLOW, to regulate the flow of items reaching the constraint operation. CONFLOW was tested under many simulating conditions (batching, parallel processing, and different line length). Compared to a push system, CONFLOW release strategy results, into significant improvement (up to 80%) in cycle time, cycle time standard deviation and WIP level at the cost of 13% reduction in throughput. CONFLOW performances were compared to common TOC strategies (SA and DBR). The results are encouraging. In the specific conditions considered, CONFLOW performances are similar to SA and slightly better than DBR.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

T_e	Mean Processing Time of items in the system
T_{e_i}	Processing time of item i
$T_{e_i}^j$	Processing time of item i at Operation j
$T_{q_i}^j$	Queue time of item i in Buffer j
σ_{PT}	Standard deviation of processing time
r_{a_j}	Arrival rate at Operation j, or the departure rate from the buffer of the preceding Operation j-1
CT_i	Cycle time of item i
r_{d_j}	Mean Departure Rate
t_{d_j}	Mean Inter-Departure Rate
t_{e_j}	Mean Processing Time of items at Operation j
t_q	Mean Queuing Time
t_q^j	Mean Queuing Time of items at Operation j
m_j	Number of machines at operation j
$\overline{CT}$	Mean Cycle Time
C_d	Coefficient of Variation of Inter-Departure Time
C_q	Coefficient of Variation Queue Time
t_d	Mean Inter-Departure Time
σ_{CT}	Standard Deviation Cycle Time
σ_d	Standard Deviation Inter-Departure Time
σ_q	Standard Deviation Queue Time
σ_{PT}	Standard Deviation Processing Time
A	Availability
AT	Arrival Time
BSCS	Base Stock Control System
BPM	Batch Processing Model
C	Operation/Machine Capacity
CA	Critical Availability
CC	Correlation Coefficient
Ccon	Constraint Capacity
CDR	Coefficient of Departure Rate Variability
CONWIP	Constant WIP
CONFLOW	Constant Flow
CR	Critical Ratio
CT	Cycle Time
CV	Coefficient of Variation
CVCT	Coefficient of Variation Cycle Time
CVI-AT	Coefficient of Variation Inter-Arrival Time
CVI-DT	Coefficient of Variation Inter-Departure Time
CVPT	Coefficient of Variation Processing Time

DBR	Drump-Buffer Rope
DT	Departure Time
EDD	Earliest Due Date
EKCS	Extended Kanban Control System
F ( )	Function of tool and operator characteristics
Fi	The amount of time job i spends in the system
FIFO	First-In-First-Out
GKCS	Generalized Kanban Control System
i	Item number
I-AT	Inter-Arrival Time
I-DT	Inter-Departure Time
j	Operation number
JIT	Just In Time
KCS	Kanban Control System
Li	The amount of time by which the completion time of job i exceeds its due date
m	Number of machines
MBS	Matched Batch Size
Mean I-AT	Average Inter-Arrival Time
Mean I-DT	Average Inter-Departure Time
MIVP	Minimum Inventory Variability Policy
MRP	Material Requirement Planning
MTBF	Mean Time between Failure
MTO	Make-to-Order
MTS	Make-to-Stock
MTTR	Mean Time to Repair
n	Total number of items
NMBS	Non Matched Batch Size
NOP	The number of operations remaining
Op1~6	Operation 1 to Operation 6
PFB	Pull From Bottleneck
PM	Preventive Maintenance
PMS	Preventive Maintenance Scheduling
PT	Process Time
QT	Queue Time
R	Lot Release
RCT	Raw Cycle Time
RPS	Recovery Performance Simulation
RTI	Relative Tardiness Importance
SA	Starvation Avoidance
SD	Standard Deviation
SIPM	Single Item Processing Model
SM1~SM5	Simulation model 1 to Simulation Model 5
SPT	Shortest Processing Time
Stdev I-AT	Standard Deviation Inter-Arrival Time
Stdev I-DT	Standard Deviation Inter-Departure Time

STU	Standard Time Unit
TBF	Time between Failure
TH	Throughput
TOC	Theory of Constraint
TT	Transport Time
TTR	Time to Repair
U	Utilization
UTIL	Utilization of available capacity
WIP	Work-In-Process

CHAPTER - 1 INTRODUCTION AND RATIONALE

1.1 Background

Production lines are designed to execute a series of operations to complete the transformation of raw material into valuable products. In the front end semiconductor industry, the lifetime of products is relatively short, with a steep decline in selling price over time. Therefore, in such volatile environment, front end semiconductor factories must achieve high productivity to increase market share and profit margin [1]. Maximizing productivity and minimizing costs depend on high utilization (U), high throughput, short Cycle Time (CT), and minimizing stock and Work-In-Process (WIP) [2]. In particular, front end semiconductor factories are frequently concerned with reducing cycle time. It improves cash flow and minimizes order lead time. The entire manufacturing process, from start to shipment, generally takes six to eight weeks. Reducing the variance of cycle time also improves the ability to meet due dates announced to customers, and consequently raises their satisfaction [3].

Three possibilities exist to reduce cycle time:

(1) The addition of capacity (e.g. adding new operators, upgrading machinery, and purchasing additional machines). Expanding the number of machines reduces cycle time regionally; nevertheless the overall cycle time of the product might not necessarily diminish. It just moves the bottleneck to the next station [4]

(2) Reducing the loading of the production line to avoid congestion and queues. Nevertheless, a low utilization of the line is contrary to the objectives of productivity and profitability.

(3) Variability reduction. Li [2] has illustrated the corrupting effect of operational time variability on cycle time. In mass production manufacturing, cycle times called out in the design phase are never achieved. Unpredictable factors delay the flow of material, information and resources. Therefore a better control of variability sources limits the delays and improves the cycle time.

Overall, reducing the effects of variability appears more efficient and relatively cheaper than purchasing additional equipment [5]. Different types of control policies (Section 2.6) can be implemented to ensure that the factory's cycle time is minimized [6].

1.2 Research Problem

Variability in a manufacturing system results in products taking more time to complete than originally planned. With increasing variability in the fab, not only the cycle time increases but also the distribution of cycle time spreads out. So the scheduling of items completion becomes problematic. This is of course not desirable from an operations manager's point of view [7] as the dates of delivery to customer must be respected.

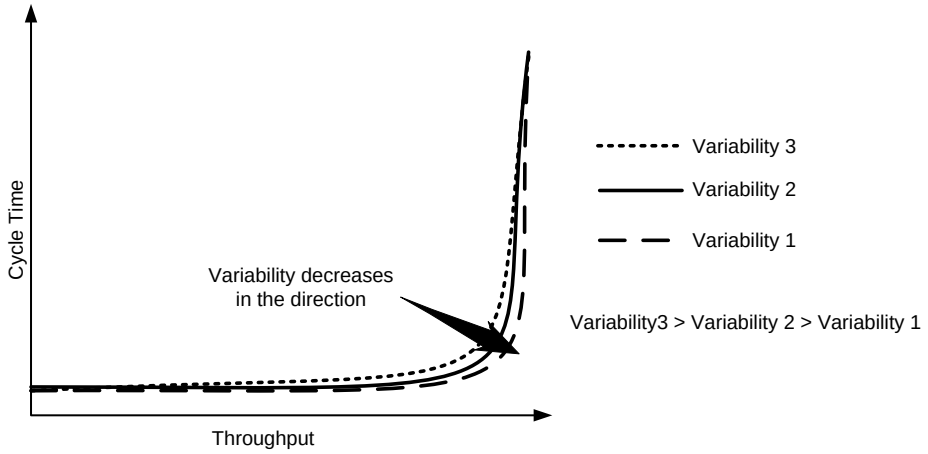


Figure 1.1: Cycle time performance curves showing the relationship between cycle time, throughput and variability [5, 8]

Figure 1.1 represents three typical performance curves obtained for different levels of variability in a front end semiconductor production line. These cycle time performance curves show the typical relationship between cycle time, throughput and variability.

A decrease in variability results in a high throughput at a low cycle time. Therefore a reduction in variability brings significant improvement in productivity.

1.2.1 What is the Problem?

Variability exists in all production systems and can have a large impact on performance. Therefore, variability measurement is critical to effective manufacturing management. A variety of factors contribute to cycle time variability in a semiconductor factory. Li [2] assembled the variability factors into two parts ─ (1) process factors and (2) flow factors:

(1) Process factors regroup the influence of all the physical assets ─ human and machine ─ on the variability. Unexpected breakdowns stop the production. Mixing products/processes introduces delays to setup machines. Operators' qualification influences their response time. Re-work lots take priority over

normal lots process schedule. All these unnecessary time-wastes cause productivity loss.

(2) Flow factors concern the organization of the production and the rules driving the movement of the WIP between the production units. Batch machines stop the flow and then suddenly release a high number of items simultaneously. A complicated process makes products re-visit the same machine several times and results in a non-linear flow. Hot lots, take priority over any attempt of regulating the flow.

Variability is clearly a complex issue and further study is necessary to understand its influence mechanisms on a semiconductor production line. Section 2.3 further describes the characteristics of the sources of variability.

Certain rules are defined in order to address such variability factors [6] and attempt to control cycle time and cycle time variability. For example, rules to determine the timing of items release into the production line, rules to prioritize items in process queues or rules to schedule preventive maintenances. These rules are referred to as scheduling policies. Nevertheless an inadequate scheduling policy causes even more disturbance on the production flow. Therefore the scheduling policy should be devised with caution and discernment. A clear understanding is necessary.

1.2.2 For whom is it a Problem?

In the front end semiconductor industry, the problem of not controlling variability is a problem first and foremost for operation managers involved in the planning and scheduling of the production. It is crucial to improve cycle time, output capacity and the

overall performance of a manufacturing facility, that they take informed decisions in applying resources. Strategic choices need to be made in term of release timing, items prioritization, capacity allocation, etc. Without understanding, an accurate fit of the scheduling policy is not possible, leaving gaps and approximations detrimental to the productivity of the line.

The research problem is also relevant for the academic community, especially for those who are interested in optimizing production line productivity through innovative scheduling control.

1.3 Purpose of the Study

The specific purpose of this dissertation is to contribute understanding on the control of cycle time and cycle time variability in a front end semiconductor production line. Many different sources of variability can be identified in a production line. Therefore, a conversation was engaged with semiconductor factory managers to identify the dominant sources. Batching and tool availability have emerged as the main disturbances in a modern front-end semiconductor line. This study will focus on those two factors and the control of their negative impact on the line's productivity. In collaboration with these factory managers, three objectives (1 to 3) were identified to fulfill this purpose. Finally, the lessons learnt from previous simulation and modeling process should be applied to produce and develop a new release policy.

1.3.1 Objectives

Factory managers pointed out that the variability created by process factors has for origin specific toolsets or operations. A metric is essential to investigate operation individually and identify sources of process variability in the production line.

Objective 1: Develop a metric to measure the amount of variability created by an operation.

Many different factors of variability can be identified in a production line (Section 2.3). Nevertheless, factory managers again pointed out that for a modern front-end semiconductor factory three factors (as justified in section2.13) are preponderant: tool availability, batching and re-entrant lines. These three factors cannot be studied simultaneously. Too many variables would considerably increase the difficulty. The scope of this study will be limited to the detailed analyses of tool availability and batching only. Then the results obtained will be tested in a re-entrant environment. Interesting areas for policy, practice and future research for the academic community, will be highlighted.

Objective 2: Understand and explain the impact of tool availability and batching on cycle time and cycle time variability.

Managers also pointed out that some operations have high output variability, but they do not have the highest impact on the overall production line. Thus, the interactions between operations are keys to the reduction of process flow variability. The circumstances easing or emphasizing variability in the flow have to be exposed.

Objective 3: Determine how the variability in the flow of items is affected by the interaction between operations.

The utilization of an appropriate release strategy appears as a promising solution to control cycle time and cycle time variability (Section 2.6). Therefore, the results obtained from the three previous objectives will be used to develop a release strategy adapted to front end semiconductor production lines.

Objective 4: Develop a new release strategy to take advantage of the interaction between operation and control batching and tool availability disturbances.

1.3.2 Significance of the Study

This study supplements existing literature on operation management by shedding new light on process control. It summarizes the knowledge available on process variability and scheduling policies. It identifies a range of mechanisms affecting cycle time and cycle time variability. In particular, it develops the concept of critical availability and provides a novel release strategy enhancing the performances of the production line.

Thus, the study brings an innovative reference to managers of front end semiconductor factories and helps the development of their scheduling policy. For the academic community, it highlights useful areas for policy, practice and future research.

1.4 Approach

A well developed methodology is useful in demonstrating what was done in the research process, and to articulate how research practices transformed observations into data, results, findings, and insights

1.4.1 Overview of Methodology

Two different approaches were adopted in this study.

First, a descriptive study was undertaken to better comprehend the nature of the problem. A front end semiconductor devices manufacturer was contacted to get access to the production data. The analysis of the data was performed using several statistical calculations. This preliminary work was needed to gain familiarity with the phenomenon in the situation and understand what is occurring. Then a model could be developed and a rigorous design was setup for comprehensive investigation.

Second, an explanatory study was undertaken to clarify the causes, the context, and consequences of the observed phenomenon. A simulation model was developed using Extend Simulation software (www.extendsim.com). The advantage of simulation is the facility to monitor and control the entire production system without doing any changes in the real environment, saving time, efforts and money. Parameters can be modified many times to determine the optimum configuration.

1.4.2 Limitations

In the exploratory study, the absence of control on the content of the data was a seriously limiting factor. The data were giving indication as to the “when”, “how many” and “how often” something occurs, but they couldn’t tell us “why” or “how”. The research could not describe what caused a situation. Moreover hypothesis and theory couldn’t be put under the test as no experiment could be run.

With simulations, the fidelity and validity of the simulations outcomes is dependent on the acquisition of valid source of information, the relevant selection of key

characteristics and behaviors, and the appropriate use of simplifying approximations and assumptions within the simulation. Moreover simulation results are hypothetical. Ideas and theories may be tested with simulations, but the transfer to real environment is not 100% sure and outcomes may differ slightly from predictions, particularly for complex systems as semiconductor processing lines.

1.5 Definition of Key Terms

The level of performance of a production line has to be measured in order to evaluate the quality of the process flow. That is how manufacturers judge the factory performance. If the production is good, then it is possible to satisfy all customers' needs. The precise sampling of shop-floor data, such as machine down times is a must.

Variability is one of the production line's characteristic that needs to be monitored because reducing the variability in the manufacturing system enables the measurement of low and predictable cycle times [7]. Variability is closely associated with randomness. Consequently, to understand the causes and effects of variability, one must understand the concept of randomness and the related subjects of probability and statistics. This study introduces the necessary ideas in as loose and intuitive a manner as possible. The necessary information, examples and data are extracted from the semiconductor industry.

These statistics should not only include first order measures like means, but also statistics that allow monitoring the variability of the manufacturing system. They can be classified in three categories: those relating to the whole production line, those related to a specific machine or operation step and finally those specific to each product or lot. In

the following sections, several experiments will use these statistics to gauge the experimental results.

1.5.1 Definition of Production Line Statistics

The line performance over any time period is measured using three parameters: the full line's cycle time, loading, throughput, and the amount of Work-In-Process (WIP) [7, 9].

Cycle Time (CT)

The cycle time is the total time required to produce a product, from entering the factory to leaving the factory. Cycle time includes time actually spent processing, as well as transport time and time spent waiting in queue. The following four key statistics, determining the cycle time of a process line, can be measured: mean effective cycle time, standard deviation, coefficient of variation, and distribution.

Loading

The production line loading is the number of items started per week of multiple product types. As cycle time increases non-linearly in utilization, the production line performance is very sensitive to the factory loading at high utilization levels. Therefore, it is not possible to run a process at 100 percent of its theoretical capacity. Factory loading should be limited to slightly less than the factory (multiple) constraint's capacity [10] to avoid 'blowing up' the system and being obliged to reduce the production rate anyway [11].

Throughput Rate (TH)

Throughput rate is average output rate of a factory or workstation. The throughput of a factory is equal to the factory loading multiplied by the average line yield.

Work-In-Process (WIP)

The WIP is the average number of units of product in the factory (or at a workstation). WIP includes units being processed on equipment, as well as units in transit, or awaiting processing at an equipment group. In other words, all the unfinished items located somewhere along the production line. If given a fixed input and output schedule, a balanced production line is one whose mean WIP does not increase over time due to randomness of machine failures and repairs.

1.5.2 Definition of Machine/Operation Statistics

A characteristic of a queuing system is that the queue will grow to infinity when the arrival rate is greater than the service rate. Thus, to effectively monitor a machine or an operation (group of identical machines working together as one), the efforts should be focused on items arrival/departure, process time and machine/operation's utilisation, capacity and downtime [1].

Lot Arrival and Lot Departure

Lot arrival and lot departure are linked together as the output of one subsystem is usually the input to one or more others. Therefore, to understand the behavior of a specific machine, the output process of the upstream subsystem must be examined [12]. It means also that many publications, studying a serial production line without connection points, only consider the lot departure of the stations and not the lot arrival.

Lot departure (or lot arrival) is either characterized by the inter-departure time (inter-arrival time) or departure rate (arrival rate) and all the statistics associated – mean, standard deviation, coefficient of variation and distribution.

Coefficient of Departure (/Arrival) Rate Variability (C_{DR}/C_{AR})

The coefficient of departure rate variability has been used as a possible measurement of the variability of departures from a station [10]. One of the issues to resolve concerning flow variability is how to characterize the variability of departures from a station in terms of information about the variability of arrivals and process time. Variability in departures from a station is the result of both variability in arrivals to the station and variability in the process times [11].

Inter-Departure Time (I-DT) / Inter-Arrival Time (I-AT)

Another way to characterize the output (/input) process is to examine the time between units leaving (/arriving) the subsystem, called the inter-departure (/inter-arrival) time. The inter-departure (/inter-arrival) time is a random variable because of processing variability. The moments of the inter-departure (/inter-arrival) distribution are important descriptors of the output (/input) process. The mean determines the throughput of the system (average throughput rate = inverse of mean inter-departure time). The variance provides a measure of the variability of the output (/input) process. The variability of the inter-departure (/inter-arrival) distribution is strongly affected by three factors: variance of processing time, line length, and skew of processing time [12].

Correlation Coefficient (C_C)

Because of the link between lot departure from one station and lot arrival to the next station, some correlation exists between the departures of consecutive stations. This correlation can be measured by calculating a correlation coefficient. This coefficient can be used to examine the effects of line length, buffer capacity, and buffer placement on the inter-departure distribution and correlation structure (autocorrelation function) of the output process of the production line. The signs and magnitudes of the correlation structure affect how the manufacturing subsystems interact. Negative correlation indicates that less storage is required to buffer the output of one subsystem from the next manufacturing subsystem than if the correlation structure were positive or zero. Additionally, information from the correlation structure can be used to generate predictors for inter-departure times [12].

Process Time (PT)

Process time is measured as the time from when a job is released into a machine or station to when it exits. Here again all the statistics associated ─ mean, standard deviation, coefficient of variation and distribution ─ can be used. The skew of the processing time distribution was found to be an important predictor for the variance of the inter-departure distribution and the correlation structure. Higher skewed processing time distribution causes greater variance of inter-departure intervals. Hendricks [12] shows that the variance of the inter-departure distribution is directly proportional to the variance of the processing time distribution. For the variance of the inter-departure distribution, the CV (Coefficient of Variation) of the processing time distribution completely explains the inter-departure variance for larger buffer capacities. However,

skew of the processing time distribution increases inter-departure variance, if buffer capacities are small [12].

Utilization (U)

The utilization of a machine is defined as the fraction of time the machine is not idle for lack of items to process. This includes the fraction of the time the machine is working on items or has items waiting and is unable to work on them because of a machine failure, setup or other detractor [11]. In other words, it is the ratio of productive time to total manufacturing time [13]. In industry, cost accounting encourages high machine utilization. Higher utilization of capital equipment means higher return on investment. However, due to the variability in the line, WIP levels and cycle time grow continuously with increasing utilization. The more variability a line has, the lower utilization must be to compensate and this is magnified on constraint and/or near-constraint tools [10].

Bottleneck

The bottleneck in a factory is defined as the machine group that has the highest long term utilization for a given product mix [11]. Some authors define bottleneck as having a utilization of 100%. However, in common use, bottleneck usually refers to the most highly used machine group. When a bottleneck occurs, it causes products to wait for processing, thereby increasing their cycle time. Nevertheless, Woolverton [1] notes that identifying the system with the highest utilization is reactive and not always indicative of problem areas. Her focus, therefore, shifted to using cycle time as an indicator of the factory's constraints; each individual operation is attributed a cycle time goal. Tool sets that continuously miss their cycle time goals are re-defined as constraint operations (bottleneck) and then managed according to their new status.

Machine Downtime

Downtime is a period of time during which the machine is not in a condition to perform. Usually, it is discriminated between scheduled and unscheduled downtime. Scheduled downtime occurs when machines are not available to perform due to planned events, such as Preventive Maintenance (PM), set-up, system testing and so on. In contrast, unscheduled downtime occurs when machines are not in a condition to perform due to unplanned events, such as random failures, technical failures and other unpredictable factors.

The machine downtime is measured using three different parameters (Figure 1.2): Mean Time Between Failure (MTBF); Mean Time To Repair (MTTR); and Availability (A), which is the fraction of time, the machine is available to process WIP.

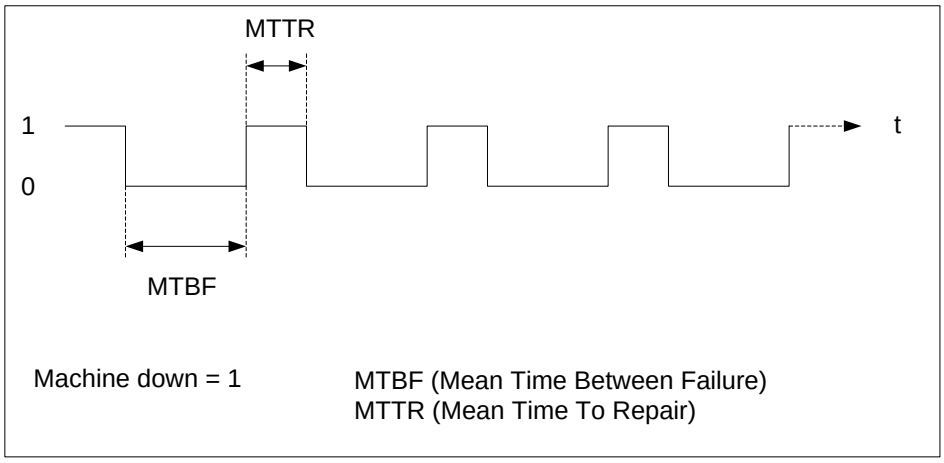


Figure 1.2: Downtime set up

$$A = \frac{MTBF}{MTBF + MTTR}$$

Equation 1.1 [11]

Machine Capacity

The capacity is the maximum throughput of a machine. In other words, the number of items a machine is able to process in an interval time, for example 420 items per week. For an operation, the capacity is the sum of the capacities of its constituent machines.

"Current capacity" can be defined as the capacity modulated by the current availability of the machine (or constituent machines). If the machine in the previous example has this week an availability of 50% then its current capacity is only 210 items per week (420 items/week x 50%).

For a factory, the capacity is the throughput rate that drives the idle time on the bottleneck to zero. Releasing work into the system at or above the capacity causes the system to become unstable (i.e. build up WIP without bound) [11]. Having a measure that easily identifies capacity-constraining machines helps managers to allocate resources or schedule preventative maintenance at these machines to reduce variability, thereby improving the cycle time [5, 8, 14].

1.5.3 Definition of Item Statistics

This part considered two item statistics, respectively, average item lateness and queue time. It is a basic approach to evaluate items delay time. The more accurate the forecast of item delivery time, the more satisfied are the customers.

Average Item Lateness

The due date of an individual item is set to the release time plus the average 'planned' cycle time of the corresponding product [15].

Queue Time (QT)

Items queue when they are waiting for a resource, e.g. workstations to be processed, transport devices to be moved, etc. Queue time represents a large fraction of the total cycle time. There again, all the statistics associated – mean, standard deviation, coefficient of variation and distribution can be used to characterize performance.

1.5.4 Summary

All these key terms refer to important measurable characteristics of any production line. The statistical analysis of the numerical data collected provides an effective monitoring of a factory performance. It allows the interpretation of the data and results in the understanding of the phenomenon observed. Therefore, all the following experiments will employ those statistics to compare and contrast the results obtained.

1.6 Thesis Structure

Given the purpose of the study, the initial objective was to research relevant past work and to contribute understanding on the control of cycle time and cycle time variability in a production line. However, as the research progressed the objectives and focus changed towards developing a release strategy and optimising cycle time under random equipment failure. The remainder of this chapter outlines the structure of the thesis by summarising the main topics discussed and developed in the succeeding chapters.

1.6.1 Chapter 2 – Literature Review

Chapter two begins with a review of the literature, which addresses the various sources of variability. Next, the studies on scheduling policy are reviewed. The purpose of this

review is to provide an understanding of the previous research in this area, as well as providing a rationale for the choice of objectives selected in the present study.

1.6.2 Chapter 3 – Pre-Study: Real Production Line Data Sample

From the literature review, a basic grasp of the possible sources of variability has been developed. In order to fulfill the goals and objectives, developing a deeper understanding of the phenomena occurring in a real production line is needed. Information about the flow of items through a process line was required to answer basic questions such as: How are the items moving from one operation to the next? How long are they staying at each operation? When and where are queues occurring? A manufacturer was contacted to get access to their production data and an exploratory study was initiated. First, work concentrated on the development of a metric for variability, then on the characterization of lot arrival and lot departure and finally on queue time analysis.

1.6.3 Chapter 4 - Methodology

The study was undertaken using simulation modeling. Chapter 4 presents the models developed to simulate various key factors, bottleneck, tool availability, and batching in a production line. Chapter 4 also exposes the data collected and the statistics used for their analysis.

1.6.4 Chapter 5 – Simulation Results

This chapter is the main experimental chapter in the thesis.

First, the basic relationship between queue time, utilization and inter-departure time is studied.

Second, four experiments are conducted to investigate the performance of the modeled batch processing line under various product loads and item release profiles in comparison to a single item processing line (without batch process).

Third, the model is extended by introducing downtime in one of the operations. The simulations analyze the interactions of batch process, tool availability and constraint operations, and highlight the issues affecting the entire line.

1.6.5 Chapter 6 – Development of a Release Strategy – Results and Discussion

Finally, a new release strategy is devised to minimize cycle time and WIP level. It avoids any variability in the flow of items in the operations preceding the bottleneck. In other words, it maintains constant the flow (CONFLOW) of items. CONFLOW's performance is compared to a standard push strategy for lines under random equipment failure. Various scenarios were considered including single item processing, batch processing and re-entrant lines.

1.6.6 Chapter 7 – Conclusion and Recommendations

Chapter 7 outlines the conclusions and implications of the study. It discusses how the aim was fulfilled and the research objectives fulfilled. It also discusses the contributions of the study for both theory and practice and proposes avenues for future research. Finally, some criticisms are directed towards the study.

CHAPTER - 2 LITERATURE REVIEW

2.1 Introduction

Virtually all manufacturing managers want on-time delivery, minimal WIP, short customer lead times, and maximum utilization of resources. Unfortunately these goals conflict [11] and compromises need to be found. Fortunately, reducing the effects of variability through the utilization of an appropriate scheduling policy appears as a promising solution for the improvement of front end semiconductor production line performances.

The research begins with a review of the literature. First, the fundamentals of queuing theory are summarized to highlight the corruptive influence of variability on a production line performance. Then, the various sources of variability in a semiconductor factory are addressed. These sources assessed with the inputs provided by operation manager of a semiconductor company. Next, the studies on scheduling policy are reviewed. The purpose of this review is to provide an understanding of the previous research in this area, as well as providing a rationale for the choice of objectives selected in collaboration with the semiconductor company.

2.2 Fundamentals of Queuing Theory

Queuing theory studies the influence of process time variability and flow variability on the overall production line. It evaluates the impact of these types of variability on the key performance measures for a line, namely, WIP, cycle time and throughput [11].

Even if the process rate is high, it is very likely that some items will be delayed by waiting in the line [16]. Actual process time (t_e) typically represents only a small fraction of the total cycle time (CT) in a plant. The majority of extra time is spent queuing (CT_q) for various resources (e.g. workstations, transport devices, or machine operators) [11].

$$CT = CT_q + t_e$$ Equation 2.1 [11]

In general, items arrive and depart at irregular intervals; hence the queue length will assume no definitive pattern unless arrivals and service are deterministic. Thus it follows that a probability distribution for queue lengths would be the result of two separate factors – arrivals and processes – which are generally assumed mutually independent [16].

2.2.1 Characteristics of Queuing Processes

A queuing system can be described as items arriving for process, waiting for process if it is not immediate, being processed, and then leaving the system. In most cases, six basic characteristics of queuing processes provide an adequate description of a queuing system [16]:

- Arrival pattern of items: either deterministic (characterised by the mean inter-arrival time) or stochastic (characterised by the inter-arrival probability distribution).
- Process pattern of operations: single item or batch processing, characterised by the process rate.

- Dispatching policies: refers to the manner by which items are selected for process when a queue has formed.
- System capacity: limit to the maximum queue size: infinite or finite (no further parts are allowed to enter the queue until space becomes available by an item processing).
- Number of process channels: number of parallel process stations which can process items simultaneously.
- Number of process stages

All these characteristics make the variety of queuing systems almost endless. Discrete part models fall into two main categories, those which model the unreliability of machines and those that do not. Most models assume exponentially distributed service, repair and breakdown times [17]. Nevertheless, regardless of the queuing system under consideration, the job of queuing theory is to characterise performance measures in terms of descriptive parameters [11].

2.2.2 Queuing Notation

A standard notation is used in the queuing literature to describe queuing processes. A queuing process is described by a series of symbols and slashes such as A/B/X/Y/Z, where A indicates the inter-arrival time distribution, B the service pattern as described by the probability distribution for process time, X the number of parallel process channels, Y the restriction on system capacity, and Z the dispatching policy. Typical values for A and B, along with their interpretations, are:

D: constant (deterministic) distribution

M: exponential (Markovian) distribution

G: completely general distribution

A common example of queuing process is M/M/1: arrival and process are exponentially distributed; there is only 1 station and an infinite queue capacity.

2.2.3 Performance Measure

Queuing theory was developed to provide models to predict behaviour of systems that provide service for randomly arising demands [16]. The queuing theory methods used to analyse production lines models give either exact or approximate results. Exact results are only available for short production lines [17]. For the solution of longer lines approximate methods are required. For example, the blocking caused by finite buffers greatly increases the complexity of the analysis of a queuing network model. Consequently, exact results for queuing network models of production lines exist for only a limited number of cases and are from a practical point of view of little use for manufacturing system design purposes [17]. Even so, exact result models are important as they provide useful qualitative insight into the behaviour of these systems. They also provide results for comparison purposes against approximate results. Also, some of these models form the basis of approximate algorithms.

A well known formula for the determination of the waiting time (CT_q) of items is the Kingman's (or VUT) equation:

$$CT_q = \left(\frac{c_a^2 + c_e^2}{2} \right) . \left(\frac{u}{1-u} \right) . t_e = V . U . T \qquad \text{Equation 2.2 [11]}$$

This formula doesn't apply to every process. It is valid for G/G/1 queues. However, it offers valuable insight into more complex and real systems [11]. This formula suggests that there are two factors causing queuing time: variability V and utilization U, t_e is the effective process time. Variability involves the so-called coefficient of variation (CV), which consists of c_a^2 and c_e^2. c_a^2 represents the variability of the arrival process. c_e^2 is the variability of the effective process time. So, the variability of arrival and/or process must be reduced to decrease the waiting times.

The second factor that could cause queuing time is utilization. Utilization is the fraction of time a workstation is busy over the long run. Higher utilization leads to longer waiting times. Utilization has the most dramatic effect on waiting times. The reason is that the VUT equation has a 1-u term in the denominator. As utilization approaches one, cycle time approaches infinity. Cycle time is very sensitive to utilization. If variability is higher and utilization approaches one, cycle time will sooner blow up.

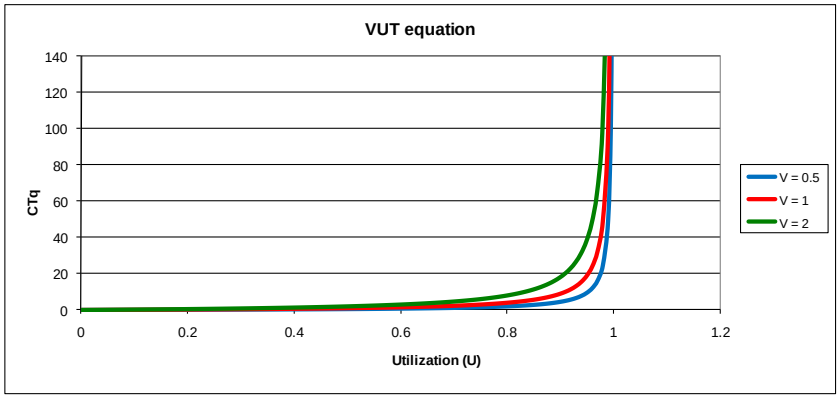


Figure 2.1: VUT equation

In summary, two factors contribute to long waiting times: high utilisation levels and high levels of variability. The G/G/1 model illustrate that both increasing effective

capacity (i.e. to bring down utilisation levels) and decreasing variability (i.e. to decrease congestion) are useful for reducing cycle time [11].

These results are also valid for queuing systems more complex than G/G/1 queues. Exact or approximate solutions can also be determined (see [16] and [17]) and their behaviour is similar to the VUT equation (APPENDIX - B).

Queuing network models are important manufacturing system design tools. However, because they are mathematically based models, their application is somewhat limited [16]. They are usually structurally inflexible, in that a particular formulation of the model is only valid for a narrow range of problems [17]. In contrast, simulation modelling is a very flexible modelling tool, and as a result is probably the most important and the most popular modelling tool available to manufacturing system designers [17].

2.3 Variability

Variability exists in all production systems and can have a large impact on performance (as mentioned earlier in Section 1.2). All sorts of factors contribute to cycle time variability in a semiconductor factory. Li [2] assembled the variability factors into two categories ─ (1) process factors and (2) flow factors, which are explored in detail in Sections 2.4 and 2.5, respectively:

2.4 Variability ─ Process Factors

Process factors regroup the influence of all the physical assets ─ human and machine ─ on the variability. These factors include random equipment failure, reworked lots, variation in operators, product/process mix, machines setup and tool dedication. Their

occurrence cannot be anticipated. All these unnecessary time-wastes cause productivity loss.

The goal is to find potential areas for productivity improvement that will yield a quantifiable increase in fab 'cycle-time-constrained capacity' [15]. Process-related and equipment-related improvements are studied to enable lower CT at individual toolsets, increase capacity and/or shortening process times [10].

2.4.1 Equipment Downtime

Machine downtimes increase the waiting times in the process. Two types of downtime exist. Scheduled Preventive Maintenance (PM) when the machine is stopped by the floor technicians to perform some maintenance and unscheduled random equipment failures.

Scheduled maintenance typically occurs between jobs, rather than during them. It is predictable and can be taken into account for controlling the production line. On the contrary, equipment failures can occur right in the middle of a job and their unpredictability renders production line control difficult [18].

Unscheduled downtime occurs from highly complex and technologically advanced processes [19], from equipment in insufficient condition to perform the intended function, from the variation of operators, technicians, and engineers, and from the unavailability of spare parts. For example, a wafer broken inside a machine could stop the production on this machine, and if it happens often, the capability to confront production requirements will also be limited [20].

Schömig's work [7] shows that reducing the downtime frequency on machine groups in the production line reduces the processing time and processing time variability. Indeed, in semiconductor manufacturing, the variability of service time and repair time are the primary causes for the variability in a machine's performance [7]. These two sources of variability are caused by service delays, lack of spare parts, long repair periods, and lack of technical experience.

- Lack of spare parts for repair is a significant problem as it results in excessively long machine downtimes. On the other hand, spare parts inventory is expensive; the total investment in spare parts can amount to a significant fraction of the value of the machine to be maintained [21].
- The high complexity of semiconductor equipments makes those equipments difficult to repair in case of breakdown. Therefore repairs are slow and require highly skilled technicians.
- Furthermore, once a machine is repaired, it still needs to be fine tuned through calibration processes. These calibrations can sometimes be even more time consuming than the repair itself.

Preventive maintenance should be properly scheduled to space them evenly and avoid that several machines stop for maintenance simultaneously [22]. A good PM schedule increases tool availability by trade-off between the planned downtime versus the risk of unscheduled downtime due to tool failure.

Moreover variability may be significantly reduced by intelligent preventive maintenance scheduling [23]. The ability to provide failure-free production for a predictable time,

results in lower variability than providing a system with a high mean time between failures. The main task in improving variability is to provide uninterrupted operation for a given period of time. After the given period of time, Preventive maintenance is performed and the process is repeated [24]. Tag [22] reported a reduction of ~20% in the availability variability by using an optimized scheduling system and ~30% reduction in CT.

Variability is generally measured using the coefficient of variation and simulated in experiments using a distribution (see APPENDIX - A). The type of distribution is subject to debate in the literature. Some argue that the actual downtime distribution type plays only a minor ─ if not negligible ─ role in the performance of the fab [7], others argue that both the variability and the shape of the distribution used for modeling Time To Repair (TTR) and Time Between Failure (TBF) have a considerable effect on the factory performance estimates, e.g. average cycle time [25]. Regardless, exponential random variable seems to be the most general distribution reported in the fab [26].

2.4.2 Rework Lots

Rework lots are made from defective wafers that fail to pass inspection and need to be corrected. They are sent back in the production line to repeat earlier processing. Rework lots typically have fewer wafers than regular lots. The processing of these lots generally has a high priority in order to re-integrate the wafers to their parent lot (lot of origin). As rework lots originate from a processing problem, they are not predictable and perturb fab organization. The entire process needs to be accurately controlled and monitored; otherwise lots will fail frequently, and then require rework. This will at least double the cycle time of impacted lots [27].

In semiconductor manufacturing, the process is very well monitored. Any machine deviating from its baseline is immediately detected. So, few lots are affected before the machine is stopped for repair. In those conditions, the main impact on the production line is the time to repair the machine and not the few lots that need rework.

2.4.3 Variation in Operators

The complexity of equipment in semiconductor fabrication requires a highly skilled operator who has the ability to monitor multiple machines at one time. The operators' response to the various situations (such as machine issues, or process issues) will depend on this ability. Significant variability can occur when a production operator is not able or not available to attend a tool in a timely fashion. One to three months is necessary to train an operator. It depends on the process and the machines he/she will have to monitor. Temporary workers have a lower cost, but lack technical skills. Permanent workers have a higher cost, but have a higher skill level. Thus, determining the required number of trained operators in a specific skill, is one of the key factors to meeting production requirements [20]. Stratman [28] assessed the production cost and quality cost impacts of various staffing mixes of temporary and permanent operators.

Here again, modern semiconductor production lines are fully automated. The role and influence of operators have been greatly reduced and can be neglected compare to other issues.

2.4.4 Product/Process Mix and Setup Variability

Under the requirement of high throughput and profitability, machines are shared by a number of different products or identical products at different steps in the production

chain. This can result in heavy loads for those machines shared by different products or used regularly at various steps in the process [29].

Moreover, the process route of two different products can be almost identical or extremely distinctive, depending on the type of products. The greater the differences, the greater the difficulties to process various products simultaneously on the same production line. It complicates the scheduling and attribution of one machine to one process or the other. Accordingly complicated queuing may occur [29].

Switching from one product or step to the next may also introduce a delay, while the machine is setup with new parameters. For example, an oven will bake products at various steps, each step requiring a different baking temperature. The temperature will have to reach and then stabilize at the value required before processing. It increases the delays to swap a machine from the production of one product to the next.

A setup is required for changeover of a machine from making one product to making another. Two possibilities exist to minimize the impact of setup time on the production output, reducing the frequency of setups and reducing the setup time. Large lot sizes can be used to keep the number of changeovers to a manageable level [11]. For example, Fowler [15] applies a setup avoidance policy to minimize the amount of setup and to improve productivity. This policy overrides the default dispatching rule (lots due date) in order to avoid performing an extra setup on the machine. Another approach consists of optimizing changeovers to reduce the setup time to the point where the time lost is negligible [11].

Each product mix has a different impact on production performance due to the equipment set availability, frequency of setups for product type conversions, control rules and loading condition [30]. Significantly, product mix introduces disparities and variability in the product cycle time. It impacts throughput and the capacity to respect due dates.

This can indeed be an issue in back-end semiconductor factories. They receive various products from various front-end factories and have to process them together on the same line. On the other hand, front-end lines often process only one or two products. And in this case, each product will run on a different line.

2.4.5 Tool Dedication

Tool dedication occurs when machines are split into groups and each group dedicated to one specific product or process step. In other words, products cannot be processed by any available machine. They have to wait until one of the dedicated machines is available. Three reasons to have dedicated tools in a semiconductor factory are: equipment location, contamination, and equipment capability.

In dedication for equipment location, a piece of equipment may be selected to run specific process recipes because of its location in the factory. Dedication for contamination occurs when there is concern that processing two different recipes or steps on the same machine will lead to contamination. A typical example in semiconductor production lines is contamination by copper particles. Such contamination may ruin a wafer. Therefore, machines processing wafers with copper, are not allowed to process wafers without copper to avoid their contamination.

Dedication for equipment capability is due to the variation in performance between the machines. Some machines may not be good enough to execute critical steps of the process. Another reason for tool dedication is to improve the matching between steps. For example, photolithography requires tremendous accuracy in overlay of the various steps. But each machine has slightly different characteristics, thus the process owner drives the products back to the same photolithographic tool in order to ensure a consistent image placement even if this tool is busy and others are available.

Thus dedication decreases the available machines in a production line from many to one, resulting in tools having many items queuing in front of them while others are idle. These items will not be moved to the available tools [27]. Thus tool dedication can reduce the capacity of operations, and therefore acts as a detractor to throughput [31]. It also impacts variability, as the availability of a single tool fluctuate much more than the average availability of a group of tool. If the dedicated tool is busy then products are queuing. They have to wait to be processed until the dedicated tool is available. Worse still, if this tool is down, the queue in its buffer will sharply increase. That explains why dedicated tool produce cycle time variability.

Fowler [15] demonstrated that change in the factory management that includes relaxation of the dedication policies, could bring a 25% reduction of cycle time (and correspondingly in inventory) without a reduction in throughput. The variability of cycle time is also decreased.

While tool dedication is an issue in a real production line and should be avoided, in terms of theoretical analysis, it is equivalent to any other constraint operation. It will therefore not be expressly studied.

2.5 Variability – Flow Factors

When variability occurs at one station, it can affect the behavior of other stations in a line by means of another type of variability, which is called flow variability. Flow refers to the transfer of jobs or parts from one station to another. Significantly, if one upstream station has highly variable process times, the flow it feeds to downstream stations will also be highly variable.

In a single input, single output system, item departures from a station will in turn be arrivals to the following station. So once the variability of arrivals to one station is described, the effects on the variability of departure from that station (and hence arrivals to other stations) can be determined. Thus the flow variability for the entire line can be characterized [11].

2.5.1 Batch Processing

For machines that can process multiple lots simultaneously, batching policy refers to the number of lots that must be present to allow processing. For cycle time analysis of batching workstations, the batching policy, in addition to the workstation utilization and variability, has to be taken into account [32]. When processing time and/or setup time are long, batch processing can be useful to increase the throughput of a machine, but batching is a particularly dramatic cause of variability [11]. Indeed batch machines act

as dams, interrupting the flow by stocking products until full and then suddenly releasing all of them simultaneously.

For instance, suppose a conveyer brings 16 jobs once per 8-hour shift to a workstation. Since arrivals always occur in this way without any randomness whatsoever, one might wrongly interpret the variability and the CV (Coefficient of Variation) to be zero. But actually, batching mingles two different effects. The first effect is due to the batching itself, the following station does not receive any jobs for 8 hours and suddenly it receives 16 jobs simultaneously. Thus the inter-arrival time is large before the first lot of a batch and zero for all following lots of the batch. So the next operation, the one receiving the processed batch, 'sees' highly variable arrivals [33]. This is not a randomness issue, but rather one of release control. Nevertheless it still creates variability in the line as the interval time between job arrivals is not constant. The second effect is the variability, or randomness, in the batch arrivals themselves [11].

Batch machines usually have the longest process time in the manufacturing line; they also easily create long queues if the machine is not available [34]. To optimize machine utilization, full batch loadings are recommended. Although, if the batch machines are not highly utilized, significant wait-to-batch time is added to the cycle time as well as increased variability in the system. For underutilized batching machines, small batch sizes are recommended to minimize cycle time [5].

Therefore it is critical to determine when running less than a full load might be the right thing to do in order to minimize the waiting time in queue [11, 15, 35]. A load level too low decreases the throughput and increases the queue time dramatically, which

adversely affects the due date performance of the fab. On the other extreme, a load level that is too high increases the waiting time significantly increasing the cycle times for the lots and the variability for the following stations [36].

The determination of the optimum load/batch policy is an extremely complex problem as it is dependent, among others, on the variability of the lots arrival, WIP level, process and setup times, and mix of products. Several authors recommend considering the incoming work-in-process from upstream operation in order to predict arrival times of the next few lots [35-37]. Simulations show reduction in queue time and lot tardiness, particularly under moderate traffic conditions (30% to 70% utilization).

2.5.2 Re-entrant Lines

In multiple steps processing, an item re-visits the same machine several times along the production line. This type of manufacturing system is termed a re-entrant line; items may spend significant time waiting for an available machine, resulting in long cycle times and low production [38, 39].

In semiconductor manufacturing, most of the re-entrances are located in the photolithography area. Several layers have to be printed on the wafers, thus they have to re-visit the same tools several times along the process route [40].

Re-entrant lines are a significant difficulty for lot release policies as one work station has to process several stages. Hence, lots at different process levels compete for the same resources. That increases the problem of how to allocate the work station processing capacity to the job stages [41].

2.5.3 Hot Lots

Hot lots refer to the highest priority lots, often requested by the customer-pressing deadlines. They often get priority at all stages of the production line. It often results in irregular flow of items and can be extremely disturbing to the regular processing time and capacity jobs [42]. It results in significant impacts on cycle time. Hot lots are particularly harmful in batch processing machines, as they must be processed immediately without waiting the arrival of the items needed to complete the batch capacity [43].

Furthermore, the disturbance results in delays for normal lots. Delays that might make them miss their delivery schedule. To avoid this, managers and supervisors give these lots high priority, they become hot lots. It is a vicious circle.

Therefore, it appears that most efficient solution to minimize the impact of hot lots is to avoid their creation in the first place. Our primary objective of improvement in cycle time and cycle time variability should improve the respect of due dates and thus remove the necessity for hot lots.

2.6 Scheduling Policies

Managers and supervisors allocate products to operations and take significant decisions affecting items flow [44]. The purpose is to optimize this flow, taking into consideration the various products at different process steps and the random variability [15]. Thus, they try to minimize inventory costs and set-up costs, assure optimal WIP, maximize the capacity and utilization, and decrease the overall cycle time.

To perform these allocations, they typically follow a set of process rules. This set is generally called the item scheduling policy of the plant. It comprises two major aspects [6, 45]. The first aspect is referred to as the item release policy. Rules determine when new lots are to be released into the production line. The second aspect is referred to as dispatching policy. For items already in the line, queuing at an operation, rules decide which item is to be processed first when a machine becomes available. The prioritization is done according to certain attributes that differentiate the lot urgency for completion. It is important to control the correct mix of products to the stations in the proper proportion at the proper time [45].

The optimization of this set of rules is the main difficulty of operation management. Re-entrant lines and mix of products render difficult the tracking of lots and the establishment of priorities, especially taking into account that all lots have to meet their delivery date to customers [46]. Many parameters have to be considered: the way stations are interconnected; the number of workstations; the presence or absence of buffers and their capacities; machines breakdowns; and variable process times [47]. To obtain stable flow between batch-processing machines and single-unit processing machines is also a big challenge.

Untimely and uncontrolled scheduling policies create flow variability in the production line, bring about queues and a higher WIP, increase the throughput time, and result in an unstable production line [45]. While effective scheduling policies attain significant reductions of cycle time (13% for Kalir [10] and 50% for Shu [48]) and cycle time variability.

2.7 Dispatching Policies

For items already in the line, one has to decide which item is to be processed first when an operation becomes available. The basic idea of job dispatching is simple: develop a rule for arranging the queue in front of each workstation that will maintain due date integrity while keeping machine utilization high and manufacturing times low. This rule is referred to as dispatching rule. Many rules have been proposed for examples Shortest Processing Time (SPT), Earliest Due Date (EDD) and First In First Out (FIFO).

Blackstone [49] has made a good survey of various dispatching rules, and tested these rules by using a simulated factory under a broad range of conditions. The measurement criteria most often used for studying dispatching rules are Cycle time and Lateness. These criteria can be defined as follows:

Cycle time (CT_i): The amount of time job *i* spends in the system.

Lateness (L_i): The amount of time by which the completion time of job *i* exceeds its due date. Lateness may be negative, indicating an early completion.

In order to lend continuity to the discussion of research results, dispatching rules have been separated into three classes, each of which will be discussed in turn, and the more promising rules of each class will be noted.

2.7.1 Rules Involving Processing Time

Under Shortest Processing Time (SPT) [39], jobs at the operation queue are sorted with the shortest jobs first in line. Thus, the job in the queue having the shortest processing time will always be performed next. The effect is to clear out small jobs and get them

through the plant quickly. SPT typically decreases average manufacturing times and increases machine utilization [11]. Problems with SPT occur whenever there are particularly long job. In such case, jobs can sit for a long time without ever being started. Thus, while average due date performance of SPT is good, the variance of the lateness can be quite high. One way to avoid this is to use a rule known as SPT^x, where x is a parameter. By this rule, the next job to be worked will be the one with the shortest processing time unless a job has been waiting x time units or longer, in which case it becomes the next job. This rule seems to yield reasonably good performance in many situations [11]. For shops, particularly, concerned about having a few jobs very late, the SPT^x rule [50-52] seem worthy of consideration.

2.7.2 Rules Involving Due Dates

This approach considers that the function of shop floor scheduling is to ensure that the realized production adheres as closely as possible to the master production schedule. Each lot on the shop floor has been assigned a completion date by the planning system and the scheduling system tries to minimize deviation from these due dates, and minimize lateness. Since this avoids making some jobs early at the expense of others being extremely late, it tends to spread the deviations evenly across all jobs [53]. Therefore, the principle advantage of due-date-based rules over processing-time-based rules is a smaller variance of job lateness, and often a smaller number of tardy jobs. But they typically exhibit higher mean CT, and higher mean tardiness than SPT. Conway [54] studied three due date based rules which are earliest due date, slack and slack-per-operation. Under EDD, the job closet to its due date is processed on next. If jobs are all approximately the same size and routing are fairly consistent, EDD exhibits reasonably

good performance. The slack for a job is its due date minus the remaining process time (including setup) minus the current time. The highest priority is the job with the lowest slack value. For the slack per operation, the slack is divided by the number of operations remaining on the routing. Again, the highest-priority job has the smallest value. Another member of the due date family of rules is the Critical Ratio rule [15]. Jobs are sorted according to an index computed by dividing the time remaining (i.e., due date minus the current time) by the number of hours of work remaining. If the index is greater than 1, the job should finish early. If it is less than 1, the job will be late; and if it is negative, it is already late. Again, the highest-priority job has the smallest value of the critical ratio, in other words the highest "lateness" [39].

$$CR = \frac{due\ date - current\ time}{remaining\ process\ time} \qquad \textbf{Equation 2.3}$$

Most researchers have found that slack-per-operation consistently outperforms the other due date based rules [39].

2.7.3 Simple Rules Involving neither Processing Times nor Due Dates

The most commonly used rule involving a shop characteristic is First-In-First-Out (FIFO). With FIFO dispatching policy, waiting items are scheduled in the order of arrival. This rule is the only one considered that does not lead to a reordering of queue items. The idea is to work on the lot or job based on its ranking as to the order of its appearance at the machine in question. In this case, it's irrelevant to know which buffer the lot or job pertains to. The only criterion is the age of the lot with respect to all the

other jobs waiting to be processed. The older the lot, the higher the priority to process it. This imposes the existence of a tracking system (database), recording arrival times of all the lots on all the machines.

A number of researchers have found that the FIFO rule performs substantially the same as a random selection with respect to mean cycle time or mean lateness, although FIFO produces a lower variance of performance measures than does random selection. In general, FIFO has been found to perform worse than processing-time rules and due date rules with respect to both the mean and variance of most measurement criteria. Nevertheless, FIFO is an attractive alternative due to its simplicity of definition and usage.

Another rule tested by Rochette [55] was the number of operations remaining (NOP). This rule performed much worse with respect to mean tardiness than all other rules tested. Cownway [56] examined two 'look-ahead' rules: NINQ, which selects the job going next to the queue having the smallest number of jobs, and WINQ, which selects the job going to the queue containing the least total work. Both rules are intended to compete with SPT as they attempt to select jobs that can be processed rapidly through the next work station. However, they have greater mean cycle time than SPT, and generally perform worse that SPT for in-process inventory criteria.

The MIVP dispatching rule minimizes the difference between the instantaneous inventory and the average inventory profile [4, 48, 57, 58]. In a production line with mixed product or re-entrant lines, the operations' buffers contain WIP of different types. For each buffer, MIVP determines the average number of WIP of each type. These WIP

averages are used as baselines. WIP are prioritized to keep the buffers as closed to their baseline as possible. If an operation has in its buffer too much WIP of a certain type, then it will process this type in priority to reduce the number. Upstream operations will also slow down the processing of this type to avoid adding to the problem. In the other hand, if an operation has in its buffer not enough WIP of a certain type, then it will slow down the processing of this type, while upstream operations will prioritize this type to feed the buffer.

2.8 Release Policies

Production systems are categorized into two main families: push and pull production systems. In a Push system, such as MRP, work releases are scheduled. In a Pull system, releases are authorized [59]. The difference is that a schedule is prepared in advance, based on estimates of future demand. It is assumed that advance demand information is available, either in the form of actual orders, or forecasts, or a combination of both [60]. On the other hand, an authorization depends on the status of the plant. Because of this, a Push system directly accommodates customer due dates, but has to be forced to respond to changes in the plant (e.g., MRP must be regenerated). Similarly, a Pull system directly responds to plant change, but must be forced to accommodate customer due date (e.g., by matching a level production plan against demand and using overtime to ensure that the production rate is maintained) [11]. In other words, push systems are inherently make-to-order and pull systems are inherently make-to-stock [61]. Section 2.9 and 2.10 provides a review of the literature in respect to Push and Pull Production Control Strategies; mechanisms, advantages and known issues. Section 2.11 reviews the

literature in relation to reported comparison studies of various production control strategies.

2.9 Push Production Systems

A Push system schedules the release of items based on demand (outside information), which is inherent make-to-order (Figure 2.2).

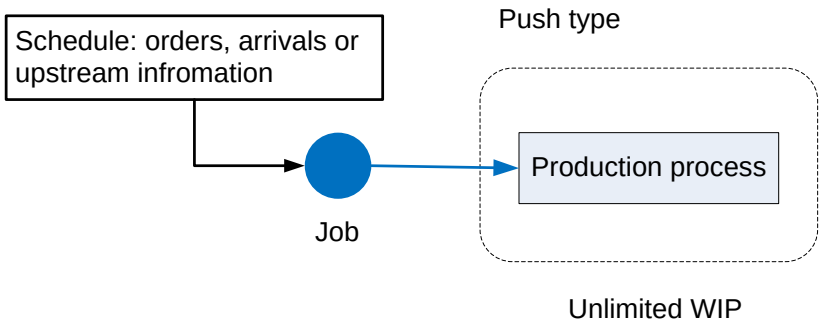


Figure 2.2: Push type production process [11]

Push system releases work into the system without a feedback loop which communicates the WIP status, thus the amount of WIP in the system can fluctuate essentially without bound. They are entirely controlled by external information (i.e., schedule). Examples of push release policies are the static policies and Material Requirement Planning (MRP).

Static policies [62, 63] release new jobs into the line at fixed interval time, independent of the line status. Thus it minimizes input variability and according to queuing theory should improve performance. MRP is working backward from a production schedule of purchase order to derive schedules for components (parts). MRP is therefore called a push system since it computes schedules of what should be started (or push) into production based on demand. First, the delivery schedule is set to meet customer

demands. All items have to be completed in time for delivery. So the items' completion date is set. From this completion date, and the process time at each operation, the start date is computed for each item at each station.

Figure 2.3 illustrates a simple example; it should be read from right to left (backward).

- Schedule target of item ABCD is 01/09/2010. To produce ABCD, components AB and CD are needed and the assembly time is 1 day. Therefore, the assembly must start the 31/08/2010 to be completed in time and both components also have to be ready the 31/08/2010
- Therefore the schedule target of components AB and CD is 31/08/2010. These components are an assembly of parts A, B, C and D. Due to assembly time, component AB assembly needs to start the 28/08/10 and component CD assembly needs to start the 29/08/2010. Parts A and B needs to be ready the 28/08/2010, parts C and D needs to be ready the 29/08/2010.
- Due to their respective processing time, part A needs to start processing the 26/08/2010, part B the 25/08/2010, part C the 28/10/2010 and part D the 25/08/2010.

In other words, the whole schedule has been determined. Parts A, B, C and D will be pushed in the line at the date scheduled without consideration of the current production line status.

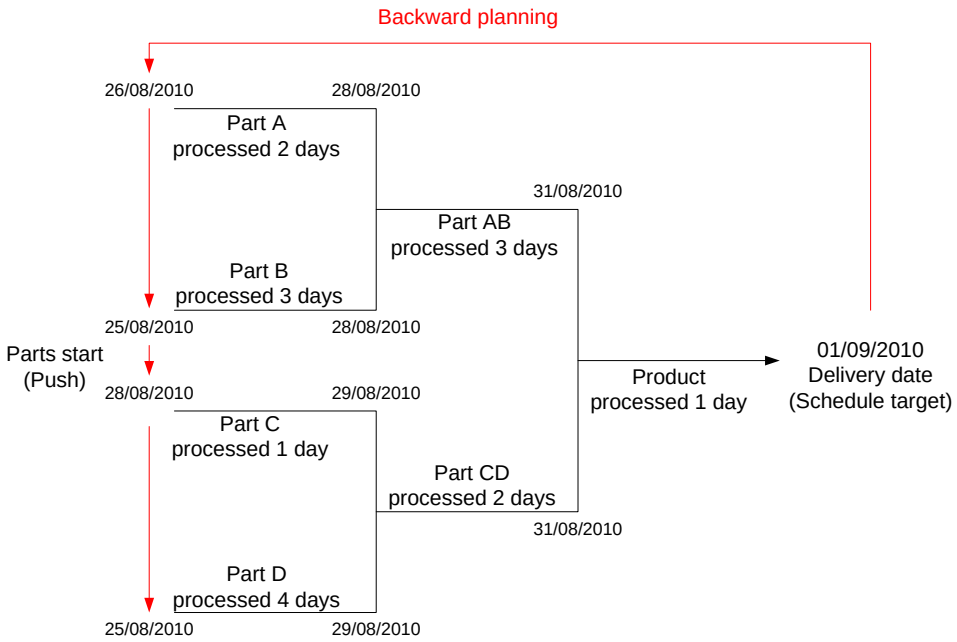


Figure 2.3: Backward requirements planning

The main focus of MRP is on scheduling jobs and purchase orders to satisfy material requirements generated by external demand. In a real factory involving many processing steps and many components, this determination becomes very complex and only computerised systems can schedule the production. MRP deals with two basic dimensions of production control: quantities and timing. The system must determine appropriate production quantities of all types of items, from final products that are sold, to components used to build final products, to inputs purchased as raw materials. It must also determine production timing (i.e., job start times) that facilitates meeting order due dates [11].

Two main issues can be reported with MRP: capacity infeasibility and long planned lead time.

- The basic model of MRP considers fixed lead time (cycle time) to compute the schedule. Fixed lead time is not a true behaviour in a real factory as the line

loading influence the lead time. MRP is taking a risk without considering the production line variability. For instance, a machine stops production due to downtime. This can render production schedules infeasible when product levels are at or near maximum capacity [59].

- For safety, a planner will typically choose pessimistic (Long) estimates for the planned lead times. For example, the average manufacturing lead time is 3 weeks. Since the actual lead times are variable, some will be less than the mean of 3 weeks and others will be greater. For safety, factory managers will plan a lead time of 5 weeks. The longer the planned lead times, the longer parts will wait for the next operation, and so the more inventory there will be in the system. Such behaviour results in a lack of responsiveness as well as high inventory level [11].

2.10 Pull Production Systems

Pull systems are closely associated with the Just-In-Time (JIT) practices. The manufacturing techniques behind the phenomenal Japanese success have become collectively known as Just-In-Time (JIT). They represent an important chapter in the history of manufacturing management. The most direct source for many of ideas represented by JIT is the work of Taiichi Ohno at Toyota Motor Company. His goal was to have each workstation acquire the required materials from upstream workstations precisely as needed or just in time. Just in time flow requires a very smoothly operating system. If materials are not available when a workstation requires them, the entire system may be disrupted. This has serious implications for the production environment. One means for avoiding disruptions is Ohno's concept of *autonomation*, which refers to

machines that are both *automated*, so that one worker can operate many machines, and *foolproofed*, so that they automatically detect problems [11].

JIT later developed into Lean production [11]. Lean goals are to improve quality and eliminate waste. Waste is any activity that consumes time, resources, or space but does not add any value to the product or service. Seven types of waste are identified:

- Transport (moving products that is not actually required to perform the processing)
- Inventory (all components, WIP and finished product not being processed)
- Motion (people or equipment moving or walking more than is required to perform the processing)
- Waiting (waiting for the next production step)
- Overproduction (production ahead of demand)
- Over processing (resulting from poor tool or product design creating activity)
- Defects (the effort involved in inspecting for and fixing defects)

Pull system authorizes the release of items based on system status (inside information), which is inherent make-to-stock (Figure 2.4). It allows in many cases a reduction of transport, inventory, motion and waiting wastes.

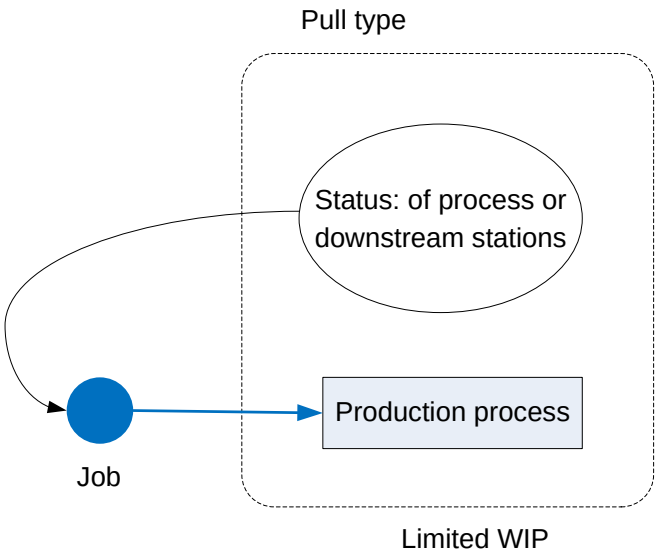


Figure 2.4: Pull type production process [11]

In pull systems, the status of the WIP level within the system is transferred up the line. This information is used to govern items release and maintain a pre-specified WIP level within the system. It triggers releases in response to insufficient WIP level, and prohibit releases when the pre-specified WIP level is reached. Hence, a Pull system will not let system WIP grow beyond the pre-specified WIP level. As a result the amount of WIP that can be in the system is controlled. To determine the optimum WIP level, shop characteristic curves (or "X" curves) representing the relationship between WIP and production rate may be developed (APPENDIX - B).

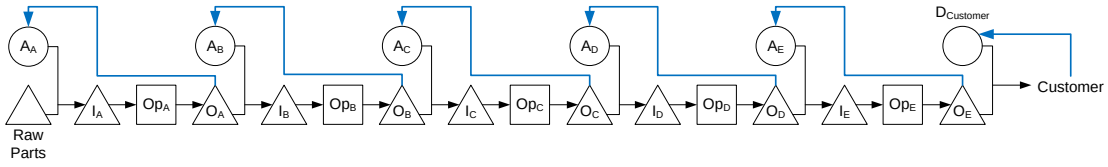
Issues reported with pull systems:

- In environment with multiple products, pull requires that a minimum inventory of each product be maintained at the output of each workstation. This can lead to proliferation of WIP inventories at each stage of the process [60].

- Pull systems involve optimising or standardising tasks and freezing production schedules. Hence, pull might not be the best material control strategy for environments with highly customised products or demands highly variable [60].

2.10.1 Kanban Control System (KCS)

The single technique most closely associated with the JIT practices of the Japanese is the "pull system" known as Kanban control system developed at Toyota. The word Kanban is Japanese for card, and in the Toyota KCS, cards were used to govern the flow of materials through the plant.



Op_i is Operation i (i: A to E) I_i is the parts Input buffer of Op_i
A_i is the queue for production's authorisations of Operation i
O_i is the parts Output buffer of Op_i $D_{customer}$ is the queue for customers' demands

Figure 2.5: Kanban control system [64]

Kanban control system (Figure 2.5) is triggered by a demand. When a part is removed from an inventory point (which may be finished goods inventory or some intermediate stock) the workstation that feeds the inventory points is given authorization to replace the part. This workstation then sends an authorization signal to the upstream workstation to replace the part it just used. Each station does the same thing, replenishing the downstream void and sending authorization to the next workstation upstream (Figure 2.5). In Kanban system, an operator requires both parts and authorization signal (Kanban) to work.

The main issues in modelling Kanban systems are: (a) determining the number of Kanbans for each product and (b) their allocation among the different stages of the manufacturing system [60]. It directly affects the trade-off between throughput and WIP inventory [65].

Kanban is difficult, or impossible to use [66] when there are

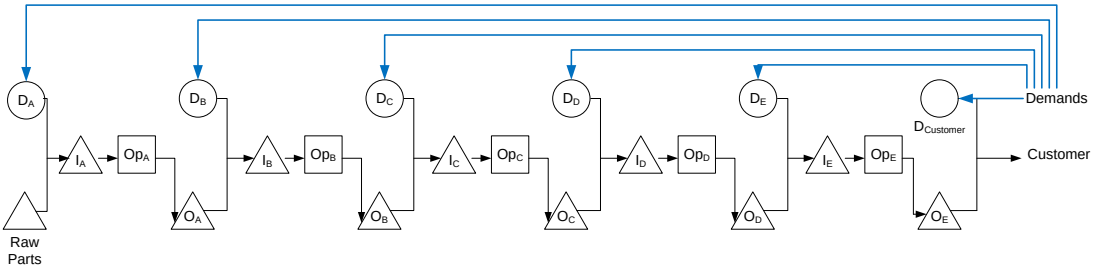
- Job orders with short production runs, or
- Significant set-ups, or
- Scrap loss, or
- Large, unpredictable fluctuations in demand

The two cards Kanban system made use of two types of cards. Production kanbans are used to authorize production within a work station, and withdrawal kanbans are used to pull parts, materials and sub-assemblies from other work stations [67]. Indeed, if the distance between the consecutive workstations is long, each work station will have separate inbound buffer and outbound buffer. Then a second card is needed to coordinate the items movement from the outbound buffer to the next work station inbound buffer [68].

2.10.2 Base Stock Control System (BSCS)

A simple pull control system used in inventory control is the Base Stock Control System (BSCS) [69]. In the BSCS (Figure 2.6), every stage has a target inventory of finished parts, called base stock. When a demand for an end item arrives to the system, it is immediately transmitted to every stage where it authorizes the release of a new part. The

advantage over KCS of this mechanism is that the system responds instantly to demand. Its disadvantage is that it provides no limit on the number of parts in the system.

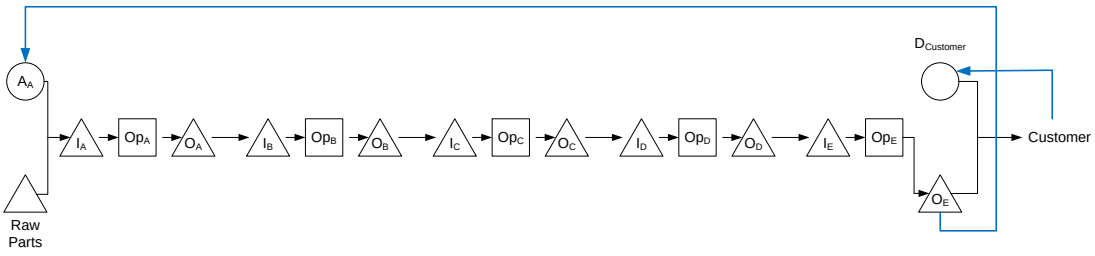


Op_i is Operation i (i: A to E)
D_i is the queue for demands at Operation i
$D_{customer}$ is the queue for customers' demands
I_i is the parts Input buffer of Op_i
O_i is the parts Output buffer of Op_i

Figure 2.6: Base Stock Control System [64]

2.10.3 Constant Work in Process Control System (CONWIP)

CONWIP (Constant Work In Process) [11, 15, 70-72] establishes a limit on the WIP in the line and simply does not allow releases into the line whenever the WIP is at or above limit. A new job is introduced to the line each time a job departs (Figure 2.7) and results in a WIP level that is very nearly constant. To be effective, a reasonable maximum level of WIP for the flow must be established. If this level is too low (i.e., near the critical WIP), throughput will suffer. If too high, then cycle time will be excessive [11].



Op_i is Operation i (i: A to E)
I_i is the parts Input buffer of Op_i
A_A is the queue for production's authorisations of the whole line
O_i is the parts Output buffer of Op_i
$D_{customer}$ is the queue for customers' demands

Figure 2.7: CONWIP release strategy [64]

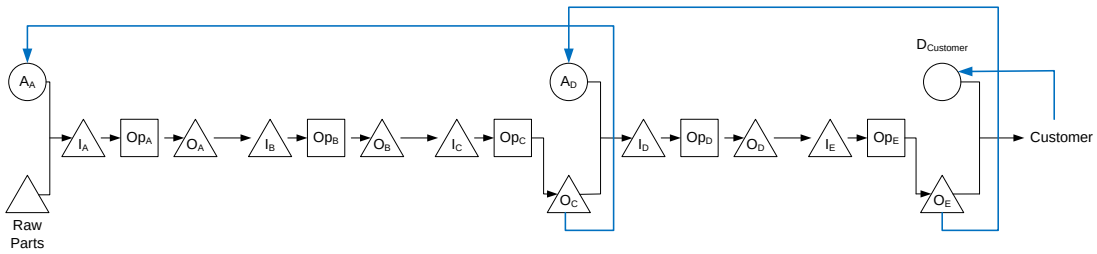
CONWIP implicitly assumes two things:

(1) The production line consists of a single routing, along which all parts flow.

(2) Jobs are identical, so that WIP can be reasonably measured in units (i.e., the number of jobs or parts in line).

In such situations, the basic CONWIP protocol (i.e., start a new job whenever one in proves finishes) can be easily and effectively used for shop floor control.

Nevertheless, very long line should not be run as a single CONWIP loop. For instance, one should not create a single CONWIP loop spanning an entire semiconductor fab ─ there are simply too many steps. A long CONWIP line begins to behave like a push system. That is, when the WIP cap is large (because the line is long) WIP can accumulate in sections of the line and be unavailable in others. This creates "WIP bubbles," which disrupt flow and thereby defeat the flow smoothing role of a pull system. Fortunately, a long line can be broken into several tandem lines. One way to do this is to control the line as several tandem CONWIP loops (Figure 2.8) separated by WIP buffers [73]. The WIP levels in the various loops are held constant at specified levels. The inter-loop buffers hold enough WIP to allow the loops to temporarily run at different speeds without affecting (blocking or starving) one another. This makes it easier for different managers to be in charge of the different loops. However, the extra WIP and cycle time introduced by the buffers degrade efficiency. This is trade-off one must evaluate in light of the particular needs of the manufacturing system. The more CONWIP loops the line is broken into, the closer its behaviour will be to Kanban.



Op_i is Operation i (i: A to E) I_i is the parts Input buffer of Op_i
A_i is the queue for production's authorisations of the loop starting at Op_i
O_i is the parts Output buffer of Op_i $D_{customer}$ is the queue for customers' demands

Figure 2.8: Tandem CONWIP loops [64]

If one loop is a clearly defined bottleneck, however, it may be decoupled from the rest of the line. This will let the loop run as fast as it can (i.e., to work ahead), subject to availability of WIP in the upstream buffer and subject to a WIP cap on the total amount of inventory than can be in the line at any point in time. Of course, this means that the WIP in the downstream buffer can float without bound, but as long as the rest of the line is consistently faster than the bottleneck loop, the faster portion will catch up and therefore WIP will not grow too large. Of course, in the long run, all the CONWIP loops will run at the same speed ─ the speed of the bottleneck loop [11].

While it is certainly simplest from a logistics standpoint if machines are dedicated to routings, other considerations sometimes make this impossible. Shared resources complicate both control and prediction of CONWIP lines. If the facility contains multiple routings that share workstations, CONWIP levels can be established along different routings.

If different jobs (product mix) require substantially different amounts of processing on the machines, then things are not so simple. The reason is that the total workload in the line may vary greatly because of the difference in processing times across products. To

use CONWIP in these settings the policy must be expanded. CONWIP levels can be stated in units of “standardized jobs”, which are adjusted according to the amount of processing they require on critical resources [11]. For example, the WIPLOAD policy measures the overall workload on the shop floor as the sum of the remaining processing times of all the items on the shop floor [74]. Each time an item go through one of the operation, the WIPLOAD is reduced by this operation processing time. New items are then released into the line to maintain the WIPLOAD constant at a prescribed level.

Therefore, CONWIP can be applied to a very broad range of production environments. Of course, greater system complexity generally implies greater variability and hence lower efficiency. Nevertheless, the WIP cap provided by CONWIP will prevent inventory from growing without bound, which will make the system more stable and manageable. The following conditions are needed for CONWIP to work well:

(1) The loop should not be too long. The line can be broken into several tandem lines.
(2) Part routing can be grouped into a small number of product flows. Each flow will make up a CONWIP loop.
(3) There must be a measure of WIP. In some systems, this can simply be a count of the units in the system. But in systems where different part types require vastly different process times, it makes sense to measure the WIP in terms of processing time required.

Two problems that can arise with CONWIP (or Kanban) in certain environments are the following:

(1) Premature releases due to the requirement that the WIP level be held constant.

(2) Bottleneck starvation due to downstream machine failures.

While the issue of premature releases is not a common problem in lines operating close to capacity, it is a major concern in low utilization routings. Even if a part will not be needed for months, a CONWIP system may trigger its release because CONWIP in the loop has fallen below its target level. In plants with many routing (e.g., a plant tending toward a "job shop" configuration), some routings may not be used for substantial periods of time. Clearly, under these conditions a constant WIP level should not be maintained along the routing, since this would result in releasing jobs that are not needed until far in the future. A simple way to prevent this is to establish an "earliest start date" for jobs in release list [11].

The problem of bottleneck starvation is at the center of the theory of constraint. Indeed, any starvation of the bottleneck results systematically into lost capacity and reduced throughput for the whole line. Therefore there should be enough inventories in the line to preclude the bottleneck starvation. Simultaneously, long queues in front of the bottleneck have to be avoided to keep low inventory. In other words, items should arrive as late as possible to the bottleneck machine, just in time to prevent the bottleneck starvation [44]. A proper scheduling of the arrivals at the bottleneck is important. It is the object of dedicated release strategies (see Theory of Constraint section, p58).

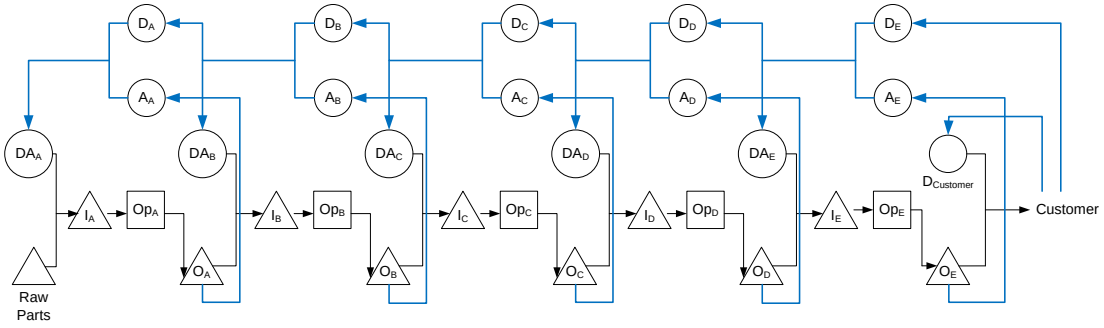
2.10.4 Two Parameter Kanban Systems

Several authors [69, 75-77] propose to mix Kanban Control System (KCS) and Base Stock Control system to introduce new control systems depending on two parameters per stage.

Indeed the characteristic of the KCS is that in every stage a single parameter, the number of kanbans, plays two roles: (i) it is the number of cards used to authorize the production of new parts at the stage; and (ii) it is the base stock of finished parts of the stage. This "two-roles-in-one-parameter" characteristic may lead to bad system performance especially when demand or effective processing times are highly variable. For instance, in a situation of high demand variability, one would like to have a "large" number of kanbans at times of high demand, to quickly respond to demand. At the same time, one would like to have a "small" number of kanbans at times of low demand, to reduce inventory costs, since the number of kanbans is equal to the target inventory of finished parts [69]. To compromise these two tendencies, one would end up setting the number of kanbans somewhere in between "large" and "small", thus meeting neither objective (quick customer response and low in-process inventories) too well. Indeed, it has often been reported that kanban control does not work well unless demand and the flow of parts are fairly constant [69].

The Extended Kanban Control System (EKCS) [69, 75, 76] and the Generalized Kanban Control system (GKCS) [77] are modified Kanban schemes where one of the parameters (K) controls the maximum WIP in the stage and the other (S) determines the number of products that should be stored at the stage's output inventory. The difference between these two systems and KCS is that in the KCS, initially, all Operation-*i*'s kanbans are attached to an equal number of finished parts in Output i, there are no free kanbans. In the GKCS (Figure 2.9) and EKCS (Figure 2.10), on the other hand, there are S_i kanbans attached to an equal number of finished parts in Output i, but there are also $K_i - S_i$ free kanbans in A_i.

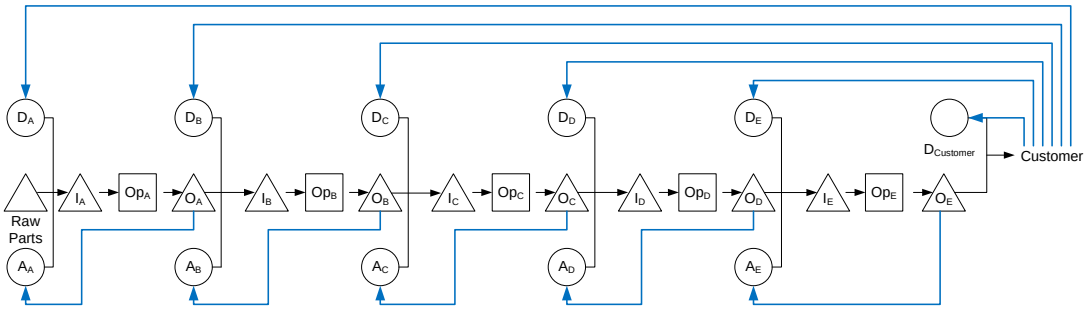
In GKCS, theses extra kanbans allow for the partial decoupling of the transfer of parts downstream of Operation i and the transfer of demands upstream to DA_i.



Op_i is Operation i (i: A to E) I_i is the parts Input buffer of Op_i
A_i is the queue for production's authorisations of Op_i
O_i is the parts Output buffer of Op_i $D_{customer}$ is the queue for customers' demands
D_i is the queue for production's demands at Op_i DA_i is the queue for authorisation-demand pairs of Op_i

Figure 2.9: Generalized Kanban Control System [64]

In EKCS, these free kanbans may authorize an equal number of new parts to be released in to Operation i, however, in order to authorize the release of any part above this number, a finished Operation-*i* part must leave Output *i*.



Op_i is Operation i (i: A to E) I_i is the parts Input buffer of Op_i
A_i is the queue for production's authorisations of Op_i D_i is the queue for production's demands at Op_i
O_i is the parts Output buffer of Op_i $D_{customer}$ is the queue for customers' demands

Figure 2.10: Extended Kanban Control System [64]

Both GKCS and EKCS require the presence of both a free Kanban and a Demand to release a part from an Output buffer i. The two systems differ in the transfer of Demands to the Operations. GKCS requires the presence of a free Kanban in A_i to transfer a

demand from D_i to D_{i-1} as in KCS. On the other hand, in EKCS a Demand is transmitted immediately to all operations as soon as it received as in BSCS.

These two parameter kanbans control systems are shown to provide more flexibility in system performance under dynamic environments with variable demands and variable processing time. EKCS operation is simpler than that of the GKCS and response time is shorter. This is however at the expense of higher bounds on the number of finished parts in the system [69].

2.10.5 Theory of Constraint (TOC)

The theory of constraints was introduced by Goldratt [78]. It is based on the premise that the rate of goal achievement is limited by at least one constraining process. Only by increasing flow through the constraint can overall throughput be increased. This can be achieved by:

1. Identify the constraint.
2. Optimize the utilization of the constraint (get the most capacity out of the constrained process $U \approx 1$).
3. Restructure the whole system organization to achieve the previous step.
4. Increase the constraint capacity
5. If, as a result of these steps, the constraint has moved, return to Step 1.

Several release strategies [11, 70, 79-85] propose various CONWIP adaptations to calculate the workload target and optimize the utilization of the bottleneck. The main difference from CONWIP is that they are pulling from the bottleneck instead of pulling from the end of the line.

The simplest version is the Pull-From-Bottleneck (PFB) strategy [11] also named Starvation Avoidance (SA) [79].

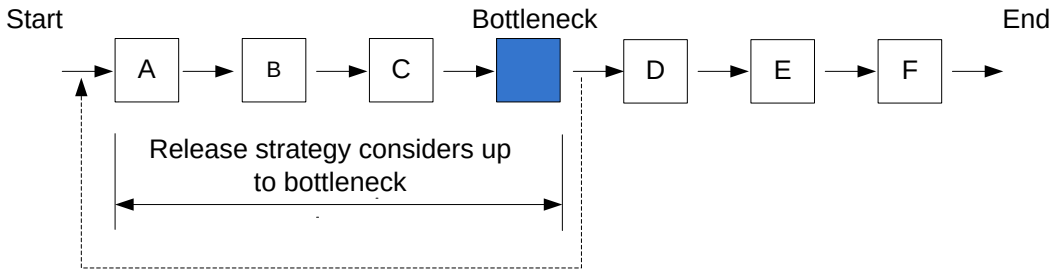


Figure 2.11: Pull from bottleneck strategy

Figure 2.11 shows such a basic chart for a single line. This mechanism differs from CONWIP in that the WIP level is held constant in the machines up to and including the bottleneck, but is allowed to float freely past the bottleneck. Since machines downstream from the bottleneck are faster on average than the bottleneck, WIP will not usually build up in this portion of the line. However, if a failure in one of these machines cause a temporary build-up of WIP, it will not cause the bottleneck to shut down, as can occur under CONWIP if card deficits are not used.

Once again, if different jobs (product mix) require substantially different amounts of processing on the machines, then things are not so simple. The same number of units does not present the same use of resources. Hence, in order to maintain a level loading on the system, it makes sense to measure the WIP in terms of time required at the bottleneck [11].

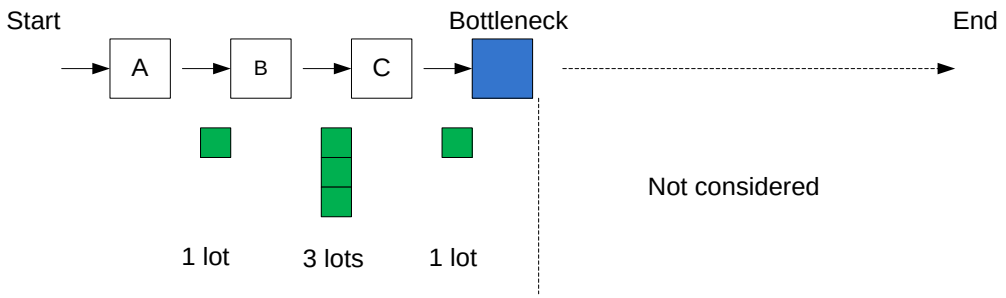


Figure 2.12: WORKLOAD regulation

Workload regulation release strategy [86] (Figure 2.12) measures the amount of processing time at the bottleneck that is currently represented by the items being processed in the fab. Items are released into the fab if the current workload plus the total amount of bottleneck processing time of this item is less than a given processing time target. So, new items are released to maintain the target workload. Each time an items leaves the bottleneck operation the workload is reduced by the bottleneck processing time.

For example, there are 5 items remaining until bottleneck operation (Figure 2.12), and the target workload is 90 minutes. Operation B has 1 item remaining; this item processing time in bottleneck operation is 20 minutes. Operation C has 3 items remaining; each item processing time in bottleneck operation is 10 minutes. And finally, the bottleneck operation has 1 item remaining whose processing time is 10 minutes. Because of these 5 items remaining until bottleneck operation, the total current bottleneck workload is 60 minutes (20 min + 30 min +10 min = 60 min). But the target is 90 minutes; there is still space to release new items. For example, three items requiring 10 minutes processing time at the bottleneck or one item requiring 20 minutes processing time plus one item requiring 10 minutes processing time. The target workload is decided by the factory manager.

If the bottleneck shifts depending on product mix, then it is not clear where the pulling point should be located, and therefore one may be just as well off pulling from the end of the line (i.e., using regular CONWIP), possibly with a card deficit policy [11].

CONLOAD release strategy [81] is a simple extension of the workload regulation. Instead of considering the amount of working time for bottleneck operation, the amount of load for bottleneck operation is computed, i.e., the sum of bottleneck processing times of the lot divided by the average cycle time of lots of this product. A new item is allowed to enter the line if the current bottleneck load plus the load introduced by the new item is less than a given target.

While in the workload regulation, the processing time target depends on the factory manager's judgment, in CONLOAD, the load's target can be calculated from the desired bottleneck utilization. It is the optimum utilization multiplied by number of machines in bottleneck operation. For example, for an optimum utilization of 0.95, and four machines in bottleneck operation, the load's target is 0.95 multiplied by 4 equal to 3.8. A new lot is introduced only if its introduction does not increase the load above 3.8. Therefore CONLOAD is much clearer about the target level compared to the workload regulation.

The Drum-Buffer-Rope release strategy [82] differs from PFB in that it is counting the number of items leaving definitively the bottleneck instead of the number of items in the line. For a simple process line without reentrancy, DBR counts the WIP processed by the bottleneck operation and releases the same amount of items into the line [82]. For

example, if in a shift the bottleneck operation processes 50 items, then the following shift releases 50 items.

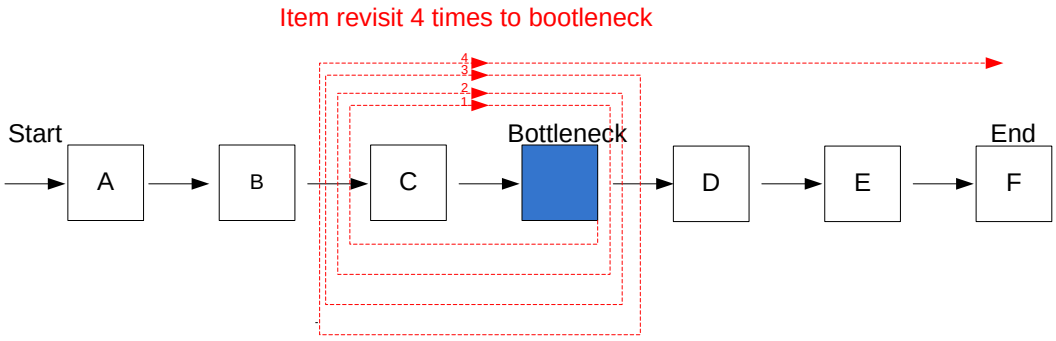


Figure 2.13: Re-entrant figure

When the production line become more complex with re-entrant lines, some of the items will revisit the bottleneck, and it is not as straightforward to calculate the release [72]. DBR needs to determine the number of items processed by the bottleneck that will not revisit it. For example, if items revisit 4 times the bottleneck (Figure 2.13), and the total output of the bottleneck operation is 16 items, the actual number of items not revisiting the bottleneck is 4 in most cases (bottleneck output/re-entrant times). And only 4 new items need to be released in the line. When the system becomes more complex, then the formula to calculate the release becomes more and more complex.

2.10.6 Hybrid Push-Pull Systems

The production of the earlier upstream stations is controlled by push type production, while the production of the later downstream stations is controlled by pull-type production (Figure 2.14). The hybrid system often compromises the conflicting performance characteristics of the push and the pull environments [87]. The general hybrid push/pull system has a series of pure push stations followed by a series of pure

pull stations. These push and pull type systems are combined at an integration point (or a junction point), which is a safety stock inventory in the form of semi-finished products after the last push station [87].

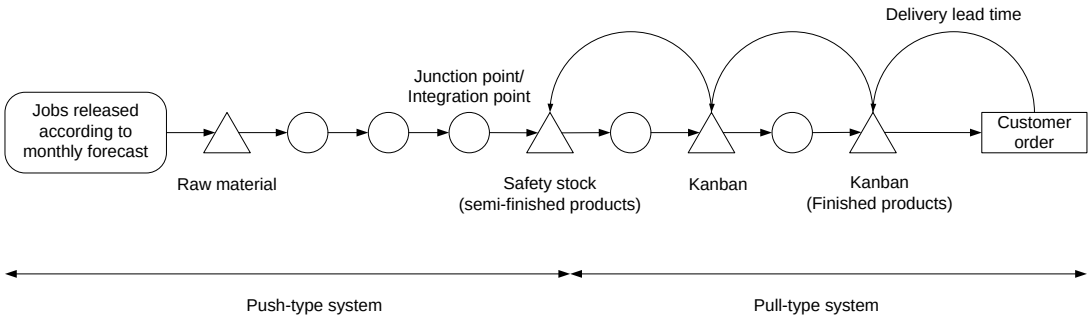


Figure 2.14: A generic hybrid push/pull manufacturing system [87]

All production quantity decisions and item releases are derived from MRP planning systems [88]. Items are pushed into the line following the MRP planning and then are processed from operation to operation until they reach the safety stock inventory. Items have to wait in the inventory until they are pulled from the inventory into the second part of the production line. The second part of the line is controlled by a Kanban system managing the items from the safety stock inventory to the finished product inventory.

Huang [89] developed a more advanced system, for manufacturing environments where items have to go through distinctive process sections. Six rules are used to determine whether a section should apply a push or pull approach. Therefore items go through a sequence of pull and push sections as determined by the rules.

The hybrid push/pull system compromise the conflicting performance characteristics from both push and pull type systems, that is high supply reliability and low inventory holding [87, 88, 90] as well as increased productivity and machine utilization [89].

2.11 Release Policies Comparison

First, it is important to understand that it is not possible to assume that one release policy performs better than another under all circumstances. In other terms, no item release policy dominates across all scenarios [6]. Nevertheless trends can be noticed.

2.11.1 Comparison of Push and Pull Production Systems

A fundamental distinction between Push and Pull systems is that Push systems control throughput and observe WIP, whereas Pull systems control WIP and observe throughput.

All pull systems ensure that, no matter what happens on the plant floor, the WIP level cannot exceed a pre-specified limit. By establishing a WIP *cap*, Pull systems place a very strong emphasis on material flow; if product stops, inputs stop [11]. If WIP is capped, then disruptions in the line (e.g. machine failures, shutdowns due to quality problems, slowdowns due to product mix changes) do not cause WIP to grow beyond a predetermined level. However, depending on what happens in the line, the output rate may vary over time.

In a pure push system, no such WIP limit exists. For example, in MRP, a master production schedule is established, which determines planned order releases. These, in turn, determine what is released into the system. Depending on what happens in the line, however, the WIP level may float up and down over time. The key point here is that in a push environment, corrective action is not taken until after there is a problem and WIP has already spiralled out of control.

Between pull and push production systems which one is better? While this is not a simple question, some observations can be made [11].

First, and fundamentally, WIP is directly observable, while throughput is not. Hence, setting WIP as the control in a pull system is comparatively simple. Items can be physically counted on the shop floor to maintain compliance with a WIP cap. In contrast, setting the release rate in a push system must be done with respect to capacity. If the rate chosen is too high, the system will be choked with WIP; too low, and revenue will be lost because of insufficient throughput. But estimating capacity is not simple. A host of detractors, ranging from machine outages to operators unavailability, are difficult to estimate with precision. This fact makes a push system intrinsically more difficult to optimize than a pull system [11].

A second argument in favour of Pull systems is that for serial lines manufacturing a single product, they are more efficient than Push systems. More efficient means that the WIP level required to achieve a given throughput is lower in a pull system than in a push system [91]. And for a given level of throughput, a push system will have longer average cycle times than an equivalent Pull system [11].

Third, Push systems also have more variable cycle times than equivalent Pull system [59]. Increased cycle time variability means that longer lead times must be quoted in order to achieve the same level of on-time delivery. This is because, to achieve a given level of on-time delivery, the mean cycle time plus some multiple of the standard deviation of cycle time must be quoted. For example, if the cycle time is 10 day +/- 1 day, then for safety an operation manager will quote a lead time of 11 days. Whereas with the same average cycle time of 10 days but fluctuations of 3 days, the operation manager will have to quote 13 days for safety. So a bigger variability implies longer lead times. Thus, for the same throughput and customer service level, lead times will be

longer in the push system for two reasons: longer mean CT and larger standard deviation of cycle time.

Finally, the most important advantage of a Pull system over a pure Push system is neither the reduction in WIP (and advantage cycle time) nor the reduction in cycle time variance, important as these are. Instead, the key advantage of pull systems is their robustness, which can be stated as follows [11]: "A CONWIP system is more robust to errors in WIP level than a pure push system is to errors in release time". In other words, errors in WIP level are less damaging on profit than errors in release rate.

These benefits urge operation managers to incorporate aspects of pull into manufacturing control systems. Unfortunately, from a planning perspective, there are drawbacks to pull systems. There is no natural link to customer due dates in a pull system. Until customers "pull" what they need, the system offers no information for planning raw material procurement, staffing, opportunities for machine maintenance, and so on. In contrast, push system can be operational nightmares but are extremely well suited to planning. There is a simple and direct link between customer due dates and order releases in a push system. MRP is generally considered to be applicable to many more manufacturing firms than is Kanban. But Kanban seems to produce superior results when it can be applied [91].

2.11.2 CONWIP vs. Kanban

CONWIP and Kanban are both pull systems in the sense that release into the line are triggered by external demands. Because both systems establish a WIP cap, they exhibit similar performance advantages relative to MRP. Specially, both CONWIP and Kanban

will achieve a target throughput level with less WIP than a pure push and will exhibit less cycle time variability. Moreover, since both are controlled by setting WIP, and as can be known that WIP is a more robust control than release rate, they will be easier to manage than a pure push system. However, there are important differences between CONWIP and Kanban. The most obvious difference is that Kanban requires setting more parameters than does CONWIP. This fact means that CONWIP is intrinsically easier to control [11].

Hall [92] pointed out that Kanban is applicable only in repetitive manufacturing environments in which material flows along fixed paths at steady rates. Large variations in either customers' orders or product mix destroy this flow and seriously undermine Kanban. This is due to the information delay that occurs in a KCS line. Therefore, the release rate is not easily adjusted to match changes in the demand rate [93]. CONWIP, while still requiring a relatively steady volume is much more robust to swings in product mix, as a result of the planning capability introduced by the process of generating a release list, and is applicable to a wider variety of production environment [72]. In Kanban, the optimal card count allocation is a function of mix. Hence, to achieve high throughput with low WIP, this may need to dynamically vary the card counts over time. This could be a difficult task. In CONWIP, the WIP will naturally accumulate in front of the bottleneck, right where it is needed. Hence, CONWIP will tend to produce higher utilisation of the bottleneck, and therefore greater throughput than KANBAN [91]. Happily, this all happens without intervention, because of the natural forces governing the behaviour of bottlenecks. If the mix of products change, and result in a change in the

bottleneck operation, then the largest queue will shift by itself to the new bottleneck. CONWIP system is fundamentally simpler to manage than a Kanban system [11].

BSCS and two parameters Kanban tend to maintain similar overall inventory levels as CONWIP. However, EKCS tend to maintain more of this inventory internally in the line, i.e. in a semi-finished state, than CONWIP. This may be either an advantage or a disadvantage and will depend on the manufacturing objectives of the organisation [93].

2.11.3 TOC vs. Non-TOC

In lines where all parts follow the same routing and processing times are such that the same process is the slowest operation for all parts, the bottleneck plays a key role in the performance of the line and therefore should be given special attention. Because throughput is a direct function of the utilization of the bottleneck, it makes sense to trigger release into the line according to the status of the bottleneck. All the articles [80, 81, 83] comparing TOC release strategies with CONWIP show that TOC release strategies outperform CONWIP with respect to throughput for a given WIP level.

Indeed, if there is one tool broken down after the bottleneck, for example operation E in Figure 2.11, WIP will be held in operation E. Eventually, no items will output from the line (operation F starves). Under those conditions CONWIP will stop releasing items into the line. If the breakdown lasts long enough, the bottleneck starves. This fatally impacts throughput as the bottleneck cannot increase its production speed to compensate for the time lost.

TOC release strategies do not take care of the WIP after the bottleneck operation. So even if operation E is broken down, they release items into the line and no capacity

(throughput) is lost at the constraint. Indeed the queue will build in front of operation E. But when operation E is up again, it can absorb the queue thanks to its high capacity (higher than constraint) [79].

If the breakdown occurs before the bottleneck (operation B in Figure 2.11), TOC release strategies and CONWIP stop the release of new items into the line. Nevertheless, TOC release strategies will react faster as the bottleneck starves earlier than operation F. Thus the number of items queuing in front of operation B is reduced. This also presents the second advantage of reducing the WIP bubble when operation B is back in working order thus helping the bottleneck to cope with it.

Nevertheless, in spite of the theoretical importance of bottlenecks, it has been experienced that few manufactures can identify their bottleneck process with any degree of confidence. The reason is that few manufacturing environments closely resemble a single-product, single-routing line. Most systems involve multiple products with different processing times. As a result, the bottleneck machine for one product may not be the bottleneck for another product. This can cause the bottleneck to "float", depending on the product mix [11]. This discussion has two important implications:

(1) Stable bottlenecks are easier to manage. A line with a distinct identifiable bottleneck is simpler to model and control than a line with multiple moving bottlenecks. A manager can focus on the status of the bottleneck and think about the rest of the line almost exclusively in terms of its impact on the bottleneck (i.e., preventing starvation or blocking of the bottleneck).

(2) Bottlenecks can be designed. Although some manufacturing systems have their bottleneck situation more or less determined by other consideration (i.e. the capacity of all key processes would be too expensive to change), often bottleneck can be proactively influenced. For instance, the number of potential bottleneck can be reduced by adding capacity at some stations to ensure that they virtually never constrain throughput. This may make sense for stations where capacity is inexpensive. Alternatively, product sharing section of their processing lines can be separated by adding additional machines and isolating each line.

The simplest plant to manage is one with separate routings and distinct, steady bottlenecks. If the line is plagued by a floating bottleneck that could be eliminated via inexpensive capacity, the addition deserves consideration. If shared routings could be separated without large cost, it should be looked into [11].

2.11.4 SA vs. DBR

SA and DBR are two types of TOC release strategies. Their performances are very similar [72]. Nevertheless SA is counting the WIP, which is easy to apply, while DBR is counting the output of bottleneck to calculate the release, which is not easy to apply when the line becomes more complex due to reentrancy. Thus SA is easy to apply like CONWIP and it shares similar performance with DBR. SA appears as a good compromise between DBR and CONWIP release strategies [72].

2.12 Literature Review Key Insights

Overall, the purpose of this study is to contribute understanding on the control of cycle time and cycle time variability in a front end semiconductor production line.

Not surprisingly, the literature review confirmed that variability exists in all production systems and can have a large impact on performance. All sorts of process and flow factors contribute to cycle time variability in a factory.

Reducing the effects of variability, through the utilization of an appropriate scheduling policy, appears as a promising solution to control cycle time and cycle time variability. Therefore, a dispatching policy and a release policy need to be determined. Firstly, dispatching policies are particularly important when various products/steps compete for the same machines. In this case, dispatching rules decide which item is to be processed first. In other words, dispatching policies are useful when the production line processes a mix of products or includes re-entrant lines. With a single product and without re-entrant line, a simple FIFO policy is sufficient. Moreover, several papers in the literature have concluded that the lot release policy has a bigger impact on the fab performance than the dispatching rule [6, 79, 83, 86]. These two reasons explain why FIFO is the most commonly used dispatching policy in the literature and also why only FIFO will be considered in this study.

Secondly, the release policy has to be determined. Release policies belonging to the theory of constraint seem to have a slight advantage in term of performances. At least, this is true in production line where the bottleneck is not fluctuating due to product mix or re-entrant mix. Nevertheless, the difference is not sufficient to choose a specific

release policy without considering the exact scenario. It appears that the performance of any release policy is mainly dependent on the peculiar characteristics of the studied production line. As mentioned earlier (p64), it is not possible to assume that one release policy performs better than another under all circumstances. In those circumstances, a release policy adapted to the various factors of variability cannot be developed until full understanding of these peculiar characteristics.

2.13 Production Managers' View and Contribution

The study was developed in collaboration with a front-end semiconductor manufacturing company and the production managers participated in the development of the research objectives. Indeed, the managers' experience of semiconductor manufacturing is valuable to determine the key issues encountered in production lines. Therefore, the literature review's key insights were presented to the production managers for an assessment.

Firstly, the managers pointed out that the variability created by process factors has for origin specific toolsets or operations. A metric is essential to investigate operation individually and identify sources of process variability in the production line.

Objective 1: Develop a metric to measure the amount of variability created by an operation.

Then, the various sources of variability presented in the review were analyzed. Discussions with factory managers revealed that their influences in front-end semiconductor line vary. Indeed, production is well monitored to detect immediately defects and reduce the amount of lot reworked. Also, the role of operators is less and

less predominant in production lines as they get automated. Items are transported and processed without human interference. In front-end lines products are not mixed. Tools dedicated can be treated as separate tool operations of low capacity. Hot lots are the result of variability and not the source of origin. Therefore, from the point of view of factory managers all those sources of variability can be excluded from the study of modern front-end semiconductor production line. As a result, three predominant sources of variability ─ unscheduled equipment failures, batch processing and re-entrant lines ─ emerge from the analysis. Therefore, in agreement with factory managers, these three sources of variability are targeted in this study for investigation.

These three factors cannot be studied simultaneously. Too many variables would considerably increase the difficulty. The scope of this study will be limited to the detailed analyses of tool availability and batching only. Then the results obtained will be tested in a re-entrant environment. Interesting areas for policy, practice and future research for the academic community, will be highlighted.

Objective 2: Understand and explain the impact of tool availability and batching on cycle time and cycle time variability.

Managers also pointed out that some operations have high output variability, but they do not have the highest impact on the overall production line. Thus, the interactions between operations are keys to the reduction of process flow variability. The circumstances easing or emphasizing variability in the flow have to be exposed.

Objective 3: Determine how the variability in the flow of items is affected by the interaction between operations.

A fourth objective emerged while the previous objectives were studied. It was not a priori objective, but was identified based on the first phases of work actually done.

Objective 4: Determine a release strategy able to control batching and tool availability disturbances through the use of the interactions between operations.

Indeed, the release policy should be determined from the lessons learnt from the study of variability in objectives 1, 2 and 3. The development of a new release policy should be a product of the previous simulation and modeling process. Moreover, according to production managers, semiconductor fabs generally work on a push system. Some factories have WIP management control which can in some cases replicate a 'pull' system of sorts, but due to the in-line variability inherent in semiconductor manufacturing, the typical results is that WIP is pushed through the fab. Consequently, this study will analyze the sources of variability using the simplest release policy, in other words static policies. Then, from the lessons learned, a release policy adapted to the circumstances will be tested.

But first, to fulfill the goals and objectives, a deeper understanding of the phenomena occurring in a real production line need to be acquired. From the literature review, a basic grasp of the possible sources of variability has been developed. However, information about the flow of items through a process line is required to answer basic questions such as: How are the items moving from one operation to the next? How long are they staying at each operation? When and where are queues occurring? A manufacturer was contacted to get access to their production data and an exploratory pre-study was initiated.

CHAPTER - 3 PRE-STUDY: REAL PRODUCTION LINE DATA SAMPLE

3.1 Introduction

This analysis was undertaken to better comprehend the nature of the problem. Access to production data was obtained from the semiconductor manufacturing company. The analysis of the data was performed using several statistical calculations. This preliminary work was needed to gain familiarity with the phenomenon in the situation and understand what is occurring, before a model is developed and a rigorous design is setup for comprehensive investigation.

The sample of data provided by the manufacturer included the movement of all the items through the whole production line for a period of one week. It was presented in an Excel document. Each row corresponded to one lot movement and included the following information: lot ID number, current operation ID number, product type, previous operation ID number, date of output from previous operation, date of arrival to the current operation, date of output from current operation. A chart (Figure 3.1) illustrates the meaning of the various entries in the spreadsheet.

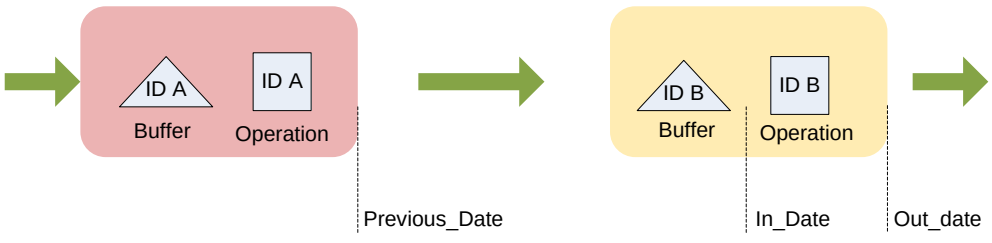


Figure 3.1: Items flow information

The interval time between In_date and Previous_date was assimilated to a queue time in the operation buffer. In reality, it also includes the transport time, but this information cannot be accessed. The manufacturer indicated that transport time was small compared to buffer time.

First, work concentrated on the development of a metric for variability, then on the characterization of lot arrival and lot departure and finally on queue time analysis.

3.2 Variability Measurement

The objective of this investigation is to analysis the variability of Inter-Departure Time (I-DT) in the production line. More accurately, the target is to find which operations create variability and which remove variability. The investigation addresses several possible metrics: Coefficient of Variation of Inter-Departure Time ($CV_{I\text{-}DT}$), difference in variability between departure and arrival times [$(CV_{I\text{-}DT}) - (CV_{I\text{-}AT})$], and their ratio [$(CV_{I\text{-}DT}) / (CV_{I\text{-}AT})$].

3.2.1 Coefficient of Variation (CV)

There are several measurements to address, lots arrival time (Previous_date), queue time, process time and departure time (Out_date). Queue time is the time between previous_date and in_date [(In_date) – (Previous_date)]. Process time is the time between In_date and Out_date [(Out_date) – (In_date)]. In the production line, lots are not following a linear sequence of operation numbers. Their route depends on the type of product and all kinds of complicated process steps. Usually they cross over to different operations, which make very difficult to trace them from one operation to the

next. Therefore, the default classification follows the time sequence of lots entering the various operations. One schedule sample is shown in Table 3.1.

Table 3.1: Sample of lots sequencing

Lot number	Operation ID	Arrival time	Queue time (min)	Process time (min)	Departure time
Lot 01	D	04/07/2007 18:00	30	30	04/07/2007 19:00
Lot 12	A	05/07/2007 00:00	40	30	05/07/2007 01:10
Lot 09	B	05/07/2007 01:30	20	30	05/07/2007 02:20

In order to exploit these data, their arrangement has to be reorganized. In the following various sections, different classifications will be used in function of the objectives pursued.

This section addresses the calculation of inter-arrival time (and inter-departure time) coefficient of variation. So the Arrival Times (AT) and Departure Times (DT) in Table 3.1 have to be re-arranged first operation by operation and second by order of arrival time (or departure time). For instance, a sample of operation ID A is shown in Table 3.2.

Table 3.2: Sample of operation ID A arrival times and departure times

Operation ID A	Arrival time	Departure time
Lot 02	04/07/2007 16:00	04/07/2007 17:00
Lot 14	05/07/2007 02:00	05/07/2007 03:10
Lot 07	05/07/2007 03:30	05/07/2007 04:20

Proceeding this way, makes easier the calculation of the Inter-Arrival Time (I-AT) and Inter-Departure Time (I-DT) based on convenient calculations in Excel. Operation ID A inter-arrival time is the time between lots (for instance, Lot 14 arrival time – Lot 02 arrival time). It is the same way for inter-departure time, with lots classified in order of

the departure time. Samples of operation ID A inter-arrival time and inter-departure time are shown in Table 3.3.

Table 3.3: Sample of operation inter-arrival and inter-departure times

Operation ID A	Arrival time	Departure time	I-AT (hour)	I-DT (hour)
Lot 02	04/07/2007 16:00	04/07/2007 17:00		
Lot 14	05/07/2007 02:00	05/07/2007 03:10	10	10.17
Lot 07	05/07/2007 03:30	05/07/2007 04:20	1.5	1.17

To obtain the coefficient of variation of the operation inter-departure time, the average inter-departure time and inter-arrival time have to be calculated first. This is easy to obtain from any calculator. Variability is then measured by the coefficient of variation (Appendix A.2). An example of calculation is given in Table 3.4.

Table 3.4: Example calculation of the coefficient of variation

Operation ID A	I-AT (hour)	I-DT (hour)	Mean I-AT (hour)	Mean I-DT (hour)	Stdev $_{I-AT}$	Stdev $_{I-DT}$	CV_{I-AT}	CV_{I-DT}
Lot 02			5.75	5.67	6.01	6.36	1.05	1.12
Lot 14	10	10.17						
Lot 07	1.5	1.17						

Above, it just one operation sample. The same approach can be applied to all operations. Actually, it can also be applied to determine the inter-arrival and inter-departure time coefficient of variation for each buffer. Spearman [11], classifies what is an acceptable range for the variability. Big (Not recommended) variability is higher than 1.33, moderate variability (not recommended but acceptable) is the value between 0.75 ~ 1.33 and small variability (recommended) value is lower than 0.75.

The coefficient of variation is an effective measure of inter-arrival and inter-departure time variability. Nevertheless, caution is necessary when using the coefficient of

variation to measure the cycle time variability of the whole production line. In an operation management point of view, the coefficient of variation may sometimes be misleading. Indeed, the key point is to determine when all the items will be ready to schedule their delivery to the customers. Therefore, a key measure is the standard deviation. Items with a CT of 7 days ± 1 day and items with a CT of 70 days ± 10 days have the same cycle time coefficient of variation. However, 10 days standard deviation is much more troublesome for scheduling the delivery and managing space in the warehouse than 1 day standard deviation. Therefore, while the coefficient of variation still provides valuable information, the standard deviation must not be neglected.

3.2.2 First Metric: Single Coefficient of Variation

Overall, by using the coefficient of variation, operations with a high variability can easily be found. However, the variability might not be coming from the operation process but from the lots themselves. For example, lots are arriving late due to defect problems. Variability may originate from the errors of previous operations. Indeed, in a serial line where departures from an operation i become arrivals to the operation i+1, $CV_{I\text{-}DT}$ of operation i is the same as $CV_{I\text{-}AT}$ of operation i+1 [11].

$$CV_{I-AT(i+1)} = CV_{I-DT(i)} \qquad \text{Equation 3.1 [11]}$$

Therefore, variability in departures from an operation is the result of both variability in arrivals to the operation and variability in the process times. The relative contribution of these two factors depends on the utilization (u) of the workstation [11].

If utilization is close to one, then the operation is almost always busy. Therefore, under these conditions, the inter-departure times from the station will be essentially identical

to the process times. Thus, $CV_{I\text{-}DT}$ is expected to be the same as the process time coefficient of variation (CV_{PT}).

At the other extreme, when utilization is close to zero, the operation is very lightly loaded. Virtually every time a job is finished, the operation has to wait a long time for another arrival to work on. Because process time is a small fraction of the time between departures, inter-departure times will be almost identical to inter-arrival times. Thus, under these conditions $CV_{I\text{-}AT}$ and $CV_{I\text{-}DT}$ are expected to be the same.

Hopp [11] provides a simple method for interpolating between these two extremes as follows:

$$CV_{I-DT(i)}^2 = u_i^2 CV_{PT(i)}^2 + (1 - u_i^2) CV_{I-AT(i)}^2 \qquad \text{Equation 3.2 [11]}$$

$$CV_{I-DT(i)}^2 = u_i^2 CV_{PT(i)}^2 + (1 - u_i^2) CV_{I-DT(i-1)}^2 \qquad \text{Equation 3.3}$$

A simple illustration of these formulas is given in APPENDIX - E. These formulas don't apply to every process. However, they offer valuable insight into more complex and real systems. For formulas taking into account parallel machines, machines failures or any other production line characteristics see Gross [16] and Papadopoulos [17].

In other words, by monitoring $CV_{I\text{-}DT}$ operations with high or low output variability are identified. However, the reasons why the variability is high or low are not known. Where is the variability coming from, and what caused it? Is the departure variability ($CV_{I-DT(i)}^2$) high because of the variability in the process time ($CV_{PT(i)}^2$) or is it because of the variability in items arrival ($CV_{I-AT(i)}^2$)? Let's take the example of an operation whose job is just to delay the lots by 1 hour. The lots arrive at the operation, wait exactly

one hour and then go to the next operation. For this operation, $CV_{I\text{-}DT}$ and $CV_{I\text{-}AT}$ (Coefficient of Variation Inter-Arrival Time) are exactly the same. If $CV_{I\text{-}AT}$ is very high, then $CV_{I\text{-}DT}$ will also be very high. If only $CV_{I\text{-}DT}$ is considered then this operation appear to have high output variability. It might be wrongly assumed that this operation disturbs the production flow. However, this operation is not creating the variability. It has just processed what it received from the previous operation. For instance, if this operation receives a very low $CV_{I\text{-}AT}$, then $CV_{I\text{-}DT}$ will also be very low. This operation just keeps the variability at the same level than the variability it receives; it has no impact on variability.

Therefore, looking at $CV_{I\text{-}DT}$ is not sufficient to determine the origin of a high variability. The variability can either originate from this operation process or from the variability of arrivals ($CV_{I\text{-}AT}$). Another metric needs to be found.

3.2.3 Second Metric: Difference Metric

From the previous point, it is clear that a way to determine if a high variability is due to the operation at hand or due to the operation before need to be found. For that, the variability received by the operation and the variability sent by the operation need to be compared. Let's define the various behaviors possible for an operation.

Bad operation behavior: received low variability but sent high variability. This operation creates variability in the line.

Good operation behavior: received high variability but sent low variability. This operation removes variability in the line.

Neutral operation behavior: sent the same variability as it received. It does not impact variability in the line.

So how are good or bad operation distinguished? The (CV_{I-DT}) - (CV_{I-AT}) difference metric will provide the answer. If the result is positive then it means this operation is bad (creates variability). If the result is negative then it means this operation is good (removes variability). If the result is almost zero then it means this operation is neutral. To obtain the CV_{I-DT} and CV_{I-AT}, the calculations follow the steps mentioned previously (Section 3.2.1).

An example of data below:

Table 3.5: Creating and removing variability example

Operation	CV_{I-AT}	CV_{I-DT}	CV_{I-DT} - CV_{I-AT}	Operations create variability	Neutral operation	Operations remove variability
A	2.520	2.520	0		A	
B	2.508	2.508	0		B	
C	1.804	1.677	-0.126			C
D	1.763	1.933	0.170	D		
E	1.875	1.875	0		E	
F	2.325	0.993	-1.332			F
G	2.751	1.705	-1.046			G
H	2.045	2.561	0.516	H		
I	1.678	2.043	0.365	I		

In Table 3.5, the CV_{I-AT} column is the operation' inter-arrival time coefficient of variation. The inter-arrival time variability is higher than 1.6 for all operations. The CV_{I-DT} column is the operation' inter-departure time coefficient of variation. For most operations the inter-departure time variability is very high except operation F. The $CV_{I-DT} - CV_{I-AT}$ column is the difference between CV_{I-DT} and CV_{I-AT}, which is the metric to distinguish between good (operation removes variability) and bad operations (Operation

creates variability). The last three columns are the conclusion on the status of each operation: good, bad or neutral.

Review again Table 3.5, there are three operations that do not create variability despite their high $CV_{I\text{-}AT}$ and $CV_{I\text{-}DT}$. Even though they are neutral, they still have got very high variability. These operations are not capable to remove the bad effects from previous operations.

Operation F is critical; it has very high $CV_{I\text{-}AT}$ (2.325). It might be wrongly interpreted as being the worst operation if the coefficient of variation was the only statistic analyzed. But in reality it is the best operation. It is the one removing the most variability (-1.332) from the line as shown by the difference metric, because the output variability is very low compared to the input variability. That means the operation itself has the best performance no matter how it received the bad variability. Another example, operation G is showing the same behavior despite its high $CV_{I\text{-}AT}$.

There are three bad operations (see fifth column in Table 3.5). Their $CV_{I\text{-}AT}$ is low but $CV_{I\text{-}DT}$ is high, which means that their output variability is higher than the input variability they receive. They are creating variability in the line. It demonstrates a poor performance from those operations.

This simple metric shows clearly how variability transfers between operations. Next, the relationship between operations and their interactions need to be determined.

3.2.4 Third Metric: Ratio Metric

Another metric to consider is the ratio of inter-departure and inter-arrival time variability. With the difference metric, it is easy to compare operations and determine

which ones create or remove more variability. With the ratio metric, it is easy to see the percentage of variability increase or decrease induced by one operation, but it is not easy to compare the impact of operations on flow variability. Table 3.6 shows the difference between both metrics. The ratio metric shows that Operation J and Operation K both increase variability by 9% but it does not show which operation creates more variability. Whereas the difference metric clearly shows that Operation K creates two times more variability than Operation J.

Table 3.6: The comparison between difference and ratio calculation

Operation	$CV_{I\text{-}DT}$	$CV_{I\text{-}AT}$	$(CV_{I\text{-}DT}) - (CV_{I\text{-}AT})$	$(CV_{I\text{-}DT}) / (CV_{I\text{-}AT})$
J	0.96	0.88	0.08	1.09
K	2.03	1.87	0.16	1.09

3.3 Characterization of Lot Arrival and Lot Departure

Lot departure (or lot arrival) is characterized by the inter-departure time (inter-arrival time) and all the statistics associated ─ mean, coefficient of variation and distribution.

An important issue is that one week of data is not sufficient to use these statistics. There are not enough data points. Results will not be representative. Thus, a new set of data for a 6-month period has been taken to calculate the new results. In the semiconductor fab, there are more than 300 operations. It is too complex to analyze the whole production line. Hence, from these 6 months of data, 5 consecutive operations were selected for observation. As can be seen in Figure 3.2, the main serial line composed of operations A, B, C, D and E was considered. Cross items' entrances from operations F, G, H, I, J, K and L were not taken into account.

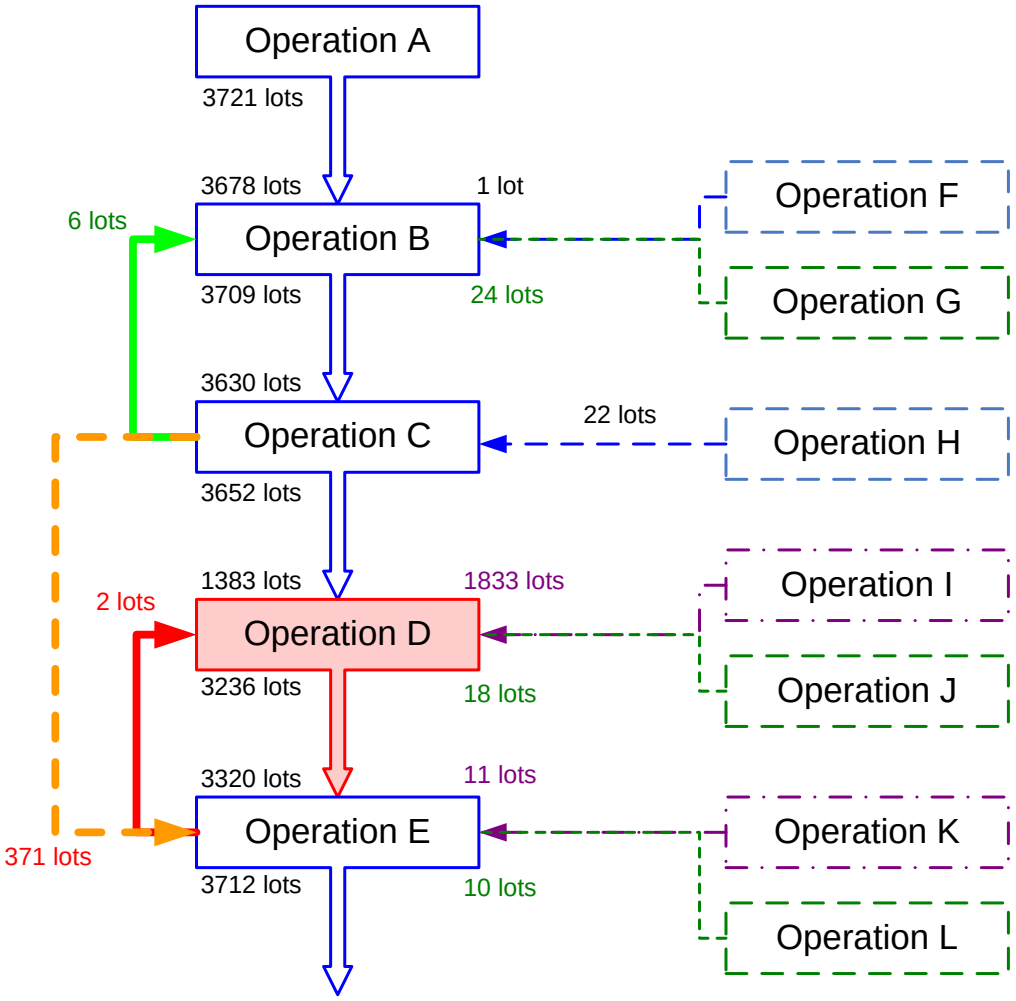


Figure 3.2: Five operations serial line

3.3.1 Mean Inter-Arrival Time

To calculate the mean inter-arrival time, a period over which the data are averaged need to be decided. What period of time provides a better understanding for the analysis? Is one week period, a good fit to measure the average inter-arrival time? Or are 12 hours period, 24 hours period, or 48 hours periods more adequate? In this analysis, the average inter-arrival times for each period and each operation were calculated (calculation refers to section 3.2.1). Then the results were compared to determine which period optimizes the data analysis.

In the 12 hours period, there are 362 times 12 hours in 6 months data (Figure 3.3 displays the first 3 months). In the 24 hours period, there are 181 times 24 hours in 6 months data (Figure 3.4). In the 48 hours period, there are 90 times full 48 hours in 6 months data. In the week period (Figure 3.5), there are 25 full weeks in 6 months data (Figure 3.6).

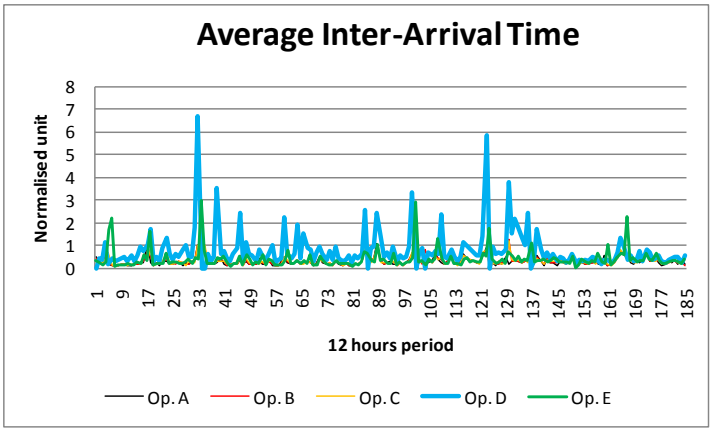


Figure 3.3: Average inter-arrival time of 12 hours period

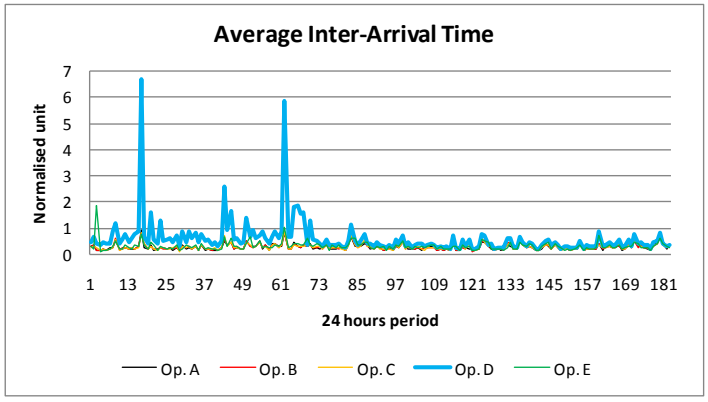


Figure 3.4: Average inter-arrival time of 24 hours period

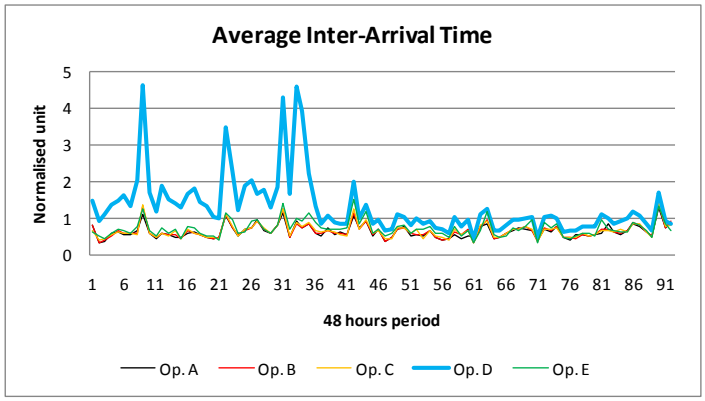


Figure 3.5: Average inter-arrival time of 48 hours period

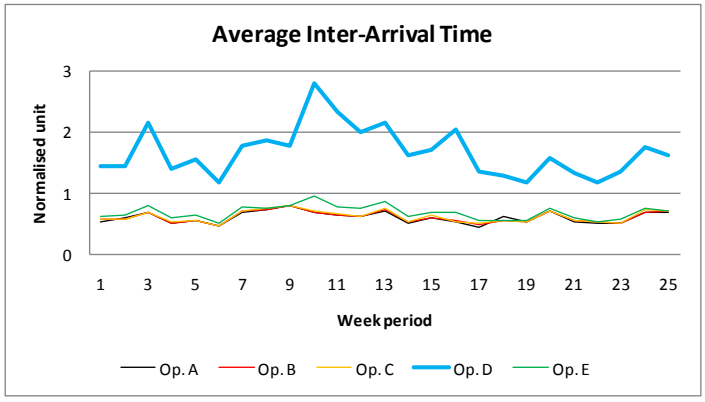


Figure 3.6: Average inter-arrival time of week period

The four graphs above respectively show the results of 12 hours, 24 hours, 48 hours and week period of average inter-arrival time. Average inter-departure time graphs (Appendix C.1) presents similar results to average inter-arrival time graphs. Data in the 24 hour period graph might be slightly too noisy but in 12 hour period graph, they are even more noisy. Another issue with the 12 hours period is that very few lots arrive in each period. There are even no lots arriving at all in some few periods. In those conditions, average values loose much meaning. Data in the week period graph are not sensitive enough and the points of interest cannot be identified. Data in the 48 hours

period graph are slightly less noisy than data in the 24 hour period graph. Therefore 48 hour-period was selected to perform the analysis.

3.3.2 Correlation Coefficient

The difference metric illustrated that variability could be transferred from one operation to the next. Therefore a further interesting point to study is the degree of correlation between the operations. For example, Hendricks [12] showed that negative correlation indicates that less storage is required to buffer the output of one subsystem from the next manufacturing subsystem than if the correlation structure were positive or zero. Additionally, information from the correlation structure can be used to generate predictors for inter-departure times.

48 hours period is a proper period to measure the average inter-arrival time, but what about the correlation coefficient? Next, the same procedure of analyze is applied for the correlation coefficient metric (Appendix A.3.4). Operation A is taken as example. The proper period to measure the correlation coefficient between average inter-departure and inter-arrival time is determined.

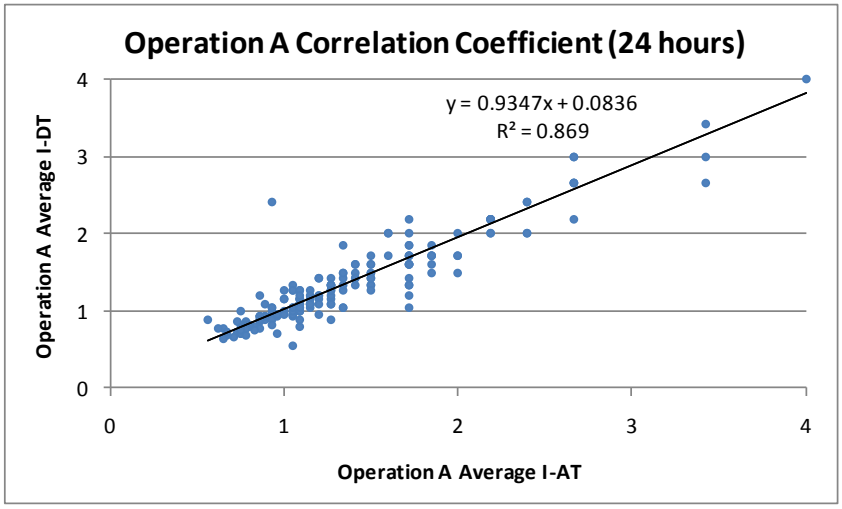


Figure 3.7: Operation A correlation coefficient between inter-arrival time and inter-departure time at 24 hours period

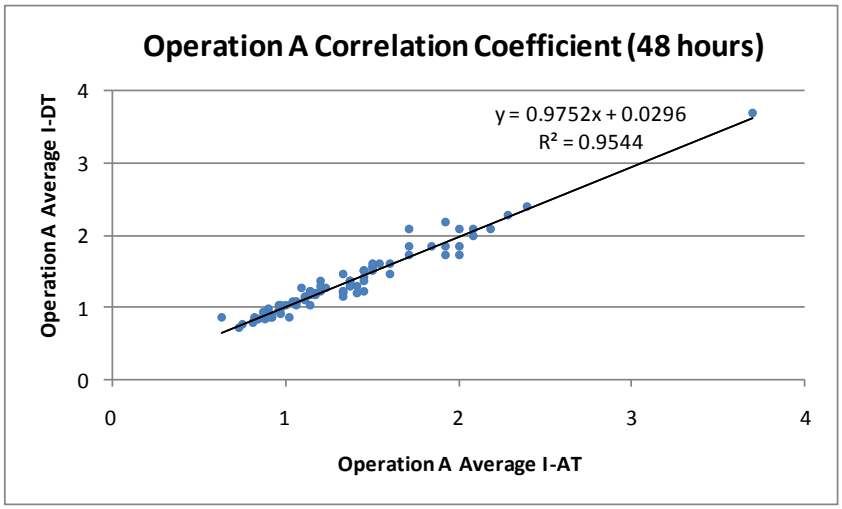


Figure 3.8: Operation A correlation coefficient between inter-arrival time and inter-departure time at 48 hours period

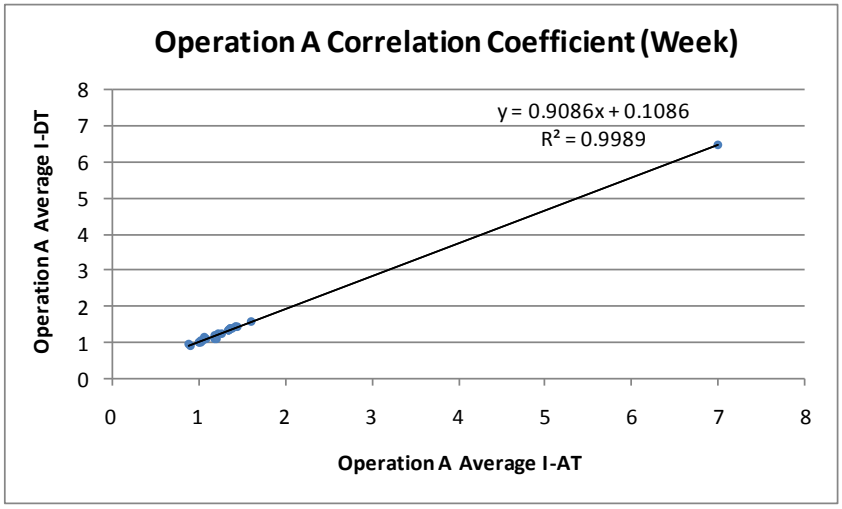


Figure 3.9: Operation A correlation coefficient between inter-arrival time and inter-departure time at week period

The three previous figures are the correlation coefficient of inter-departure and inter-arrival time in operation A. There again the data from the 48 hours period look most promising. There aren't enough data points on the week period to be certain of the regression results. 24 hours period, is noisy as illustrated by the lower correlation between departures and arrivals. The same comments can be made about operations B, C, D and E (Appendix C.2, C.3, C.4 and C.5). Therefore, a 48 hours period will be applied to calculate the correlation coefficient for all operations (Figure 3.10 to Figure 3.14).

All operations are showing high correlation coefficient, except operation D (Figure 3.13).

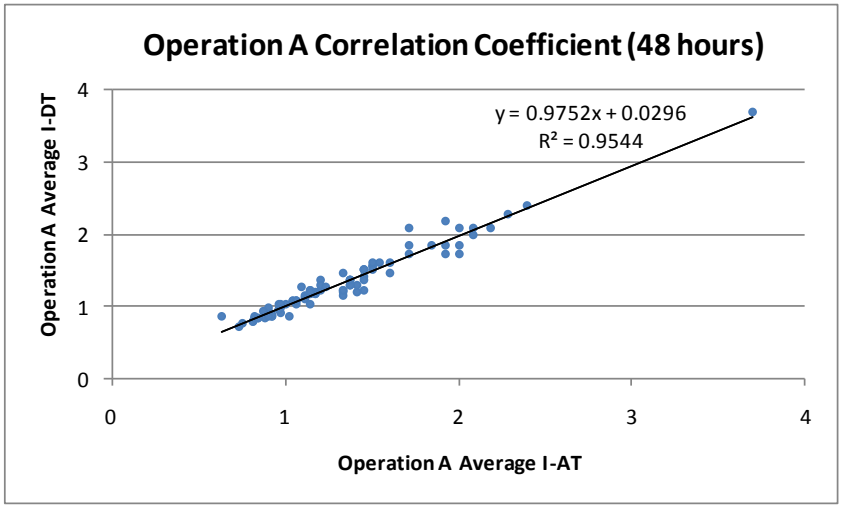


Figure 3.10: Operation A correlation coefficient between average inter-arrival time and inter-departure time at 48 hours period.

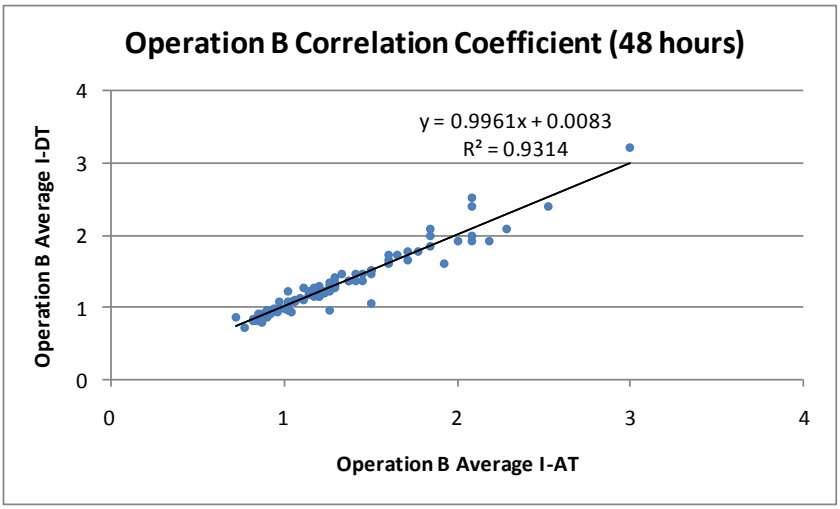


Figure 3.11: Operation B correlation coefficient between average inter-arrival time and inter-departure time at 48 hours period.

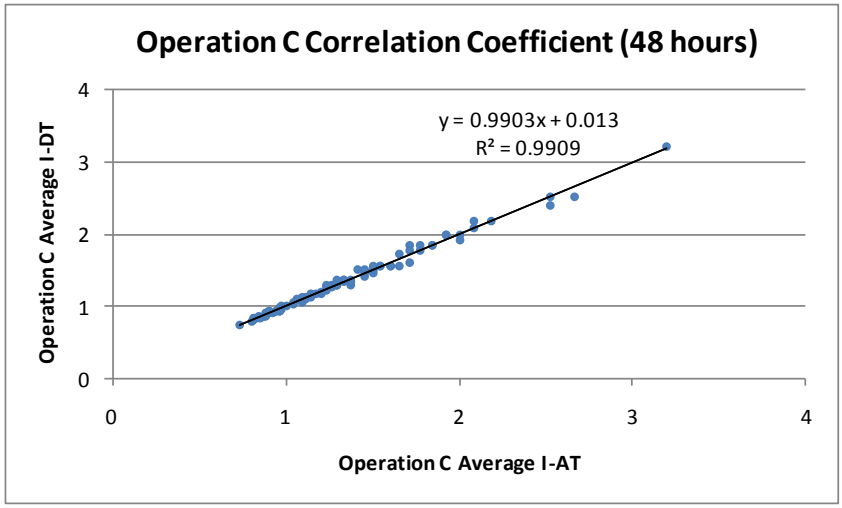


Figure 3.12: Operation C correlation coefficient between average inter-arrival time and inter-departure time at 48 hours period.

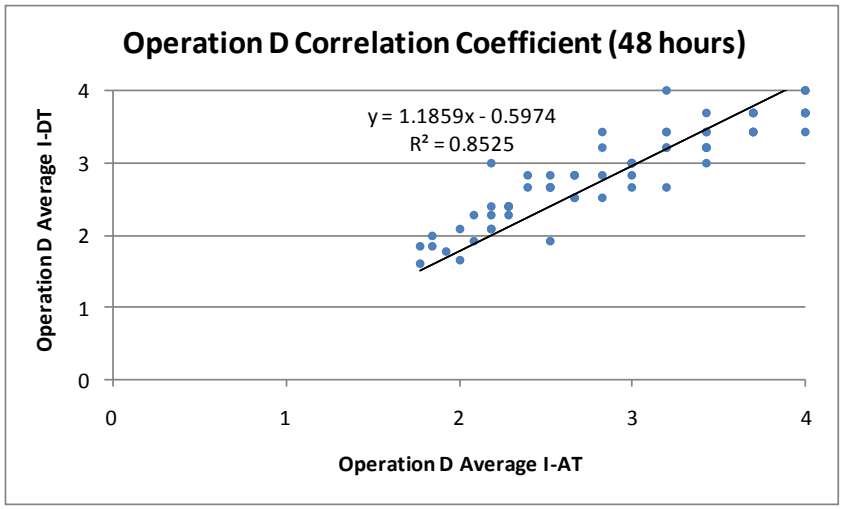


Figure 3.13: Operation D correlation coefficient between average inter-arrival time and inter-departure time at 48 hours period.

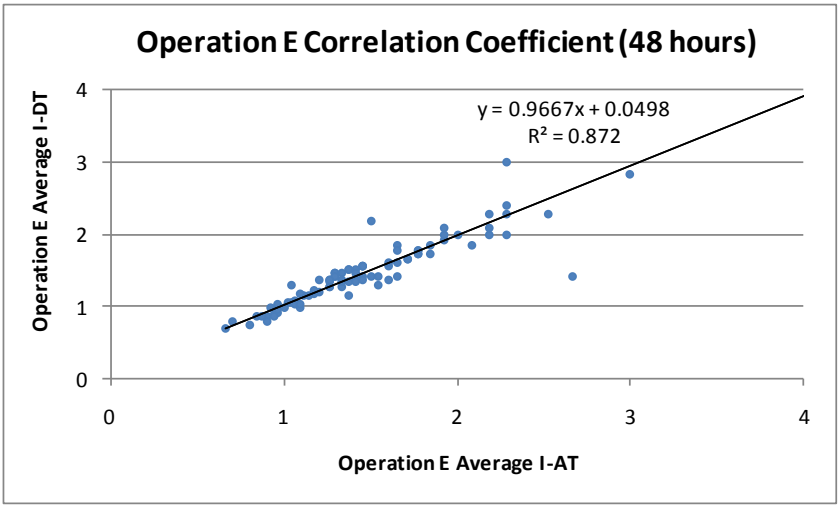


Figure 3.14: Operation E correlation coefficient between average inter-arrival time and inter-departure time at 48 hours period.

As can be seen, compared to other operations, in operation D the dots are more spread out in the graph, and showing a weaker correlation (Appendix A.3) between average inter-arrival time and inter-departure time. To find an explanation, a return to the graph of the 5 operations main structure (Figure 3.2) is necessary. All other operations have a single source of input items (single line). But operation D gets 50% of its input items from a re-entrant line. This could explain the reduced correlation coefficient as items from several operations are crossing each other.

Overall, the correlation between inter-arrival and inter-departure is a good indicator of sources of variability in the production line as illustrated by the corruptive effect of the re-entrant line.

3.3.3 Inter-arrival time distribution

The inter-arrival time distribution can be modeled using random distributions. The issue is to determine which statistical distribution is suitable for the data. Extend [94] includes Stat::Fit, a distribution fitting package from Geer Mountain Software

(www.geerms.com) that help analysts determine which statistical distributions, if any, offer a good fit for underlying random processes.

In Stat::Fit, the fits and validity tests are totally inaccurate for fewer than 10 data points and don't achieve good accuracy until 100 data points or so. On the order of 200 data points seems to be optimum. For large data sets, greater than 4000 data points, the validity tests can become inaccurate, occasionally rejecting a proposed distribution when it is actually a useful fit [94].

Each operation receives approximately 200 items each week. So one week of data is the optimum time unit to determine with Stat::Fit the inter-arrival time and inter-departure time distributions. For each week of the 6 months data, the best fits for both inter-departures time and inter-arrival time distributions were determined for the five operations. Thus 25 fits (there are 25 weeks in 6 months) for each operation's inter-departures time and inter-arrival time distributions were obtained (in total: 25 fits x 5 operations x 2 distributions = 250 fits). An example of operation A is given in Figure 3.15 and Table 3.7 for inter-arrival time. An example for all others operation inter-departure/arrival time is given in APPENDIX - C (Section C.6). An exponential distribution was in most cases (>80%) the best fit and it was an acceptable fit for all cases.

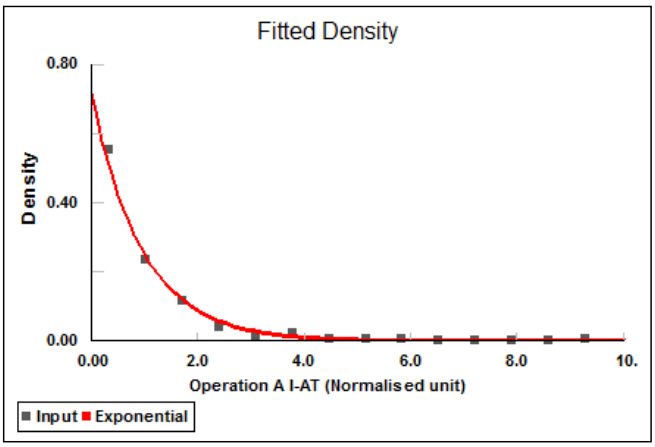


Figure 3.15: Exponential fit of the inter-arrival time distribution (Week based).

Table 3.7: Exponential fitted data of the inter-arrival time distribution (Week based)

Exponential			
Minimum = 0. [fixed]		Beta = 0.954887	
Kolmogorov-Smirnov		Anderson-Darling	
data points	179	data points	179
ks stat	5.38E-02	ad stat	0.581
alpha	5.00E-02	alpha	5.00E-02
ks stat (179,5.e-002)	0.101	as stat (5.e-002)	2.49
p-Value	0.654	p-Value	0.666
result	**Do Not Reject**	result	**Do Not Reject**

3.4 Queue time analysis

Items with a total cycle time higher than 40 hours through the whole five operations were identified. They represent 1.5% of the item's population in the data. For each of those items, the queue time at each operation was calculated. Therefore, the operation where each item spent the most time was identified. For each operation, the number of lots spending the longest time could be counted.

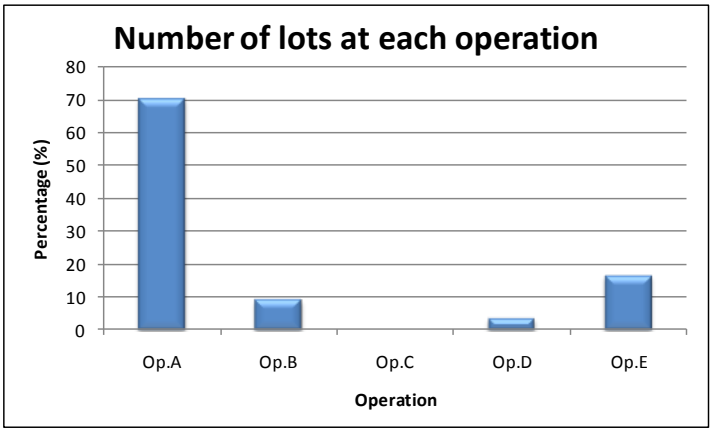


Figure 3.16: Number of lots having the longest queue at the operation

As can be seen in Figure 3.16, 70% of the items spent the longest time queuing in operation A. 9% of the items spent the longest time queuing in Operation B. No items spent the longest time queuing in operation C. 4% of the items spent the longest time queuing in Operation D and finally 17 % of the items spent the longest time queuing in operation E.

Let's explain why Operation A generates so many high cycle time lots by considering the relationship between tool availability, average Process Time (PT), Queue Time (QT) plus Transport Time (TT).

As can be seen from the highlighted area on Figure 3.17, availability seems to impact queuing. In the period from period 112 to 120 (Figure 3.17) queue time shows a sharp peak. In this situation, one would expect utilization to increase reasonably high as well to compensate. But that cannot be seen in the Figure 3.18. The maximum of the utilization is 0.5. So why does queuing not impact utilization? Furthermore, why does availability not always have the same impact?

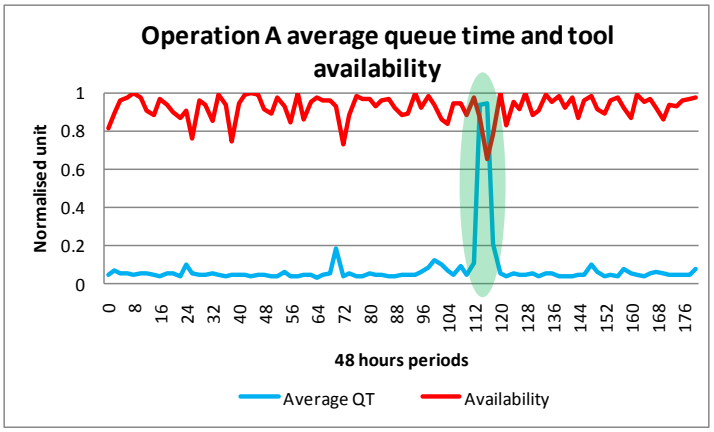


Figure 3.17: Operation A tool availability analysis

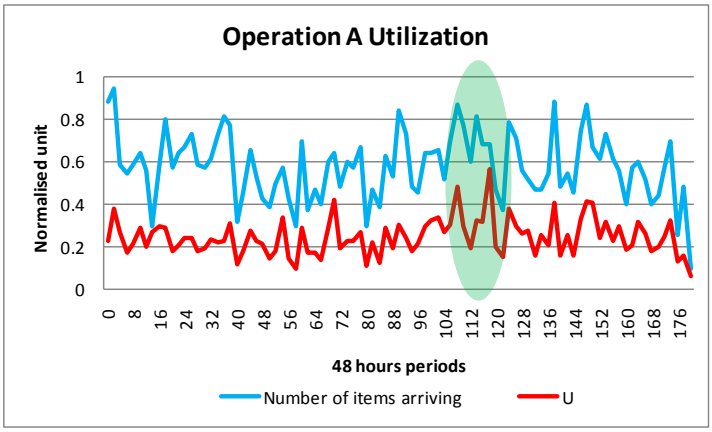


Figure 3.18: Operation A utilization

The data have been reviewed again and particularly, the period between period 112 and 120. There was no downtime or high utilization involved. So, the high queue during this period was not caused by machines downtime or an arrival of lots exceeding the capacity of the operation.

3.5 Conclusion

A real production system was considered. Six months data were collected on one product route. Five operations were observed and their inter-departure times were monitored. Several metrics to measure variability and more importantly to determine the

origins of variability have been developed. When operations have a high variability, it could be caused by the previous operation transferring its own variability or it could be caused by the operation itself creating variability. Therefore, propagation of variability from one operation to the next has been illustrated and a metric, the coefficient of correlation, has been exposed to evaluate it. Using a 48 hours period as the base for the calculation was found to be the best compromise to evaluate the data. Overall, Inter-arrival time and Inter-departure time correlation is quite strong for each operation. Nevertheless some phenomena observed in the line remain unexplained. Some periods showed particularly high queuing, which could not be related to any downtime or utilization peak.

One issue is the lack of accuracy of the lot monitoring. For example, lot departure corresponds to the time the operator logs the departure in the monitoring system. Actually this lot might have been completed several hours ago, but no operator was available to unload it from the operation and log its completion. No explanations are recorded in the system for the delays seen on the lots. Therefore it is hard to follow the progress of the lots through the line. Lack of information impedes learning from real system data and limits the generalization of findings. Another solution has to be found to understand the behavior of the production line. In particular, each parameter needs to be isolated, their influence analyzed and optimum configurations devised.

This can only be achieved in a controlled environment, in other words using simulations. Simple simulation models are needed to gain insights and then conduct further analyses on more complex model.

CHAPTER - 4 METHODOLOGY

4.1 Introduction

The concept of using a simulation model was based on the questions raised from the real data analysis in Chapter - 3. Various approaches can be used to study a production line: experimental, theoretical modeling using the queuing theory (APPENDIX - B) and simulation modeling (APPENDIX - D). It has been shown in the pre-study that an experimental approach in a semiconductor production line is too complex to isolate the influence of all the parameters.

Theoretical modeling provides insights and understandings of production line behaviors; nevertheless, it is insufficient to predict with accuracy the response to modifications brought to the line (APPENDIX - B). Therefore, this study was undertaken using simulation modeling.

Simulation is a sound approach to gain insights of the dynamics of complex systems without costly physical experiments. The advantages of simulation include: identification; incorporation or elimination of system parameters; fast experiments and what-if analysis; low cost and low risk [13]. The simulation models considered in this research are discrete, dynamic, and stochastic and will henceforth be called discrete-event simulation model.

- A discrete system is one for which the state variables change instantaneously at separated point in time [95].

- Dynamic simulation models: A dynamic simulation model represents a system as it evolves over time [95].
- Stochastic simulation models: If a simulation model contain probabilistic (i.e., random) components, it is called stochastic. Stochastic simulation models produce output that is itself random, and must therefore be treated as only an estimate of the true characteristics of the model [95].

Discrete-event simulation concerns the modelling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time. These points in time are the ones at which an event occurs, where an event is defined as an instantaneous occurrence that may change the state of the system.

A process approach to simulation modelling was adopted. It views the simulation in terms of the individual entities involved and describes the "experience" of a "typical" entity as it "flows" through the system [95]. It requires the use of special-purpose simulation software (Extend). A production line was modeled using Extend^{TM} v6 simulation software (APPENDIX - D).

4.2 Real Production Line Setup

Before modeling and running simulations, an understanding of a real semiconductor production line is needed. What are the different types of operations? How are they processing the items? What are the processing times and typical availabilities? How are the items moving from one operation to the next? All these questions need answers in order to select the proper blocks in Extend and also to set all their parameters to realistic

values. A real production pattern of semiconductor manufacturing is shown in Figure 4.1. There are six main production areas.

- Diffusion for oxidation, diffusion, deposition, anneals and alloy
- Photolithography for deposition of patterned photoresist layers
- Etching for layer removal
- Ion Implant for ions implantation
- Thin films for layer deposition
- Polish

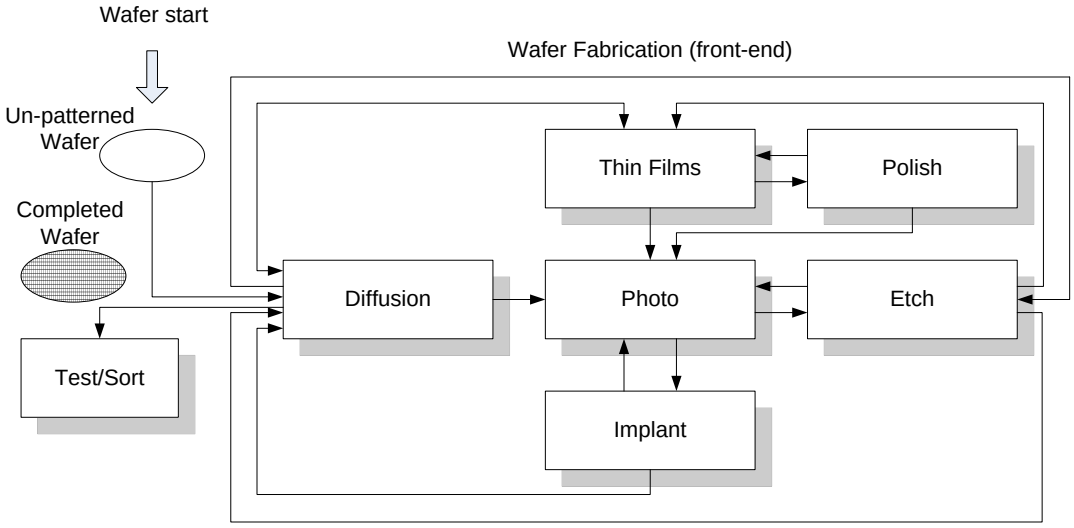


Figure 4.1: Six major production areas in the semiconductor manufacturing fabrication [96]

Figure 4.1 is a standard picture of the process route; it may changes slightly based on the product type. Otherwise, all items have to go through all 6 production areas in several occasions. For each step of the process, the items will go in a specific sequence through some of the production areas to complete the step. There is no repetitive sequence of events or cycle that can be identified and simulated. The diffusion area batches several items together to process them. The photo area is typically the bottleneck of the line.

Items go through it more often than in the other areas. Therefore photo area has many more wafers to process and this limits the capacity of the whole line.

In summary, a real production line is composed of six operations. One of them is a batching operation (Diffusion area) and another one is a bottleneck (Photolithography area). The items' movement sequence is complex with multiple re-entrant lines.

4.3 Model Development

Simulation model of a theoretical production line has been constructed using the Extend 6^{TM} simulation software. All the items are initially assumed to go through all the areas in sequential order without re-entrant lines. This is to allow a detailed analysis of tool availability and batching, free from re-entrant lines influences. Then, the results obtained are tested in a re-entrant environment and interesting areas for policy, practice and future research for the academic community are highlighted.

Thus the simulation model was built with six successive operations (serial line), each representing one of the six areas found in the real environment. One operation is a batch operation to simulate the diffusion area and one operation is a constraint operation to simulate the lithography area. Three variations of the model were developed to introduce progressively these various operations. First the constraint operation, then batch processing and finally tool availability were simulated as may be seen in Figure 4.2, Figure 4.3 and Figure 4.4.

Several simulation experiments are conducted to assess the impact of item release strategies and production load on the performance of the line with respect to queuing time and cycle time. Literature mentioned that the loading should be limited to slightly

less than the constraint capacity [10] to avoid "blowing up" the system. Therefore, the maximum loading will not exceed the capacity of constraint operation.

4.3.1 Reference Model: Single Item Processing Model (SIPM)

The first model (Figure 4.2) is similar to Murray's original model [97]. It is used as reference model to analyze the characteristics of the two following models. A buffer is positioned in front of each operation to allow items to queue when all the machines are busy. The queue follows the First-in-First-out rule (Section 2.7).

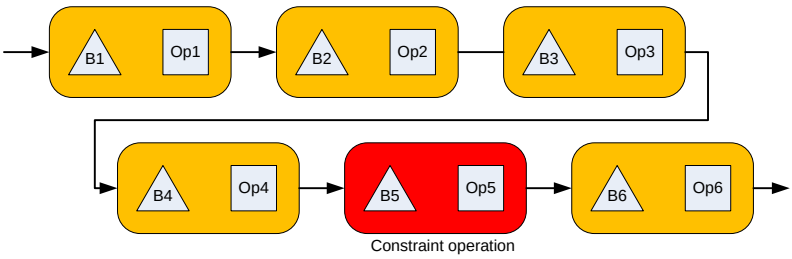


Figure 4.2: Single Item Processing Model (SIPM)

The default operation's capacity is set to 420 items/week (factory target). It is assumed that all operations have the same capacity, except the constraint. This assumption isolates the influence of capacity to the constraint operation only and facilitates the results' analysis. The capacity of the constraint operation is set at two third of the others (280 items/week). By choice, the constraint was placed near the end of the line (Operation 5). Thus in the following models, it will be submitted to the disturbance generated by batching and availability to the flow of items.

Each operation contains several identical machines in parallel as previously described by Murray [97]. The number of machines is given in Table 4.1. The processing time of each operation is assumed to be normally distributed. The mean processing time is

calculated from the definition of capacity given in Equation 4.1 for one week (10080 minutes) in order to control the capacity. As there are a different number of machines at each operation, each one has a different mean processing time (see Table 4.1) to obtain the targeted capacity. For Operation 2 refer to Op2 SIPM column.

$$Capacity = \frac{10080\,\text{min} \times m}{PT} \qquad \textbf{Equation 4.1}$$

m : Number of machines
PT : Processing Time (Normal Distribution)

Table 4.1: Simulation models set up for each operation

Operation setup		Op1	Op2 SIPM	Op2 BPM	Op3	Op4	Op5	Op6
Number of machines (m)		8	15	3	13	11	8	14
Normal Distribution	Mean processing time (min)	192	360	360	312	264	288	336
	Processing time standard deviation	9.33	10.8	10.8	21.4	18.1	14.1	16.4
Capacity (items)		420	420	420	420	420	280	420
Batch Size (items)		1	1	5	1	1	1	1

Op2 SIPM: setup of Operation 2 in single item processing model

Op2 BPM: setup of Operation 2 in batch processing model

4.3.2 Batch Processing Model (BPM)

The second simulation model introduces batching process in Operation 2 (Figure 4.3). This is to take into account what is occurring in the diffusion area. The batch was placed at the beginning of the line in order to affect the whole line and break the regularity of the flow. This provides simulation conditions closer to what can be seen in the real environment. And the impact of batching on the constraint operation can be studied. The

batch size is assumed to be five items. Items are processed simultaneously. If a batch is not complete, items have to wait until the next arrival and completion of the batch to be processed Figure 4.3 provides a flow diagram for the batch processing model. This batch processing is a parallel process; therefore, the five items are processed at the same time, then un-batched when the process is completed. The set-up of Operation 2 batch processing (Op2 BPM column) is shown in Table 4.1.

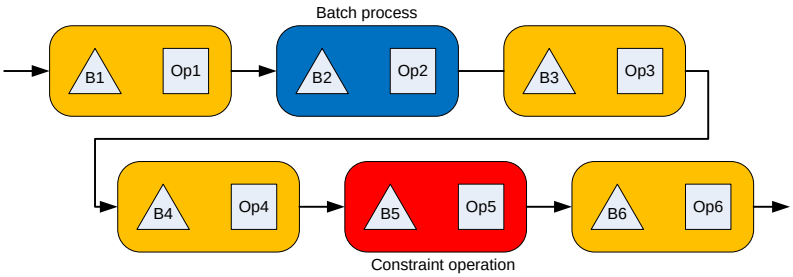


Figure 4.3: Batch Processing Model (BPM)

4.3.3 Downtime Simulation Model (DSM)

The third simulation model (Figure 4.4) introduces tool availability process in Operation 3, it is affected by downtime. Operation 2 is still a batch process. Items are batched by 5 and processed simultaneously. If a batch is not complete, items have to wait until the next arrival to be processed. Operation 5 is still the constraint; it has the lowest capacity at 280 items per week. The rest of the operations have the same 420 items per week capacity. Processing time, batch size, and capacity for the basic setup are shown in Table 4.1 (Operation 2 setup is identical to Op2 BPM column).

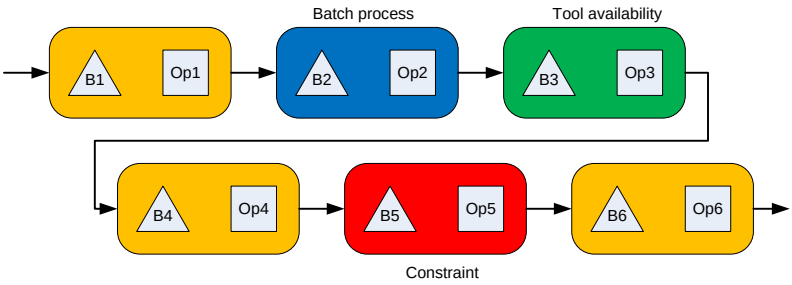


Figure 4.4: Tool availability model

4.4 Data Collection

The simulation model collects, in a database, all the information relative to the flow of each item in the line. That is, for the buffers, the arrival time, queue time and departure time of each item, and for the operations, arrival time, processing time, and departure time. In the final development of the simulation models, the SDI library of Extend was used. It includes a comprehensive data management system. This improved the performance of the models, lowering the total execution time from twenty minutes (fifteen minutes runtime and five minutes data processing) to approximately three minutes (fifteen to fifty seconds runtime and approximately two minutes data processing).

4.5 Data Analysis

From this database, the statistics (mean, standard deviation and coefficient of variation) for cycle time, process time, queue time, utilization and inter-departure time were calculated. Standard definitions of these key performance measures are given in Appendix A.4.

A simulation is a computer-based statistical sampling experiment. Thus, if the results of a simulation study are to have any meaning, appropriate statistical techniques must be used to design and analyze the simulation experiments [95].

Since characteristics of the model are assumed not to change over time, the simulations run in this research belong to the non-terminating category. A non-terminating simulation is one for which there is no natural event to specify the length of a run [95]. A measure of performance for such simulation is said to be a steady-state parameter, for example the steady-state mean $\nu = E(Y)$ of some output stochastic process Y_1, Y_2... One difficulty in estimating a steady-state parameter is that the observations Y_1, $Y_2 \ldots$ Y_m are dependent on the initial conditions. This cause an estimator of the steady-state based on those observations not to be "representative". For example, the sample mean $\bar{Y}(m)$ will be a biased estimator of $\nu = E(Y)$ for all finite values of m. This problem is called the problem of the initial transient or the startup problem in the simulation literature.

The technique most often suggested for dealing with this problem is called warming up the model or initial-data deletion [95]. The idea is to delete some number of observations from the beginning of a run and to use only the remaining observations to estimate ν. For example, given the observations Y_1, $Y_2, \ldots,$ Y_m, it is often suggested [95] to use

$$\bar{Y}(m,l) = \frac{\sum_{i=l+1}^{m} Y_i}{m-l}$$

Equation 4.2 [95]

($1 \leq l \leq m - 1$) rather than $\bar{Y}(m)$ as an estimator of ν. In general, one would expect $\bar{Y}(m,l)$ to be less biased than $\bar{Y}(m)$, since the observations near the "beginning" of the simulation may not be very representative of steady-state behavior due to the choice of initial conditions.

If properly applied, this approach should give reasonable good statistical performance. Moreover, it applies to all type of output parameters and can be used to estimate several different parameters for the same simulation model. And finally, it can be used to compare different system configurations.

The Initial-Data Deletion method has been employed to conduct and analyze the results of the simulation model. The simulations simulated 18 months of production. The initial 3 months of data were ignored to eliminate the influence of the warm up period. Therefore 15 months of data were used for the analysis.

In order to compare the various system configurations investigated, common random numbers (CRN) were used. The basic idea is that the alternative configurations should be compared under similar experimental conditions. It increases the confidence that any observed differences in performance are due to differences in the system configurations rather than to fluctuations of the "experimental conditions" [95]. In this research, these experimental conditions are the generated random MTTR and MTBF of the machine's failures. The same basic random numbers are used to drive each of the alternative configurations through time.

4.6 Summary

This chapter explained the model formation, setup, shape and structure in order to follow next experiments. It also described how the data will be gathered and analyzed. Further information on the process will be obtained from the production simulations. Any unexpected or badly comprehended production behavior can be analyzed, and the understanding of the production flow can be improved. Next chapter, several scenarios will be introduced one by one. The results will be analyzed and their effects on the behavior of real factory production will be discussed.

CHAPTER - 5 SIMULATION RESULTS

5.1 Introduction

Previously (Section2.13), batch processing and tool availability were identified to be the main factors influencing production performance. Several simulation models (Chapter - 4) have been set to investigate the relationship between batch, constraint and tool availability (affected by downtime) operations. These models are studied in the following two sections. First, the effect of batch process operation is studied. It is then followed by the study of tool availability.

5.2 Simulation of the Effect of a Batch Process Operation on a Production Line with Constraint Operation

The basic relationship between queue time, utilization and inter-departure time has been studied, as well as their interaction with each other. They are inseparably linked together. If the behavior of any one is changing, then it will definitely affect the others. It was explained in Section 2.5 that batching is a significant source of variability due to irregular releases. It significantly affects the performance of a production line. As mentioned in the research objectives (Section 1.3.1), the objective of this simulation study is to determine the effect of batch processing on the production line and determine a compromise between loading, release policy and batching.

Therefore, four experiments are conducted to investigate the performance of the modeled line under various product loads, item release rates and batching policies in comparison to a single item processing line (without batch process).

5.2.1 Scenario 1: Fixed (High) Production Load, Variable Release Profile in SIPM and BPM Models

The purpose of this experiment is to examine the impact of release rate on performance of both models given a high production load (280 per weeks). The simulation models tested in this scenario are respectively single item and batch processing model (SIPM and BPM), their structure and setup were described in Section 4.3 (p102) and represented in Figure 4.2 (p103) and Figure 4.3 (p105). Performance measures were defined in Section 4.5 (p106). Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5, p106). Total simulation run time is 18 months.

Six different item release profiles were considered; once/week, twice/week, once/day, twice/day, four times/day, and five times/day with a fixed production load identical to the constraint operation capacity, 280 items/week, for both the single item processing model and the batch processing model. The production load was purposely chosen at a critical loading of the constraint, outside the safety zone, in order to amplify and highlight the effect of batching. The objective here is not to obtain a smooth and efficient production but it is to locate any possible corruptive effect of batching on the items flow. The details of the release profiles, for a loading of 280 items per week, are given in Table 5.1.

Table 5.1: Release profiles

Release Frequency	Period (min)	Number of items
Once/week	10080	280
Twice/week	5040	140
Once/day	1440	40
twice/day	720	20
Four times/day	360	10
Five times/day	288	8

Table 5.2 and Table 5.3 depict the results for mean cycle time and cycle time variability for single item processing and batch processing models. Both single item processing and batch processing models have similar mean cycle time and cycle time variability. The batch slightly increases the cycle time, but the main factor affecting the cycle time is the profile of the release. The results show that the more regular and frequent the item release is the more cycle time reduces. In other words, the highest the release variability, the highest the cycle time. Nevertheless, some saturation effect can be seen. A release rate more frequent than once/day only marginally improves the cycle time.

Table 5.2: Scenario 1 simulation results: Mean cycle time of SIPM and BPM

Mean CT (min)	SIPM	BPM
once/week	6331.8549	6363.7117
twice/week	4041.8023	4066.6012
once/day	2409.6941	2449.7855
twice/day	2106.2633	2133.5507
Four times/day	1954.3814	1983.0295
Five times/day	1896.5631	2026.9700

Table 5.3: Scenario 1 simulation results: Cycle time coefficient of variation of SIPM and BPM

CV_{CT}	SIPM	BPM
once/week	0.4201	0.4192
twice/week	0.3342	0.3282
once/day	0.1627	0.1573
twice/day	0.0939	0.0933
Four times/day	0.0617	0.0556
Five times/day	0.0532	0.0700

The cycle time includes the transport time, queue time and processing time. Transport time was not considered in the simulation. In fully automated transport system, variation in transport time is negligible according to factory managers. Table 5.4 depicts the results for mean queue times for the single item processing and the batch processing models. Again, the impact of the release variability is obvious. It can be seen that the variations seen in cycle times originate from the queue times. Processing time variations are negligible compare to queue time variations.

From Table 5.4, it can be seen that the introduction of batch processing model results in queuing in Buffer 2, as items must wait until five items are present to complete the batch. Buffer 3 is showing a small queue as operation 3 process the five items arriving simultaneously one at a time. These increases in buffer 2 and 3, explain the slight increase in cycle time of the batch processing model. Buffer 4 has no queues, because Operation 4 has the same capacity as Operation 3. Operation 5 has a long queue because it is the constraint operation. Nevertheless, a reduction of the queue time compared to single item processing model can be seen. Indeed, batching reduced the throughput rate of Operation 2 creating a small queue in Buffer 2. This reduces the arrival rate into

Operation 5, and thus shortens the average queue time in Buffer 5. The distribution of queuing time in the buffers has changed with little impact on the total queue time.

Note: from Table 5.2 and Table 5.3 above, mean cycle time and cycle time variability, the release policy of twice/day appears to be a good compromise. It generates low cycle time and low variability and simultaneously it keeps simple the release profile for operators. Therefore, twice/day will be used as standard release policy in the following simulations.

Table 5.4: Scenario 1 simulation result (from buffer 1 to buffer 3): Mean queue time of SIPM and BPM

Mean QT (min)	B1		B2		B3		B4		B5		B6	
Model	SIPM	BPM	SIPM	BPM	SIPM	BPM	SIPM	BPM	SIPM	BPM	SIPM	BPM
once/week	2859.7145	2860.2649	40.9533	89.7407	28.6638	50.2305	24.6328	23.0640	1626.7739	1589.3225	0	0
twice/week	1388.3013	1388.9227	35.2513	81.2804	22.0627	38.4424	16.4125	16.1228	828.6618	790.7536	0	0
once/day	336.3818	336.4728	16.4514	62.3027	16.3560	21.2444	6.7939	8.1025	282.5948	270.5625	0	0
twice/day	133.4223	133.5741	6.2247	41.9402	7.4026	17.4724	3.4225	5.2605	204.6525	184.2136	0	0
four times/day	33.7934	33.7731	0	56.9647	0	0	0	0	169.4648	141.1835	0	0
five times/day	0	0	3.9913	76.5425	0.2528	1.7327	0	0	141.1930	197.5823	0	0

5.2.2 Scenario 2: Fixed Release Profile, Variable Production Load in SIPM and BPM Models

The purpose of this experiment is to examine the impact of production load on the performance of the models given a specific release profile (twice/day). Five different production loads were explored; 280 items/week, 240 items/week, 200 items/week, 160 items/week, and 120 items/week.

The simulation models tested in this scenario are respectively single item (SIPM) and batch processing model (BPM), their structure and setup were described in Section 4.3 (p102) and represented in Figure 4.2 (p103) and Figure 4.3 (p105). Performance measures were defined in Section 4.5. Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5). Total simulation run time is 18 months.

Figure 5.1 shows that the mean cycle time has been increased slightly by introducing batch processing. As production load increases, Coefficient of variation cycle time (Figure 5.2) increases for the single item processing model and decreases for the batch processing model. The two models show similar performances at 280 items/week. Once again, all these observations on cycle time can be explained by looking at the items queues in each operation (Figure 5.3 and Figure 5.4).

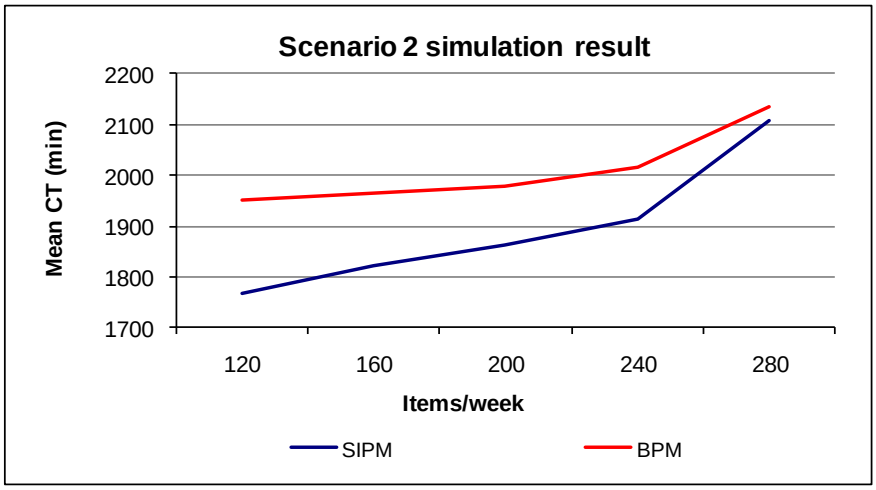


Figure 5.1: Scenario 2 simulation results: Mean cycle time of SIPM and BPM

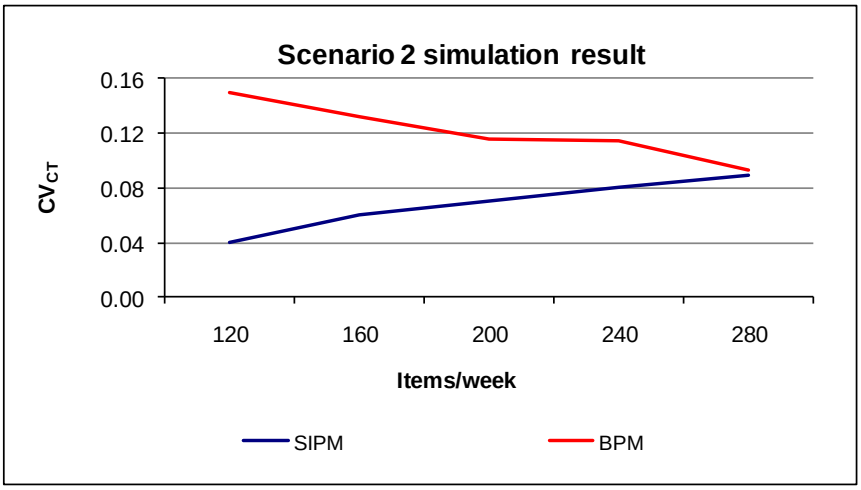


Figure 5.2: Scenario 2 simulation results: Coefficient of variation cycle time of SIPM and BPM

Figure 5.3 and Figure 5.4 illustrate the mean queue time results respectively for single item processing and the batch processing models. The queues seen in Op1 are simply due to the fact that a number of items are released simultaneously in the line while Op1 is a single item processing operation. The difference seen between production loads corresponds to the different number of items released simultaneously.

In the batch processing model, it should be noted that the highest production load exhibits the smallest mean queue time at the batch processing operation. This is because

the rate of introduction of new items into the line means that there are sufficient items available to form complete batches at any instant. As production load decreases, the mean queue time increases as insufficient items are released together into the line to form a full batch. Un-batched items must wait almost twelve hours, until the next release, to complete batching.

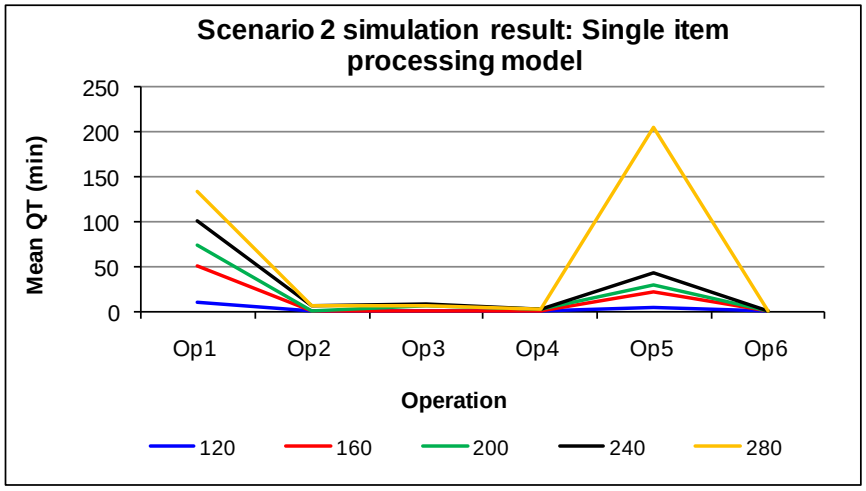


Figure 5.3: Scenario 2 simulation results: Mean queue time of single item processing model

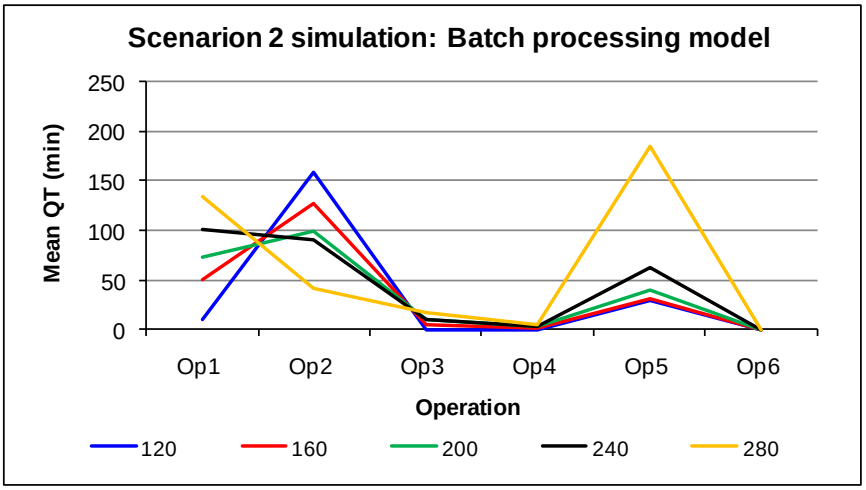


Figure 5.4: Scenario 2 simulation results: Mean queue time of Batch processing model

As the total production load is reduced, the small number of items that have to wait almost twelve hours for batching becomes a larger percentage of the total number of

items released and the queuing time of these items has a larger impact on determining the mean queuing time of items as outlined in Figure 5.5. In Figure 5.5, blue lots represent lots processed immediately, whereas red, green and yellow lots represent the lots in insufficient number to complete a full batch and have to wait until the next release.

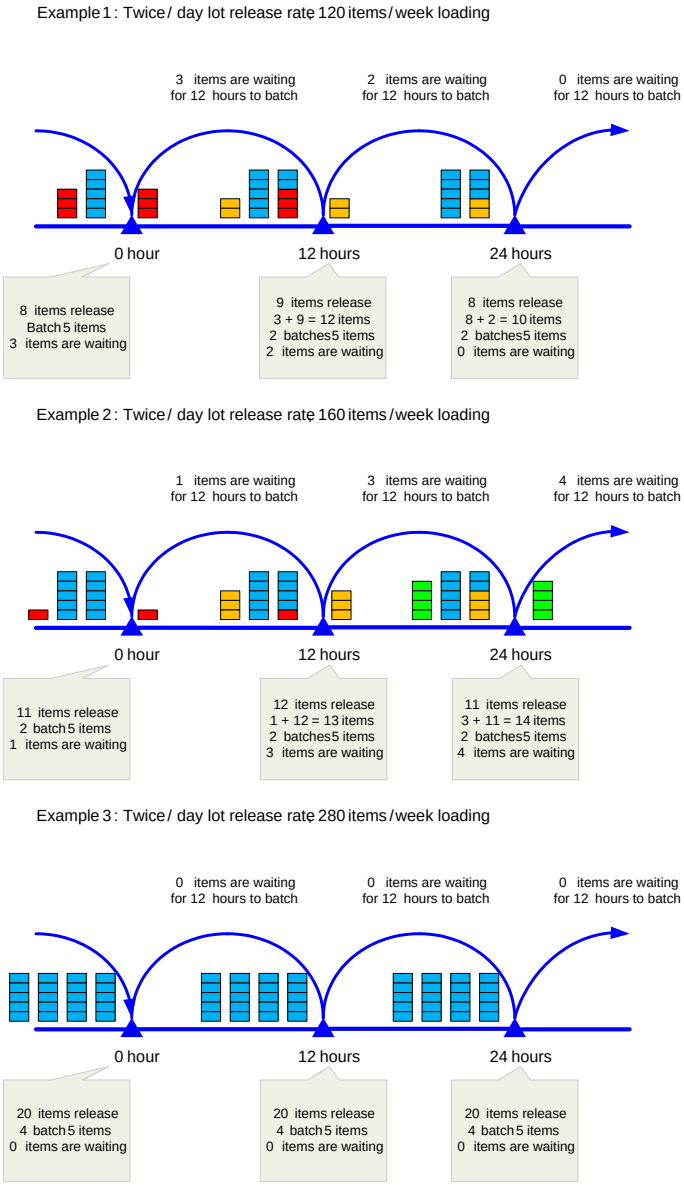


Figure 5.5: Waiting to batch example

In conclusion, the lot release policy appears to be critical when a batch operation is involved. These phenomena will be further investigated in the following scenario.

5.2.3 Scenario 3: Initial Assessment of Item Release Rates which Minimize Queuing for Batching in BPM Model

To reveal the extent to which the phenomenon outlined in Figure 5.5 affects performance, the production loads of 70, 140 and 210 items/week were studied. Indeed, these three production loads release items in multiples of the batch quantity.

Only the batch processing model is tested in this scenario; its structure and setup was described in Section 4.3 (p102) and represented in Figure 4.3 (p105). As a reminder, the batch size is 5 items. Performance measures were defined in Section 4.5, p106. Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5, p106). Total simulation run time is 18 months. Release profile is fixed at twice/day.

Figure 5.6 gives the mean queue times for each buffer. It shows that releasing the items into the line in quantities that are multiples of the batch size (MBS), reduce queue times at Operation 2. This is exhibited by the grouping on the graph of the solid lines (MBS releases).

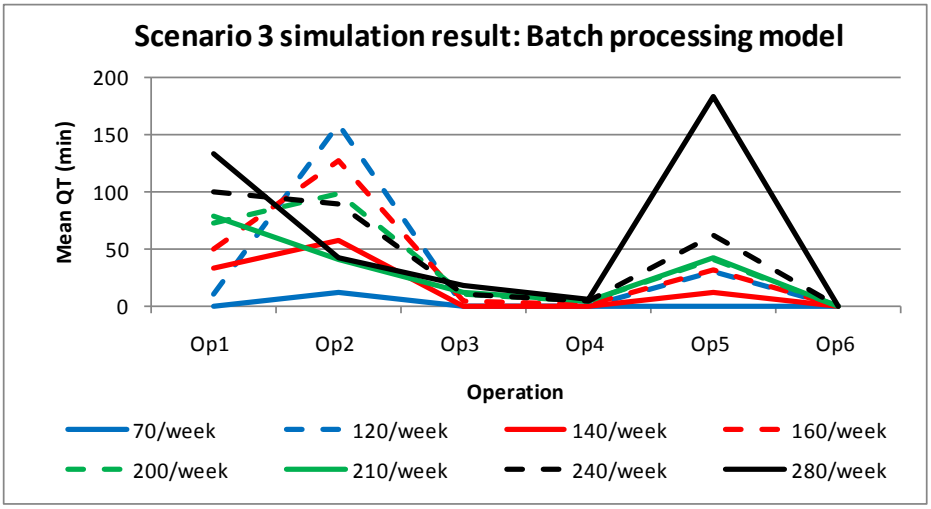


Figure 5.6: Scenario 3 simulation results: mean queue time of batch processing model

5.2.4 Scenario 4: Further Assessment of Item Release Rates which Minimize Queuing for Batching in BPM Model

Given the results from Scenario 3, the performance of the batch processing line when the release strategy is designed to suit the batch process independent of the production load was explored. This was achieved by releasing the items twice a day in differing quantities which were multiples of the batch size and which fulfilled the load in terms of the total number of items introduced in a given week.

Only the batch processing model is tested in this scenario. Its structure and setup was described in Section 4.3 (p102) and represented in Figure 4.3 (p105). As a reminder, the batch size is 5 items. Performance measures were defined in Section 4.5 (p106). Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5, p106). Total simulation run time is 18 months. Release profile is fixed at twice/day. Four different production loads were explored; 120 items/week, 160 items/week, 200 items/week and 240 items/week. Originally, these four loads were

released without considering the batch size (NMBS release policy) in Section 5.2.2. This time, these four loads will be tested with a MBS release policy. Then, the results will be compared to determine the optimum release function in batch processing model simulation.

Figure 5.7 compares the results from these improvements to those previously obtained, dashed lines representing the policies not releasing in multiples of the batch size (NMBS). These shows that poor release strategies can increase the queue at the batching operation by a factor of 3.

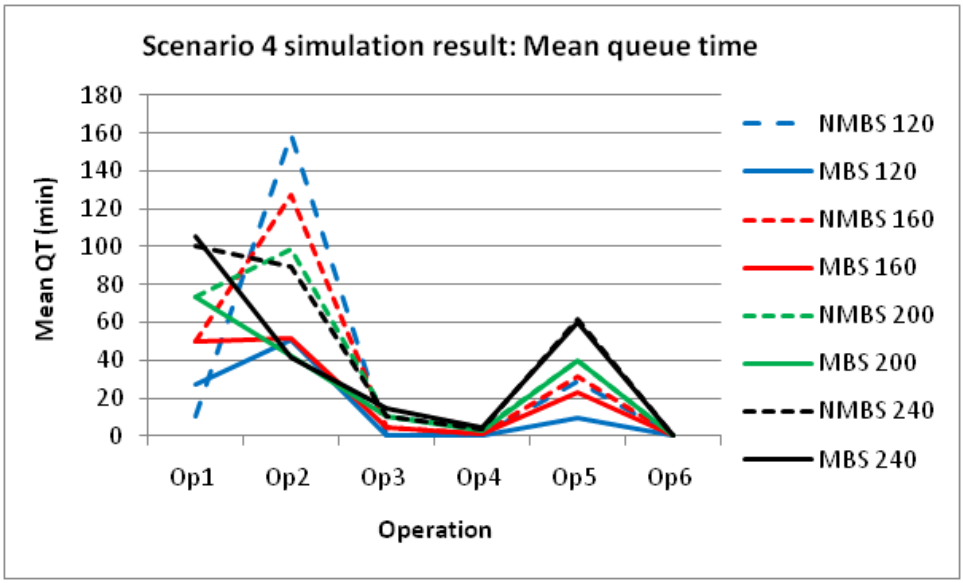


Figure 5.7: Scenario simulation results: Mean queue time for the batch processing model; non-matched batch size and matched batch size release policies (NMBS and MBS)

Figure 5.7 compares the queuing time at the constraint operation for the Matched Batch Size (MBS) and Non Matched Batch Size policies (NMBS) for the four production loads. A clearly significant reduction in queuing for light loads results from designing the release strategy to account for the batch size, resulting in a double benefit at

operations 2 and 5. However, as the loadings increase, this benefit at the constraint is eroded leaving only the savings in queue time at the batch processing operation.

It is expected that this improvement in queuing at the constraint process results from a better spacing of the arrivals as a result of the improved flow through the batching operation. This is facilitated by the low loading of the line, giving effective spare capacity at the constraint, to cope with almost simultaneous arrival of a group of items from a single batch. As the production load approaches the capacity of the line, this facility disappears and so there is no marked improvement at the constraint operation. This conjecture requires further investigation to establish the exact nature of the phenomenon.

Figure 5.8 and Figure 5.9 show that as production load increases the mean cycle time and cycle time variability, in the matched batch size item release, are much smoother than the non-matched batch size item release. This means that the performance of the line, as loadings change, will be much more stable if the batch size is considered when start quantities are decided.

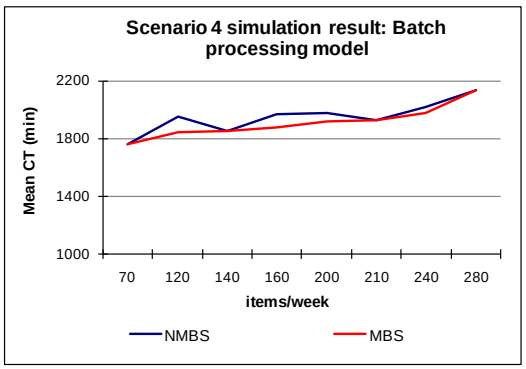


Figure 5.8: Scenario 4 simulation results: mean cycle time for the batch processing model; non-matched batch size and matched batch size release policies (NMBS and MBS)

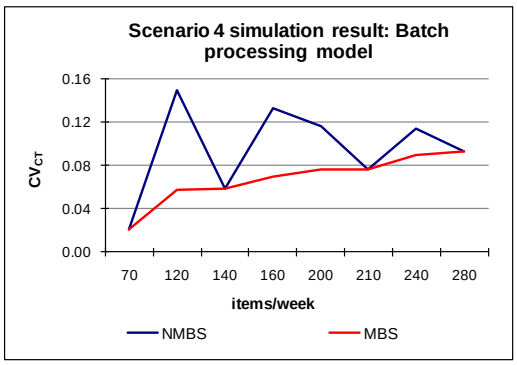


Figure 5.9: Scenario 4 simulation results: cycle time coefficient of variation for the batch processing model; non-matched batch size and matched batch size release policies (NMBS and MBS)

5.2.5 Key Insights from Investigation of Batching Operations

The results show that with a batch processing operation, the release profile affects the cycle time and cycle time variability of a line. The impact of the extra delays incurred in the batch formation may be exacerbated by poor control of the arrival of items into the batch processing operation. Here the issue is not simply that of inter-arrival times of individual items for processing, but rather that a sufficient quantity of items to allow for complete batch formation should arrive in reasonable proximity to each other. This was achieved in the simulations by controlling the release of items into the line. In an actual semiconductor production line, planning the arrival of items to the batch operation might be more difficult. Indeed, re-entrant lines introduce variability into the items arrival. Thus, items would not arrive to the batch in multiple of the batch size, unless the release takes into account the re-entrant lines. This would make controlling the release much more complex.

The line model used here has the particular characteristic that, due to the large overcapacity in processes other than the constraint, items are delivered from the start of

the line to the constraint process with little delay. This means that the inter-arrival times are not altered significantly from entry to the constraint process for the single item processing model. It is believed that the reduced queue time at the constraint process exhibited in the models with a batch processing operation arises from the changes in the departure rate distribution from that operation. In essence, for the line simulated in this model, the batch processing operation distributes favorably the arrival of the items into the constraint operation. This benefit increases as the loading of the line drops, as the excess capacity in the constraint operation can deal with the items arriving together from a single batch before the arrival of the next group. At higher loadings this is not the case and the items must queue at the constraint.

5.3 Simulation of a Tool Availability's Impact on a Production Line with Constraint Operation and a Batch Process Operation

This simulation extends the previous model by introduced downtime in operation 3. Therefore, it includes batch process, tool availability and constraint operations in the analysis. This simulation will analyze the interactions of batch process, tool availability and constraint operations, and highlight the issues affecting the entire line. Model structure and setup are represented respectively in Figure 4.4 (p106) and Table 4.1 (p104). Performance measures were defined in Section 4.5 (p106). Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5). Total simulation run time is 18 months. Twice per day release profile has been mentioned as a good compromise in a previous simulation (Section 5.2.1) and a loading of 280 items

per week, at the maximum capacity of the constraint operation, is interesting to investigate the system under stress conditions. Therefore, the loading in the following experiments is 280 items per week, and items are released twice per day (MBS).

Only one operation will be subject to downtime in order to isolate the effect on the line, and particularly on the constraint operation. Operation 3 was considered for tool availability simulation. Each machine in operation 3 has a MTBF (mean time between failures) and a MTTR (mean time to repair). The downtime profile is represented as shown in Figure 1.2. When a machine goes down, it has to wait for the current lot to finish processing before going down.

An analysis of the fab data shows that the availability of tools is generally between 70% and 90%, therefore availability was fixed to the median value of 80% for the simulation.

Op3 has 13 machines, each with an availability of 80%. Thus, the average capacity of Op3 is 336 items per week. It also means that on average, there should be 2.6 machines down at any given time. Therefore despite its reduced capacity, Op3 capacity remains on average higher than the constraint operation capacity.

5.3.1 Experiment 1: Impact of Downtime Frequency - Experiment Design

This experiment is to study the impact of downtime frequency on mean cycle time and cycle time variability. The machine's TBF and TTR have an exponential distribution (Appendix A.1.5), characterized by their mean value. As was explained in Section 2.4 (p26), the exponential distribution seems to be the most general distribution reported in

the fab [26]. Five different downtime frequencies were designed, respectively, shift (12 hours), day, week, month, and 6 months.

Availability is kept constant by calculating the appropriate TBF and TTR mean values (Table 5.5). Therefore, on one extreme, machines fail one time each shift for a short period of time (144 minutes), and on the other extreme, machines fail only one time every 6 months but for a long period of time (48384 minutes). For comparison, a simulation without downtime (availability is 100%) was also included.

Table 5.5: Operation 3 TBF and TTR input data

Distribution	Down time frequency type		MTBF (min)	MTTR (min)	Availability
None	No downtime	0	∞	0	1
Exponential	Shift (12 hrs)	720	576	144	0.8
Exponential	Day (24 hrs)	1440	1152	288	0.8
Exponential	Week	10080	8064	2016	0.8
Exponential	Month	40320	32256	8064	0.8
Exponential	6 Month	241920	193536	48384	0.8

5.3.2 Experiment 1: Impact of Downtime Frequency - Experiment Results

Figure 5.10 gives a visual representation of the results. It displays the mean cycle time and cycle time variability obtained from each simulation.

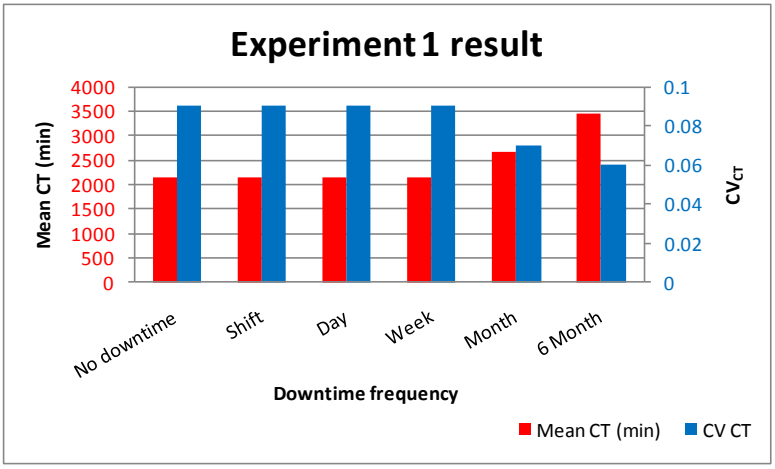


Figure 5.10: Experiment 1 results: mean cycle time and coefficient of variation cycle time

Shift, day, and week downtime frequency are showing almost no impact on the mean cycle time. It remains almost identical to the results obtained without downtime. On the other hand, month and 6 months downtime frequency are showing a considerable increase in cycle time.

Long downtimes are typically unscheduled failures of equipment. The various reasons that could prevent a fast repair were given in Section 2.4. On the other hand, preventive maintenances are short and frequent downtimes. The results (Figure 5.10) show that they do not impact cycle time.

What are the implications for a real factory? Frequent maintenances of the machines preventing the occurrence of infrequent long failure periods will considerably improve the cycle time. In other words, thorough and efficient maintenance schedules should be implemented in all factories.

A detailed analysis of the data shows that the increase in cycle time increase mainly originates from buffer 5 (Table 5.6).

Table 5.6: Experiment 1 results: Mean Queue Time In operation 5 (Buffer 5)

Downtime frequency type	Op5 Mean QT (min)
No downtime	184.21
Shift (12 hrs)	178.61
Day (24 hrs)	174.18
Week	185.42
Month	661.21
6 Month	1422.71

Items are queuing in Op5 buffer to be processed. So, downtime in Op3 is actually affecting the constraint operation (Op5). What is the relationship between these two operations? The results show that the cause is operation 3's output variability. The average availability of Op3 was fixed at 80 %. It does not mean that the availability is constant. Each shift the number of machines down is different. For example, for the 1 month frequency simulation, the availability is fluctuating greatly, from values as low as 46% up to 96% (recorded from simulation data).

A loading of 280 items/week means a release of 20 items each shift. Operation 3's capacity of 420 items/week means a capacity of 30 items per shift. If the availability for the shift goes under the critical availability of 66.66% (20 items ÷ 30 items = 0.6666 = 66.66%) then the capacity for this shift is inferior to the loading. The output of operation 3 is dropping and a queue is formed in buffer 3 (Table 5.7).

Let's take the 46% availability as example:

- In this shift, arrival to operation 3 is 20 items. Operation 3 will only be able to complete 13 items (30 items x 0.46 = 13.8). So operation 3 output is 13 items and 7 items will remain in buffer 3.

- Next shift there will be 27 items to process (7 items left + 20 items from loading). If availability in the next shift is good, operation 3 is able to process all these items giving an output of 27 items.

Operation 5 is the constraint. Its capacity is only 20 items per shift. If 27 items suddenly arrive during a shift, operation 5 will not be able to process them. They will have to queue. Operation 5 being the constraint, it cannot easily clear any queue forming in its buffer. Therefore items will have to queue for a long time in Op5 buffer.

Table 5.7: Interaction between tool availability with high capacity operation and low capacity operation

High capacity operation 3 (420 items)		Low capacity operation 5 (280 items)
30 items/shift		20 items/shift
Tool Availability		
95%	28 items	20 items
90%	27 items	20 items
85%	25 items	20 items
80%	24 items	20 items
75%	22 items	20 items
70%	21 items	20 items
66%	20 items	20 items
60%	18 items	20 items
55%	16 items	20 items

Safe variability

Unsafe variability

This explains the differences between the different downtime frequencies. Shiftly downtime only fluctuates slightly around 80%. It is only exceptionally under 66.66% as the downtime is spread between all the shifts. As the downtime frequency reduces, the downtime is no more spread and some shifts are impacted more heavily than others. The data show that the proportion of shifts whose availability is under 66.66% increases and also the minimum availability reached is decreasing. This is illustrated by the coefficient of variation of Op3 shiftily availability (Table 5.8).

Table 5.8: Variability of Op3 shiftily availability

Downtime frequency type	Coefficient of variation of Op3 shiftily availability
No downtime	0.00
Shift (12 hrs)	0.07
Day (24 hrs)	0.10
Week	0.11
Month	0.13

5.3.3 Critical Availability Definition

From now on, this study will refer frequently to the idea of Critical Availability (CA) of an operation. This idea needs to be defined clearly:

The Critical Availability (CA) of an operation of capacity C is the operation's availability that limits the production output of this operation to the loading level (R). It is given by $CA = \frac{R}{C}$.

As illustrated in Table 5.7, if the loading is R = 20 items/shift and an operation has a capacity of C = 30 items/shift then the operation critical availability is CA= 20/30= 66%. If the availability is lower than 66% then the operation cannot process all the items received.

If during a shift, the availability (A) of an operation is lower than its critical availability, a queue will appear in the operation buffer. When the availability of the operation returns to its standard level, the operation will quickly process the queuing items and a 'bubble' of items will be transferred down the line. When this bubble reaches the constraint operation, it generates long queues (CT) and CT variability.

5.3.4 Experiment 1: Impact of Downtime Frequency - Key Insights

These results show that an operation manger should consider the shiftly operation availability and its variability instead of the average availability of the operations. Queue must be avoided in front of any high capacity operation. Indeed fluctuations in the output of a high capacity operation will have grave consequences when they reach the constraint operation in the line.

5.3.5 Experiment 2: Impact of Repair Time Variability - Experiment Design

This experiment is to study the impact of repair time variability (TTR) on mean cycle time and cycle time variability for two downtime frequencies (daily and weekly). The model and the basic set-up are identical to the previous experiment. Model structure and setup are represented respectively in Figure 4.4 (p106) and Table 4.1 (p104). Performance measures were defined in Section 4.5 (p106). Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5, p106). Total simulation run time is 18 months. The loading input is 280 items per week, and items are released twice per day.

Table 5.9: Day downtime frequency

Downtime frequency type	Lognormal distribution					Availability
	TBF (min)		TTR (min)			
	Mean	Standard deviation	Mean	Standard deviation		
Day	1152	1152	288	100%	288	80%
				75%	216	
				50%	144	
				25%	72	
				0%	0	

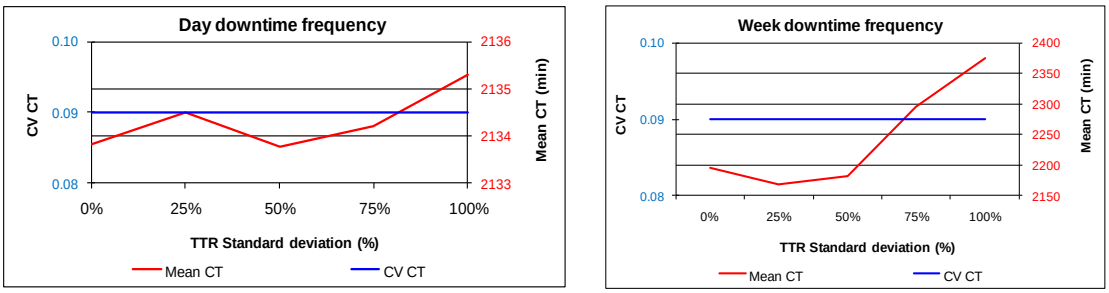
Table 5.10: Week downtime frequency

Downtime frequency type	Lognormal distribution					Availability
	TBF (min)		TTR (min)			
	Mean	Standard deviation	Mean	Standard deviation		
Week	8064	8064	2016	100%	2016	80%
				75%	1512	
				50%	1008	
				25%	504	
				0%	0	

In this experiment, only two downtime frequencies are run: day and week downtime frequencies (Table 5.9 and Table 5.10). Availability is again kept constant at 80 % by calculating the appropriate TBF and TTR mean values. The changes are in operation 3 downtime set-up. The distribution is now a lognormal distribution (Appendix A.1.6) which is characterized by its mean value and its standard deviation. In order to study the impact of repair time variability, the TTR standard deviation was varied. Five simulations are run using as standard deviation 100%, 75%, 50%, 25% and 0% (constant TTR) of the mean TTR value. TBF standard deviation is kept constant at the same value than mean TBF.

5.3.6 Experiment 2: Impact of Repair Time Variability - Experiment Results

Average cycle times and cycle time coefficient of variations were calculated for each standard deviation of the repair time both for the day downtime frequency and week downtime frequency. Results are exposed in Figure 5.11



(a) (b)

Figure 5.11: Experiment 2 results: (a) Mean cycle time and coefficient of variation cycle time vs variability of repair time; Day downtime frequency (b)Mean cycle time and coefficient of variation cycle time vs. variability of repair time; Week downtime frequency

Figure 5.11 shows that TTR variability (TTR Standard deviation) for daily downtime has no significant impact on the cycle time. Mean cycle time remains between 2133 and 2136 minutes, thus a very small variation. However, Figure 5.11shows that weekly downtime is affected. There is a significant variation of mean cycle time.

The explanation is similar to the one for experiment 1. Again, the increase in cycle time is due to the queue in buffer 5 (Table 5.11 and Table 5.12).

Table 5.11: Experiment 2 results: Mean queue time in operation 5 (Day downtime frequency)

Downtime frequency type	Lognormal distribution TTR (min) Standard deviation		Op5 Mean QT (min)	Op5 CV CT
Day	100%	288	174.99	0.4
	75%	216	174.85	0.4
	50%	144	175.53	0.4
	25%	72	177.47	0.4
	0%	0	177.12	0.4

Table 5.12: Experiment 2 results: Mean queue time in operation 5 (Week downtime frequency)

Downtime frequency type	Lognormal distribution TTR (min) Standard deviation		Op5 Mean QT (min)	Op5 CV CT
Week	100%	2016	394.74	0.30
	75%	1512	318.53	0.31
	50%	1008	206.34	0.37
	25%	504	193.63	0.35
	0%	0	219.42	0.36

TTR variability means that the repairs will sometimes take longer, reducing the availability for the affected shift under the critical 66.66% and thus increasing queue in Op5. Daily downtime is not affected as the shift availability is at a safe margin above the critical availability.

5.3.7 Experiment 2: Impact of Repair Time Variability - Key Insights

Controlling the variability of the repair time becomes critical if the availability is close to critical availability. This conclusion can probably be extended to any source of variability. If the availability of a high capacity operation is at a safe margin above the critical availability, variability in this operation will not affect the cycle time. On the

other hand, if the availability is close to the critical availability, then any source of variability will create considerable strain on the constraint operation.

CHAPTER - 6 DEVELOPMENT OF A RELEASE STRATEGY

6.1 Introduction

Up to now, the important characteristics of a production line behavior were discovered and studied. In a second phase of this project, the aim is to apply the knowledge acquired to improve and optimize the performance of the production line. How can the results obtained be transformed and applied in a real environment?

The approach to the problem should be modified. So far, the variables considered were down time, availability, and regular loading. But in a real environment, these are given and a manager cannot easily adapt or modify them. A customer orders a given number of items; the fab manager has to produce them as fast as possible in order to have the production line available for the next customer. The only parameters, that can easily be modified to optimize the production, are the number of items introduced daily in the fab, the number of items extracted from the various buffers to be produced at each operation and the scheduling of the preventive maintenance on the machines. One could think about the number of machine as well, but this is generally fixed by the fab floor area, facilities, transport access to the designated area, and budget, as some of the machines cost several millions Euros. Therefore, increasing the capacity by adding new machines can only be a long term plan. So new variables have been defined: line loading, buffer lot release and maintenance schedule.

What could be the target?

It could be to increase the output rate of the line while maintaining, or ideally decreasing, the level of work-in-process inventory.

How can it be achieved?

The primary variable that controls the performance of a production line is the release of items into the beginning of the process. Implementing a release rule with a specific parameter value will lead to improved performance measured in terms of variables such as the average rate of output from the line, and the level of WIP within the line. By varying the parameter, one observes a set of (output rate, WIP level) pairs that is achievable by the policy. One policy is superior to another policy if, for every output rate within a target range, the policy requires a lower level of work-in-process inventory [70].

A new release strategy needs to be devised. This new release strategy should be a product of previous simulations and modeling processes. Therefore according to Experiment 1 key insights (Paragraph 5.3.3, p131), it should avoid any variability in the flow of items in the operations preceding the bottleneck and consider the availability of an operation. And according to experiment 2 key insights (Paragraph 5.3.7, p135), this new release strategy should consider the critical availability of the operations.

6.2 Constant Flow (CONFLOW) Release Strategy

It is assumed that avoiding the formation of a queue at a high capacity operation should greatly improve CT and CT variability. How can it be achieved? There are two possibilities. First, maintaining the availability of the high capacity operations above

their critical value $A > CA$, through the management of machine downtimes both scheduled and unscheduled. Second, monitoring the availability of the high capacity operations and momentarily reducing the number of items released when the availability is under its critical value. This second solution defines a new release strategy that needs to be defined in details and tested.

6.2.1 Strategy Definition

A new hybrid push-pull approach for the release strategy, named Constant Flow (CONFLOW), can be devised. As its name indicates, it aims to control the flow of items arriving at the bottleneck station. This is achieved by releasing, as baseline, a regular number of items in the line (push system). This number is reduced when a queue appears at a high capacity operation preceding the bottleneck (pull system). Operations past the bottleneck are not considered.

CONWIP, TOC and CONFLOW policies all aim to limit the amount of WIP in the production line. They all authorize the release of items based on the current status of the line, thus they are all pull strategies. They differ in the methodology employed to determined this status and control the release of items.

As was explained in Section 2.8, CONWIP consider the WIP of the whole production line, whereas TOC policies only consider the operations preceding the constraint operation. Thus CONFLOW policy can be said to belong to the TOC family.

SA, workload, CONLOAD and DBR all measure the output of the bottleneck and release in the line the same amount. They differ in the type of output measured: number of item (SA and DBR), working time (workload) or load (Conload).

The CONFLOW policy differs in that it not only evaluates the output of the bottleneck. It also evaluates the output of all preceding operations. Indeed, with TOC policies, a disturbance in the production line will not be detected until it propagates down the line and reached the bottleneck. Only then will the number of item released be corrected. By controlling all operations, CONFLOW detect the disturbance as soon as it occurs and immediately correct the number of item released. In other words, CONFLOW reacts faster than other TOC policies.

CONFLOW policy might also have an advantage over DBR when the production line processes a mix of products. Indeed, if the proportion of the various products changes, then the position of the bottleneck may shift at short notice. This is an issue for DBR, as the release of items is dependent on the bottleneck output. In CONFLOW policy, the release is not dependent on the bottleneck itself but on the operations preceding it. Even if the bottleneck shifts, most of the operations preceding it will still be monitored and CONFLOW will still provide some level of stabilisation.

So how does CONFLOW evaluate the operation outputs? Actually, as for TOC policies, several possibilities may be considered depending on the line characteristics: items number, working time, etc. This study will use the operations availability as explained in the following chapter. The simulation model only includes one operation with downtime.

6.2.2 CONFLOW Operating Protocol with One Operation Availability

In the model only one operation is submitted to downtime. Therefore only this operation’s availability needs to be considered. The operation’s availability is measured

each shift. The counter i designates the shift number and thus A_i designates the availability during shift i. Each shift, the availability results are compared to the critical availability (Figure 6.1). Two possibilities exist:

- The shift's availability is bigger than the critical availability, and then no action is taken.
- The shift's availability is smaller than the critical availability, then the number of items released in the following shift needs to be reduced (Figure 6.1).

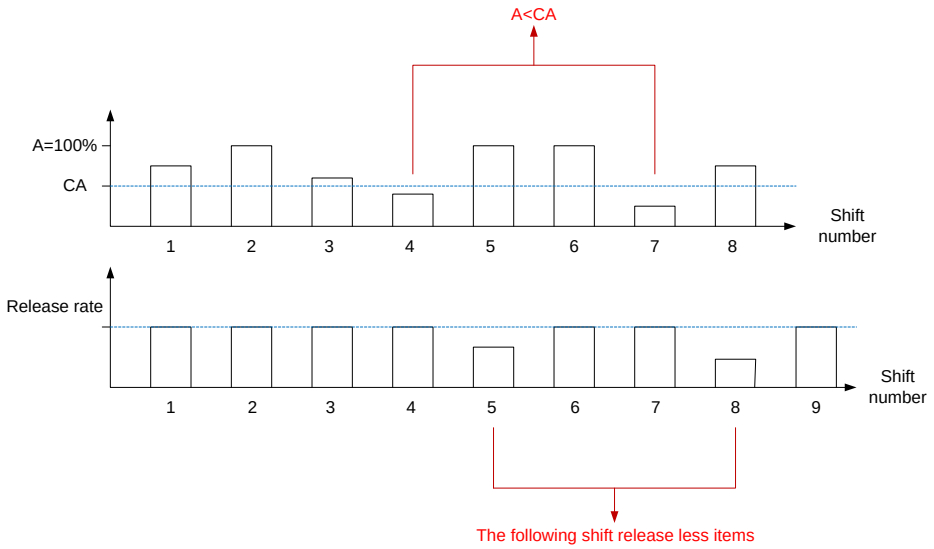


Figure 6.1: Momentarily reducing the number of items released by monitoring the availability level.

In the event of an excursion (availability of a machine drops below CA) then the WIP release strategy should be adjusted to avoid the temporary accumulation of WIP (WIP bubbles).

6.2.3 CONFLOW Operating Protocol with One Operation Subject to Downtime

One example (Figure 6.2) will be used to explain how to compute three different release options for CONFLOW. These three options will be compared to the push system.

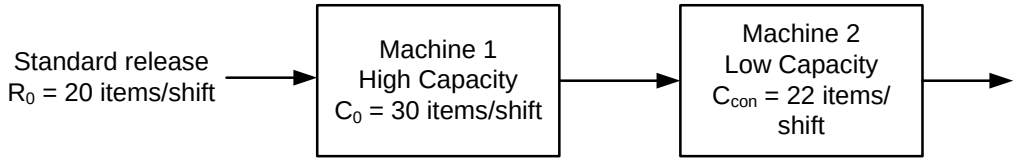


Figure 6.2: CONFLOW release modulation – introduction model

Suppose that the standard release (R_0) is 20 items per shift. Machine 1 has a capacity C_0 of 30 items/shift and Machine 2 has a capacity (C_{con}) of 22 items/shift. Machine 2 (low capacity machine) is almost running at full capacity. Machine 1 is submitted to downtime.

The previous shift availability (A_{i-1}) is assumed to be 50%. How many items should be released in the current shift (R_i) to compensate? Four cases are discussed. Table 6.1 provides the nomenclature used in the example.

Table 6.1: Definition of variables considered in CONFLOW release strategy

R_o	Standard Release
R_i	Release during shift i
C_o	Machine 1 maximum capacity (no downtime)
C_i	Machine 1 capacity during shift i (downtime adjusted)
C_{con}	Constraint capacity
A_i	Machine 1 availability during shift i

Push System

The push system is a static release policy, (p43), new jobs are released into the line at fixed interval time, without considering the status of the line. So, this system does not compensate for downtime and continues to push WIP at the standard rate.

In the push system, the standard release (R_0) is maintained in all shifts (Equation 6.1).

$$R_i = R_0 \qquad \textbf{Equation 6.1}$$

In the example, the release remains 20 items for all the shifts as illustrated in Figure 6.3.

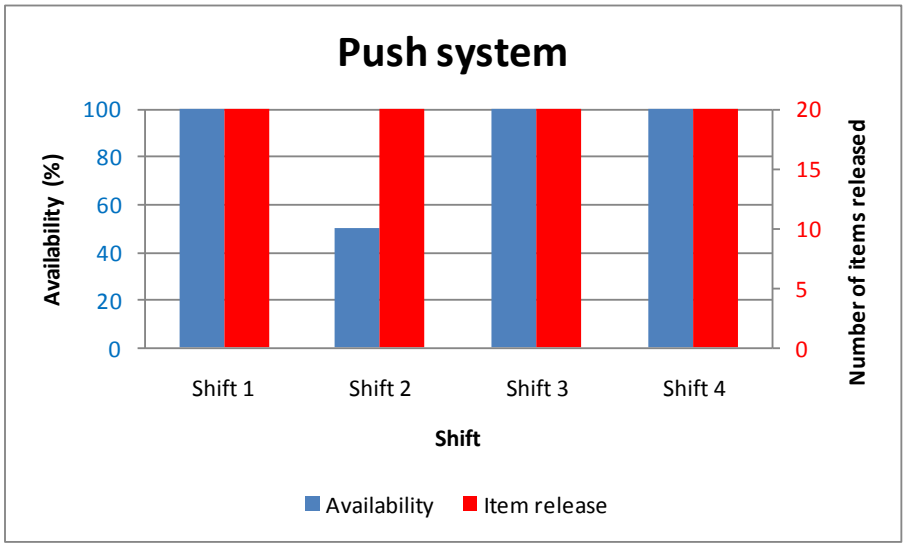


Figure 6.3: Push system

CONFLOW Option 1

CONFLOW option 1 is the first option of CONFLOW release strategy. As mentioned earlier (6.2.2), the release will consider the availability level (machine 1) in order to calculate the release quantity in every shift.

This works as follows: if in the previous shift the machine availability is A_{i-1} = 50%, then how many items should be sent in this shift? Option 1 of CONFLOW system considers that if the availability is 50% in the previous shift, then only 50% of the standard release should be sent in this shift.

In more generic terms, in CONFLOW option 1 the release in shift i is given by Equation 6.2 where A_{i-1} is the availability in the previous shift of the machine with downtime and R_0 the standard release.

$$R_i = A_{i-1} \times R_0 \qquad \textbf{Equation 6.2}$$

So, in the example the release quantity is

R_i = 50% × 20 items = 10 items

In the shift following the downtime, the release will be reduced to 10 items (Figure 6.4). In all other shifts, the release remains standard (20 items).

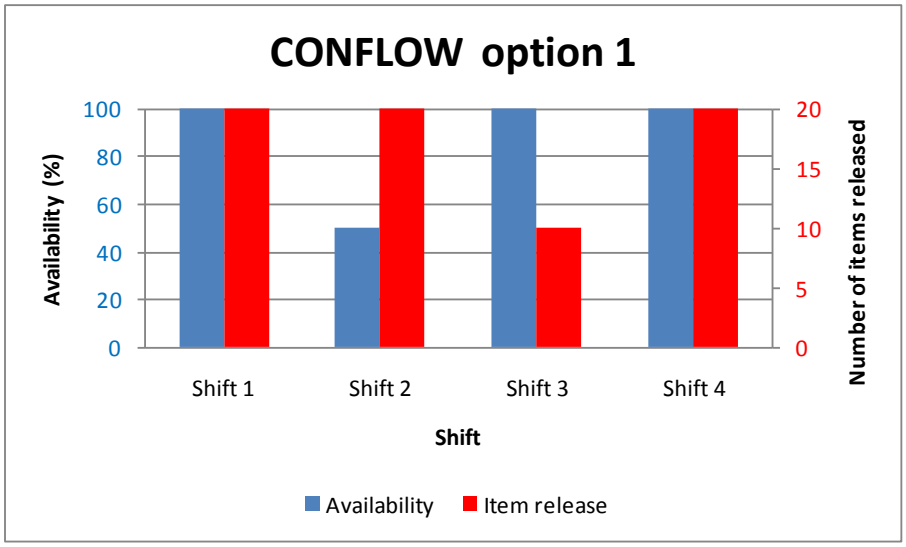


Figure 6.4: CONFLOW release modulation – CONFLOW Option 1

CONFLOW Option 2

CONFLOW option 2 is the second option of CONFLOW release strategy. The difference with option 1 is that option 2 is not only considering the availability from the previous shift. Option 2 also considers the queue of items remaining from the previous shift to calculate the release quantity in this shift.

This is an attempt to maintain constant the number of items arriving at the constraint operation, in other words a Constant Flow (CONFLOW).

First, the capacity (C_{i-1}) of machine 1 in the previous shift needs to be evaluated.

C_{i-1} = 30 items × 50% = 15 items.

Where, 30 items is Machine 1 standard capacity (C_0) and 50% is Machine 1 availability in the previous shift (A_{i-1}).

Therefore,

$$C_{i-1} = C_0 \times A_{i-1}$$ **Equation 6.3**

And the size of the queue is, Queue (Q_{i-1}) = 20 items ─ 15 items = 5 items, where 20 items is the standard release rate (R_0). So,

$$Q_{i-1} = R_0 - C_{i-1} = R_0 - (C_0 \times A_{i-1})$$ **Equation 6.4**

How many items (R_i) have to be sent during this shift to maintain the number of items processed by machine 1 during this shift to its standard level of 20 items?

R_i= 20 items ─ 5 items = 15 items

Where 20 items is standard release rate (R_0) and 5 items is the queue from the previous shift.

So,

$$R_i = R_0 - Q_{i-1} \qquad \textbf{Equation 6.5}$$

Transfer Equation 6.4 into Equation 6.5,

$$R_i = R_0 - Q_{i-1} = R_0 - [R_0 - (C_0 \times A_{i-1})] \qquad \textbf{Equation 6.6}$$

Therefore in CONFLOW option 2, the release in shift i is given by Equation 6.7 where A_{i-1} is the availability in the previous shift and C_0 the standard capacity of the machine with downtime.

$$R_i = C_0 \times A_{i-1} \qquad \textbf{Equation 6.7}$$

In the example, in the shift following the downtime the release will be reduced to 15 items (Figure 6.5). In all other shifts, the release remains standard (20 items).

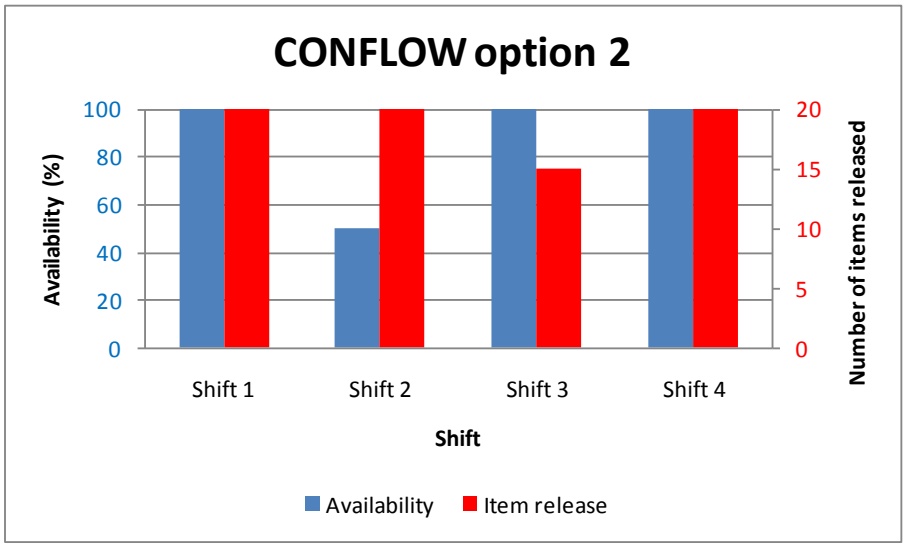


Figure 6.5: CONFLOW release modulation – CONFLOW option 2

CONFLOW Option 3

CONFLOW option 3 builds on option 2 in that it not only considers the availability and queue at the constraint (Machine 2) but also its capacity in determining the quantity to release in a given shift. The objective for a shift, therefore, is to release the maximum number of items the constraint can handle (R_i).

Machine 2 capacity is 22 items, so 22 items have to be processed by Machine 1. So how many items should be released in this shift?

R_1 = 22 items – 5 items = 17 items,

Where, 22 items is the Capacity of the Constraint ($C_{con.}$), and 5 items is the queue from the previous shift (Q_{i-1}). So,

$$R_i = C_{con} - Q_{i-1}$$ **Equation 6.8**

Transfer Equation 6.4 into Equation 6.8,

$$R_i = C_{con} - [R_0 - (C_0 \times A_{i-1})]$$ **Equation 6.9**

Therefore in CONFLOW option 3, the release in shift i is given by Equation 6.10 where A_{i-1} is the availability in the previous shift of the machine with downtime, C_{con} is the constraint capacity, C_0 the standard capacity of the machine with downtime and R_0 the standard release rate.

$$R_i = C_{con} - R_0 + (C_0 \times A_{i-1})$$ **Equation 6.10**

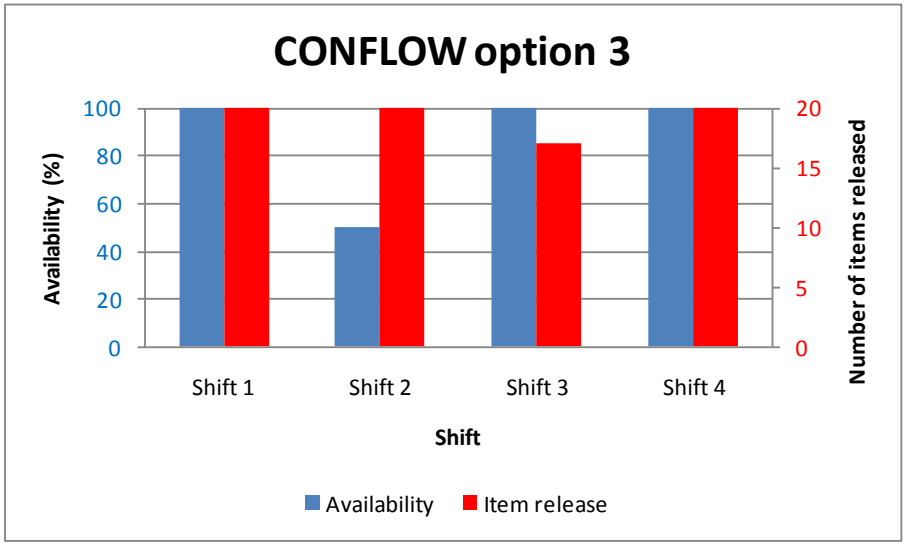


Figure 6.6: CONFLOW release modulation – CONFLOW option 3

In the example, in the shift following the downtime, the release will be reduced to 17 items (Figure 6.6). In all other shifts, the release remains standard (20 items).

6.3 Experiment Design

Seven scenarios are simulated. These are listed in Table 6.2, and are discussed in more detail in subsequent sections 6.3.1-6.3.7.

In **Scenario 1**, recovery performance simulations are run to compare the recovery time of the various release strategies. Afterward, several simulations test the efficiency of these strategies. Simulations start with a two machines model: availability and constraint. Then, batching is introduced followed by parallel processors.

The following step, **scenario 2**, is to increase the length of the line to 5 machines.

Then, **scenario 3** determines the influence of the availability, batch and constraint machines position.

Scenario 4 further compares PUSH and CONFLOW policies by matching their throughput and studying the differences in CT and WIP.

Scenario 5 studies the influence of a re-entrant line on CONFLOW performances.

Scenario 6 introduces failures on multiple machines.

Finally, **scenario 7** compares the performance of CONFLOW to other TOC strategies (SA and DBR).

The high capacity operations have a capacity of 30 items/shift. One machine/operation is affected by random downtimes with an exponential distribution (Appendix A.1.5) (tool availability machine/operation) and its mean availability is fixed at 80% (Section

5.3). Under those conditions, the mean capacity of machine (or operation) affected by downtime is reduced to 80% of 30 items/shift or 24 items/shift.

For each release strategy, the impact of the constraint operation capacity will also be studied. The following capacities will be studied for the constraint: 20, 21, 22, and 24 items/shift (Table 5.7).

The capacity of 20 items per shift for the constraint represents the most severe constraint on the system as it is equal to the targeted output of 20 items per shift. If the capacity of the constraint was lower than this level then management would either have to correspondently reduce the output target or find means to augment the capacity of the constraint to meet the output target. The other values for the capacity of the constraint represent gradually reducing the severity of the constraint on the system while ensuring that this operation still remains the overall constraint within the system on the output achievable.

In summary, several models will be studied. Each model will be exposed to four different release strategies (Push system, CONFLOW option 1, CONFLOW option 2 and CONFLOW option 3), and the behavior of each release strategy will be examined under four different constraint capacities (Table 6.2). Moreover in scenario 3, each combination of position (availability, constraint, and batch) will be studied.

For all simulations, the total simulation run time is 18 months. Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5). The default loading is 20 items per shift. Next section will introduce each simulation model.

Release strategies will be compared based on the average rate of output from the line, and level of WIP within the line.

Table 6.2: Simulation models overview: 4 release strategies and 4 constraint capacities

Scenario	Model	Number of machines/operations	Downtime	Batch	Constraint	Release strategies	Constraint Capacity
1	RPS	2 machines	Yes	No	Yes	Push system CONFLOW option 1 CONFLOW option 2 CONFLOW option 3	20, 21, 22, 24
	SM1	2 machines	Yes	No	Yes		
	SM2	2 machines	Yes	Yes	Yes		
	SM3	2 operations	Yes	No	Yes		
2	SM4	5 machines	Yes	No	Yes		
3	SM5 (B/C/T permutations)	5 machines	Yes	Yes	Yes		
4	Matched throughput strategy	5 machines	Yes	Yes	Yes		
5	Re-entrant	5 machines	Yes	Yes	Yes		
6	Failures on multiple machines	5 machines	Yes	Yes	Yes		
7	TOC vs. CONFLOW	5 machines	Yes	Yes	Yes	SA, DBR	

Where RPS is Recovery Perfomance Simulation, SMi is Simulation Model i and re-entrant is re-entrant simulation.The following sections will further describe the detail of each scenario.

6.3.1 Scenario 1: Two Machines/Operations Model

The standard model is illustrated in Figure 6.7. It is composed of two machines/operations. The first operation is a high capacity operation and is submitted to downtime. The second operation is the constraint.

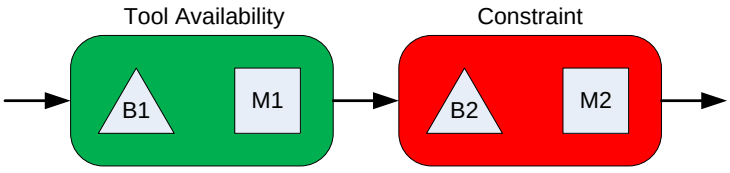


Figure 6.7: two machines/operations simulation model

This model will be tested with four different simulations: Recovery Performance Simulation (RPS), Simulation Model 1 (SM1), Simulation Model 2 (SM2) and

Simulation Model 3 (SM3). For each simulation, Operation 1 and Operation 2 setups are given respectively in Table 6.3 and Table 6.4.

Table 6.3: Operation 1 setup

Operation 1				
Model	Number of machine	Processing time (min)	Capacity (items/shift)	Batch size (items)
RPS	1	24	30	1
SM1	1	24	30	1
SM2	1	120	30	5
SM3	13	312	30	1

Table 6.4: Operation 2 setup

		Operation 2							
		Cap. 1		Cap. 2		Cap. 3		Cap. 4	
Model	Number of machines	Capacity (items/shift)	Processing time (min)	Capacity (items/shift)	Processing time (min)	Capacity (items/shift)	Processing time (min)	Capacity (items/shift)	Processing time (min)
RPS, SM1, SM2	1	20	36	21	34.29	22	32.73	24	30
SM3	8	20	288	21	274.29	22	261.82	24	240

In the following simulations, the process time is fixed. It is kept constant when the model is run. This is actually what occurs in a real factory. Automatic machines repeat the same tasks with a constant process time. Variations can occur only if the type of product processed change. This is not part of the current study.

By default 20 items are released at the beginning of every shift. Therefore the critical availability for machine 1 is 66.66%. This level of release should avoid the building of high queue in the constraint buffer and simultaneously keeps the constraint busy most of the time, avoiding lost capacity. This release will be modulated according to the high capacity machine availability.

Recovery Performance Simulation (RPS)

The simulation model is illustrated in Figure 6.7. Machine 1 and machine 2 setup are given in Table 6.3 and Table 6.4. Machine 1 is down only once for 10080 minutes (week). The downtime occurs in the middle of the simulation (9 months). This simulation studies how long it takes for a system to recover baseline performance in function of the different release strategies applied and the constraint capacity (Table 6.2).

Response to Random Downtime

Downtime (Table 6.5) is random and follows an exponential (Appendix A.1.5) distribution. The average time between failures is 8064 min and the average time to repair is 2016 min. It corresponds to 80% availability (Section 5.3) and to one failure each week. These values also result in 9% of the shift availabilities under the critical availability with shift availabilities as low as 1.45%.

Table 6.5: Machine 1 downtime setup

Distribution	Down time frequency type		MTBF (min)	MTTR (min)	Availability
Exponential	Week	10080	8064	2016	0.8

The second machine is a low capacity machine (constraint/bottleneck). Different release strategies and constraint capacities will be applied from Table 6.2.

For reference, a baseline simulation is run without applying any downtime to machine 1. In this case, there are no differences between the four release strategies as the availability is constant (100%) and never decreases under the critical availability. This simulation is the baseline against which the four release strategies are evaluated.

Simulation Model 1 (SM1): Availability Machine 1 and Constraint Machine 2

The purpose of this model is to test the performance of the release strategies in the simplest configuration: one high capacity machine affected by downtime and one constraint machine. The system is composed of two machines and their buffer (Figure 6.7). Details of the setup are given in Table 6.3 and Table 6.4.

Simulation Model 2 (SM2): Availability and Batch Process Machine 1, Constraint Machine 2

The purpose is to test the behavior of the release strategies in the presence of a batch machine. The difference with the previous model is the introduction of batch process in machine 1. So machine 1 is not only influenced by downtime, but also by batch process. The batch size is set to five items. This batch processing is a parallel process; therefore, the five items are processed at the same time, then un-batched when the process is completed. Details of the setup are given in Table 6.3 and Table 6.4.

Simulation Model 3 (SM3): Parallel Process Simulation Model with Availability Operation 1 and Constraint Operation 2

The previous models were considering only single machines. Here, each operation is constituted of several machines in parallel. The specific number of machines set for each operation is a representative figure informed by production practice at the industrial partner's fabrication facility. This model is intended to test the release strategies when parallel processing is involved. Will there be any difference between this model and simulation model 1?

Setup details are given in Table 6.3 and Table 6.4. Different release strategies and constraint capacities will be applied from Table 6.2.

Operation1 has a high capacity of 420 items per week, which is 30 items per shift (12 hours). For each machine of operation 1, downtime is random and follows an exponential distribution. The average time between failures and the average time to repair (Table 6.5) were kept to the values used in previous models to also have the average operation availability at 80%. But due to the 13 machines and their averaging effect on the whole operation, the variability of the operation's availability from one shift to the next is considerably reduced compare to the two previous models. The operation's availability is less often under the critical availability than in the two previous models. There are around 7.5% of shifts under the critical availability and the lowest shift availability is significantly higher around 41%. Therefore the impact of the various release strategies is reduced in model 3 compare to models 1 and 2. Smaller differences between the push system and CONFLOW Option 1, 2 and 3 are expected.

6.3.2 Scenario 2: 5-Stage Serial Line with Constraint and Downtime (SM4)

So far, the previous models only considered a line composed of 2 machines (or operations). Here, the line will extend to 5 machines. This model is an extension of scenario 1, simulation model 1. The purpose is to test what could happen when the line is longer. What are the differences between short and long line simulation models?

The system is composed of five machines and their buffer. One machine is subject to downtime and another one is a constraint. Setup details are given in Table 6.6. Different release strategies and constraint capacities will be applied as per Table 6.2.

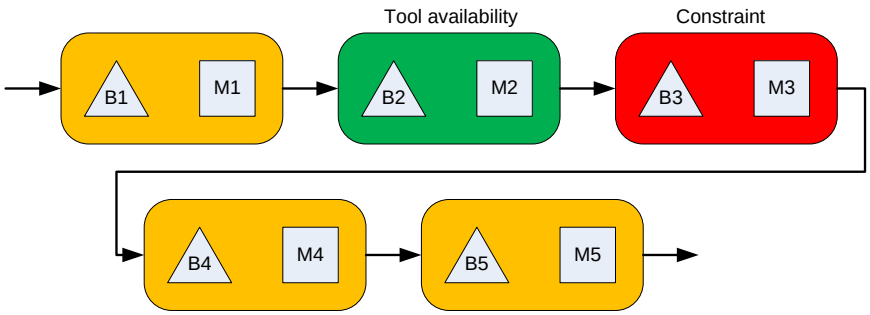


Figure 6.8: Simulation model 4

Table 6.6: Simulation model 4 (SM4) setup

Setup	M1	M2	M3				M4	M5
			Cap. 1	Cap.2	Cap.3	Cap. 4		
Number of machines	1	1	1				1	1
Processing time (min)	24	24	36	34.29	32.73	30	24	24
Capacity (items/shift)	30	30	20	21	22	24	30	30
Batch size (items)	1	1	1				1	1

The capacity of all machines, except the constraint machine, is 420 items per week, which is 30 items per shift (12 hours).

For the availability machine, downtime (Table 6.5) is random and follows an exponential distribution. The average time between failures is 8064 min and the average time to repair is 2016 min. It corresponds to 80% (Section 5.3) availability and to one failure each week. These values also provide us 9% of the shift availabilities under the critical availability with shift availabilities as low as 1.45%.

By default 20 items are released at the beginning of every shift. Therefore the critical availability for machine 2 is 66.66%. This level of release should avoid the building of

high queue in the constraint buffer and simultaneously keeps the constraint busy most of the time, avoiding lost capacity. This release will be modulated according to the machine availability.

6.3.3 Scenario 3: 5-Stage Serial Line with Batch, Downtime and Constraint (SM5)

The system is composed of five machines and their buffer. One machine is subject to downtime, one is a constraint and one is a batch process. The original model position is coming from previous earliest experiment model (Figure 4.4). In this scenario, the simulation model is designed with 5 machines instead of 6 operations. The line becomes shorter because the interesting point is the relationship between batch process, tool availability and constraint. Adding high capacity machines increases uniformly the items cycle time. It does not perturb the line flow.

A full factorial set of simulations are run to study the impact of the positions of tool availability, constraint and batch machines in the line. The standard model is illustrated in Figure 6.9. Each machine setup is shown in Table 6.8. Six simulation models are tested by swapping various machine positions. The machine sequence for each model is represented in Table 6.7.

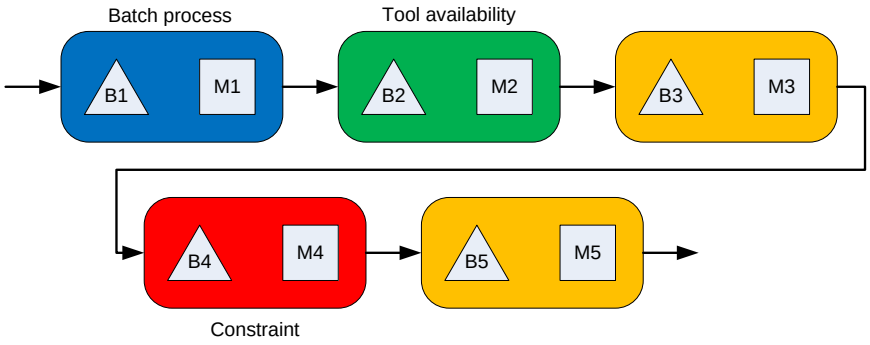


Figure 6.9: Five-stage model

Table 6.7: Six simulation models (SM5 x B/C/T permutations)

Model	Sequence				
	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
SM5 BTC	Batch	Tool availability		Constraint	
SM5 BCT	Batch	Constraint		Tool availability	
SM5 TBC	Tool availability	Batch		Constraint	
SM5 TCB	Tool availability	Constraint		Batch	
SM5 CTB	Constraint	Tool availability		Batch	
SM5 CBT	Constraint	Batch		Tool availability	

CONFLOW (Section 6.2, p138) was defined as a TOC release policy. Therefore, it should not consider what happens after the constraint. Nevertheless, for testing purpose CONFLOW strategy will be applied to the whole line in scenario 3. It means in particular, that the release will be modulated by the operation availability even when the operation is after the constraint.

Table 6.8: Setup used for the five machines

Setup	Batch	Tool Availability	M3	Constraint				M5
				Cap. 1	Cap.2	Cap.3	Cap. 4	
Number of machines	1	1	1	1				1
Processing time (min)	120	24	24	36	34.29	32.73	30	24
Capacity (items/shift)	30	30	30	20	21	22	24	30
Batch size (items)	5	1	1	1				1

The capacities of all machines, except the constraint machine, are 420 items per week, which is 30 items per shift (12 hours).

The batch size is assumed to be five items. This batch processing is a parallel process; therefore, the five items are processed at the same time, then un-batched when the process is completed.

For the availability machine, downtime (Table 6.5) is random and follows an exponential distribution. The average time between failures is 8064 min and the average time to repair is 2016 min. It corresponds to 80% availability and to one failure each week. These values also provide us 9% of the shift availabilities under the critical availability with shift availabilities as low as 1.45%.

By default 20 items are released at the beginning of every shift. Therefore the critical availability for the operation affected by downtime is 66.66%. This level of release should avoid the building of high queue in the constraint buffer and simultaneously keeps the constraint busy most of the time, avoiding lost capacity. This release will be modulated according to the machine availability. Different release strategies and constrain capacities will be applied from Table 6.2.

6.3.4 Scenario 4: Push and CONFLOW Policies Matched Throughput

This scenario is based on the simulation model 5 BTC (order: batch, tool availability and constraint). All the operations' characteristics (machine number, capacity, downtime, batching…) are similar. The aim of this experiment is to facilitate the comparison of Push and CONFLOW. Therefore, the release rate of the push model was reduced to

obtain the same throughput than CONFLOW. The appropriate rate was determined by gradually reducing the simulation release until the throughputs matched. 17 items per shift are released for the push system.

6.3.5 Scenario 5: 5-Stage Serial Line with Batch, Downtime, Constraint and Re-entrant Line

CONFLOW was developed for serial lines without product mix or re-entrant lines. Nevertheless, in practice in front-end semiconductor manufacturing, re-entrant lines are an important factor of variability (Section 2.5, p33). Therefore, this scenario tests CONFLOW in a re-entrant system. Will CONFLOW be able to handle the problem of re-entrant lines?

In a real system, re-entrant lines follow a very complex pattern (Figure 4.1, p101). Complex modeling cannot be attempted directly. CONFLOW need to be evaluated with a simpler re-entrant system. The model 5 BTC is used as a reference. The aim is to demonstrate whether reentrancy is an issue that warrants further exploration, e.g. in future work.

In the model studied (Figure 6.10), all items will go through the whole line twice. In other words, when an item is completed by Operation 5 for the first time, it will be sent to Operation 1 buffer. When an item is completed by Operation 5 for the second time, it will exit the line.

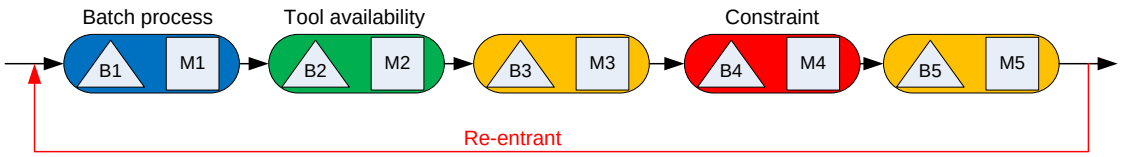


Figure 6.10: Re-entrant line

All the operations' characteristics (machine number, capacity, downtime, batching…) are similar with scenario 3 (simulation model 5 BTC). The amount of items released is halved to 10 items/shift. As all items have to be processed twice, each operation still has to process 20 items/shift. So the overall amount of work remains identical to scenario 3.

6.3.6 Scenario 6: 5-Stage Model with Failures on Multiple Stages

This scenario is based on the simulation model 5 BTC but all machines preceding the constraint (machines 1, 2 and 3) are affected by downtimes (Figure 6.11). Again, BTC is used to demonstrate normal conditions. The aim is to determine whether failures-on-multiple-machines is an issue that warrants further exploration, e.g. in future work. Downtime characteristics are the same than in model 5 BTC. Failures are random and follow an exponential distribution. The average time between failures is 8064 min and the average time to repair is 2016 min. All the other machines' characteristics (machine number, capacity, batching…) are similar to model 5 BTC.

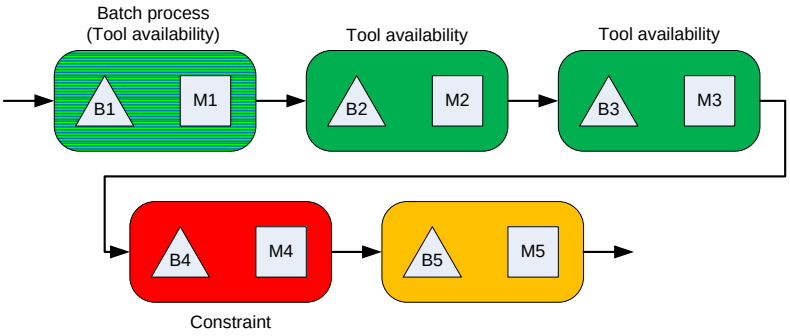


Figure 6.11: Five operations model with failures on multiple operations

In order to calculate the number of items released in the shift, CONFLOW compares the availability of machine 1, 2 and 3 in the previous shift. The lowest availability is used to calculate the number of items released according to the formula given in section 6.2.3.

6.3.7 Scenario 7: TOC vs. CONFLOW

CONFLOW performance needs to be evaluated against well known TOC policies such as Starvation Avoidance (SA) policy and Drum-Buffer-Rope policy (DBR). For a single type of product (no mixed product) and in the absence of re-entrant lines, CONFLOW will be compared to the SA policy. Indeed, it was shown that SA is easy to use and well adapted to those conditions. Then a re-entrant line is introduced and CONFLOW is compared to the Drum Buffer Rope policy as it is better adapted to re-entrant lines.

Starvation Avoidance Policy Setup

The model was built from the model developed in SM5 BTC (Figure 6.9). All the operations' characteristics (machine number, capacity, downtime, batching…) are similar.

The release policy differs. In compliance with the SA policy, the number of WIP from the start of the line down to the constraint machine is maintained constant at a target

WIP level (20 items). New items are released into the line at the beginning of every shift to meet the WIP target. The number of WIP after the constraint machine is not monitored.

Drum Buffer Rope Policy Setup

The model was built from the model developed in scenario 5 (Figure 6.10). All the operations' characteristics (machine number, capacity, downtime, batching…) are similar.

The release policy differs. Initially (first shift), the line was loaded with 10 items. Thus each machine still has to process 20 items each shift (10 items x 2 due to re-entrant line) like in all previous models. Then for each shift, the number of items leaving the constraint machine for the second time is counted. These are the items which already went through the re-entrant line. They are completing their process and will not come back to the constraint machine. In compliance with the DBR policy, at the beginning of the following shift, the same number of items is released in the line.

6.4 Experiment Results

The results for the seven scenarios described are presented. Firstly, the 2 machines/operations model, then the 5 machines serial line with constraint and downtime, then batching is added. A re-entrant line and failures on multiple machines are also tested. Finally, CONFLOW is compared to SA and DBR release policies.

6.4.1 Scenario 1: Two Machines (Operations) Simulation Models

Two types of simulations are studied. Firstly recovery performance is analysed and then response to downtime.

Recovery Performance Simulation

Again, the simulation model is illustrated in Figure 6.7. Machine 1 and machine 2 setup are respectively in Table 6.3 and Table 6.4.

Figure 6.12 represents the results for the four release strategies for capacity 1 of Machine 2. In the push system, the WIP never recovers. If some downtime appears in machine 1 (high capacity machine in front of constraint machine), some items will have to wait due to machine failure. The queue will start to build up in front of machine 1. When machine 1 is back up to normal condition (no downtime), it transfers this queue to the constraint machine. This is called a WIP bubble. Due to limitation of capacity, the constraint machine cannot handle this bubble added to the standard release in the shift. Indeed the standard release rate is 20 items per shift and the constraint capacity is also 20 items per shift. There is no spare capacity to process the queue. This results in an over loading of the constraint machine and the WIP will never disappear.

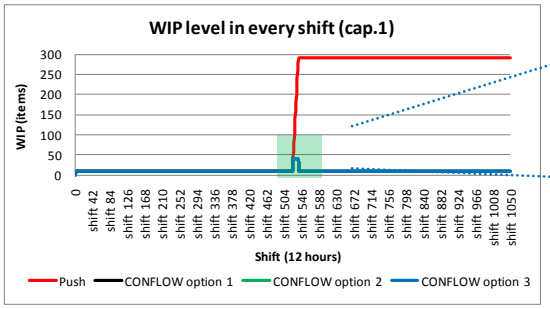


Figure 6.12: Recovery performance simulation results for constraint capacity 1.

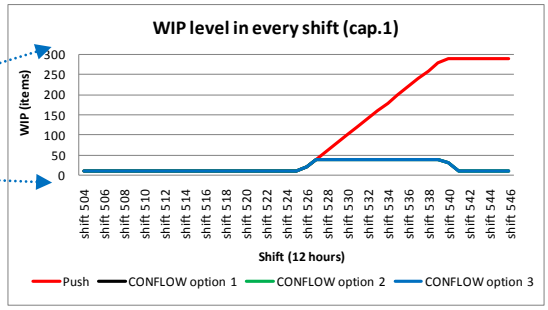


Figure 6.13: Zoom on recovery performance simulation results for constraint capacity 1.

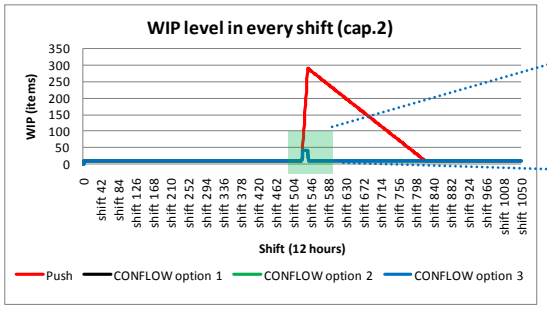


Figure 6.14: Recovery performance simulation results for constraint capacity 2

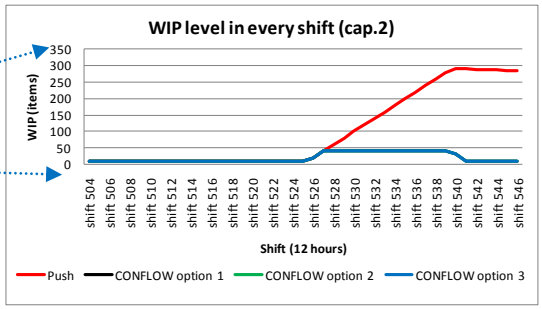


Figure 6.15: Zoom on Recovery performance simulation results for constraint capacity 2

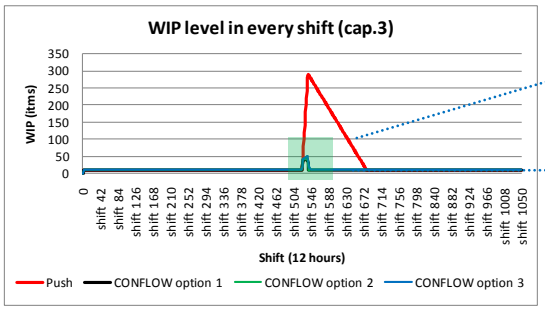


Figure 6.16: Recovery performance simulation results for constraint capacity 3

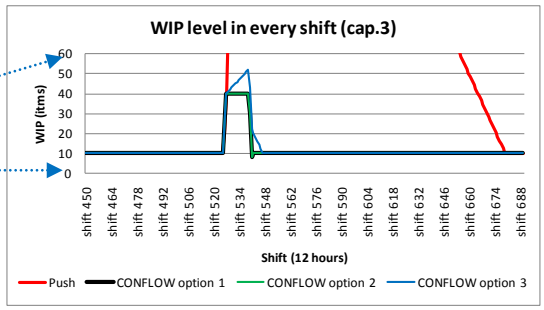


Figure 6.17: Zoom on Recovery performance simulation results for constraint capacity 3

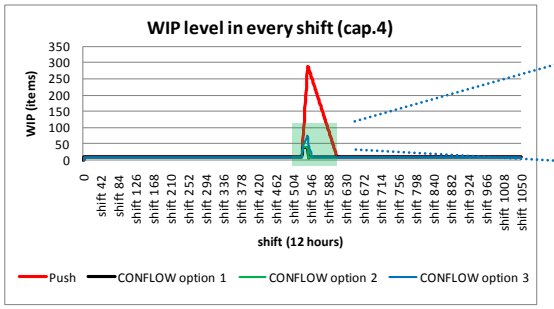


Figure 6.18: Recovery performance simulation results for constraint capacity 4

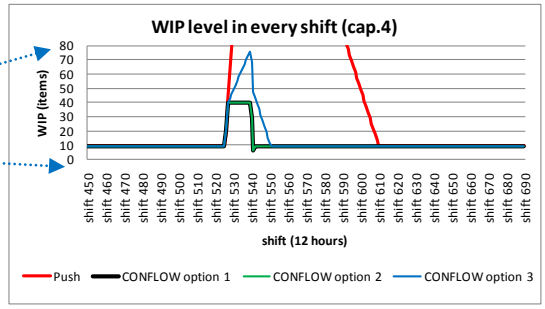


Figure 6.19: Zoom on Recovery performance simulation results for constraint capacity 4

For the constraint capacities 2, 3 and 4 (Figure 6.14, Figure 6.16 and Figure 6.18), the constraint machine capacity is more than 20 items per shift. It is higher than standard release rate. Therefore, machine 2 (constraint) can slowly recover the WIP bubble even in the push system. With constraint capacity 2 the push system needs approximately 20 weeks to recover the WIP. With constraint capacity 3 the push system needs approximately 11 weeks to recover the WIP. With constraint capacity 4, the push system needs approximately 6 weeks to recover the WIP. Overall, with constraint capacity 4, the push system needs a shorter time to recover the WIP because the capacity is higher than constraint capacity 1, 2 and 3.

In CONFLOW option 1 and 2 when the downtime is detected then the release is stopped. No items are sent until the availability comes back above the critical availability. Only then new items are sent into the line. Therefore the WIP level is not increasing while machine 1 is down. In CONFLOW option 3, some few items are still released if the constraint machine's capacity is higher than the standard release (Equation 6.10, p148) as can be seen in Figure 6.17 and Figure 6.19. Nevertheless the WIP number only grows by approximately 10 items for Capacity 3 and 30 items for Capacity 4 during the downtime period (one week). As a result, CONFLOW option 1 and 2 recover the WIP in one shift, once machine 1 is repaired. CONFLOW option 3 needs approximately 10 shifts.

The recovery performance of CONFLOW policy to isolated disturbances is therefore excellent. The next step is to test CONFLOW in a random environment.

Response to Random Downtime

The results obtained for simulation model 1, simulation model 2 and simulation model 3 are similar. Simulation model 1 will be analyzed in details. For simulation model 2 and 3, only differences with simulation model 1 will be given. Figure of the results are displayed in APPENDIX - F.

Simulation Model 1 (SM1): Availability Machine 1 and Constraint Machine 2

First, as can be noticed from Figure 6.20 and Figure 6.21, the baseline is showing the best results. This was expected as there is no downtime applied. The line is able to process all items released. The output rate is identical to the input rate (20 items/shift or 0.0278 items/min). What was less expected is that the push system shows similar results to the baseline when the capacity of the constraint becomes close of machine 1 effective capacity (capacity 3 and 4). Indeed when a bubble of items arrives to the constraint, it is able to process it at a higher throughput and compensate for the starvation preceding the bubble. For capacity 1 and capacity 2, starvation is translated into lost capacity and lower output rate.

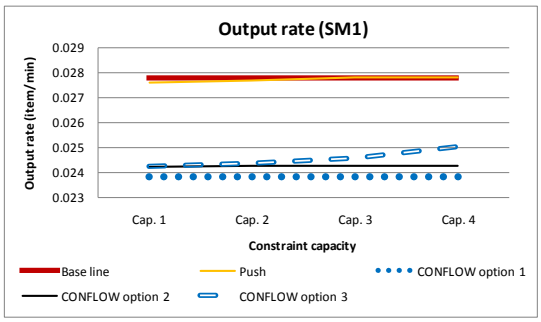


Figure 6.20: Simulation model 1 output rate for all release strategies

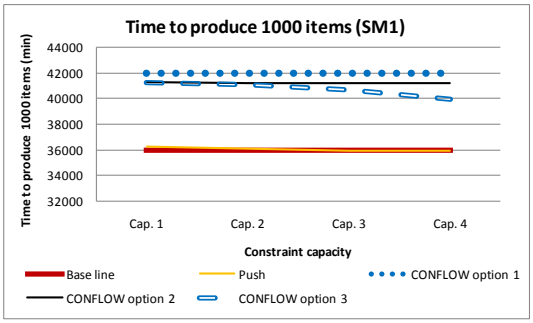


Figure 6.21: Time to produce 1000 items in simulation model 1 for all release strategies

These results also show that CONFLOW option 1, option 2 and option 3 have a slower throughput than the push system by approximately 13%. The push system needs approximately 25 days at capacity 1 (36166.34 min / [60×24]) to produce 1000 items. But CONFLOW option 1, option 2 and option 3, need four days more than push system to produce 1000 items. In summary, CONFLOW option 1, option 2 and option 3 are slowed down by the controlling release strategy. This again is explained by lost capacity. The constraint machine is starved thus reducing the output rate. Among these three release strategies, CONFLOW option 2 and 3 are slightly improving the throughput compare to option 1. This was expected, as the utilization of the constraint machine has been optimized from option 1 to option 2 and then to option 3 (Section 6.2).

Nevertheless the major advantages of the release strategies are that CONFLOW option 1, option 2 and option 3 improve the average cycle time dramatically (Table 6.9 to Table 6.12) and also improve the WIP level in the line (Figure 6.22 and Figure 6.23) in comparison to the push system. This is particularly sensitive at lower capacity of the constraint. For capacity 1, the mean cycle time is improved by as much as 86% and the WIP number by 88%. For capacity 4, they are still respectively at 54% and 61%.

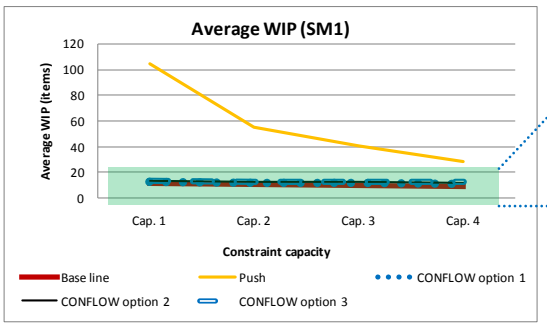


Figure 6.22: Average WIP in simulation 1 for all release strategies

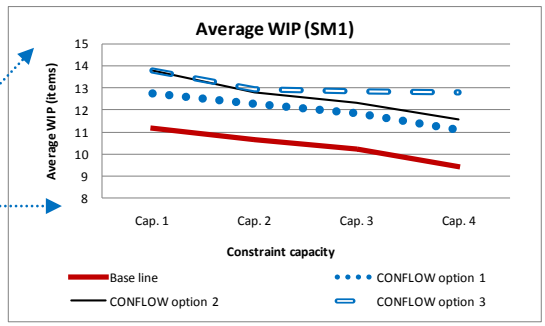


Figure 6.23: Zoom on baseline, CONFLOW option 1, 2 and 3. Average WIP in simulation model 1

Table 6.9: Performance of the policies with respect to cycle time for capacity 1

	Mean	Stdev	Min	Max
Base line	401.995	207.606	60	744
Push	3762.350	872.936	1766.80	5586.712
CONFLOW option 1	525.047	559.022	60	5586.712
CONFLOW option 2	560.146	579.980	60	5586.712
CONFLOW option 3	559.935	579.670	60	5586.712

Table 6.10: Performance of the policies with respect to cycle time for capacity 2

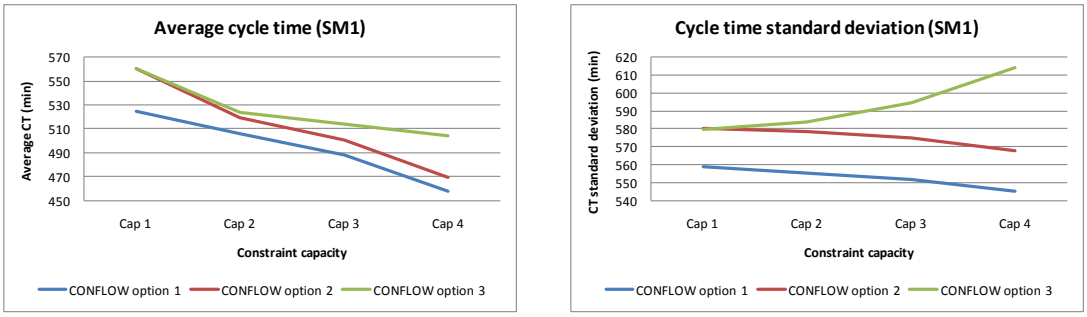
	Mean	Stdev	Min	Max
Base line	384.042	197.739	58.29	709.8
Push	1985.571	1186.314	58.29	5552.512
CONFLOW option 1	505.566	555.052	58.29	5552.512
CONFLOW option 2	519.513	578.285	58.29	5552.512
CONFLOW option 3	524.193	583.575	58.29	5552.512

Table 6.11: Performance of the policies with respect to cycle time for capacity 3

	Mean	Stdev	Min	Max
Base line	367.665	188.737	56.73	678.6
Push	1451.818	1123.226	56.73	5521.312
CONFLOW option 1	488.042	551.760	56.73	5521.312
CONFLOW option 2	500.787	574.746	56.73	5521.312
CONFLOW option 3	514.252	594.637	56.73	5521.312

Table 6.12: Performance of the policies with respect to cycle time for capacity 4

	Mean	Stdev	Min	Max
Base line	339.005	172.990	54	624
Push	1014.332	1004.655	54	5466.712
CONFLOW option 1	457.432	545.341	54	5466.712
CONFLOW option 2	469.352	567.728	54	5466.712
CONFLOW option 3	504.486	613.943	54	5466.712



(a) (b)

Figure 6.24: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

The min CT (Table 6.9 to Table 6.12) for Baseline, Push and CONFLOW Option 1, 2 and 3 are identical except for Capacity 1 in Push. Indeed the values obtained correspond exactly to the processing time. The items have been processed by the machines without doing any queue. In the case of Push Capacity 1, after 3 months running the line (initial-data deletion method), a queue is already existing in front of the constraint, and it never disappears. Therefore no any lots can go through the line without queuing. It explains the higher Min CT.

The max CT (Table 6.9 to Table 6.12) for Push and CONFLOW Option 1, 2 and 3 are identical. Indeed they correspond to the longest failure of machine 1. When the machine fails, items have to wait until the machine is back in working order to be processed. Actually, this failure is so long that in the push system, the queue in front of the constraint has the time to clear before the machine is repaired. Items queue in front of

the down machine and then are immediately processed in machine 2 without further queuing. Therefore the results are identical for Push and CONFLOW.

There is one main difference however. In the push system, while the machine is down, a huge queue is building in the first buffer. This queue (bubble) is then transferred to the second buffer and it remains there. All the items following the failure have long CT. Figure 6.25 displays the cycle time of every item obtained with capacity 1. The release pushes the items all the time. When Machine 1 is down and the availability for the shift is lower than the critical availability then a queue of items is formed. Then, a bubble of items is sent to the constraint when the machine is back up (Figure 6.27). This bubble increases the queue in front of machine 2. The items have to wait longer in the queue and so their CT increase. These CT increase corresponds to the jump seen on Figure 6.25. As can be seen the CT keeps level between the jumps as machine 2 is not able to reduce the queue. The cycle time continues to increase. This is an important observation.

In CONFLOW, no queue is built while the machine is down. Only the items already in line when the machine fails are affected and have long CT. Indeed, the release is controlled in CONFLOW option 1, 2 and 3 in order to reduce the queue in Machine 2. No WIP bubble is created (Figure 6.27) Nevertheless, as can be seen in Figure 6.26 (differences between option 1, 2 and 3 are negligible), some items still have a very high cycle time. They are the items trapped in front of machine 1 when it goes down. They are already in the line and have to queue. The following items won't have to queue. They have low CT. Therefore, average cycle time (Table 6.9 to Table 6.12) is low in CONFLOW option 1, 2 and 3 in comparison to the push system.

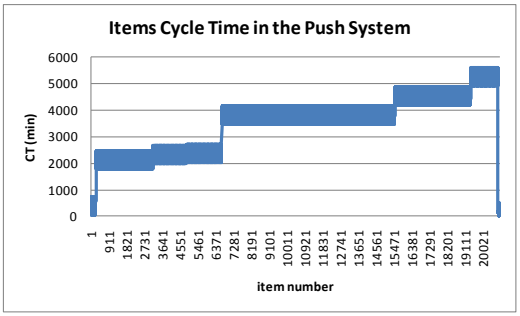


Figure 6.25: Items cycle time for the push system in simulation model 1

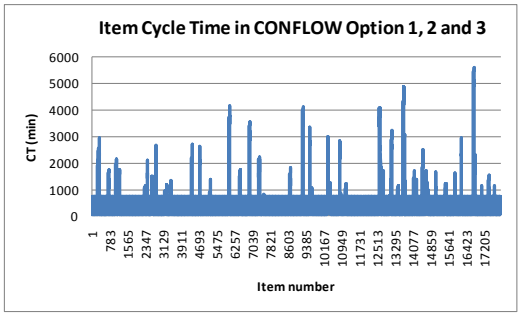


Figure 6.26: Items cycle time for CONFLOW Option 1, 2 and 3 in simulation model 1

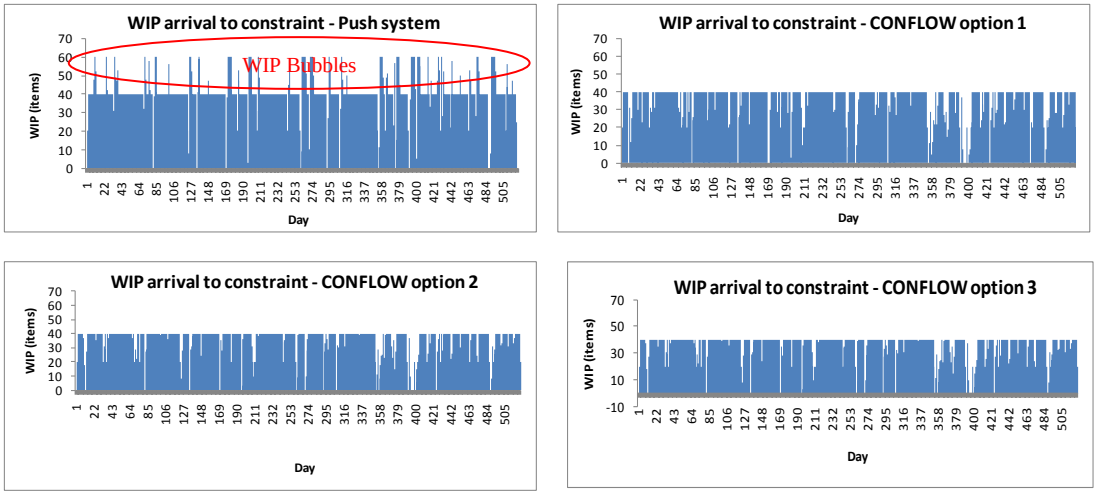


Figure 6.27: WIP arrival to constraint machine for all release strategies in SM1 model

Figure 6.28 zoom on one WIP bubble for the push system. The constraint receives 60 items per day for a period of 4 days (day 185 to day 188). For capacity 1 (40 items/day), it is an excess of 20 items/day. At the end of the 4 days, the queue in front of the constraint as increased by 80 items.

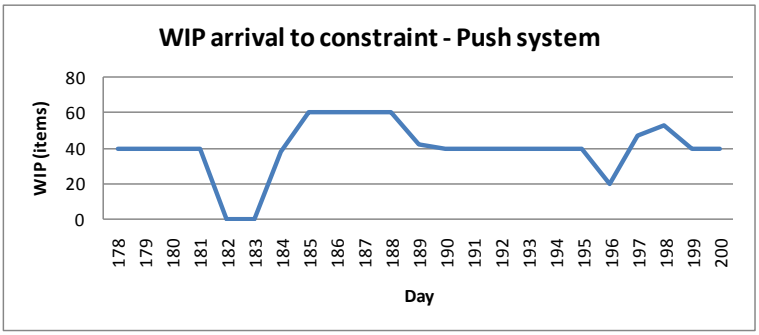


Figure 6.28: Zoom on Push system – WIP arrival to constraint machine

Figure 6.29 represents the distribution of the items cycle time for all release strategies (note the different scales applied for Push and CONFLOW). It can be seen that for CONFLOW Option 1, 2 and 3 most of items are grouped but few items have a much higher cycle time. While, for the push system cycle times are more evenly spread, therefore the cycle time standard deviation is higher (by as much as 53% for capacity 2) than in CONFLOW (Table 6.9 to Table 6.12).

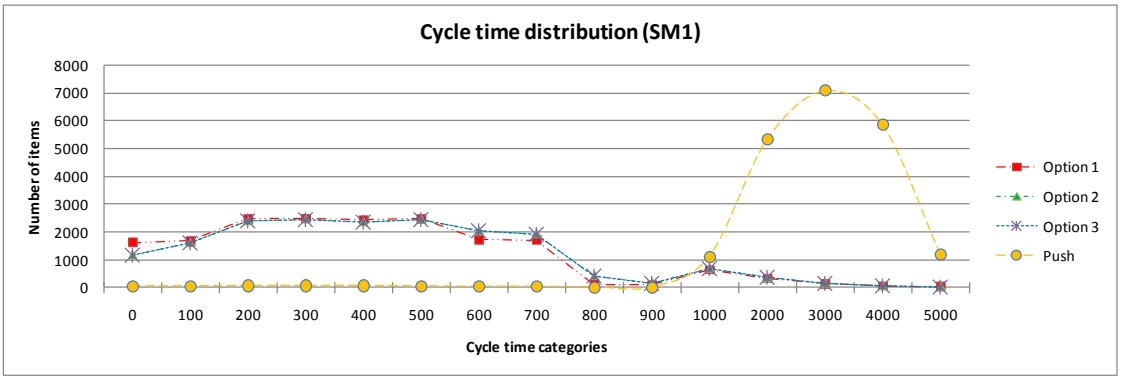


Figure 6.29: Cycle time distribution in simulation model 1 for all release strategies

The push system's results can be compared with those obtained with the simulation model called one buffer one machine (APPENDIX - E). Their setups are similar, both are push systems and do not control the release. Let's review some few results from Appendix E.3. First, if a machine is always kept busy — utilization is 1 — then the inter-departure time is constant and the variability is nil. Second, if the loading is really

low, the release time is slower than the machine process time, then there is no queue but the machine will transfer the inter-arrival time variability to inter-departure time variability. This is another approach to explain why CONFLOW option 1, 2 and 3 cycle time coefficient of variation is higher than the push system. The release rate is controlled to reduce the queue, but it results in high cycle time coefficient of variation. In a real factory, it's very difficult to balance the relationship of queue time, coefficient of variation and utilization. If a low coefficient of variation is targeted, machines cannot be starved then high queue time will appear. If a low queue time is targeted then high coefficient of variation will appear because machine will be idle sometimes. This un-regular feeding will increase the variations in the production line.

So which one is best? Each production manager has to decide in function of his factory objectives and his customers' demand. Reducing cycle time and WIP levels greatly improves the running cost and the predictability of the line, but reduced throughput increase the lead times for customers' delivery.

Simulation Model 2 (SM2): Availability and Batch Process Machine 1, Constraint Machine 2

Results of model 2 (Figure 6.30 to Figure 6.36) are very similar to those of model 1. To avoid repetition in the analysis only the conclusions are given in this chapter. Detailed result tables can also be found in F.1.1.

Model 2 results also show that CONFLOW option 1, 2 and 3 have a slower throughput than the push system by approximately 18%. The push system needs approximately 25

days at capacity 1 (36231 min / [60×24]) to produce 1000 items. But CONFLOW option 1, 2 and 3, need five days more than the push system to produce 1000 items.

For capacity 1, the mean cycle time is improved by as much as 81%, its standard deviation by 40% and the WIP number by 85%. For capacity 4, they are still respectively at 50%, 40% and 60%. The cycle time standard deviation behaves identically than in model 1.

This simulation was testing the release strategies when a batch machine is involved. Indeed previous batching simulations have shown the importance of properly sequencing the release of items for a batch machine. So do the results, seen for simulation model 1, still apply when a batch is involved? The answer is yes. Despite the batch machine, CT and WIP level are considerably improved. On the down side, throughput is reduced due to the starvation of the constraint.

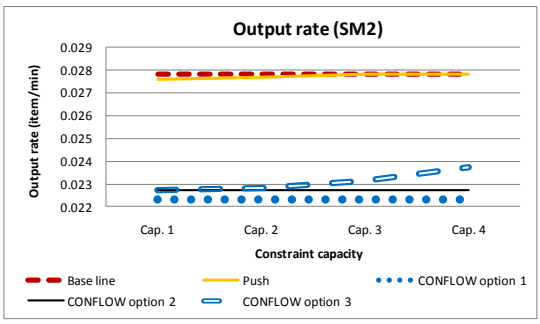


Figure 6.30: Simulation model 2 output rate for all release strategies

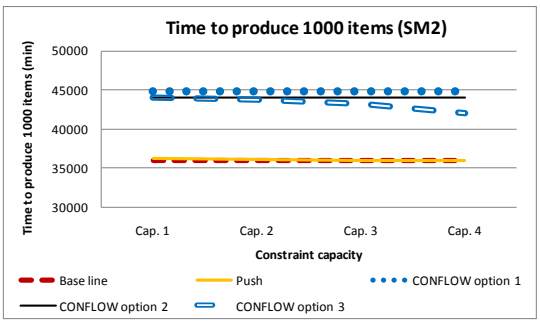


Figure 6.31: Time to produce 1000 items in simulation model 2 for all release strategies

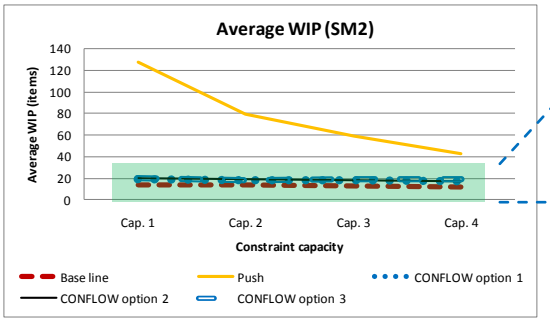


Figure 6.32: Average WIP in simulation model 2 for all release strategies

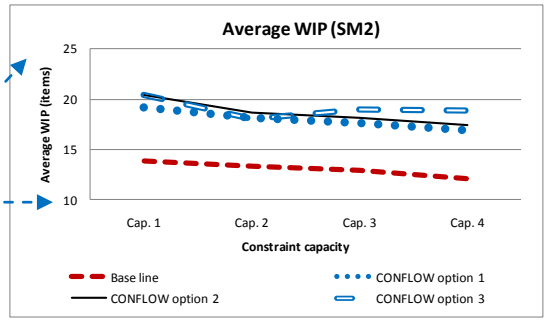
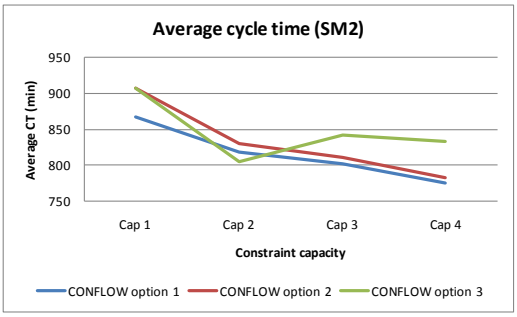
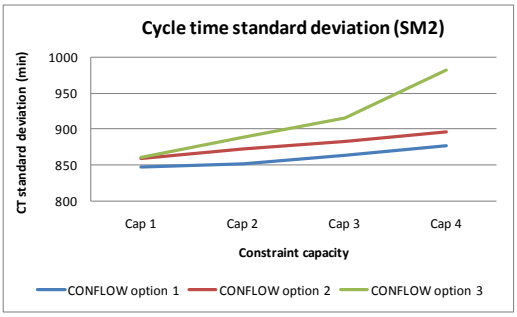


Figure 6.33: Zoom on baseline, CONFLOW Option 1, 2 and 3. Average WIP in simulation model 2



(a)



(b)

Figure 6.34: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

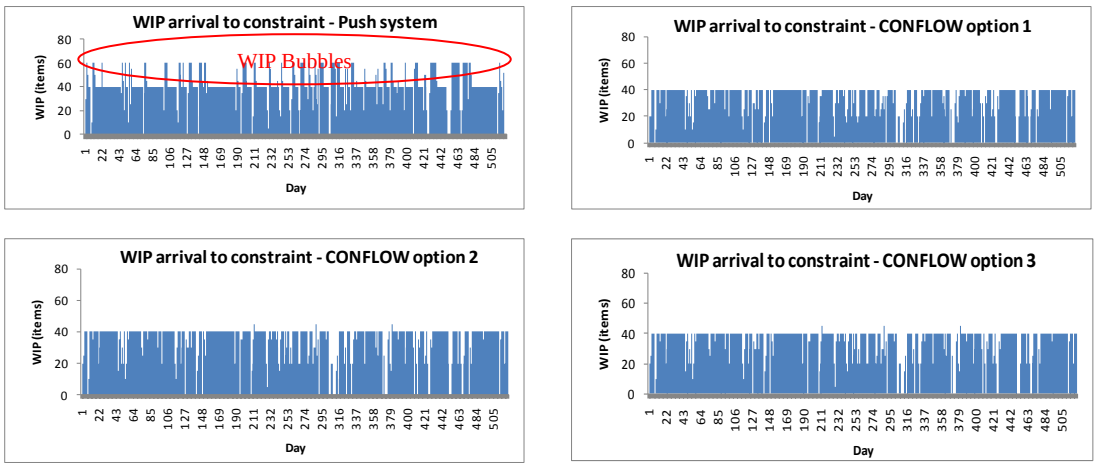


Figure 6.35: WIP arrival to constraint machine for all release strategies in SM2 model

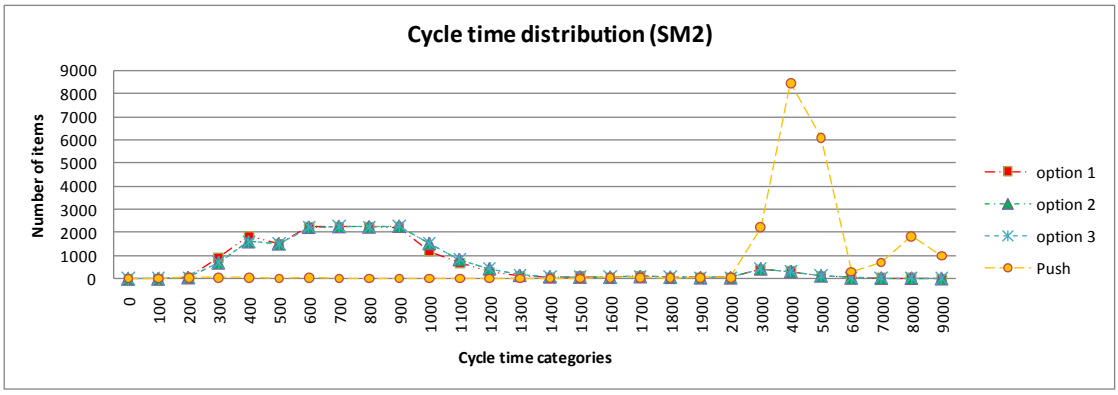


Figure 6.36: Cycle time distribution in simulation model 2 for all release strategies

Simulation Model 3 (SM3): Parallel Process Simulation Model with Availability Operation 1 and Constraint Operation 2

Results of model 3 (Figure 6.37 to Figure 6.43) are similar to those of model 1. To avoid repetition in the analysis only the conclusions are given in this chapter. Note that the four release strategies show much less variation compared to the baseline for the reason explained in the experiment design (p154). Detailed result tables can also be found in F.1.2.

Model 3 results show that CONFLOW option 1, 2 and 3 have a slower throughput than the push system. The push system needs approximately 25 days (36009 min / [60×24])

to produce 1000 items. CONFLOW Option 1, 2 and 3 need several hours more than the push system to produce 1000 items.

For capacity 1, the mean cycle time is improved by as much as 20%, its standard deviation by 8% and the WIP number by 22%. For capacity 4, they are respectively at 1%, 1% and 4%. The cycle time standard deviation behaves identically than in model 1.

Results show similar behaviors than in simulation model 1. Nevertheless as expected, there are fewer differences between the push system, CONFLOW option 1, 2 and 3. Increasing the number of machines in an operation not only increases the capacity of the operation, it also reduces the variability of the availability. Thus it minimizes the WIP bubble effect.

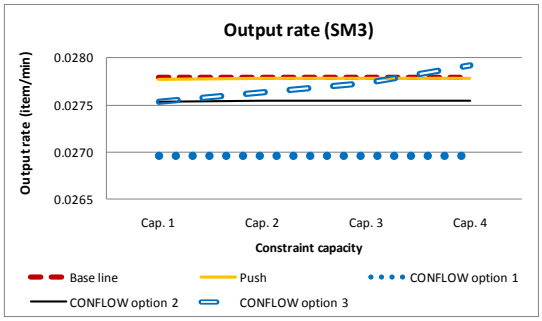


Figure 6.37: Simulation model 3 output rate for all release strategies

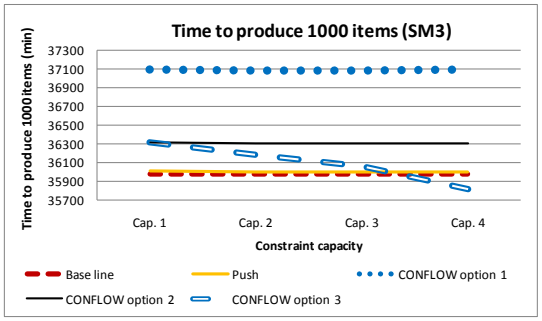


Figure 6.38: Time to produce 1000 items in simulation model 3 for all release strategies

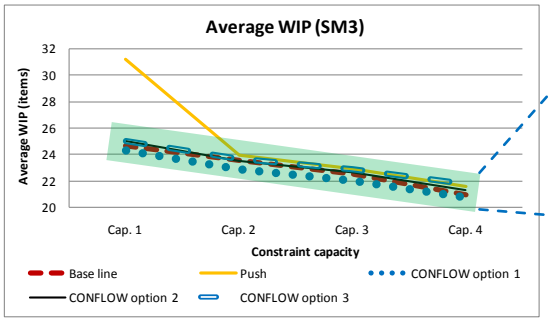


Figure 6.39: Average WIP in simulation model 3 for all release strategies

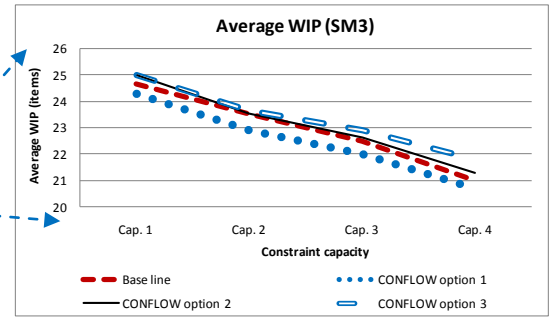
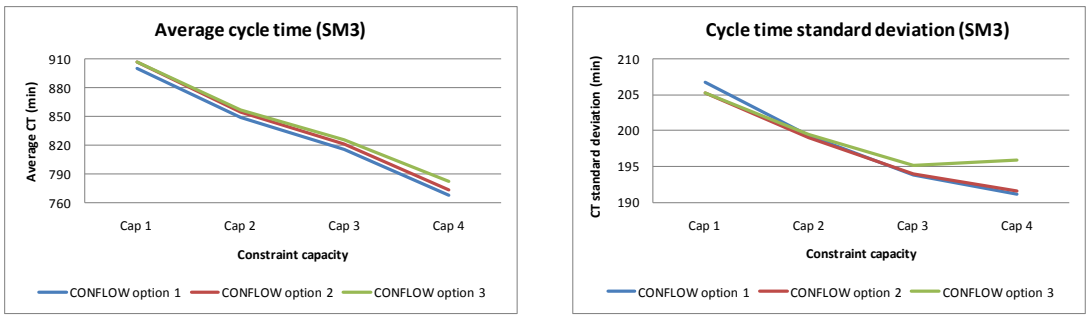


Figure 6.40: Zoom on baseline, CONFLOW Option 1, 2 and 3. Average WIP in simulation model 3



(a) (b)

Figure 6.41: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

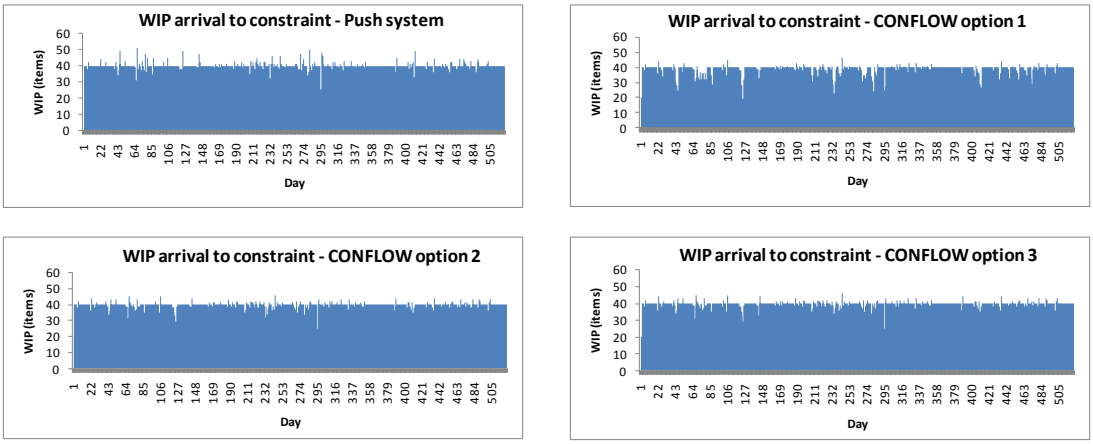


Figure 6.42: WIP arrival to constraint operation for all release strategies in SM3 model

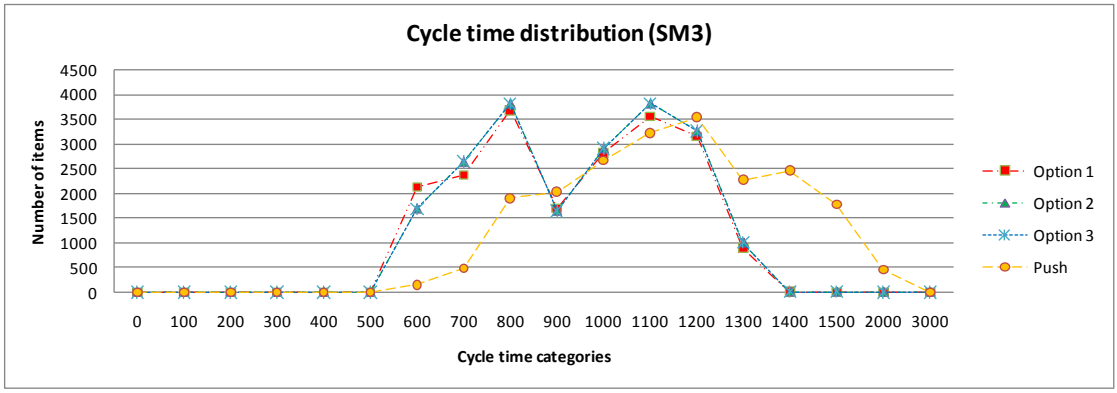


Figure 6.43: Cycle time distribution in simulation model 3 for all release strategies

6.4.2 Scenario 2: 5-Stage Serial Line with Constraint and Downtime (SM4)

Results of scenario 2 (Figure 6.44 to Figure 6.50) are very similar to those of scenario 1 simulation model 1. To avoid repetition in the analysis only the conclusions are given in this chapter. Detailed result tables can also be found in F.2.

Scenario 2 results also show that CONFLOW option 1, 2 and 3 have a slower throughput than the push system by approximately 13%. The push system needs approximately 25 days at capacity 1 (36166 min / [60×24]) to produce 1000 items. But CONFLOW option 1, 2 and 3 need five days more than the push system to produce 1000 items.

For capacity 1, the mean cycle time is improved by as much as 84%, its standard deviation by 30% and the WIP number by 86%. For capacity 4, they are still respectively at 57%, 45% and 50%. The cycle time coefficient of variation behaves identically than in scenario 1 model 1.

The results show only negligible differences with simulation model 1. Therefore, adding high capacity machines before and after the pair (availability tool – constraint tool) does not affect the results.

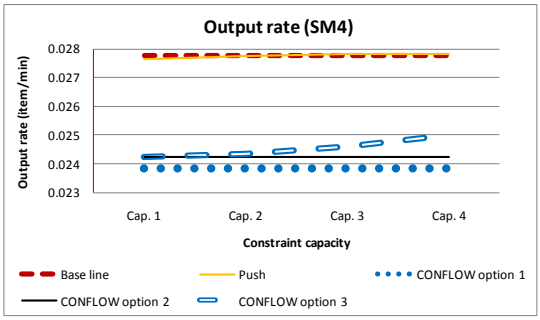


Figure 6.44: Simulation model 4 output rate for all release strategies

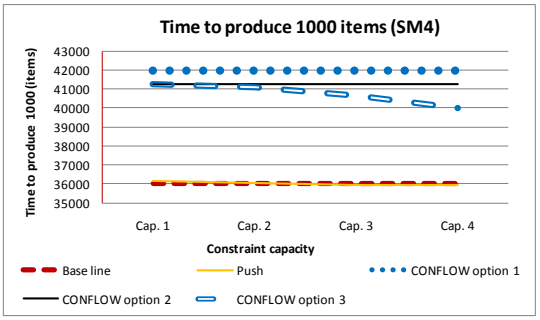


Figure 6.45: Time to produce 1000 items in simulation model 4 for all release strategies

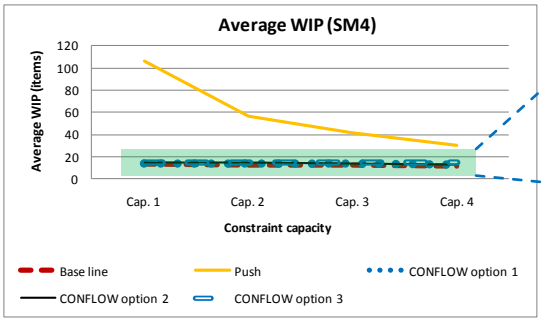


Figure 6.46: Average WIP in simulation model 4 for all release strategies

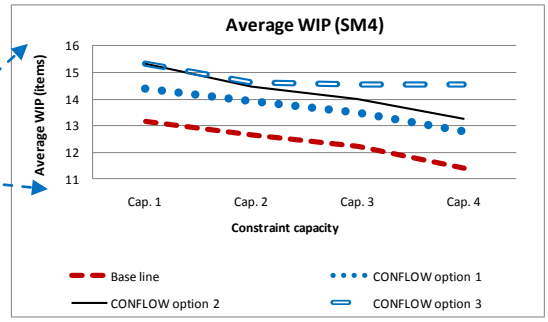
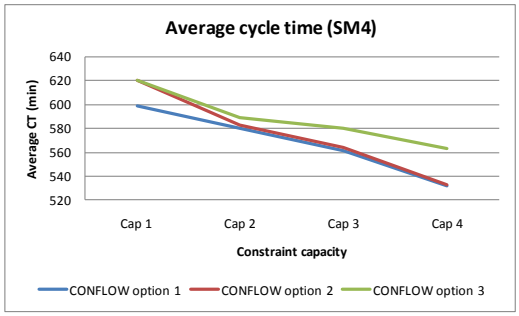
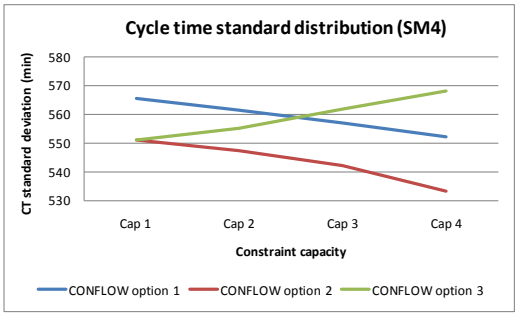


Figure 6.47: Zoom on baseline, CONFLOW Option 1, 2 and 3. Average WIP in simulation model 4



(a)



(b)

Figure 6.48: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

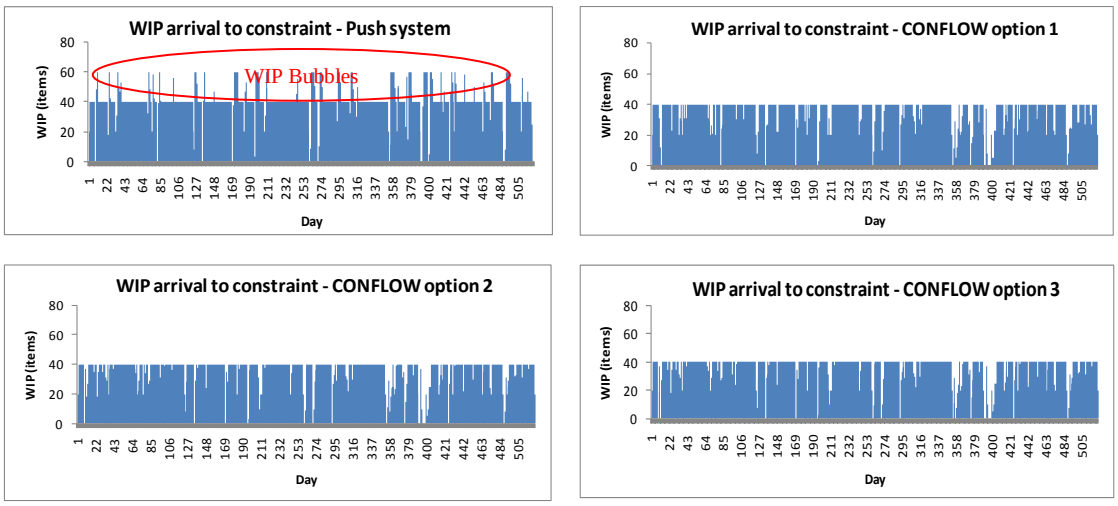


Figure 6.49: WIP arrival to constraint machine for all release strategies in SM4 model

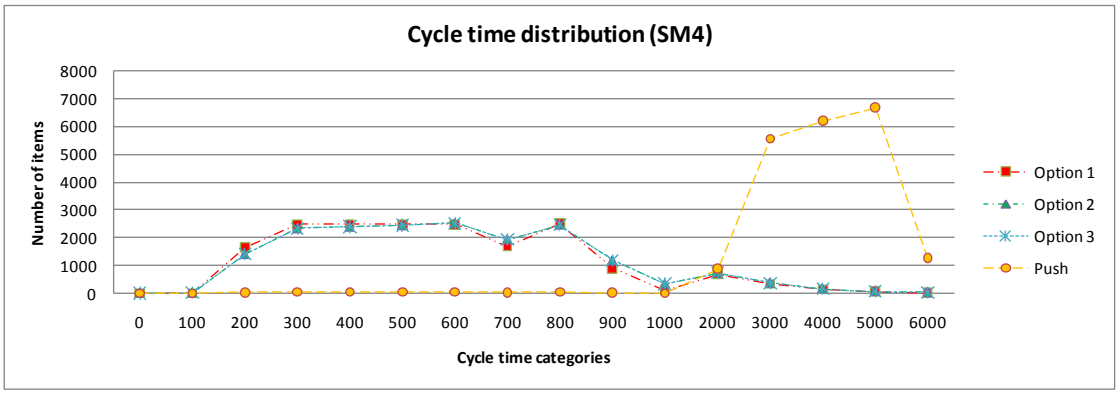


Figure 6.50: Cycle time distribution in simulation model 4 for all release strategies

6.4.3 Scenario 3: 5-Stage Serial Line with Batch, Downtime and Constraint (SM5)

The influence of the batch, constraint and downtime positions is tested through the simulation of the 6 possible order sequences.

Simulation model 5 BTC (Order: Batch, Tool availability and Constraint sequence)

Results of scenario 3 model 5 BTC (Figure 6.51 to Figure 6.57) are very similar to those of scenario 1 simulation model 1. To avoid repetition in the analysis only the conclusions are given in this chapter. Detailed result tables can also be found in F.3.1.

These results show that CONFLOW option 1, 2 and 3 have a slower throughput than the push system by approximately 14%. The push system needs approximately 25 days at capacity 1 (36156 min / [60×24]) to produce 1000 items. But CONFLOW option 1, 2 and 3 need four days more than the push system to produce 1000 items.

For capacity 1, the mean cycle time is improved by as much as 78%, its standard deviation by 42% and the WIP number by 82%. For capacity 4, they are still respectively at 38%, 38% and 48%. Min CT still corresponds to the total processing time without queuing in the buffers. Max CT is slightly higher in Push than CONFLOW; however it still corresponds to the longest machine failure.

The results obtained in simulation model 5 BTC are similar to the results of simulation model 1. Adding high capacity machines, including batch machine, does not fundamentally change the results obtained when considering only the pair (availability – constraint).

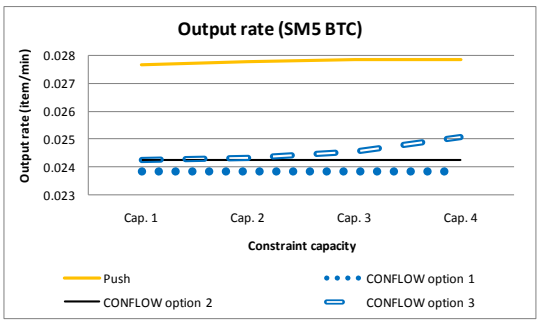


Figure 6.51: Simulation model 5 BTC output rate for all release strategies

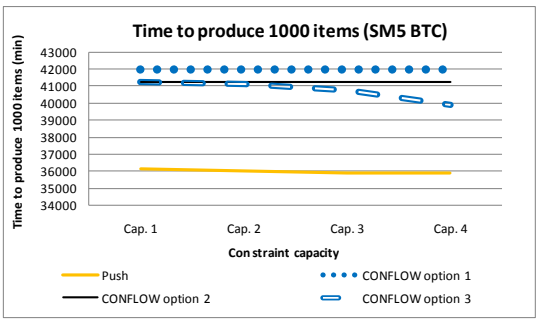


Figure 6.52: Time to produce 1000 items in simulation model 5 BTC for all release strategies

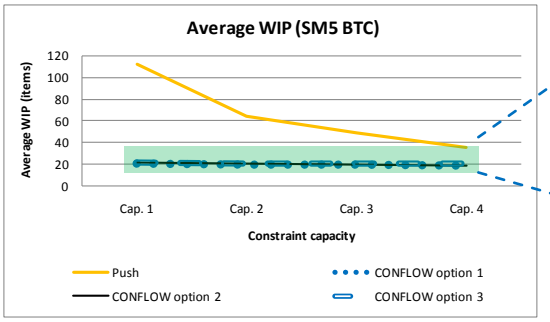


Figure 6.53: Average WIP in simulation model 5 BTC for all release strategies

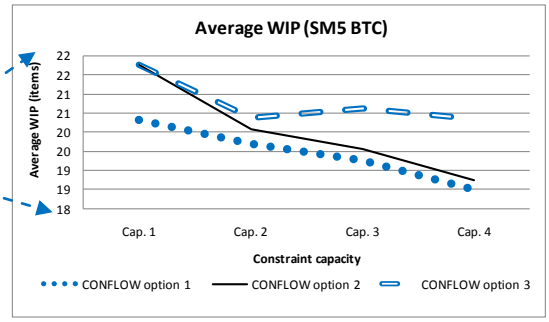
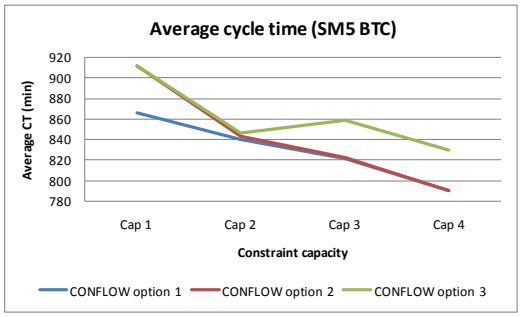
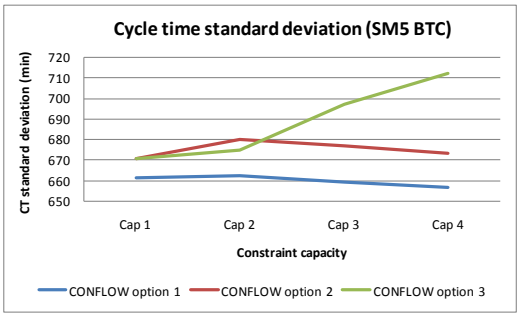


Figure 6.54: Zoom on CONFLOW Option 1, 2 and 3. Average WIP in simulation model 5 BTC



(a)



(b)

Figure 6.55: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

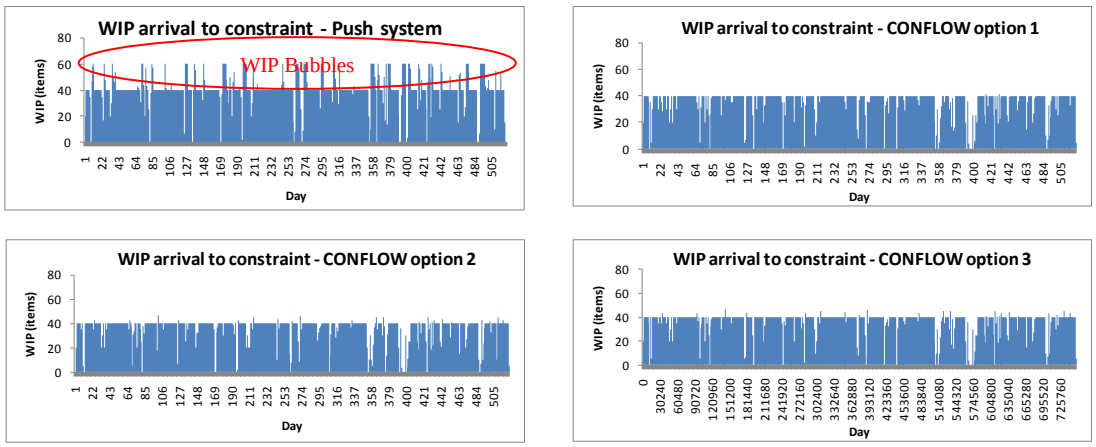


Figure 6.56: WIP arrival to constraint machine for all release strategies in SM5 BTC model

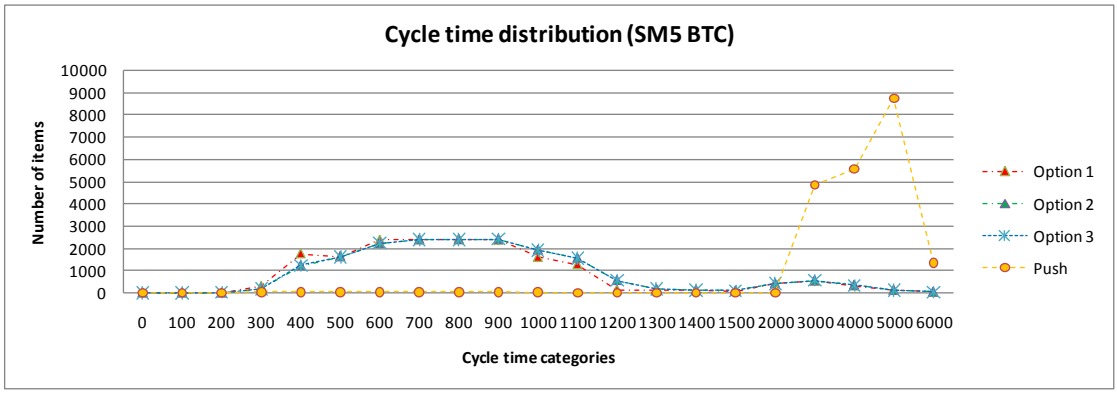


Figure 6.57: Cycle time distribution in simulation model 5 BTC for all release strategies

Simulation Model 5 BCT (Order: Batch, Constraint and Tool Availability Sequence)

In this model (order: Batch, Constraint and Tool availability) there is less difference between the four release strategies (Table 6.13 to Table 6.16 and Figure 6.58). Indeed, the push system's cycle time is lower than in all previous simulation models. It means that the downtime has less impact on the production line.

Table 6.13: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	1035.5707	904.6079	252	6018.7115
CONFLOW option 1	862.0145	677.8931	252	5994.7115
CONFLOW option 2	860.3791	690.8072	252	5994.7115
CONFLOW option 3	860.6285	686.1600	252	5994.7115

Table 6.14: Performance of the policies with respect to cycle time for Capacity 2

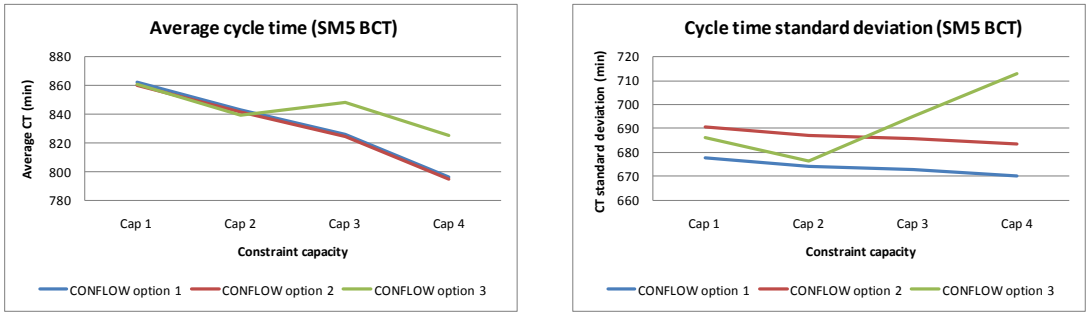
	Mean	Stdev	Min	Max
Push	1015.7581	901.4726	250.290	6018.7115
CONFLOW option 1	842.7529	674.0112	250.290	5994.7115
CONFLOW option 2	841.1177	687.0370	250.290	5994.7115
CONFLOW option 3	839.2012	676.4298	250.290	5994.7115

Table 6.15: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	999.4452	899.7499	248.730	5994.7115
CONFLOW option 1	825.9194	672.8233	248.730	5994.7115
CONFLOW option 2	824.3943	685.7764	248.730	5994.7115
CONFLOW option 3	848.1168	695.1769	248.730	5994.7115

Table 6.16: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	971.3575	898.5309	246	5994.7115
CONFLOW option 1	795.9812	670.3731	246	5994.7115
CONFLOW option 2	794.5814	683.3734	246	5994.7115
CONFLOW option 3	824.9310	712.9964	246	5994.7115



(a) (b)

Figure 6.58: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

This is explained by the fact that the constraint machine precedes the machine affected by downtime. Indeed, let's take the analogy of a pipe and a grain of sand in Figure 6.59.

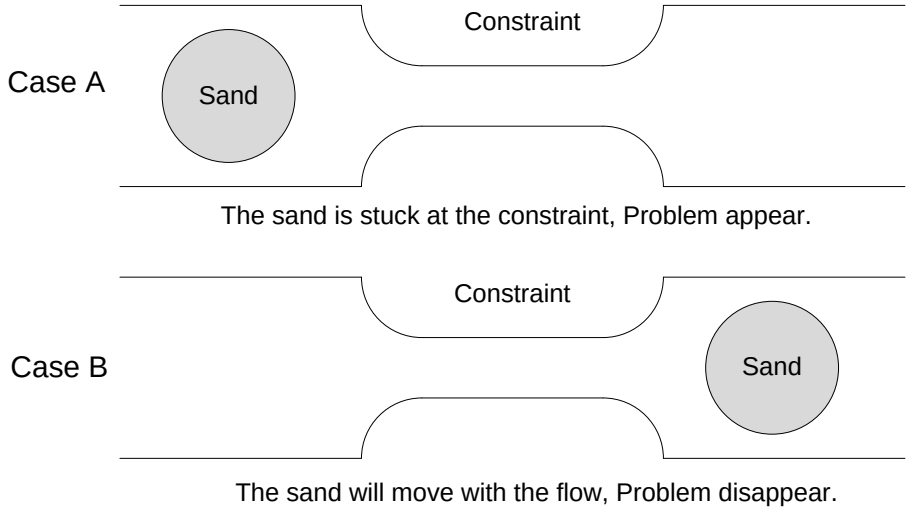


Figure 6.59: Analogy of sand flow in a pipe

In Figure 6.59, case A is in the same situation as when the constraint machine is behind machine affected by downtime. Downtime creates a queue in front of the tool availability machine which result later in a huge release (grain of sand) when the machine is back up. When this grain of sand reaches the constraint machine, it becomes stuck and a problem appears, resulting in a long cycle time.

On the other hand, case B is in the same situation than when the constraint machine is in front of the tool availability machine. Whatever the downtime of the tool availability machine, the grain of sand is flushed by the flow down the pipe; it is never blocked by the constraint machine. The flow of items arriving to the constraint is stable, no WIP bubbles can be seen (Figure 6.60). In this situation, the flow is only limited and fully controlled by the constraint machine.

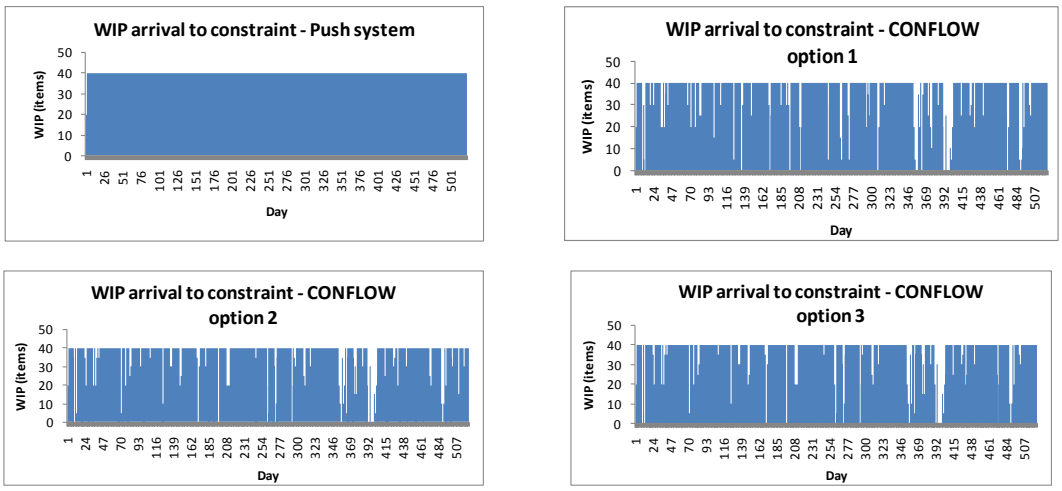


Figure 6.60: WIP arrival to constraint machine for all release strategies in SM5 BCT model

This is the reason why the push system, CONFLOW option 1, 2 and 3 only show smaller differences for all capacities in cycle time (Table 6.13 to Table 6.16), output rate (Figure 6.61) and WIP (Figure 6.63) than all previous models.

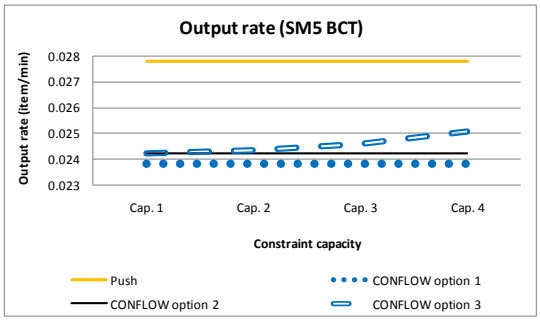


Figure 6.61: Simulation model 5 BCT output rate for all release strategies

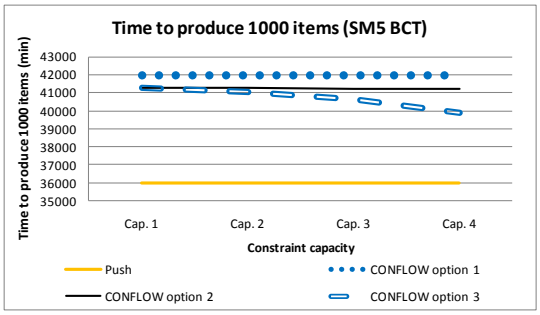


Figure 6.62: Time to produce 1000 items in simulation model 5 BCT for all release strategies

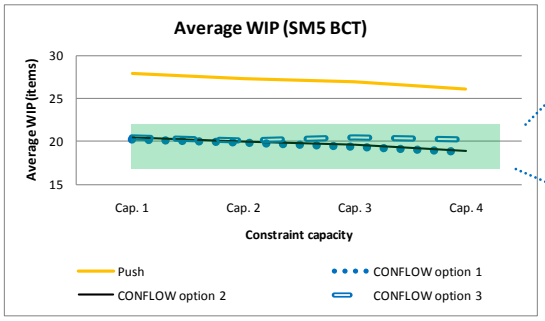


Figure 6.63: Average WIP in simulation model 5 BCT for all release strategies

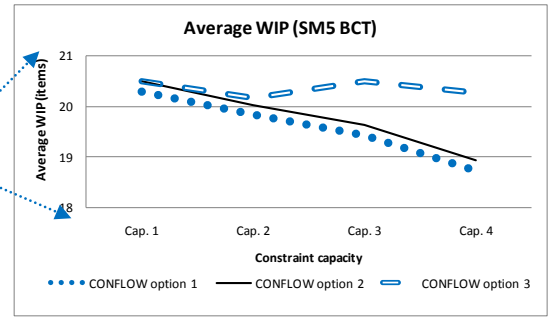


Figure 6.64: Zoom on CONFLOW Option 1, 2 and 3. Average WIP in simulation model 5 BCT

Figure 6.65 represents the distribution of the items cycle time for all release strategies. It can be seen again that differences between release strategies have been greatly reduced.

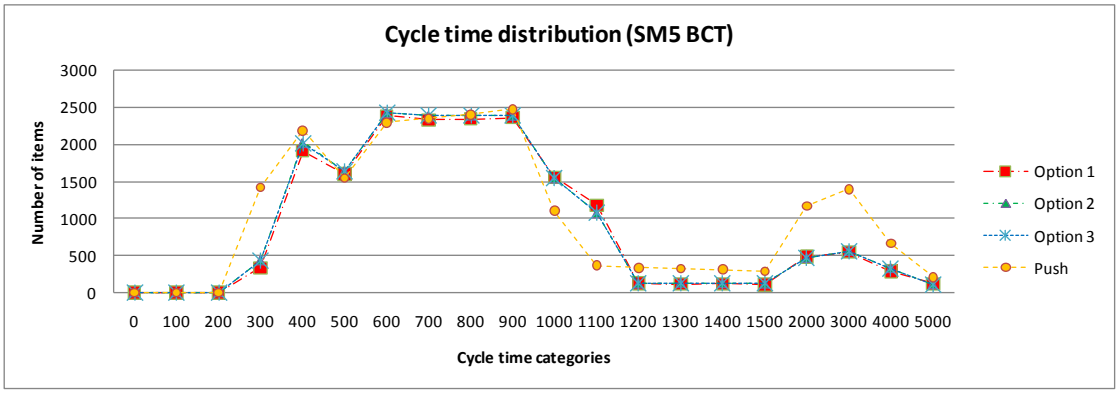


Figure 6.65: Cycle time distribution in simulation model 5 BCT for all release strategies

In those conditions, CONFLOW release strategies (CONFLOW option 1, 2 and 3) still improve cycle time and WIP level at the cost of a decrease in throughput and higher variability. But the gain and loss are much smaller as the impact of downtime has been reduced. These results confirm the TOC approach. The line only needs to be monitored down to the constraint. Events happening after the constraint (for example downtime) have much less impact.

Simulation Model 5 TBC (Order: Tool Availability, Batch and Constraint Sequence)

Results of scenario 3 model 5 TBC (Figure 6.66 to Figure 6.72) are very similar to those of scenario 1 model 1. To avoid repetition in the analysis only the conclusions are given in this chapter. Detailed result tables can also be found in F.3.2.

These results show that CONFLOW option 1, 2 and 3 have a slower throughput than the push system by approximately 14%. The push system needs approximately 25 days at

capacity 1 (36166 min / [60×24]) to produce 1000 items. But CONFLOW option 1, option 2 and option 3, need four days more than push system to produce 1000 items.

For capacity 1, the mean cycle time is improved by as much as 79%, its standard deviation by 49% and the WIP number by 82%. For capacity 4, they are still respectively at 39%, 42% and 48%. The cycle time standard deviation behaves identically than in scenario 1 model 1.

In this case, availability is again before the constraint. This is Case A in Figure 6.59. So the results are similar to simulation model 5 BTC or simulation model 1.

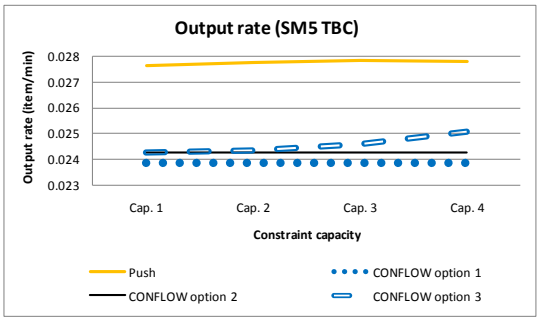


Figure 6.66: Simulation model 5 TBC output rate for all release strategies

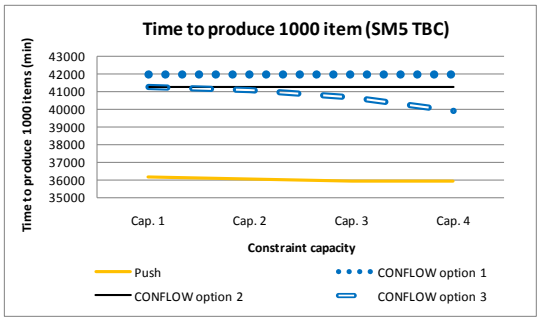


Figure 6.67: Time to produce 1000 items in simulation model 5 TBC for all release strategies

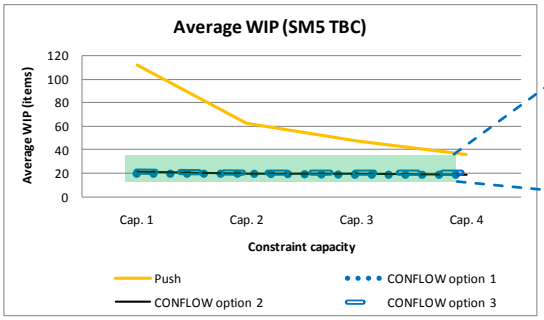


Figure 6.68: Average WIP in simulation model 5 TBC for all release strategies

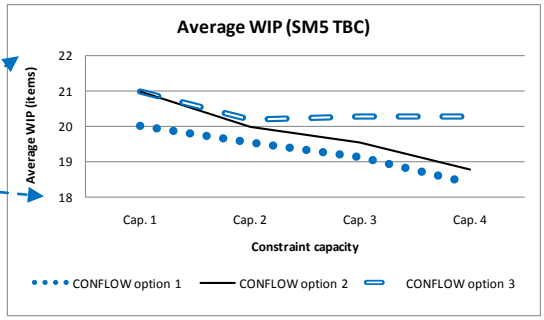
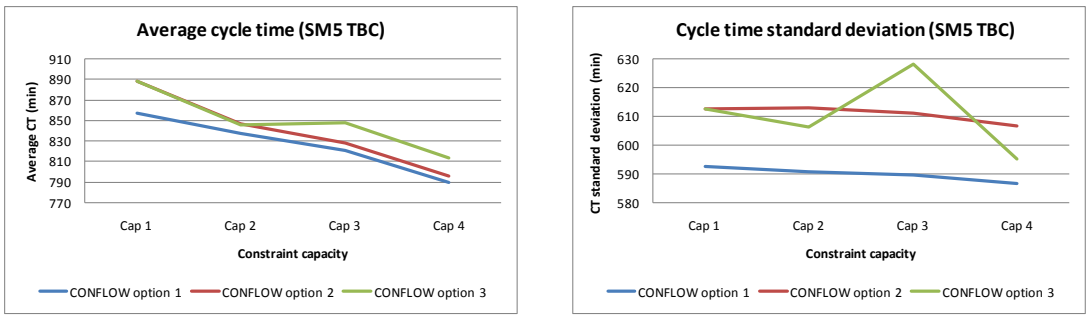


Figure 6.69: Zoom on CONFLOW Option 1, 2 and 3. Average WIP in simulation model 5 TBC



(a) (b)

Figure 6.70: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

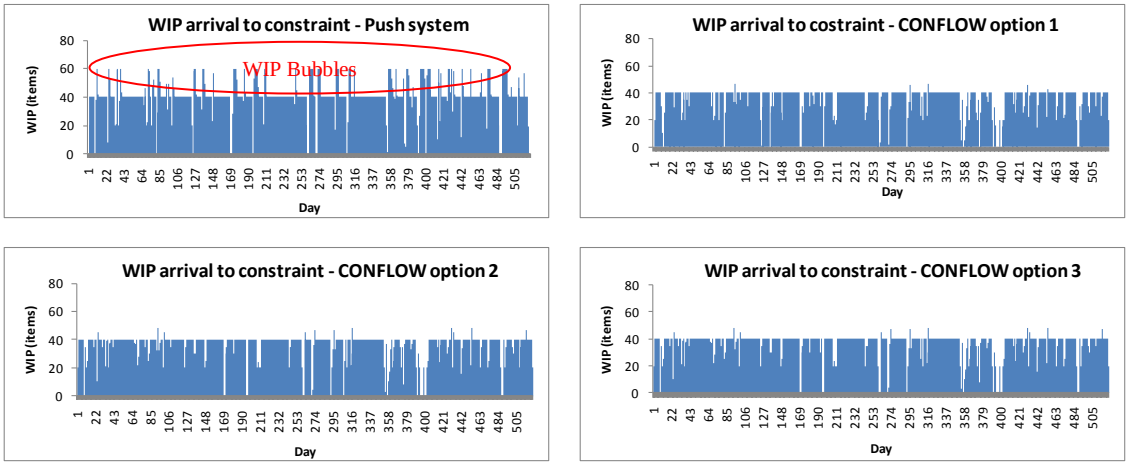


Figure 6.71: WIP arrival to constraint machine for all release strategies in SM5 TBC model

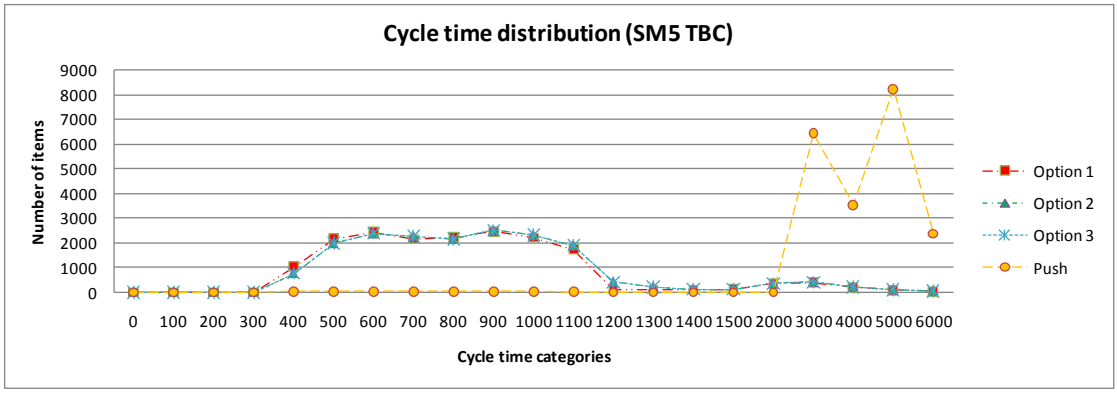


Figure 6.72: Cycle time distribution in simulation model 5 TBC for all release strategies

Simulation Model 5 TCB (Order: Tool Availability, Constraint and Batch Sequence)

Results of scenario 3 model 5 TCB (Figure 6.73 to Figure 6.79) are very similar to those of scenario 1 model 1. To avoid repetition in the analysis only the conclusions are given in this chapter. Detailed result tables can also be found in F.3.3.

These results show that CONFLOW option 1, 2 and 3 have a slower throughput than the push system by approximately 14%. The push system needs approximately 25 days at capacity 1 (36166 min / [60×24]) to produce 1000 items. But CONFLOW option 1, option 2 and option 3, need four days more than push system to produce 1000 items.

For capacity 1, the mean cycle time is improved by as much as 79%, its standard deviation by 44% and the WIP number by 82%. For capacity 4, they are still respectively at 39%, 44% and 48%. The cycle time standard deviation behaves identically than in scenario 1 model 1.

In this case, availability is again before the constraint. This is Case A in Figure 6.59. So the results are similar to simulation model 5 BTC or simulation model 1.

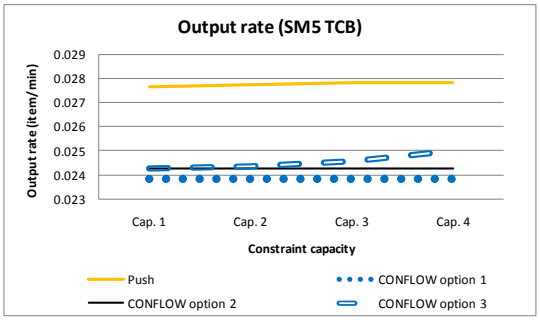


Figure 6.73: Simulation model 5 TCB output rate for all release strategies

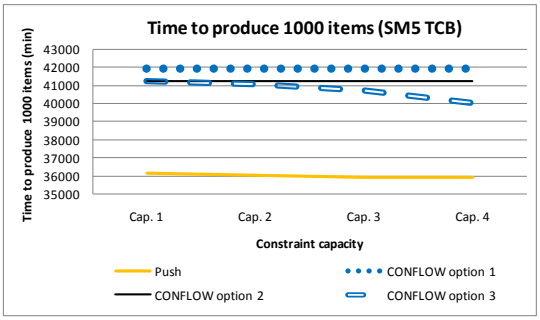


Figure 6.74: Time to produce 1000 items in simulation model 5 TCB for all release strategies

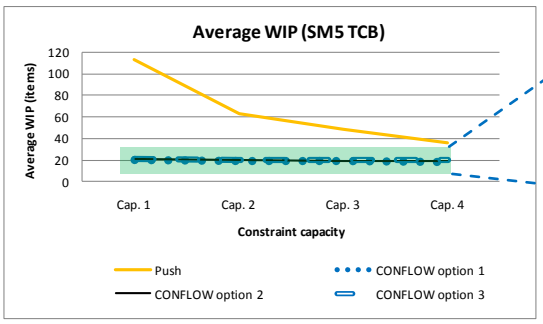


Figure 6.75: Average WIP in simulation model 5 TCB for all release strategies

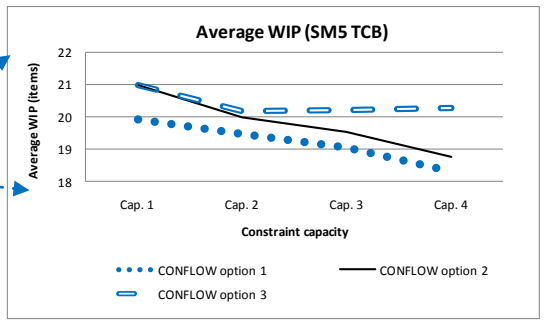
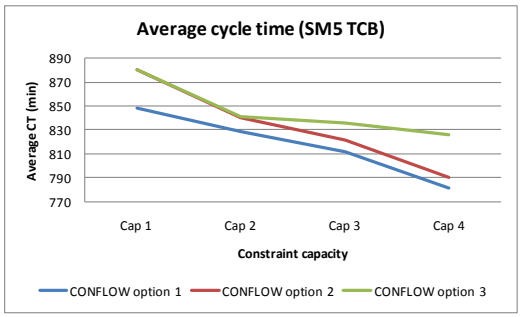
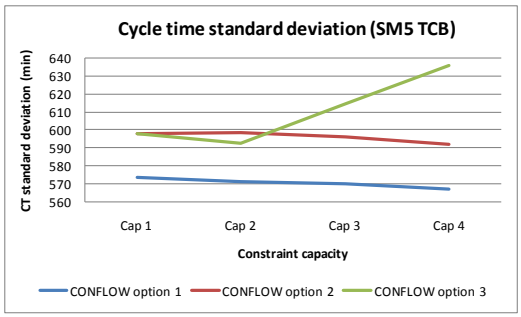


Figure 6.76: Zoom on CONFLOW Option 1, 2 and 3. Average WIP in simulation model 5 TCB



(a)



(b)

Figure 6.77: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

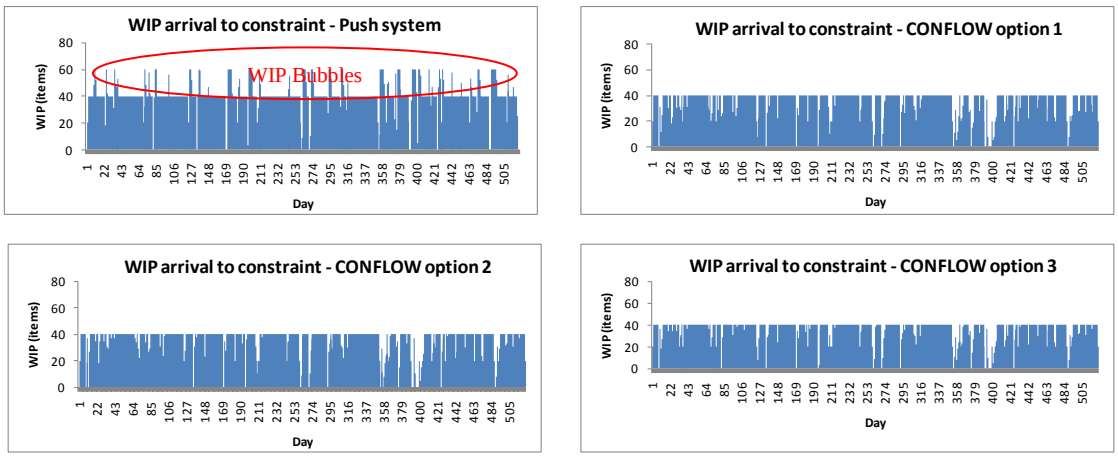


Figure 6.78: WIP arrival to constraint machine for all release strategies in SM5 TCB model

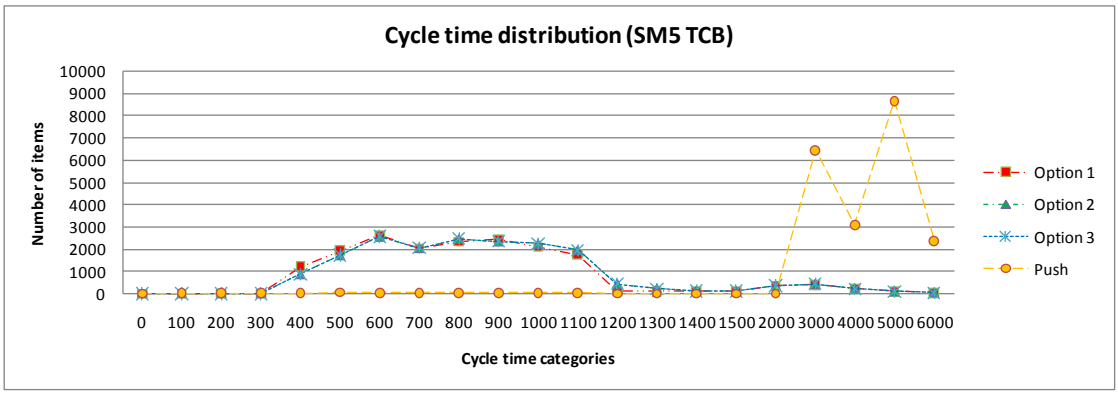


Figure 6.79**: Cycle time distribution in simulation model 5 TCB for all release strategies**

Simulation Model 5 CTB (Order: Constraint, Tool Availability and Batch Sequence)

Results of scenario 3 model 5 CTB (Figure 6.80 to Figure 6.86) are very similar to those of scenario 3 model 5 BCT. To avoid repetition in the analysis only the conclusions are given in this section. Detailed result tables can also be found in F.3.4.

The constraint machine precedes the machine affected by downtime which is a similar situation to Case B in Figure 6.59. The WIP bubbles due to downtime are created after the constraint machine. They can be processed by all following machines without delays

as they are all high capacity machines. There is no major queue forming. It means that the downtime has less impact on the production line.

Results are similar to the simulation model 5 BCT, CONFLOW release strategies (CONFLOW option 1, 2 and 3) still improve cycle time and WIP level at the cost of a decrease in throughput and higher variability. But the gain and loss are much smaller as the impact of downtime has been reduced. These results confirm the TOC approach. The line only needs to be monitored down to the constraint. Events happening after the constraint (for example downtime) have much less impact.

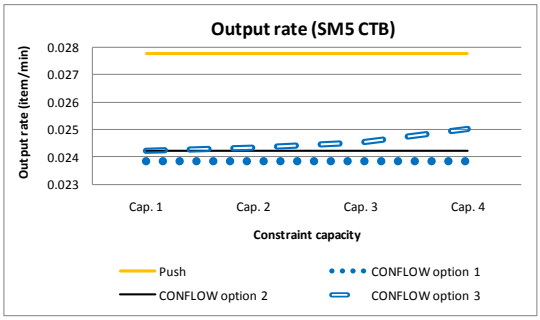


Figure 6.80: Simulation model 5 CTB output rate for all release strategies

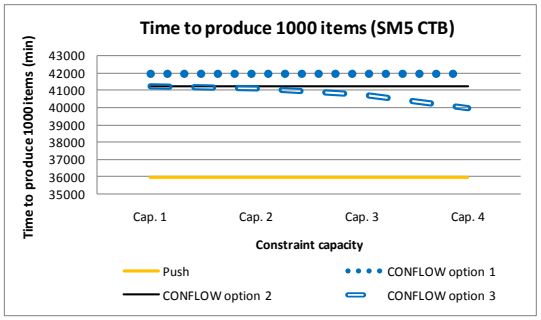


Figure 6.81: Time to produce 1000 items in simulation model 5 CTB for all release strategies

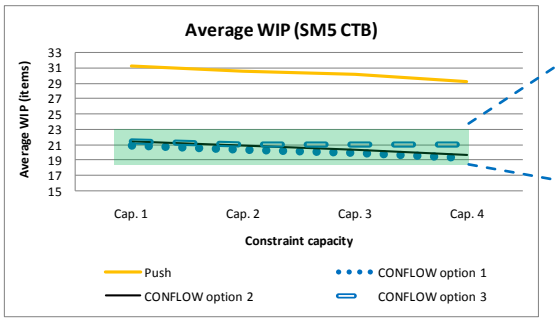


Figure 6.82: Average WIP in simulation model 5 CTB for all release strategies

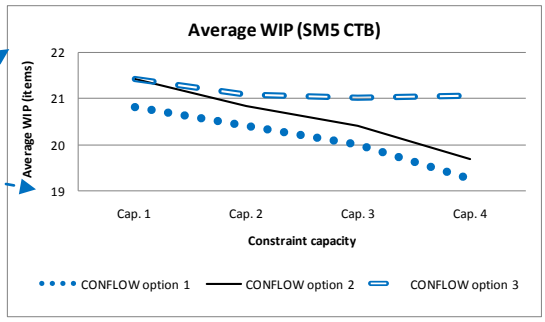
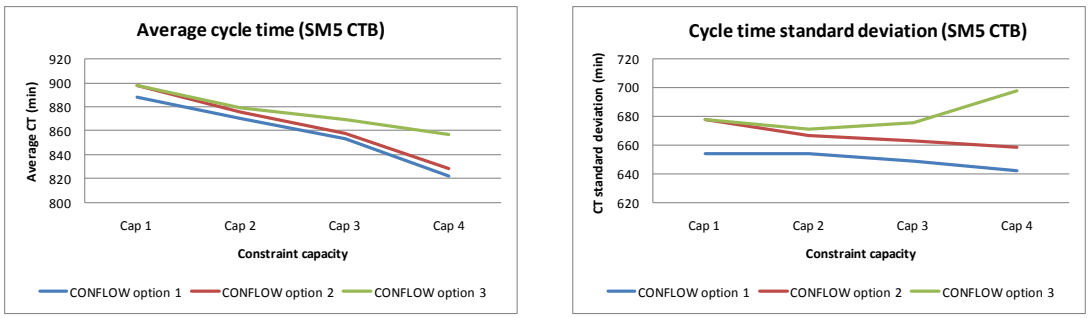


Figure 6.83: Zoom on CONFLOW Option 1, 2 and 3. Average WIP in simulation model 5 CTB



(a) (b)

Figure 6.84: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

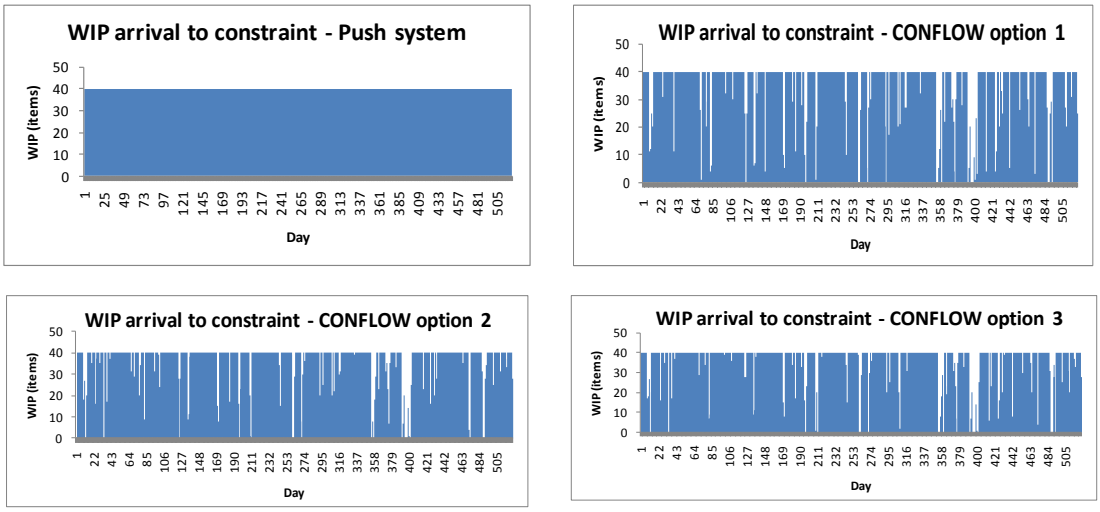


Figure 6.85: WIP arrival to constraint machine for all release strategies in SM5 CTB model

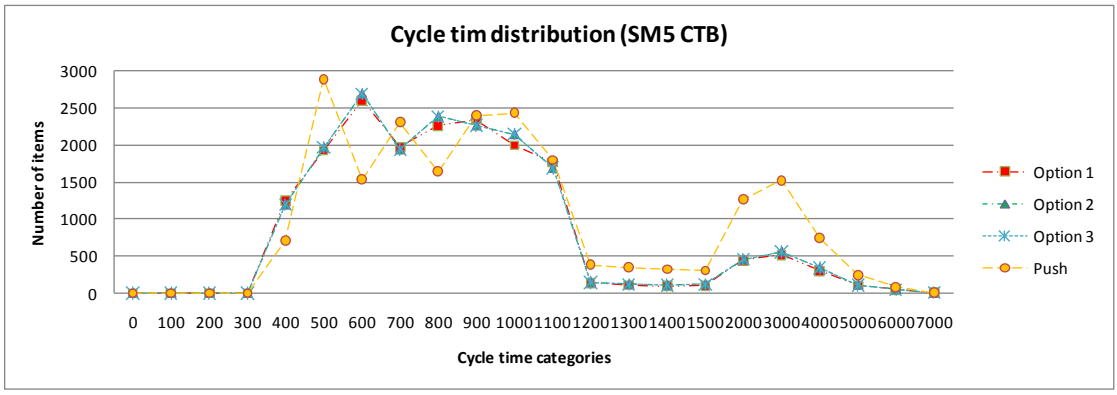


Figure 6.86: Cycle time distribution in simulation model 5 CTB for all release strategies

Simulation Model 5 CBT (Order: Constraint, Batch and Tool Availability Sequence)

Results of scenario 3 model 5 CBT (Figure 6.87 to Figure 6.93) are very similar to those of scenario 3 model 5 BCT. To avoid repetition in the analysis only the conclusions are given in this chapter. Detailed result tables can also be found in F.3.5.

The constraint machine precedes the machine affected by downtime which is a similar situation to Case B in Figure 6.59. The WIP bubbles due to downtime are created after the constraint machine. They can be processed by all following machines without delays as they are all high capacity machines. There is no major queue forming. It means that the downtime has less impact on the production line.

Results are similar to the simulation model 5 BCT, CONFLOW release strategies (CONFLOW option 1, 2 and 3) still improve cycle time and WIP level at the cost of a decrease in throughput and higher variability. But the gain and loss are much smaller as the impact of downtime has been reduced. These results confirm the TOC approach. The line only needs to be monitored down to the constraint. Events happening after the constraint (for example downtime) have much less impact.

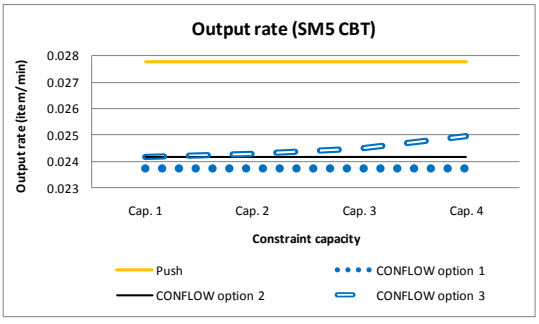


Figure 6.87: Simulation model 5 CBT output rate for all release strategies

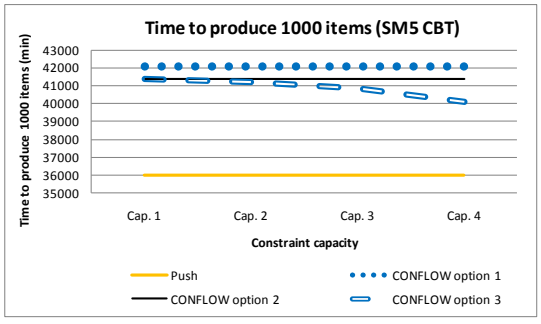


Figure 6.88: Time to produce 1000 items in simulation model 5 CBT for all release strategies

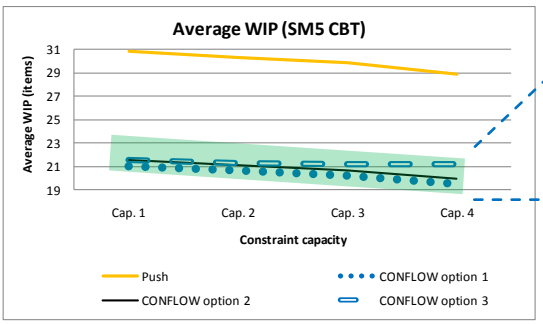


Figure 6.89: Average WIP in simulation model 5 CBT for all release strategies

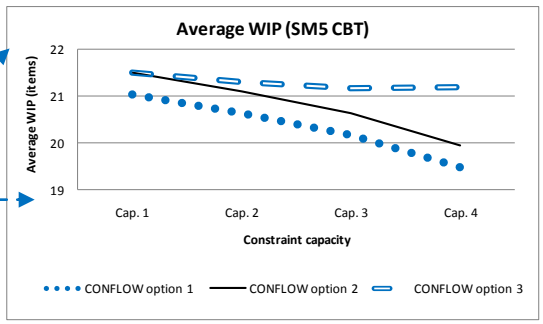
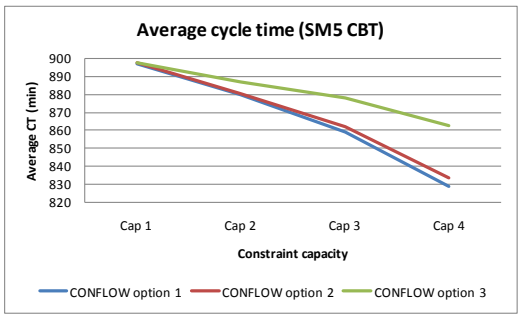
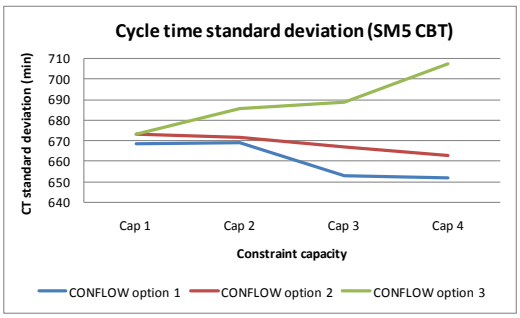


Figure 6.90: Zoom on CONFLOW Option 1, 2 and 3. Average WIP in simulation model 5 CBT



(a)



(b)

Figure 6.91: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

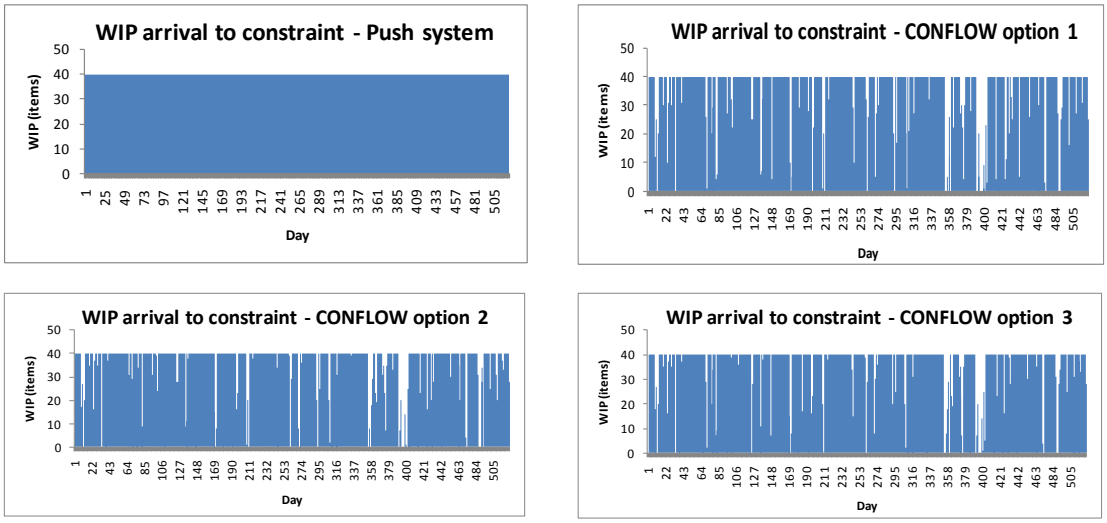


Figure 6.92: WIP arrival to constraint machine for all release strategies in SM5 CBT model

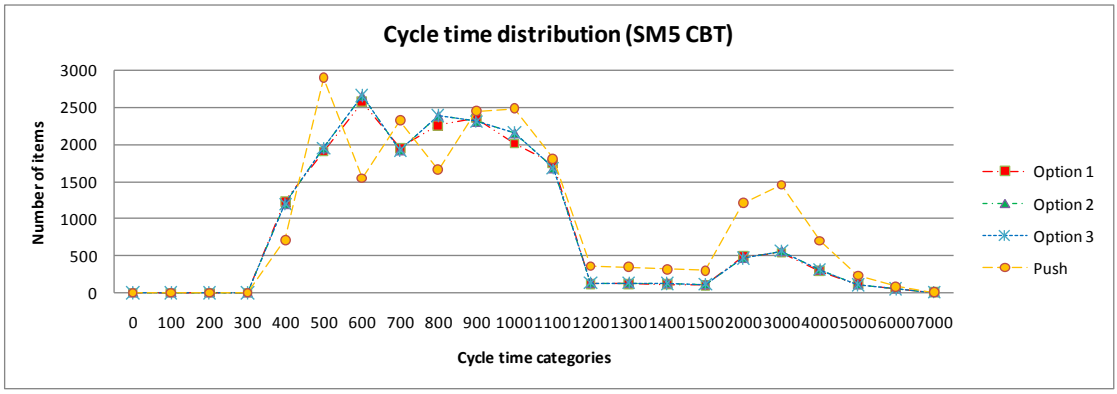


Figure 6.93: Cycle time distribution in simulation model 5 CBT for all release strategies

Simulation Model 5 – Six B/C/T Permutations Summary

Taking into account all the models and simulations, it can be deduced that:

Result 1. The most important parameter is the order in the line of the machine affected by downtime and the constraint machine. If the constraint is before an operation affected by downtime then, the cycle time is not affected. If the downtime is before the constraint then, cycle time is greatly affected by the WIP bubble created. Nevertheless, cycle time and WIP can be greatly improved by using a CONFLOW release policy and modulating the release as a function of the downtime.

Result 2. The number of machines in a production stage does not affect result 1.

Result 3. The number of high capacity production stages in the line does not affect result 1.

Result 4. The positions in the line of unreliable production stage and the constraint stage do not matter. Only their respective order matters.

Result 5. Batching does not modify result 1.

These results provide to line managers an operating procedure to improve the average cycle time and reduce the WIP on the production line. First, the constraint stage needs to be identified in the production line. Then, the availability of the production stages preceding the constraint stage should be monitored. Finally, the release of items in the line can be modulated using a CONFLOW release policy.

6.4.4 Scenario 4: Push and CONFLOW policies matched throughput

This scenario is based on the simulation model 5 BTC (order: batch, tool availability and constraint). In order to facilitate the comparison of Push and CONFLOW, the release rate of the push model has been reduced to 17 items per shift. In those conditions, the output rates of push and CONFLOW policies are similar (Figure 6.94 and Figure 6.95). The push policy has actually a slightly lower throughput. Simultaneously (Table 6.17 to Table 6.20), it shows higher cycle time: 40% for capacity 1 down to 25% for capacity 4. It shows higher cycle time standard deviation: 40% for capacity 1 down to 25% for capacity 4. It also shows higher WIP (Figure 6.96): 39% for capacity 1 down to 23% for capacity 4. It proves the better performance of CONFLOW in comparison to the push system.

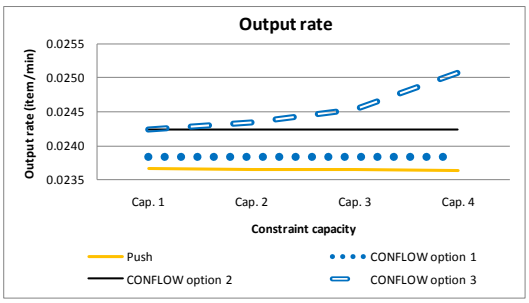


Figure 6.94: Output rate

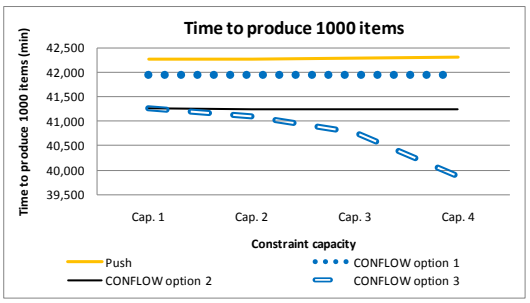


Figure 6.95: Time to produce 1000 items

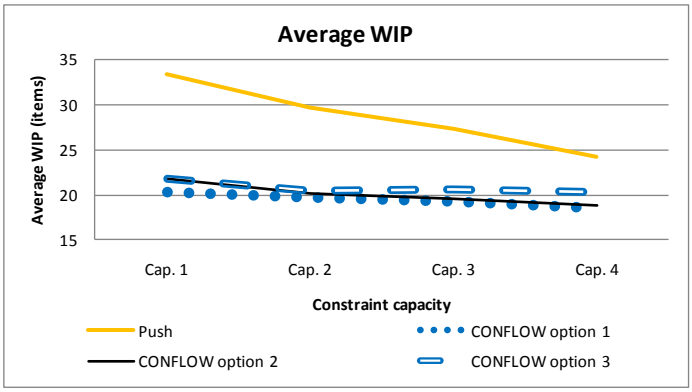


Figure 6.96: Average WIP

Table 6.17: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	1439.9331	1083.0587	252	5982.7115
CONFLOW option 1	865.6927	661.1602	252	5874.7115
CONFLOW option 2	911.3226	670.8861	252	5874.7115
CONFLOW option 3	911.3226	670.8861	252	5874.7115

Table 6.18: Performance of the policies with respect to cycle time for Capacity 2

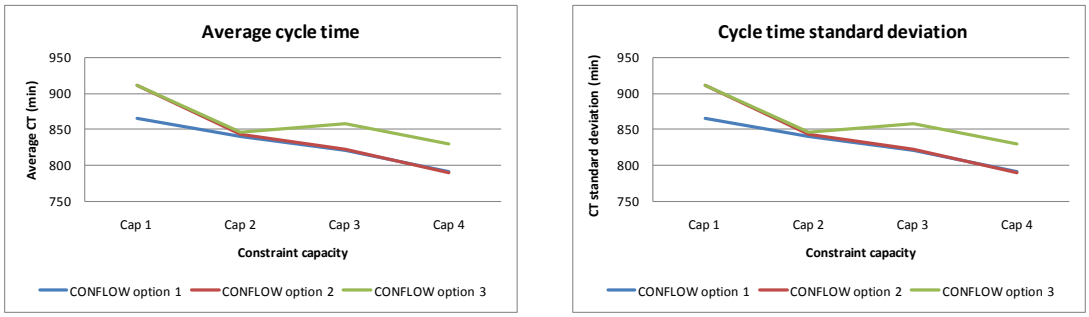
	Mean	Stdev	Min	Max
Push	1291.4572	1030.4781	250.290	5967.3215
CONFLOW option 1	839.9461	662.1305	250.290	5898.8315
CONFLOW option 2	843.4163	680.0887	250.290	5898.8315
CONFLOW option 3	846.4156	675.0765	250.290	5898.8315

Table 6.19: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	1190.406	984.7085	248.730	5953.2815
CONFLOW option 1	821.6120	659.4503	248.730	5887.8815
CONFLOW option 2	822.5528	677.0339	248.730	5887.8815
CONFLOW option 3	858.4838	696.9786	248.730	5887.8815

Table 6.20: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1059.026	917.2736	246	5928.7115
CONFLOW option 1	790.6707	656.9071	246	5898.7115
CONFLOW option 2	789.9143	673.1742	246	5898.7115
CONFLOW option 3	829.7558	711.9993	246	5898.7115



(a) (b)

Figure 6.97: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

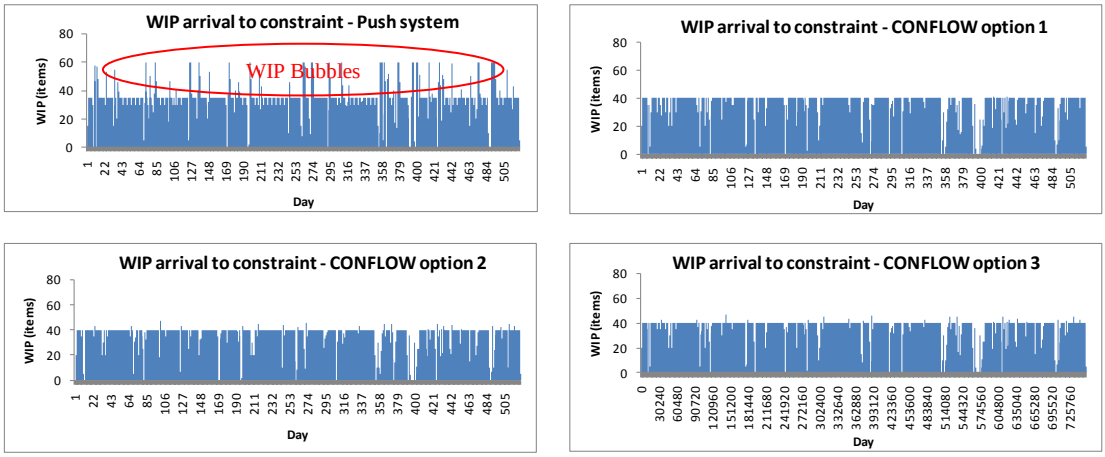


Figure 6.98: WIP arrival to constraint machine for all release strategies

The cycle time distribution profiles (Figure 6.99) shows that the push system has many more items with a cycle time around 3000min. This is due to the WIP bubbles (Figure 6.98). Once the queue is built in front of the constraint due to some downtime, it reduces very slowly and affects many items. In CONFLOW only few items are affected.

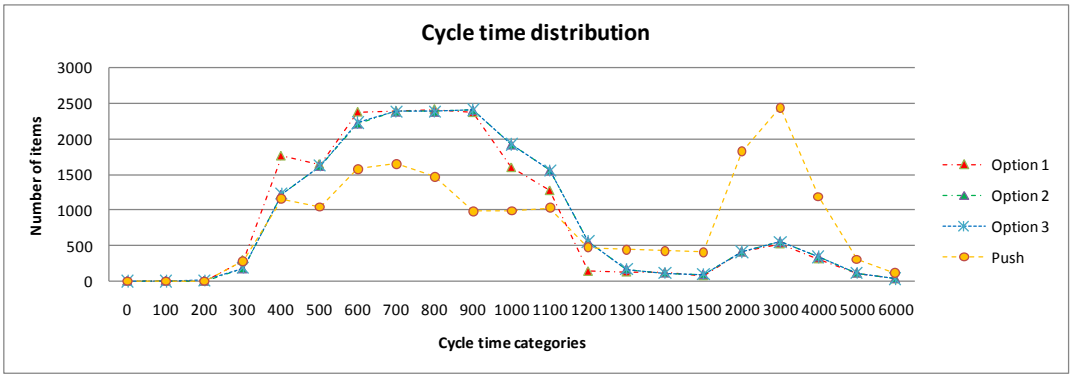


Figure 6.99: Cycle time distribution for all release strategies

6.4.5 Scenario 5: 5-Stage Serial Line with Batch, Downtime, Constraint Machine and Re-Entrant Line

In scenario 5, all the experimental conditions were kept identical to scenario 3 BTC except for the re-entrant line. Results display similar behaviors, and most conclusions from scenario 3 BTC are still valid. The analysis will therefore focus on the few differences and their meanings for the management of re-entrant lines.

Figure 6.100 and Figure 6.101 display respectively the output rate and the time to produce 1000 items. In comparison to scenario 3 BTC, the output rate is halved and the time to process 1000 items is doubled for all release strategies and capacities. This is not surprising as the items release has been halved to 10 items/shift. In the push system, the output rate corresponds almost to 10 items/shift, (above 9.95 items/shift for all capacities). In other words, all the items released are processed. With CONFLOW strategies, the output rate is slightly lower because the number of items introduced in the line is modulated when the availability is too low. It results in a lower input rate and thus a lower output rate.

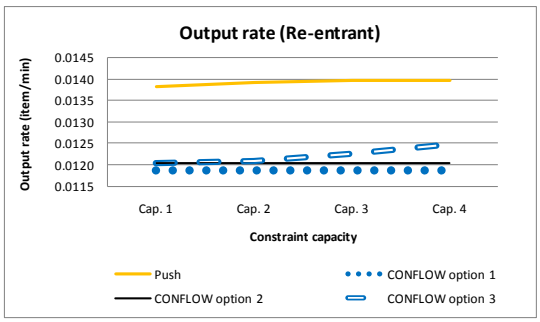


Figure 6.100: Re-entrant line model output rate for all release strategies

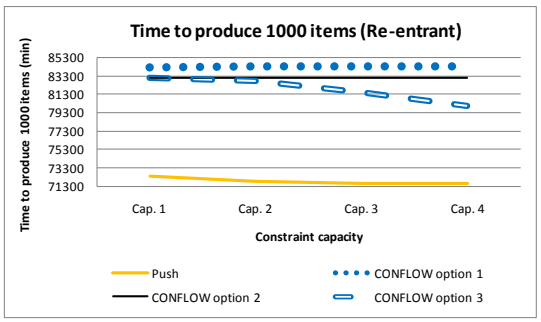


Figure 6.101: Time to produce 1000 items in re-entrant line model for all release strategies

As expected, the cycle time has increased (Table 6.21 to Table 6.24). Items have to go through the whole line twice. Nevertheless the increase is proportionally much higher in the push system, than in CONFLOW option 1, 2 and 3, particularly at low constraint capacity. In other words, the re-entrant line affects much less the cycle time when CONFLOW release strategies are employed than when the items are just pushed in the line.

Table 6.21: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	7814.9159	1036.6311	5082.7115	10818.7115
CONFLOW option 1	1170.9161	712.8278	636	8648.4911
CONFLOW option 2	1332.3800	757.1382	600	8792.4911
CONFLOW option 3	1332.3800	757.1382	600	8792.4911

Table 6.22: Performance of the policies with respect to cycle time for Capacity 2

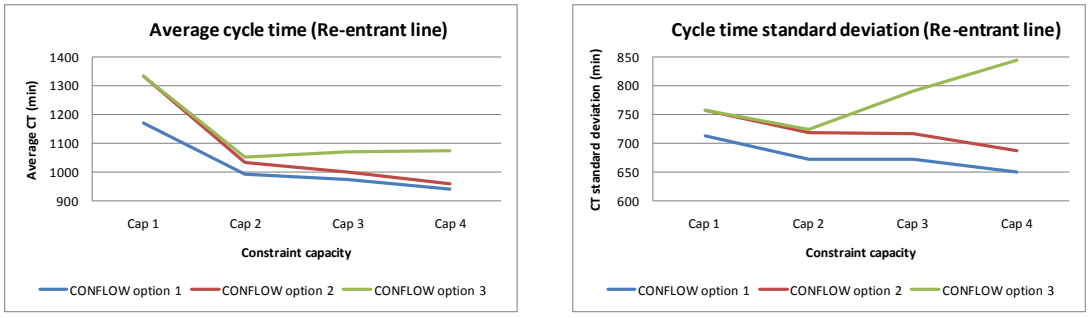
	Mean	Stdev	Min	Max
Push	4006.4730	1654.5738	589.740	9068.5513
CONFLOW option 1	992.7091	671.7229	589.740	8463.3611
CONFLOW option 2	1033.5143	718.8958	589.740	8429.0711
CONFLOW option 3	1052.3329	724.0782	589.740	8566.2311

Table 6.23: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	2771.6779	1690.0564	580.380	8640.6013
CONFLOW option 1	972.8176	671.2001	580.380	8458.6811
CONFLOW option 2	999.5913	717.4581	580.380	8425.9511
CONFLOW option 3	1071.8331	790.9980	580.380	8589.6011

Table 6.24: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1826.6397	1447.7357	564	8017.4231
CONFLOW option 1	938.5053	650.1203	564	7284.4701
CONFLOW option 2	960.1884	686.5027	564	7356.4701
CONFLOW option 3	1074.3350	844.8125	564	7970.4911



(a) (b)

Figure 6.102: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

In scenario 5 capacity 1, CONFLOW release strategies (CONFLOW option 1, 2 and 3) improve the cycle time by as much as 85% in comparison to the push release strategy. CT was 'only' improved by 78% in scenario 3 BTC capacity 1. For all other capacities, CONFLOW release strategies also improve CT more in comparison to the push strategy than it had in scenario 3, BTC. In other words, CONFLOW is even more effective with a re-entrant line. Once again the gain in CT is due to the elimination of WIP bubbles with CONFLOW (Figure 6.103).

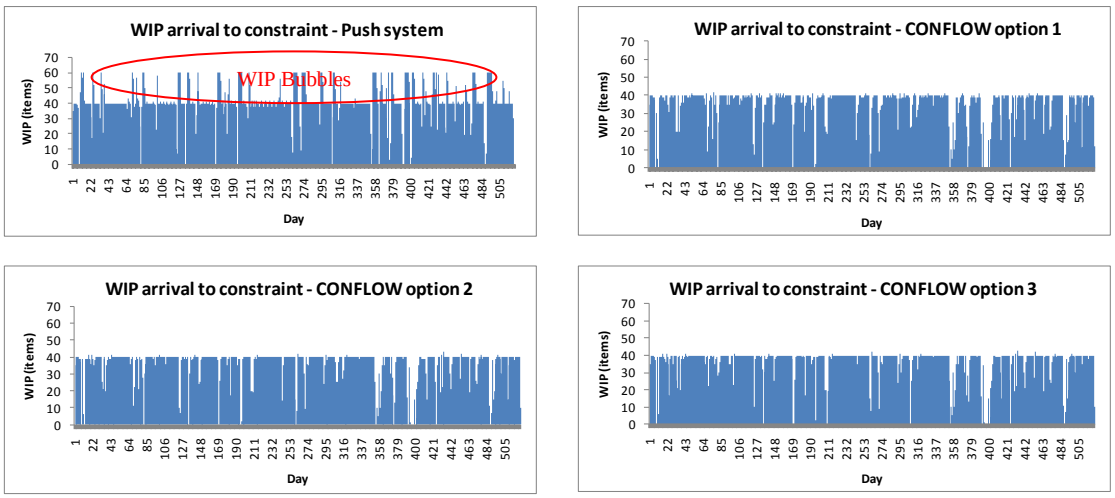


Figure 6.103: WIP arrival to constraint machine for all release strategies in re-entrant line model

This time, Max CT is different for all release strategies. Indeed, all the items have to go through the machine affected by downtime twice. Depending on the release strategy, items affected by the longest failure on their first passage are encountering different queuing conditions when there are coming back for their second passage.

Once again, if the various options of CONFLOW are compared (Figure 6.102), it can be seen that Option 1 has the lowest cycle time and cycle time standard deviation, whereas Option 3 has the highest. Option 2 is in between.

Compared to scenario 3 BTC (Figure 6.104 compared to Figure 6.53), there is a slight WIP reduction, 6~10 items (Figure 6.105 compared to Figure 6.54), for all release strategies and capacities. This is simply due to the lower release of 10 items/shift instead of 20 items/shift. Even if the machines still have to process 20 items/shift, there are fewer items in the line.

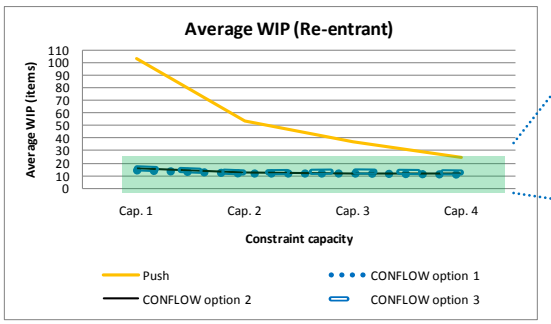


Figure 6.104: Average WIP in re-entrant line model for all release strategies

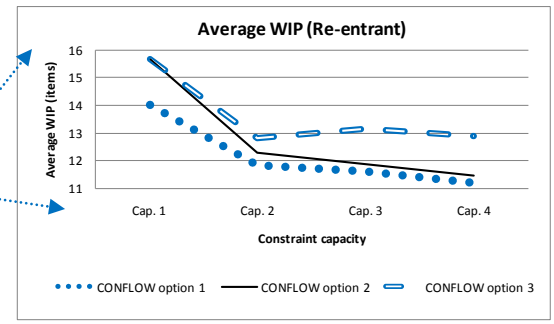


Figure 6.105: Zoom on CONFLOW Option 1, 2 and 3. Average WIP in re-entrant line model

CONFLOW release strategies (CONFLOW option 1, 2 and 3) reduce the WIP by 86% for capacity 1 and 47% for capacity 4 in comparison to the push system. It is slightly better than in scenario 3, BTC.

Figure 6.106 represents the distribution of the items cycle time for all release strategies (note the different scales applied for Push and CONFLOW). As in scenario 3 BTC, for CONFLOW Option 1, 2 and 3 most items are grouped but few items have a much higher cycle time. While, for the push system cycle times are more evenly spread, therefore the cycle time standard deviation is higher than in CONFLOW (Table 6.21 to Table 6.24).

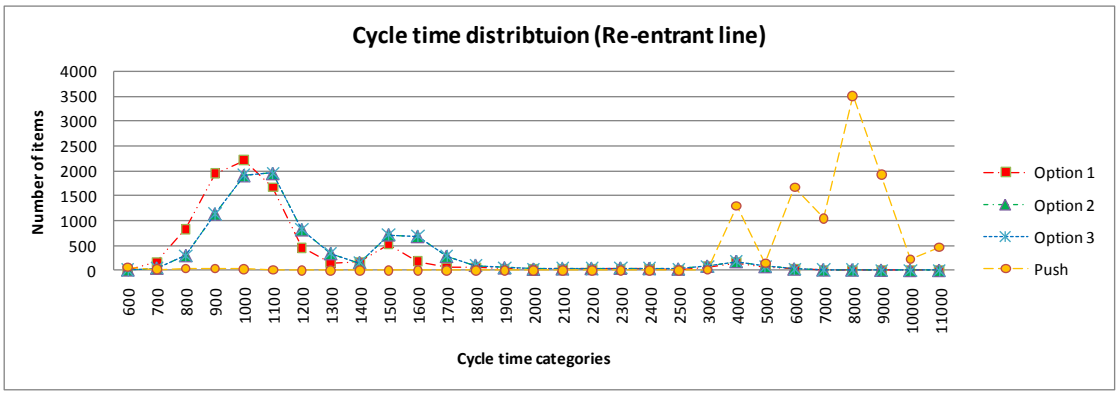


Figure 6.106: Cycle time distribution in re-entrant line model for all release strategies

In conclusion, the results obtained in scenario 5 are similar to those of scenario 3 BTC. Adding a re-entrant line, does not fundamentally change the results obtained when considering only a straight production line. CONFLOW still improves CT considerably at the cost of a lower throughput. Among the three CONFLOW options (Figure 6.102), Option 1 has the lowest throughput but also the lowest WIP, lowest CT and lowest CT standard deviation. Option 3 has the highest throughput, WIP, CT and CT standard deviation. Option 2 compromises throughput, WIP and CT.

6.4.6 Scenario 6: 5-Stage Model with Failures on Multiple Stages

The results are very similar to those of scenario 1 simulation model 1. Figure 6.107 and Figure 6.108 show that CONFLOW option 1, 2 and 3 have a slower throughput than the push system by approximately 44%. The push system needs approximately 25 days at capacity 1 (36646 min / [60×24]) to produce 1000 items. But CONFLOW option 1, option 2 and option 3, need 18 days more than push system to produce 1000 items.

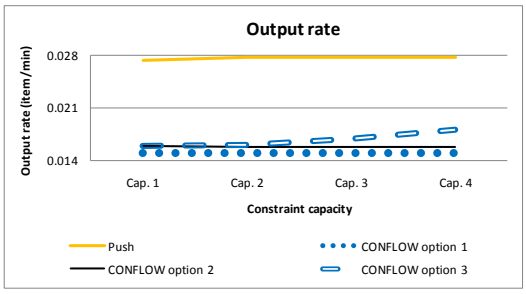


Figure 6.107: Scenario 6, Output rate

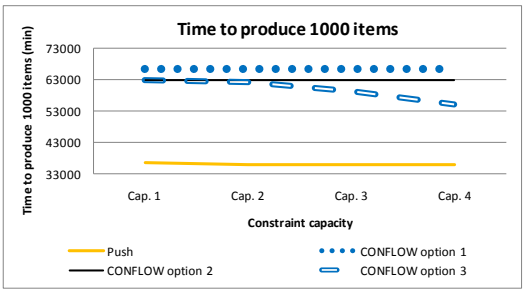


Figure 6.108: Scenario 6, Time to product 1000 items

For capacity 1, the mean cycle time (Table 6.25) is improved by as much as 86%, the standard deviation by 57%, and the WIP number (Figure 6.109) by 92%. For capacity 4 (Table 6.28), they are still respectively at 69%, 42% and 82%. The Min CT is much

higher in the push system due to the queue already existing at the constraint after 3 months (initial data deletion) production.

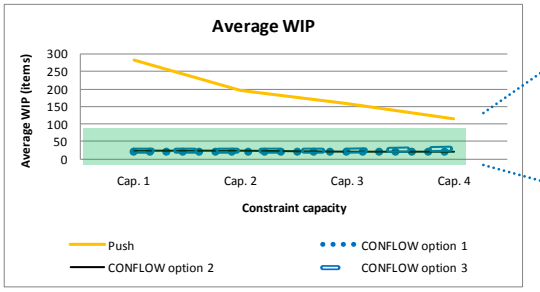


Figure 6.109: Scenario 6, Average WIP

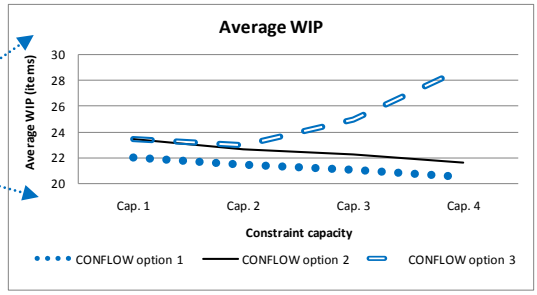


Figure 6.110: Scenario 6, Average WIP. Zoom on CONFLOW Option 1, 2 and 3.

Table 6.25: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	10218.1866	3852.2675	3707.5959	15538.4218
CONFLOW option 1	1421.6877	1646.7959	252	14998.4218
CONFLOW option 2	1454.1265	1698.3572	252	14998.4218
CONFLOW option 3	1448.8967	1698.9944	252	14998.4218

Table 6.26: Performance of the policies with respect to cycle time for Capacity 2

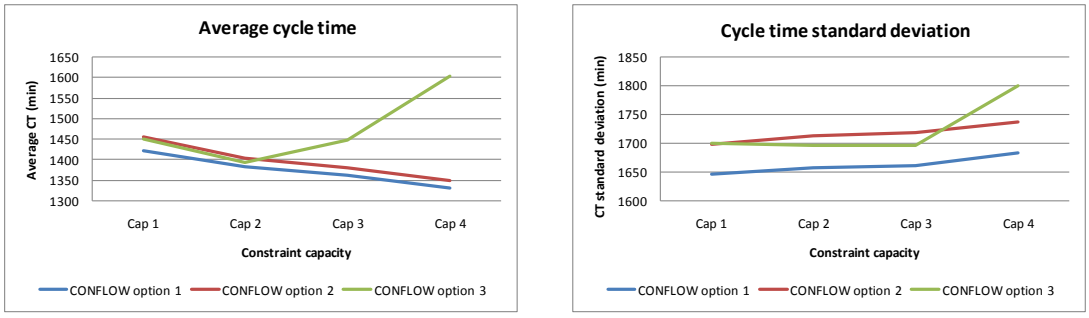
	Mean	Stdev	Min	Max
Push	7226.9023	3120.0576	1198.9729	15504.2218
CONFLOW option 1	1382.0513	1656.8061	250.290	14989.9618
CONFLOW option 2	1402.6830	1713.1900	250.290	14989.9618
CONFLOW option 3	1394.0764	1695.3824	250.290	14989.9618

Table 6.27: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	5840.3115	2907.9725	566.8129	15473.0218
CONFLOW option 1	1361.1403	1661.2920	248.730	14982.1318
CONFLOW option 2	1379.5218	1717.6937	248.730	14982.1318
CONFLOW option 3	1447.5388	1696.2024	248.730	14982.1318

Table 6.28: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	4273.8033	2890.5373	246	15418.4218
CONFLOW option 1	1331.4267	1682.5217	246	14998.4218
CONFLOW option 2	1348.2935	1737.8036	246	14998.4218
CONFLOW option 3	1603.9236	1800.5096	246	14998.4218



(a) (b)

Figure 6.111: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

With more machines failing, differences between the three CONFLOW options are increased. Nevertheless, Option 1 remains the option with the lowest throughput (Figure 6.107) but also lowest WIP (Figure 6.109) and CT (Table 6.25 to Table 6.28), while Option 3 has the highest throughput and also highest WIP and CT. Option 2 is a compromise between the two other options.

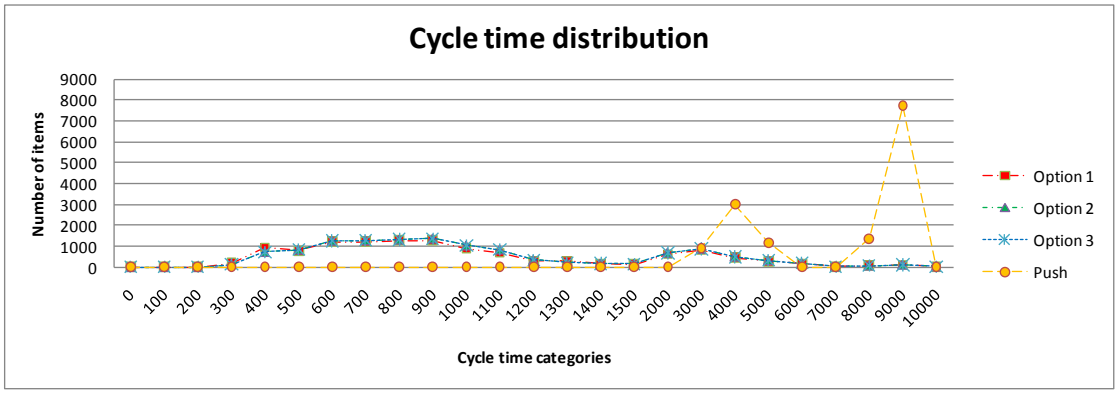


Figure 6.112: Cycle time distribution for all release policies

By monitoring the availability of all the machines preceding the bottleneck, CONFLOW can handle failures on multiple machines. CONFLOW ensures that the constraint machine does not receive WIP bubbles whichever machine fails or even if several machines fail simultaneously. The difference in performance between Push and CONFLOW is increased when several machines are affected by downtime. The throughput of CONFLOW is much smaller but the cycle time, its standard deviation and the WIP level are greatly improved. The differences between the three CONFLOW options are also increased. Option 1 provides the best WIP and CT while Option 3 provides the best throughput. Option 2 compromises WIP, CT and throughput.

6.4.7 Scenario 7: TOC vs. CONFLOW

SA vs. CONFLOW

SA shows a lower throughput (Figure 6.113) but also a lower WIP (Figure 6.115) and lower CT (Table 6.29 to Table 6.32 and Figure 6.116).

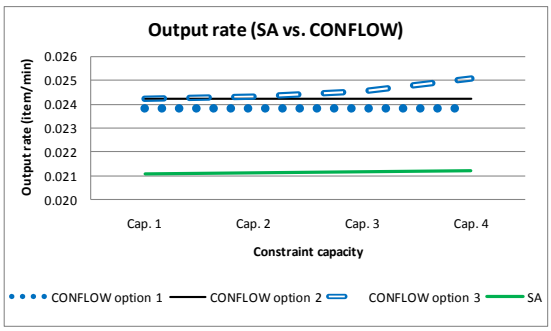


Figure 6.113: Output rate, SA vs. CONFLOW

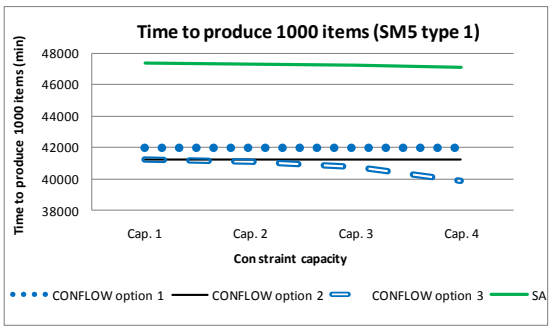


Figure 6.114: Time to produce 1000 items, SA vs. CONFLOW

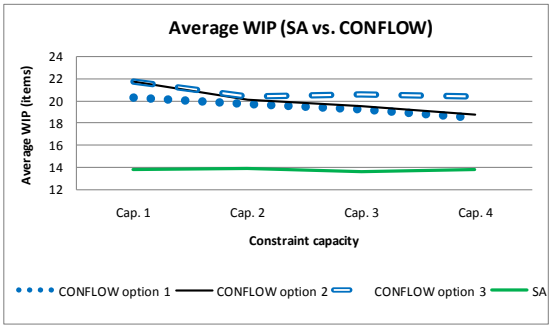


Figure 6.115: Average WIP, SA vs. CONFLOW

Table 6.29: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
SA	667.0839	534.0695	252	5838.7115
CONFLOW option 1	865.6927	661.1602	252	5874.7115
CONFLOW option 2	911.3226	670.8861	252	5874.7115
CONFLOW option 3	911.3226	670.8861	252	5874.7115

Table 6.30: Performance of the policies with respect to cycle time for Capacity 2

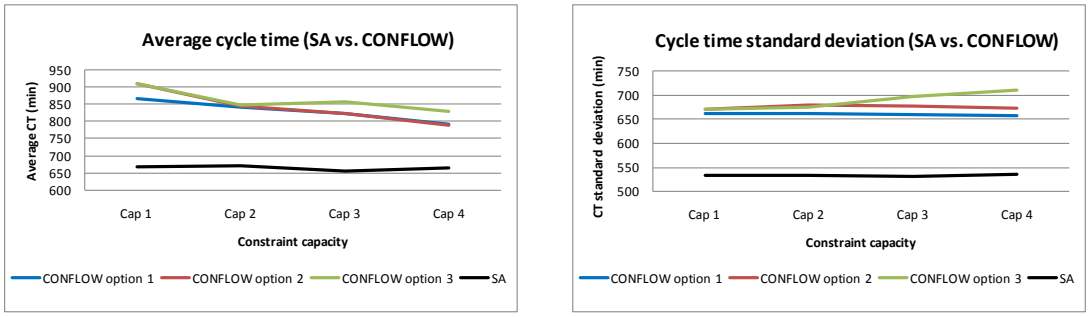
	Mean	Stdev	Min	Max
SA	669.1082	532.2813	250.290	5864.4515
CONFLOW option 1	839.9461	662.1305	250.290	5898.8315
CONFLOW option 2	843.4163	680.0887	250.290	5898.8315
CONFLOW option 3	846.4156	675.0765	250.290	5898.8315

Table 6.31: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
SA	654.2785	530.2142	248.730	5855.0915
CONFLOW option 1	821.6120	659.4503	248.730	5887.8815
CONFLOW option 2	822.5528	677.0339	248.730	5887.8815
CONFLOW option 3	858.4838	696.9786	248.730	5887.8815

Table 6.32: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
SA	665.0143	534.876	246	5958.7115
CONFLOW option 1	790.6707	656.9071	246	5898.7115
CONFLOW option 2	789.9143	673.1742	246	5898.7115
CONFLOW option 3	829.7558	711.9993	246	5898.7115



(a) (b)

Figure 6.116: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

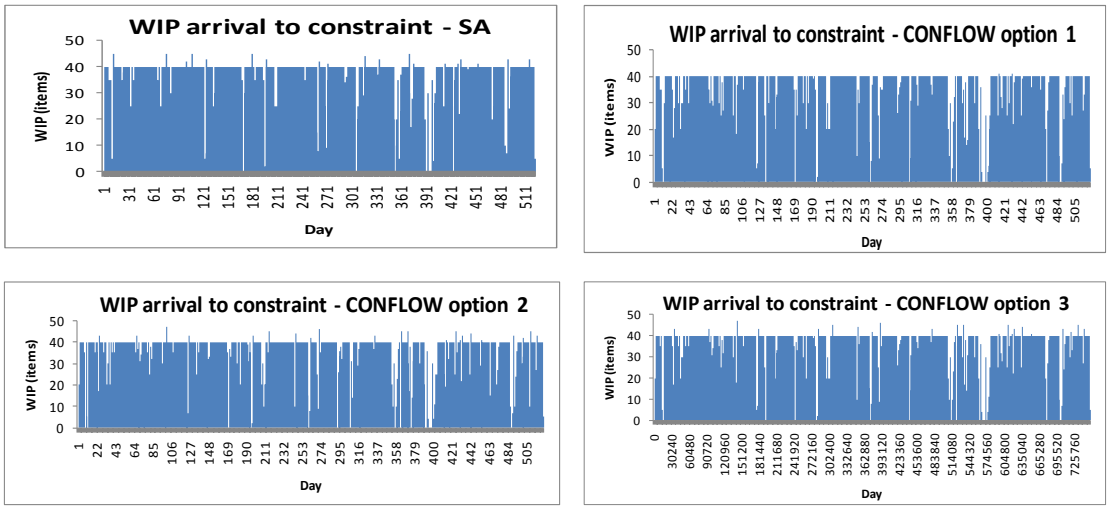


Figure 6.117: WIP arrival to constraint machine, SA vs. CONFLOW

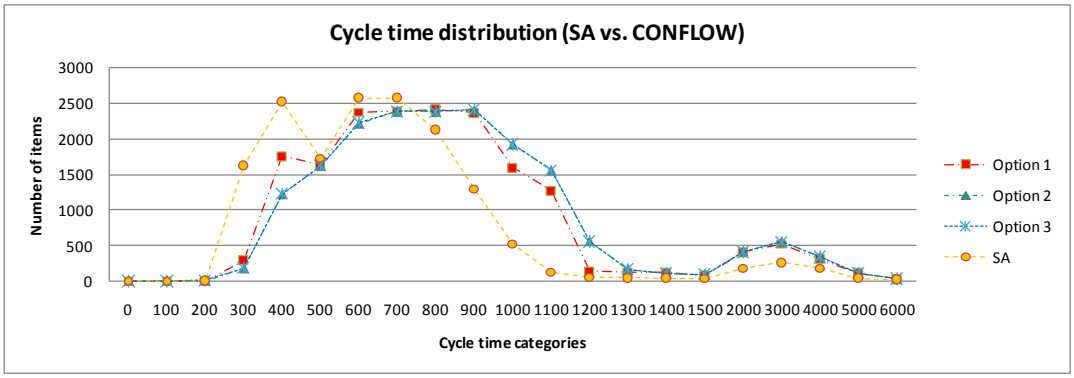


Figure 6.118: Cycle time distribution for all release policies

These results are not specific enough to conclude about the superior performance of one policy above the other. Nevertheless, they show that their performance should be relatively similar. Indeed, if the throughput of SA is increased, then WIP and CT will also increase.

For a better comparison of SA and CONFLOW, the setup of the SA simulation must be adjusted to obtain the same throughput as CONFLOW. The throughput of SA can be increased by applying a higher target WIP. A higher target will also increase the WIP level. Then the impact on the cycle time will be determinant for the comparison of the release policies. This simulation should be addressed in future work

SA should also be tested in its best position, i.e. settings optimizing SA performances should be determined. The results should then be compared with those obtained with CONFLOW. This simulation should also be addressed in future work.

DBR vs. CONFLOW

DBR shows a lower throughput (Figure 6.119) but also a lower WIP (Figure 6.121) and slightly lower CT (Table 6.33 to Table 6.36 and Figure 6.122).

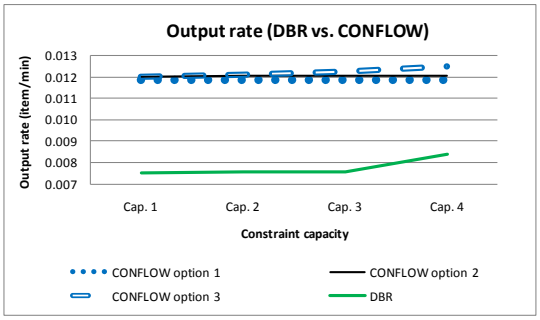


Figure 6.119: Output rate, DBR vs. CONFLOW

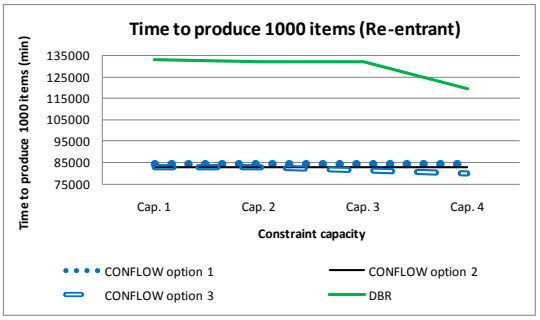


Figure 6.120: Time to produce 1000 items, DBR vs. CONFLOW

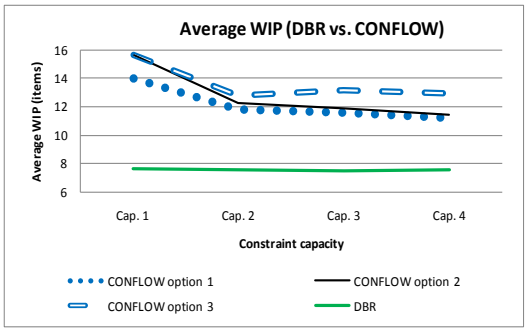


Figure 6.121: Average WIP, DBR vs. CONFLOW

Table 6.33: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
DBR	873.0849	560.2583	756	8321.3417
CONFLOW option 1	1170.9161	712.8278	636	8648.4911
CONFLOW option 2	1332.3800	757.1382	600	8792.4911
CONFLOW option 3	1332.3800	757.1382	600	8792.4911

Table 6.34: Performance of the policies with respect to cycle time for Capacity 2

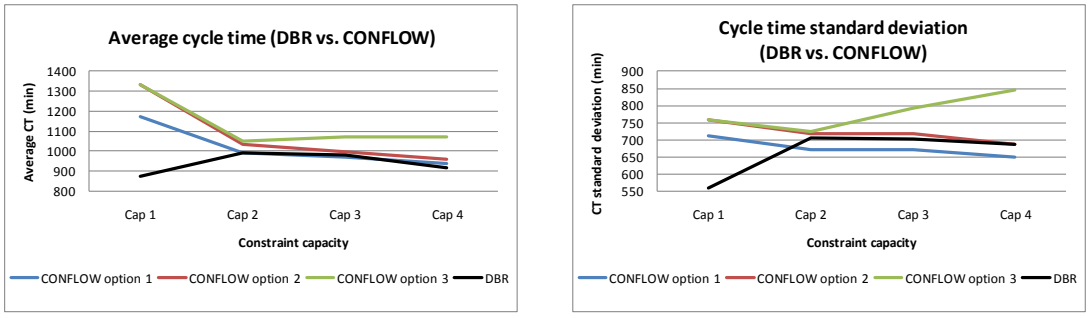
	Mean	Stdev	Min	Max
DBR	992.9691	704.9332	589.740	8429.0711
CONFLOW option 1	992.7091	671.7229	589.740	8463.3611
CONFLOW option 2	1033.5143	718.8958	589.740	8429.0711
CONFLOW option 3	1052.3329	724.0782	589.740	8566.2311

Table 6.35: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
DBR	982.8978	701.5839	580.380	8393.2211
CONFLOW option 1	972.8176	671.2001	580.380	8458.6811
CONFLOW option 2	999.5913	717.4581	580.380	8425.9511
CONFLOW option 3	1071.8331	790.9980	580.380	8589.6011

Table 6.36: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
DBR	914.9310	686.1010	564	8244.2989
CONFLOW option 1	938.5053	650.1203	564	7284.4701
CONFLOW option 2	960.1884	686.5027	564	7356.4701
CONFLOW option 3	1074.3350	844.8125	564	7970.4911



(a) (b)

Figure 6.122: Performance of CONFLOW Options across the Capacities Investigated in terms of (a) Mean Cycle Time and (b) Standard Deviation of Cycle Time

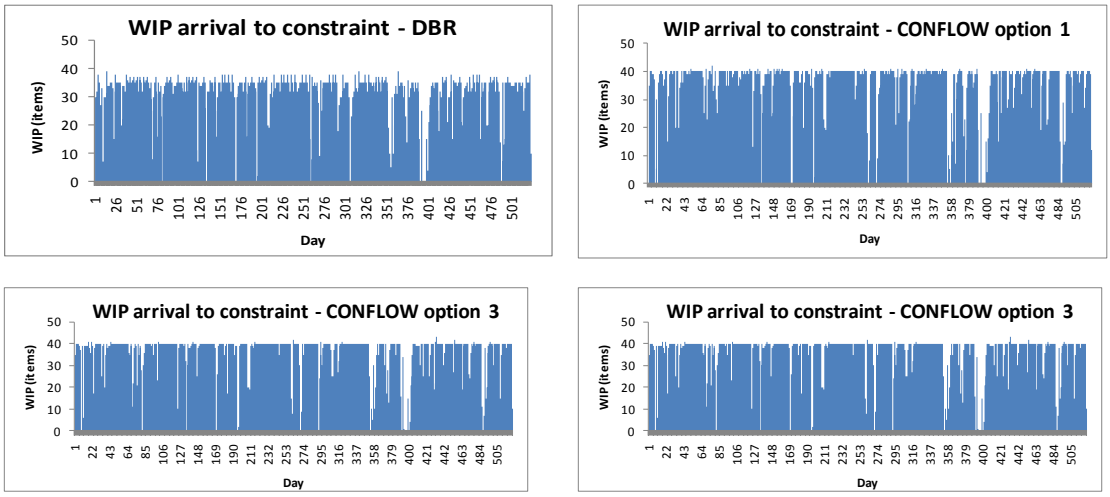


Figure 6.123: WIP arrival to constraint machine, DBR vs. CONFLOW

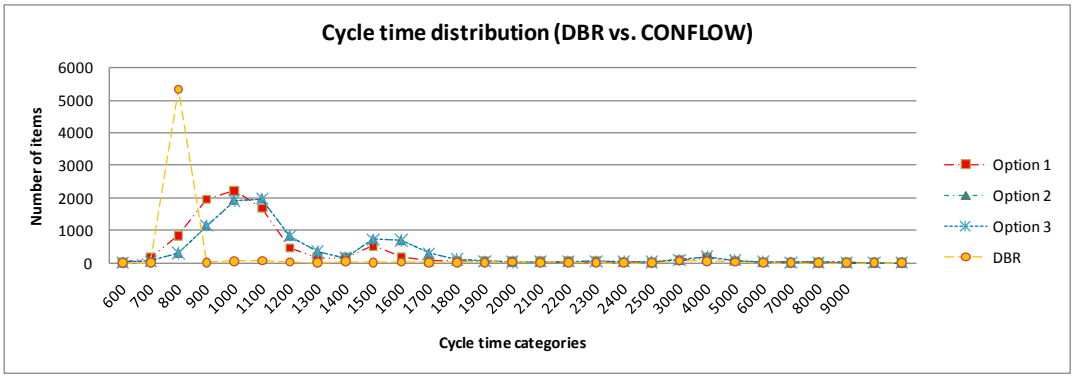


Figure 6.124: Cycle time distribution for all release policies

These results are not specific enough to conclude about the superior performance of one policy above the other. Nevertheless, they show that their performance should be relatively similar with maybe a slight advantage for CONFLOW. Indeed, if the throughput of DBR is increased, then WIP and CT will also increase. The cycle time and its standard deviation will probably become higher than for CONFLOW, particularly for higher capacities.

For a better comparison of DBR and CONFLOW, the setup of the DBR simulation must be adjusted to obtain the same throughput as CONFLOW. The throughput of DBR can be increased by increasing the initial WIP. It will naturally also increase the WIP level. Then the impact on the cycle time will be determinant for the comparison of the release policies. This simulation should be addressed in future work

DBR should also be tested in its best position, i.e. settings optimizing DBR performances should be determined. The results should then be compared with those obtained with CONFLOW. This simulation should also be addressed in future work.

CHAPTER - 7 DISCUSSION OF RESULTS

An overall map of the study's simulations is given in Table 7.1. With simulations, the fidelity and validity of the simulations outcomes is dependent on the acquisition of valid source of information, the relevant selection of key characteristics and behaviors, and the appropriate use of simplifying approximations and assumptions within the simulation. Moreover simulation results are hypothetical. Ideas and theories may be tested with simulations, but the transfer to real environment is not 100% sure and outcomes may differ slightly from predictions, in complex systems as semiconductor processing lines. In particular, all the models considered are relatively simple compare to real production lines. Therefore, precautions must be taken in the results' transfer to real environment.

Table 7.1: Summary of experiments

Experiment		Page	Key concept
Variability Measurement	Inter-departure time coefficient of variation	79	Detection of production stages with high ouput variability
	Difference metric	81	Detection of production stages creating variability
	Ratio metric	83	Determination of how much variability a production stage creates
	Correlation Coefficient	88	Variability propagation in the production line
Batch Process Production Stage Investigation	Scenario 1: Fixed (High) Production Load, Variable Release Profile in SIPM and BPM Models	111	Examine the impact of release rate on performance of SIPM and BPM models
	Scenario 2: Fixed Release Profile, Variable Production Load in SIPM and BPM Models	116	Examine the impact of production load on the performance of SIPM and BPM models given a specific release profile (twice/day)
	Scenario 3: Initial Assessment of Item Release Rates which Minimize Queuing for Batching in BPM Model	120	Test the performance of particular release rates naturally adpated to the BPM model
	Scenario 4: Further Assessment of Item Release Rates which Minimize Queuing for Batching in BPM Model	121	Test the performance of release rates specifically designed for BPM model
Unreliable Production Stage Investigation	Experiment 1: Impact of Downtime Frequency	126	Study the impact of downtime frequency on mean cycle time and cycle time variability
	Experiment 2: Impact of Repair Time Variability	132	study the impact of repair time variability (TTR) on mean cycle time and cycle time variability for two downtime frequencies (daily and weekly)
Constant Flow (CONFLOW) Release Strategy Evaluation	Recovery Performance Simulation (RPS)	164	Determine how long it takes for a system to recover baseline performance in function of the different release strategies applied and the constraint capacity
	Simulation Model 1 (SM1): Availability Machine 1 and Constraint Machine 2	167	Test CONFLOW's response to random downtime in a two-stage configuration
	Simulation Model 2 (SM2): Availability and Batch Process Machine 1, Constraint Machine 2	174	Test CONFLOW's response to random downtime in a two-stage configuration with one batching stage
	Simulation Model 3 (SM3): Parallel Process Simulation Model with Availability Operation 1 and Constraint Operation 2	176	Test CONFLOW's response to random downtime in a two-stage configuration with parallel processing
	Scenario 2: 5-Stage Serial Line with Constraint and Downtime (SM4)	179	Test CONFLOW's response to random downtime in a five-stage configuration
	Scenario 3: 5-Stage Serial Line with Batch, Downtime and Constraint (SM5)	181	Test CONFLOW's response to random downtime in a five-stage configuration with one batching stage
	Scenario 4: Push and CONFLOW policies matched throughput	199	Compare Push and CONFLOW response to random downtime when their throughput is matched in a five-stage configuration with one batching stage
	Scenario 5: 5-Stage Serial Line with Batch, Downtime, Constraint and Re-entrant Line	202	Test CONFLOW's response to random downtime in a five-stage configuration with one batching stage and one re-entrant line
	Scenario 6: 5-Stage Model with Failures on Multiple Stages	207	Test CONFLOW's response to random downtime in a five-stage configuration with one batching stage and failures on multiple stages
	Scenario 7: TOC vs CONFLOW - Starvation Avoidance Policy Setup	210	Compare Push and SA response to random downtime in a five-stage configuration with one batching stage
	Scenario 7: TOC vs CONFLOW - Drum Buffer Rope Policy Setup	213	Compare Push and DBR response to random downtime in a five-stage configuration with one batching stage and one re-entrant line

7.1 Variability Metric

Real data were observed and the correlation coefficient and the difference (or ratio) metrics were used to analyze them. The difference (or ratio) metric shows clearly the variability's origins and the correlation coefficient metric highlights the relationship between two operations. Therefore, variability generated by an operation can be measured using the difference (or ratio) metric and the correlation between operations can be measured (Chapter - 3).

The operation adding the biggest contribution to variability was identified, but the cause of this variability could not be identified. Therefore, it was decided to work in a more controlled environment using simulations models.

7.2 Tool Availability and Batching Influence on Cycle Time and Cycle Time Variability

In the six operations simulation model, the experiments show that the release rate affects the cycle time and cycle time variability of a line with a batch processing operation. The impact of the extra delays incurred in the batch formation may be exacerbated by poor control of the arrival items into the batch operations. So, the issue is not simply that of inter-arrival time of individual items for processing, but rather that a sufficient quantity of items to allow for complete batch formation should arrive in reasonable proximity to each other. Batch processing operation improves the arrival of the items into constraint operation. This benefit increases as the loading of the line drops, as the excess capacity in the constraint operation can deal with the items arriving together from a single batch before the arrival of the next group (Section 5.2).

If tool availability is considered in the simulation model, results show that an operation manager has interest considering the shift (12 hours) operation availability, and its variability instead of the average availability of the operation (Section 5.3).

7.3 Variability and Interactions between Operations

Queue must be avoided in front of any high capacity operation, because the fluctuations in the output of such operations will have grave consequences when they reach the constraint operation in the line (Section 5.3).

Controlling the variability of the repair time becomes critical if the availability is close to the critical availability (shift availability that reduces the shift capacity down to the operation loading). This conclusion can probably be extended to any source of variability. If the availability of a high capacity operation is at a safe margin above the critical availability, variability in this operation will not affect the cycle time. On the other hand, if the availability is close to the critical availability, then any sources of variability will create considerable strain on the constraint operation. It results in the notion of critical capacity. If, despite variability, the shift capacity of an operation is always above the critical capacity, then variability has no major impact on cycle time and cycle time variability (Section 5.3).

Therefore, monitoring the availability of high capacity operations and momentarily reducing the number of items released when the availability is under its critical value should greatly improve CT by avoiding the formation of a queue at a high capacity operation.

7.4 Constant Flow (CONFLOW) Release Strategy

7.4.1 A Novel Release Strategy

A novel hybrid push-pull release strategy, named Constant Flow (CONFLOW) was developed. CONFLOW avoids variability and regulates the flow of items reaching the constraint operation. It is achieved by monitoring the availability of all operations preceding the constraint operation. When downtime is detected, the release is modulated to avoid the creation of a WIP bubble. The elimination of WIP bubbles considerably stabilizes the flow of items into the line.

Three different options were proposed to calculate the number of items to be released in the line. CONFLOW option 3 has the best throughput, followed by Option 2 and finally Option 1. On the other side Option 3 has the highest WIP level and highest CT, followed by Option 2 and finally Option1. Therefore production line managers have to choose Option 1 if their priorities are WIP and CT, Option 3 if their priority is throughput, and for a compromise, they can choose Option 2.

7.4.2 A Robust Release Strategy

CONFLOW was tested under many conditions (batching, parallel processing, different line length, re-entrant line) and the results are robust.

When several machines are affected by failures, CONFLOW can still be applied by monitoring the availability of all the machines preceding the constraint. The lowest availability is then used to determine the number of items released and avoid the formation of a WIP bubble.

CONFLOW release strategy is compatible with batch operations. As CONFLOW only affect the release on few shifts, most of the releases keep the standard matched batch size. Therefore the improvements described are still achieved with a batch operation.

The results obtained, when CONFLOW release strategy is applied in a system with a re-entrant line, are encouraging. Results are similar to those in a system without re-entrant lines. Significant improvements in WIP and CT are seen with some loss in throughput. Further testing is required to confirm the results obtained in systems with complex re-entrant lines.

7.4.3 CONFLOW vs Push

The results show that CONFLOW release strategy allows a quick recovery of the WIP created by downtime even in extreme circumstances where a constant push policy could never recover.

CONFLOW release strategy brings significant improvement over a constant push policy. Across various scenarios, improvement of up to 86% in mean cycle time and reduction of WIP by as much as 88% could be seen. However, throughput can drop by 14% with the CONFLOW release strategy compared to the push system. It is due to lost capacity at the bottleneck with CONFLOW release strategy. In the push system, the queue created at the bottleneck prevents the loss of capacity. In terms of variability, the CT coefficient of variation is higher because few items have a very long CT compared to the others.

Compared to a push release policy, CONFLOW improves Cycle time and WIP level at the detriment of throughput (Chapter - 6).

7.4.4 CONFLOW vs TOC

CONFLOW was compared to other TOC policies (SA and DBR). The results are not specific enough to conclude about the superior performance of one policy above the other. Nevertheless, they show that their performance should be relatively similar. CONFLOW might even achieve slightly better performances. To confirm these results, further simulations should be run with setups matching the throughputs of all the various release policies.

CHAPTER - 8 CONCLUSION AND RECOMMENDATIONS

8.1 Introduction

The objective of this study was to determine how the cycle time can be shortened and the variability of the overall production line reduced. Four goals were targeted for achieving production line improvement. First, develop a metric to measure the variability of operations. Second, understand the impact of tool availability and batch variability. Third, understand the operations' interactions and their impact on cycle time and variability. Finally, develop a release strategy to control the production flow. The conclusions drawn from this research are presented next in Section 8.2 and opportunities for further research are delineated in Section 8.3.

8.2 Conclusions

A new hybrid push pull release strategy (CONFLOW) was developed. CONFLOW regulates the flow of items reaching the constraint operation. It is achieved by monitoring the availability of all operations preceding the constraint operation. When downtime is detected, the release is modulated to avoid the creation of a WIP bubble. The elimination of WIP bubbles considerably stabilizes the flow of items into the line. Three different options were considered. One allows a better control of WIP; one improves the throughput and the last one compromise between throughput and WIP level.

Compared to a push system, CONFLOW release strategy results, into significant improvement in cycle time, cycle time standard deviation and WIP level at the cost of

reduction in throughput. Improvements become particularly relevant as the loading comes closer to the constraint machine maximum capacity. CONFLOW has been tested under many operating conditions including parallel processing, batching, several production line lengths and re-entrant lines. The results are consistent in all these operating conditions.

It was also shown that the higher performance of CONFLOW compare to a push system was heightened when the production line becomes longer with several machines simultaneously affected by downtime. This is a promising result for the application of CONFLOW to real production lines. Indeed, front end semiconductor production lines are made of many more operations than used in the simulations.

CONFLOW performances were compared to common TOC strategies (SA and DBR). The results are encouraging. In the specific conditions considered, CONFLOW performances are similar to SA and slightly better than DBR.

The factors of variability in a production line were identified from the literature review. Three objectives were extracted as fundamental for the improvement of production lines: (1) describe the impact of tool and batch variability on the process flow, (2) understand the interaction between operations, and (3) determine a proper metric to measure the variability. These objectives were accomplished through the analysis of real production data and the use of model simulations.

From the observation of real data, it was shown that the origin of variability can be traced by measuring the difference (or ratio) metrics, and the relationship between two operations can be seen by measuring the correlation coefficient. Presented simulations

explain the relationships between line loading, batch operation, constraint operation and machine downtime.

A compromise between variability and queue time can be adjusted by controlling the correct loading of operation. This can be accomplished by setting either a correct utilization or a correct inter-departure time target. Therefore, this can provide line manager a good reference to determine how many items to input in the process line.

When batch process is taken into account, the input release profile clearly affects the cycle time and cycle time variability of the production line. This is caused by irregular arrivals. The lack of items to group into full size batches aggravates the queue time. Hence, the release quantity needs to be adjusted to form full batches without any items left in the buffer. It also has the added benefit of improving arrivals to the constraint operation.

Results show that an operation manger should consider the shift availability of an operation instead of longer term statistics. Through the monitoring of tool availability on short periods, the detection and resolution of problems is much quicker. And the formation of queue in front of any high capacity operation affected by downtime is avoided. This is fundamental, because the fluctuations in the output of such operations will greatly impact on the constraint operation.

This lead to the introduction and definition of critical availability. Availability has to stay in a safe margin above the critical availability, then operations will not affect to the cycle time. Otherwise, if the availability is close to the critical availability, then any sources of variability will create considerable strain on the constraint operation.

These results prove that high capacity operations can also be considered as source of improvement for the line. Most of the current literature concentrates exclusively on the constraint operation and neglect the benefits that might be possible through better management of the high capacity operations.

8.3 Recommendations for Future Work

CONFLOW does not solve all the problems of a factory manager. It improves cycle time and WIP level at the cost of higher variability and lower throughput. Reducing cycle time and WIP levels greatly improves the running cost and the predictability of the line, but reduced throughput increase the lead times for customers' delivery. Each production manager has to decide in function of his factory objectives and his customers' demand.

In its current definition, CONFLOW only considers downtime for several independent machines. The release policy needs to be extended to include downtime on any number of operations with parallel processing. Product mix was also not considered. It might be possible to modify CONFLOW to solve this issue.

Results show improved performance on longer production lines. However, there might be a maximum limit in the length of the production line for the application of CONFLOW. This possibility has not been studied.

In this study, CONFLOW was monitoring the operations through their availability. But other possibilities exit, such as number of item processed, working time or number of items in the buffer. In particular, working time might provide a solution to apply CONFLOW to product mix.

In this study, CONFLOW modulates a static push policy. However, it could as well modulate more complex push policies such as MRP. The efficiency of such system should be tested.

Re-entrant lines were studied with a very simple model. Further studies should be completed to valid CONFLOW release strategy in more complex production line. Larger scale simulations should be attempted to confirm CONFLOW validity before its test in real environment.

Further comparisons of CONFLOW with common TOC policies must be pursued to determine under which conditions each strategy outperforms the others.

REFERENCES

1. Woolverton A., Whitaker L., Shear J., Geniesse J., Brostuen D., and Schoepke D. *Fast cycle time in high-mix technology development and manufacturing*. in *Semiconductor Manufacturing Conference Proceeding, IEEE International Symposium*. 1999.
2. Li N., Zhang L., Zhang M., and Zheng L. *Applied factory physics study on semiconductor assembly and test manufacturing*. in *Semiconductor Manufacturing. ISSM, IEEE International Symposium*. 2005.
3. Bard J.F., Srinivasan K., and Tirupati D., *An optimization approach to capacity expansion in semiconductor manufacturing facilities.* International Journal of Production Research, 1999. **37**(15): p. 3359 - 3382.
4. Palmeri V. and Collins D.W. *An analysis of the "K-step ahead" minimum inventory variability policy using SEMATECH semiconductor manufacturing data in a discrete-event simulation model*. in *Emerging Technologies and Factory Automation Proceedings*. 1997.
5. Delp D.R. *A new X-Factor contribution measure for identifying machine level capacity constraints and variability*. in *Advanced Semiconductor Manufacturing, ASMC, IEEE Conference and Workshop*. 2004.
6. Sandell R. and Srinivasan K. *Evaluation of lot release policies for semiconductor manufacturing systems*. in *Winter Simulation Conference*. 1996.
7. Schömig A.K. *On the corrupting influence of variability in semiconductor manufacturing*. in *Simulation Conference Proceedings*. 1999.
8. Delp D., Si J., and Fowler J.W., *The development of the complete X-Factor contribution measurement for improving cycle time and cycle time variability.* Semiconductor Manufacturing, IEEE Transactions, 2006. **19**(3): p. 352 - 362.
9. Aurand S.S. and Miller P.J. *The operating curve: A method to measure and benchmark manufacturing line productivity*. in *Advanced Semiconductor Manufacturing Conference and Workshop, IEEE/SEMI*. 1997.
10. Kalir A. and Bouhnik S. *Achieving reduced production cycle times via effective control of key factors of the P-K Equation*. in *Advanced Semiconductor Manufacturing Conference, IEEE/SEMI*. 2006.
11. Hopp W.J. and Spearman M.L., *Factory Physics*. Third edition ed. 2008: McGraw-Hill.
12. Hendricks K.B. and McClain J.O., *The output processes of serial production lines of general machines with finite buffers.* Management Science, 1993. **39**(10): p. 1194 - 1201.
13. Zhang M.T., Fu J., and Zu E. *Dynamic capacity modelling with multiple re-entrant workflows in semiconductor assembly manufacturing*. in *Automation Science and Engineering, IEEE International Conference*. 2005.
14. Delp D., Si J., Hwang Y., and Pei B. *A dynamic system regulation measure for increasing effective capacity: The X-Factor theory*. in *Advanced Semiconductor Manufacturing Conference and Workshop*. 2003.

15. Fowler J.W., Brown S., Gold H., and Schömig A. *Measurable improvements in cycle-time-constrained capacity*. in *Semiconductor Manufacturing Conference Proceedings, IEEE International Symposium*. 1997.
16. Gross D. and Harris C.M., *Fundamentals of queueing theory* 2nd ed. 1985: New York : Wiley.
17. Papadopoulos H.T., Heavey C., and Browne J., *Queueing theory in manufacturing systems analysis and design*. 1993: New York : Chapman & Hall.
18. Babbs D. and Gaskins R. *Effect of reduced equipment downtime variability on cycle time in a conventional 300mm fab*. in *Advanced Semiconductor Manufacturing Conference, IEEE/SEMI*. 2008.
19. Uzsoy R., Church L.K., and Ovacik I.M. *Dispatching rules for semiconductor testing Operations: A computational study*. in *Electronics Manufacturing Technology Symposium, IEEE/CHMT* 1992.
20. Baum S.S. and O'Donnell C.M. *An approach to modelling labour and machine down time in semiconductor fabrication*. in *Simulation Conference*. 1991.
21. Davis A.E., Hamlin M., McCullough R.D., Teyner T., and Uzsoy R., *A decision support system for spare parts management in a wafer fabrication facility.* Semiconductor Manufacturing, IEEE Transactions 2001. **14**(1): p. 76 - 78.
22. Tag P.H. and Zhang M.T., *E-Manufacturing in the semiconductor industry.* IEEE Robotics & Automation Magazine, 2006. **13**(4): p. 25 - 32.
23. Mosley S.A., Teyner T., and Uzsoy R.M., *Maintenance scheduling and staffing policies in a wafer fabrication facility.* Semiconductor Manufacturing, IEEE Transactions, 1998. **11**(2): p. 316 - 323.
24. Reid J.M. *Predicting failure modes to improve reliability*. in *Reliability and Maintainability Symposium*. 1990.
25. Rose O. *Modelling tool failures in semiconductor fab simulation*. in *Simulation Conference*. 2004.
26. Tullis B., Mehrotra V., and Zuanich D. *Successful modelling of a semiconductor R&D facility*. in *Semiconductor Manufacturing Science Symposium, IEEE/SEMI*. 1990.
27. Woods R.H. *A cost benefit analysis of photolithography and metrology dedication in a metrology constrained multipart number fabricator*. in *Advanced Semiconductor Manufacturing Conference and Workshop, IEEE/SEMI*. 1998.
28. Stratman J.K., Roth A.V., and Gilland W.G., *The deployment of temporary production workers in assembly operations: A case study of the hidden costs of learning and forgetting.* Journal of Operations Management, 2004. **21**(6): p. 689 - 707.
29. Chung S.-H., Lee A.H.I., and Pearn W.L., *Product mix optimization for semiconductor manufacturing based on AHP and ANP analysis.* The International Journal of Advanced Manufacturing Technology, 2004. **25**(11-12): p. 1144 - 1156.
30. Chung S.-H., Lee A.H.I., and Pearn W.L., *Analytic network process (ANP) approach for product mix planning in semiconductor fabrication.* International Journal of Production Economics, 2005. **96**(1): p. 15 - 36.

31. Rohan D. *Machine dedication under product and process diversity*. in *Winter Simulation Conference Archive, Proceedings of the 31st conference on Winter Simulation*. 1999.
32. Jacobs J.H., Van Bakel P.P., Etman L.F.P., and Rooda J.E., *Quantifying variability of batching equipment using effective process times.* Semiconductor Manufacturing, IEEE Transactions, 2006. **19**(2): p. 269 - 275.
33. Curry G.L. and Deuermeyer B.L., *Renewal approximations for the departure processes of batch systems.* IIE Transactions, 2002. **34**(2): p. 95 - 104.
34. Bhatnagar R., Chandra P., Loulou R., and Qiu J., *Order release and product mix coordination in a complex PCB manufacturing line with batch processors.* International Journal of Flexible Manufacturing Systems, 1999. **11**(4): p. 327 - 351.
35. Glassey C.R. and Weng W.W., *Dynamic batching heuristic for simultaneous processing.* Semiconductor Manufacturing, IEEE Transactions, 1991. **4**(2): p. 77 - 82.
36. Nemoto K., Akcali E., and Uzsoy R.M., *Quantifying the benefits of cycle time reduction in semiconductor wafer fabrication.* IEEE Transactions on Electronics Packaging Manufacturing, 2000. **23**(1): p. 39 - 47.
37. Mönch L. and I. Habenicht. *Simulation-based assessment of batching heuristics in semiconductor manufacturing*. in *Proceedings of the Winter Simulation Conference, IEEE*. 2003.
38. Felipe D. and Daniel E., *A model predictive control approach for real-time optimization of reentrant manufacturing lines.* Computers in Industry, 2001. **45**(1): p. 45 - 57.
39. Rose O. *WIP evolution of a semiconductor factory after bottleneck work center breakdown*. in *Proceedings of the Winter Simulation Conference, IEEE*. 1998.
40. Rezaie K., Eivazy H., and Nazari-Shirkouhi S. *A novel release policy for hybrid make-to stock/make-to-order semiconductor manufacturing systems*. in *Developments in eSystems Engineering (DESE)*. 2009.
41. Choi J.Y. and Reveliotis S.A., *A generalized stochastic Petri net model for performance analysis and control of capacitated re-entrant lines.* Robotics and Automation, IEEE Transactions 2003. **19**(3): p. 474 - 480.
42. Narahari Y. and Khan L.M., *Modelling the effect of hot lots in semiconductor manufacturing systems.* Semiconductor Manufacturing, IEEE Transactions, 1997. **10**(1): p. 185 - 188.
43. Gupta A.K., Ganesan V.K., and Sivakumar A.I. *Hot lot management: minimizing cycle time in batch process*. in *Engineering Management Conference, IEEE International Publication*. 2004.
44. Lozinski C. and Glassey C.R., *Bottleneck starvation indicators for shop floor control.* Semiconductor Manufacturing, 1988. **1**(4): p. 147 - 153.
45. Liu W., Chua T.J., Cai T.X., Wang F.Y., and Yan W.J., *Practical lot release methodology for semiconductor back-end manufacturing.* Production Planning & Control, 2005. **16**(3): p. 297 - 308.
46. Kizil M., Ozbayrak M., and Papadopoulou T.C., *Evaluation of dispatching rules for cellular manufacturing.* The International Journal of Advanced Manufacturing Technology, 2006. **28**(9 - 10): p. 985 - 992.

47. Tan B., *Agile manufacturing and management of variability.* International transactions in operational Research, 1998. **5**(5): p. 375 - 388.
48. Shu L., Tang T., and Collins D.W., *Minimum inventory variability schedule with applications in semiconductor fabrication.* Semiconductor Manufacturing, IEEE Transactions, 1996. **9**(1): p. 145 - 149.
49. John H. Blackstone, Don T. Phillips, and Hogg G.L., *A state-of-the-art survey of dispactching rules for manufacturing job shop operations.* International Journal of Production Research, 1982. **20**(1): p. 27 - 45.
50. Eilon S. and Cotterill D.J., *A modified SI rule in job shop sequencing.* International Journal of Production Research, 1968. **7**(2): p. 135.
51. Eilon S., Chowdhury I.G., and Serghiou S.S., *Experiments with SI rule in job shop scheduling.* Simulation, 1975. **24**(2): p. 45-48.
52. Oral M. and Malouin J.L., *Evaluation of the shortest processing time scheduling rule with truncation process.* IIE Transactions, 1973. **5**(4): p. 357 - 365.
53. Upasani A.A., Uzsoy R., and Sourirajan K., *A problem reduction approach for scheduling semiconductor wafer fabrication facilities.* Semiconductor Manufacturing, IEEE Transactions 2006. **19**(2): p. 216 - 225.
54. Conway R.W., *Priority dispatching and job lateness in a job shop.* Journal of Industrial Engineering, 1965 a. **16**(4): p. 228.
55. Rochette R. and Sadowski R.P., *A statistical comparison of the performance of simple dispatching rules for a particular set of job shops.* International Journal of Production Research, 1976. **14**(1): p. 63.
56. Conway R.W., *Priority dispatching and work-in-process inventory in a job shop.* Journal of Industrial Engineering, 1965 b. **16**(2): p. 123.
57. Collins D.W., Williams K., and Hoppensteadt F. *Implementation of minimum inventory variability scheduling 1-step ahead policy in a large semiconductor manufacturing facility*. in *Emerging Technologies and Factory Automation Proceedings.* 1997.
58. Flores-Godoy J.-J., Wang Y., Collins D.W., Hoppensteadt F., and Tsakalis K. *A mini-fab simulation model comparing FIFO and MIVP schedule policies (outer loop), and PID and H machine controllers (inner loop) for semiconductor diffusion bay maintenance*. in *Proceedings of the 24th Annual Conference of the IEEE*. 1998.
59. Spearman M.L., Woodruff D.L., and Hopp W.J., *CONWIP: A pull alternative to Kanban.* International Journal of Production Research, 1990. **28**(5): p. 879 - 894.
60. Krishnamurthy A., Suri R., and Vernon M., *Re-examining the performance of MRP and Kanban material control strategies for multi-product flexible manufacturing system.* The International Journal of Flexible Manufacturing Systems, 2004. **16**(2): p. 123 -150.
61. Chow L.K. and Ong E.C. *A Novel Push-Pull sampling methodology for test production in semiconductor manufacturing Industries*. in *Electronic Manufacturing Technology Symposium (IEMT) 33rd*. 2008.
62. Gilland W.G., *A simulation study comparing performance of CONWIP and bottleneck-based release rules.* Production Planning & Control, 2002. **13**(2): p. 211-219.

63. Suri R. and Sanders J.L., *Performance evaluation of production networks*. Handbooks in OR and MS, ed. S.C. Graves. 1993, Amsterdam: Elsevier Sciences Publishers.
64. Liberopoulos G. and Dallery Y., *A unified framework for pull control mechanisms in multi-stage manufacturing systems.* Annals of Operations Research, 2000. **93**(1-4): p. 325 - 355.
65. Panayiotou C.G. and Cassandras C.G., *Optimization of kanban-based manufacturing systems.* Automatica, 1999. **35**(9): p. 1521 - 1533.
66. Spearman M.L. and Zazanis M.A., *Push and Pull Production Systems: Issues and Comparions.* Operations Research, 1990. **40**(3): p. 521.
67. Aytug H. and Dogan C.A., *A framework and a simulation generator for kanban-controlled manufacturing systems.* Computers Industry Engineering, 1998. **34**(2): p. 337 - 350.
68. Kumar C.S. and Panneerselvam R., *Literature review of JIT-Kanban system.* The International Journal of Advanced Manufacturing Technology, 2007. **32**(3-4): p. 394 - 408.
69. Dallery Y. and Liberopoulos G., *Extended kanban control : Combining kanban and base stock.* IIE Transactions, 2000. **32**(4): p. 369 - 386.
70. Gilland W.G., *A simulation study comparing performance of CONWIP and bottleneck-based release rules.* Production Planning & Control, 2002. **13**(2): p. 211 - 219.
71. Rose O. *CONWIP-like lot release for a wafer fabrication facility with dynamic loads changes*. in *Proc. SMOMS 01*. 2001.
72. Spearman M.L., Eoodruff D.L., and Hopp W.J., *CONWIP: A pull alternative to kanban.* International Journal of Production Research, 1990. **28**(5): p. 879 - 894.
73. Otenti S. *A modified Kanban system in a semiconductor manufacturing environment*. in *Advanced Semiconductor Manufacturing Conference and Workshop. ASMC 91 Proceedings. IEEE/SEMI*. 1991.
74. Chao Qi, Appa Iyer Sivakumar, and Gershwin S.B., *An efficient new job release control methology.* International Journal of Production Research, 2009. **47**(3): p. 703 - 731.
75. Chang T.M. and Yih Y., *Determining the number of kanbans and lot sizes in a generic kanban system: A simulated annealing approach.* International Journal of Production Research, 1994. **32**(8): p. 1991 - 1994.
76. Chang T.M. and Yih Y., *Generic kanban systems for dynamic environments.* International Journal of Production Research, 1994. **32**(4): p. 889.
77. Frein Y., Mascolo M.D., and Dallery Y., *On the design of generalized kanban control systems.* International Journal of Operations & Production Management, 1995. **15**(9): p. 158-184.
78. Goldratt E.M. and Cox. J., *The Goal: A process of ongoing improvement*. 1986: Croton-on-Hudson, NY: North River Press.
79. Glassey C.R. and Resende M.G.C. *Closed-loop job release control for VLSI circuit manufacturing*. in *Semiconductor Manufacturing, IEEE*. 1988.
80. Zhongjie Wang and Chen J. *Release control for hot orders based on TOC theory for semiconductor manufacturing line*. in *Asian Control Conference*. 2009.

81. Rose O. *CONLOAD - A new lot release rule for semiconductor wafer fabs*. in *Proceedings of the 31st conference on Winter Simulation*. 1999.
82. Kayton D., Semicon. H., and Findlay O. *Using the theory of constraint's production application in a semiconductor fab with a reentrant bottleneck*. in *Electronics Manufacturing Technology Symposium, IEEE/CPMT*. 1998.
83. Glassey C.R. and Resende M.G.C., *A scheduling rule for job release in semiconductor fabrication.* Operation Research Letters, 1988. **7**(5): p. 213 - 217.
84. Jain S. and Chan S. *Experiences with backward simulation based approach for lot release planning*. in *IEEE Computer Society*. 1997.
85. Enns S.T. and Costa M.P., *The effectivness of input control based on aggregate versus bottleneck work loads.* Production Planning & Control, 2002. **13**(7): p. 614 - 624.
86. Lawrence W.M. *Scheduling semiconductor wafer fabrication*. in *IEEE Transactions on Semiconductor Manufacturing*. 1988.
87. Ghrayeb O., Phpjanamongkolkij N., and Tan B.A., *A hybrid push/pull system in assemble-to-order manufacturing environment.* Journal of Intelligent Manufacturing, 2009. **20**(4): p. 379 - 387.
88. Xiong G. and Nyberg T.R., *Push/pull production plan and schedule used in modern refinery CIMS.* Robotics and Computer-Integrated Manufacturing, 2000. **16**(6): p. 397 - 410.
89. Huang C.C. and Kusiak A., *Manufacturing control with a push-pull approach.* International Journal of Production Research, 1998. **36**(1): p. 251 - 275.
90. Dr G.X., Xiong G.-Y., and Dr. T.R.N. *Push/pull based production plan & Schedule strategy*. in *Emerging Technologies and Factory Automation*. 2001.
91. Spearman M.L. and Zazanis M.A., *Push and pull production systems: issues and comparisons.* Operations Research, 1992. **40**(3): p. 521.
92. Hall R.W., *Zero Inventories*. 1983: Homewood, IL: Down Jones-Irwin.
93. Geraghty J. and Heavey C., *A review and comparison of hybrid and pull-type production control strategies.* OR Spectrum, 2005. **27**(2-3): p. 435 - 457.
94. Bob Diamond, Steve Lamperti, Dave Krahl, and Nastasi A., *Extend V6: Professional simulation tools*. 2002.
95. Law A.M. and Law A.M., *Simulation modeling and analysis* 3rd ed. McGraw-Hill series in industrial engineering and management science. 2000: Boston : McGraw-Hill, c2000.
96. Michael Quirk and Serda J., *Semiconductor manufacturing technology*. International Edition ed. 2000: Pearson Education, Limited, 200. 666.
97. Murray S., Young P., Geraghty J., and Sievwright S.S. *Impact of lot release strategies on 'make-to-order' production line performance*. in *Semiconductor Manufacturing, ISSM 2007*. 2007.
98. Montgomery D.C., Rungen G.C., and Hubele N.F., *Engineering statistics*. 1998: John Wiley & Sons, Inc.
99. Davies O.L. and Goldsmith P.L., *Statistical methods in Research and Production*. 1972.
100. Smith D.L. and Naberejnev D.G., *Confidence intervals for the lognormal probability distribution.* Nuclear Instruments and Method in Physics Research

Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2004. **518**(3): p. 754 - 763.

101. Smith B.E. and Merceret F.J., *The lognormal distribution.* The College Mathematics Journal, 2000. **31**(4): p. 259 - 261.
102. Gravetter F.J. and Wallanu L.B., *Statistics for the Behavioural Sconces*. Four Edition ed. 1996.
103. Relilly J., *Understanding Statistics: and its applications in Business, Science and Engineering*. 1997.
104. Martin D.P. *How the law of unanticipated consequences can nullify the theory of constraint: The case for balanced capacity in a semiconductor manufacturing line*. in *Advanced Semiconductor Manufacturing Conference and Workshop, IEEE/SEMI Publication*. 1997.
105. Martin D.P. *The advantage of using short cycle time manufacturing (SCM) instead of continuous flow manufacturing (CFM)*. in *Advanced Semiconductor Manufacturing Conference and Workshop*. 1998.
106. Martin D.P. *Maximizing productivity improvements using short cycle time manufacturing (SCM) concepts in a semiconductor manufacturing line*. in *Advanced Semiconductor Manufacturing and Workshop, IEEE/SEMI*. 2000.
107. Jacobs J.H., Etman L.F.P., Rooda J.E., and Van Compen E.J.J. *Quantifying operational time variability: The missing parameter for cycle time reduction*. in *Advanced Semiconductor Manufacturing Conference, IEEE/SEMI*. 2001.
108. Atherton R.W., Turner F.T., Atherton L.F., and Pool M.A. *Performance analysis of multi-process semiconductor manufacturing equipment*. in *Advanced semiconductor Manufacturing Conference and Workshop, ASMA proceedings, IEEE/SEMI*. 1990.

APPENDIX - A RANDOM EVENT THEORY ANALYSIS

The field of statistics deals with the collection, presentation, analysis, and use of data to make decisions and solve problems. Because many aspects of engineering practice involve working with data, obviously some knowledge of statistics is important to any researcher. Statistical techniques can be a powerful aid in developing and improving production processes. Statistical methods are used to help us describe and understand "variability". By variability, we mean that successive observations of a system or phenomenon do not produce exactly the same result. We all encounter variability in our every day lives, and statistical thinking can give us a useful way to incorporate this variability into our decision making processes. Statistics gives us a framework for describing this variability and for learning about which potential sources of variability are the most important or which have the greatest impact on performance [98].

A.1 Probability Distribution

In a random experiment, a variable whose measured value can change ─ from one replicate of the experiment to another ─ is referred to as a random variable (X). When a number of repeat measurements are made, they may be regarded as a sample of the results from the population of results which might have been obtained. From such a sample of observations, we can calculate the sample mean and standard deviation, which are estimates of the population or true value. The probability distribution describes the range of possible values that a random variable can attain and the probability that the value of the random variable is within any (measurable) subset of that range. There are

various probability distributions, such as normal, lognormal or exponential, that show up in various different applications. [99]

The statistics most commonly used to represent the properties of a distribution fall into the following categories:

A.1.1 Measure of Location or Central Value

This measure gives the location of some central or typical value. An example is the arithmetic mean; it is simply the sum of all the observation divided by their number. [99]

Arithmetic mean = $\bar{x} = \sum x / N$ **Equation A.1**

A.1.2 Measure of Dispersion

This measure shows the degree of spread of the data round the central value. An example is the standard deviation (σ), positive squared root of the variance where the variance (V) of a population is the mean squared deviation of the individual values from the population mean. [99]

$$\sigma^2 = V = \frac{\sum (x - \mu)^2}{N}$$ **Equation A.2**

Where,

μ : Mean ($\bar{x}$)

x : Value of distribution

N : Number of the value

A.1.3 Measure of Skewness

Skewness means lack of symmetry, and measures of Skewness show the extent to which the distribution departs from symmetry. A distribution will not in general be completely symmetrical; the frequency may fall away more rapidly on one side of the mode than on the other. When this is so the distribution is said to be skewed. The distribution shown in Figure A.1 is described as Positively Skewed, because the long tail is on the side of the high values of x. Similarly, if the long tail is on the side of low values of x, the distribution is said to be Negatively Skewed (Figure A.2). Positive Skewness is more common than negative; for example, the distribution of the number of items waiting in a queue and the distribution of molecular chain lengths in a polymer usually exhibits this shape. If a distribution shows a large Skewness, then mean and standard deviation are not really useful. Instead a more practical solution is to resort to the use of well-defined confidence intervals [99, 100]. Moreover, it also implies that such distribution analysis requires large samples to obtain a good representation of the tail [101].

In the case of Skewness, we need to distinguish between mean, mode, and median. One measure of Skewness (Pearson's Skewness) is defined by:

$$\frac{Mean - Mode}{\sigma}$$ **Equation A.3**

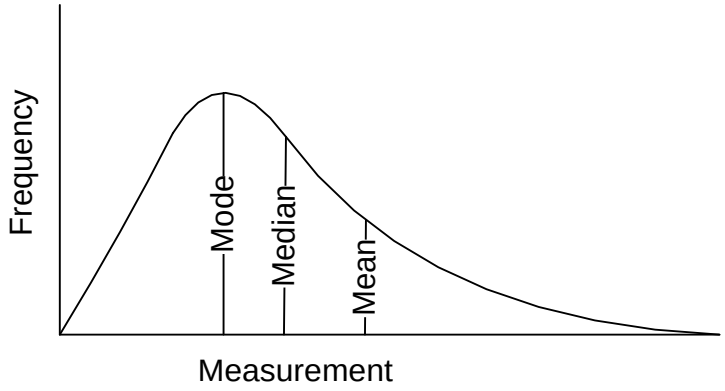


Figure A.1: Location statistics for distribution with positive Skewness

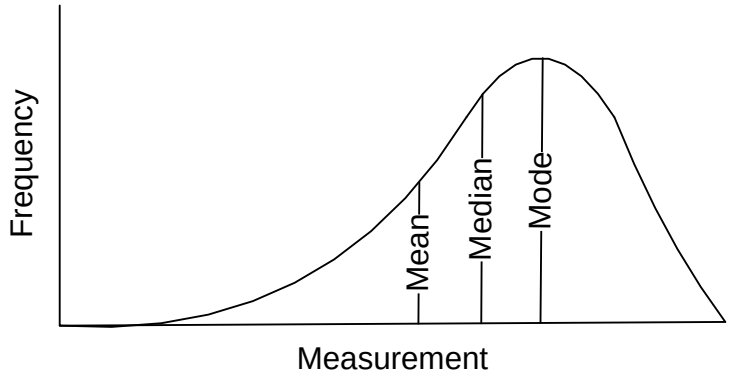


Figure A.2: Location statistics for distribution with negative Skewness

Mean

The mean, commonly known as the arithmetic average, is computed by adding all the scores in the distribution and dividing by the numbers of scores.

Mode

The mode is the value of the variate which occurs most frequently, i.e. for which the frequency is a maximum. In a frequency distribution, the mode is the score or category that has the greatest frequency. There are no symbols or special notation used to identify the mode or to differentiate between a sample mode and a population mode. In addition, the definition of the mode is the same for a population and for a sample distribution. An

approximate value can be obtained by plotting a frequency diagram, drawing a smooth curve through it and noting the point of maximum frequency.

Median

If the data are arranged in order of magnitude, the median is the central member of the series, i.e. there are equal numbers of observations greater than and less than the median. The median is the score that divides a distribution exactly in half. Exactly 50% of the individuals in a distribution have scores at or below the median. There are no symbols or notation, instead, the median is simply identified by the word median. [99]

A.1.4 Measures of Kurtosis

Kurtosis may be defined as "peakedness", and a measure of kurtosis serves to differentiate between a flat distribution curve, and a sharply peaked curve. [99]

A.1.5 Exponential Distribution

Consider machine downtimes that occur on the production line. This is an example of events (such as downtime) that occurs randomly in an interval (such as time). The number of events over interval (such as number of downtime that occur in one hour) is discrete random variable that is often modeled by a Poisson distribution. The length of the interval between events (such as the time between downtime) is often modeled by an Exponential distribution.

The exponential distribution is often used in reliability studies as the model for the time until failure of device. [98]

A.1.6 Lognormal Distribution

The lognormal distribution is very useful in representing inherently positively skewed continuous variables, particularly when knowledge of these variables is limited to estimates of the mean value and standard deviation [100]. Indeed the lognormal distribution tends to approximate well, for a wide range of conditions, empirical outcomes that can be thought of as the multiplicative product of many independent random positive variables or perturbations. Because of the skewness of the lognormal distribution, it is necessary to use very large samples to obtain accurate estimates of the parameters of these distributions. Typically, samples of size 20,000 or more would be reasonable [101].

A.2 Coefficient of Variation (CV)

The coefficient of variation is the standard deviation expressed as a percentage of the arithmetic mean:

$$C.V. = \left(\sigma / \mu \right) \times 100 \qquad \textbf{Equation A.4}$$

It is regarded as a measure of stability or uncertainty, and can indicate the relative dispersion of data in the population to the population mean. The main use of the coefficient of variation is to compare the variability of groups of observation with widely differing mean levels. It is also invaluable when dealing with properties whose standard deviation rises in proportion to the mean. It is a dimensionless measure of scatter or dispersion and it is readily interpretable, as opposed to other commonly used

measures such as standard deviation, mean absolute deviation or error factor, which are only interpretable for the lognormal distribution. [11, 99]

A.3 Correlation

In general statistical usage, correlation or co-relation refers to the departure of two random variables from independence. Correlation is a statistical technique that is used to measure and describe the relationship between these two variables. In this broad sense, there are several coefficients, measuring the degree of correlation, adapted to the nature of the data. The best known is the Pearson product-moment correlation coefficient, which is obtained by dividing the covariance of the two variables by the product of their standard deviations. Correlation requires two scores for each individual (one score from each of the two variables), denoted by X and Y score (they present graphically in a scatter plot). A correlation measures three characteristics of the relationship between X and Y: the direction, the form and the degree of the relationship. [102, 103]

A.3.1 The Direction of the Relationship

Correlations can be classified into two basic categories: Positive and Negative. The direction of a relationship is defined by the sign of the correlation. A positive value (+) indicates a positive relationship; a negative value (–) indicates a negative relationship.

Positive Correlation

The two variables tend to move in the same direction: When the X variable increase, the Y variable also increases; if the X variable decreases, the Y variable also decreases. For example, a number of temperatures, measured on both the Celsius (C) and Fahrenheit (F) scales have a positive correlation of 1.

X (C): 180, 200, 230, 250, 280

Y (F): 356, 392, 446, 482, 536

Figure A.3 is clearly showing the natural characteristic of positive correlation.

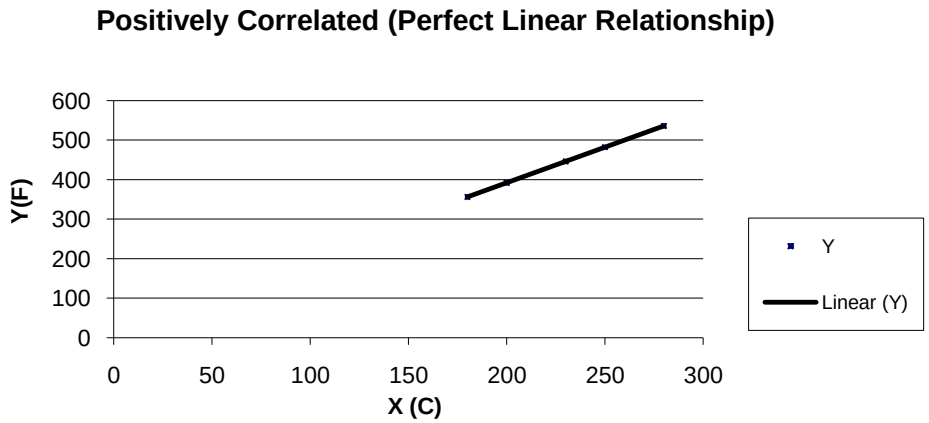


Figure A.3: Positive Correlation

Figure A.3 shows that higher values of Celsius temperature are associated with higher values of Fahrenheit temperature: we say that the two variables are positively correlated. A straight line could be drawn through all the points without missing any: we say that the two variables are perfectly correlated.

Negative Correlation

The two variables tend to go in opposite directions. As the X variable increases, the Y variable decreases. That is, it is an inverse relationship. For example, the two related variables concerning the age (in years) and the value (in £) of a machine have a negative correlation of -1. The machine was purchased for £12,000 and £ 2,000 was written off its value each year.

X: 0, 1, 2, 3, 4, 5, 6.

Y: 12000, 10000, 8000, 6000, 4000, 2000, 0.

Figure A.4is clearly showing the natural characteristic of negative correlation.

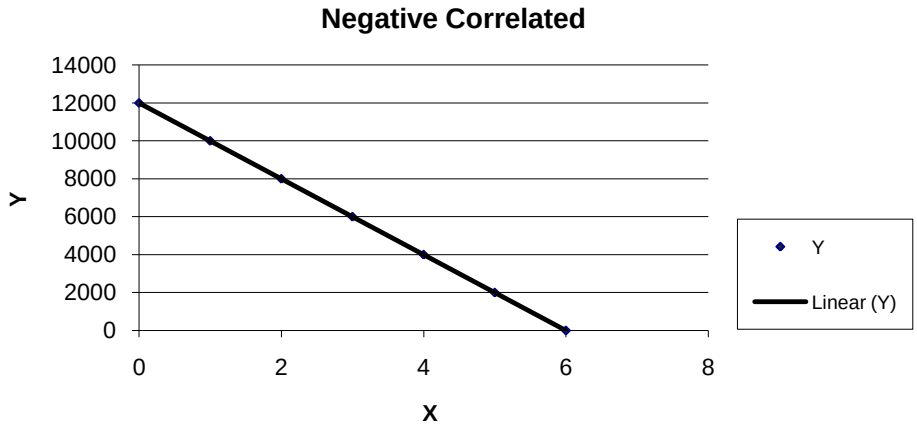


Figure A.4: Negative Correlation

These two variables are also perfectly correlated, but this time the correlation is negative: the higher the age, the lower value. [102, 103]

A.3.2 The Form of the Relationship

The most common use of correlation is to measure straight-line relationship. However, we should note that other forms of relationship do exist and that special correlations are used to measure them.

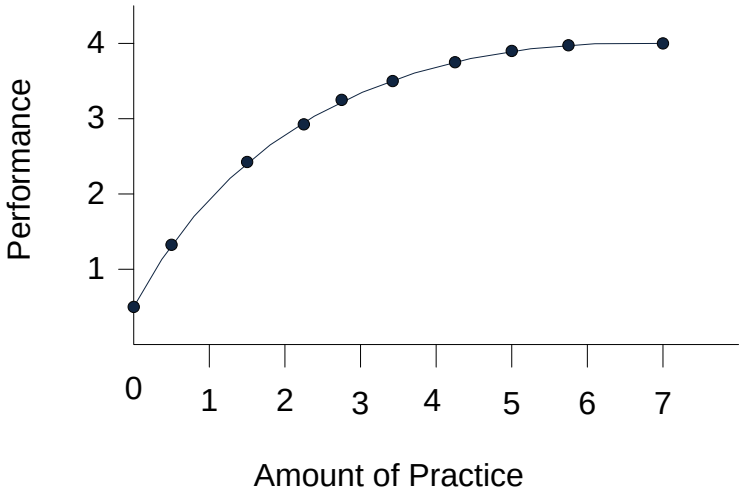


Figure A.5: Relationship between performance and amount of practice

Figure A.5 shows the typical relationship between practice and performance. This is not a straight-line relationship. The graphic is gradually increasing. This means that with a great deal of practice, the improvement in performance becomes less noticeable.

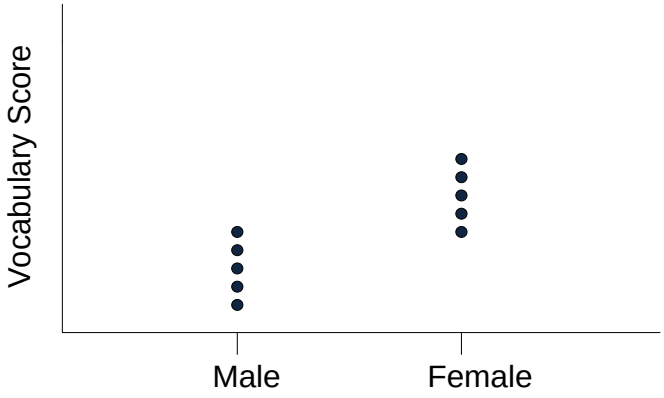


Figure A.6: Relationship between vocabulary score and gender

Figure A.6 shows the relationship between vocabulary scores and gender for five-year-old children. Again, this is not a straight line relationship. These data show a tendency for females to have a higher score than males. Many different types of correlations exist; each one is designed to evaluate a specific form of relationship. In this study, we will concentrate on linear correlation. [102, 103]

A.3.3 The Degree of the Relationship

A correlation measures how well the data fit the specific form being considered. For example, a linear correlation measures how well the data points fit on a straight line. A perfect correlation always is identified by a correlation of 1 and indicates a perfect fit, whereas a correlation of 0 indicates no fit at all. Intermediate values represent the degree to which the data points approximate the perfect fit. The numerical value of the correlation also reflects the degree to which it is consistent to predict a relationship between the two variables. Again, a correlation of 1 (or -1) indicates a perfectly consistent relationship. [102, 103] Examples of different values for linear correlation are shown in Figure A.7, Figure A.8, Figure A.9 and Figure A.10.

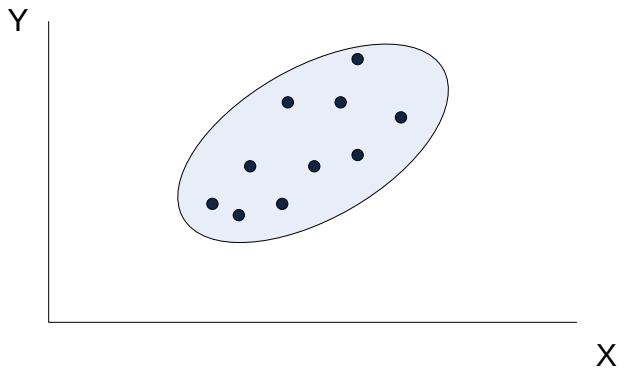


Figure A.7: A strong positive relationship, approximately + 0.90

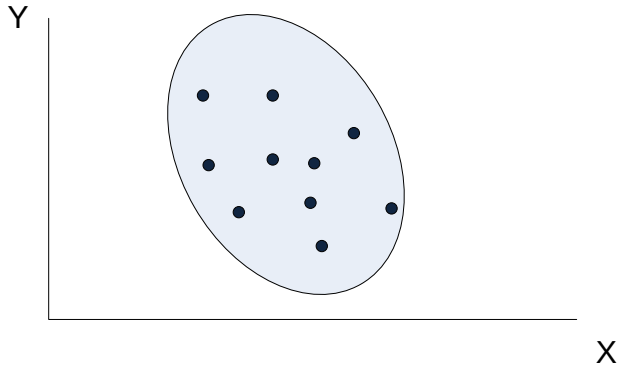


Figure A.8: A relatively weak negative correlation, approximately -0.40

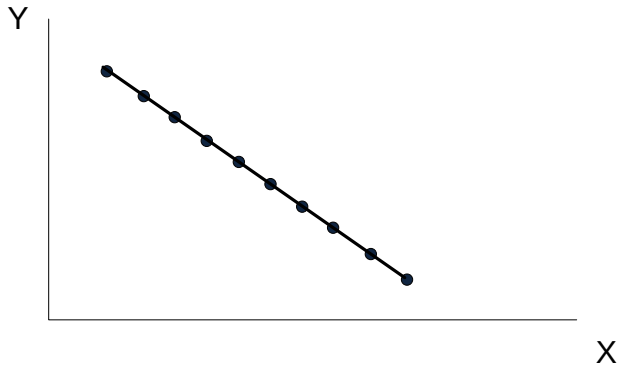


Figure A.9: A perfect negative correlation, -1.00

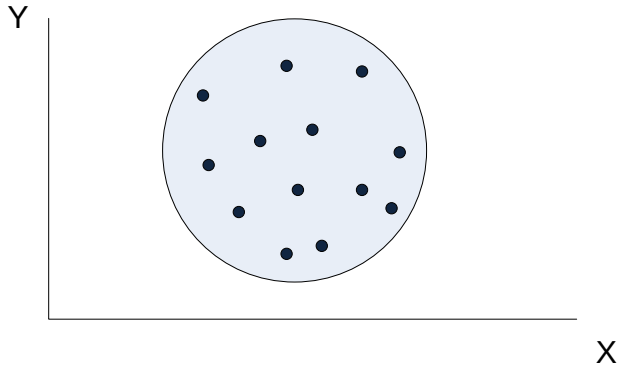


Figure A.10: No linear trend, 0.00

A.3.4 The Correlation Coefficient

Sometimes we need to find out if a linear relationship exists between two variables. Also it can be useful to have a statistic which measures the degree of linearity. The correlation coefficient (or Pearson's product-moment correlation coefficient), denoted

by γ , is such a statistic. The value of γ always lies between -1 and 1. A positive value for γ indicates positive correlation; a negative value for γ indicates negative correlation. The magnitude of γ ($|\gamma|$) indicates the strength of the correlation: values close to zero indicate that the correlation is weak (i.e. the points are widely scattered); values close to 1 or -1 indicate that the correlation is strong (i.e. the points lie close to a straight line): a value of 1, or -1, indicates a perfect linear relationship. For example, in the case of the "temperature data" presented earlier, $\gamma = +1$. [102, 103]

$$\gamma = \frac{n\sum xy - \sum x \sum y}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right] \times \left[n\sum y^2 - \left(\sum y\right)^2\right]}}$$ **Equation A.5**

The formula above is γ calculation. Many softwares, such as Excel, can automatically calculate the value of γ when used in statistical mode.

A.4 Data Analysis Statistics

A.4.1 Mean Cycle Time

The mean Cycle Time ($\overline{CT}$) is the average time from release of a job at the beginning of the line until it reaches the end and is given by Equation A.6 below. Equation A.7 gives the cycle time of item *i.*

$$\overline{CT} = \frac{\sum_{i=1}^{n} CT_i}{n}$$ **Equation A.6**

Where:

n: Total number of items

CT_i: Cycle time of item i

$$CT_i = \sum_{j=1}^{6} T_{e_i}^{j} + T_{q_i}^{j} \qquad \textbf{Equation A.7}$$

Where:

$T_{e_i}^{j}$: processing times of item i in Operation j

$T_{q_i}^{j}$: Queue time of item i in Buffer j

i: Item number

j: Operation number

A.4.2 Coefficient of Variation (CV) Cycle Time

One relative measure of the variability of a random variable is the standard deviation divided by the mean, which is called the coefficient of variation (CV). Equation A.8 below provides the means to determine the coefficient of variation of the cycle time.

$$CV_{CT} = \frac{\sigma_{CT}}{\overline{CT}} \qquad \textbf{Equation A.8}$$

A.4.3 Mean Processing Time and Coefficient of Variation Processing Time

Equation A.9 is used to determine the mean processing time of items in the system and is essentially the mean cycle time of items in the system minus the mean queuing time of items. Equation A.11 determines the processing time variability.

$$T_e = \frac{\sum_{i=1}^{n} T_{e_i}}{n}$$ **Equation A.9**

Where:

T_e : Mean Processing Time of items in the system

T_{e_i} : Processing time of item *i*

n: Total number of items

i: Item number

$$t_{e_j} = \frac{\sum_{i=1}^{n} t_{e_i}^{j}}{n}$$ **Equation A.10**

Where:

t_{e_j} : Mean Processing Time of items at Operation *j*

$t_{e_i}^{j}$: Processing time of item *i* at Operation *j*

j: Operation number.

$$CV_{PT} = \frac{\sigma_{PT}}{T_e}$$ **Equation A.11**

Where:

σ_{PT} : Standard deviation of processing time

A.4.4 Mean Queue Time

The mean queuing time of items in the system is given by Equation A.12 and Equation A.13 gives the mean queuing time of items at a given Operation j.

$$t_q = \sum_{j=1}^{6} t_q^j$$ **Equation A.12**

$$t_q^j = \frac{\sum_{i=1}^{n} T_{q_i}^j}{n}$$ **Equation A.13**

Where:

t_q : Mean queuing time

t_q^j : Mean queuing time of items at Operation *j*

n: Total number of items

i: Item number

j: Operation number

A.4.5 Coefficient of Variation Queue Time

Just as there is variability in processing times, there is also variability in queue times. A reasonable variability measure for queue times can be defined in exactly the same way as for process times. If σ_q is standard deviation of the queue times, then the coefficient of variation of queue times c_q is given by Equation A.14.

$$c_q = \frac{\sigma_q}{t_q}$$ **Equation A.14**

A.4.6 Utilization

The utilization of Operation *j* is given by Equation A.15.

$$U_j = \frac{r_{a_j} t_{e_j}}{m_j}$$ **Equation A.15**

Where:

r_{a_j} : Arrival rate at Operation *j*, or the departure rate from the buffer of the preceding Operation *j*-1.

t_{e_j} : Mean processing time of Operation *j*.

m_j : Number of machines at Operation *j*.

j: Operation number.

A.4.7 Mean Inter-Departure Time

The starting point for studying flow is the arrival of jobs to an operation. The departure from this operation will in turn be arrivals to the following operation. Therefore, to characterize the flow variability for the entire line, first the variability of arrivals to one operation has to be described. Then, its influence on the variability of departures from this same operation has to be determined. Hence arrivals to the following operation will have been described. The first descriptor of departures from an operation is the departure rate, measured in jobs per unit time. The departure rate can be characterized

from Operation *j* by either the mean time between departures, which we denote by t_{d_j} or the average departure rate denoted by r_{d_j}.

A.4.8 Coefficient of Variation Inter-Departure time

Just as there is variability in processing times, there is also variability in inter-departure times. A reasonable variability measure for inter-departure times can be defined in exactly the same way as for process times. If σ_d is standard deviation of the time between departures, then the coefficient of variation of the inter-departure times c_d is given by Equation A.16.

$$c_d = \frac{\sigma_d}{t_d}$$ **Equation A.16**

APPENDIX - B QUEUING THEORY AND OPERATING CURVE

Maximizing productivity and minimizing costs depend on high utilization, high throughput, short cycle time, minimizing stock and minimizing WIP. Unfortunately, utilization, throughput, cycle time and WIP are not independent variables that can be optimized separately. Two fundamental relationships govern the production line.

First, a relationship among WIP, cycle time, and throughput mathematically proved by J. D. C. Little [11]. WIP is equal to the product of throughput and cycle time. Little's Law can be applied to a single station, a line, or an entire fab.

$$WIP = TH \times CT \qquad \textbf{Equation B.1}$$

Second, the relationship introduced by Martin [104], linking the capacity utilization of the line and the cycle time,

$$X = \frac{CT}{RCT} = F() * \frac{1 - UTIL/2}{1 - UTIL} \qquad \textbf{Equation B.2}$$

Where CT is cycle time, RCT is the raw cycle time, F () is a function of tool and operator characteristics, and UTIL is the utilization of available capacity. The RCT of a line is the theoretical minimum amount of time that one lot would take to move from the beginning to the end of the line [9]. The normalized cycle time, which is the average cycle time divided by the raw processing time, is commonly referred to as ‘X factor’ [15].

It can be seen from Equation B.1 that the cycle time increases in a highly non-linear fashion if the loading of the fab increases [11]. Therefore, a trade-off between cycle time performance and throughput needs to be found [9].

The plot of the cycle time versus the loading of the production line is known as the factory operating curve. It provides to managers, the production line response to increased loading level and thus is an effective tool to predict and adjust the performance of the fab. But the exact shape of the plot is partly a function (F term in Equation B.2) of the production line's characteristics.

Therefore, how can the correct curve be determined? A first solution is to monitor and measure the actual production line loading and cycle time, to draw point by point the curve [9]. But while the cycle time is easily obtained from the lot tracking system available in semiconductor fabs, the loading is much less accurate. A second solution is to use waiting line models from the queuing theory literature to determine F() and generate analytical approximation of the operating curve. This has been done taking into consideration various characteristics of the production line such as capacity [5, 14, 15, 104, 105]; re-entrant lines [13]; tool, operator and parts availability [106]; tool dedication [15]; variability [5, 7, 8, 10, 107] or batching [32]. While this method provides insights and understandings of production line behaviors, it is insufficient to generate a curve accurate enough to predict the exact response to any modification brought to the line.

APPENDIX - C PRE-STUDY: DATA RESULTS

C.1 Mean Inter-Departure Time

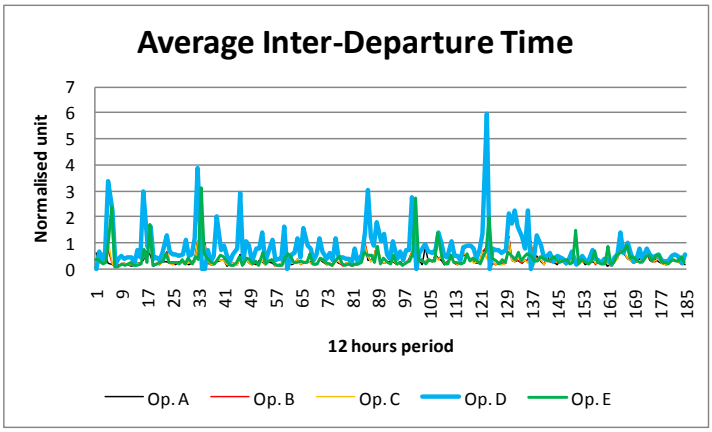


Figure C.1: Average inter-departure time of 12 hours period

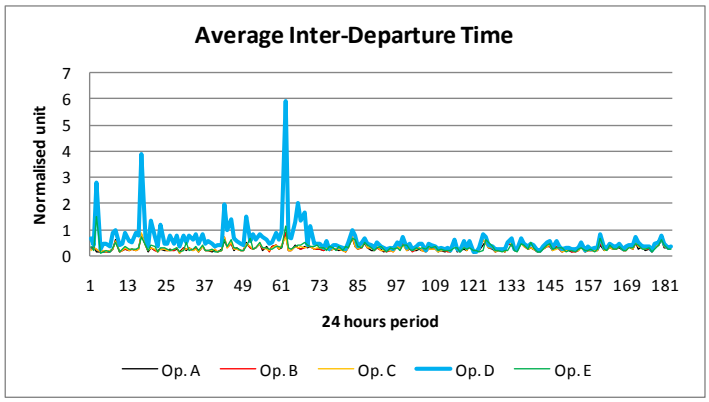


Figure C.2: Average inter-departure time of 24 hours period

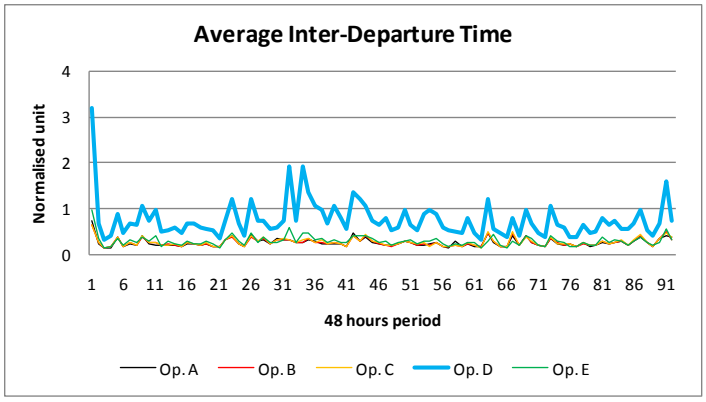


Figure C.3: Average inter-departure time of 48 hours period

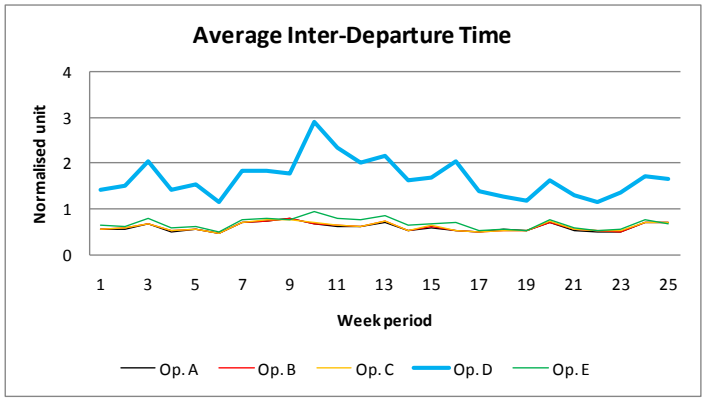


Figure C.4: Average inter-departure time of week period

C.2 Operation B Correlation Coefficient

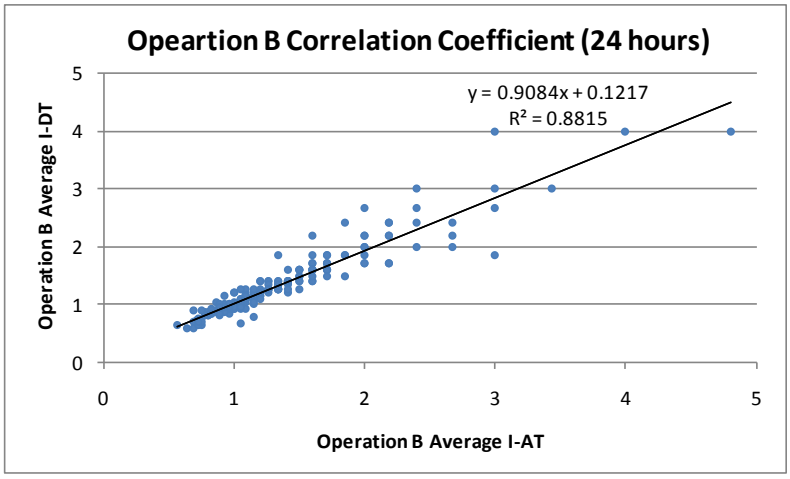


Figure C.5: Operation B correlation coefficient between inter-arrival time and inter-departure time at 24 hours period

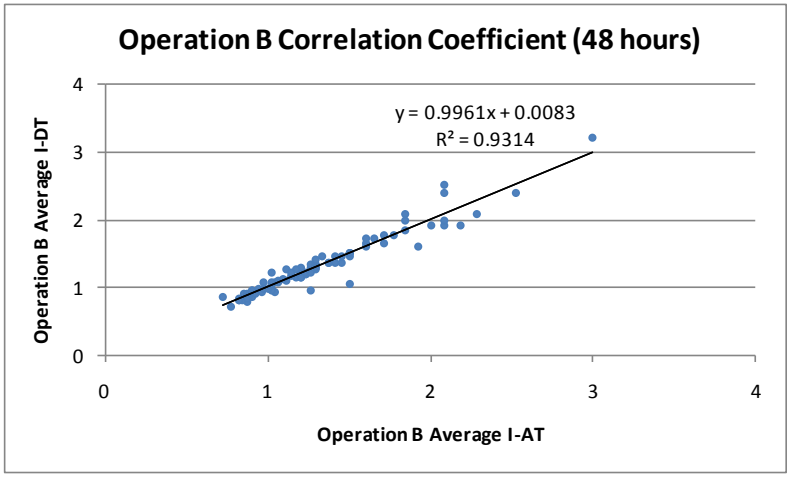


Figure C.6: Operation B correlation coefficient between inter-arrival time and inter-departure time at 48 hours period

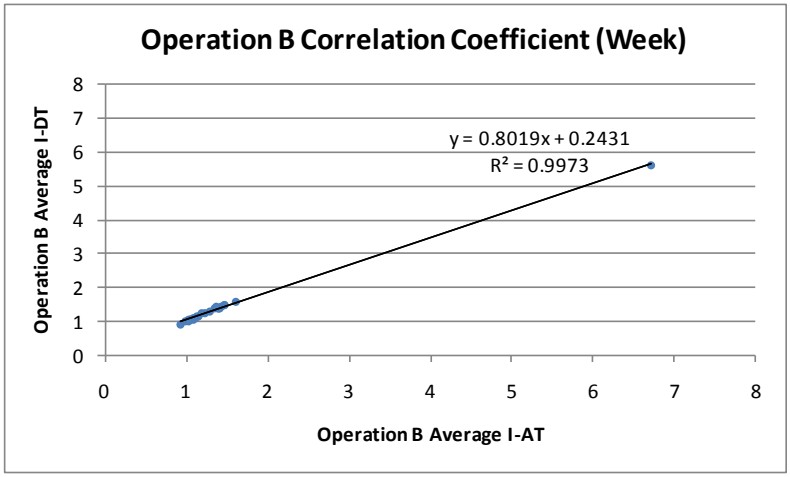


Figure C.7: Operation B correlation coefficient between inter-arrival time and inter-departure time at week period

C.3 Operation C Correlation Coefficient

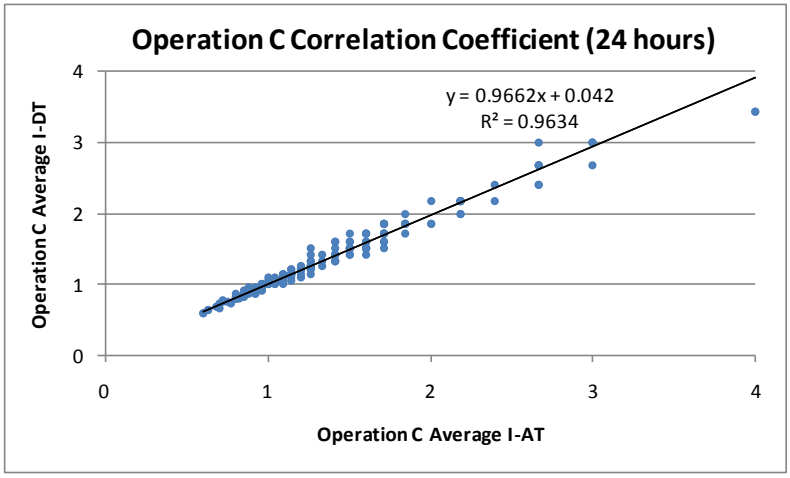


Figure C.8: Operation C correlation coefficient between inter-arrival time and inter-departure time at 24 hours period

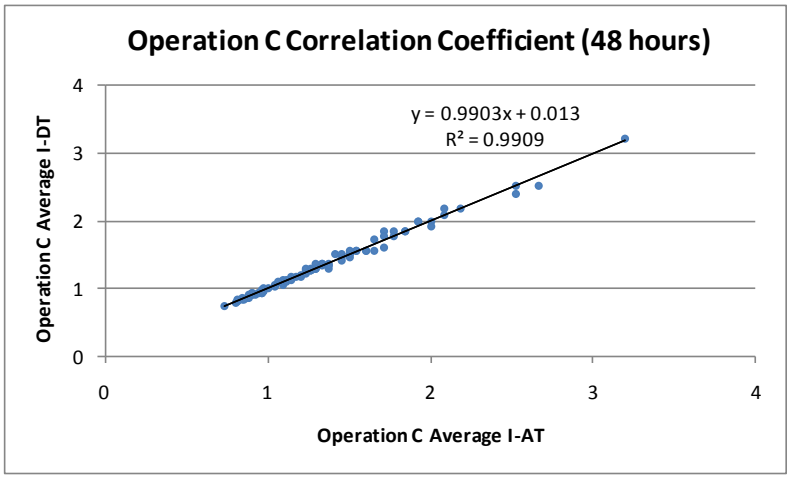


Figure C.9: Operation C correlation coefficient between inter-arrival time and inter-departure time at 48 hours period

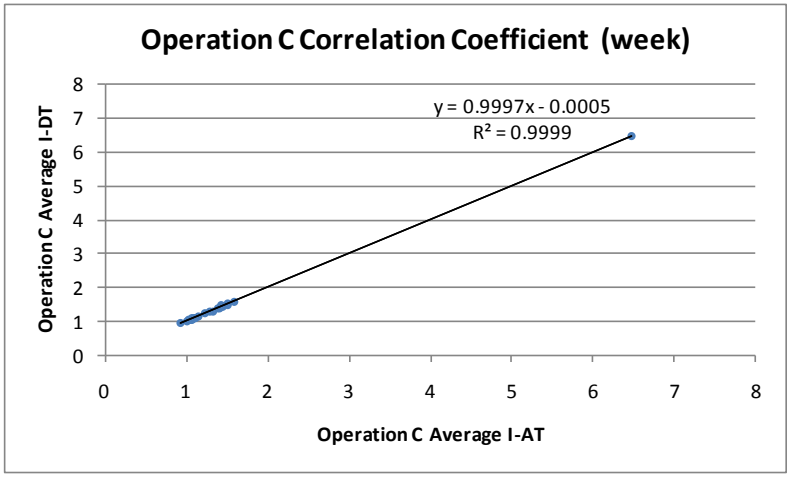


Figure C.10: Operation C correlation coefficient between inter-arrival time and inter-departure time at week period

C.4 Operation D Correlation Coefficient

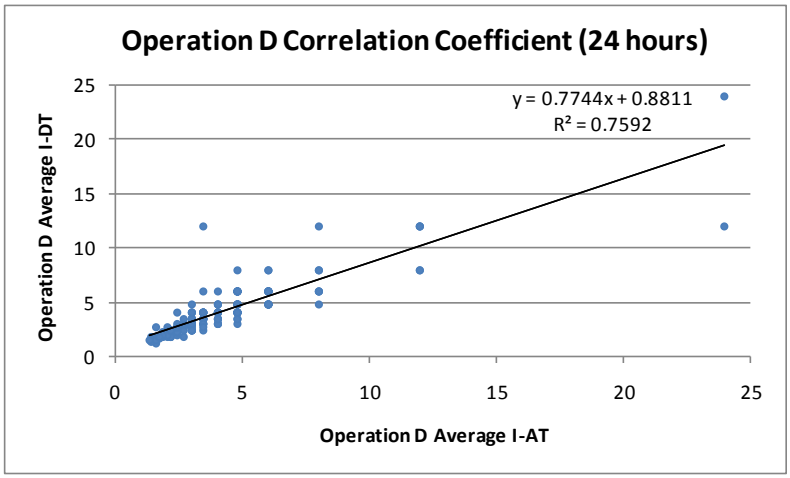


Figure C.11: Operation D correlation coefficient between inter-arrival time and inter-departure time at 24 hours period

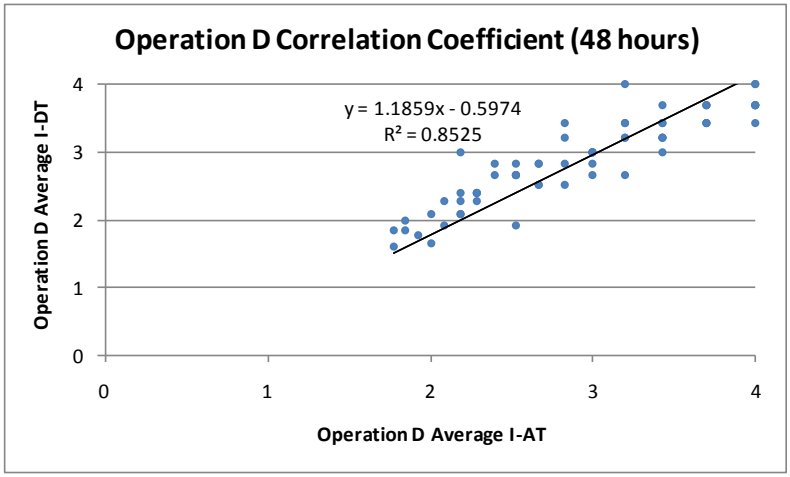


Figure C.12: Operation D correlation coefficient between inter-arrival time and inter-departure time at 48 hours period

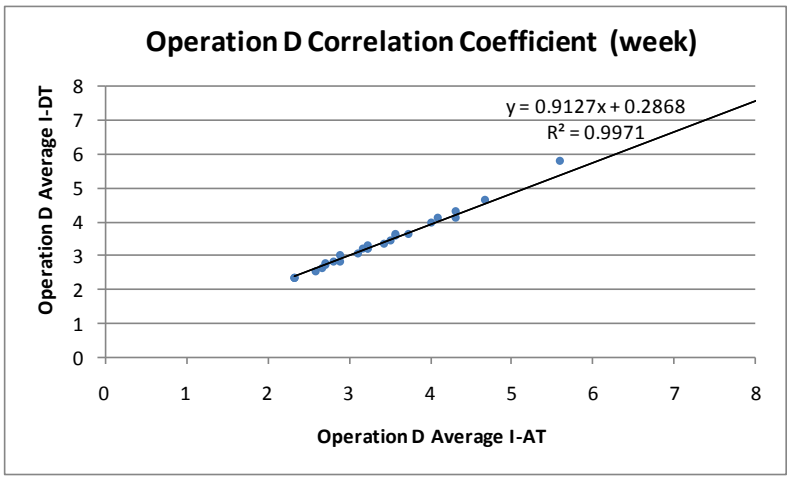


Figure C.13: Operation D correlation coefficient between inter-arrival time and inter-departure time at week period

C.5 Operation E Correlation Coefficient

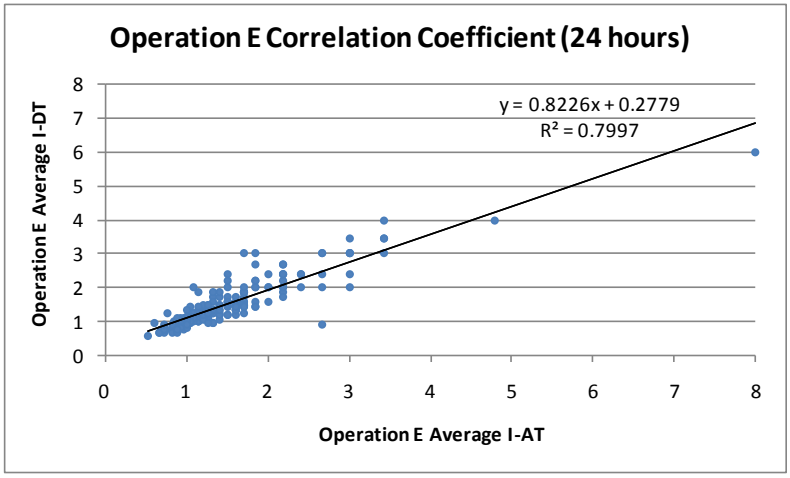


Figure C.14: Operation E correlation coefficient between inter-arrival time and inter-departure time at 24 hours period

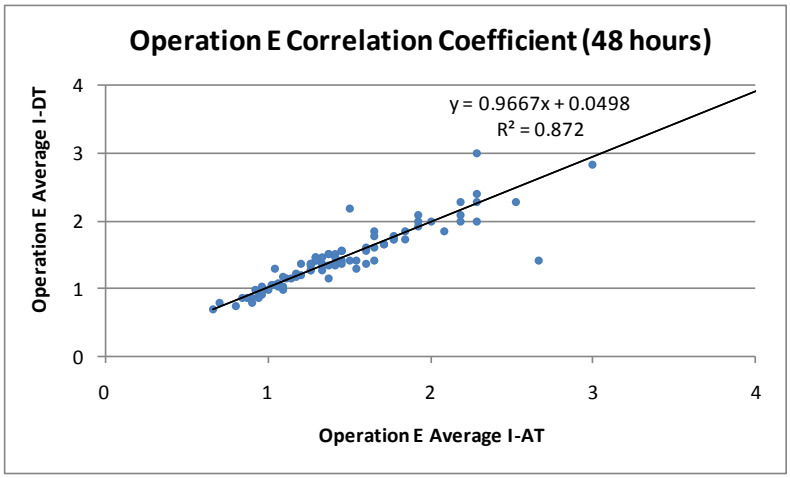


Figure C.15: Operation E correlation coefficient between inter-arrival time and inter-departure time at 48 hours period

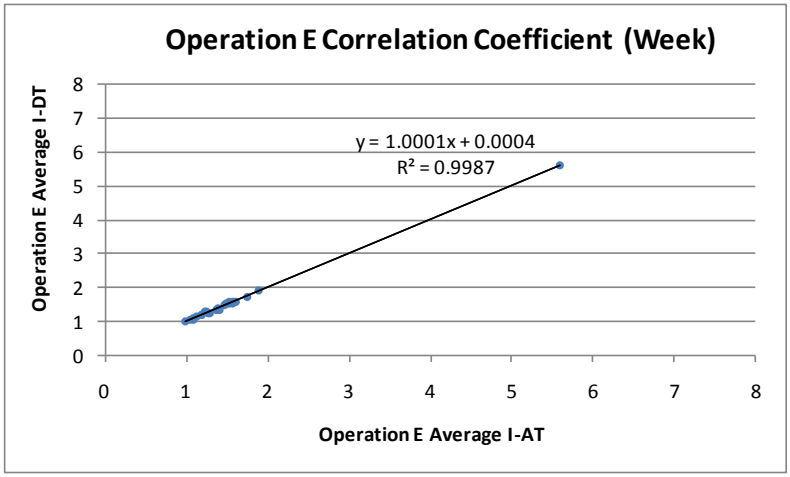


Figure C.16: Operation E correlation coefficient between inter-arrival time and inter-departure time at week period

C.6 Inter-departure/arrival time distribution

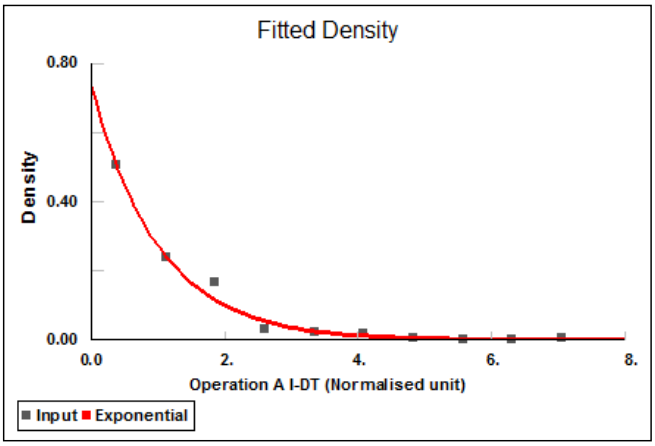


Figure C.17: Fitted distribution for Operation A inter-departure time (Week based)

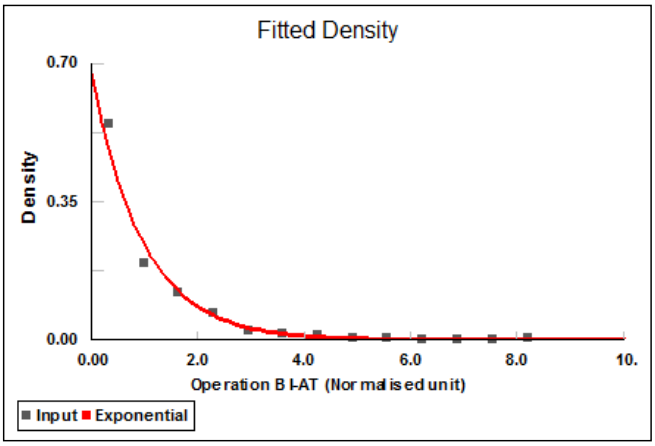


Figure C.18: Fitted distribution for Operation B inter-arrival time (Week based)

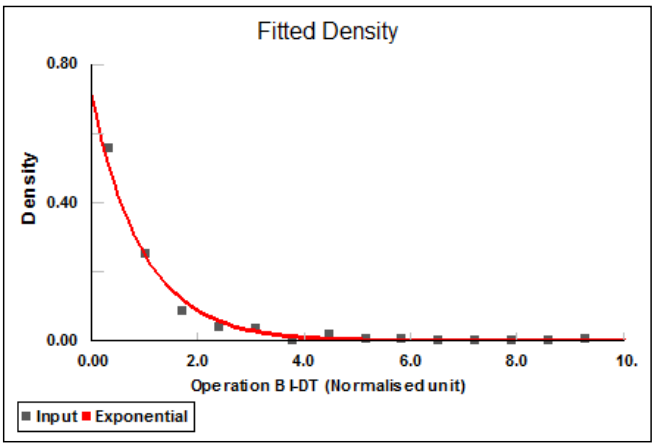


Figure C.19: Fitted distribution for Operation B inter-departure time (Week based)

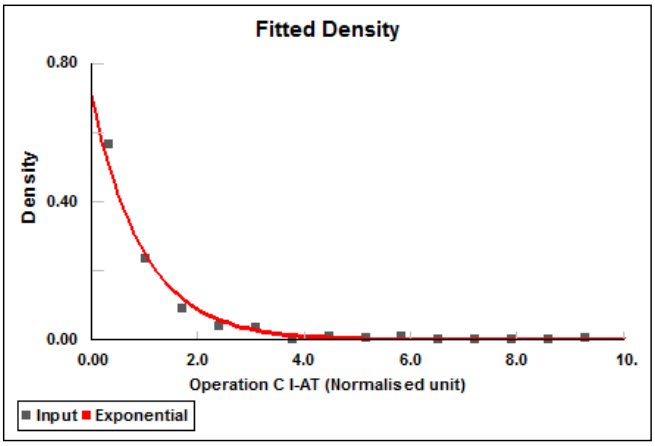


Figure C.20: Fitted distribution for Operation C inter-arrival time (Week based)

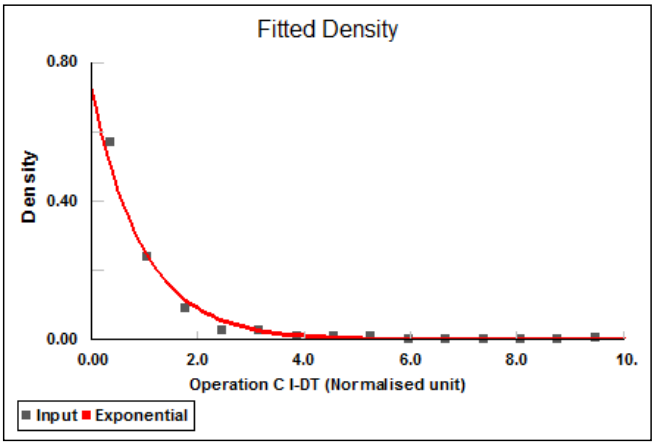


Figure C.21: Fitted distribution for Operation C inter-departure time (Week based)

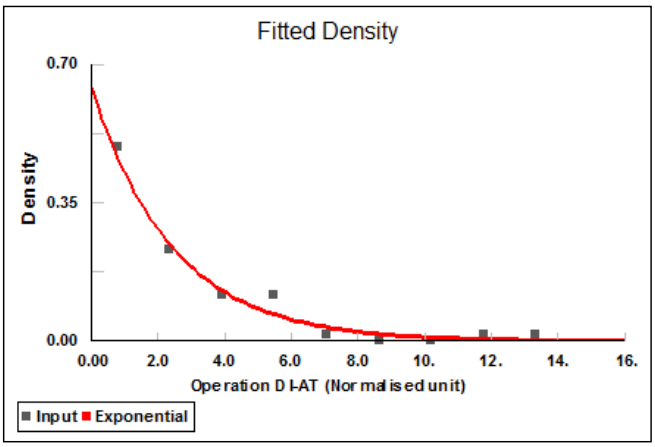


Figure C.22: Fitted distribution for Operation D inter-arrival time (Week based)

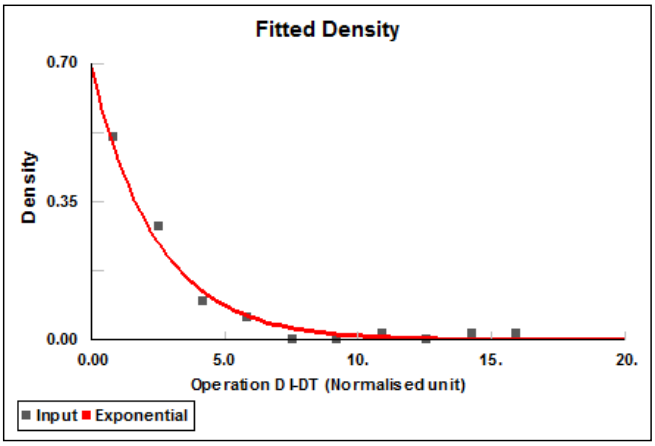


Figure C.23: Fitted distribution for Operation D inter-departure time (Week based)

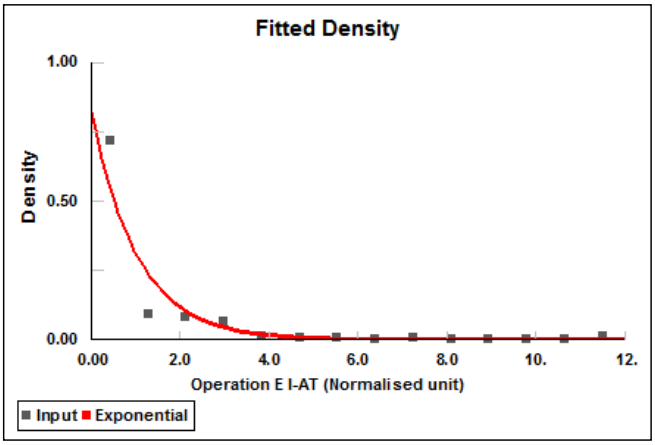


Figure C.24: Fitted distribution for Operation E inter-arrival time (Week based)

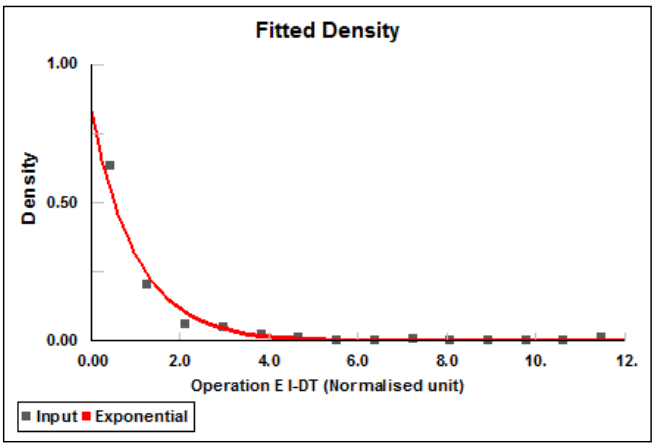


Figure C.25: Fitted distribution for Operation E inter-departure time (Week based)

APPENDIX - D SIMULATION MODELING

Most real-world systems are too complex to allow realistic models to be evaluated analytically and obtain exact information. These models must be studied by means of simulation. Data are gathered in order to estimate the desired true characteristics of the model [95].

Simulation is a sound approach to gain insights of the dynamics of complex systems without costly physical experiments. Simulations are useful in many contexts, including simulation of technology for performance optimization, safety engineering, testing, training and education. They can estimate the eventual real effects of alternative conditions and courses of action. Overall, the high-level advantages of simulation include [95]:

- Most complex, real-world systems with stochastic elements cannot be accurately described by a mathematical model that can be evaluated analytically. Thus, a simulation is often the only type of investigation possible.
- Simulation allows one to estimate the performance of an existing system under some projected set of operating conditions.
- The entire production system can be easily monitored and controlled without doing any changes in the real environment, saving time, efforts and money. Parameters can be modified many times to determine the optimum configuration for the process flow. It reduces the risk of errors when manufacturers decide to modify any process, product or facilities in the factory.

- Alternative proposed system designs (or alternative operating policies for a single system) can be compared via simulation to see which best meets a specified requirement.
- In a simulation, much better control can be maintained over experimental conditions than would generally be possible when experimenting with the system itself.
- Simulation allows to study a system with a long time frame in compressed time, or alternatively to study the detailed workings of a system in expanded time.

Nevertheless, simulations are not without their drawbacks:

- Simulations need to be handled with care, particularly during the elaboration of the model and the choice of parameters. If an error occurs, it might render completely invalid the system, and the model will have to be rebuilt from the beginning, resulting in the whole project delay. And so the key in simulation is to address several issues including: (1) relevant selection of key characteristics and behaviours, (2) the use of simplifying approximations and assumptions within the simulation and (3) assessing the fidelity and validity of the simulations outcomes.
- Simulation results are hypothetical. Their transfer to real environment is not 100% sure and outcomes may differ slightly from predictions, particularly for complex systems such as semiconductor processing lines. Such an enormous manufacturing industry does not allow any process mistake, because small discrepancies could result in millions loss.

- Each run of a stochastic simulation model produces only estimates of a model's true characteristics for a particular set of input parameters. For this reason, simulation models are generally not as good at optimisation as they are at comparing a fixed number of specified alternative system designs [95].
- The large volume of numbers produced by a simulation study or the persuasive impact of a realistic animation often creates a tendency to place greater confidence in a study's results than is justified [95].

D.1 Simulation packages

A simulation is essentially a controllable statistical experiment technique that, with a model, is used to obtain approximate answers for questions about complex problem. It is useful when analytical and numerical techniques are unable to provide answers.

Atherton [108] proved in 1990 the efficiency of such simulation applied to cluster tools. But at the time computers calculation power was far too insufficient to even consider applying the same methodology to even a small section of a semiconductor line. Nowadays powerful simulation packages, such as Enterprise Dynamics (www.incontrolsim.com) and Extend (www.extendsim.com), allow this type of analyses.

Simulation packages deal in a very literal manner with the interactions of products and resources. The operations are modelled in terms of fundamental events and their interaction. A detailed-simulation model mimics each and every event in the operations sequence [108]. They are a powerful simulation tool, which can be used to develop dynamic models of real processes in the factory.

Most contemporary simulation packages use the process approach to simulation modelling [95]. A process is a time-ordered sequence of interrelated events separated by intervals of time, which describes the entire experience of an "entity" as it flows through a "system" [95]. The process corresponding to an entity arriving to and being served at a single server is shown in Figure D.1.

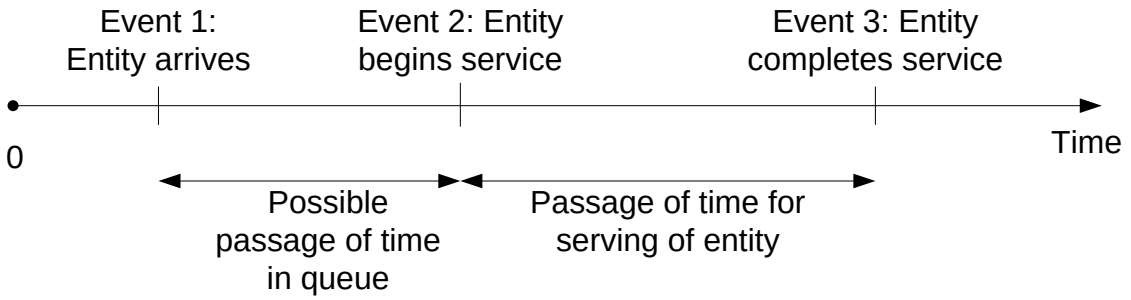


Figure D.1: Process describing the flow of an entity through a system [95]

A simulation using the process approach evolves over time by executing the events in order of their time of occurrence. An entity is created, travels through some part of the simulated system, and then is usually destroyed. Entities are distinguished from each other by their attributes, which are pieces of information stored with the entity. As an entity moves through the simulated system, it requests the use of resources. If a requested resource is not available, then the entity joins a queue.

Simulation packages have some advantages over general-purpose programming language [95]:

- Simulation packages automatically provide most of the features needed to build a simulation model: generating random numbers, generating random variates from a specified probability distribution, advancing simulated time, collecting

output statistics or reporting results. It results in a significant decrease in "programming" time.

- They provide user friendly interface and a visual structure to the model.
- Simulation models are generally easier to modify and maintain.

They provide better error detection because many potential types of errors are checked for automatically. Since fewer modelling constructs need to be included in a model, the chance of making an error will probably be smaller.This study is using the simulation software package available in the university, Extend software (www.extendsim.com), to conduct experiments. The simulation capacity of Extend is significant and easily put in practice to build models. Further information on Extend is introduced in the next section.

D.2 Extend™ V6 Simulation Software

Extend is a general-purpose simulation package marketed by Imagine That, Inc. Extend (www.extendsim.com) can develop dynamic models of a real production line system in any industry. It can be used to create models from building blocks, explore the processes involved, and see how they relate. Thus, Extend helps model-builder to design new systems, and it also allows us to improve existing ones. This simulation provides a method for checking one's understanding of the factory and helps model-builders achieving better results faster. With Extend, a block diagram of a process can be created where each block describes one part of the process. Extend's iterative technique lets model-builders create models of real manufacturing processes that are too complex to be easily represented. Models can also be created quickly because Extend comes with all the blocks needed for most simulations. These blocks act like macros, so models can be built without even having to type an equation. Many blocks are assembled into a single

model. For illustration a series of simple blocks will be used to introduce the definitions to model building.

D.2.1 Model Building

To build a simulation model, one browses through Extend's extensive block libraries to find the blocks corresponding to each operation of the production line. Each block has a different function. The blocks required are selected, and dragged onto the working space. The blocks are connected to indicate the flow of items through the system. Each block's parameters can be adapted to the user requirements using dialog boxes. The internal ModL language can be used to customize existing blocks and to create new blocks. There are an essentially unlimited number of random-number streams available in Extend. Furthermore, the user has access to 18 standard theoretical probability distributions and also to empirical distributions.

This section will show how to build an Extend model for the manufacturing system.

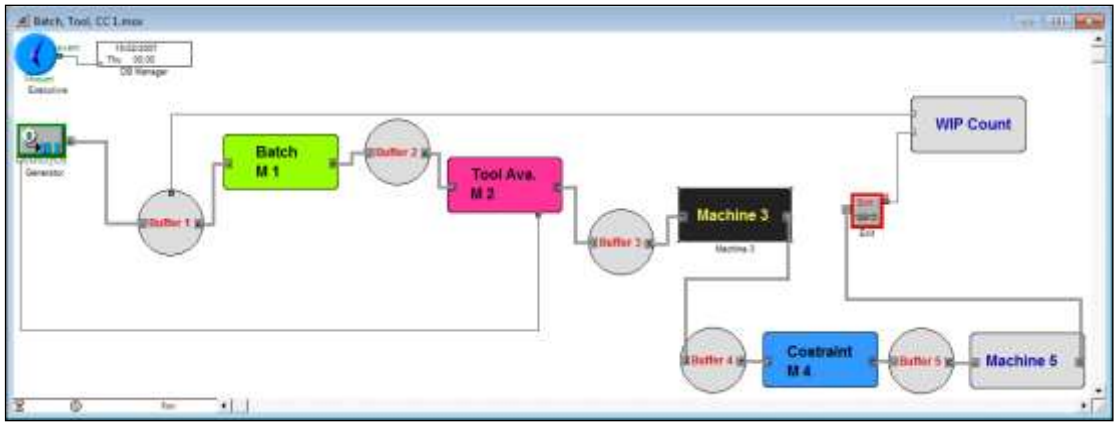


Figure D.2: Five machines serial line with batch, downtime and constraint simulation model in scenario 3

Figure D.2 shows the required blocks and connections for the model. The thick connections correspond to the flow of items, and thin connections are used for the

transmission of values (e.g. sending an observation drawn from a probability distribution to a block). The "Executive" block is the event list for an Extend model, while the "Generator" block is used to create items having constant inter-departure times (Figure D.3).

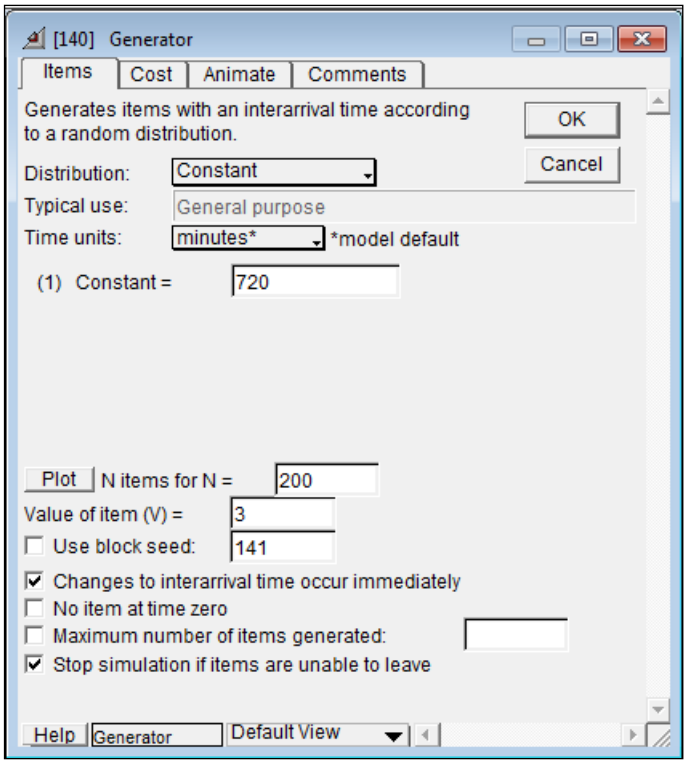


Figure D.3: Dialog box for the Generator block

The "Database Manager" (Figure D.4) block is the main user interface for the Database which is generating, displaying, editing and deleting, importing and exporting database from within Extend.

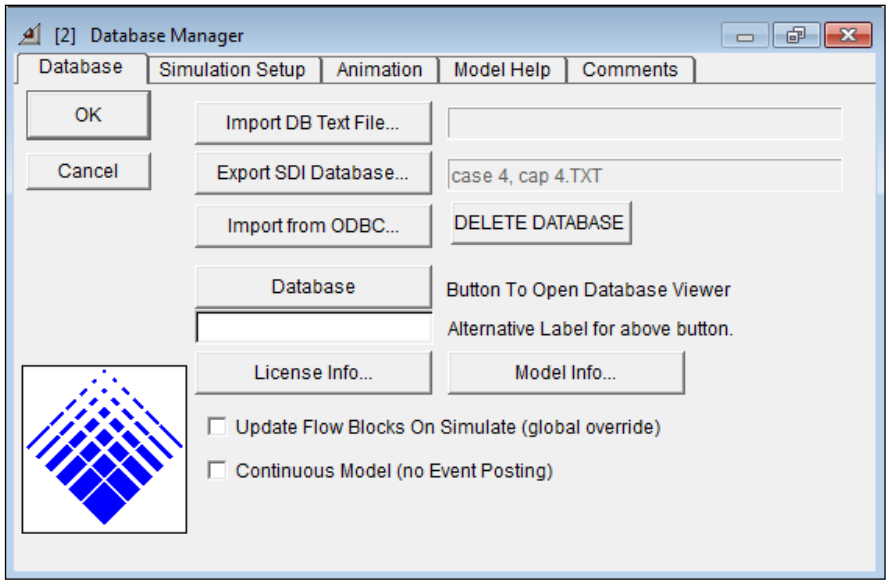


Figure D.4: Dialog box for the Database Manager block

The "Exit" block is the output of the simulation. The rest of the blocks were created by grouping multiple blocks together into second level units (Hierarchical block). By double clicking these units, the original blocks can be seen. For instance, Machine 3 is shown in Figure D.5. There are Timer block, Machine blocks and DB Write block. Timer block (Figure D.6) computes the arrival time, departure time and processing time in system of each item. Machine block (Figure D.7) processes items for a specified processing time. DB Write blocks transfer arrival time, processing time and departure time data to the database.

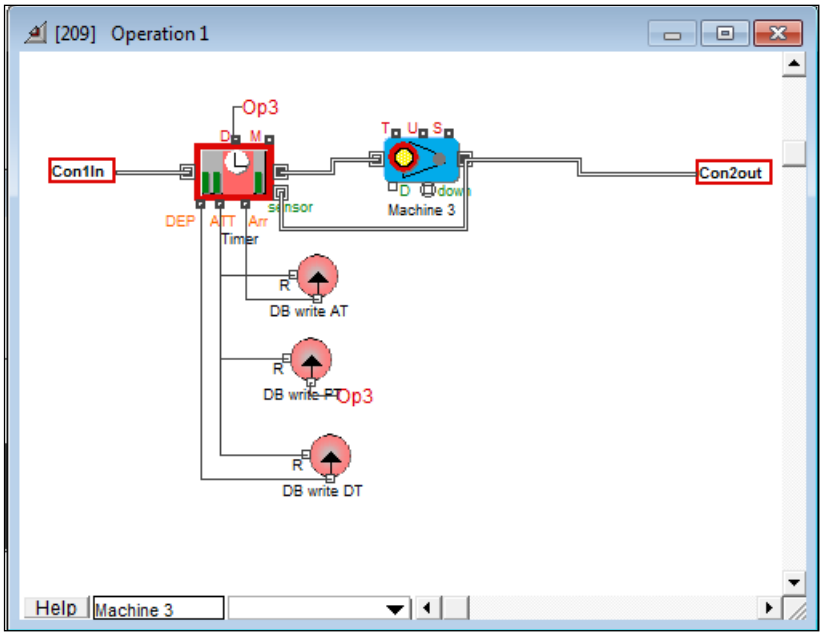


Figure D.5: Hierarchical structure for Machine 3

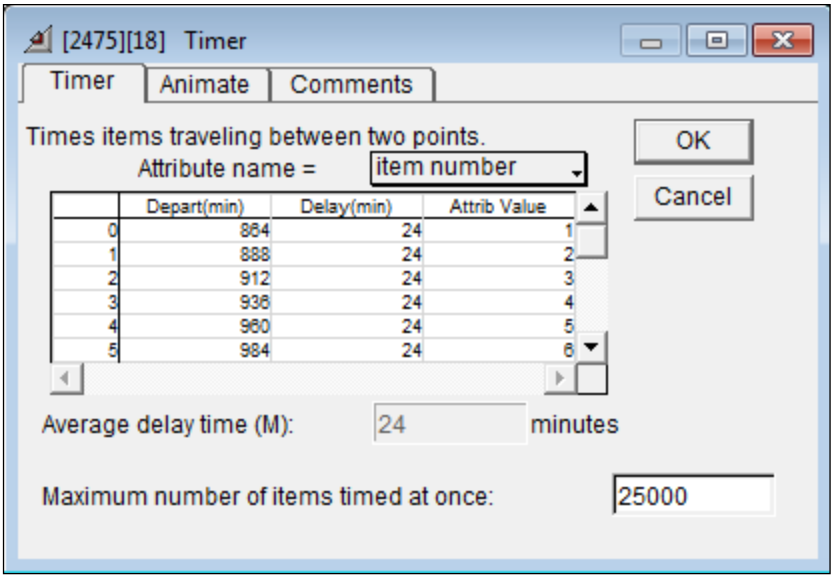


Figure D.6: Dialog box for the Time block

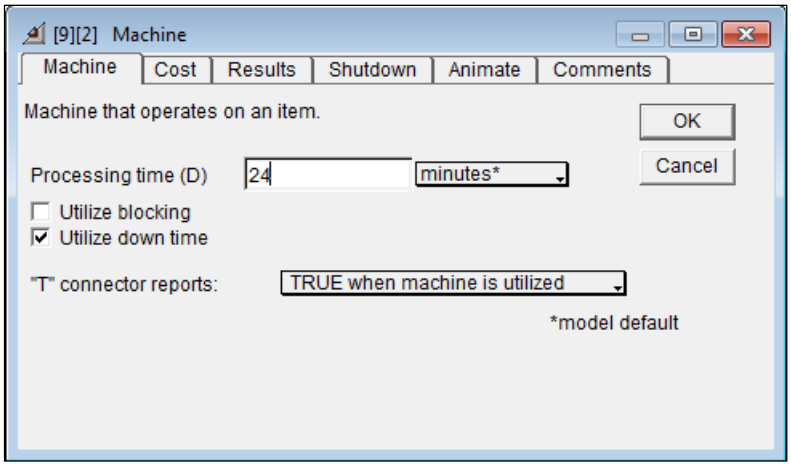


Figure D.7: Dialog box for machine block

D.2.2 Simulation Running

When the model is ready, a simulation can be run. Any data (such as process time, queue time, or item number) can be plotted or gathered in an Excel spreadsheet or a database using appropriate blocks. The simulation's running time is dependent on the model's complexity. The more complex is the model, the longer the running time. Therefore, care should be taken during the model creation to optimize the speed of simulation, particularly around the collection and management of the results. The consequent size of the files can dramatically affect the performance of the computer. The use of a database over Excel is greatly recommended. Data can always be transferred to Excel later for analysis.

For example, while the simulation is running, the data are recorded in the database as indicated in Figure D.8. After the simulation, all these data from Extend database can be transferred automatically through a text file into a pre-formatted Excel sheet (Figure D.9). Pre-formatting allow the analysis of the data, with for example automatic statistics calculation and graphics drawing.

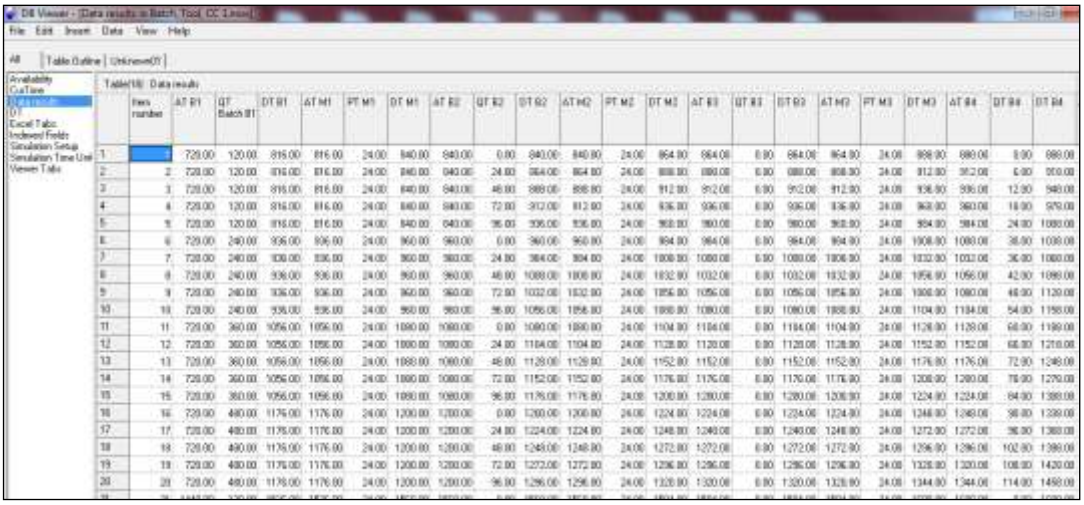


Figure D.8: Simulation data dialog box in Extend Database

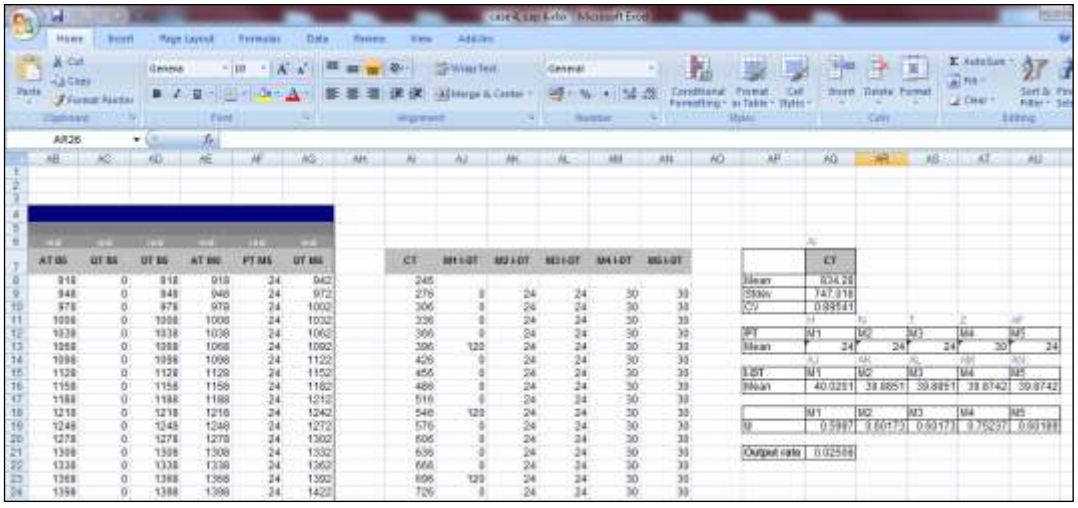


Figure D.9: Simulation data in Excel sheet

APPENDIX - E ONE BUFFER, ONE MACHINE SIMULATION

This scenario studies the relationship between Queue Time (QT), Utilization (U), and Inter-Departure Time (I-DT) for a basic setup: one machine, one buffer and fixed processing time. This simple model studies the basic principles of the items progression into the production line. The objective is to understand, the interactions of the buffer and the machine. For instance, how items go through the buffer to reach the machine. Why items are waiting in the buffer? Why machines cannot process all items from the buffer? A simple model gives basic answers that can help comprehending more complex models

E.1 Model

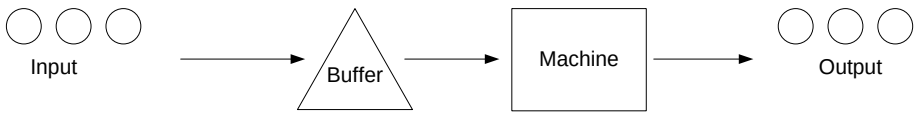


Figure E.1: One buffer and one machine model

- The input releases the items one by one
- Buffer follows the first-in-first-out (FIFO) rule.
- Machine capacity is fixed at 420 items/week

The machine processing time is constant 24 minutes (calculation reference in Equation 4.1).

E.2 Scenario 1: fixed release's interval time

In scenario 1, the input releases the items with a fixed period (fixed release's interval time). The effect of loading on utilization, QT and inter-departure time is expected to follow the graphic of Figure E.2.

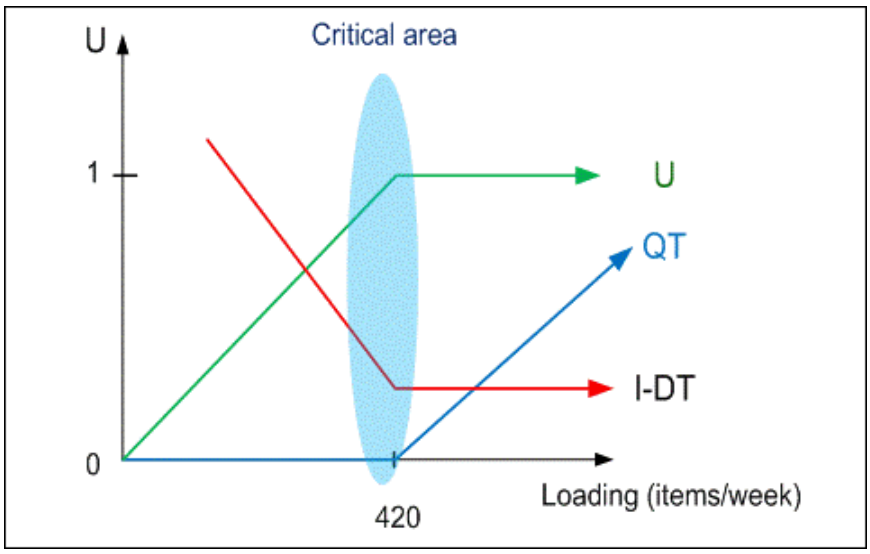


Figure E.2: Hypothesis on the relationship between Utilization, Queue, and Inter-departure time

Indeed the machine's utilization should increase with the loading until it reaches the machine's maximum capacity (420 items per week). Then, the machine will work at maximum capacity (100% of the time), utilization is 1. It remains 1 even when the loading is much higher than the capacity.

Moreover, the item release is done at fixed time interval, one item by one item. As long as the release's time interval remains longer than the processing time, the items are not making a queue. Therefore with a loading lower than 420 items per week, queue time is zero. On the contrary, with loading higher than 420 items per week, the release's interval is shorter than the processing time. Thus, items have to queue, the higher the loading, the longer the queue.

Finally, when the release's interval time is longer than the processing time (loading inferior to 420 items per week), items are processed as soon as they arrive at the machine (no queue). Therefore as the processing time is constant, the inter-departure time from the machine is identical to the release's interval time. When the release's interval time is shorter than the processing time (loading superior at 420 items per week), the machine works at full capacity (U=1). Therefore, the inter-departure time from the machine is identical to the processing time whatever the loading.

E.2.1 Experimental Conditions

The hypothesis on the relationship between Utilization, Queue Time and Inter-departure time needs to be tested. Therefore, simulations will be run for several loading levels, respectively, 100, 200, 300, 420, 500, 600 and 700 items per week. Experiment setup is shown in the table below.

Table E.1: Scenario 1 simulation setup

Total Simulation Run Time	Capacity (items/week)	Processing Time (min)	Loading (items/week)	Release period (min)
9 months	420	24	100	100.8
	420	24	200	50.4
	420	24	300	33.6
	420	24	420	24
	420	24	500	20.16
	420	24	600	16.8
	420	24	700	14.4

Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5).

E.2.2 Simulation Results

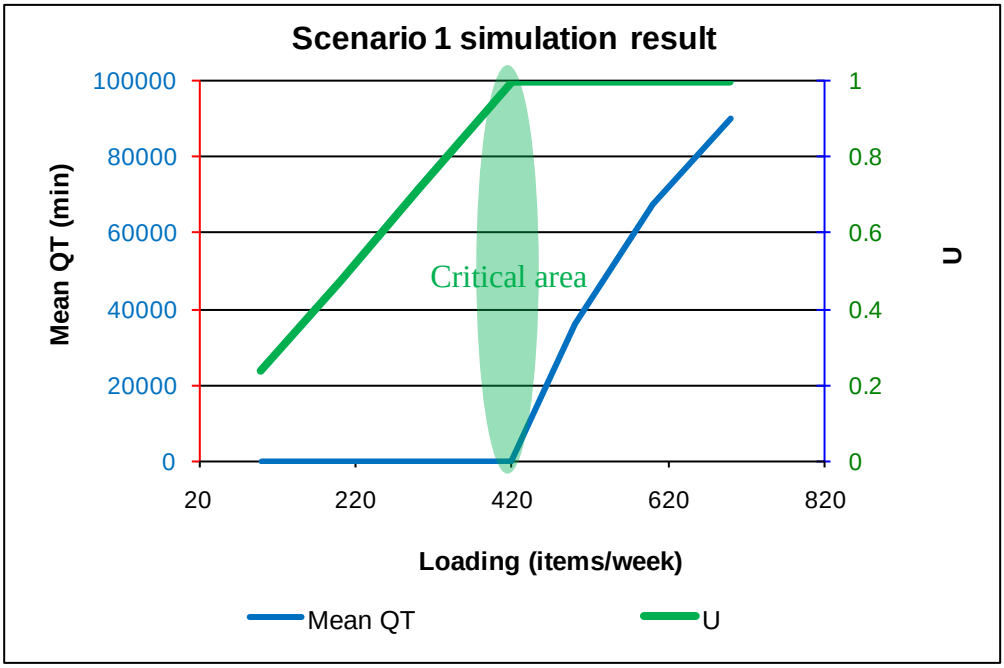


Figure E.3: Scenario 1 simulation results: Mean queue time and utilization

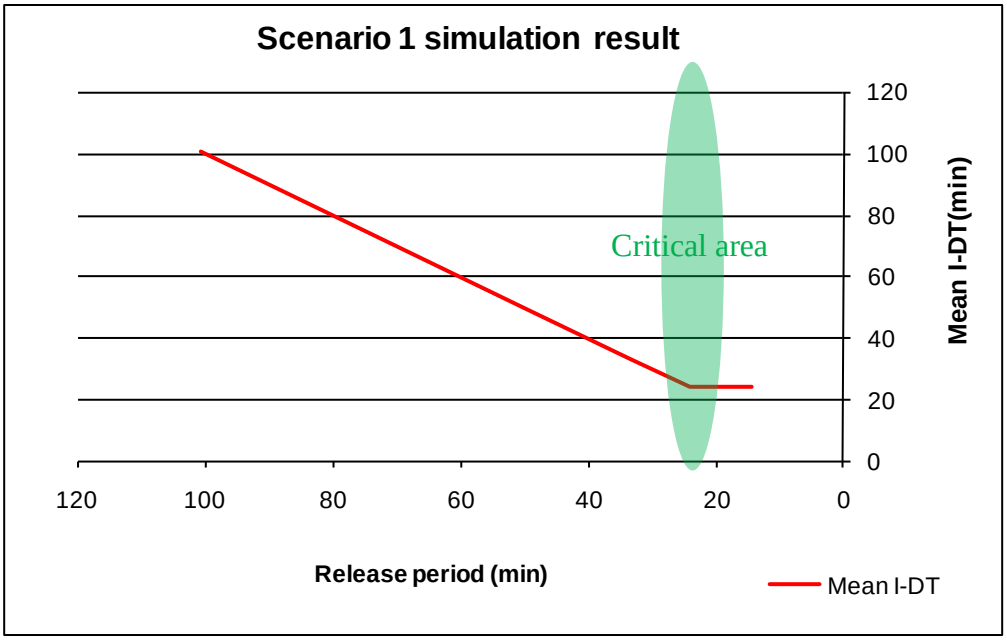


Figure E.4: Scenario 1 simulation results: Mean inter-departure time

These results confirm the hypotheses and validate the model (Figure E.2). On all those graphics (Figure E.2, Figure E.3 and Figure E.4) a critical area can be located when the loading is close of the machine capacity. Dramatic changes in QT, U and I-DT behaviors can be noticed.

E.2.3 Key Insights

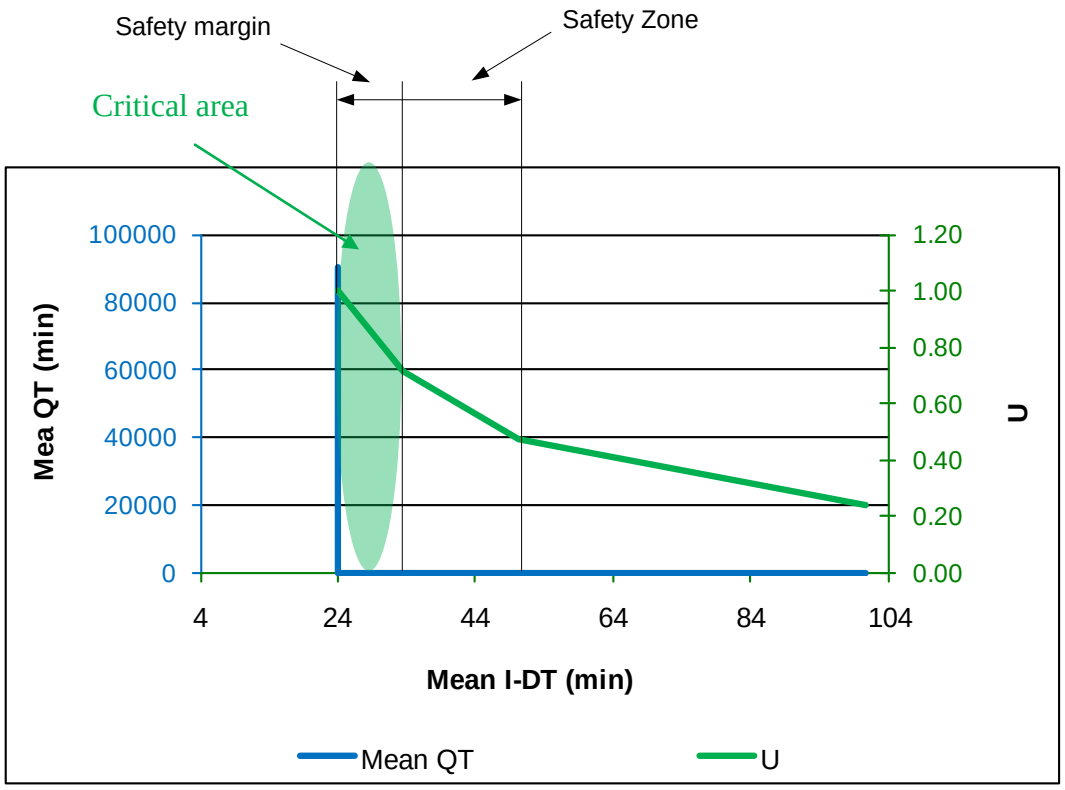


Figure E.5: Safety margin and safety Zone

These results show the interesting relationship between loading and machine capacity, and their effect on utilization and queue time. When loading is lower than machine capacity, then queue time will be reduced eventually down to zero. Indeed the machine has free time available to process queuing items. If the loading exceeds the maximum

capacity, the queue time (cycle time) will start to increase. In order, to keep a safety margin toward process variability, the loading should be kept in a safety zone away from the formation of long queues but close enough of the optimum inter-departure. Thus, it will optimize simultaneously cycle time and throughput.

Indeed, Figure E.5 results also show that inter-departure time can be used to monitor an appropriate loading of a single machine. The loading is optimum when the inter-departure time is close to the processing time. Nevertheless, in this case a safety margin is even more important, as the mean inter-departure time cannot discriminate between a small and a large queue.

As an operation composed of several machines can be assimilated to one machine with a shorter processing time (except for downtime behavior analyses), it would be interesting to extend the previous conclusion and control the production flow by monitoring the inter-departure time at each operation. Indeed in this case, if the inter-departure time from a station is too low compared to its current capacity (refer Machine Capacity p16), then the loading of this station needs to be increased. On the other hand, if the inter-departure time of a station is coming too close to its current capacity, then its loading needs to be reduced. The following simulations will test if this conclusion can be extended to more complex models and particularly to a constraint operation in multiple operations systems.

E.3 Scenario 2: Varying Release Period Following a Lognormal Distribution

Variability will be addressed in this simulation, and the relationship between QT, U and I-DT will be determined when the inter-arrival time (release period) is varying. In addition, the inter-departure-time variability will be interpreted in order to characterize and measure it.

E.3.1 Experimental conditions

To include randomness in the model, the release rate will follow a probability distribution. This distribution should have several characteristics:

- be continuous
- allows no negative values
- be a two-parameter distribution to allow independent mean and standard deviation variations

Several common probability distributions respect these characteristics: Weibull, Gamma, Lognormal, and Erlang distributions. However, the inter-departure time of an operation can be thought of as the multiplicative product of many independent random positive variables. Indeed, the inter-departure time is affected by all the random perturbations occurring in each of the preceding operations. Therefore, the lognormal distribution (Appendix A.1.6) was selected.

Data calculation will omit the simulation's warm-up time (Initial-data deletion method, Section 4.5). The following loadings, 100, 200, 300, 340, 380 and 420 items/week are

simulated by setting the adequate mean value for the distribution (Table E.2). The 340 and 380 loadings where added to obtain more result points in the critical area. Variability is studied by slowly increasing the coefficient of variation of inter-arrival time ($CV_{I\text{-}AT}$) to the machine, until its impact on queue time becomes evident. The coefficient of variation was thus successively set at 0, 0.05, 0.1, 0.2 and finally 0.5. The mean inter-departure time is determined from the loading. The inter-departure time standard deviation is calculated from the mean inter-departure time and the coefficient of variation.

Table E.2: Scenario 2 simulation setup

Total Simuation Run Time	Capacity (items/week)	Processing Time (min)	Loading (items/week)	Release Period				
				Lognormal Distribution				
				Mean (min)	Standard Deviation (SD)			
					CV=0.05	CV=0.1	CV=0.2	CV=0.5
9 months	420	24	100	100.8	5.04	10.08	20.16	50.4
	420	24	200	50.4	2.52	5.04	10.08	25.2
	420	24	300	33.6	1.68	3.36	6.72	16.8
	420	24	340	29.6	1.48	2.96	5.92	14.8
	420	24	380	26.5	1.32	2.65	5.3	13.25
	420	24	420	24	1.2	2.4	4.8	12

E.3.2 Simulation Results

Mean Queue Time

As can be seen from Table E.3, the variability of the release period affects the queue time only when the loading is close to the machine's maximum capacity. Mean queue time is almost zero for the 100, 200 and 300 load; there is almost no queue. This is due to a loading lower than the machine capacity, of 420 items per week.

Table E.3: Scenario 2 simulation result: Mean queue time

Mean QT (min)	100 load	200 load	300 load	340 load	380 load	420 load
CVI-AT = 0	0.00	0.00	0.00	0.00	0.00	0.00
CVI-AT = 0.05	0.00	0.00	0.00	0.00	0.01	97.32
CVI_AT = 0.1	0.00	0.00	0.00	0.02	0.32	197.76
CVI_AT = 0.2	0.00	0.00	0.11	0.56	2.67	406.04
CVI_AT = 0.5	0.01	0.46	3.75	8.34	24.89	1055.34

So the buffer will always maintain a very low queue time. The queue is starting to build from 420 loading. Especially, when the standard deviation is increasing, the queue time is increasing as well. For example, 300 loading means a mean release period of 33.6 minutes (see in Table E.2). When $CV_{I-AT} = 0.05$, data show that the release period ranges between 31.92 and 35.28 minutes. Even the minimum release period is higher than the machine processing time of 24 minutes. Therefore there is no queue. But if the $CV_{I-AT} = 0.5$, then it ranges between 16.8 and 50.4 minutes. Therefore, in this case the release period is shorter than the machine processing time and a small queue is created. That's why in 100, 200 and 300 loads a very low queuing time is obtained, but there is still a possibility to increase the queue time when the standard deviation is large, creating occasional release periods shorter than the processing time. Finally, if the loading is higher than 420 items per week, obviously the queue is building and the mean queue time is gradually increasing at a rate depending on the loading. The higher the loading the quicker the queue time is building.

Mean Inter-Departure Time (I-DT)

The mean inter-departure time (Table E.4) is only marginally affected by the variability in the release period. The mean inter-departure time is gradually decreasing when the loading is increasing until it reaches a value equal to the processing time.

Table E.4: Scenario 2 simulation results: Mean inter-departure time

Mean I-DT (min)	100 load	200 load	300 load	340 load	380 load	420 load
$CV_{I\text{-}AT} = 0$	100.80	50.40	33.60	29.59	26.49	24.00
$CV_{I\text{-}AT} = 0.05$	100.90	50.44	33.60	29.59	26.49	24.01
$CV_{I\_AT} = 0.1$	101.01	50.47	33.59	29.58	26.48	24.02
$CV_{I\_AT} = 0.2$	101.23	50.53	33.57	29.55	26.46	24.04
$CV_{I\_AT} = 0.5$	101.78	50.67	33.48	29.45	26.36	24.08

Indeed, the mean inter-departure-time is constant when the loading exceeds 420 items/week. Those loadings are higher than the maximum capacity of the machine, therefore the mean inter-departure time is fixed by the maximum output capacity of the machine, and the inter-departure time is equal to the processing time. Loading a machine above its maximum capacity only increases the mean queue time. Under 420 loading, the items are not queuing therefore the mean inter-departure time is almost identical to the mean inter-arrival time. So in this case, it is identical to the release period.

Inter-Departure Time Variability

Let's recall the formulas given in Section 3.2.2.

$$CV_{I-DT(i)}^2 = u_i^2 CV_{PT(i)}^2 + (1 - u_i^2) CV_{I-AT(i)}^2 \qquad \text{Equation E.1 [11]}$$

$$CV_{I-DT(i)}^2 = u_i^2 CV_{PT(i)}^2 + (1 - u_i^2) CV_{I-DT(i-1)}^2 \qquad \text{Equation E.2}$$

In this model, the processing time is fixed ($CV_{PT(i)}^2 = 0$). Therefore, the previous formulas become:

$$CV_{I-DT(i)}^2 = (1 - u_i^2) CV_{I-AT(i)}^2 \qquad \text{Equation E.3}$$

$$CV^2_{I-DT(i)} = (1 - u_i^2)CV^2_{I-DT(i-1)}$$ Equation E.4

For loading higher than 420 items/week, the utilization (u) is 1. Therefore (Equation E.3), the inter-departure time variability is zero. As mentioned earlier, if the machine works at maximum capacity, the inter-departure time is fixed (Figure E.6). Therefore the variability is nil. When the items are not queuing (low loading u≈0), the inter-departure time remains identical to the release period as was explained previously. Therefore the variability remains the same as the release variability (0, 0.05, 0.1, 0.2, and 0.5). It can also be deduced from Equation E.4: $CV_{I-DT(i)} \approx CV_{I-DT(i-i)}$

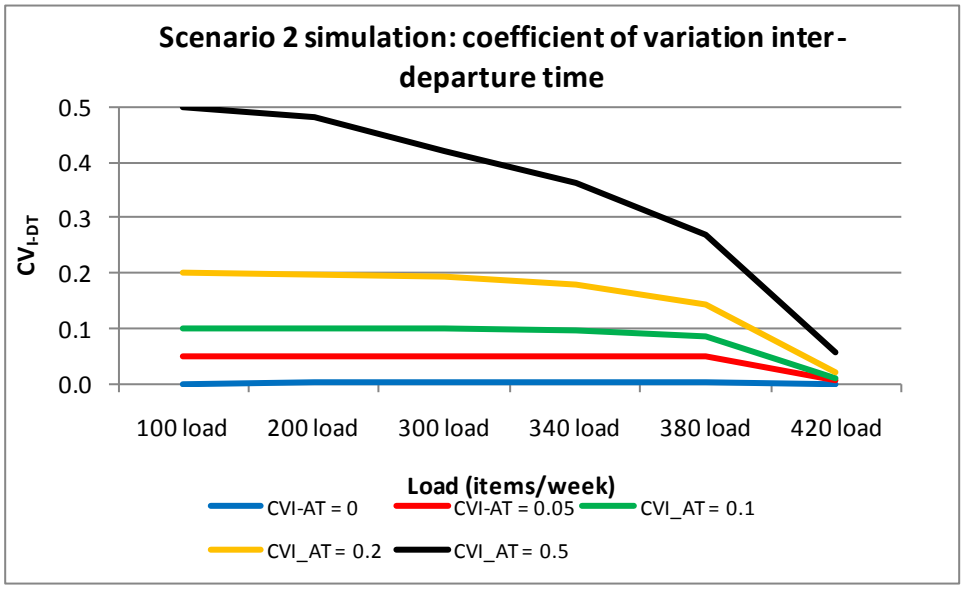


Figure E.6: Scenario 2 simulation results: Coefficient of variation inter-departure time (CV_{I-DT})

When some items start to queue, the variability decreases (Figure E.6).The curves follow Equation E.3. The machine acts as a dam and the buffer as a reservoir. Variability in the release means that the flow of items is fluctuating up and down. When the release's flow is high, items are stocked in the buffer and the machine's output flow remains constant (maximum output capacity). When the release flow is low, the queue

in the buffer decrease but the machine's output remain constant (maximum output capacity) until the queue is empty. Thus the machine inter-departure time variability is lower than the inter-arrival time variability. Therefore queuing reduces inter-departure time variability. There is a compromise to be found between long queue times and high variability.

This result can also be applied if a high capacity machine with high inter-departure time variability is followed by a constraint machine. If a queue appears in front of the constraint, then the inter-departure time of the constraint has a lower variability than the inter-departure time of the high capacity machine. The constraint stabilizes the flow of items. Here also a compromise has to be found between long queue times and high variability.

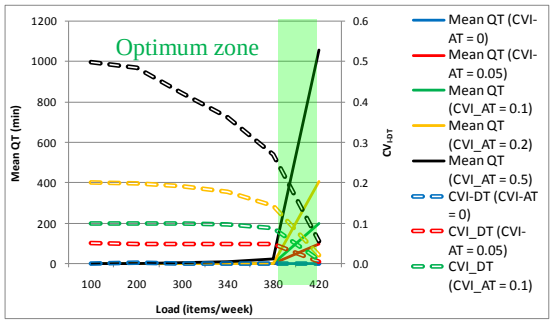


Figure E.7: Mean queue time, mean inter-departure time and coefficient of variation inter-departure time

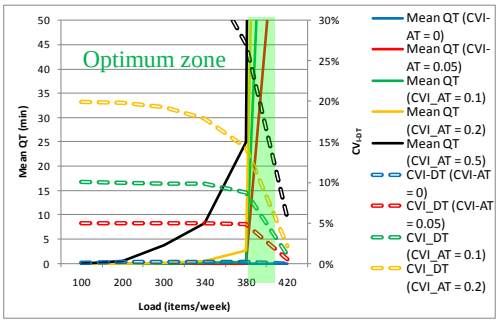


Figure E.8: Zoom on Mean queue time, mean inter-departure time and coefficient of variation inter-departure time

Figure E.7 plots the inter-departure time coefficient of variation and mean queue time versus the loading. From Figure E.7, an optimum value for loading should be identified. Ideally, this optimum loading value would give minima values for both QT and $CV_{I\text{-}DT}$. Unfortunately, it can be seen that when QT is at its minimum then $CV_{I\text{-}DT}$ is at its maximum. For example $CV_{I-AT} = 0.5$, QT is minimum for a loading of 100 items/week (QT= 0.01min) but for the same loading CV_{I_DT} is maximum ($CV_{I\text{-}DT} = 0.5$). Reversely,

when $CV_{I\text{-}DT}$ is at minimum, QT is at its maximum. For example $CV_{I-AT} = 0.5$, CV_{I_DT} is minimum for a loading of 420 items/week ($CV_{I\_DT} = 0.06$) but for the same loading QT is maximum (QT = 1055min). It is therefore impossible to have simultaneously $CV_{I\text{-}DT}$ and QT at their minimum. A compromised loading value needs to be found where both $CV_{I\text{-}DT}$ and QT have intermediary values (neither minimum nor maximum).

Figure E.8 zooms on such optimum values for loading. In the highlighted green zone, a compromise is found between mean QT and coefficient of variation. For example $CV_{I-AT} = 0.5$, for a loading of 380 items/week QT= 24.9min and $CV_{I\text{-}DT} = 0.27$. Any loading value around 380 items/week provides a good compromise (green zone).

Utilization

Overall, when the loading reaches the maximum machine capacity, utilization becomes 1 (Table E.5), the machine will never be idle and the queue is building. Otherwise a lower loading gives a lower utilization

Table E.5: Scenario 2 simulation results: Utilization

U	100 load	200 load	300 load	340 load	380 load	420 load
$CV_{I\text{-}AT} = 0$	0.24	0.48	0.71	0.81	0.91	1
$CV_{I\text{-}AT} = 0.05$	0.24	0.48	0.71	0.81	0.91	1
$CV_{I\_AT} = 0.1$	0.24	0.48	0.71	0.81	0.91	1
$CV_{I\_AT} = 0.2$	0.24	0.47	0.71	0.81	0.91	1
$CV_{I\_AT} = 0.5$	0.24	0.47	0.72	0.81	0.91	1

Mean Cycle Time

Mean cycle time is sum of mean processing time and mean queue time. Processing time is fixed at 24 minutes. The variation in CT is result of the queue time. So, CT behavior is the same than mean queue time (Table E.3). The higher is queue time, then the higher is cycle time.

Table E.6: Scenario 2 simulation results: Mean cycle time

Mean QT (min)	100 load	200 load	300 load	340 load	380 load	420 load
$CV_{I\text{-}AT}$ = 0	0.00	0.00	0.00	0.00	0.00	0.00
$CV_{I\text{-}AT}$ = 0.05	0.00	0.00	0.00	0.00	0.01	97.32
CV_{I_AT} = 0.1	0.00	0.00	0.00	0.02	0.32	197.76
CV_{I_AT} = 0.2	0.00	0.00	0.11	0.56	2.67	406.04
CV_{I_AT} = 0.5	0.01	0.46	3.75	8.34	24.89	1055.34

E.3.3 Key Insights

On the one hand, if the machine is always kept busy ─ utilization 1 ─ then the inter-departure time is constant and the variability is nil. But the queue time is building. On the other hand, if the loading is really lower than the machine capacity, there is not any queue but the machine will transfer the inter-arrival time variability to the inter-departure time. A compromise needs to be found between variability and queue by adjusting correctly the loading of the machine. This can be done by either setting a correct utilization target or a correct inter-departure time target. Another solution might be to dampen the variability of a machine by limiting the capacity of the following machine.

APPENDIX - F CONFLOW RESULTS

F.1 CONFLOW Results – Scenario 1: Two Machines (Operations) Model

F.1.1 CONFLOW Results: Scenario 1 Model 2

Table F.1: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Baseline	522.0055	207.5873	180	864
Push	4541.7554	1412.4229	2711.9209	8237
CONFLOW option 1	867.1520	846.4133	180	8100
CONFLOW option 2	907.5116	859.4674	180	8099
CONFLOW option 3	907.9330	860.6236	180	8099

Table F.2: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Baseline	504.0502	197.7269	178.290	829.8000
Push	2890.9652	1685.9442	178.290	8203.0188
CONFLOW option 1	818.4585	852.0994	178.290	8098.2900
CONFLOW option 2	830.0732	871.8305	178.290	8077.2810
CONFLOW option 3	805.5013	888.9839	178.290	8077.2810

Table F.3: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Baseline	487.6700	188.7315	176.730	798.6000
Push	2246.6512	1571.7410	176.730	8171.8188
CONFLOW option 1	801.9781	863.1376	176.730	8096.7300
CONFLOW option 2	811.1795	883.1399	176.730	8089.7010
CONFLOW option 3	842.3825	915.7914	176.730	8089.7010

Table F.4: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Baseline	459.0046	172.9894	174	744
Push	1613.8155	1448.8951	174	8117
CONFLOW option 1	774.5496	876.4319	174	8094
CONFLOW option 2	783.1874	896.2683	174	8094
CONFLOW option 3	833.4336	983.0151	174	8087

F.1.2 CONFLOW results: Scenario 1 Model 3

Table F.5**: Performance of the policies with respect to cycle time for Capacity 1**

	Mean	Stdev	Min	Max
Baseline	887.9824	203.6592	600	1176.0000
Push	1123.1691	224.7099	672	1584.0000
CONFLOW option 1	900.9217	206.6832	600	1463.9163
CONFLOW option 2	907.4673	205.3214	600	1463.9163
CONFLOW option 3	907.4582	205.3243	600	1463.9163

Table F.6: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Baseline	846.8541	194.6884	586.290	1134.8700
Push	862.0405	201.0495	586.290	1529.1600
CONFLOW option 1	849.2285	199.3477	586.290	1450.2063
CONFLOW option 2	854.7823	199.0452	586.290	1450.2063
CONFLOW option 3	856.8424	199.3968	586.290	1450.2063

Table F.7: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Baseline	809.4456	187.8998	573.820	1097.4600
Push	826.8340	195.7153	573.820	1509.8200
CONFLOW option 1	816.1244	193.8328	573.820	1437.7363
CONFLOW option 2	821.3771	193.9224	573.820	1437.7363
CONFLOW option 3	825.7020	195.1808	573.820	1437.7363

Table F.8: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Baseline	754.7876	184.2723	552	1032
Push	777.2719	193.4400	552	1488
CONFLOW option 1	768.0387	191.0813	552	1488
CONFLOW option 2	773.0681	191.6368	552	1488
CONFLOW option 3	781.7663	195.9181	552	1488

F.2 *CONFLOW Results – Scenario 2: 5-Stage Serial* Line with Batch and Constraint Machine

Table F.9**: Performance of the policies with respect to cycle time for Capacity 1**

	Mean	Stdev	Min	Max
Push	3771.6040	808.8040	2042.7582	5634.7115
CONFLOW option 1	599.1006	565.5064	132	5490.7115
CONFLOW option 2	619.9779	551.1174	132	4923.1494
CONFLOW option 3	619.9779	551.1174	132	4923.1494

Table F.10: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Push	1965.4437	1159.5709	130.290	5600.5115
CONFLOW option 1	579.8021	561.5781	130.290	5497.6415
CONFLOW option 2	582.9224	547.2027	130.290	4888.9494
CONFLOW option 3	589.3698	555.1860	130.290	4888.9494

Table F.11: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	1476.1942	1127.4975	128.730	5569.3115
CONFLOW option 1	561.8965	556.8293	128.730	5471.1215
CONFLOW option 2	564.2018	541.9418	128.730	4857.7494
CONFLOW option 3	579.8622	561.7811	128.730	4857.7494

Table F.12: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1086.8562	1017.4927	126	5514.7115
CONFLOW option 1	531.7431	552.0407	126	5484.7115
CONFLOW option 2	532.5748	533.0544	126	4803.1494
CONFLOW option 3	563.1777	568.2690	126	5000.4911

F.3 CONFLOW Results – Scenario 3: 5-Stage Serial Line with Batch, Tool Availability and Constraint Machine

F.3.1 CONFLOW Results: Scenario 3 Model 5 BTC

Table F.13**: Performance of the policies with respect to cycle time for Capacity 1**

	Mean	Stdev	Min	Max
Push	4013.9345	1144.6899	2606.7582	6018.7115
CONFLOW option 1	865.6927	661.1602	252	5874.7115
CONFLOW option 2	911.3226	670.8861	252	5874.7115
CONFLOW option 3	911.3226	670.8861	252	5874.7115

Table F.14: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Push	2243.4034	1145.3783	250.290	6001.6115
CONFLOW option 1	839.9461	662.1305	250.290	5898.8315
CONFLOW option 2	843.4163	680.0887	250.290	5898.8315
CONFLOW option 3	846.4156	675.0765	250.290	5898.8315

Table F.15: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	1725.5769	1151.6675	248.730	5986.0115
CONFLOW option 1	821.6120	659.4503	248.730	5887.8815
CONFLOW option 2	822.5528	677.0339	248.730	5887.8815
CONFLOW option 3	858.4838	696.9786	248.730	5887.8815

Table F.16: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1294.0810	1062.3581	246	5958.7115
CONFLOW option 1	790.6707	656.9071	246	5898.7115
CONFLOW option 2	789.9143	673.1742	246	5898.7115
CONFLOW option 3	829.7558	711.9993	246	5898.7115

F.3.2 CONFLOW Results: Scenario 3 Model 5 TBC

Table F.17: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	4012.5023	1171.0253	2282.7582	5874.7115
CONFLOW option 1	856.9251	592.4940	348	5778.7115
CONFLOW option 2	887.9654	612.5632	348	5778.7115
CONFLOW option 3	887.9654	612.5632	348	5778.7115

Table F.18: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Push	2202.5450	1163.6893	346.290	5840.5115
CONFLOW option 1	837.7164	590.6955	346.290	5782.2215
CONFLOW option 2	846.2421	612.9685	346.290	5782.2215
CONFLOW option 3	845.7435	606.4142	346.290	5778.8015

Table F.19: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	1709.0377	1134.2878	344.730	5809.3115
CONFLOW option 1	820.2942	589.4557	344.730	5785.3115
CONFLOW option 2	827.5740	610.9417	344.730	5785.3115
CONFLOW option 3	847.7438	628.2308	344.730	5778.7715

Table F.20: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1314.8769	1025.7059	342	5754.7115
CONFLOW option 1	789.8092	586.6764	342	5760.7115
CONFLOW option 2	796.1457	606.6277	342	5760.7115
CONFLOW option 3	813.6464	595.1478	342	5252.4911

F.3.3 CONFLOW Results: Scenario 3 Model 5 TCB

Table F.21: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	4036.5023	1030.8340	2330.7582	5874.7115
CONFLOW option 1	847.8531	573.3430	348	5790.7115
CONFLOW option 2	880.6096	598.1374	348	5790.7115
CONFLOW option 3	880.6096	598.1374	348	5790.7115

Table F.22: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Push	2223.1250	1163.1774	387.450	5840.5115
CONFLOW option 1	828.6278	571.4128	346.290	5782.2215
CONFLOW option 2	840.1495	598.2436	346.290	5782.2215
CONFLOW option 3	840.6476	592.4296	346.290	5778.8015

Table F.23: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	1726.4977	1133.8534	379.650	5809.3115
CONFLOW option 1	811.5449	569.9760	344.730	5776.5815
CONFLOW option 2	821.4439	596.2116	344.730	5776.5815
CONFLOW option 3	836.1439	614.2630	344.730	5794.0415

Table F.24: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1326.8769	1025.3906	366	5754.7115
CONFLOW option 1	781.7236	566.9745	342	5778.7115
CONFLOW option 2	790.0622	591.9711	342	5778.7115
CONFLOW option 3	826.0424	635.8282	342	5760.7115

F.3.4 CONFLOW results: Scenario 3 model 5 CTB

Table F.25: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	1153.6346	902.1240	396	6162.7115
CONFLOW option 1	887.6632	653.7282	348	6090.7115
CONFLOW option 2	898.2479	677.5078	348	6090.7115
CONFLOW option 3	898.2479	677.5078	348	6090.7115

Table F.26: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Push	1131.0840	900.2238	387.450	6138.7115
CONFLOW option 1	870.5168	653.7503	346.290	6090.7115
CONFLOW option 2	875.5303	666.7142	346.290	6090.7115
CONFLOW option 3	879.3144	670.9188	346.290	6090.7115

Table F.27: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	1114.1543	900.8613	379.650	6138.7115
CONFLOW option 1	853.0216	648.7898	344.730	6090.7115
CONFLOW option 2	857.4393	663.1597	344.730	6090.7115
CONFLOW option 3	869.6386	675.8538	344.730	6090.7115

Table F.28: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1080.3267	899.6938	366	6114.7115
CONFLOW option 1	822.0958	641.7840	342	6090.7115
CONFLOW option 2	828.1236	658.7034	342	6090.7115
CONFLOW option 3	856.4300	697.8259	342	6090.7115

F.3.5 CONFLOW results: Scenario 3 model 5 CBT

Table F.29: Performance of the policies with respect to cycle time for Capacity 1

	Mean	Stdev	Min	Max
Push	1147.1525	896.1558	396	6090.7115
CONFLOW option 1	897.3182	668.5683	348	6090.7115
CONFLOW option 2	897.8091	673.0030	348	6090.7115
CONFLOW option 3	897.8091	673.0030	348	6090.7115

Table F.30: Performance of the policies with respect to cycle time for Capacity 2

	Mean	Stdev	Min	Max
Push	1127.8051	894.5101	387.450	6090.7115
CONFLOW option 1	879.9648	669.0228	346.290	6090.7115
CONFLOW option 2	880.4961	671.4319	346.290	6090.7115
CONFLOW option 3	886.7719	685.5267	346.290	6066.7115

Table F.31: Performance of the policies with respect to cycle time for Capacity 3

	Mean	Stdev	Min	Max
Push	1110.8520	898.9462	379.650	6090.7115
CONFLOW option 1	858.9767	652.6845	344.730	5994.7115
CONFLOW option 2	862.2952	666.8091	344.730	5994.7115
CONFLOW option 3	877.8141	688.8950	344.730	6066.7115

Table F.32: Performance of the policies with respect to cycle time for Capacity 4

	Mean	Stdev	Min	Max
Push	1075.9066	895.9796	366	6090.7115
CONFLOW option 1	829.0475	651.7751	342	5994.7115
CONFLOW option 2	833.4591	662.5939	342	5994.7115
CONFLOW option 3	862.5195	707.5426	342	6080.4911