A Scheduling Policy Experiment for Lean Implementation

by

Sawan P. Deshpande

B.E. (Electrical), University of Pune (1997)

Submitted to the Technology and Policy Program
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1999

Signature of Author . . . .
Technology and Policy Program
August 13, 1999

Certified by . . . .
Stanley B. Gershwin
Senior Research Scientist
Thesis Supervisor

Accepted by . . . .
Richard de Neufville
Chairman, Technology and Policy Program

A Scheduling Policy Experiment for Lean Implementation

by

Sawan P. Deshpande

Submitted to the Technology and Policy Program
on August 13, 1999, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

This thesis is about a new scheduling policy to achieve lean improvements for a factory. The policy is concerned with the allocation of finite resources and its effects on the due date and inventory performance of the factory. The policy makes use of time and finite buffers to make the scheduling decisions for the factory.

The goal of the thesis is to gain more insights into the behavior of the policy. We use an experiment in a factory to show the applicability to a real-factory scenario. We also use discrete event simulations to study the performance of the policy for manufacturing systems.

Thesis Supervisor: Stanley B. Gershwin
Title: Senior Research Scientist

Contents

List of Figures

Chapter 1

Introduction

This thesis is about a new scheduling policy for a factory. The thesis is particularly concerned with the due date and inventory effects of the policy. The policy makes use of time and finite buffers to make the scheduling decisions in a factory. To give an intuitive feel for what the scheduling policy is trying to achieve, we start with some analogies drawn from very different fields.

In [Goldratt, 1994], the cause and effect analysis of all the undesirable effects in the factory point to one basic cause: "Managers are trying to run their factories by striving to achieve local optima". However, most managers will say that the local optima are as important as the global optima for running a factory. While the global optima keep the company running, the local optima keep the machines and the people running. Neglecting either of them is suicidal. In our analogy, the decision processes at each point observe the local information and the global information, and factor both into their decisions.

Consider a manager's typical day in the office. There are dozens of people meeting him during the day with several inputs. There are several tasks to be done, and the time available is finite. The tasks come in at random times, and every now and then, the manager would like to sit down and figure out the schedule of things to do. The first thing the manager makes sure of is that the most important tasks are at the top of the list. The next thing to be checked is when the tasks need to be completed and the estimate of the time required to complete the tasks. Based on this, the manager can decide whether it is *too early* to start work on the task, or whether it is *ready* to be done. If a task is early, it can wait, otherwise it should be looked at as soon as possible. Then the manager goes about doing the *ready* tasks in the order in which they appear on the list. Using mental calculations to do all the scheduling may not be possible at times. The manager may use day-planner or a palm-pilot in that case.

Another example is the control of incoming flights at a busy airport. During a busy time of the day, there are many flights scheduled to arrive at very close times intervals. However, this schedule is subject to significant amount of randomness. It may be due to poor weather or events at other airports. There may be more than one plane asking for permission to land *at the same time*. The control tower needs to decide the schedule for landing these planes. The first calculation the tower does is the difference between the scheduled arrival times for the flights and the time remaining. If a flight is early it can wait. For the remaining flights, it checks for the ones with top-priority, depending on different criteria: the amount of fuel left, the number of passengers on-board, the length of the trip etc. Thus, the landing sequence for the flights is decided.

In both above cases, the calculations done at real time are logical and also very simple. We are concerned with similar scheduling issues: allocation of finite resources and its effect on number of dissatisfied customers. The control methodology used, like the above examples, makes intuitive sense and the goal of this document is to gain more insights into its performance. We use an experiment in a factory to show the applicability to a real-factory scenario. We also use discrete event simulations to understand the behavior of the control system.

1.1 Literature review

There are excellent books available on the topic of factory scheduling and management. Important *systems* problems in factory design and scheduling are explored in [Gershwin, 1994]. Using decomposition techniques, *quick and reliable* analytical techniques are developed for studying the problems. [Hopp and Spearman, 1996] provides a very useful reference guide to develop practical ideas for managing factories. [Baudin, 1990] is a small but comprehensive reference for computer-aided management of manufacturing lines. In [Pinedo, 1995] a comprehensive analysis of the theory and application of different scheduling techniques in manufacturing is provided.

Manufacturing Resource Planning (MRP II) is the leading scheduling technique used in factories of today. [Hopp and Spearman, 1996] and [Baudin, 1990] provide a detailed treatment of the logic and working of the MRP II. There is a lot of literature available which studies the problems associated with MRP II. [Yeung et al., 1998] provides a complete review of the research. [Miltenburg, 1997] discusses how the Theory of Constraints (TOC) [Goldratt and Fox, 1984] and Toyota Production System (TPS) techniques [Monden, 1998] can be embedded into MRP II, without dismantling the MRP II system.

There is a substantial literature on token-based control polices for manufacturing systems. [Buzacott and Shanthikumar, 1993] present a common framework to describe these policies using production authorization cards (PAC). All the token-based policies (KANBAN, basestock, CONWIP) can be considered as special cases of the PAC. [Liberopoulos and Dallery, 1998] provide a unifying framework for studying all these policies in a multi-stage manufacturing system. There is an immense amount of KANBAN literature, and one survey paper is [Berkley, 1992]. Descriptions of the Toyota Production System, of which KANBAN control is a part, can be found in [Monden, 1998]. [Berkley, 1991] shows that a common model of KANBAN systems is equivalent to a traditional tandem production line with finite buffers. [Gershwin, 1994] provides rigorous mathematical treatment of such systems. [Spearman et al., 1990] introduce the constant work-in-progress (CONWIP) policy. CONWIP limits the total inventory of all parts in the system and allows the inventory location to vary as appropriate. [Hopp and Roof, 1998] discuss the use of statistical throughput control (STC) techniques for setting WIP levels in CONWIP lines. [Kimball, 1988] discusses the basestock policy in a simple and accessible from. [Bonvik, 1996] compares the token-based policies and their hybrids (combination of two pure policies) using simulations.

[Gershwin, 1999] traces the history of research for the hedging point theory. Using the earlier, more general work of [Rishel, 1975], [Kimemia and Gershwin, 1983] establish the general form of the hedging point theory for a single stage unreliable machine for constant demand. [Bielecki and Kumar, 1988] and [Akella and Kumar, 1986] obtain the complete analytical solutions for single machine two part-type cases. [Gershwin, 1999] follows the subsequent research for extensions to multiple stages and multiple part-types to conjecture the policy for multiple stages and multiple part-type systems, which is the *control point policy (CPP).* The policy has all three versions: surplus-based, token-based and time-based. The work presented in this document is a *time-based* extension of the work of [Gershwin, 1999]. [Gershwin, 1999] also conjectures equivalence between the token-based and time-based versions of the policy.

Time-based scheduling policies have received a lot of attention recently. A comprehensive review of the research in this area is provided in [Blackstone et al., 1982]. The different control rules which can be used are analyzed in [Gutzmann and Wysk, 1986]. [Lu and Kumar, 1991] describe and get good performance for the *least slack policy.* [Salegna, 1996] uses simulation techniques to compare the performance for different time-based policies for a job-shop environment.

1.2 Outline

Chapter 2 describes the issue of factory scheduling from different perspectives. It discusses the applicability of scheduling policy-types to different manufacturing environments. It also presents a critical appraisal of MRP II: the most popular scheduling system of today.

Chapter 3 introduces and defines the control point policy (CPP). Later in the chapter, the policy is compared with other commonly used time-based and token-based scheduling policies and intuitive reasons are given for better performance of the policy.

Chapter 4 serves as a guide for future CPP implementations at other factories. The sequence of implementation steps and the possible modifications at each step are discussed. The system metrics are also discussed in the same chapter.

Chapter 5 deals with experiment at Boeing. The Boeing facility at Portland, Oregon hosts the 777 flap support business unit, where the CPP was implemented for the first time. We go through each step of implementation using the framework discussed in Chapter 4. The results and the conclusions of the experiment are discussed at the end of the chapter.

Chapter 6 presents the results of the simulation studies. We report results for a three stage single part-type system under CPP-control. We also discuss the results of the comparison studies of different policies for the case of a multiple stage two part-type assembly process.

Chapter 7 concludes the thesis.

Chapter 2

Factory Scheduling

2.1 Introduction

All manufacturing companies want on-time delivery, minimal work-in-progress, short lead times and the maximum utilization of resources. Unfortunately, these goals conflict. It is much easier to finish parts on time if the utilization of resources is low. Lead times can be essentially made zero, if an enormous amount of finished good inventory is maintained. The goal of the scheduling policy is to strike a profitable balance between these conflicting objectives.

Scheduling concerns the allocation of limited resources over time. To be fully effective, a scheduling system needs to be incorporated within a control methodology which enables shop-floor performance to be analyzed and, when necessary, corrective actions to be formulated. However, the manufacturing system performance is not only dependent upon short-term planning decisions, but is also constrained by the long term capability for which it is designed. Hence, any control methodology should be based on a consistent set of performance measures and well defined procedures which help to integrate decision making at all levels.

[Gershwin, 1994] describes such a control hierarchy, by dividing the control or management of a manufacturing system into a hierarchy consisting of different levels. The levels of the hierarchy correspond to classes of events that have distinct frequencies of occurrence. Each level is characterized by the length of the planning horizon and the kind of data required for the decision making process at that level. Higher levels of the hierarchy have long time horizons and use highly aggregated data, while lower levels have shorter time horizons and use more detailed information. The nature of uncertainties at each level of control also varies.

This chapter discusses the issue of factory scheduling from different perspectives. Section 2.2 discusses the role of the scheduling policy in the manufacturing hierarchy. Section 2.3 discusses the common symptoms of poor scheduling in a factory. In Section 2.4 we classify the commonly used scheduling policies using a framework which is used throughout the document. An important, yet relatively unexplored, area of research has been the applicability of the scheduling policies to different manufacturing environments. Section 2.5 provides such a classification system for manufacturing environments, which can be used along with the framework discussed in Section 2.4. Since the focus of this document is on the time-based scheduling policies, Section 2.6 reviews the research in this area. In Section 2.7, we present a critical appraisal of MRP II: the most popular, but widely criticized scheduling system of today. Section 2.8 concludes the chapter.

2.2 Role of scheduling

[Sethi and Zhang, 1994] discuss the real-time hierarchical control of manufacturing systems in great detail. In the control hierarchy, the scheduling function has to interface with several other important functions above and below it. Figure 1.1 [Pinedo, 1995] depicts a diagram for the information flow and the position for the scheduling function in the manufacturing hierarchy. On the top, it is affected by the production planning and material planning processes, which handle the medium to long range planning for the entire organization. This process must consider the forecasts for demand, sales, inventory levels, and resource requirements to optimize, at a higher level, the product mix and the long-term resource allocation. The scheduling function also receives input from the lower level: shop-floor. Unexpected events on the shop-floor, such as machine breakdowns, or longer processing times, maintenance shutdowns have to be taken into account to optimize, at a lower level, performance of the production system. The goals of the scheduling function is then to be able to communicate efficiently with both levels and:

1. To be predictable and reliable.

2. To work efficiently. Get the most work done, while using minimum resources.

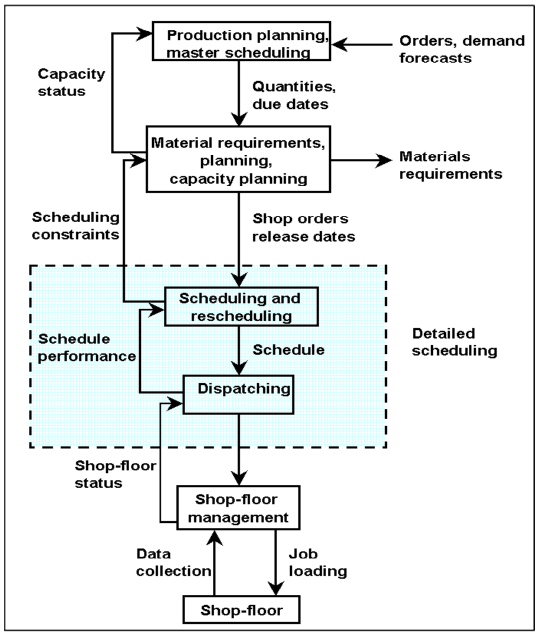


Figure 2-1: Manufacturing control

3. To be reasonable. Set targets and provide solutions which are realistic.

4. To order on time. Bring in raw material or purchased parts only when needed, thus, minimizing raw material inventory and storage space.

5. To release orders on time. Make effective capacity decisions, smooth the flow of work through the shop-floor, and reduce WIP levels.

6. To ship the manufactured goods just in time. Keep the customers satisfied and also control the finished goods inventory and thus minimize possible damage or loss to finished goods.

7. To provide a path out of major production disruptions.

8. To provide metrics and directions for continuous improvements.

2.3 Signs of poor scheduling

Before we delve into the details of scheduling, some very common signs of poor scheduling in a factory are listed below:

1. More reactive than predictive.
2. Schedule followed on the shop-floor is different to that planned.
3. Long lead times and excessive WIP.
4. Inability to give a firm due dates. Premium shipments to customers and from vendors.
5. Expediting is a common practice.
6. Unplanned overtime and unplanned sub-contracting.
7. Continuously missed production deadlines particularly when less important parts went out on time.
8. Too many calls from sales representatives (or customers) asking for status information.
9. Panic schedule reorganization as the materials were not ordered in time.
10. "Miracle weeks" at the end of the month (larger number of shipments than other weeks).
11. Inability to give a clear picture of the forward work-load for each work-center.
12. None or only a mental schedule in the head of the factory manager.

They exist in many organizations and are indicators of incorrect or inadequate control on the system. The worst end of the scale, and yet, a very common sight in many organizations is the last symptom: None or only a mental schedule in the head of the factory manager. These organizations will exhibit all or most of the pitfalls detailed above. They have a particular weakness in that they are heavily dependent on the factory manager being present. The schedules will be poor as it is generally impossible to control more than a few parts at a time with this system. part priority is determined by who screams the loudest. Often, factory managers then see their part as filling in for the role of the scheduling function.

2.4 Classification of scheduling policies

A very useful way of classifying scheduling polices is to treat them as *token-based*, *time-based* or *surplus-based*. The goal of this section is to only classify them; all the policies are discussed in detail in Section 3.6.

1. **Token-based policies**

Token-based policies involve the movement of tokens to indicate and trigger events. When an event is complete (e.g. a demand for a product arrives or an operation is finished), a token to indicate the above events is either created or taken from a specific location and then either destroyed or moved into the another location. The tokens used may either be physical tokens like a card or a container or a visual signal like a chart or an electronic signal using a bar-code.

The flow of tokens in the system has two types: *global flow* and *local flow.* In *global flow*, tokens move directly between individual manufacturing stages (machines or buffers) and the demand or order process, without going through intermediate manufacturing stages. In *local flow,* tokens connect individual manufacturing stages to adjacent manufacturing stages.

The most popular example of this type system is the KANBAN system, made legendary by Toyota [Monden, 1998]. The finite buffer systems studied in [Gershwin, 1994] are a simple version of the KANBAN system. CONWIP [Spearman et al., 1990] is another example in which operations are not triggered when there are too many parts in a group of buffers. Another example, the basestock policy [Bonvik, 1996], triggers operations at machines depending on how far production is ahead of demand. The production authorization card (PAC) technique [Buzacott and Shanthikumar, 1993] studies the most general token system, involving tokens in the manufacturing system to trigger events. [Liberopoulos and Dallery, 1998] provide a unified framework for studying all these policies.

The important thing to note in all the above token-based policies is that a token needs to indicate only binary information i.e. whether a particular event has occurred or not. It does not need to contain information on the time, the nature and other characteristics of the event. This information is not necessary for the scheduling decisions. For example, consider that the n^{th} demand d_n, on arrival into the system, triggers a demand token. The system would then know that demand for a part has arrived. However, it is not necessary to know the arrival time, the due date, or the serial number n of the demand.

The binary information on the occurrence of the event is *necessary and sufficient* for the working of the token-based policies.

2. **Time-based policies**

Time-based policies operate on the basis of time and use the status of the parts in relation to their arrival times, due dates or their schedule to make the scheduling decisions.

The time-based policies have two versions: *forward scheduling* and *backward scheduling.* Forward scheduling begins with the first date a part can start and plans when the parts can get done ahead of that date. Backward scheduling starts with the date the part needs to be completed, i.e. its due date, and works back from there. Since most work in manufacturing is planned based on the due date, backward scheduling is used more frequently.

Example of time-based polices include the first in first out (FIFO), earliest due date (EDD), least slack policy (LSP), shortest processing time (SPT), and critical ratio calculation (CR). The version of the control point policy (CPP) discussed in this document is also an example of this type.

It is *necessary but not sufficient* for the policies to know that a particular event has occurred. These policies need more knowledge about the event. They also need to know the time of the event, and its relation with other events in the system. By using this information and calculations based on this information, the scheduling decisions are made.

3. **Surplus-based policies**

The surplus-based policies make the scheduling decisions based on far production is ahead or behind the demand process. Information on the production surplus or the production backlog is used to control the production process.

The hedging point theory [Kimemia and Gershwin, 1983], the two boundary and the basestock (it can be argued) policies are examples of this type of policies. They are the least popular of the three types, and find most of their application in theoretical methods used to develop the scheduling algorithms.

2.5 Applicability to manufacturing systems

While there has been extensive research in the area of production scheduling, there is no unifying framework to evaluate the applicability of the scheduling techniques to different manufacturing systems. Traditional industrial scheduling techniques have been used satisfactorily in a particular group of industries or manufacturing systems, but have major limitations when they are used outside these systems. Typical examples include:

- Lot sizing methods such as Economic Order Quantity (EOQ) and the Wagner-Whitin extension [Wagner and Whitin, 1958] work well for high volume assemblers of varied products such as industrial pumps or TV sets, but they have limited use when there are variable demand and yield issues.
- KANBAN and its various extensions can be used for deterministic and discrete production (and assembly) lines with great efficiency. But they have severe limitations in a job-shop type environment.
- Network, review, and analysis technologies such as Critical Path Method or Program Evaluation Review Technique are used only for planning one of a kind custom projects [Merrle, 1997].

These examples indicate that there is a need for taxonomy of manufacturing systems that can be used along with a catalogue for scheduling techniques to determine appropriate applications and also directions for further research.

The problem with this approach is that it is very difficult to find a single classification system, which can completely and accurately apply to all the scheduling policies and manufacturing systems. [Dietrich, 1991] proposes a detailed taxonomy to classify manufacturing environments. Manufacturing environments are characterized in terms of three categories: *the production process*, *the system management* and *the system behavior*. In this section, we follow the methods to conjecture a classification system. It can be used along with the classification system for the scheduling policies discussed in Section 2.4.

2.5.1 Manufacturing environments

For the sake of simplicity, consider a factory producing a single product. Consider d_n, the n^{th} demand arriving into the system. The demand d_n is a vector of information (t_n, q_n, δ_n). t_n denotes the arrival time for the demand d_n, q_n is the required quantity and δ_n is the time at

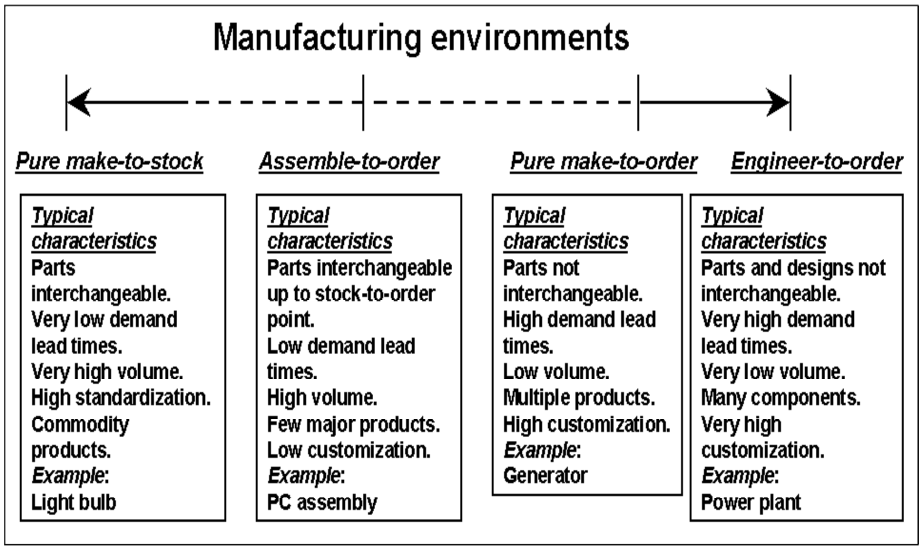


Figure 2-2: Classification of manufacturing environments

which the quantity q_n is required (due date). The difference ($l_n = \delta_n - t_n$) is the difference between the due date and the arrival date for the demand and is called the *demand lead time* for the n^{th} demand [Hariharan and Zipkin, 1995]. In this discussion, we assume the quantity q_n is unit quantity for all demand d_n*'s.*

The criteria we use are the two factors the company uses to determine the quantity of each product to manufacture: as a customer order or a desired level of production stock. There are four important types we identify:

1. Pure make-to-stock

2. Assemble-to-order

3. Pure make-to-order

4. Engineer-to-order

Figure 2-2 shows the classification system with the typical characteristics of each classification. The two extremes are the *pure make-to-stock* and the *engineer-to-order*. In the next paragraphs, we discuss the characteristics which define these classifications.

1. **Pure make-to-stock**

In a *pure make-to-stock* system, the production goal is to replenish the stock in the finished goods buffer. Parts in the finished goods buffer and within a production stage in the system are interchangeable. The customer lead time l_n is essentially zero; i.e. the demand d_n has to be served the instant it arrives. When the demand d_n arrives, if there is a part in the finished goods buffer, it is served, else the customer walks back dejected. The part served to the customer could have been the last part available in the finished goods stock, or it could have been one among hundreds present there. It does not make any difference to the company or the customer.

Typical examples of this type of manufacturing environment are the production of commodity goods like light bulbs or beverage bottles. For such a system, the best place to maintain inventory in the system, if any, is the finished goods buffer. Also parts (and thus tokens) within any production stage in the system can be use as buffer stock for other parts at the production stage.

The only information needed for the system is whether a demand d_n has arrived or not; no information is needed on the arrival time of demand. Also, the demand lead time l_n is zero ($\delta_n = t_n$). Therefore, the d_n information vector can be reduced to a single dimensional vector (q_n), which is (1), if there is a demand. Such a system can be easily represented using binary tokens.

2. **Pure make-to-order**

In a *pure make-to-order* system, work on the n^{th} part is not initiated until a customer places an order for it i.e. a demand d_n arrives. Once the demand d_n arrives, it is attached to the part n. They cannot be separated after that. Therefore, parts within a production stage inside the system are not interchangeable; each part n is earmarked for a particular customer n. Also, the customer n is ready to wait for some time to receive the part n, after he has placed his order ($l_n > 0$).

Typical examples of this type of system are an electric generator or a tailored suit: each is made to a customer specification. In this case, the best place to hold the inventory, if any, is in the raw material buffer. Once parts are issued into the system, they cannot be used as a buffer-stock for other parts. Parts (and thus tokens) are not interchangeable within a production stage.

The demand d_n for the part n is a three dimensional vector $(t_n, 1, \delta_n)$. The demand lead time l_n quoted has to take into the total cycle time from the raw material buffer to the finished goods buffer. Such a demand vector can not be represented by a binary token system. The information on the arrival time t_n and the due date δ_n is needed by the company to make important scheduling decisions.

3. **Assemble-to-order**

However, most companies, lie in a spectrum in between these two pure classifications. They have a *pure make-to-stock* system until a certain step in their manufacturing process, but beyond that they operate on a *pure make-to-order* basis. We call this point the *stock-to-order point*, up to which the company stocks the standard components in the product. Parts up to this point are interchangeable, beyond this point the cannot be interchanged.

An *assemble-to-order* system is a common example of this type. The company makes all the components which go into the assembly according to a *pure make-to-stock* system, but the final product is assembled only when the customer orders. Typical example of *assemble-to-stock* system is a PC assembler like Dell Computers. All the standard components are stocked, and the final product n is assembled when the order d_n arrives. Auto companies, for example, are trying to shift from a *pure make-to-stock* system to an *assemble-to-order* system.

The *stock-to-order* point in the company can lie at any point in the manufacturing process and depends on the type and the nature of the industry. The later the stock-to-order point, the less is the demand lead time l_n. The earlier the stock-to-order point, the more is the customization for each customer. The best place to store inventory, if any, for such a system is at the *stock-to-order* point. Parts and tokens within a production stage are interchangeable only up to the stock-to-order point.

The demand d_n for the part n is a three dimensional vector $(t_n, 1, \delta_n)$. The demand lead time l_n quoted will then be the time required for the assembly of the product. Therefore, the demand lead time l_n is shorter than in the case of a make-to-order system. Binary tokens can be used to communicate all the necessary information upstream of the *stock-to-order* point, but they can not be used downstream of the *stock-to-order* point.

4. **Engineer-to-order**

An extreme version of the *make-to-order* system is *engineer-to-order* system, where each part n is *designed and made* to the customer's specifications. There is no part interchangeability at any point in the system.

The manufacturing systems of this type usually produce expensive customized items. A power plant or a construction project are the typical examples of this type. In this type of a system, there is no point in holding inventory at any point in the system.

The demand d_n for the part n consists of has all three dimensions (t_n, q_n, δ_n) in addition to other information specific to the customer order. However, the demand lead time quoted l_n is larger than in a *pure make-to-order* case, because it has to take into account the total cycle time from the design process to the finished goods buffer. A simple system like a binary token cannot be used to represent the demand process or to schedule work.

2.5.2 Scheduling applicability

Based on the discussions of the above two sections, we conjecture an applicability spectrum for the different types of scheduling policies. Figure 2-3 shows Figure 2-2 with the applicability spectrum added to it. The following observations can be made:

1. The token-based policies can be used only for *make-to-stock* systems. The scheduling of these types of manufacturing systems requires only binary information about events in the system. Also parts and their tokens within a production stage are interchangeable. This type of control can be provided very efficiently by the token-based policies..

2. The token-based policies are not suitable in a *pure make-to-order* environment and any variation on its right hand side in Figure 2-3. The information needed for the scheduling of these systems cannot be provided by the binary token system. Once inside the system, parts and their tokens are not interchangeable. The time-based policies are the only ones which can be used in these cases.

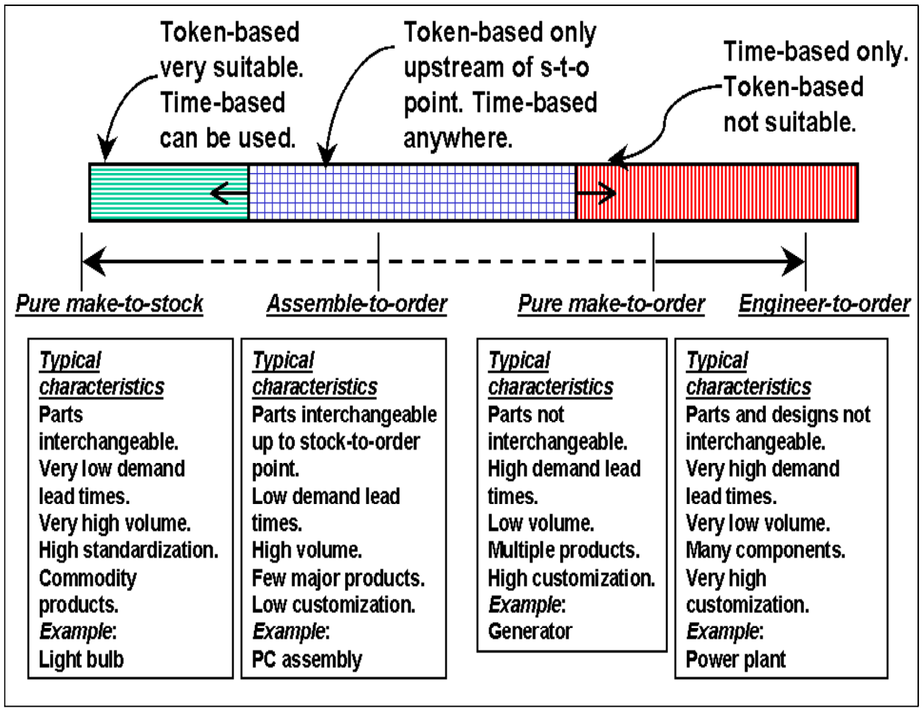

Token-based very suitable. Time-based can be used.
Token-based only upstream of s-t-o point. Time-based anywhere.
Time-based only. Token-based not suitable.
Pure make-to-stock
Assemble-to-order
Pure make-to-order
Engineer-to-order
Typical characteristics Parts interchangeable. Very low demand lead times. Very high volume. High standardization. Commodity products. Example: Light bulb
Typical characteristics Parts interchangeable up to stock-to-order point. Low demand lead times. High volume. Few major products. Low customization. Example: PC assembly
Typical characteristics Parts not interchangeable. High demand lead times. Low volume. Multiple products. High customization. Example: Generator
Typical characteristics Parts and designs not interchangeable. Very high demand lead times. Very low volume. Many components. Very high customization. Example: Power plant

Figure 2-3: Applicability of scheduling policies to manufacturing environments

3. Time-based policies can be used anywhere in the spectrum. The information needed for the scheduling of all types of manufacturing environments is maintained and provided by the time-based policies. The information is more than sufficient for *pure make-to-stock* systems, while it is sufficient for *pure make-to-order* systems.

4. In the wide spectrum between the *pure make-to-stock* and *pure make-to-order*, the location of *stock-to-order point* decides the applicability of the policies. The time-based policies can be used throughout such a system. On the other hand, the token-based policies can be used only upstream of the *stock-to-order point.* Many manufacturing companies lie in this spectrum and the stock-to-order point lies at different points in the manufacturing system. Since it is convenient to have only a single scheduling system in the entire organization, they tend to use time-based policies[1].

[1]This also may be the reason for the widespread popularity of MRP II.

2.6 Time-based scheduling research

The focus of this document is on the time-based version of the control point policy (Chapter 3). Therefore, in this section, we briefly review the research in time-based scheduling. Since the early 1950s, a considerable amount of research has been conducted in the area. This research can be classified into four basic areas:

1. Process of ordering
2. Process of releasing material
3. Assignment of due dates to parts
4. Dispatching rules inside the system

A comprehensive reviews of the scheduling literature have been compiled for all the areas. [Song and Zipkin, 1996] and [Melynk and Ragatz, 1988] provide a discussion of the research in order review strategies and the use of supply conditions. The decisions regarding the releasing orders to the manufacturing shop-floor are reviewed in [Wisner, 1995]. [Cheng and Gupta, 1989] review of the due date assignment research, while [Blackstone et al., 1982] provide a review of the dispatching research.

Though most research has been divided into these categories, the areas are highly interrelated. [Wein, 1988] varies the bottleneck conditions in a semiconductor facility to test combinations of several release rules and 14 dispatch rules. He found, in all cases, the release rule based on the bottleneck machine loading resulted in the best performance, irrespective of the dispatch rules inside the factory. [Ragatz and Mabert, 1988] argue that uncontrolled release of orders to the shop-floor places additional pressure on dispatch rules inside the factory to control the timely progress of orders through the shop. Conversely, [Baker, 1984] points out that restricting the number of orders on the shop-floor reduces the dispatching options available and creates a potential for lower overall production effectiveness. [Bertrand, 1983] tests different release strategies with two due date determination methods and two dispatch rules. The tests show that release rules did not significantly affect lateness variance, but rather act as amplifiers of the effects of dispatching and due date determination.

2.7 MRP II appraisal

In spite of being widely criticized in many articles [Turbide, 1995], Manufacturing Resource Planning (MRP II) is still the dominant, and almost a standard, software for today's manufacturing management. Many leading companies, including cutting-edge exponents like Dell Computers [Fine, 1998], rely on MRP II. The study of MRP II systems has received a lot of attention, and the literature on this subject is vast. However, there are a lot of misconceptions and ambiguities about the exact structure of the system. Our aim, in this section, is to provide a candid analysis of MRP II.

MRP II is a time-based backward infinite planning process, which assumes that due dates (intermediate and final) can be met. The original MRP was and is Material Requirements Planning, a calculation technique for planning purchase orders and manufacturing orders according to what is needed to complete a high-level master production schedule. MRP II, the next version of the program, was created to address the short-comings of the MRP and further integrate business functions into a common framework. MRP II has a very general control structure which breaks down the control problem into a hierarchy based on time scale and product aggregation. Enterprise Resource Planning (ERP) represents the next generation of MRP II systems [Turbide, 1995]. It includes MRP II functionality, plus new application areas (maintenance, quality, field service, marketing support), and a number of technology requirements.

Since we have taken up the issue of hierarchical production planning, we analyze MRP II within the same framework. Section 2.7.1 provides a brief description of the control hierarchy of the MRP II. The emphasis is on describing the lowest level of the MRP II control hierarchy, because it is the area of primary interest from the scheduling point of view. Section 2.7.2 discusses the short term control function. Section 2.7.3 discusses the inherent short-comings in the underlying model and assumptions of MRP. In section 2.7.4, we look at the measures which have been taken by the MRP II vendors to rectify the faults associated with MRP.

2.7.1 MRP II hierarchy

Figure 2-4 shows the general planning hierarchy of MRP II. The dark-shaded blocks represent the additions and improvements which have been made in MRP II over the original MRP in

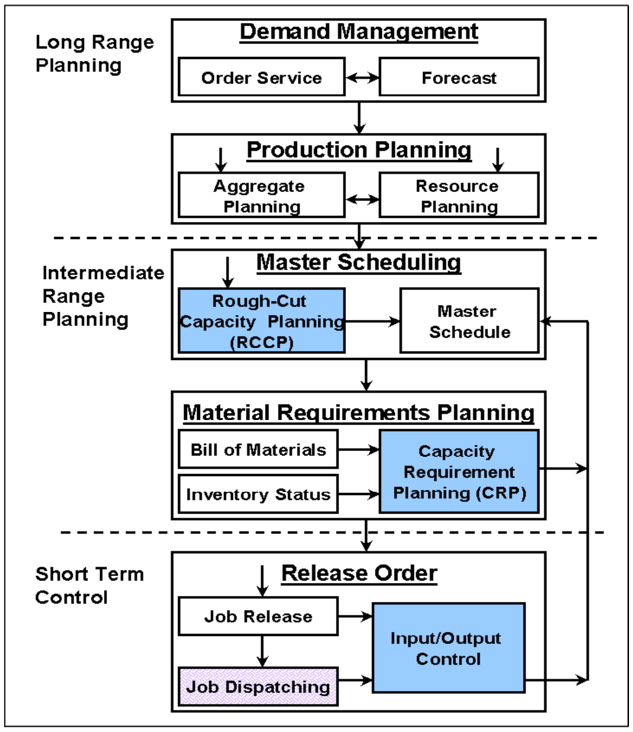


Figure 2-4: Control hierarchy of MRP II

the lower two levels of the hierarchy. The light shaded block represents the dispatching logic of the MRP II. Our discussion is restricted to these shaded blocks in Figure 2-4.

2.7.2 Short term control

The most important output of the intermediate range is the planned order release (POR). A POR does not correspond with any individual customer demand and most often consists of parts belonging to different orders, which have been pooled together using the different lot-sizing rules. This will determine the sequence of releasing orders into the shop-floor.

Once the POR is released into the internal manufacturing facility, the short term control takes over to ensure that each order within the POR is completed in the right quantities at the right time. If the order is an out-sourced part, then it is a simple task of monitoring the arrival and tracking the outstanding orders. If the order is within the internal manufacturing facility, then the queue in front of each work-center needs to be arranged in an order that will maintain the due date integrity while keeping the machine utilization and the lead times low. This is done by the *job dispatching function.*

Job dispatching rules are simple rules for deciding the loading schedule of a work-center. Since this is done at real time, the calculations are normally very simple and transparent to the user. There are various dispatching rules which are used to decide the sequence of loading. [Blackstone et al., 1982] provides a detailed analysis of the dispatching rules. The most commonly used ones are the following:

1. **First in first out (FIFO)**

   The part which has waited the longest in the queue is assigned top priority.

2. **Shortest processing time (SPT)**

   The part in the queue which has the shortest processing time on the work-center will be done first.

3. **Earliest due date (EDD)**

   The part in the queue which is closest to its due date is loaded first.

4. **Least slack policy (LSP)**

   The slack for a part is defined as its due date minus the remaining processing time in the system. The part with the least slack value is given the first priority.

5. **Critical ratio (CR)**

   parts are sorted according to the index computed by dividing the time available (due date minus the current time) by the time remaining. If the index is greater than one, it can wait longer. If it is less than one, then the part is late. If it is negative, it was already due at an earlier time. The part with the smallest value of the critical ratio is loaded first.

These are the most commonly used dispatching rules. There are many more rules which can be used.

2.7.3 Problems with MRP

Most of the problems associated with MRP are due to the underlying assumptions of MRP. They are also the reasons for the wide-spread dissatisfaction with the program. In this section, we list out the 4 major flaws with the program.

1. **Infinite capacity**

   MRP assumes that resource capacities are infinite and are independent of the work-load on the resource. The effect of failures, batching, set-ups and resource-contention on the capacity of the resource are not taken into consideration, while deciding the load schedule on the machine. In other words, it assumes infinite resource capacity while actual physical capacity is finite and dependent on the above factors.

2. **Fixed lead times**

   Another flawed assumption that MRP makes is that the lead times are fixed and not variable. It holds the lead times for the parts at each stage of operation, and uses these lead times, to determine the release and dispatching times for the parts. However, any factory manager knows that lead times are function of other parts on the schedule, the order in which they are scheduled and other random events in the factory. MRP treats the factory resources as if they are deterministic and infinite while calculating the lead times.

3. **System nervousness**

   Nervousness in MRP system occurs when a small change in the master production schedule (MPS) results in a large change in the planned order releases. For example, when the planned lead times of MRP are changed due to shop floor conditions, the schedule created by the MRP becomes unstable. [Vollmann et al., 1992] and [Yeung et al., 1998] provide detailed treatment of the problem.

4. **Infinite buffers**

   The MRP part dispatching rules do not take into account the size and the location of the WIP downstream of the work-center, while deciding the load schedule. This leads to excessive inventory accumulating into the factory, and the MRP system can not detect this inventory until the MPS is run the next time. Also, because of absence of finite buffers, unavailability of resources are not propagated in the system.

2.7.4 MRP II improvements

The full-shaded blocks in Figure 2-4 include the three functions that have been added to the MRP II hierarchy [Hopp and Spearman, 1996] to rectify the problems associated with original MRP. In this section, we analyze and critique the functions.

1. **Rough-cut capacity planning (RCCP)**

The RCCP is used to provide a quick capacity check of few critical resources to ensure the feasibility of the MPS. It sums the number of total hours to process all the items at work-center in a given time period, and checks whether the aggregate time available at the work-center is greater than the time required.

However, the RCCP does not regard how the work is scheduled within the work-center and its effect on the capacity. Also, it does not perform any time off setting. Therefore, the fundamental problems with MRP (infinite capacity and fixed lead times) are still not solved.

2. **Capacity requirement planning (CRP)**

CRP provides a more detailed capacity estimate check on the generated production plans than the RCCP. Necessary inputs include all planned release orders, existing WIP positions, routing data as well as capacity and lead times for all the process centers. CRP predicts the part completion times for each process, and thus computes a predicted loading at the work-centers over time. The loading schedules are then compared to the available capacities.

However, no corrections is made for the overloading situation. Even the CRP assumes that time required for processing at a work-center is fixed.. Also it tells the planner that there is a problem, it offers no information about the cause of the problem or what can be done to alleviate it. It again goes back to the fundamental wrong assumption in MRP: assuming lead times are fixed and not dependent on the loading on the work-center.

3. **Input/output control**

A new addition to MRP II has been the input/output control, which is a method to keep the lead times under control. The I/O control works in the following way:

1. Monitor the WIP level at each work-center.
2. If the WIP goes above a certain level, then the current release rate is too high, so reduce it.
3. If the WIP level goes below a certain level, then the current release rate is too low, so increase it.
4. If it stays between these levels, then the release rate is correct for the current conditions.

It provides a simple way to check releases against available capacity. However, by waiting until the WIP levels have become excessive, the system, has in many respects, already gone out of control.

2.8 Conclusions

This chapter introduces the concepts required for the following chapters. We start with the requirements of a scheduling policy, followed by a classification system which helps users to identify the applicability of different scheduling policies to their manufacturing system. We then discuss research in time-based scheduling, which is the focus of this document. A candid appraisal of the MRP II clears the ambiguities associated with the exact structure of the program.

In the next chapter, we introduce the control point policy.

Chapter 3

Control point policy

3.1 Introduction

In Chapter 2, we discuss the role of a scheduling policy in managing a factory and the requirements of a good scheduling policy. We then carry out a critical appraisal of MRP: the most commonly used scheduling policy in the factories of today. Despite MRP's wide-spread popularity, users are highly dissatisfied with its performance and are always looking for better and more advanced techniques for manufacturing scheduling. There is no common consensus on which policy is the best. [Bonvik, 1996] shows that a hybrid policy (combination of two or more pure policies like KANBAN, CONWIP and basestock) gives the best performance (with respect to attaining high service rate and throughput goals at low inventories) in many cases.

Reaching the customer *on time* is an attribute where the companies of today are facing increasing pressure. Companies like Dell Computers are making their due date performance an advertisement slogan. Accordingly, there has been considerable interest in the development and the implementation of a pull-type scheduling mechanism driven by the due date in the recent years. A due date driven pull mechanism typically responds to the arrival of a demand (or order) and uses the attached due date to schedule the production. Also, many industries (typically those with long lead times) need a pull-type mechanism driven by future demand (in the form of forecasts) in addition to actual demands. Due dates include the ones which are promised by the company and the ones demanded by the customer. Also they include due dates internal to the organization and the ones associated with the final customer. Previous studies

([Salegna, 1996]) indicate that the scheduling rules which incorporate the due date information perform the best in such production environments.

[Hariharan and Zipkin, 1995] study the optimal use of advance-order information on inventory control. Their results indicate an intuitively appealing conclusion. Early information resolves future uncertainty and this information can be used to reduce inventory control costs effectively (and in some cases optimally). [Karaesmen et al., 1999] propose a mechanism for production coordination and control for a multi stage single part-type make-to-stock system which incorporates information on future customer orders. They also provide numerical examples for a single stage system which verify the effectiveness of the mechanism. [Duenyas and Hopp, 1995] study the impact and importance of quoting good customer lead times for factories. Incorporating order and release information in the scheduling mechanism has been studied in great detail from several different perspectives in [Tsai et al., 1997]. [Ragatz and Mabert, 1988] argue that uncontrolled release of orders to the shop-floor places added pressure on the internal despatch rules to control the timely progress of orders through the shop. [Wein, 1988, Wein] observes that the best place to control the inventory is at the incoming-material buffer. The decisions that are made once the material is inside the factory are less important (assuming they are reasonable). Overall, the issues of due date based scheduling, due date assignment, customer lead times, order releasing, and inventory management are becoming topics of increasing research.

These factors become more emphasized and complex in case of multiple products, with each having its own demand-pattern and processing through the factory. Different products hold different dollar-values to the company, and the management would like the scheduling policy to take this into account while making the scheduling decisions for a factory. New products are being introduced at a faster rate and huge premiums are associated with these products; the consequences of being late with a new product are significant ([Hendricks and Singhal, 1997]). A company would like to give priority to these new products, while limiting the disruptions to the flow and performance of the current products.

In this chapter, we introduce the *control point policy (CPP)*, which is born out of all the above concepts. The control point policy has three versions: surplus-based version, token-based version and time-based version. Though there is a difference in the applicability and implementation of each version to different manufacturing environments, the underlying theory (hedging point theory) is the same for all three versions. The primary focus of this document is on the time-based version of the policy[1]. Suitable analogies and relations can be easily drawn to the other two versions, and it is done wherever necessary. [Gershwin, 1999] provides a detailed discussion of all the versions of the policy.

The control point policy is a pull-type control mechanism driven by demand information in the form of due dates. The policy limits the flow of a product into the system and further downstream depending on two limitations which are imposed on each part-type. The first is a limit on surplus i.e. how far ahead of demand (or schedule) each part-type[2] is allowed to get. The second is a limitation on the inventory. A part-type is not allowed to flow into the system, or further downstream, if there are already too many of that part-type in the system. Part-types are rank ordered in advance and the high ranked part-types are given top priority in their flow through the system. This gives the company the choice to favor part-types depending on different concerns, which include holding costs, profit-margins, market considerations and space issues.

This chapter provides a detailed description of the control point policy. Section 3.2 gives an outline of the derivation of the policy from the underlying hedging point theory. Section 3.3 provides a formal statement of the time-based version of the policy. Section 3.4 briefly describes the token-based version of the policy. In Section 3.5, the policy is described in greater details: the assumptions, requirements and the different terms used are explained. In Section 3.6, the CPP (time-based and token-based) is compared with other commonly used time-based and token-based scheduling policies and intuitive reasons are given for better performance of the policy. Section 3.7 concludes the chapter.

3.1.1 Summary of the policy

The policy is described in detail in Section 3.2 and Section 3.3; here we summarize the key features of the simplest version of the policy. The time-based version of the policy, in its simplest form, can viewed as a *super-set* of the MRP and the KANBAN scheduling policies. The MRP logic controls the manufacturing stage in relation with the overall demand (global information),

[1]It was the version which was implemented at Boeing. Unless explicitly stated, the *control point policy* or the *CPP* refers to the time-based version in this document.

[2]A *part-type* is a synonym for *product*.

while the KANBAN logic controls the status with respect to its immediate neighborhood (local information). However, the CPP logic is more intelligent as compared to these policies, especially MRP, with the lead times *calculated and optimized* taking into account the system capacity and variability. The other advantage it has over these scheduling policies is that it provides a definite and simple way of selecting which part to work on at any work-center at any time. Also, unlike these policies, it provides a clear and definite strategy to get out of a crisis: relying little on the personnel's discretion or intuition. The real-time calculations are simple to perform. The policy does not need major changes in the factory.

The policy follows a two step process: establishing a fixed, prior set of rules; and executing a real-time algorithm. In advance, it gives a fixed ranking to all the part-types that flow through the work-centers. Then in real-time, at each work-center, the policy checks the following three conditions:

a. If a part is present in the upstream buffer;

b. If there is space in the downstream buffer;

c. If the part is ready according to a conservative schedule;

If and only if the answer is "Yes" to all the above conditions, the policy assigns a *green* flag to the part (now called green part). If the answer is "No" to any one of the three conditions, the policy puts a *red* flag against the part (now called red part). This is done for each part, from the top to bottom of the ranking list, and a list of *green parts* and *red parts* is produced. The highest ranking *green part* is loaded first on to the work-center. The next highest ranking *green part* comes next, and so on so forth.

All three conditions are essential. The first must be satisfied since a part that is not present cannot be worked on. The second prevents excess inventory from collecting on the floor. The third prevents the factory from working on parts that are early while others are late, and also limits inventory. The product ranking ensures that if more than one part survives the three filters, the highest ranking product gets the first priority.

The policy is modified to deal with the need for batching multiple parts at some work-centers. The proposed batching policy at work-centers, which require batching of parts, is as follows:

i. Check the three conditions for all the parts.

ii. Form logical batches from all the parts (including *red parts*).

iii. Load the batch with the highest ranking *green part* (the batch with the next highest ranking *green part* will come next and so on so forth).

iv. After processing a batch in the work-center, only release the *green parts* in the batch further downstream. Do not release the *red parts* until they pass the three conditions.

3.2 Derivation from first principles

The control point policy is an approximation and an extension of the *hedging point theory.* Readers may refer to [Kimemia and Gershwin, 1983] and [Gershwin, 1994] for the details of the hedging point theory. In this section, we briefly describe the single stage, single part-type application of the hedging point theory and then follow its extension to multiple stages and multiple part-types. Readers may refer to [Gershwin, 1999] for a detailed treatment of the analysis presented in this section. The control point policy is the outcome of these extensions, and it is stated after that.

3.2.1 Hedging point theory for single stage, single part-type

Consider an unreliable single stage manufacturing system M with a finished goods buffer B, producing a single part-type as shown in Figure 3-1. Let d be the demand rate, u be the instantaneous operating rate of machine M and μ be the maximum rate at which machine M can do an operation on a part. In this example, the demand rate d is kept constant for simplicity; it is more likely that it varies with time. Let x be the system surplus. It is the difference between cumulative production of the machine M and the cumulative demand d. It is not bounded below or above. If it is negative, the system is lagging behind the demand and the customers are dissatisfied. If it is positive and large, there is a large amount of inventory in the buffer B. Keeping x near zero is the major objective. Physically, it means that we want to meet all our demand, but at the same time we want to carry as little inventory in the finished goods buffer B as possible. There is a fixed positive penalty g_- for backlogging and a fixed positive penalty g_+ for surplus, such that:

$$g(x) = g_+x^+ + g_-x^- \tag{3.1}$$

where,

$$\begin{aligned} \text{if } x &\geq 0, x^+ = x \\ \text{if } x &< 0, x^+ = 0 \end{aligned} \tag{3.2}$$

$$\begin{aligned} \text{if } -x &\geq 0, x^- = -x \\ \text{if } -x &< 0, x^- = 0 \end{aligned} \tag{3.3}$$

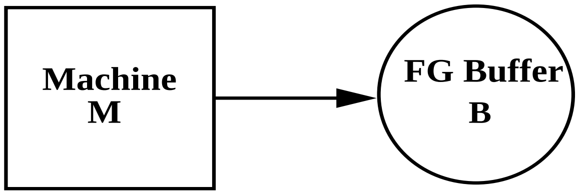


Figure 3-1: Single stage manufacturing system with single part-type

For this manufacturing system, the *hedging point theory* [Bielecki and Kumar, 1988] states that there is a number Z, called the *hedging point*, such that the optimal control law for the machine is:

$$\begin{aligned} \text{if } x &> Z, u = 0 \\ \text{if } x &= Z, u = d \\ \text{if } x &< Z, u = \mu \end{aligned} \tag{3.4}$$

whenever machine M is up. When the machine is down, $u = 0$.

It means that once the system surplus reaches Z, machine M needs to be operated at the demand rate d. When system surplus x exceeds Z, the machine should be kept idle and nothing should be produced. When x is less than Z, Machine M is to be operated as fast as possible to produce as much as possible.

Figure 3-2 illustrates the dynamic behavior of the hedging point theory for the above case. The horizontal axis represents time and the vertical axis represents cumulative production (P) and cumulative demand (τ) for the part. Since the demand rate is assumed to be constant, the cumulative demand during the time interval [0,t] is dt. The heavy broken line represents the cumulative production $P(t)$ of machine M in time period [0,t]. Since the surplus $x(t)$ is the difference between the cumulative production and the cumulative demand, the cumulative production at time t is

$$P(t) = x(t) + dt \tag{3.5}$$

When the machine is down, $u(t) = 0$ the cumulative production stays constant (since nothing is produced). When the machine is up and the surplus $x(t)$ is less than Z, $u(t)$ takes on its maximum value μ and the $P(t)$ graph is steepest. When the machine is up and $x(t)$ is equal to Z, $x(t)$ remains constant and

$$P(t) = Z + dt \tag{3.6}$$

3.2.2 Interpretation of Z

Equation 3.7 below shows the relation between surplus x, the penalty g_- for backlog and the penalty g_+ for surplus, if the above single stage, single part-type manufacturing system is operated under the hedging point control.

$$p(x \leq 0) = \frac{g_+}{g_+ + g_-} \tag{3.7}$$

Intuitively, a low g_+ and a high g_- represents a high company-commitment to a high service-rate. Depending on what service rate the company aims to achieve for the manufacturing system

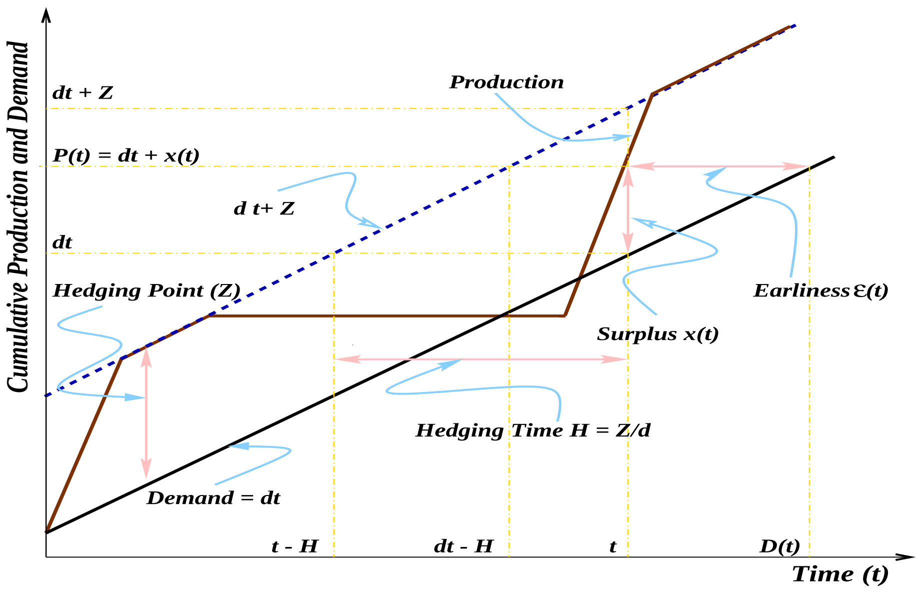


Figure 3-2: Dynamic behavior of *hedging point theory* for single stage, single part-type system

shown in Figure 3-1, the penalties for backlog (g_-) and surplus (g_+) would be decided. Based on these penalties the hedging point Z is calculated.

3.2.3 Time-based hedging point theory for single stage, single part-type

There is another way of analyzing the argument made in Section 3.2.1. Referring to Figure 3-2, at time t, the amount of material that has been demanded since time 0 is dt. The amount of material that was actually produced in time t is:

$$P(t) = dt + x(t) \tag{3.8}$$

The maximum that could have been produced according to the policy if the machine up time was not unfavorable, is $dt + Z$. It means that the required material dt, could have been produced earlier. The earliest it could have been produced under this policy was at time t^Z, such that

$$dt^Z + Z = dt \tag{3.9}$$

Dividing both sides of Equation 3.9 by d, we get

$$t^Z = t - HT \tag{3.10}$$

in which we define the *hedging time*,

$$HT = Z/d \tag{3.11}$$

This suggests the alternate statement of the hedging point policy. Define

$$D(t) = P(t)/d \tag{3.12}$$

The is the time we should have produced a total of $P(t)$ units. In manufacturing terms, $D(t)$ is the due date of the part which is being processed at machine M at time t. Again for simplicity, we have assumed that the due date allocation process to be deterministic, where the due date for the n^{th} part coming out of machine M, D_n is:

$$D_n = D_{n-1} + 1/d \tag{3.13}$$

Also, define

$$\varepsilon(t) = \frac{P(t)}{d} - t = D(t) - t \tag{3.14}$$

This quantity should be less than or equal to the hedging time HT.

$$D(t) - t \leq HT \tag{3.15}$$

The quantity $\varepsilon(t)$ represents the *earliness* of the part at time (t). The inequality says that the earliness of the part should be less than or equal to the hedging time. In other words, machine M should work on the part if and only if the present time is later than or equal to the due date of the part minus the hedging time. We can therefore interpret the hedging time HT as a conservative estimate of the time we should allot for the process at the Machine M. It represents how much earlier than a part's due date we would like to start the operation on it at Machine M. It is not simply the processing time of the part at Machine M, but it also takes into account the possibility of failure machine M and the penalty for backlog (g_{-}) and surplus (g_{+}).

3.2.4 Extension for multiple stages, multiple part-types

Generalizations to multiple part-types and multiple production stages have proved to be difficult. [Gershwin, 1999] provides a detailed outline on the research in this area. This research has been focussed on the *surplus-based version* of the policy. Our goal, in this section, is not to analyze the research, but to give an outline of its findings, and translate them to corresponding implications for the *time-based version* of the policy.

Extension for single stage, multiple part-types

Suitable approximations and simplifications have to be made in order to extend the hedging point policy to multiple part-type production in a single machine. [Srivatsan and Dallery, 1998] were able to obtain a solution for a two part-type system, although under the special case that the hedging point ($Z_2 = 0$). Assuming that the policy can be extended to a single stage, n part-type system (without set-up costs), the policy is a special case of [Rishel, 1975] and [Kimemia and Gershwin, 1983]. Translating this surplus-based policy statement into a time-based statement, the resulting time-based version states:

0. Assign a fixed ranking to each part-type before executing the policy. Give the highest ranking to the most valuable part-type.

1. Produce the highest ranking part-type with earliness less than or equal to its hedging time. Do it until the earliness of the part-type is equal to the hedging time. The other part-types fall behind their respective hedging times.

2. Keep the earliness of highest ranking part-type at the hedging time. Devote the rest of the capacity to the next highest ranking part-type until its earliness reaches its hedging time. The lower-ranked part-types fall further behind their hedging times.

3. Keep the earliness of the two highest ranking part-types at their hedging times. Devote the rest of the capacity to the third highest ranking part-type, until its earliness reaches its hedging point. The lower-ranked part-types fall still further behind their hedging times.

4. etc.

5. If two or more parts have equal ranking, produce the part whose earliness is furthest behind its hedging time (i.e. the one with the earliest due date) while the earliness of all other higher ranking part-types is at their hedging times.

The statement can be simplified to: *Produce the highest ranking part-type, whose earliness is less than its hedging time.* The part-type rankings follow the decreasing values for the penalty that the business suffers for each unit of part-type, per unit time for which it is backlogged.

Extension for multiple stage, single part-type

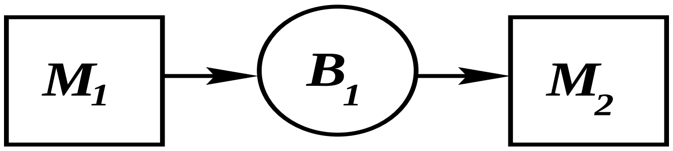


Figure 3-3: Two stage manufacturing system with single part-type

[Ryzin, 1987] and colleagues [Ryzin et al., 1993] approximated numerical solutions for two stage, single part-type systems as shown in Figure 3-3. [Gershwin, 1997] approximates an optimal control policy for the same two stage, single part-type systems. The solution is similar to the one obtained for a single stage single part-type system (surplus-based version), but has to be modified to take into account the presence of the finite buffer $B_1$of size N between the two machines. Since buffer level b_1 represents the difference between instantaneous surplus of Machine M_2 and Machine M_1 i.e. $x_1 - x_2$, we must have $0 \leq Z_1 - Z_2 \leq N$. The policy takes the form:

For $i = 1, 2,$

$$\text{if } x_i > Z_i, u_i = 0; \tag{3.16}$$

$$\begin{aligned}\text{if } x_i &= Z_i, u_i = d;\\ \text{if } x_i &= Z_i, \text{ choose } u_i \text{ as large as possible.}\end{aligned}$$

In this expression, "as large as possible" means

- if $0 < x_1 - x_2 < N$, then $u_1 = \mu_1$ and $u_2 = \mu_2$;
- if $x_1 - x_2 = N$, then $u_1 = min(\mu_1,\mu_2)$ and $u_2 = \mu_2$; and
- if $x_1 - x_2 = 0$, then $u_1 = \mu_1$ and $u_2 = min(u_1,\mu_2)$.

Intuitively, it means that if buffer B_1 is neither non-empty nor non-full, then both machines should work as fast as possible. If buffer B_1 is full, then machine M_2 should work as fast as possible, and machine M_1 can not work any faster than M_2 (because of blockage). If the buffer B_1 is empty, then machine M_1 works as fast as possible and machine M_2 can not work any faster than this rate (because of starving).

[Lou and Ryzin, 1989] extend Van Ryzin's results to tandem three machine systems with single part-type and conjecture near optimality for the policy. [Lou and Kager, 1989] and [Yan et al., 1996] apply the models to a semi-conductor manufacturing facility and get good performance.

Following this literature, we can generalize the policy statement to lines of arbitrary length. The buffer information for the two-stage system described above has be to be extended to each and every stage in the line. Also, the control policy discussed in Section 3.2.4.1 needs to be followed for multiple part-types. The policy becomes decentralized i.e. each machine has to take a decision based on the local dynamic information. For a machine, this information consists of the surplus status (the earliness of the parts in the upstream buffer) and the status of downstream buffer.

3.3 Statement of the control point policy

The control point policy is based on the above extensions of the hedging point theory to multiple stages and multiple part-types discussed in Section 3.2. We conjecture that the policy should give good performance. The emphasis, in this section, is on giving an accurate statement of the simplest version of the policy, issues of modifications and implementability are addressed in the subsequent sections and in Chapter 4.

So far, we had assumed that a work-center operates on one part at a time. But there are a few processes which require the grouping of more than one parts. The policy needs to be extended slightly for these batching work-centers to take into account the batching of more than one part. Therefore, there are two statements of the policy. The first statement is for the *non-batching work-centers* which operate on a single part at a time. The second statement is for *batching work-centers* which operate on more than one part at a time[3].

We define a *work-center* as either single machine or a group of machines or operations, which can be treated as a single stage in the production. Also, the reentrant nature of the material flow makes it essential to specify both the part-type and the production stage that it has reached. We therefore define *part sq* to be part-type q at stage s.

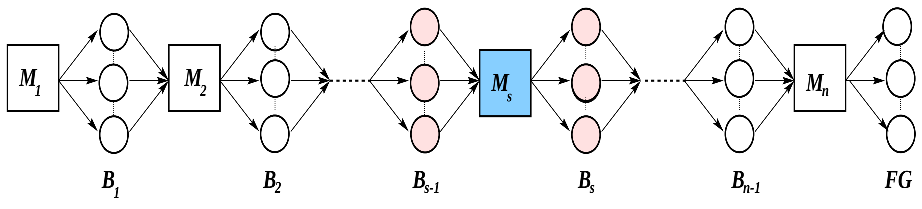


Figure 3-4: Mutiple stage and multiple part-type manufacturing system

Consider the system shown in Figure 3-4. Like most other control policies, the control point policy has two elements: the preparatory phase (Step 0), where the parameters are selected and the real-time phase (Step1 - Step 5), where the decisions are made. Figure 3-5 shows a flow-chart for the real-time algorithm of the control point policy.

3.3.1 Policy for non-batching work-centers

0. Before executing the policy, calculate the input parameters for the policy. The *CPP input parameters* are the following:

a. Ranking of *part sq's*

[3]Unless otherwise explicitly stated, the *control point policy* or *CPP* refers to the policy for *non-batching work-centers.*

b. Buffer sizes (B_{sq})

c. Hedging times (HT_{sq})

The real-time calculation need not be done at each time instant. It can be executed when a work-center completes an operation or when a failed work-center gets repaired (i.e. when it becomes idle). At each work-center M_s, use the following five step real-time algorithm:

1. Get a list of all parts present in $B_{(s-1,q)}$. If the there are no parts present in $B_{(s-1,q)}$, M_s is starved. This is called the *starving condition.* Exit calculation.

2. Put a *red* flag against all parts in $B_{(s-1,q)}$ for which buffer $B_{(s,q)}$ is full. This condition is called the *blocking condition.* The *red parts* are also called *not ready parts.*

3. In the remaining parts, put a *red* flag against all the parts whose earliness $\varepsilon_{(s-1,q)}$ exceeds the hedging times $HT_{(s-1,q)}$. This condition is called the *earliness condition.* Mark a *green* flag against the remaining parts. The *green parts* are also called *ready parts.*

4. The *red parts* can be ignored. Amongst the *green parts*, select the highest ranking part-type to load on Machine M_s. If there is more than one *green part* with the same ranking, operate on the part in $B_{(s-1,q)}$, which has the earliest due date.

5. If there are no *green parts* in $B_{(s-1,q)}$, wait until a part does become *green*, and then go to Step 4.

3.3.2 Policy for batching work-centers

The above real-time algorithm is sufficient for work-centers, which do not need batching of compatible parts. The algorithm needs to be modified for work-centers which batch more than one part for their operation. Step 0 - Step 3 and Step 5 remain the same as above, but the policy is modified in Step 4. The policy for batching work-centers is:

1. Same as above

2. Same as above

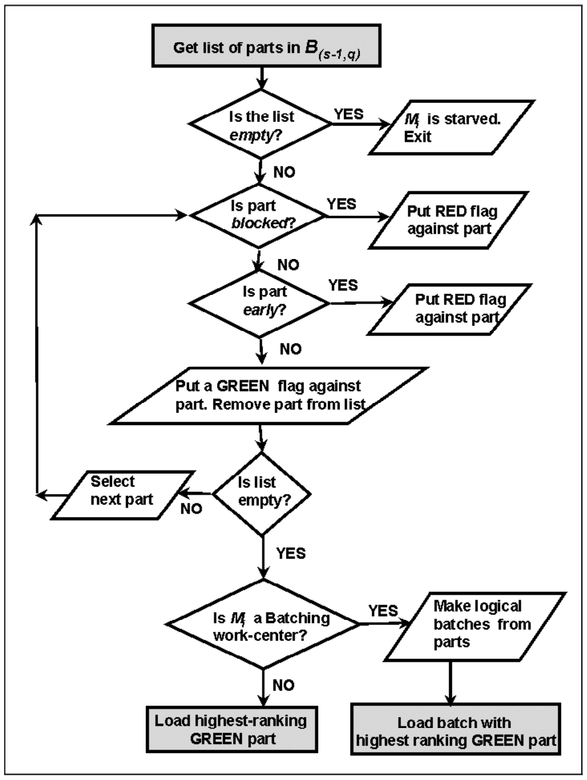

Get list of parts in $B_{(s-1,q)}$
Is the list empty?
YES
M_s is starved. Exit
NO
Is part blocked?
YES
Put RED flag against part
NO
Is part early?
YES
Put RED flag against part
NO
Put a GREEN flag against part. Remove part from list
Select next part
NO
Is list empty?
YES
Is M_s a Batching work-center?
YES
Make logical batches from parts
NO
Load highest-ranking GREEN part
Load batch with highest ranking GREEN part

Figure 3-5: CPP flowchart for selecting the first part to load

3. Same as above

4. a. Form logical batches from the parts that are present in buffer $B_{(s-1,q)}$. These may include *red parts.*

b. Load the batch with the highest-ranking *green part* first. The batch with the next highest-ranking *green part* will come next and so on so forth.

c. After processing a batch in the work-center, only release the *green parts* in the batch further downstream. Do not release the *red parts* until they become *green.*

5. If there is not batch with a *green part*, wait until a part becomes *green* and then got to Step 4.

The dynamic behavior of the policy for a part sq at a machine M_s is illustrated in Figure 3-6. The vertical axis represent the cumulative demand for type q (d_qt) and the cumulative production of type q parts at stage s (the heavy broken line).

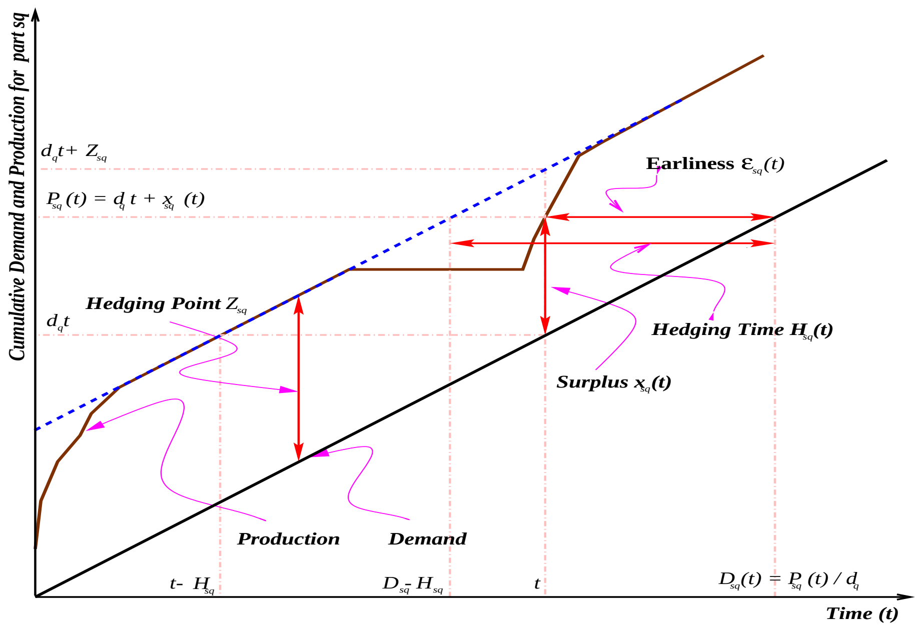


Figure 3-6: CPP - dynamic behavior for a part at a work-center

The figure is similar to Figure 3-2. However, the production line does not follow a single slope each time it makes it back to the hedging point line $(d_q t + Z_{sq})$ i.e. its earliness $(D_{sq} - t)$ catches up with its hedging time (H_{sq}). Its slope changes each time the capacity is shared with a new part, each time an upstream buffer changes from empty to non-empty (or vice versa), and each time a downstream buffer changes from full to non- full (and vice versa).

3.4 Token-based statement of the policy

An important and popular class of real-time scheduling policies are token-based (e.g. KANBAN, CONWIP, basestock). In this section, we introduce and briefly describe the token-based version of the control point policy. The intention is to provide a framework adequate for comparison with the other token-based policies in Section 3.6.

Consider a simple three machine production line with single part-type as shown in Figure 3-7. We super-impose a token-circulating system on this manufacturing system. The resulting figure is shown in Figure 3-8. The dark part of the figure represents the actual physical system, while the light part with dashed arrows represents the token-control system.

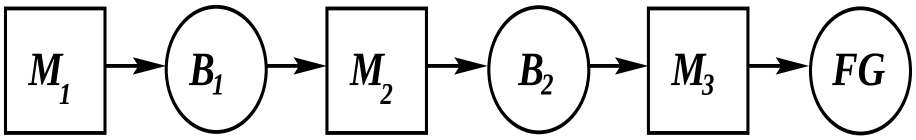


Figure 3-7: Simple three machine tandem line

Machine D represents the demand arrival process. When the sales department confirms an order, a token is placed into each buffer BL_i. When Machine M_i completes an operation, it moves a part from buffer B_{i-1} to buffer B_i, and it also creates a token and adds it to buffer SP_i. If buffer B_{i-1} is empty, or if buffer B_i is full or if buffer SP_i is full, then machine M_i cannot perform an operation. Synchronizing machine S_i is an infinitely fast and 100% reliable assembly machine, which takes a token each from buffer SP_i and buffer BL_i and assembles them. It can operate only if buffer SP_i and BL_i are non-empty. Buffers SP_i and BL_i can never be simultaneously non-empty, because if there is a token in both buffers, enough tokens will be removed from both of them by machine S_i, so that at least one of them becomes empty. Machine S_3 is a bit different; it also regulates the delivery of completed parts to the customers from the finished goods buffer FG.

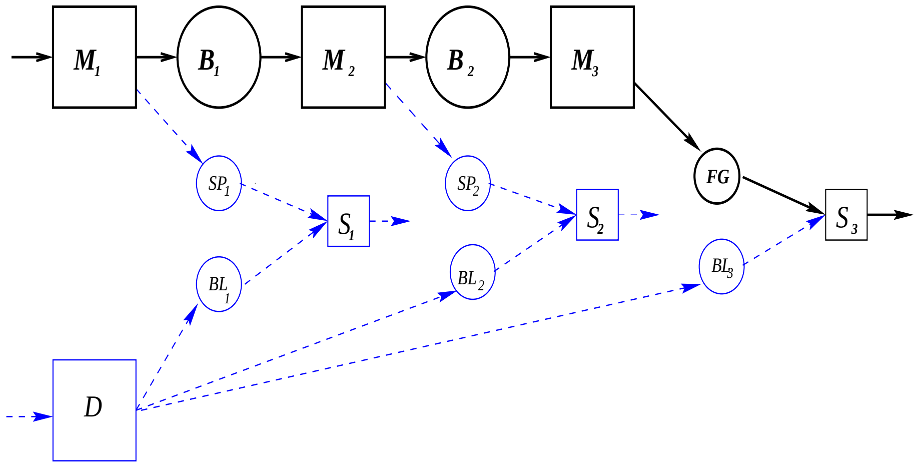


Figure 3-8: Token-based CPP for three machine line

The buffers SP_i in the above structure, which have size Z_i, perform the function of the local hedging points for each machine M_i. They regulate how far ahead of demand each machine M_i gets. That is they control the local surplus of each machine. The buffers BL_i represent the local backlog for each machine M_i.

If we reduce the above three stage system to a single stage one, consisting only of machine M_3, buffers B_3, FG and BL_3 and synchronizing machine S_3, [Bonvik, 1996] proves that the behavior of this system is identical to the system of Section 3.2.1 (single stage, single part-type hedging point control). It can be treated analytically for a given Z, and this Z can be determined.

The token-based system described in the above paragraphs is an extension and generalization of this single stage system to a three stage system. It can be extended to more than three stages. Also a similar token system for the second (and third, and so on so forth) part-type can be constructed in addition to the token-system for the first part-type to regulate the flow of the multiple part-types. We also rank the different part-types in advance, with the highest ranked part-type being operated at a machine if it is not blocked or starved at a machine. We claim that the behavior of this token-based system is identical to the time-based CPP (for a constant demand and due date allocation process)[4].

The information flow is of both types: global information flow and local information flow. *Global information* flows directly connect individual stages (machines or buffers) to the demand process, without going through intermediate manufacturing stages. *Local information* flows connect machines and buffers to adjacent machines and buffers.

3.5 Working

In the previous sections, we first derive and extend, conjecture and then gave a formal statement of the control point policy. In this section, we look at the CPP from an implementation point of view. Section 3.5.1 outlines the assumptions we have made. Section 3.5.2 analyses the requirements for establishing the policy in a factory. In Section 3.5.3, we introduce and explain the terms which will be used from this point onwards.

[4]Detailed discussion in Section 7.1.

3.5.1 Assumptions

In this section, we discuss the assumptions we have made regarding the type of manufacturing system and the products. The first three are absolutely necessary for the CPP implementation, but they are applicable to most factories of today. The next five are preliminary assumptions, which we have made for the sake of simplicity. They are not absolutely rigid and can be relaxed. Out of these five, the first two were, in fact, relaxed and suitably modified during the course of the Boeing experiment. The next three assumptions were less important in the Boeing experiment and, therefore, did not have to be modified. But these are flexible too and can be addressed in the future implementations of the policy.

1. **Capacity of the manufacturing system**

   The policy assumes that the capacity of the manufacturing system is finite and exceeds the demand on the system. The capacity of the system is the maximum total production for all the products. This should exceed the maximum total demand of all the products. This assumption is necessary for the stable performance of any scheduling policy. By finite we mean that the different resources though flexible, cannot be expanded, contracted or reallocated in an unconstrained manner.

2. **Due dates for products**

   Each product has a due date. The due date may be a time-target which is applicable for the shop-floor, the plant or the company depending on the operations and the products.

3. **Static Priority ranking**

   Each part sq has a ranking which is fixed in advance (*in Step 0*). If the flow is reentrant, each part sq at each stage of operation has a fixed ranking.

4. **Batching**

   In the preliminary formulation of the policy we had not dealt with the issue of batching. But it became clear during our implementation at Boeing, that batching of multiple parts was a major issue at some work-centers. The batching policy stated in Section 3.3.2 is one possible one (which was implemented at Boeing); other modifications are possible.

5. **Finite and homogenous buffers at each stage**

   We assume that there are finite buffers for each part-type at each stage of manufacturing. The buffer maybe physically present. However, it is not necessary and the buffer can be *only logical* that is the CPP is able to calculate the buffer levels at each stage. The assumption of having these finite and homogenous buffers *at each stage* of manufacturing had to be relaxed[5] during the implementation at Boeing.

6. **Multiple products sharing work-centers**

   We assume that there are multiple products in the factory, each of which has its own routing through the factory. Each product also has different processing times and batching polices at different work-centers. It is not a total part shop: the products, the routings and processes are reasonably known in advance. This is the system we encountered during our experiment with Boeing. Application to other kind of systems should be possible, though extensions and modifications may have to be done.

7. **Set-ups and scrapping**

   The issues of set-up (costs and times associated with change-over between parts at a work-center) and scrapping (parts being discarded mid-way) have not been dealt with in this formulation of the policy. Both these issues were not of major concern in the implementation at Boeing. However, they may be important under a different manufacturing environment. These issues are currently under investigation.

8. **Limiting resources**

   We have assumed that machines and parts and not other resources (e.g. people, tools) are the limiting constraints in the model. This is also the assumption of the underlying hedging point theory. If these resources are the limiting resources in the factory, then the model needs to be modified.

[5]Details in Section 5.6.6.

3.5.2 Requirements

This section describes the information that a factory has to maintain and provide in order to implement the control point policy. We can organize the information into three categories: *up-front*, *real-time* and *additional*.

1. **Up-front information**

Up-front information is the information needed for the preparation phase of the policy (*Step 0*). This information is required to decide the *CPP input parameters*. The information is as follows:

- List of part-types with details[6]
- List of work-centers
- Process flows through the factory for each part-type
- Processing times for each part-type at each work-center
- The reliability for each machine i.e. the Mean Time to Failure (MTTF) and the Mean Time to Repair (MTTR)
- Batch sizes and compatibilities at a batching work-center
- Inter-operation travelling time estimates for parts
- Set-up policy and set-up times for the work-centers
- Scrap rates at each work-center

This information is presently used to conjecture the *CPP input parameters* in *Step 0* of the policy. Section 4.2.2 provide details on how the up-front information is used to conjecture the rankings for each part-type at each work-center. Section 4.2.5 describes the details on the use of the up-front information for conjecturing the buffer sizes and the hedging times for each part-type at each work-center. In due course of time, the up-front information will be used to *calculate and optimize* the *CPP input parameters*.

2. **Real-time information**

[6]The details needed are discussed in Section 4.2.2.

The real-time information is the information needed each time the CPP algorithm (Step 1-Step5) is run for the factory.

- Operating state of each work-center
- List of parts waiting at each work-center
- List of parts in each buffer
- Due date for each part

3. **Additional information**

Additional information is the information collected as feedback from the system under CPP control. This information is required to decide the implementation issues for the policy. It is also used to fine-tune the *CPP input parameters*, until they are able to be analytically calculated. The additional information required is as follows:

a. Frequency of failing the *blocking condition* for a part at a work-center

b. Frequency of failing the *earliness condition* for a part at a work-center

c. Average queuing time for a part at a work-center

d. Frequency of over-riding the CPP output

e. Performance of *batching policy*

Section 4.4.2 describes the details on the additional information and its utility.

3.5.3 Definitions

In this section, we define the different terms which will be used in this document henceforth. Some of the terms have been introduced and used in the previous sections, but they are exactly defined in this section again.

Definition 1 ***Control point***

A *control point* is a point (work-center, department or a mechanism) where we apply the policy fully. At this point, we use the control point policy to decide the schedule for processing of work. The other resources in the factory can be then called *non-control points.*

Definition 2 ***Hedging time***

A *hedging time* is the time we allow a part between the control point and the end of the process i.e. finished goods area. It is a conservative estimate of the lead time between these two points. It takes into account the effect of machine failures, queuing, set-ups, batching, yield and resource-contention on the system. It also takes into account the conflicting risks of missing the delivery dates and carrying high inventories for a part-type. It is definitely greater than the sum of the remaining processing times in the system.

Definition 3 ***Earliness***

The *earliness* of the part is the difference between the due date of the part and the current time. Depending on needs of the manufacturing system, the unit for earliness could be months, days, hours, minutes or seconds.

Definition 4 ***Available work-center***

A work-center is *available* if it is idle i.e. it is in a good working condition and it is not working on anything. In other words, if required, a part can be loaded on it immediately.

Definition 5 ***Available part***

A part is *available* at a control point if it is present and can be loaded on to the work-center, and if the buffer of that part immediately downstream of the control point is not full. In other words, the control point is not starved or blocked for that part. It also means that the part would pass the *starving condition* and the *blocking condition* of the CPP logic.

Definition 6 ***Ready part***

A part is *ready* at a control point if it is available *and* if its earliness is less than the hedging time for the part. The part is also called a green part i.e. it would pass *all three conditions* of the CPP.

Example *The current time is 8.00 PM on 1st January and the due date for the part is 12.00 PM on 16th January, giving an earliness of 15 days and 4 hours for the part. The part is ready, if it is available and if the hedging time for the part is greater than 15 days and 4 hours.*

Definition 7 ***Disorderly mingling of parts***

Companies normally release parts into a system in an orderly fashion depending on their due dates, TAKT times and estimated lead times. However, once into the system, the parts are exposed to a variety of random events, such as machine failures, different routings, different processing, traveling and queuing times, different batching and set-up policies in addition to the random availability of other resources (people, tools). All these events lead to a re-sequencing of parts. We define this effect in a factory as the *disorderly mingling of parts.*

Definition 8 ***Resource contention***

If the capacity at a resource is finite and there is more than one part which can be loaded on to the resource, then the parts are said to *contend for the resource.* Once a resource is assigned to a part, the effective capacity of the resource seen by the other parts is reduced.

Definition 9 ***Lateness***

Lateness for a part is defined as the difference in time between its due date and the time it reaches the finished goods buffer, if the part arrives there after its due date. We define it to be zero, if the part arrives the finished goods buffer before its due date.

Definition 10 ***High-value products***

Products with high priority in the CPP ranking procedure are referred to as *high-value products.* The company can use different criteria to rank the products. These criteria include holding costs, profit margins, and space issues (Refer to Section 4.2.2 for details).

3.6 Comparison with other policies

In this section, we compare the CPP with all the policies introduced in Section 2.3. We provide intuitive reasons for expecting better performance of the CPP over other policies. *We expect the CPP to provide better performance as compared to other scheduling policies, whenever the effect of disorderly mingling of parts can be large.*

The time-based statement of the policy will be used to compare it with the time-based scheduling polices, while the token-based model will be used to compare it with the token-based policies.

3.6.1 Time-based policies

1. **First in first out (FIFO)**

The policy aims to preserve the order in which parts arrive. It is very simple to implement because the only data required is the history of arrival times; that is also the reason many companies use it. However its usefulness stops at that; companies can and do maintain more information about the parts in the system and this information can be used to sequence parts in a more refined and sophisticated way. The FIFO policy fails in any system in which the disorderly mingling of parts is even reasonably high, because the order in which parts arrive at an intermediate work-center is most often not the correct order. Also the policy does not take account into the value of the part-type, the part's status according to the due date and schedule.

2. **Earliest due date (EDD)**

The earliest due date (EDD) policy is very frequently used in practice. Previous studies [Salegna, 1996], have found that the scheduling policy which incorporates the due date information performs the best, with EDD giving the best performance among these.

The EDD has a very simple rule: At a machine M_i select the part in the queue with the earliest due date (D_{nq}). For a single stage single part-type system, [Baudin, 1990] proves that EDD minimizes the maximum lateness for the parts.

This is done irrespective of the penalties for backlog and surplus for the part-type, though the same lateness is worth a lot more for the high-value part-types as compared to the lower ranked part-types. Also, the policy does not take into account how the part is doing according to a schedule. Even if a part is ahead of schedule, it will be done first because it has the earliest due date, when experience indicates that it may be better to do a part that has more work ahead of it. There is no real-time feedback from the system, about the current earliness status of the part, or about information on the buffer status. This becomes more emphasized in case of re-entrant manufacturing systems with multiple products having which hold different dollar-values for the company.

Whenever there is large disorderly mingling of parts inside the system, the performance of EDD will be hampered because of its inability to account the status of the part according to the schedule.

3. **Least slack policy**

Another policy, which has received considerable attention over the last few years is the least slack policy (LSP). At a machine M$_{i,}$ it assigns an expected amount of time ζ_{sq} (*time required*) before a part sq leaves the system. In the least slack policy, as defined in [Lu and Kumar, 1991], the *slack* for a part sq is defined as the difference between its due date (D_{nq}) and the ζ_{sq}. The machine M$_i$ then chooses the part sq with the least slack. [Lu and Kumar, 1991] also claim good performance for the policy. For a single stage single part-type system, [Baudin, 1990] proves that the LSP maximizes the minimum lateness of the parts.

The LSP and the CPP are very similar, if the *time required* ζ_{sq} in the LSP and the *hedging time* HT_{sq} in the CPP are the same. However, there is a difference in the way these two quantities are determined under each policy. ζ_{sq} in the LSP is the expected lead time from that point onwards. It is often simply the *sum of the standard lead times* of the operations to be performed between sq and the finished goods buffer. HT_{sq} in the CPP is a *conservative estimate of the lead time* from that point onwards. It takes into account the effect of failures, resource contention, batching and set-ups on the lead times of the operations in addition to the penalties for backlog and surplus.

There are other differences:

a. The LSP assumes that each part-type is equally valuable to the company, which is not true in most cases.

b. The LSP offers some advantages over the EDD policy, because it takes into account the current status of the part according to the schedule. However, the LSP does not distinguish between parts which are earlier than the slack time ($D_{nq} - \zeta_{sq}$) and those that are not. Instead, it ranks the parts in the order of their slack time and selects the part with the smallest slack time first - the one with the smallest value $(D_{nq} - \zeta_{sq})$, even if this difference is later than the current time. This might lead to parts being done earlier than they should be done. And with no *blocking condition* like the CPP, this might to lead unnecessary flooding of downstream work-centers with material. Since no other criterion is used to select the part, it can lead to delays for the high-value products. This effect is emphasized when the utilization (demand/capacity ratio) and resource contention is high and the need do load the right parts at the right time is crucial. In the CPP, all other things being equal, priority is given to the most valuable part-type.

3.6.2 Token-based policies

In this section, we use the token-based version of the CPP to compare it with the other token-based policies. The framework of local and global information flow is used for comparison. We give intuitive reasons for the better performance of the CPP. We also show that all the token-based polices discussed below can be looked as special cases of the CPP. Intuitively, since the CPP contains the other policies as special cases, finding its parameters will be an optimization over a larger space than in these policies. It follows that the best parameter choice for the CPP policy will perform at least as well as the best of the other policies with optimal parameter choices.

1. **KANBAN policy**

KANBAN control uses local information flows only, as shown in Figure 3-9. It is often referred to as a “pull” policy, because each production stage in the chain reacts only to demands from the next downstream buffer.

The KANBAN policy, in its simplest version, is similar to the tandem line with finite buffers [Berkley, 1991]. It ensures the overall production rate by making a few parts ahead of time and using the finite buffers to limit the amount of inventory in the system. It also uses the buffers to control, how far ahead of the demand process the part is allowed to get. The flow of information is *only local* in nature, and the only way the demand perturbations can reach an individual machine is by propagating up to it as blockages or starvations. This is a very expensive way (in terms of inventory and time) of communicating demand fluctuations.

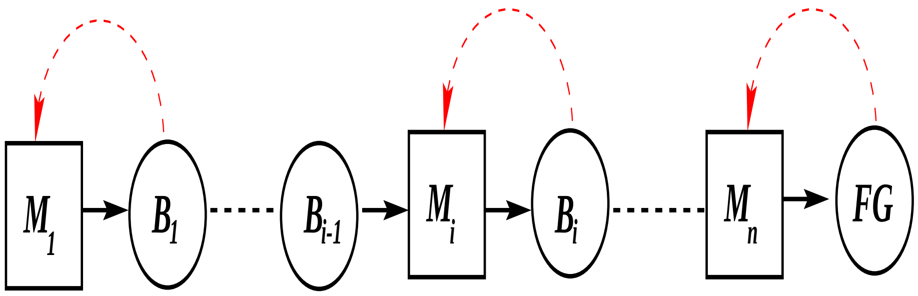


Figure 3-9: Information flow in KANBAN control

Therefore, the performance of the policy under fluctuating demand process is not satisfactory because the response is slower.

On the other hand, the CPP regulates the production rate by conveying the demand process to each machine in the form of a token (global information flow). In cases where the demand process is variable, the CPP should give better performance than the KANBAN system. The CPP should also outperform the KANBAN policy under high utilization and varied demand patterns for different products. The ranking system under the CPP accelerates the high-value part-types through the system.

Referring to Figure 3-8, we can see that local information flow is identical to the KANBAN flow (if $SP_i = B_i$). When the demand is constant, the token-network of the Figure 3-8 becomes redundant if the parts are released into M_1 at the demand rate. The network then reduces to a simple tandem line with finite buffers. Therefore, KANBAN can be looked as special case of CPP, when the demand is highly deterministic.

2. **Basestock policy**

Basestock control limits the amount of inventory between each production stage and the demand process as indicated in Figure 3-10. Each machine tries to maintain a certain amount of material in its output buffer, subtracting backlogged finished goods demand, if any. This amount is called the *basestock level* of the machine. There are no intermediate buffer limits and the flow of information is *only* global in nature. Therefore, the machine failures and other random events do not propagate up or down the line. This leads to high inventory levels for the same target production rate in case of lines with unreliable machines.

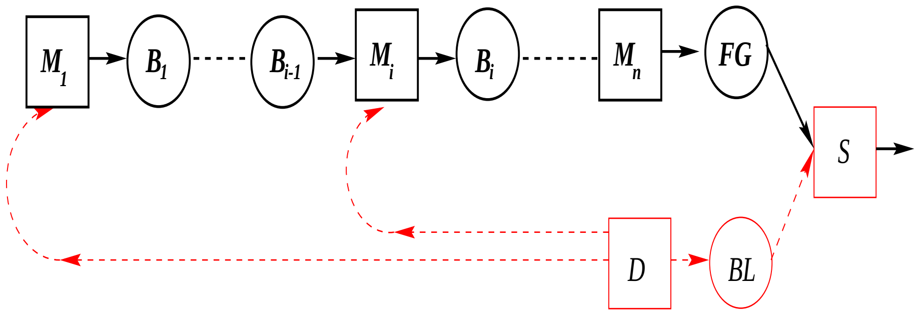


Figure 3-10: Information flow in basestock control

However, this feature is useful in recovery from failures: When a machine fails, the demand process continues to remove parts from the finished goods inventory, and the machines downstream of the failure operate normally until they become starved of parts to process. The machines upstream of the failed machine receive demand information directly from the demand process and operate as usual. There is, therefore, a build-up of inventory in front of the failed machine until it is repaired. When this occurs, the machine finds sufficient material to catch up with demand in its input buffer.

The basestock control for *single machine and single part-type* is closely related to the hedging point control for *single machine and single part-type* developed from optimal control formulations [Bai, 1991] and [Gershwin, 1994]. The information flow is the same as in basestock control, but we refer to the resulting control as a basestock control if it responds to discrete demand events, and as a hedging point control if it tracks a target demand rate.

For multiple stages and multiple part-types, the *global flow of information* in basestock control (with backlog) is identical to the CPP. If the internal buffers (B_i, SP_i, BL_i) in the CPP system shown in Figure 3-8 are infinite, then there is *no local flow of information* and the basestock control becomes identical to the CPP. Therefore, the basestock control can be looked upon as a special case of CPP, with no local information flow i.e. with infinite physical and token buffers.

3. **CONWIP policy**

CONWIP control maintains a constant work in progress (WIP) i.e. the total amount of inventory in the system [Spearman et al., 1990]. When the preset WIP level is reached, no new part is authorized for release into the system until a part leaves. Demand that arrives when the system is full is usually kept in an external backlog. When a demand is served, a message is sent to the first stage in the line, authorizing the release of a new part to the system as shown in Figure 3-11. The intermediate buffers inside the system can be considered infinitely large, or at least as big as the preset WIP level. In other words, the flow of information is only global, but the flow is restricted to *only the first machine* in the line.

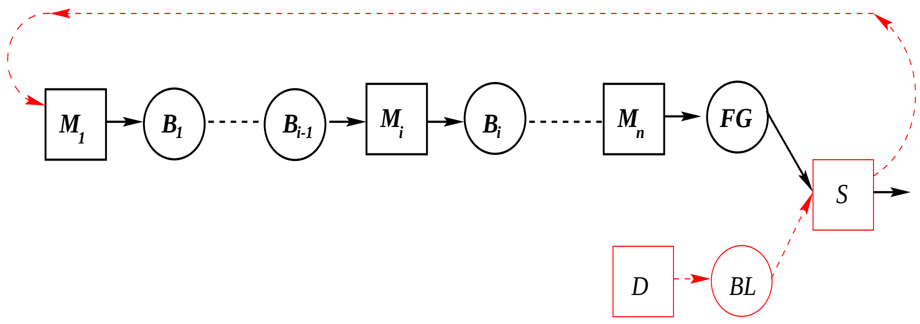


Figure 3-11: Information flow in CONWIP control

The CONWIP control gives good performance in many cases [Hopp and Roof, 1998]. However the policy suffers, when there are very unreliable machines in the line, because the failure recovery process is not very efficient. If a machine fails in a CONWIP line, the parts downstream of it are eventually removed from the system by the demand process. These demand events cause the release of new parts to the system. If the machine stays down long enough, these parts and the parts already in the system upstream of the failed machine accumulate in the buffer immediately upstream of this machine. CONWIP control stops admitting new parts to the system when the inventory limit is reached and this can start immediately after the failure has occurred, if there are already many parts inside the system.

On the other hand, CPP (and basestock) control allow parts to accumulate in front of a failed machine because demand tokens are sent to each and every machine. This is the difference between CONWIP and the CPP control. Therefore, the CPP performance should be better than CONWIP in case of lines with unreliable machines. Also since there is no product priority, the high-value products do not get an accelerated flow through the system.

For multiple stages and multiple part-types the global flow of information in the CPP control is identical to the CONWIP control if in Figure 3-8, the following changes are made:

1. For $i > 1$, buffers $B_i = \infty, SP_i = \infty, BL_i = \infty$.
2. The demand machine D triggers a token when a demand is served from the FG buffer rather than when a demand arrives.
3. The size of buffer SP_1 buffer is made equal to the CONWIP limit and the size of buffer B_1 is made ∞ (or equal to SP_1). The size of BL_1 is made equal to the size of buffer BL in Figure 3-11.

Therefore, the CONWIP control can be looked upon as a special case of CPP control with the modification in the way in which demand tokens are triggered and that tokens are communicated only to the first machine.

3.7 Conclusions

This chapter has laid the foundations for the further chapters in the thesis. We have provided a formal statement of the control point policy, and conjectured good performance for the policy. Also qualitative explanations for better performance over other commonly used policies were discussed.

The policy is still in its incipient stages of development, with many of the details still in the process of being formulated. As we find in the later chapters, there are some adjustments and modifications which have to be made to the policy as stated in this chapter, to make it implementable. Also there are subtle differences in the different versions (token-based and time-based) of the policy, which make their applicability to different manufacturing environments a case for more study.

Chapter 4

CPP implementation in a factory

4.1 Introduction

In Chapter 3, we introduce the control point policy and explain the requirements to establish the policy in a factory. Now that we have developed the foundation, the next step is to investigate its application to a real factory environment. In Chapter 3, we had also say that the policy "as stated" cannot be implemented in many factories and needs to be modified to make it implementable. As we realized during our experiment with Boeing[1], we came up with many issues which were not addressed in the original statement of the policy.

This chapter is about establishing and using the CPP as a scheduling tool in a real factory: we go through each and every step of implementation. It will serve as a guide for future CPP implementations at other factories. Possible modifications and variations to the original policy are discussed; which of them would apply is specific to the factory under concern. Section 4.2 discusses the steps involved in the prior preparation activities for the CPP. In Section 4.3, we provide the details of the real-time CPP program from an implementation perspective. The systems metrics which need to be measured are discussed in Section 4.4. Section 4.5 concludes the chapter.

[1]Details on the experiment in Chapter 5.

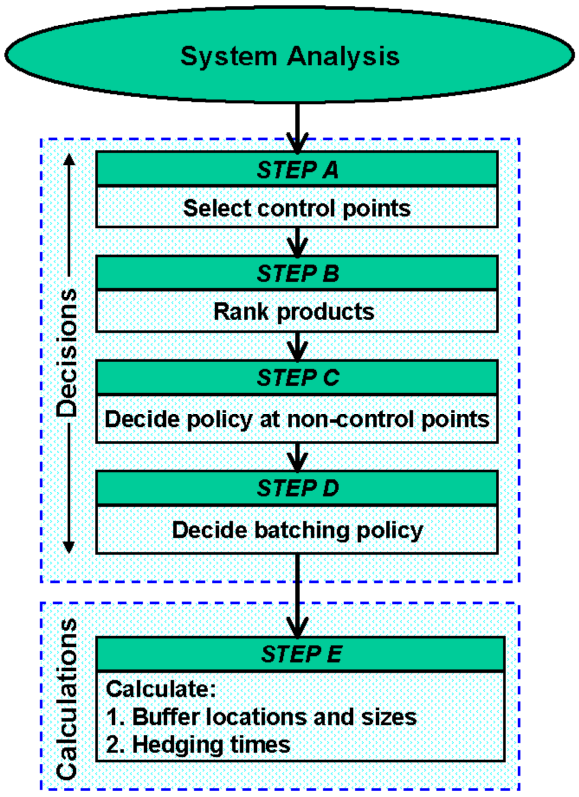


Figure 4-1: Steps for prior preparation activities

4.2 Prior preparation activities

Step 0 of the control point policy involves decisions and calculations based on these decisions. It is important to understand the factory system *as a whole* during the prior preparation activities. Figure 4-1 provides a flow chart for the prior preparation activities. The *analysis* of the system examines the flow of the material throughout the entire system from the supplier to the customer, the *decisions* made, based on this analysis, then designs the control system. The *calculations*, based on these decisions, then run the control system. If the analysis is not made correctly the decisions will not be correct. If the decisions are not correct, the calculations will be futile.

The analysis of the system precedes the prior preparation activities. At this stage, not thinking of the system as a whole leads to sub-optimal configuration and decision-making process. The analysis process combines the processes and the products in the factory, two domains which have been usually treated separately. All the inputs, processes and resources available to the factory need to be viewed as linked to each other. Based on this, we are able to identify the *vital links* in the chain, where we need to exert control so that the system as whole performs well. These are then the *control points* for the factory. Analysis of the products, their commercial value, and their flows through the system leads to the *ranking of products*. The last step of the process, *the calculations*, ensure that the decisions made in the previous steps are followed by the factory in real-time.

Accordingly, there are five steps *Step A to Step E*, in the prior preparation activities of the CPP, which follow the analysis of the system. The first four steps, *Step A to Step D*, are *decisions* which need to be made by the company management. These decisions are specific to the system, and are a function of the complexity and nature of the system. The discussions below serve as a tool for guiding the decision-making process. The first two: *Step A and Step B* are critical; the next two: *Step C and Step D* are less important. The last step, *Step E,* involves the *calculations* performed thereafter, based on these decisions. Currently these calculations are rough estimates. In due course of time, they will be the results of accurate mathematical solutions.

In the following sections, we will discuss each of these steps in great detail. Issues and guidelines are provided at each step. These steps are highly interdependent and have to be balanced carefully; or it will lead to an unsuccessful implementation.

4.2.1 *Step A*: Selection of control points

In Section 3.3, we had assumed that each and every work-center in a factory is a control point. This may be necessary in many manufacturing systems. However, in some manufacturing systems, as we realized during our Boeing experiment, it may not be necessary to exert control at each and every work-center. So the first step of the prior preparation activities is to determine the control points in the factory.

Also, a control point is any resource in the factory where we use the policy to schedule the work flow; it does not have to be a physical machine or a work-center. It can be a department or a decision making mechanism, called *pseudo control point*. For example, the in-coming raw material section in a factory, where we decide to introduce parts into the manufacturing system, can use the CPP to decide the order and time of launching parts. *Thus a control point in a factory can be looked upon as any point in the factory, where there is a need to regulate or re-sequence the flow of parts.* If it is already present as work-center or a department, then the additional work needed is minimal. If it is a pseudo control point specifically created for the CPP, then additional work is required. The additional work involves creating the pseudo control point in the factory (it may not be physical, but must be recognized by the CPP) and developing a mechanism to regulate and hold parts at the pseudo control point. In most cases, existing work-centers or departments can be used as control points.

It is also important to note the difference between a non-batching control point and a batching control point. Though the real-time calculations are made before loading the work-center in both cases, the *red parts* (Section 3.3) are held in the upstream buffer in case of a non-batching work-center, while the *red parts* are held in the downstream buffer for a batching control point.

There are two decisions which need to be made while selecting the control points in a factory:

1. Location of control points

2. Number of control points

Both the parameters are a function of the size, complexity and variability associated with the factory and the products. There are no rigid requirements for deciding the control points in the factory, there are certain guidelines which can help the process of deciding the control points.

Location of control points

We list the guidelines for deciding where the control points should be located in the factory:

a. Disorderly mingling of parts may reach a high level.

b. All or most of the part-types pass through the control point (once or more than once).

c. The point needs to be constantly monitored for the effective performance of other work-centers.

d. It should not be very late into the process flow, so that the bull-whip effect[2] can be reduced at the earliest.

Based on the above guidelines, the following points in a factory can generally be used as control points. It can be easily observed that all of them satisfy at least one of the above guidelines.

1. **Point of convergence**

A point of convergence in a factory is a work-center which is a normally a general-purpose resource through which all the parts in the factory flow at some point of time during their processing. This might be an assembly process, where different parts come in from different areas in the factory and get assembled or a process like heat-treatment or painting, which all the parts do visit at some point in their flow. The control point is then able to re-sequence the flow for *all the parts.* Also if the parts flow to different work-centers downstream after the control point, all the downstream work-centers are able to receive the parts in the correct sequence.

2. **Re-entrant work-center**

If a work-center is visited by many parts more than once during their flow through the factory, it is a good place to locate the control point. Different part-types q at different stages of production s tend to get mixed in the upstream buffer of such a work-center, causing them to lose the correct sequence. Putting them back into the correct sequence, becomes an important task. The CPP can do the re-sequencing.

3. **Point of release/re-entrance**

If parts leave the factory for some external processing, then that point is a good place to locate a control point. Most companies tend to closely monitor the flow of parts *to and from* this external process. Depending on how reliable the processing outside the factory is, we can

[2]The *bull-whip effect* or the *variability amplification effect* [Tabe et al., 1980] is the phenomenon where the variability seems to get amplified as one looks from further away from the apparent source of the variability.

decide to have the control point before the parts leave the factory or at the point where they arrive back into the factory. If the outside resource is a very reliable one which maintains the sequence in which parts are released into it, then it is important to release parts to it in the right sequence. In that case, it is good to locate the control point at the *point of release.* If the outside source is unreliable or has its own sequencing rules, then parts tend to lose the sequence in which they are released into it. In that case, it is better to locate the control point at the *point of re-entrance.*

4. **Shared work-center**

If the parts belonging to your business unit share a work-center in the factory with parts from other business unit/units, it is often a great source for disorderly mingling of parts. Parts belonging to your business unit tend to be grouped together for processing at the work-center, making the flow coming out of the work-center very lumpy and sporadic. In that case, it is good to locate a control point *at the exit* of such a work-center.

5. **Constraining work-center**

Another good candidate for a control point is at the constraining work-center in a factory [Goldratt and Fox, 1984]. In most factories, it is difficult to identify the constraint because with constant introduction of new products and rapidly changing demands, bottlenecks tend to float around in the system. Also, in most cases suitable measures have already been taken to reduce the effects of the constraint (capacity-increase, out-sourcing, extra-shifts etc.).

However, if there is a constraint in the system, it is always important to use it in the best way to ensure that its limitation on the production system is reduced as much as possible. It is important to ensure that the constraint works on the right part at the right time. Therefore, locating the control point at a constraint is a good choice. Also, since the queue in front of the constraint normally tends to be very long, it is a good place to re-sequence.

6. **Batching work-center**

A work-center, which batches multiple part-types or a number of the same part-type together, is another good place to locate a control point (using the policy for batching work-centers). The flow coming out of the work-center tends to be very lumpy. The parts belonging to a batch tend to flow together throughout the factory after that, even if all of them do not need to. Therefore, it is good to break the batch here and to release only the *green parts* (Section 3.3) in a batch downstream.

7. **Work-center with long processing times**

Work-centers, which have a long processing times (e.g. some chemical treatment processes) are a good choice for a control point. There is a long time difference between the loading of two parts (or batches), and that makes it necessary to load them in the right sequence. This ensures that the possible negative effect of this work-center on the rest of the operations is reduced.

8. **Point of issue of material**

In many cases, the arrival process of material into the factory itself is very disorderly. An example this is a factory which carries out the final assembly of parts from large number of sources or a factory with unreliable vendors. This point could either be at the start of the process where raw material is issued or at an intermediate point, where accessories or sub-parts are issued in the flow. In that case, the material-issuing department can be the control point to launch parts into the system in the right sequence.

Number of control points

The number of control points is a function of the amount of control the company wants to exercise on the factory. Some systems may need to have each work-center in the factory as a control point, when the variability and the effect of disorderly mingling of parts at each stage is high. But it may not be necessary or economically justifiable to exercise so much control on the system. It also can make the re-sequencing process redundant at many points if there is not much variability in the system.

The guidelines for choosing the number of control points in the factory are more loosely defined. There are only two guidelines:

a. The first control point should not be very late into the process flow, so that the bull-whip effect can be reduced at the earliest.

b. The number of control points should be such that there is *at least one* potential source of disorderly mingling of parts between each pair of control points. Otherwise, one of the pair is redundant.

In due course of time, calculating the optimal location and number of control points for a system will become a part of the analytical calculation process. Then, this decision will become a part of the control problem. Until such time, the above guidelines can serve as good rules of thumb to select the control points. It is important that the selection of control points be kept flexible: so that they can be changed based on performance of the system.

4.2.2 *Step B*: Ranking of products

Having decided the control points in the factory, the next step is to rank the different products at the control points. The ranking of the products reflects the value of the different products for the company. All other things being equal, the highest ranked product will always get the priority at the control point.

Depending on the manufacturing environment and the products, the difference in value between the products can be very large. Many times the same product has different versions, which are valued (priced) differently. Also the cost attached to a product can change as it flows through the system.

It may seem natural to have uniform product ranking all the control points in the factory, but that may not be necessary. The ranking system at different control points for different products can have variations.

Ranking criteria

It makes sense to link the product ranking to the direct economic value of the products for the company, but that may not be necessary. Other criteria can be equally important for the factory, and the savings associated with the criteria can result in greater overall benefits for the company. The following criteria can be used to rank the products at the control points:

1. **Premium margins**

Products are ranked according to the profit margins associated with them. The products with a high premium are given top rankings and they get top priority in the system. This will tend to improve the profits for the factory.

2. **Inventory holding costs**

   Products can be ranked according to the cost of manufacturing for the product. Since the cost of manufacturing is normally used to decide the inventory holding costs for the product, this tends to reduce the overall inventory costs for the factory by speeding the high cost products through the system.

3. **New products**

   When a company introduces a new product into the market, the company would like that product to receive top priority in the factory, because the implications of reaching the market late are large.

4. **Size of the parts**

   The space occupied by the parts belonging to a product are issues of major concern for many factories and the company would like to speed the large parts through the system. The space occupied by the product can then be used as the ranking criteria at the control point.

5. **Scrap rates**

   In many processes, the yield affects the production rate for the work-center. The ranking scheme should take this into account. The products with poor yield rates get a high ranking at the work-center.

6. **Preferred customers**

   Many companies have products for customers whom they value more than other customers and on-time delivery performance for these products is very important for the business.

7. **Seasonal parts**

   Certain products have very seasonal demand patterns, and it is very important to get good performance for these products during peak times.

8. **Parts with short lives**

   The life of the product (e.g. food processing and packaging industry) can be used to decide the ranking for the products. This will ensure that products with shorter lives get to the market faster than the others.

As seen above, there are several criteria which can be used at each control point, and the ranking procedure can get very complicated if we decide to use *more than one and different* ranking criteria at each control point. The ranking procedure can get especially very complicated, when there are many products with reentrant flows and assembly processes which join two or more products. The procedure for ranking should be kept as simple as possible: the simpler the better. In most manufacturing systems, a uniform ranking system for all products at all control points should be sufficient. A *single* ranking criteria should be used at a control point and if possible at all the control points. Classes of rankings, if necessary, can be created. In case of re-entrant systems, the same product q can also have the same ranking at different stages of production s.

4.2.3 *Step C*: Batching policy

The next step in the decision-making process is to decide the policy at the batching work-centers (if there are any in the system). As stated in Section 3.3, the policy algorithm is different at these work-centers. Though many companies have tried to establish single-part flows in the system, there are some processes which still require batching of multiple parts. Processes like heat-treatment and certain chemical processes are the most common examples.

As explained in Section 3.3, the first two steps of the CPP algorithm stay the same. The next step includes the making of logical batches (as per the current batching strategy used) out of all the parts available at the work-center. Once batches are made, the schedule of loading the batches on to the work-center needs to be decided. The policy of loading the batch with highest ranked *green part* first, followed by the batch with the second highest ranked *green part* and so on so forth is just one of the several possible policies. It is not difficult to imagine other policies. Following is a list of alternative loading policies:

1. Load the batch with the highest ranked *green part* first.

2. Load the batch with the maximum number of parts first.

3. Load the batch with the maximum number of *green parts* first.

4. Load the batch with the least number of *red parts* first.

5. Only load batches with number of *green parts* greater than a certain threshold. Do not load the other batches.

The important part of the algorithm is that after the batch is processed, only the *green parts* in the batch should be released further downstream. The *red parts* should not be released further downstream, until they become *green parts.* As long as this done, we claim that it does not make a big difference which policy out of the above five is used.

4.2.4 *Step D*: Policy at non-control points

The last step of the decision-making process is to decide the policy at the non-control points. If Steps A, B and C have been carried out successfully, nothing elaborate needs to be done at the non-control points. The control points will take care of controlling the flow of parts through the system and ensure that the non-control points get parts in the correct sequence. A simple policy like *first in first out (FIFO)* should be sufficient at the non-control points. It is important not to allow the non-control points to be unnecessarily idle.

4.2.5 *Step E*: Selection of hedging times and buffer sizes

The next step of the prior preparation activity is to select the hedging times and the buffer sizes for the parts at the control points. In the discussions of the Section 3.2, we had observed that for a two stage single part-type system, the calculation of the hedging point (and thus the hedging time) takes into account the size of the finite buffer between two stages. Any change in the buffer size would affect the calculation of the hedging time.

Calculation of buffer locations and sizes

Buffers serve the purpose of acting as decouplers between the manufacturing stages upstream and downstream of it. They also serve the purpose of limiting how far ahead of the demand the manufacturing system can get. [Gershwin, 1994] shows that the production rate of a simple tandem line with finite buffers present between the machines is much greater than a line with no buffers. Also, placing buffers of right sizes at the correct places in the line can maximize the production rate with minimum buffer space [Schor, 1995]. In spite of that many factories tend not to maintain buffers in their system.

[Schor, 1995] also shows that for a simple tandem line the buffer space should be *spread all over* the factory to maximize the production rate. The research has been restricted to simple tandem lines with single part-types and no control on the system apart from the buffers. It seems reasonable to assume that the same result should also apply for complex manufacturing lines with multiple part-types. However, in many cases it may not be possible to maintain finite and homogeneous buffers after each and every work-center in a factory. Physical space limitations along with the difficulty with monitoring and maintaining the buffers may prohibit such attempts. In such a case, the blocking logic as stated in the original statement of the policy (Section 3.3) should be modified.

In Step A of the prior preparation, we may select only a few work-centers in the factory to be control points. If only a few work-centers are selected as control points then the location and sizes of the buffers (for control points and non-control points) also must be modified. Different strategies (called *buffer strategies*) can be used to determine the location and sizes of buffers and they are discussed below.

The flow of information for the *blocking condition* of the CPP algorithm gets modified, depending on the buffer strategy used. In each of the following cases, it is indicated using a dashed arrow. It is important to remember that it is not the only flow of information in the system, the *earliness condition* ensures that there exists global information flow. The global information flow for the *earliness condition* remains unchanged under each modification.

1. **CONWIP loop between adjacent control points**

In this case, a maximum WIP limit between two control points in the factory for *all part sq's* or *each part-type q* is calculated. When the actual WIP in the loop equals the limit, a part is blocked at the upstream control point. Figure 4-2 shows the information flow for the *blocking condition* for this strategy. This type of blocking information flow limits how far ahead

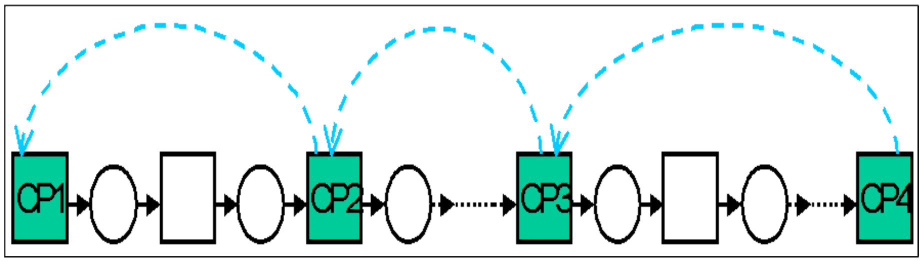


Figure 4-2: CONWIP loops between adjacent control points

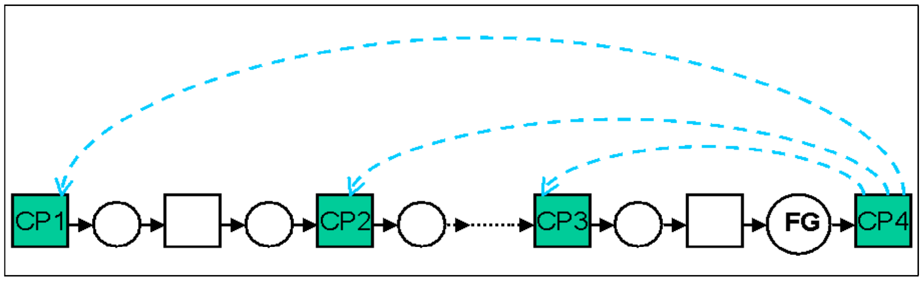


Figure 4-3: CONWIP loops between control points and shipping area

a control point is allowed to get of the next downstream control point [Spearman et al., 1990]. The information flow for the blocking condition then becomes *semi-local*, and not purely local as in the original formulation of the policy.

2. **CONWIP loop between control points and shipping area**

A maximum WIP limit is set for *all part sq's* or *each part-type q* between the each control point and the end of processing. A part will be blocked at the control point, if the total WIP downstream of the control point exceeds the limit. Figure 4-3 shows the flow of the information for the *blocking condition* in the system. It controls how far a control point is allowed to get ahead of the shipping process [Spearman et al., 1990]. The information flow for the *blocking condition* also becomes global with this type of modification. So there is *no local information flow* for this type of a system.

3. **Buffers between each work-center (control points and non-control points)**

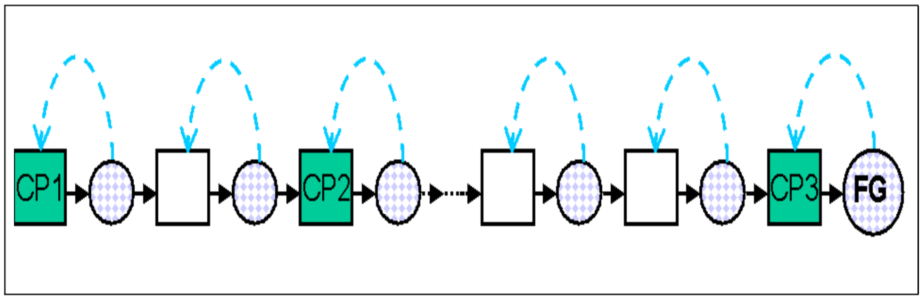


Figure 4-4: Finite and homogenous buffers between each stage of manufacturing

Another alternative is to have finite and homogenous buffers *at every stage* of manufacturing, whether it is a control point or not. At the control points, the flow of information for the blocking condition is *totally local* and becomes as shown in Figure 4-4. In addition, the non-control points also have local information flow because of the presence of buffers between them. It is similar to a simple KANBAN system. The flow of information controls how far a control point is allowed to get ahead of the next downstream work-center and also the next downstream control point because of the finite buffers in between them.

4. **Buffers after control points only**

We can decide to have finite and homogenous buffers *only after* the control points and decide to have no buffers at the non-control points (*simple KANBAN at control points only*). The flow of information for the blocking logic then becomes as shown in Figure 4-5. The flow of information for the blocking condition is *totally local*. It controls how far the control point is allowed to get ahead of the next downstream work-center.

A factory can use any of the buffer strategies which suits them the best. It is premature to say which of the above strategies is optimal. The advantages and drawbacks of each system should be similar to those discussed in Section 3.6. The optimal location and sizes of the buffers or the CONWIP limits, should be eventually calculable by analytical means. Once these analytical techniques are developed, we will be able to locate the buffers (or CONWIP limits) of appropriate sizes accurately. Until such time, reasonably calculated conjectures can be used.

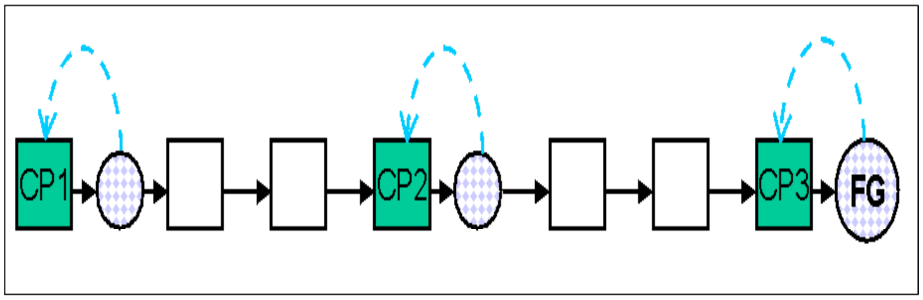


Figure 4-5: Finite and homogenous buffers between control points only

Calculation of hedging times

The next part of the process is to calculate the hedging times at the control points. The calculation of the hedging time for a single stage, single part-type system is an optimal calculation based on the machine parameters, the penalty for backlog and the penalty for surplus. However, the hedging time calculations for multiple stage, multiple part-type systems becomes an optimization problem subject to the following *additional* constraints:

1. Size of the buffers
2. Demand lead time for the part-types (Section 6.3 for details)
3. Hedging times at other control points (Section 6.3 for details)
4. Effect of resource contention (Section 6.3 for details)
5. Set-up policy at different work-centers
6. Batching of multiple parts at work-centers

In due course of time, we will be able to calculate the optimal hedging times.

Guidelines for calculating buffer sizes and hedging times

Since, at present, accurate calculations can not be performed, we can use the following rules of thumb to decide the buffer sizes and hedging times in the system. These are *broad guidelines* which were found as a result of the Boeing experiment and the simulation studies[3].

1. The hedging time at the first control point should be less than or equal to the demand lead time for the product. A high demand lead time in a system increases the service rate for the product, but it also increases the WIP levels and lead times. Therefore, the first step is to set the correct demand lead times for each product.

2. A high hedging time accelerates the part at a control point, while a small hedging times makes the part wait before it is processed. The service rate (and lateness) for a product with high hedging times tends to be small. Therefore, the hedging times for high-value products should be higher than those for the lower ranked products.

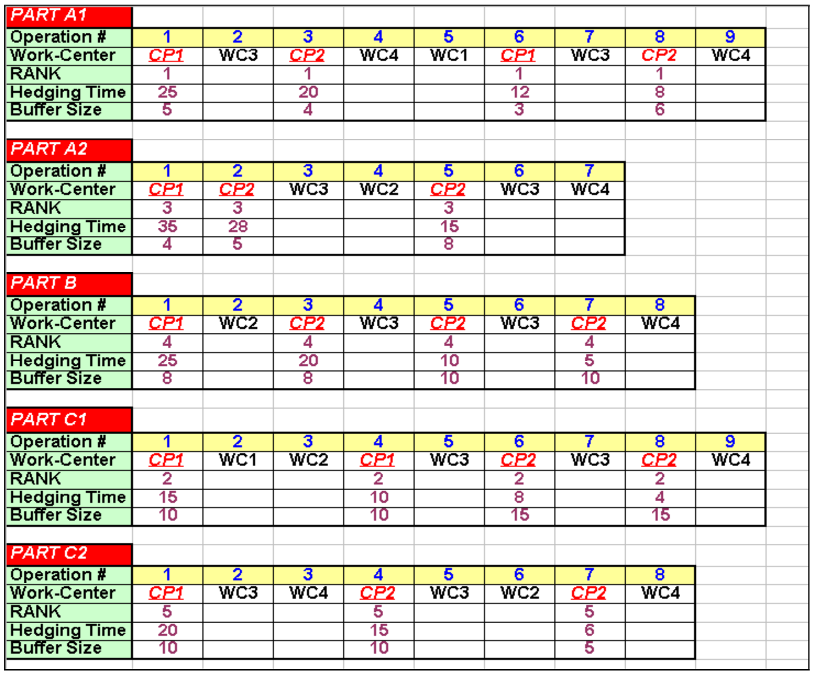
3. The greater the buffer sizes, the higher the inventory levels. Therefore, it is good to have small buffer sizes for the high-value products, and have large buffers for the lower ranked products.

4. If the hedging times at all the control points are high, the parts accumulate in the finished goods buffer and other downstream buffers. If the hedging times are low, the inventory stays in the upstream buffers. If proper buffer limits are not imposed, the inventory levels may tend to increase indefinitely in either case.

5. The top ranked products always get priority at the control points. Therefore, the capacity of the control points seen by them is close to the maximum. On the other hand, the capacity seen by the lower ranked products gets smaller as we go further down the ranking list. Having very high hedging times and large buffer sizes for the top ranked parts amplifies this effect.

These steps complete the prior preparation activities for the CPP. It is good to maintain all the *CPP input parameters* in the form of a database (*called CPP database*). Figure 4-6 shows the appearance of a sample CPP database. The different products are placed along the rows and the control points are placed along the columns. Then, for each product at each control point the rankings, hedging times, and the buffer sizes are indicated. If the ranking is uniform

[3]Details in Chapter 5 and Chapter 6 respectively.



PART A1									
Operation #	1	2	3	4	5	6	7	8	9
Work-Center	CP1	WC3	CP2	WC4	WC1	CP1	WC3	CP2	WC4
RANK	1		1			1		1	
Hedging Time	25		20			12		8	
Buffer Size	5		4			3		6	

PART A2							
Operation #	1	2	3	4	5	6	7
Work-Center	CP1	CP2	WC3	WC2	CP2	WC3	WC4
RANK	3	3			3		
Hedging Time	35	28			15		
Buffer Size	4	5			8		

PART B								
Operation #	1	2	3	4	5	6	7	8
Work-Center	CP1	WC2	CP2	WC3	CP2	WC3	CP2	WC4
RANK	4		4		4		4	
Hedging Time	25		20		10		5	
Buffer Size	8		8		10		10	

PART C1									
Operation #	1	2	3	4	5	6	7	8	9
Work-Center	CP1	WC1	WC2	CP1	WC3	CP2	WC3	CP2	WC4
RANK	2			2		2		2	
Hedging Time	15			10		8		4	
Buffer Size	10			10		15		15	

PART C2								
Operation #	1	2	3	4	5	6	7	8
Work-Center	CP1	WC3	WC4	CP2	WC3	WC2	CP2	WC4
RANK	5			5			5	
Hedging Time	20			15			6	
Buffer Size	10			10			5	

Figure 4-6: Information stored in the CPP database

at all control points, then it may not be necessary to include the ranking information at each control point. Another database can just maintain the list of products and their rankings.

4.2.6 Repeating the prior preparations

It is very important that the CPP input parameters be available for updates and changes. Since, at present, the buffer sizes and the hedging times stored in the database are conjectures they need to be fine-tuned as the system gets stabilized. The leverage metrics (Section 4.4) collect real-time statistics from the system and can be used to fine-tune the parameters to their optimal values. These are ongoing changes which can be carried out in an incremental manner as the factory gets more familiar with the system and its performance. It does not need a major change to the CPP database.

On the other hand, if the manufacturing system or the business undergoes a major change, then the prior preparation activities need to be performed again. The information stored in the CPP database needs to be completely overhauled. The changes in the system could be internal as well as external to the factory. These include:

1. Need for different or additional new control points.
2. Introduction of new products.
3. Changes in market behavior changes the demand or the value of products.
4. Changes in processes and flows.
5. Physical changes in the manufacturing system: additional machinery, different processes etc.

The CPP database can be reviewed on a monthly, quarterly or annual basis to assess the needs for any major changes. If there is a need for changing it, then Step A to Step E need to be repeated again.

4.3 Real-time CPP program

Once system has been designed correctly in the prior preparation phase, the real-time algorithm ensures that the system behaves according to the design. The real-time algorithm presented in Section 3.3 needs to be executed every time the loading schedule for a control point needs to be decided. First, the algorithm needs to find the current status of parts at the control point. The algorithm then compares the current status at the control point with that in the *CPP database*, by filtering each part through the three conditions. The parts which survive all the three conditions then form the loading schedule at the control point.

In this section we discuss the real-time CPP program in detail. We look at the algorithm from an implementation perspective followed by other implementation issues which are very important for the successful performance of the program.

4.3.1 CPP program and output

The algorithm is slightly modified from that discussed in Section 3.3 to make into a more implementable one. The only change is that the sorting of parts according to the ranking is done right at the start of the program in this case. There is no logical difference between the two algorithms, this algorithm is easier to implement and also the output is produced in a more presentable manner. An implementable version of the algorithm is as follows:

Step 1. Get a list of all parts present in the upstream buffer of the control point.

Step 2. Sort the parts according to product ranking. If there are no parts present, exit.

Step 3. Put the parts through the blocking and earliness conditions of the CPP. This is done for each part from the top to the bottom of the list.

Step 4. Attach a *green* or a *red* flag against the part depending on their status.

a. If it is a non-batching control point, load the highest-ranked *green part* in the list.

b. If it is a batching control point, make logical batches out of the parts. Using the *batching logic* selected, load the first batch.

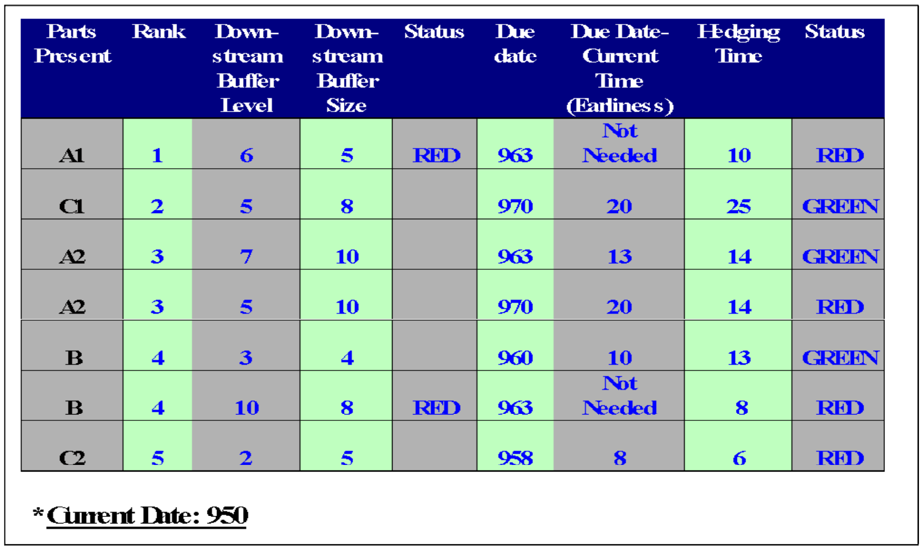
The output of the program at a control point would look like Figure 4-7. Referring to the table, part A1 should *not be loaded* (even though it is the top ranked part-type) because it has failed the blocking condition.. Once it has failed the blocking condition, it becomes a *red part* and the earliness condition does not need to be checked. Part C3 is the *first part* which should be loaded on to the control point, because it is the highest ranked *green part.* Its buffer level is less than the maximum buffer size allocated, and also its earliness (20) is less than the hedging time (25). The part A2 should be loaded next on to the control point because, it has passed all the conditions. Part C2 should be loaded next, because it is the next highest ranking *green part.* Parts B2, B1 or C1 are not loaded, they are *red parts.*

4.3.2 Real-time issues

The CPP program is not completely described by its real-time algorithm and the output of the algorithm. The manner and the frequency in which the schedules are produced are also important factors for its performance. There are three issues which should be considered:

a. What should be the interaction between the CPP system and the MRP system[4]?

[4]We assume that some version of a MRP system is used in the factory.



Parts Present	Rank	Down-stream Buffer Level	Down-stream Buffer Size	Status	Due date	Due Date-Current Time (Earliness)	Hedging Time	Status
A1	1	6	5	RED	963	Not Needed	10	RED
C1	2	5	8		970	20	25	GREEN
A2	3	7	10		963	13	14	GREEN
A2	3	5	10		970	20	14	RED
B	4	3	4		960	10	13	GREEN
B	4	10	8	RED	963	Not Needed	8	RED
C2	5	2	5		958	8	6	RED

Current Date: 950

Figure 4-7: Sample CPP program output

b. How frequently do we use the CPP program to decide the load schedules?

c. Who needs to run the CPP program?

The first issue is concerned with the incorporation of the CPP system (CPP database and CPP program) into the present system in the factory. The next two issues are related to the proper and disciplined use of the CPP program. We discuss all three issues in the following paragraphs.

Interaction with MRP system

One of the major advantages of the CPP program is that it is very simple, has very small requirements, and can be introduced into the factory without causing any major changes to the existing set-up. The CPP calculations, once the list of parts present at a control point is available, are very simple and do not take much computation resources. The major portion of the computation time is spent in making the *real-time information* available to the CPP program. This involves database queries which are normally very time consuming. Therefore, the total real-time computation time is a function of the data-transaction procedure between the CPP system and the MRP system.

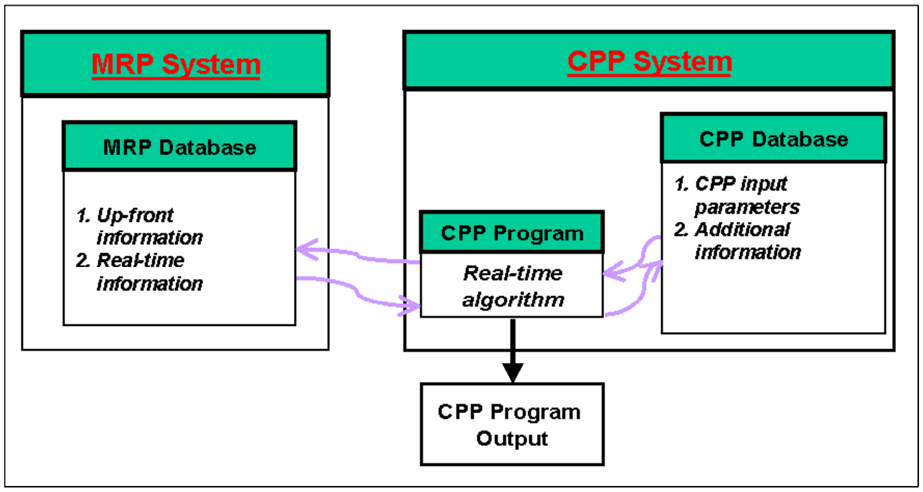


Figure 4-8: CPP system in subscription mode

There can be three levels of interaction between the CPP system and the MRP system, leading to three kinds of architectures. They are discussed in the following paragraphs along with the benefits and the disadvantages associated with each architecture.

1. **Subscription mode**

In subscription mode, the CPP system essentially lives off an existing MRP system, only extracting information from it whenever required. The only information the CPP database maintains separately is the *CPP input parameters* and the *additional information*. The up-front information and real-time information are maintained in the MRP database. Figure 4-8 shows the organization of this type of system. When the CPP real-time program is executed, it queries the MRP database to get the real-time information. After this data extraction, there is no interaction between the CPP program and the MRP system.

The advantages of this architecture is that it is very easy to develop. The MRP system does not need to be changed; only the CPP program needs to developed. The disadvantages are that its accuracy depends on the accuracy of the MRP database. Also, if the MRP system is large, the real-time computation time becomes large.

2. **Totally independent**

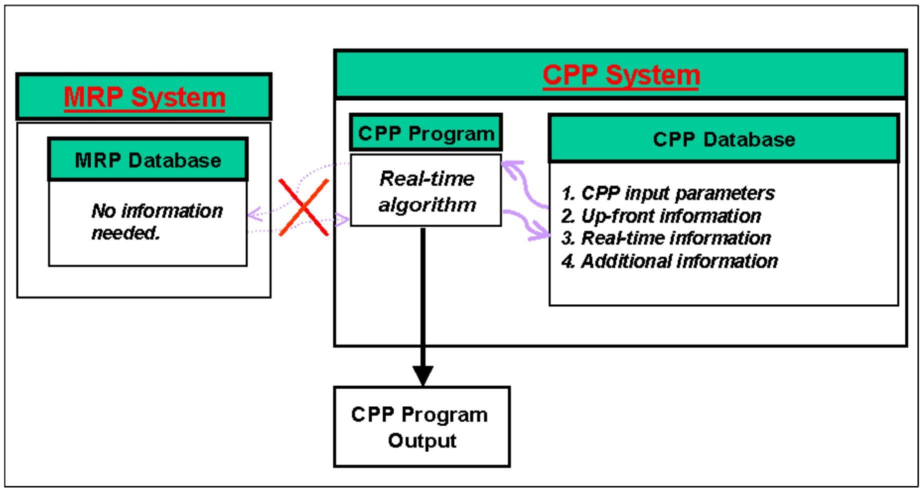


Figure 4-9: Totally independent architecture

In this type of an architecture, the CPP completely maintains and uses its own database. This type of an architecture may be used if there is no MRP database available, or if available, it is it is not practical for it to provide the information required for the CPP. In such a case, it is necessary to construct a CPP database which is self-contained; it contains all the information required for the CPP program. The CPP real-time program queries the CPP database when required. Figure 4-9 shows the organization of this type of a system.

The advantage of this system is that it is totally independent and does not rely on the MRP system for information. Also, the computation time is shorter. The disadvantage is that the database needs to maintain and track all the information for the factory, which normally the MRP database maintains. This costs a large time and effort to establish.

3. **MRP embedded**

In this type of an architecture, the CPP system is totally contained in an existing MRP system. There is no need to create an external CPP system. The CPP database can be extended to the MRP databases. Also, the CPP program can be used in place of an internal MRP dispatching algorithm. Figure 4-10 shows the interaction of the CPP system with the MRP system for this architecture[5].

[5]Thanks to John Shields of Boeing, Portland, for informing that this architecture was possible and could be

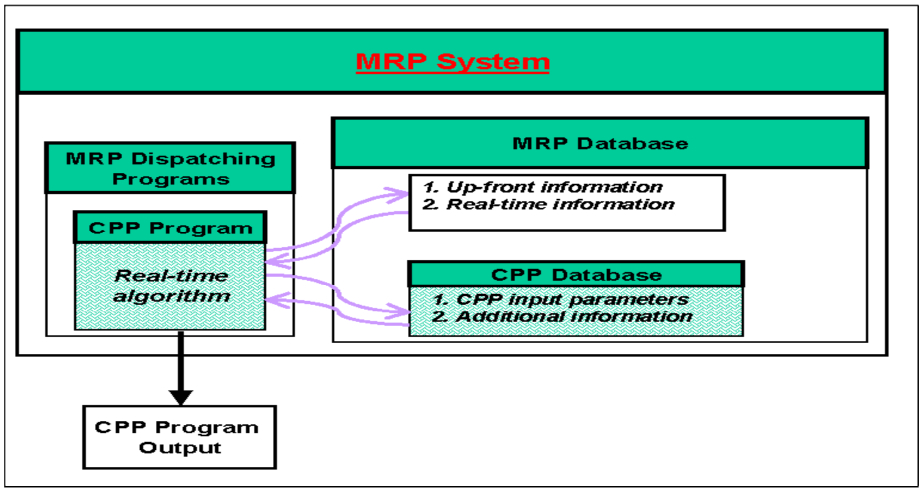


Figure 4-10: MRP embedded architecture

The advantage of the architecture is that it there is no need to create a separate CPP system in addition to the MRP system. Most of the MRP systems of today are flexible enough for the user to extend them as required. So the system is easy and quick to develop. However, the real-time program can become time-consuming in this case. Also the reliability of the CPP system is dependent on that of the MRP system.

The company can decide to use any one of the architectures, depending on which suits them the best. The subscription mode was implemented at Boeing and took a short time to implement.

Frequency of running the CPP program

Deciding the frequency of executing the CPP algorithm is another major issue of concern. It is a good discipline to run the program at a regular events, rather than running it whenever convenient. The program can be run after a fixed time interval or the status of the upstream queue can be used as an indicator to run the CPP program. The following is a list of events which can be used to decide the frequency of running the program at a control point:

easily developed.

1. Run it after every fixed time interval (e.g. every hour, every shift, every day, every half a week).

2. Run it every time the control point is available.

3. Run it every time the queue in front of the control point reaches a certain size.

4. If parts arrive in batches, run the program every time a fresh batch of parts arrives.

5. Run it every time all (or certain fixed number) of the *green parts* on the current list have been worked on.

Accessibility

This is another issue to related to the proper use of the CPP program and the discipline in the factory. The issue of accessibility to the program can also decide how frequently the CPP output is followed and the frequency of running the program. There are two methods of making the CPP program output available to the shop-floor supervisors and the operators.

1. **Periodic dispatching**

The CPP loading schedules can be periodically produced by the scheduling engineers and handed out to the shop-floor personnel. This is the easiest approach, because it does not require the end-user to have access to a computer terminal. Also, they do not need to bother about the frequency and discipline of running the CPP. *This approach is suitable when the CPP program is run on a fixed time interval basis.*

2. **On-line version**

The other method is to have an on-line version of the CPP program available to the shop-floor personnel. In this case, the loading schedules are produced on-demand by the shop-floor personnel. The end-user needs access to a computer terminal for running the programs. Also the programs should not take a very long time to run. This approach filters the decision making process to lowest possible level, making it very transparent to the end-users. *If the state of the control point queue is being used to decide the frequency of running the program, this approach suits better.*

4.4 System metrics

Once the CPP has been established and is used to schedule the work at the control points, the factory should start responding to the system. The metrics which are affected are very straightforward, because the policy goal is also very simple: *Produce parts on time with minimum inventory costs.* Also, some system metrics need to be tracked to assist the fine-tuning of the input parameters. Accordingly, there are two types of metrics which need to be measured for a system under CPP control:

1. **Performance metrics**

The *performance metrics* demonstrate the effectiveness of the CPP as a scheduling policy. Some of these improvements can be measured while some are intangible and cannot be quantified. The performance metrics should improve as the CPP input parameters are fine-tuned and optimized.

2. **Leverage metrics**

The *leverage metrics* (addition information) are the *system statistics or indicators* under CPP control, which need to be monitored in order to tune up the system as we proceed. They are used to improve the performance metrics and to help in solving the implementation issues discussed in Section 4.3.

4.4.1 Performance metrics

1. **Service rate**

The service rate for a product is the frequency with which the parts (belonging to that product) are produced on or before their due date. Again, there are two ways of measuring this metric:

a. Increase in the service rates for different products with the same operating costs.

b. Decrease in the operating costs for the same service rates for the products.

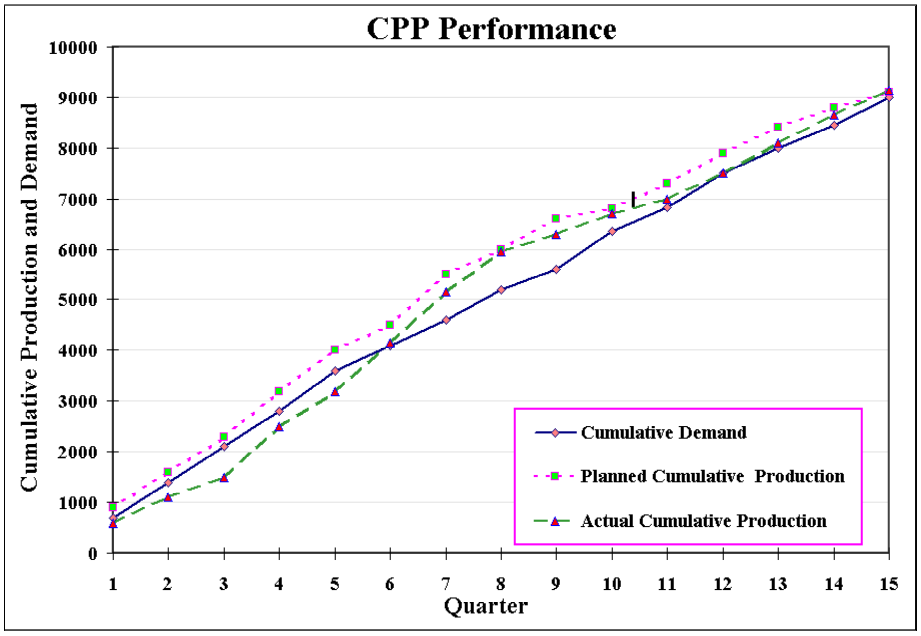


Figure 4-11: CPP performance chart for a product

In either case, the CPP is delivering results. The service rates for all products should improve with the CPP. However, if the capacity utilization is high, the CPP ensures the best service rates for the high-value products by accelerating them through the system.

A good way of tracking the above metric is by adding the actual cumulative production to the hedging point graph. Figure 4-11 shows the graph for a product. The full lines show the cumulative demand for each quarter. The dotted lines show the planned cumulative production. The dashed lines show the actual cumulative production, which is the response of the system.

Optimally, the actual cumulative production should coincide with the cumulative demand. However, due to non-availability of resources and other disruptions they can not do so at all times. The CPP tries to match the two curves closer, by working along planned cumulative production curve. Therefore, how closely the cumulative demand and the actual cumulative production curves match under the CPP will be a measure of the effectiveness of the CPP. This analysis can be done for each product.

2. **Operating costs**

Operating costs include the inventory costs and the labor costs. The CPP should be able to improve both the metrics for the factory.

a. **Inventory costs**

The inventory costs of the system should decrease due to the CPP. The CPP should be able to achieve these because of the following reasons:

i. The *ranking of products* gives priority to the high-value products. They tend to flow through the system faster than the others and their inventory levels and lead times in the system decrease.

ii. The *earliness condition* prevents the control points from working on parts that are early while others are late, and thus limits the excess inventory from being let into the system.

iii. The *blocking condition* prevents excess inventory from being pushed into the floor.

b. **Labor and management costs**

Another area where the CPP helps is in the reduction of labor costs and the need for manual intervention in the scheduling.

i. The reduction in labor costs should include overtime costs and the cost of using expediters to schedule the flow of work through the system. By ensuring that the operators are working on the right parts at the right times, their efforts and time are used efficiently.

ii. CPP reduces the need for manual scheduling and monitoring of flow of parts through the factory. It automatically determines good scheduling decisions for the factory and ensures a smooth flow of parts, without the need for the personnel spending their time on it. Thus, it reduces the load on the managers and the supervisors.

iii. When a production crisis[6] occurs, the CPP shows a clear and definite path out of it. This alleviates management problems, because using the CPP as a tool they can plan the easiest way back to regular flow in the factory.

[6]A production crisis means a large disruption in the flow of parts (e.g. a big failure at work-center, non-arrival of parts for a long time, factory shut down for maintenance, etc.).

iv. Another feature is the ease with which the management can control the system by changing the product rankings, hedging times and the buffer sizes. For example, during a business down-turn, the buffer sizes and the hedging times can be reduced. Or when there is a scheduled maintenance coming up, the factory can be operated with higher buffer levels to offset the downtime. This gives the management the flexibility to plan for forecasted business changes or scheduled activities.

3. **Production rate**

The production rate for a product is the rate of producing complete parts (belonging to that product) into the finished goods buffer. There are two ways of measuring the metric:

a. Increase in production rates with the same inventory costs.

b. Decrease in operating costs for the same production rates.

In either case, the CPP has resulted in an increase in the efficiency of the system. The objective of the CPP is not to maximize the production rates for the products but to keep them as closely tied to their demand rates as possible. The performance should improve for all the products. In certain cases, where the utilization is very high, it may improve only for the high-value products and may leave the performance for the lower ranked products as it is.

4. **Utilization of the resources**

The capacity utilization of the work-centers (control points and non control points) in the factory should not change drastically because of the CPP.

4.4.2 Leverage metrics

The additional information constitutes the leverage metrics for the CPP. Since the CPP input parameter are estimates, the leverage metrics help fine-tune and optimize these parameters.

1. **Distribution of inventory**

A metric which indicates the satisfactory performance of the system is the distribution of the inventory in the factory. The CPP divides the factory into local control loops, which are able to dampen the effect of disorderly mingling of parts and re-sequence the parts at each control point. This should result in certain behavior patterns in the inventory, if the parameters are selected correctly. There are two types of distributions we can look at:

a. **Spatial Distribution of inventory**

If, on the average, the inventory for all the parts is *evenly distributed* in the buffers (between two control points) it indicates that the CPP is working for the two control points. On the other hand if the inventory regularly gets piled up in the loop between two control points, it means the CPP is not performing as designed. Either the upstream control point is pushing parts in too early or the downstream control points is waiting too long before processing the parts or the buffer sizes are not correct. In that case, the hedging times and buffer sizes need to be adjusted for the two control points.

b. **Product-wise distribution of inventory**

The CPP tries to accelerate the high-value products in the system. Therefore, there should be very small inventory levels for the high-value products in the system. The inventory for these high-value products, if any, should accumulate in the finished goods buffer. On the other hand, inventory levels should be higher for the low ranked products because they tend to spend a longer time in the system. The inventory for the low ranked products should be distributed more or less evenly throughout the buffers in the system.

This information can be tracked by collecting the frequency of failing the blocking and earliness conditions of the CPP for a part sq at a control point (Section 3.5.2). If the frequency is too high, the part tends to get accumulated in front of the control point and spends a long time at the control point. If a part sq *consistently* fails the blocking condition too often, then the either the buffer sizes downstream need to be adjusted or the hedging times of the next downstream control point needs to be increased. If the part *consistently* fails the earliness condition, then the part is reaching the control point too early, and the hedging times upstream of the control points should be decreased or the hedging time at that control point should be increased.

2. **Frequency of following the CPP output**

This is another measure of the success of the CPP in scheduling work for the shop-floor. *It measures how frequently the schedule produced by the CPP at a control point is the one which is followed by the shop-floor.* If the schedule produced by the CPP does not make logical sense to the shop-floor personnel, they are going to neglect it. In most cases[7], the schedule provided by the CPP should be the one to be followed. If the frequency is poor, then the CPP input parameters need to be changed.

3. **Batching policy performance**

We believe that as long as the *red parts* are not released downstream into the system after the processing of batch and the work-center is not idle or under-filled unnecessarily, the exact *batching policy* at the control point is not very important. The percentage of green-to-red parts being processed at a batching control point should be monitored for that. If the percentage is high, the batching policy is working well. However, if the percentage of *red parts* is large and the buffer downstream of the control point tends to be very high, the batching policy at the batching control point needs to be changed.

4.5 Conclusions

The chapter provides a *"How-to"* document for implementing the CPP in a factory. Using the document as a framework, the policy can be implemented at a factory site. Presently, the policy is in the process of being defined and relies on the user's judgement for deciding many of the parameters. When the research is complete, the parameters of the policy can be calculated and optimized.

In the next chapter, we describe Boeing experiment. We apply the framework of this chapter step-by-step, to see how the policy was successfully implemented at the Boeing factory.

[7]In certain extraordinary circumstances, the schedule provided by the CPP may need to be overridden, as it is not applicable. This decision must be left to the discretion of the manager. If possible, a system and record for overriding the CPP should be maintained.

Chapter 5

Experiment at Boeing

We experimented with the implementation of the control point policy at the Boeing Commercial Airplane Factory at Portland, Oregon. The experiment was conducted between March 1998 and July 1999, with research being done both at the factory-site and at MIT. The experiment was the first factory experience for the policy. It was a partial success: the policy was successfully implemented, and it gave satisfactory performance, as long as it was in use. Unfortunately, due to certain unrelated developments, use of the policy had to be aborted very early in the process.

This chapter deals with experiment at Boeing. The Boeing facility at Portland, Oregon hosts the 777 Flap Support Business Unit, where the CPP experiment was conducted. Section 5.1 describes the goals of the experiment. Section 5.2 provides a description of the factory and product in detail. Section 5.3 briefly describes the other lean initiatives in the factory. In Section 5.4, the existing scheduling policy and the prior projects for improving the scheduling are discussed. Section 5.5, Section 5.6 and Section 5.7 discuss the details of the CPP implementation: the first gives the background for the implementation, Section 5.6 deals with the prior preparation, while the last section deals with the real time program.

The policy was used briefly to schedule work at the control points. Section 5.8 reports on the use of the policy, while Section 5.9 gives the details on the abrupt termination of the experiment. There are no quantitative results to present. However, results of the survey conducted among the management, which are presented in Section 5.10, show promising feedback for the policy. In the final section, Section 5.11, we conclude the chapter with the achievements and lessons from the experiment.

5.1 Goals

The objectives of project were the following:

1. **Feasibility of the idea**

To translate the CPP in to an implementable solution for a factory. The policy had a proven theory and made intuitive sense to all the people involved with the project. The policy was in the process of preliminary formulation during the course of the experiment, and we wanted to ensure whether it would work in a real factory situation.

2. **Impact of the policy**

To observe its effects on the performance metrics of the factory. Ascertain whether it is a good policy and results in lean improvements for the factory.

3. **Policy parameters**

To derive insights in to the process of optimization for the parameters. The parameters used for the experiment were not optimal parameters; but were initial conjectures made by the Boeing management in consultation with MIT. By observing the response of the factory to changes in the parameters, develop an understanding of the behavior of the parameters.

4. **Future implementations**

To provide a benchmark for future implementations. The lessons and observations gained during the course of the experiment will help future implementation at other factory sites.

5.2 Factory details

The Boeing 777 flap support is one of the business units located at the Boeing facility located in Portland, Oregon. The business unit produces flap supports for assembly on to the Boeing 777 range of commercial airplanes. The unit manufactures, processes and finally assembles the flap support mechanism. These are then shipped for the final assembly on to the airplanes at Everett, Washington. There are approximately150 individual part-types (known as part-numbers)[1], which are then assembled in to fourteen final assemblies (called ship-set) per airplane.

5.2.1 Product

The 777 flap support is a complex mechanism, primarily made up of aluminium. It transmits motion from the shaft-drive to the flaps for maneuvering purposes, during the process of take-off and landing. The flap supports are mounted symmetrically on the wings of the aircraft. There are four stations symmetrically located on each wing of the plane, which house the flap supports. The corresponding stations on each wing have the exact mirror-assemblies of the other wing, making a total of eight flap supports per plane. In addition to the flap supports, there are three rear spar attachments on each wing.

The rear spar attachment is an assembly, also, primarily made up of aluminium. It attaches the flap supports to the wing body at the different stations along the wing. Like the flap support assemblies, the rear spar attachments too have the wing symmetry. However, unlike the flap supports, the inner-most station on each wing does not have a rear spar attachment. The flap-support attaches itself to the wing-body at these stations; it does not need a rear spar assembly.

The eight flap support assemblies together with the six rear spar attachment assemblies constitute a *ship-set*. Each ship-set is earmarked for a particular 777 final product. The ship-sets are transported to the final 777 airplane assembly in Everett by trucks.

5.2.2 Operations

The 777 flap support business unit performs the entire processing for the above assemblies: from the raw material machining to the final assembly. The raw material consists of huge aluminium blocks, which are machined, cleaned, processed and assembled into the end-product before shipping to the customer. Parts need to be inspected after each major operation. The various operations are machine intensive as well as labor intensive. The different work-centers

[1]To be consistent with Boeing terminology, the term *part-number* is used in this chapter. The term *part-type* or *product* can replace it.

in the factory can be categorized into five broad types:

1. **Machining**
    a. CNC machining
    b. Boring
2. **Cleaning**
    a. Deburring
    b. Hand finish
3. **Processing**
    a. Plating
    b. Shot peen
    c. Painting
    d. Heat treatment
4. **Assembly**
5. **Inspection**

We provide a brief description of each operation:

1a. **CNC machining**

There are a set of 4 CNC machines, which carry out the cutting and milling processes for the different parts. These are enormous machines, which have to be programmed every time a new part is loaded on to the machine. They process material at the start and at intermediate steps of production for the part-numbers.

1b. **Boring**

The boring operations are done at different stages of production for the part-numbers. Most of the operations are done within the 777 business-unit, while some of them are out-sourced to other business-units in the same facility.

2a. **Deburring**

Immediately following the machining operations parts need to be deburred. This is done at the deburring work-center. The operations are manual as well mechanical. The operations are called *mass media deburring.*

2b. **Hand Finish**

Hand finish involves a set of operations to prepare the just-machined parts for further processing. The operations include cleaning, removing metal markings, edge preparation, shot peen preparations, chamfering and deburring. It also includes some visual inspection activity.

3a. **Plating**

Plating involves various kinds of chemical surface treatments to make the parts ready for painting and shot peening.

3b. **Shot Peen**

Shot peen[2] involves the bombarding of parts with high-speed peening media (steel shots, glass beads or ceramics) using nozzles or spinning wheels. The process surface-hardens the part by putting compressive stresses on the surface. The process also reduces the stress corrosion and improves the overall strength of the part. Shot peen includes a set of program-driven machines, which batch parts based on a common shot peening recipe. This work-center does the operation for parts belonging to other business units as well.

3c. **Painting**

Different parts are painted at intermediate stages as well at the final stage at the work-center.

[2] Please refer to (www.shotpeener.com) for further details.

3d. **Heat treatment**

Some of the part-numbers need to undergo heat treatment at intermediate steps. They are sent outside the business unit for this purpose.

4. **Assembly**

The assembly process involves mating and attaching of parts and sub-assemblies and the preparation activity needed for it. It involves intermediate assembly as well as final assembly. The assembly process is mostly manual. There has been a cellular flow established in the assembly area with areas devoted to certain part families and operations. On the completion of final assembly, parts are put into the finished goods buffer, for shipping to the customer.

5. **Inspection**

Inspection of parts is done after each major operation. It involves visual and physical tests to assure the quality and dimensions of the parts. Parts may be rejected at inspection. The same part-numbers tend to be batched together at inspection, making the flow coming out of the inspection stage very lumpy.

5.2.3 Process flows

All the work-centers are shared by the different part-numbers: they visit the work-centers at different stages of production. Each part-number has its own unique routing through and processing times at the different work-centers. The *general flow* for the parts through the system is as shown in Figure 5-1. The flow is highly re-entrant for most of the part-numbers; with many loop-backs as shown in the figure. The detailed flow for each part-number is very complicated. Parts visit work-centers more than once and have different processes to be performed each time they visit the work-center.

Parts are issued and first undergo processing at CNC machining. After that, they go to different deburring and hand finishing operations followed by inspection. After these operations, they proceed to different processing centers: plating, heat treatment, shot peen and painting. This is again followed by inspection. After they have been accepted by inspection, they proceed to assembly. Some part-numbers, in between, also have the boring operation performed. This

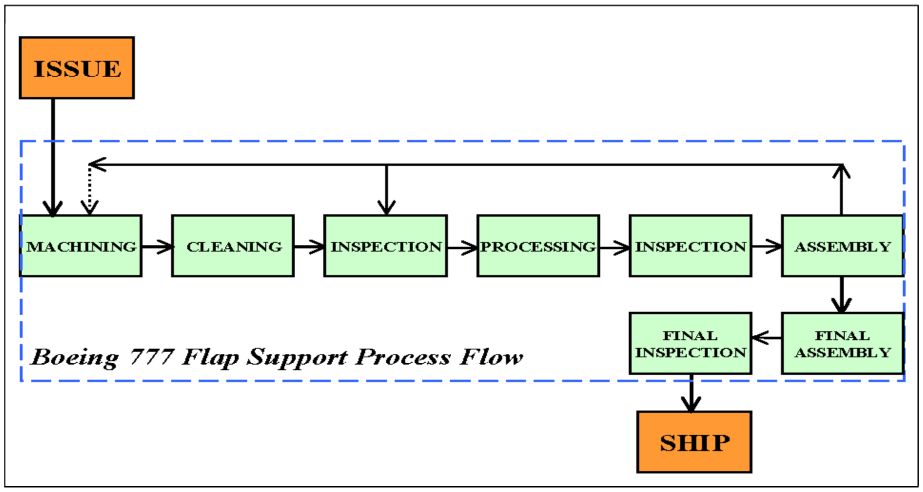


Figure 5-1: General process flow

cycle is repeated (not necessarily in the same order) three to six times for a part-number. Sub-assemblies, fixtures and other parts are introduced at intermediate stages. Parts, on final assembly, undergo final inspection before they are shipped out.

The total flow times (i.e. lead times) for the different part-numbers range between 15 and 25 days: from issue to final inspection. All the work-centers except shot peen and painting are completely devoted to 777 flap support parts. The heat treatment process and some of the boring operations require the parts to leave the business unit. Overall, the effect of *disorderly mingling of parts* in the system is very high. The reentrant nature of the flow made the process of tracking parts in the system difficult.

The only work-centers which batch more than one part are shot peen, boring and inspection. Shot peen batches different parts based on common recipes, in order to maximize the capacity utilization for each shot peening operation. On the other hand, boring and inspection tend to batch the same part-numbers together, because the change-over times are high. At other work-centers, each part-number has an individual set-up, which needs to be performed, irrespective of the sequence of loading part-numbers.

The rejection rate for inspection ranges between 5 to 20%, with the exact rejection rate being very part-number specific. Depending on the nature of the fault and the cost of rework, the rejected parts may be either scrapped or reworked. Once a part is rejected, there is a 20%

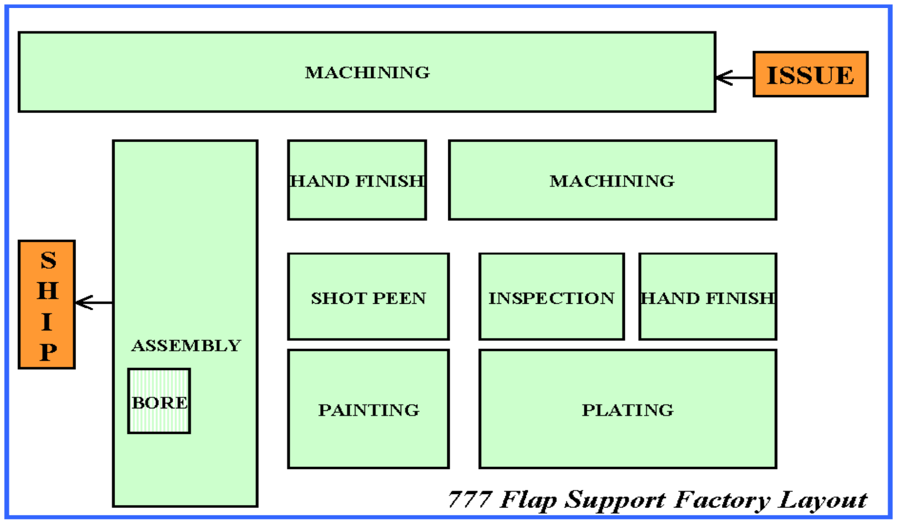


Figure 5-2: Layout of the factory

probability of being scrapped. All the inspection and rejection process is done before the final assembly; parts once shipped to the customer, are never called back for quality work.

5.2.4 Layout and travel

The layout of the factory is as shown in Figure 5-2. A work-center (also called cost-center) consists of machines or work-benches and resources (tools, fixtures, operators) necessary to carry out the particular operation. Parts are moved between work-centers in carriages by the moving personnel. They make regular trips to the different work-centers to pick up parts. Parts are normally kept in a certain demarcated area at each work-center to indicate finished items. Parts are *clocked-out* at each work-center; i.e. the information is entered into the MRP database to indicate the completion of the operation.

5.3 Other lean initiatives

A survey was conducted among the management to collect the status and the success for some of the major lean projects. The survey indicates a high level of success for most of the initiatives, and ongoing projects for further improvements.

1. **Cellular and grouping techniques**

   The factory was designed with cellular concepts: all similar operations have been arranged together into work-centers (Figure 5-2). There have been limited attempts towards *grouping part-numbers and processes*, if any, they have all been restricted to the assembly area. The enormous size of the machinery, along with the presence of general purpose work-centers shared by large number of parts-numbers, does not allow *grouping* for the entire factory. On the other hand, in assembly, part-number families have been created and grouped together in cells. The management survey indicates that the flow in the factory is probably as cellular as it can get.

2. **Set-up reduction**

   Set-up times and costs have been and continue to be reduced at all work-centers. Most of the set-ups are external set-ups. They have to be performed for all the parts irrespective of the loading sequence. The boring work-center has large set-ups, and there is a long-term plan to reduce them at this work-center.

3. **Single-part flow**

   All work-centers (except for shot peen and boring) have established a single-part flow in the system. At shot peen and boring, reasons explained in Section 5.2.2, prohibit attempt towards establishing a single-part flow.

4. **Lead time reduction**

   The lead times for the part-numbers have been continuously reduced over the last three years, with some lead times being reduced by almost 60%. An analysis of the present lead times showed that, even now, some parts spend 50% of their total flow time in non-value added activities. There is an ongoing project which is looking into further cutting down the lead times.

5.4 Scheduling

The factory uses an MRP II system to schedule the flow of work through the factory. The Master Production Schedule (MPS) is updated twice a week and the Bill of Materials (BOM) is produced every night. The BOM process takes about 45 minutes to run. Once the MPS is run, the schedulers decide which parts to launch into the system depending on the due dates and the reject situation.

Once the parts are introduced into the system, the Daily Dispatch List (DDL) decides the schedule of work for the shop-floor. The DDL uses the *critical ratio calculation* to decide the loading sequence at the work-centers. The total processing time for a part at a work-center was taken as the sum of the travel time, queue time, setup time and run time.

However, the DDL sequence was very rarely followed by the shop-floor, because the sequence indicated by the DDL is very often incorrect. The shop-floor supervisors used their discretion to schedule the flow of work. As a result, the scheduling inside the factory was very informal, with expediters being used regularly. Individual parts had to be monitored and scheduled manually. This also made the factory very sensitive to crisis: if there was a major production disruption (e.g. long machine failures, maintenance shut-downs), it took a lot of time and effort to get the flow back to normal.

5.4.1 Prior projects

The 777 flap support management had implemented several projects, which had yielded very good results for the factory. Most of the projects exploited the deterministic nature of the demand faced by the unit. In this section, we provide a brief description of the projects:

1. **Ordering process - trucking CONWIP loops with vendors**

Realizing the importance of the order process of the factory, the unit had established CONWIP loops of fixed size with the vendors to regulate the flow into the system. A fixed number of carriages for each part-number q (thus a CONWIP loop) rotated between the company and the vendors. A carriage was emptied at the Boeing end every $TAKT_q$ time. The empty carriage, when it reached the vendor, indicated a demand for part-number q. This ensured that the part-ordering process was self-regulated.

The unit also maintained trucking CONWIP loops with the customer (in Everett) to regulate the flow out of the system.

2. **Release process - visual control**

Part-number *q's* were released into the factory floor every $TAKT_q$ time. Since there are many parts-numbers in the system, monitoring the release process was not a simple task. However, using the Visual Control Chart, the management was able to monitor the status of the release process, and control it accordingly. The Visual Control Chart was a large board mounted in the middle of the factory which indicated the status and release dates for the various work-orders. This gave transparency to the shop-floor and also regulated the release process in to the factory.

3. **Internal control - KANBAN system**

The large number of part-numbers and the re-entrant nature of the flow made it very difficult to maintain *physical* finite buffers for all part-numbers at all points in their flow. However, the management had established finite buffers (i.e. KANBAN tokens) for the important part-numbers in the factory at key intermediate points. A fixed number of carts (color-coded for each part-number) rotated between two operations. An empty cart at the upstream operation indicated a demand for the part-number, while all carts full indicated blockage for the upstream operation.

4. **Waste reduction - reusable carriages and packaging**

As a part of their efforts to reduce the waste in the factory, the company had re-designed the carriages and packaging used while transporting material to and from the factory. This made it possible to reduce the costs of transportation and also to eliminate waste.

5.5 CPP implementation background

These projects ensured that the order and the release process for the factory was controlled. The unit was able to achieve significant WIP and due date performance improvements with the above projects. However, due to the long flow times in the factory and the large effect of *disorderly mingling of parts*, the parts were losing their sequence through the factory. The part flows, by the time they reached final assembly became highly skewed. This made the part of the assembly manager very difficult because he was not able to forecast and anticipate the arrival of part-numbers at final assembly. The assembly manager was quoted "I wish I could get parts in the same sequence as they were released into the system". Therefore, the management felt there was a strong need to re-sequence the flow at intermediate points in the flow to make the arrival process at the final assembly more predictable.

The management also felt that there was strong case for further reduction of inventory levels and to make the process of scheduling more disciplined and smoother.

The next three sections describe the details of the implementation of the CPP at Boeing. Each step of implementation is explained using the framework presented in Chapter 4. MIT's involvement with the project was as an advisor, with the Boeing management responsible for most of the decisions and implementation. Section 5.6 describes the prior preparation activities (Refer to Section 4.2 for details). In Section 5.7, we discuss the working and other issues regarding the CPP program (Refer to Section 4.3 for details). Section 5.8 gives a description of the use of the policy for the brief time that it was in use.

5.6 Prior preparation

5.6.1 Analysis

From an analysis of the system, the Boeing management came up with the following requirements for the policy:

1. To ensure a more predictable arrival process for the final assembly.

2. To reduce the inventory costs for the factory.

3. To reduce the need for manual intervention in the scheduling of the factory.

The prior preparation activities converted these objectives into parameters for the policy.

5.6.2 *Step A*: Selection of control points

We realized that the final assembly was too late in the process to control the variability: it had to be controlled at points prior to the final assembly. We also realized that control was not required at all work-centers; it was needed at key intermediate point in the flow. Since the part-numbers were many, it was necessary to locate control points at work-centers through which all the part-numbers flowed, at one point or the other (Section 4.2.1 for details).

Location of control points

1. **Shot Peen**

Shot peen was an obvious choice for a control point because it was:

a. *Point of convergence*: All the part-numbers flow through this work-center.

b. *Reentrant work-center*: All part-numbers visit the work-center more than once.

c. *Shared work-center*: The 777 flap support parts share the work-center with parts from other business units, making the flow coming out of the work-center very lumpy and sporadic.

d. *Batching work-center*: The parts batched at this work-center tend to flow together through the remaining processes in the factory.

Shot peen typically preceded the assembly process (final and intermediate), with the inspection process between them. Therefore, the assembly supervisor had to constantly monitor the flow of work at shot peen to foresee his arrival process. The management strongly believed that the shot peen operation set the tone for the rest of the factory. Therefore, it was an ideal control point.

2. **Assembly**

The second control point was assembly work-center because it was:

a. *Point of convergence*: Parts and sub-assemblies arrived from various sources in the factory and also from outside vendors.

b. *Reentrant work-center*: Parts visited the work-center at various intermediate stages and also for final assembly.

3. **Machining**

The machining process was chosen as the third control point because it was:

a. *Point of issue of material*: Most of the parts are introduced into the factory at the work-center.

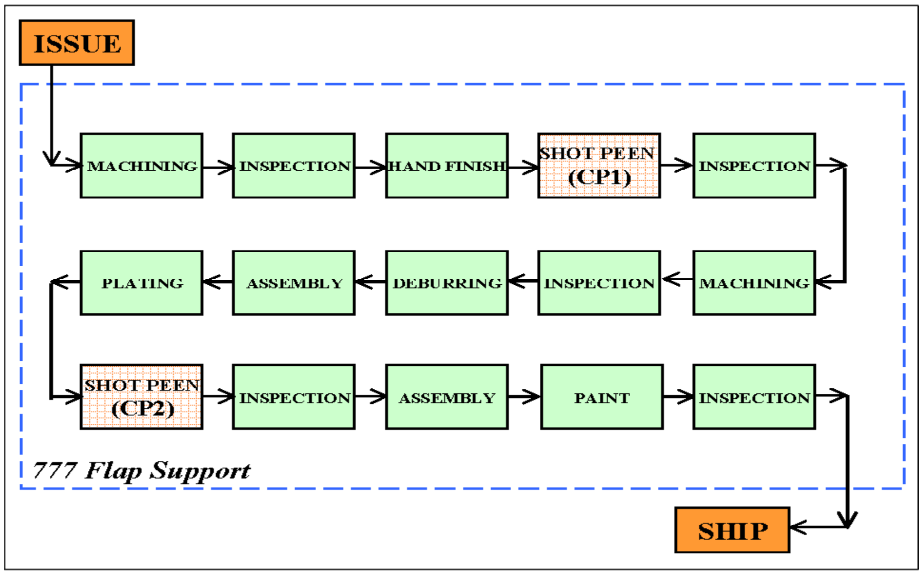


Figure 5-3: Final location of control points for a sample part-number

b. *Reentrant work-center*: Many parts visit the work-center for intermediate machining operations.

Number of control points

The management felt it was sufficient to exert control at these work-centers. The remaining work-centers could maintain the sequence of parts; the above work-centers were the major sources of randomness. Even amongst these three control points, shot peen was the most crucial one. Therefore, the plan was to start using only the shot peen control point and assess the performance. If needed, the assembly and machining work-centers could be added as the other two control points. Figure 5-3 shows the control points in the flow for a part-number.

The number and location of the control points were decided by the Boeing management. These were initial conjectures: the location and number could be changed later.

5.6.3 *Step B*: Ranking of part-numbers

Part-numbers were ranked based on the *cost of manufacturing* for the part-numbers (Section 4.2.2 for details), since the inventory costs for the factory were calculated as a percentage of the cost of manufacturing. The parts and the assemblies processed by the 777 flap support unit had high manufacturing costs attached to them. Also, the cost-distribution was large, with part costs ranging from hundreds of dollars to tens of thousand of dollars. The part-numbers were given a uniform ranking at all the control points.

This strategy tends to accelerate the high costs part-numbers through the system, which was desired by the company. The ranking scheme was again decided by the Boeing management. The yield rate was another criterion considered, but the cost of manufacturing was preferred over it.

5.6.4 *Step C*: Batching policy

Shot peen was a batching work-center. The management decided to use the *batching policy* (Section 4.2.3 for details) of loading the batch with highest ranked *green part* first, followed by the batch with the second highest ranked *green part* and so on so forth.

Shot peen batches part-numbers based on the batching color-codes attached to them. Part-numbers having the same color-code can be batched together. Therefore, in addition to the rankings, information on the batching color-codes was also stored against the part-numbers in the *CPP database*.

5.6.5 *Step D*: Policy at non-control points

The last step of the decision-making process was to decide the policy at the non-control points. The *first in-first out (FIFO)* policy was thought to be sufficient at the non-control points.

5.6.6 *Step E*: Calculation of hedging times and buffer sizes

Since we had not developed analytical techniques for calculating the buffer sizes and the hedging times, the Boeing management came up with a reasonable strategy to conjecture the hedging times and the buffer sizes. The strategy required the calculation of the hedging times prior to the buffer sizes. The strategy was as follows:

Calculation of hedging times

The flow times for the part-numbers maintained in the MRP database were not accurate. However, based on their observations, the management was aware of the total time a part spends at a work-center, once it enters it. Their estimates took into account the set-up and run times, and also the travel times and delays associated with queuing and waiting. Thus, they were able to conjecture conservative estimates for the flow times associated with each work-center. A list of work-centers with the flow time associated with them, was made. The hedging time at a control point was then simply the sum of the flow times between the control point and the final assembly. The hedging times (HT_{sq}) were thus calculated for each part-number q at each control point s.

Calculation of buffer sizes

The *buffer strategy* of having CONWIP loops between adjacent control points was fixed. The size of the CONWIP loops between two control points was calculated using the formula:

$$B_{sq} = \frac{HT_{sq} - HT_{s+1,q}}{TAKT_q} \tag{5.1}$$

In other words, the size of the CONWIP loop for a part-number q between two control points was simply the product of flow time between the control points and the demand rate (d_q) for the parts. The demand rate (d_q) for the part-number was the demand rate for the for the end-product, that the part-number was finally assembled into.

The CPP input parameters, along with the information on the batching color-codes was maintained in the *CPP database*.

5.7 CPP program

Boeing developed the CPP program with help and guidance from MIT. When run for a particular control point,the program goes through the following steps:

Step 1. Accesses the MRP II database and gets a list of all parts present in the buffer upstream of the control point.

Step 2. Looks up the CPP database for the cost of the part-numbers and sorts them according to it.

Step 3. Puts the sorted part numbers through the algorithm described in Section 4.3.1.

Step 4. Generates a list of *ready* (*green*) *parts* and *not ready* (*red*) *parts*. At the shot peen control point, the batching color-codes are also printed against the parts.

The program on execution generates three lists.

List #1. **List of ready parts**

This is a list of parts which have passed all the three conditions of the CPP algorithm and are *ready* to be loaded on to the work-center.

List #2. List of early parts

This is the first sub-list of *not ready parts*. These parts are early according to the hedging time and should not be worked on until they get promoted to the list of ready parts. (Exception: shot peen, where the policy for batching work-centers is used).

List #3. **List of blocked parts**

This is the second sub-list of *not ready part*-numbers which should not be worked on until they are promoted to the list of ready parts (Exception: shot peen, where the policy for batching work-centers is used). For all the parts in this list, the actual number of parts present between the control point and the next downstream control point for that part-number is greater than or equal than the CONWIP size allocated.

At the shot peen control point, the batch consisting of the highest ranked *ready part* should be loaded first. The batch with the next highest ranking *ready part* should be loaded next and so on so forth. After processing, only the *ready parts* in the batch would be released downstream. The *not ready parts* in the batch would be held back in the downstream buffer. At the remaining two control points, the *ready parts* should be loaded in the sequence they appear on the list.

The *subscription mode* (Section 4.3.2 for details) architecture was used for the interaction of the program with the MRP database. The program took approximately 180 man-hours for development. At real time, the program took a total run time of approximately 4 minutes. 65% of the time was spent in extracting data from the MRP database, while the remaining time was spent in putting the extracted information through Step2 - Step 4 of the program.

The program was configured in such a way that it could be run for any other work-center in the factory. This would help Boeing to change the control points as and when necessary.

Shot peen was the only control point which was used while the policy was in use. It was initially decided to run the program twice a shift, once at the start and then at the middle of the shift. However, by observing the leverage metrics, it became evident that, it was not necessary to run it so frequently. The program was then run only once: at the start of each shift. Initially, the program output was dispatched to the shot peen work-center, every time the program was run. However, the shot peen supervisors became comfortable with it soon and started using the program on their own.

5.8 Policy use

The policy took approximately 10 months to implement from start to having the CPP program available for the users. The policy was used to schedule work at the shot peen control point for approximately two weeks. The shop supervisor found it very helpful tool, particularly when there was a major production disruption and he had to recover from it. It reduced the time and the effort required to get the flow back to normal.

5.9 Experiment termination

However at this point, the implementation was interrupted by major a re-organization in the 777 flap-support team. Most of the management was transferred. Also, around the same time there was a down-turn in demand, making it no longer necessary to use a tool like the CPP to schedule the work. The utilization was reduced to such an extent, that scheduling optimally was not an issue. As a result, the CPP project was aborted at a very early stage.

5.10 Survey

We conducted a survey among seven 777 flap support team members to get their views about the CPP and a variety of other issues. Overall, the response indicated a positive feedback for the policy. This section analyzes the replies to the survey. *Appendix A* shows the survey forms with the actual replies.

1. **MRP II**

When questioned about their level of satisfaction with the MRP system, the response was mixed. The top management and the shop-floor personnel were unanimous: they were highly dissatisfied with it. The middle management and the MRP system administrators, on the other hand, had no complaints with it. The major complaints with the MRP system were that it was very inaccurate and inflexible. Another complaint was that it did not reflect and, therefore, did not encourage any shop-floor improvement measures carried out by the management (e.g. the trucking CONWIP loops, and the visual control system). Six out of seven users thought there were definite advantages in maintaining finite buffers in the system, even thought parts were being introduced every TAKT time.

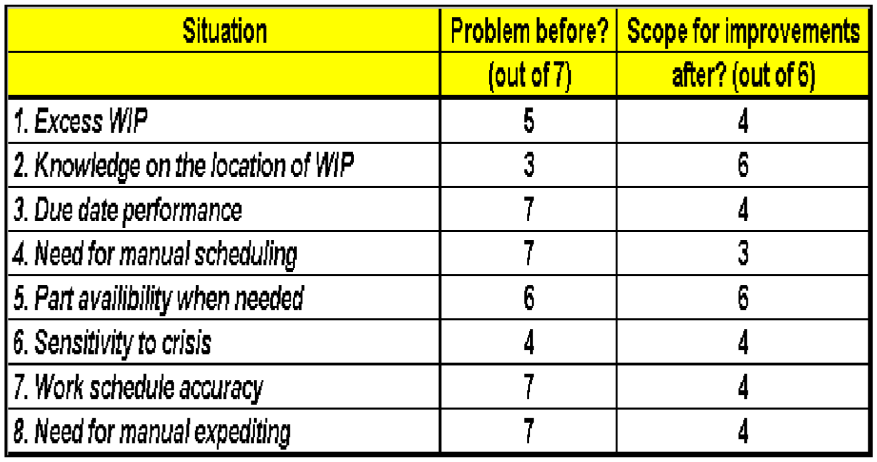
2. **Factory problems and CPP**

Figure 5-4 shows the results of the survey in which the users were questioned about the problems associated with the factory, and whether they thought the CPP could definitely[3] alleviate the problems for them. More than 50% of the users thought the CPP would improve the situation in all cases. The response was best for the part availability (#5) and the knowledge on the location of WIP (#2), which indicates that the CPP should make factory operation become smoother and more predictable.

3. **Feedback on CPP**

Figure 5-5 shows the survey-rankings for the different features of the CPP. Most users thought its best value was in driving down the inventory levels in the factory. Also, its utility

[3]Therefore, "Dont know's" and "Maybe's" have been excluded.



Situation	Problem before? (out of 7)	Scope for improvements after? (out of 6)
1. Excess WIP	5	4
2. Knowledge on the location of WIP	3	6
3. Due date performance	7	4
4. Need for manual scheduling	7	3
5. Part availibility when needed	6	6
6. Sensitivity to crisis	4	4
7. Work schedule accuracy	7	4
8. Need for manual expediting	7	4

Figure 5-4: Before and after comparison

as a re-sequencing mechanism received consistent high ranks from all the users. This reiterates the result from the Figure 5-4 on part availability and location of WIP.

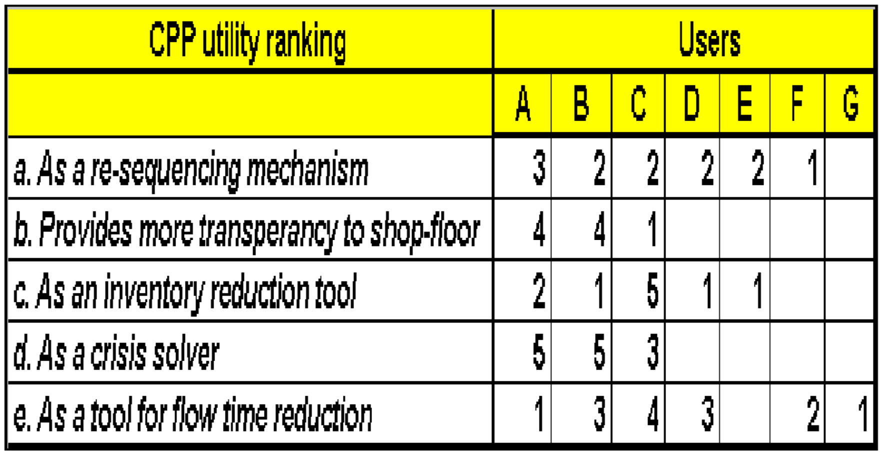
The results from both surveys provide a confirmation to our claim that the CPP will be provide very good performance when there is a high level of *disorderly mingling of parts*. In terms of their complaints with the CPP, many users (four out of seven) thought the buffer sizes and hedging times need to be optimized.

5.11 Conclusions

Though the policy seemed to be on a promising track, the CPP program was never used for long enough time to achieve demonstrable lean improvements for the factory. It was not a success in those terms, but it was a success in terms of the achievements and lessons we gathered from this experiment. They will help in future CPP implementations. In this section, we discuss the achievements and the lessons gained from the experiment.

5.11.1 Achievements

1. We were able to apply the hedging point theory to a real factory situation. Though the policy had to be modified along the way, the underlying theory remained intact.



CPP utility ranking	Users						
	A	B	C	D	E	F	G
a. As a re-sequencing mechanism	3	2	2	2	2	1	
b. Provides more transperancy to shop-floor	4	4	1				
c. As an inventory reduction tool	2	1	5	1	1		
d. As a crisis solver	5	5	3				
e. As a tool for flow time reduction	1	3	4	3		2	1

Figure 5-5: CPP utility analysis

2. We were able to deliver a program which Boeing used briefly and could use in future.

3. Although we were not able to observe any quantitative improvements in the performance, the users saw it as a very useful scheduling tool.

Looking back at the original goals of the experiment (Section 5.1), Goal #1 and Goal #4 were completely achieved, while Goal #2 was achieved to a small extent. Unfortunately, the policy did not have a long enough life, to make any progress on fine-tuning of the policy parameters (Goal #3).

5.11.2 Lessons

1. We had started with the token-based version of the policy. But we soon realized that it was very difficult to implement the token-network because of the reentrant nature of the flow and the large number of part-types. The time-based version is also more suitable for this kind of make-to-order manufacturing system.

2. An important lesson was that all the work-centers need not be control points. It was necessary to choose a only a few critical points where the policy needs to be applied. The location and number of control points will again depend on the manufacturing environment.

3. *Batching policy*: In the original formulation of the policy, we did not know how to deal with batching. The modification for batching work-centers came up during the course of the experiment.

4. The different *buffer strategies* had to be designed because all the work-centers were not control points. The CONWIP loops between the control points is a very helpful modification.

5. The interaction of the CPP program with the MRP system was an important issue in this case and will be important for future implementations too. The subscription mode used in this experiment was easy to develop and operated efficiently at real time. This architecture should also suit future implementations.

In the Boeing experiment, we spent a major portion of the initial time trying to formulate the policy, rather than having the policy ready. We proposed the policy to Boeing in the fifth month of the project, which was 50% of the total lead time (10 months) for the implementation of the project. In terms of the time-distribution, we spent the first five months to come up with a proposal, the next three months were spent in the prior preparation activities, while the remaining two months were used to develop the real time program. This experience will help to cut down on this time, especially the time spent on the first two activities. We will start at a higher point on the learning curve and the lead time for a similar project should be significantly shorter.

Chapter 6

Simulation Studies

The previous chapters of the thesis deal with factory scheduling and the details of the control point policy. The purpose of this chapter is to present a collection of simulation results to understand the behavior of manufacturing systems under CPP-control. The approach followed is to understand the behavior for very simple manufacturing systems and based on these conclusions, to conjecture behavior for more complicated systems. We also compare the performance of manufacturing systems under CPP and other commonly used control policies discussed in Section 3.6. In the same section, we make the following claim: *The CPP will provide better performance as compared to other scheduling policies, whenever the potential effect of disorderly mingling of parts in the system is large.* We present simulation results for a line which confirm this claim.

The advantage of simulation is that we can easily represent arbitrary systems and control policies without introducing approximation errors. Its inherent appeal is largely due to its flexibility as compared to rigid mathematical models. Complex models and behaviors can be easily constructed and introduced into the models. Despite its strengths, simulation is not without pitfalls. Lengthy development time and run time are the major problems associated with it.

Section 6.1 lists the goals of the simulation studies. In Section 6.2, we briefly describe the software and the computing resources used for the simulation studies. Section 6.3 is a small section which defines the terms which are used in the simulation models. Section 6.4 and Section 6.5 describe the experiments and the results for the single part-type simulations. Section 6.6 -

Section 6.9 describe the simulation comparison studies of the different policies, for the case of a two part-type assembly process. Finally, Section 6.10 concludes the chapter.

6.1 Goals

The goals of the simulation exercise were the following:

1. To test the performance of and establish confidence in the simulation package (SIMUL8)[1].
2. To understand the behavior and performance of simple CPP-controlled tandem lines with single part-type.
3. To compare the performance of CPP with other control policies.
4. To provide directions for future research.

6.2 Software and systems

Discrete event simulation is a very CPU-intensive task. A model of the manufacturing system is represented by a program incorporating a pseudo-random number generator, and statistics of the system are collected by executing the program. It is necessary to run the simulation for a very long time to get good estimates of the performance measures of the system under study. For example, if one is simulating a scheduling policy for a manufacturing system with random machine failures, the simulation must cover a period with many failures of different lengths before getting a good estimate of steady-state effect of the scheduling policy. Depending on the type and frequency of the discrete events in the model, the length of the simulation run needs to be decided.

We used a software package called SIMUL8 to perform the simulation studies described in the chapter. It is available on a Microsoft Windows platform with a convenient GUI interface. Appendix B provides a critique of the software, especially as regards comparison with the in-house developed unix-based simulations.

[1]© 1998 Visual Thinking Int. Ltd. Appendix B contains an appraisal of the program.

The software package allows the user to build different models of manufacturing systems very easily. Different kind of manufacturing architectures, process flows, and scheduling rules can be easily created. A user can do this by introducing external code-modules into the simulation program. There is a simple software language which provides the external interface. We used our own modules to design the flow of parts through the system and the scheduling policy at the work-centers. Also, we did not rely on the statistics collected by the package. We performed all the data collection and analysis with our own programs.

The experiments were carried over a period of 6 months. The computing resources used for the experiments were the COLUMBIA2 WINDOWS NT server (Dell Optiplex GX1P PII 450 MHz Processor, 284 MB RAM). Also, the computing resources in the MIT Operations Research Center (ORC) and the MIT Center for Innovation and Product Development (CIPD) were used for the experiments[2].

6.3 Definitions

In this section, we introduce some of the terms which are used in the simulation models. Many of them are generic manufacturing terms. However, there is a lot of confusion, which still persists regarding the exact definition of the terms and their interpretation. Therefore, they are defined again in this section for absolute clarity.

Definition 11 ***Demand lead time*** l_q

The *demand lead time* (l_q) for a part-type q is defined as the time difference between the arrival date of demand and the due date for parts belonging to that part-type.

Definition 12 ***Service rate*** σ_q

The *service rate* (σ_q) for a part-type q is defined as the frequency of the n^{th} part belonging to the part-type q reaching the finished goods buffer on or before its due date.

Definition 13 ***Lead time*** γ_q

[2] Thanks to Omar Gzouli (ORC) and John Hull (CIPD) for making this possible.

The *lead time* (γ_q) for a part-type q is defined as the time interval between when a part belonging to the part-type is released into the factory and the time it is shipped to the customer. It is a metric of the manufacturing system.

Definition 14 ***Work In Progress*** WIP

Work in progress (WIP) includes all the material present in the factory including the raw material (material procured by the ordering department) and the finished material (material in the finished goods buffer before it is shipped out of the factory).

Definition 15 ***Demand arrival distribution*** DA_q

On the average, a demand (d_{nq}) for n^{th} part belonging to part-type q arrives every $TAKT_q$ time for the part-type q. However, there is variability associated with the demand arrival process for the part-type. The *demand arrival distribution* (DA_q) describes the variability. In our simulation, we have assumed it to be a normal distribution with a mean zero and a specified standard deviation.

Definition 16 ***Due date allocation distribution*** DDA

On the average, the due date (D_{nq}) to a part n belonging to part-type q is assigned using the formula:

$$D_{nq} = d_{nq} + l_q \tag{6.1}$$

However, there is variability associated with the due date allocation process[3]. The deviation from the above formula is defined using the *due date allocation distribution* (DDA_q). We have assumed it to be a normal distribution with mean zero and a specified standard deviation.

Definition 17 ***Order hedging time*** HT_{0q}

In many cases, there is a time difference between the arrival time of the demand (d_{nq}) and time the demand is accepted by the factory. In some cases, it is necessary and, in other cases,

[3] A company may assign different due dates (D_{nq}) to different part n's. Also, different customers may demand certain due dates.

the company deliberately chooses to do so. For example, if the demand for a n^{th} part arrives very early as compared to $(D_{nq} - l_q)$, the company may chose to wait for sometime before accepting the part into the system. This is done using the *order hedging time* (HT_{0q}). The order hedging time (HT_{0q}) is a constant for a part-type q.

6.4 Single part-type simulations

The objective of this section is to analyze the behavior of a simple three stage single part-type manufacturing system under CPP control. Each stage in the manufacturing system is a control point. The objective is to develop an intuitive understanding of the system and to interpret the policy parameters accurately. The experiments presented in the following section illustrate the important results obtained.

6.4.1 Model details

The simulation model is as shown in Figure 6-1[4]. Machine M_1, M_2 and M_3 are machines in the factory. Each machine has a deterministic processing time of 1 time unit. The only source of variability in the system are machine failures and repairs. Each machine has an exponentially distributed time to fail with a mean 100 time units[5]. The time to repair for each machine is also exponentially distributed with mean 10 units. Buffers B_1 and B_2 have a fixed size of 10. Buffer B_3 is infinite (actually 1000, but this can be treated as infinite for our simulation models). Buffers B_0 is also infinite, except in case of Experiment 4.

The capacity of the above line was found to be 0.81[6]. We have assumed demand rate of *0.5* $(TAKT = 2)$ for the part-type. The demand-arrival process in the system is deterministic. The orders arrive into the system every TAKT time interval; i.e.e the demand arrival distribution (DA) is $N(0.0, 0.0)$. Therefore, the n^{th}demand arrives at time:

[4]Since there is only a single part-type q in the system, the subscript q is dropped for all parameters.

[5]The failures in the model are time-dependent rather than operation-dependent, because that is the type provided by the software.

[6]Using the *Explongline* models. It is an approximation, but, in any case, the demand rate (0.5) is not close to capacity.

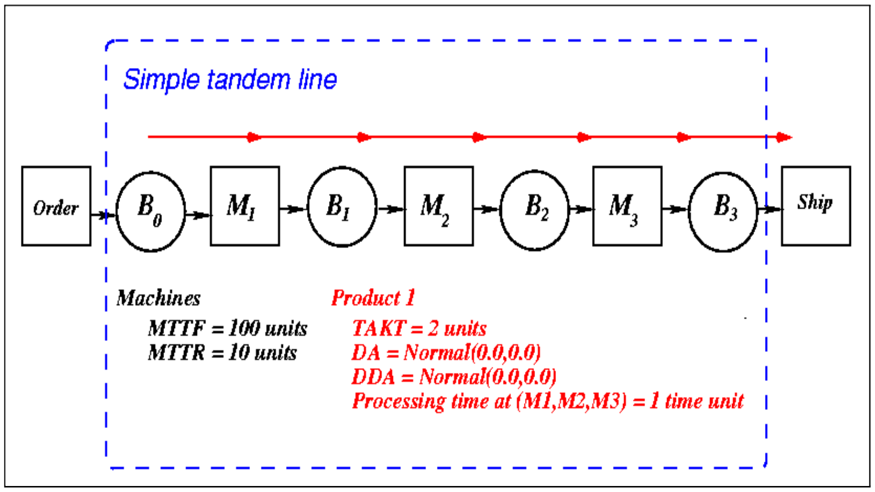


Figure 6-1: Model for three stage, single part-type simulation

$$d_n = A' + nTAKT + DA \tag{6.2}$$

$$= A' + nTAKT \tag{6.3}$$

A' is a positive constant for the system. *Order Machine* is a pseudo machine representing the ordering department of the company. It decides when to order raw parts into the factory i.e. buffer B_0. *Ship Machine* is another pseudo machine representing the shipping department of the factory. It controls the finished goods buffer (B_3) for the factory, and lets parts out of the buffer after their due dates. If the parts in the finished goods buffer (B_3) have already passed their due date when they arrive there, they are immediately shipped by the *Ship Machine*. If not, they wait till then. Both the *Order Machine* and the *Ship Machine* are 100% reliable[7]. Parts arrive at the *Order Machine* every TAKT time. The due dates are allocated deterministically; i.e. the due date allocation distribution (DDA) is $N(0.0, 0.0)$. The n^{th} part has a due date:

$$D_n = A + nTAKT + DDA \tag{6.4}$$

[7]For the purpose of simulations, these machines also perform the function of logging statistics for the system.

$$= \quad A + nTAKT \qquad (6.5)$$

A is a positive constant for the system. The difference $(A - A')$ is thus the *demand lead time* l for the parts. On arrival of the demand (d_n), the *Order Machine* may or may not immediately order the part into the system, depending on whether the part passes the CPP logic at the machine. When is does order the part into the system, the part is put into the B_0 buffer. The part then becomes WIP for the system until it moves out of the finished goods buffer B_3.

The CPP logic is used at the *Order Machine* and *each machine* M_1, M_2 and M_3, to schedule the loading of parts on to the respective machines. Thus, there are four control points in the factory. Since there is only a single part-type in the system, there is no part-type ranking.

6.4.2 CPP logic for single part-type

The CPP logic is modified because of the single part-type and the deterministic nature of the system. It performs the following steps *once every time instant* before loading a part on to a machine:

Step1. Parts are sorted according to their due dates in the upstream buffer. The part with the earliest due date is placed at the top of the buffer.

Step 2. If the upstream buffer is empty, the machine is starved. Exit.

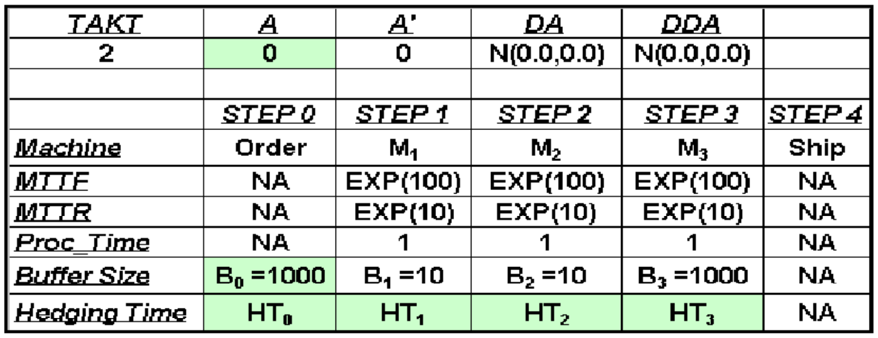
Step 3. If the machine is idle, the Step 4 to Step 6 are checked *for the first part* in the queue[8]. Else exit.

Step 4. The first part is checked for the *blocking logic* i.e. if the buffer downstream of the machine is full. If blocked, exit.

Step 5. If not blocked, the *earliness logic* is checked. If early, exit.

Step 6. Else the part is loaded on to the machine.

[8]If the first parts fails the CPP conditions, all other parts would fail the them also.



TAKT	A	A'	DA	DDA	
2	0	0	N(0.0,0.0)	N(0.0,0.0)	
	STEP 0	STEP 1	STEP 2	STEP 3	STEP 4
Machine	Order	M_1	M_2	M_3	Ship
MTTF	NA	EXP(100)	EXP(100)	EXP(100)	NA
MTTR	NA	EXP(10)	EXP(10)	EXP(10)	NA
Proc_Time	NA	1	1	1	NA
Buffer Size	B_0 =1000	B_1 =10	B_2 =10	B_3 =1000	NA
Hedging Time	HT_0	HT_1	HT_2	HT_3	NA

Figure 6-2: Reference table for single part-type simulations

6.5 Experiments

In this section, primarily, we study the impact of two parameters on the system performance:

1. Demand lead time (l)

2. Hedging times (HT_s)

We have restricted the results presented in the section to these parameters, because there are many parameters in the system and we found the performance of the system to be most interesting in its response to them. Figure 6-2 provides the list of default input parameters for the system. The parameters in the shaded boxes are the ones which are changed during the experiments. The rest are kept constant in all the experiments.

6.5.1 Effect of demand lead time

Experiment 1

In this experiment, we have varied the demand lead time ($A - A'$). All the hedging times (HT_0 to HT_3) are uniform and are higher than the maximum ($A - A'$). So the part never fails the *earliness condition* of the CPP at any machine. It passes straight through the system into the buffer B_3, *unless* it gets blocked at some point.

Figure 6-3 shows the effect of ($A - A'$) on the production rate and the service rate. The production rate stays constant and is equal to the equal the demand rate. However the service

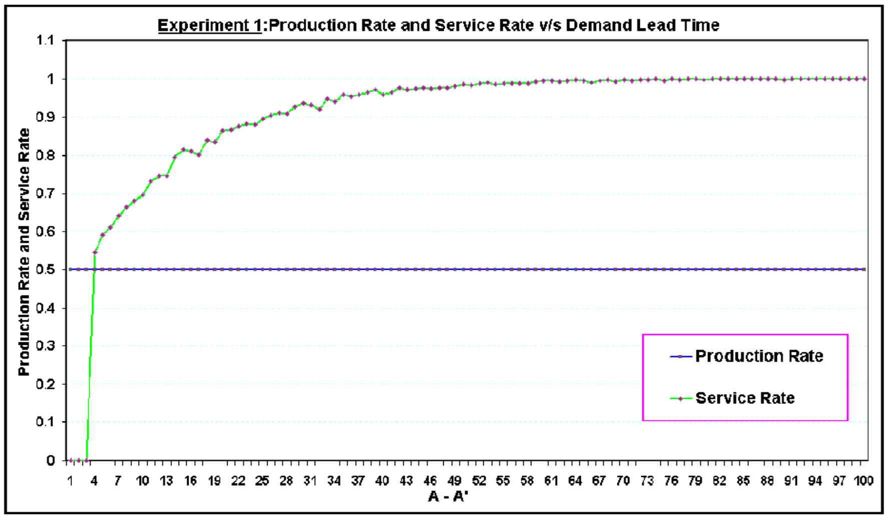


Figure 6-3: Effect of demand lead time on production rate and service rate

rate goes on increasing, before saturating close to 100% at $(A - A' \sim 70)$. The marginal improvements at high values of $(A - A')$ are small. Figure 6-4 shows the effect on the cycle time and the lateness. The cycle time for the system remains constant at approximately 7.0. The lateness decreases rapidly initially, and at very high levels $(A - A' \sim 70)$, the lateness is almost zero. But that comes with a price. Figure 6-5 shows the effect of $(A - A')$ on the lead time γ and the WIP: they both increase (almost linearly) with $(A - A')$. The Figure 6-5 also shows the increase in the average B_3 level. The increase in the WIP level is due to all the inventory being put into B_3. All three curves in Figure 6-5 continue increasing even though the lateness and service rate have saturated at lower values of $(A - A')$.

These graphs illustrate the classic dilemma managers face: their natural reaction to dissatisfied customers is to quote higher demand lead times for the products. That improves their service rate performance, but it leads to higher inventories and time-to-market. Also the marginal improvement in service rate with higher levels of demand lead times are much smaller as compared to the increase in the WIP levels and the lead times. The appropriate demand lead time is a trade-off between these conflicting behaviors.

The lateness and the service rate achieved for a particular value of $(A - A')$ are the best values which can be achieved for the system with that demand lead time.

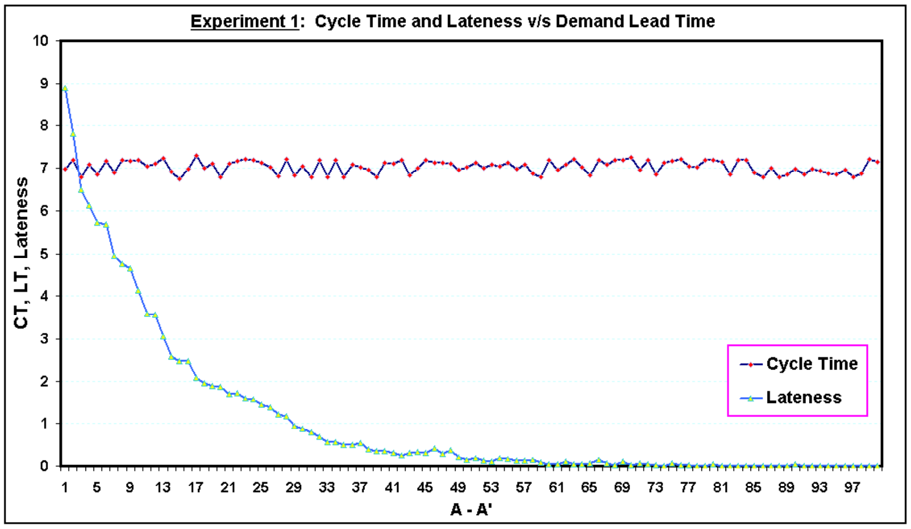


Figure 6-4: Effect of demand lead time on cycle time and lateness

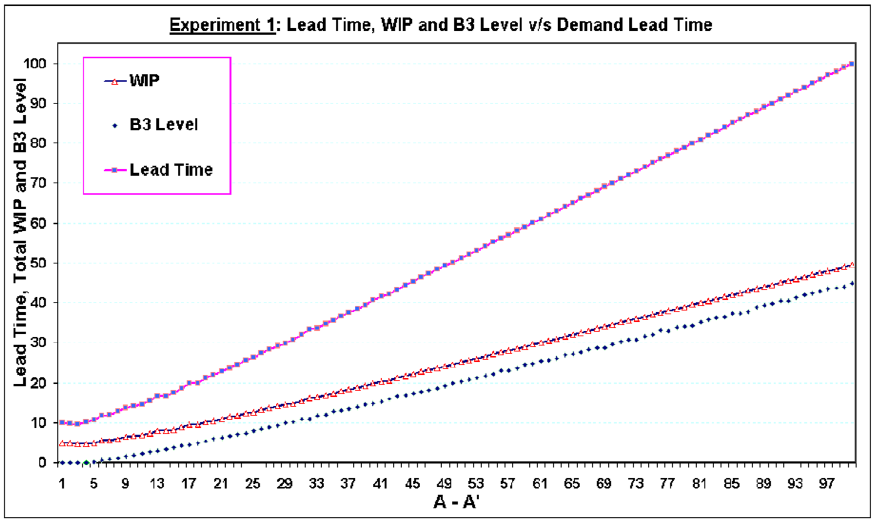


Figure 6-5: Effect of demand lead time on lead time, WIP and B_3 buffer level

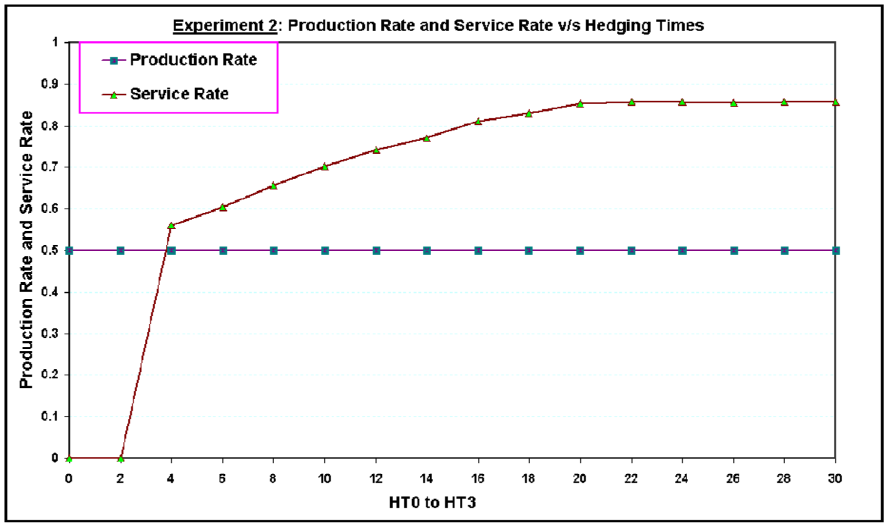


Figure 6-6: Effect of hedging times on production rate and service rate

6.5.2 Effect of hedging times

Experiment 2

In this experiment, the demand lead time ($A - A'$) is kept constant at 20. The hedging times for the *Order Machine* and machines M_1, M_2 and M_3 (HT_0 to HT_3) are increased *together* from 0 to 30. Since, all the hedging times are uniformly increased, once the part passes the earliness logic at the *Order Machine*, it passes it at all the machines downstream. Figure 6-6 shows the effect of increasing the hedging times on the production rate and the service rate. The production rate stays constant, however, the service rate improves as the hedging times are increased. As the hedging times increase, the parts get processed earlier in the system, and the service rate improves. However, it stops improving when ($HT = HT_0 = HT_1 = HT_2 = HT_3 = A - A'$).

Figure 6-7 shows the effect on the lead time, WIP and lateness. As the material is let into the factory earlier, the amount of time it spends in the system, before it is passes out of buffer B_3 (at its due date), increases. The lead time and the WIP levels increase. The graphs for the lead time and the WIP follow Little's Law [Little, 1961][9].

[9] *Little's Law*: The relationship between the production, or arrival, rate (λ), WIP (L), and lead time (W): $L = \lambda W$.

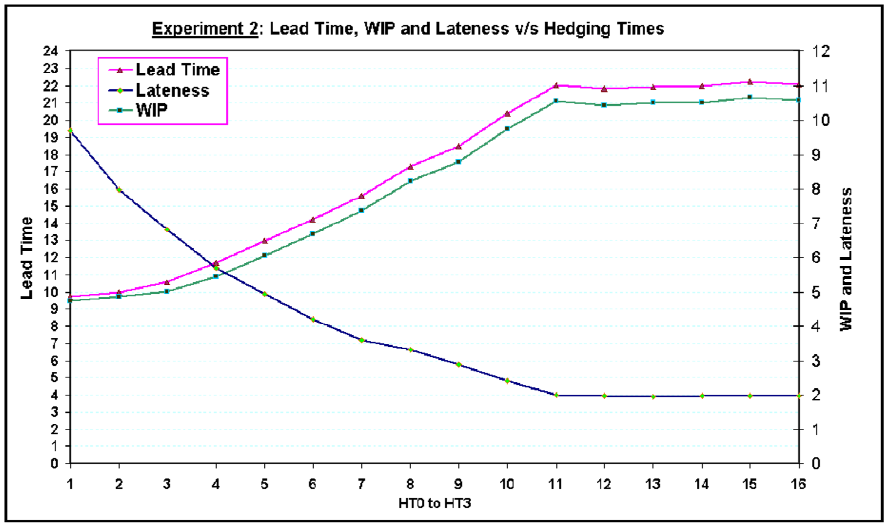


Figure 6-7: Effect of hedging times on lead time, WIP and lateness

There is an upper bound to the changes in the above metrics at ($HT = A - A'$). Also, the best values for lateness and service rate at that point are equal to those obtained in Experiment 1.

6.5.3 Relation between hedging times and demand lead time

Experiment 3

Experiment 2 was repeated for ($A - A' = 10$) and the two results were compared in Experiment 3. Figure 6-8 compares the service rate for ($A - A' = 20$) [CASE 1] and ($A - A' = 10$) [CASE 2]. The dotted lines in the graph indicate results for CASE 2. The two curve coincide with each other until the hedging times are increased up to 10. Beyond that point, the improvement ceases for CASE 2, while the improvement for CASE 1 continues until ($HT = 20$).

Figure 6-9 shows the effect on the lateness and WIP for CASE 1 and CASE 2. The two curves match each other exactly until the hedging times are 10. The graphs illustrate that the policy performance is not only the effect of the hedging times but also their relation to demand lead time.

There is a small constant error in the graphs. Otherwise, the two curves should exactly coincide. This error is due to the rounding-errors of the statistics collection during the simulation runs.

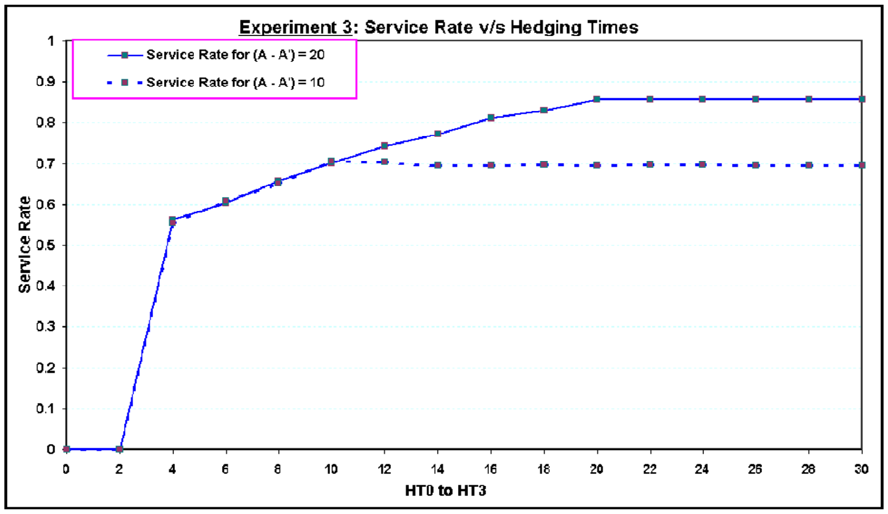


Figure 6-8: Service rates for CASE 1 and CASE 2

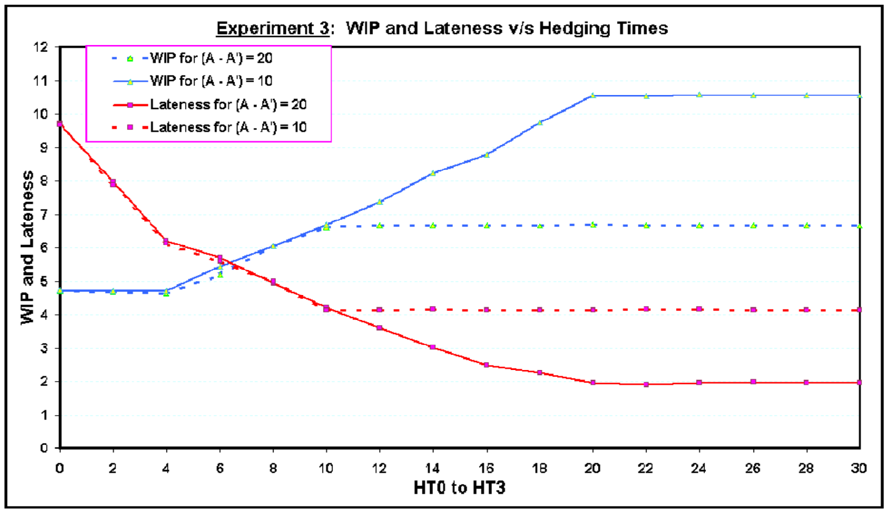


Figure 6-9: WIP and lateness for CASE 1 and CASE 2

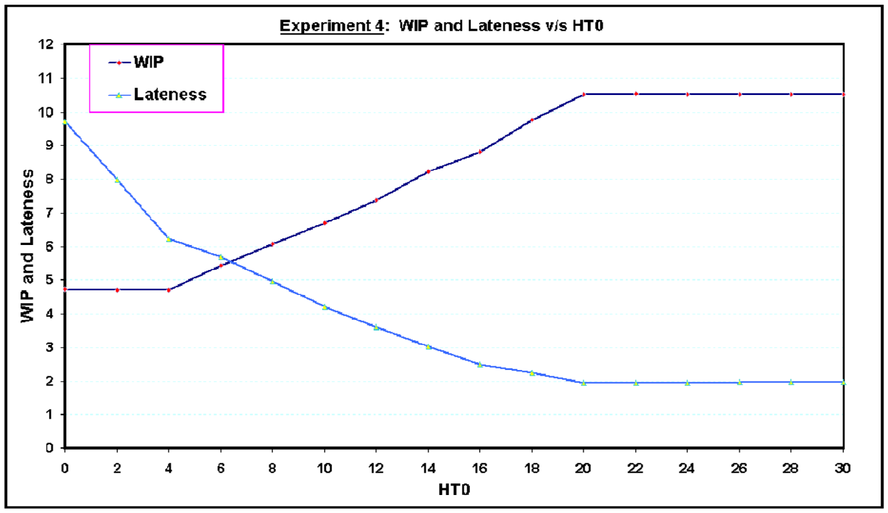


Figure 6-10: Effect of HT_0 on WIP and lateness

It indicates that for any time-based policy, the first thing to decide is the demand lead time for the part-type. The demand lead time decides the upper bound for the service rate (assuming the policy used inside the system is reasonable). This make the process of quoting the demand lead time for the products the most important thing for a company: not only to meet the service targets but also to drive the performance of the scheduling policy.

6.5.4 Effect of order hedging time and the B_0 buffer

Experiment 4

One of the claims made in Section 2.5.1 is that, for a make-to-order environment, when the supply process is reasonably deterministic, the best place to hold and control the inventory is buffer B_0 . The order hedging time (HT_0) along with the in-coming material buffer B_0, provide the necessary control for the *process of ordering parts* into the buffer B_0. The order hedging time (HT_0) decides how early the material is ordered, while the size of buffer B_0 determines the amount of inventory that can be ordered for the factory. The internal hedging times HT_1, HT_2 and HT_3, on the other hand, control the *process of releasing parts* into the manufacturing system from buffer B_0 . In this section, we present the results of Experiment 4, which illustrate the importance of HT_0 and B_0.

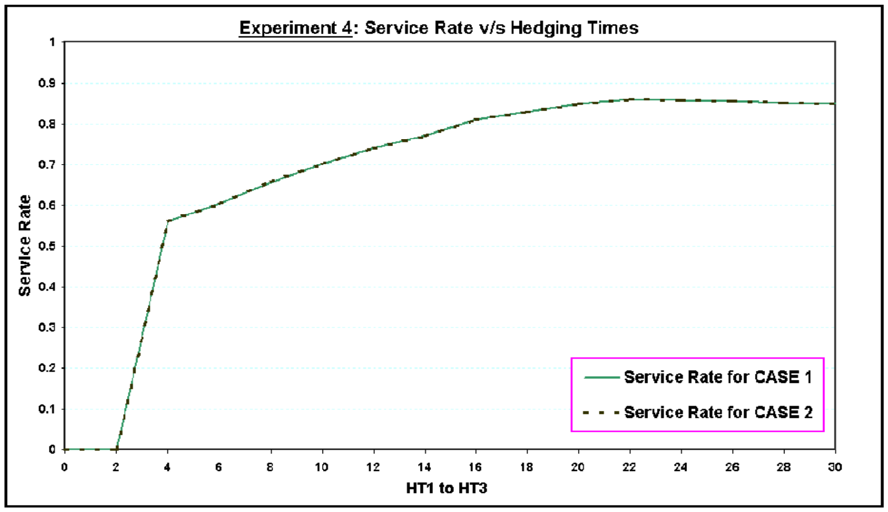


Figure 6-11: Service rate for CASE 1 and CASE 2

In the first part of Experiment 4, the order hedging time (HT_0) is increased from 0 to 30. The internal hedging times HT_1, HT_2 and HT_3 are kept constant ($HT_1 = HT_2 = HT_3 = A - A' = 20$). The B_0 buffer size is infinite in this experiment. Therefore, once the part is ordered by the Order Machine, it flows through the system without interruption (unless blocked at B_1 or B_2). Figure 6-10 shows the effect of ordering early on the WIP and lateness. The WIP level in the system increases because material is put into the B_0 earlier as HT_0 increases. For the same reason, the lateness performance of the system improves. At small values of HT_0, the ordering process is delayed and therefore, the lateness is high and the WIP level is low.

In the second part of the Experiment 4, the order hedging time HT_0 is kept constant and ($HT_0 = A - A' = 20$). The internal hedging times HT_1, HT_2 and HT_3 are uniformly increased from 0 to 30. In CASE 1, the buffer B_0 is infinite, while in CASE 2, the buffer B_0 is finite and has size 5. Figure 6-11, Figure 6-13 and Figure 6-12 compare the performance measures for CASE 1 and CASE 2. As seen in Figure 6-11, the service rate is identical for both the systems[10]. Therefore limiting the inventory in the buffer B_0 does not affect the service rate.

Figure 6-12 shows the effect on buffer level B_0 and buffer level B_3 for CASE 1 and CASE 2. The full lines are the curves for CASE 1, while the dotted lines are the curves for CASE 2.

[10]Also, the lateness and the lead times for the system are found to be identical in both cases.

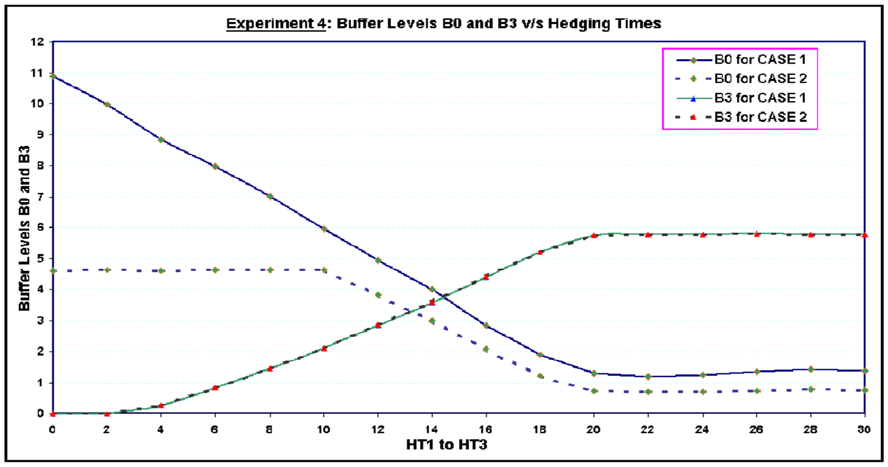


Figure 6-12: B_0 buffer level and B_3 buffer level for CASE 1 and CASE 2

First consider only CASE 1. As HT_1, HT_2 and HT_3 are increased, an increasing amount of material is transferred from B_0 to B_3. The parts spend less time in B_0 and spend more time in B_3. The change is linear up to ($HT_1 = HT_2 = HT_3 = A - A'$), after which the effect ceases. We call this effect of displacement of material from one buffer to another the *buffer-transfer effect*.

The dotted lines in Figure 6-12, show the effect for CASE 2. In this case too, the *buffer-transfer effect* takes place from B_0 to B_3. The behavior for buffer B_3 is the same in either case, but by limiting the amount of inventory in the buffer B_0, we are able reduce the raw material inventory. The B_0 level is lower in CASE 2 as compared to CASE 1. The effect is observed until the average inventory level in the buffer B_0 is less than 5.

Figure 6-13 shows the effect on WIP level performance for the two cases. The WIP levels for the CASE 2 are better than that for CASE 1; the improvements are larger for small values of the internal hedging times. By clamping down on the B_0 buffer, we are able to achieve inventory reductions for the system. It is important to note that the service rate is identical for both the cases.

This is another important lesson for factory managers. Many of them do not realize the importance of the order and release process and the control of the B_0 buffer. Ordering parts too early leads to better service performance, but is also leads to higher WIP levels. Most factories

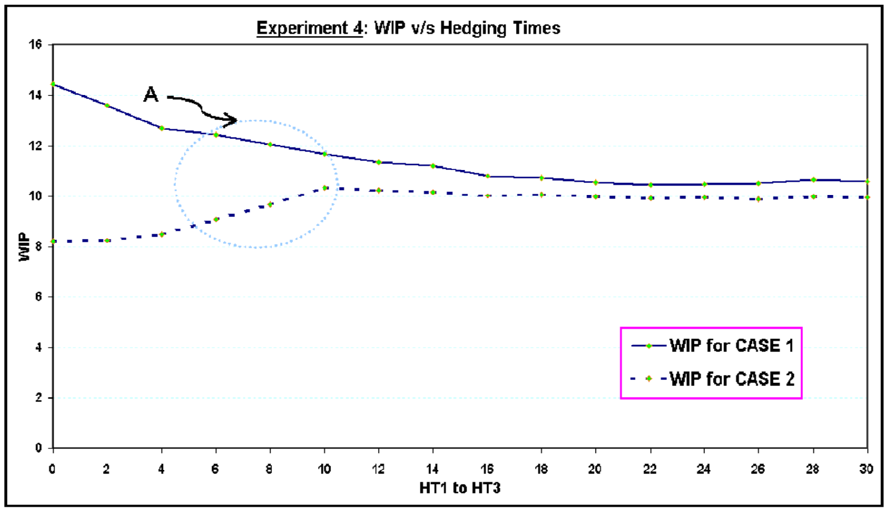


Figure 6-13: WIP for CASE 1 and CASE 2

operate in the vicinity of area A in Figure 6-13 i.e. they order far in advance of the release process. If it is accompanied by proper control of the B_0 buffer, significant improvements in WIP can be achieved[11].

6.5.5 Effect of internal hedging times

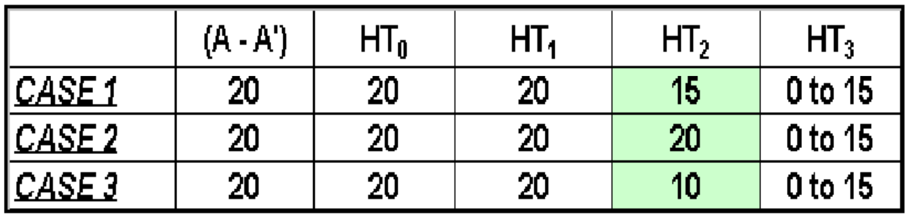
Experiment 5

In Experiment 5, we study the effect of the internal hedging times on the system. We focus on HT_3 and HT_2. Figure 6-14 shows the reference table for the experiment.

First consider only CASE 1, where HT_3 is increased from 0 to 15 for ($HT_2 = 15$). Figure 6-15 shows the WIP and the buffer levels for CASE 1.

As the HT_3 is increased the *buffer-transfer effect* takes place between the B_2 and the B_3 buffers. As HT_3 is increased, material from B_2 is processed earlier and transferred into B_3. The rate of decrease in buffer B_2 and the rate of increase in buffer B_3 are identical. Therefore, the WIP in the system should remain the same, because the B_0 level and B_2 level are the same.

[11]This effect is similar to what the control points in the CPP seek to achieve. By creating intermediate loops in the factory, they try to replicate the sequencing and WIP-control process at these points, like that at the Order Machine and the finite B_0 buffer.



	(A - A')	HT_0	HT_1	HT_2	HT_3
CASE 1	20	20	20	15	0 to 15
CASE 2	20	20	20	20	0 to 15
CASE 3	20	20	20	10	0 to 15

Figure 6-14: Reference table for Experiment 5

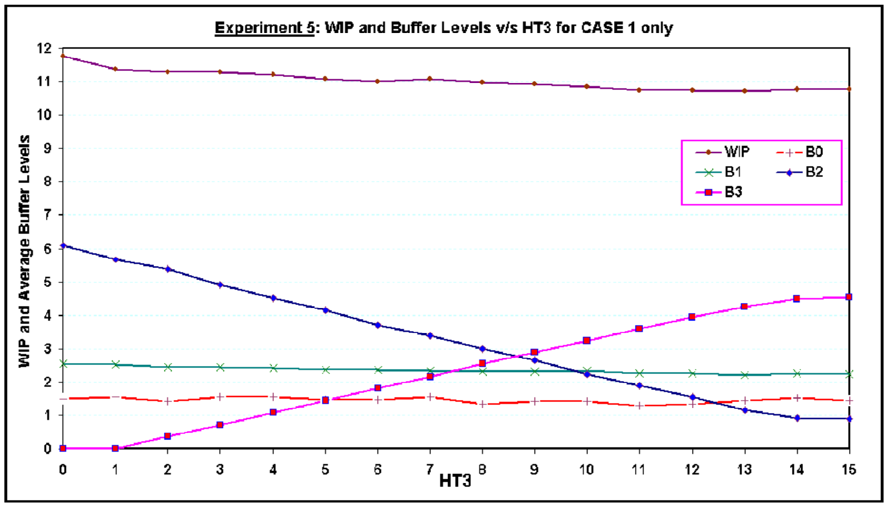


Figure 6-15: WIP and buffer levels for CASE 1

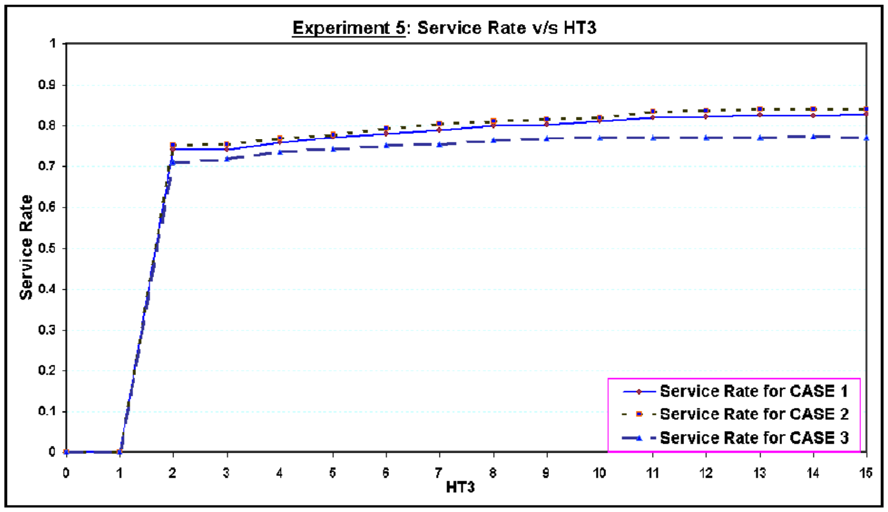


Figure 6-16: Service rate for CASE 1, CASE 2 and CASE 3

However, the WIP in the system decreases slightly. This is because the parts also spend a unit time, on the average, at machine M_3. Therefore, the curve for B_3 buffer is displaced to the right by one time unit. Otherwise, the WIP level would have remained the same.

Figure 6-16 shows the service rate for the three cases. The service rate for CASE 2 is the best followed by that for CASE 1 and CASE 3 for each values of HT_3. It indicates that the effect of hedging time HT_3 (and thus HT_{i+1}) at machine M_3 depends on the hedging time HT_2 (and thus HT_i) of the next upstream machine M_2. Similarly, it can be shown that the effect of hedging time HT_2 *(*and thus HT_i) at the next upstream M_2 machine, in turn, depends on the hedging time HT_1 (and thus HTHT_{i-1}) of the next-to-next upstream machine M_1. This chain-effect continues further upstream. The effect of hedging time HT_{i+1}, thus, depends on the hedging times (HT_0 to HT_i) at the upstream machines (*Order Machine* to M_i)

This is further validated by Figure 6-17. It shows the *buffer-transfer effect* between B_2 and B_3 for all three cases. The increase in levels for the B_3 buffer is the same for all of them. The drop in B_2 levels have the same slopes, but the B_2 buffer levels are different in each case. We had conjectured that the equation (similar to the invariant equation [Gershwin, 1992]):

$$B_2 \approx \frac{HT_2 - HT_3}{TAKT} \tag{6.6}$$

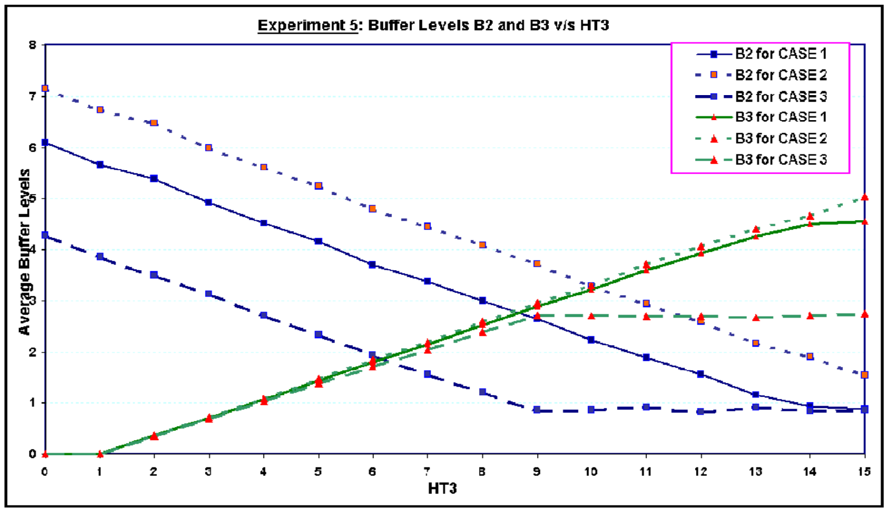
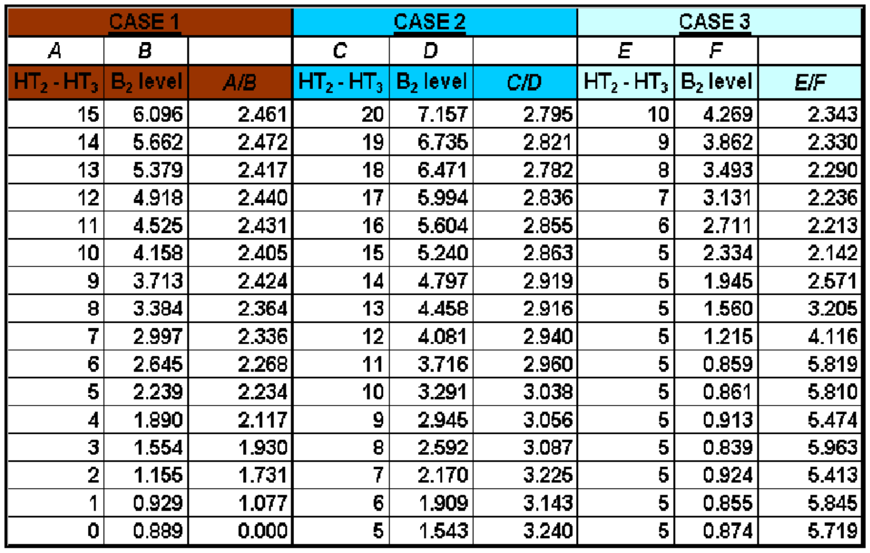


Figure 6-17: Buffer level B_2 and buffer level B_3 for CASE 1, CASE 2 and CASE 3

would hold true. However, it behaves differently for different values of HT_2. Figure 6-18 shows the calculations for the right hand-side of Equation 6.6, for the three cases. The third column in each case should have been equal to the ($TAKT = 2$) if Equation 6.6 was valid. However, the values in the three columns are different and also vary over different values of $(HT_2 - HT_3)$ in each case. The relation cannot be intuitively understood.

6.6 Comparison with other policies

The strong point of the CPP lies in its ability to re-sequence parts at intermediate points in a manufacturing line. Therefore, any system in which this needs to be done, provides an ideal hunting ground for the policy. This feature of the policy was exploited to the maximum for the comparison studies of this section. We conducted a series of experiments on a manufacturing system, where the effect of *disorderly mingling of parts* has reached a large level. Here, we present results from the experiments.



CASE 1			CASE 2			CASE 3		
A	*B*		*C*	*D*		*E*	*F*	
HT_2 - HT_3	B_2 level	*A/B*	HT_2 - HT_3	B_2 level	*C/D*	HT_2 - HT_3	B_2 level	*E/F*
15	6.096	2.461	20	7.157	2.795	10	4.269	2.343
14	5.662	2.472	19	6.735	2.821	9	3.862	2.330
13	5.379	2.417	18	6.471	2.782	8	3.493	2.290
12	4.918	2.440	17	5.994	2.836	7	3.131	2.236
11	4.525	2.431	16	5.604	2.855	6	2.711	2.213
10	4.158	2.405	15	5.240	2.863	5	2.334	2.142
9	3.713	2.424	14	4.797	2.919	5	1.945	2.571
8	3.384	2.364	13	4.458	2.916	5	1.560	3.205
7	2.997	2.336	12	4.081	2.940	5	1.215	4.116
6	2.645	2.268	11	3.716	2.960	5	0.859	5.819
5	2.239	2.234	10	3.291	3.038	5	0.861	5.810
4	1.890	2.117	9	2.945	3.056	5	0.913	5.474
3	1.554	1.930	8	2.592	3.087	5	0.839	5.963
2	1.155	1.731	7	2.170	3.225	5	0.924	5.413
1	0.929	1.077	6	1.909	3.143	5	0.855	5.845
0	0.889	0.000	5	1.543	3.240	5	0.874	5.719

Figure 6-18: Calculations for CASE 1, CASE 2 and CASE 3

6.6.1 Model details

The manufacturing line, under study, is as shown in Figure 6-19. The line can be looked upon as the final assembly process in a factory. Two products: Product 1 and Product 2 undergo final assembly in the line, before they are shipped to the customer. There are three machines in the line. Each of them has an exponentially distributed time to failure with mean 100 time units. The time to repair for each machine is also exponentially distributed with a mean 5 time units. The processing times at each assembly machine are exponentially distributed.

The two products are processed and put into the buffer B_3 , before being shipped. Both the products undergo a complex machining process upstream of the final assembly process. By the time they reach the Order Machine of the assembly process, their arrival has become highly variable.

Product 1 is a *new product* introduced by the company and has demand rate of 0.125 i.e. $(TAKT_1 = 8)$. The company wants to reach the market before anyone else. The consequences of being late for Product 1 are immense. But the company has done a very hasty job in promising the due dates to the customers $(DDA_1 = N(0.0, 4.0))$. Also due to a lot of re-engineering work and introduction delays, the arrival of Product 1 at the Order Machine is highly variable

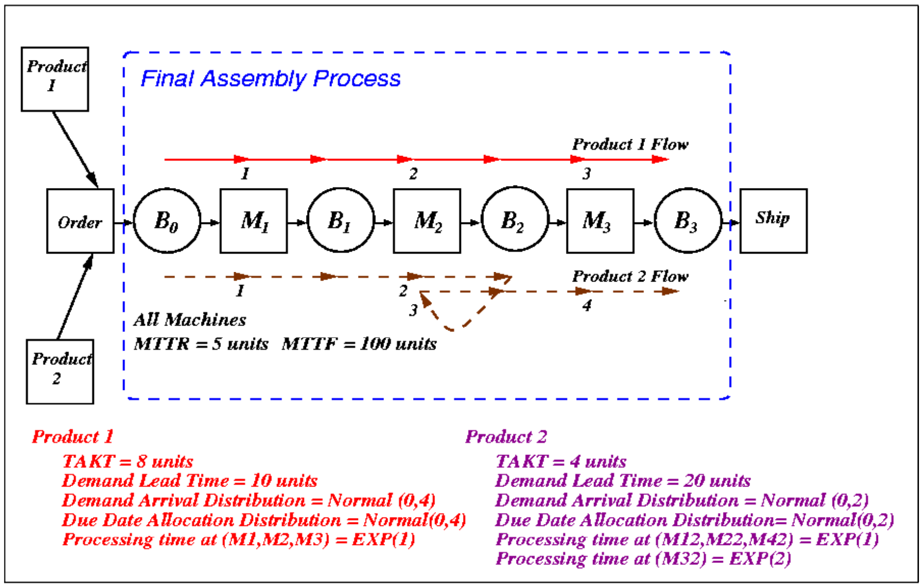

Product 1
Final Assembly Process
Product 1 Flow
1
2
3
Order
B_0
M_1
B_1
M_2
B_2
M_3
B_3
Ship
Product 2 Flow
1
2
3
4
All Machines
MTTR = 5 units MTTF = 100 units
Product 2
Product 1
TAKT = 8 units
Demand Lead Time = 10 units
Demand Arrival Distribution = Normal (0,4)
Due Date Allocation Distribution = Normal(0,4)
Processing time at (M1,M2,M3) = EXP(1)
Product 2
TAKT = 4 units
Demand Lead Time = 20 units
Demand Arrival Distribution = Normal (0,2)
Due Date Allocation Distribution= Normal(0,2)
Processing time at (M12,M22,M42) = EXP(1)
Processing time at (M32) = EXP(2)

Figure 6-19: Model for comparison studies

$(DA_1 = N(0.0, 4.0))$. The company has assigned it a demand lead time $(A_1 - A'_1 = 10)$ in this part of the process. Therefore, the n^{th}demand at the *Order Machine* arrives at time:

$$d_{n1} = 0 + 8n + N(0.0, 4.0) \tag{6.7}$$

The n^{th} part has a due date:

$$D_{n1} = 10 + 8n + N(0.0, 4.0) \tag{6.8}$$

Product 1 has a simple tandem flow through the assembly machines, with a mean processing time of single time unit at each stage.

Product 2, on the other hand, is a relatively stable product of the company, with a demand rate of 0.250 i.e. $(TAKT_2 = 4)$. The arrival process $(DDA_2 = N(0.0, 2.0))$ and the due date assignment process $(DDA_2 = N(0.0, 2.0))$ are less variable. The company has assigned Product 2 a demand lead time $(A - A' = 20)$ in this part of the process.

Therefore, the n^{th}demand at the *Order Machine* arrives at time:

$$d_{n2} = 0 + 4n + N(0.0, 2.0) \tag{6.9}$$

The n^{th} part has a due date:

$$D_{n2} = 20 + 4n + N(0.0, 2.0) \tag{6.10}$$

Product 2 has a reentrant flow as shown in Figure 6-19. It goes through Machine M_2 twice as shown. It has a mean processing time of two time units the second time it comes to the machine M_2. The remaining mean processing times are all single unit.

Problem. Get the best possible performance for *Product 1*, without causing disruption to the performance Product 2.

6.7 Strategy

Due to the TAKT times for each product, there are less number of Product 1*'s* randomly interspersed in the system among the many Product 2*'s*. The performance for these Product 1*'s* should be the best possible, without affecting the performance for Product 2.

The manufacturing system was scheduled using the *commonly used* scheduling policies: KANBAN, CONWIP, basestock, EDD, LSP and CPP. There are many input parameters which can be tuned for each policy, and optimizing over such a large space is an impossible task. Therefore, a suitable strategy had to be developed to get to within a close range of the optimal numbers. The following strategy[12] was adopted *for each policy*:

Step 1. Keep the input parameters for *Product 1* constant. Get the best possible performance for Product 2 by varying the input parameters for Product 2.

Step 2. Do not change the above input parameters for Product 2. Adjust the input parameters for *Product 1* to get the best possible performance for *Product 1*.

[12]We used the trial-and-error method to design the above strategy. The results collected along the way indicated that this strategy was a fair one. However, this strategy is definitely not 100% fool-proof and the results obtained cannot be taken as the absolutely optimal numbers.

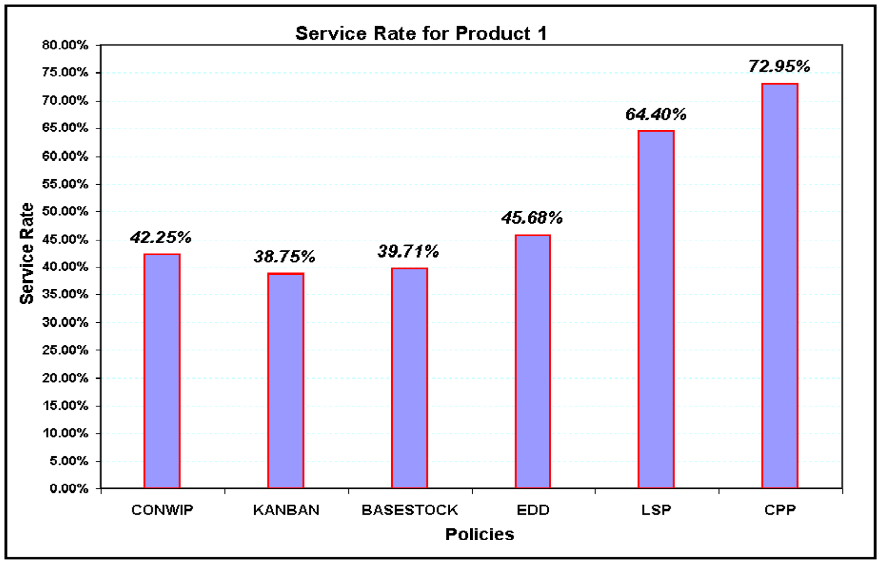


Figure 6-20: Service rates for *Product 1*

Step 3. Check that **Step 2** does not affect the performance obtained for Product 2 in **Step 1**.

a. If it does not, choose these set of input parameters as the result parameters for the problem and exit.

b. If it does, revaluate strategy for *Product 1* in **Step 1** and,

c. Repeat **Step 1** and **Step 2**.

Fortunately, **Step 3b** and **Step 3c** did not have to be repeated more than a few times for most cases. "Best possible performance" in the above strategy means the best service rate. Getting the minimum lateness would have been another option. However, we chose to use the service rate because it is the apparent preferred measure in the industry [Spearman and Zhang, 1999].

We discuss **Step 1** and **Step 2** and **Step 3b** in the above strategy for each policy in detail in Appendix C.

6.8 Graphs for comparison studies

Each trial took about 3 hours to run. Each trial had a warm up period of 25,000 time units and a results collection period of 75,000 time units. It was repeated six times with

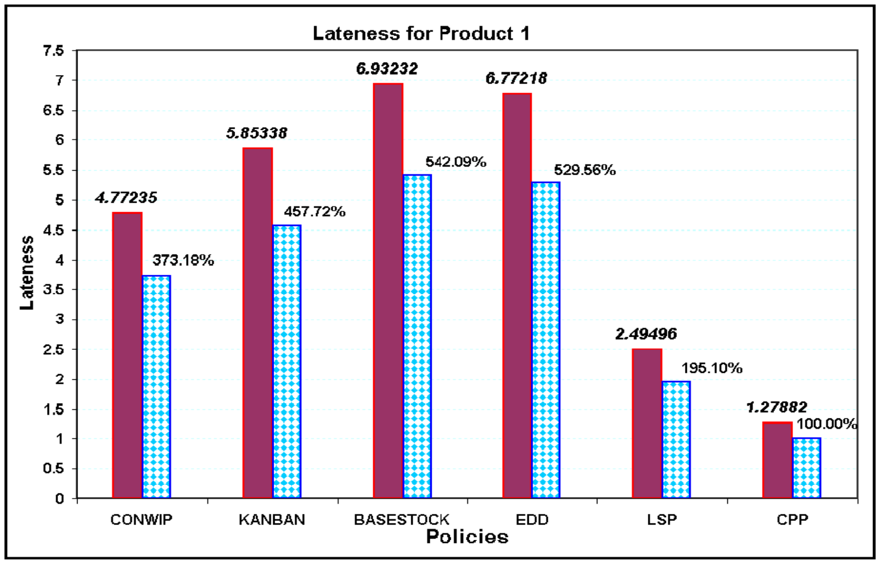


Figure 6-21: Lateness for *Product 1*

different random seeds, and the average values were collected from these runs. The results of the comparison studies are presented in this section. Figure 6-20 - Figure 6-23 show the results for Product 1, while Figure 6-24 shows the results for Product 2.

The CPP performs the best for Product 1 for all the metrics. Figure 6-20 shows the comparison for the service rate (Section 6.3 for definition). The service rates for all the policies are not close to 100%, because of the high level of *disorderly mingling of parts*. The service rate obtained in the CPP is maximum. It is significantly better (almost 1.5 times) as compared to the token-based policies, and better than the other time-based policies.

The differences in lateness (Section 6.3 for definition) are much more drastic. Figure 6-21 compares the lateness for the different policies: the full-shaded bars indicate the absolute values of lateness, while the spotted bars show the lateness for different policies as a percentile of the lateness obtained for the CPP. The lateness obtained in case of basestock and EDD is greater than 500% of that obtained for the CPP. This indicates that for the other policies, when the parts miss the due dates, they miss them by a large margin.

Another important measurement measure is the lead time achieved and its relation with the demand lead time ($A_1 - A_1' = 10$). As seen in Figure 6-22, the difference between the lead

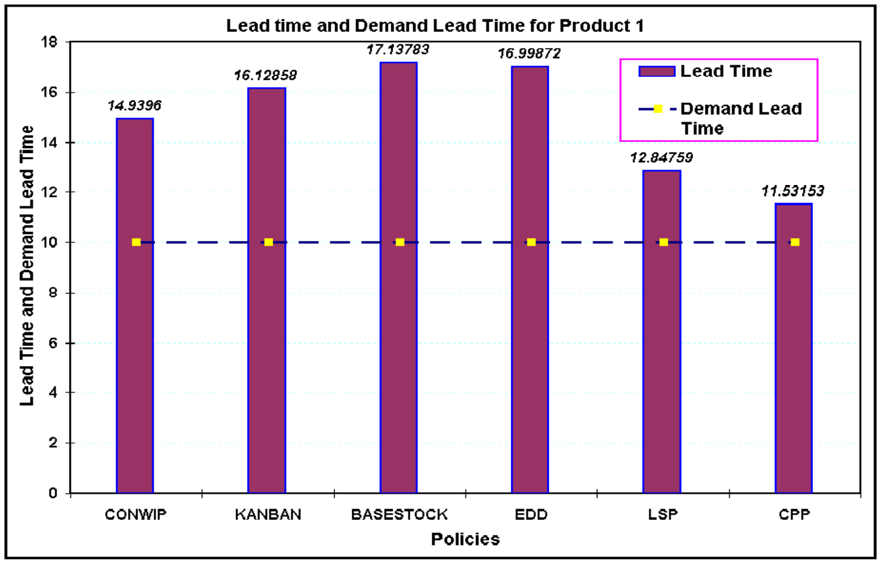


Figure 6-22: Lead time for *Product 1*

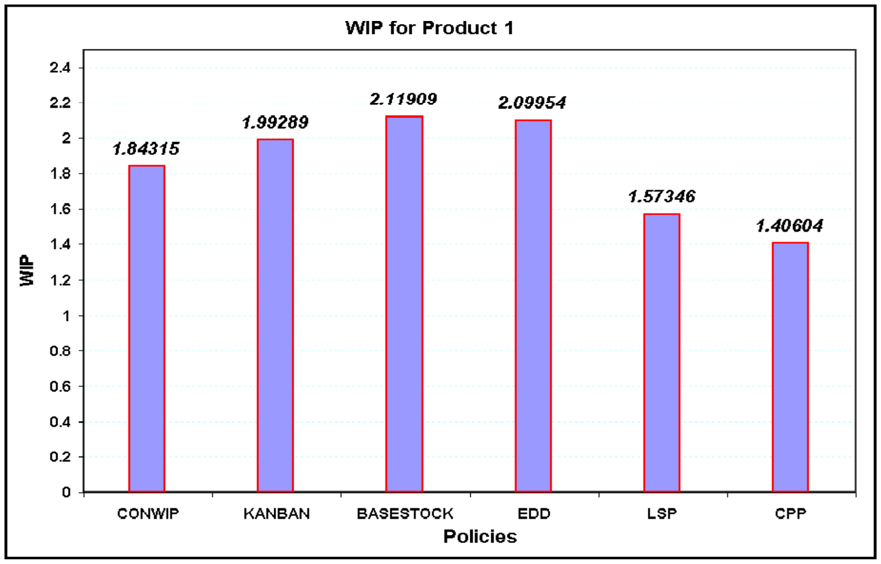


Figure 6-23: WIP for *Product 1*

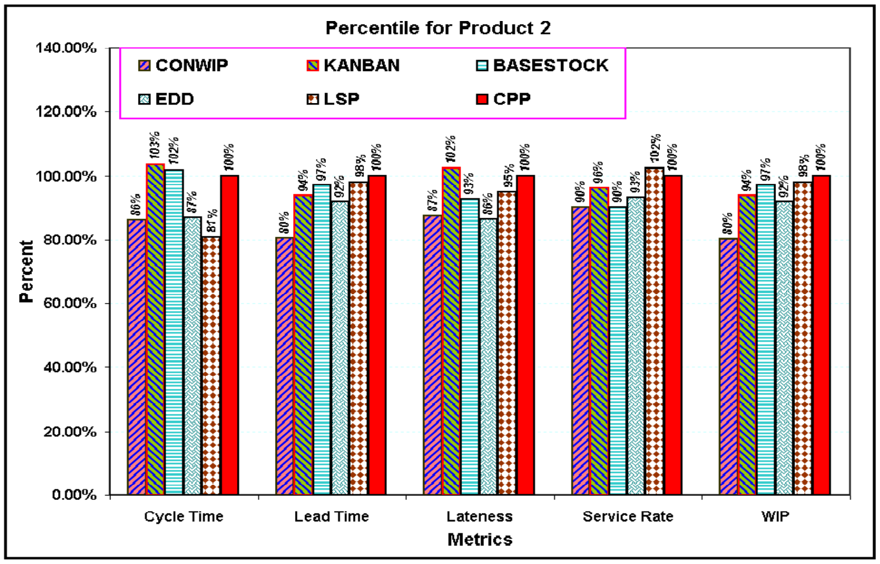


Figure 6-24: Percentile metrics for *Product 2*

time (Section 6.3 for definition) and the demand lead time (Section 6.3 for definition) is the minimum for the CPP. It means that, with the CPP, the company is able to achieve demand lead time quoted for the customers better than the others

Finally, the last metric, WIP (Section 6.3 for definition) for the Product 1 is compared in Figure 6-23. The WIP levels in all the policies are small. The WIP levels are the lowest for the CPP. Even small WIP improvements are significant for Product 1, because any in-process inventory means that it is not reaching the customer.

Figure 6-24 compares the percentile metrics for Product 2 in the above system. All the metrics are measured as a percentile of the metrics for the CPP. The metrics are roughly the same for all the policies, the maximum deviation is 20% from the CPP for any of the metrics. In other words, the performance obtained by the CPP is never the best, but never far away from the best. The CPP performance is especially very good for the two important metrics: the service rate and lateness are within 10% of the best performance. In both cases, there are policies which perform worse than the CPP.

6.9 Results

Thus, the CPP out-performs all other policies by a large margin for Product 1, while its performance in case of Product 2 is reasonably close to that of the other policies. By assigning a high priority to Product 1, and by re-sequencing parts at the machines, the policy is able to process them in the correct order.

6.10 Conclusions

Following are the conclusions of the simulation studies. They also include information on the status of current research and directions for further research.

1. **Demand lead time**

   Experiment 1 investigates the effect of the demand lead time $(A - A')$ on the performance metrics. The demand lead time has a major impact on the performance of the policy (for that matter on any time-based policy). Therefore, future research for the CPP should start with calculating the optimal demand lead times for the system. There is a lot of research available in the area. [Glasserman and Wang, 1998] discusses the trade-offs between the demand lead time and the inventory for an assemble-to-order system. [Duenyas and Hopp, 1995] and [Spearman and Zhang, 1999] study the importance and develop models for deciding optimal lead times. [Gong et al., 1994] studies the planning of optimal lead times in serial production systems.

2. **Order and release process**

   As we found in Experiment 4, the process of controlling the buffer B_0 is critical. There are significant improvements which can be achieved, if it is controlled correctly. There is a host of literature which discusses the optimal order and release process. [Song and Zipkin, 1996] studies the effect of supply conditions on WIP control. [Bergamaschi et al., 1997] studies the order release process for a job-shop environment, and [Tetzlaff, 1998] studies its effect for asynchronous production systems. [De'Mello et al., 1999] use simulation techniques for studying the optimal release times. Finally, [Hariharan and Zipkin, 1995] study the relation between the order process, the demand lead time and inventory in the system.

3. **Effect of buffer size**

In most of the simulation experiments, the blocking logic of the CPP was redundant i.e. the buffers were not performing the part of blocking parts effectively (as required by the policy). The role played by the finite and homogenous buffers is very important, especially in case of multiple part-type systems.

The results of [Schor, 1995] need to be extended to multiple part-types for that. The decomposition models for the multiple part-types studied by [Nemec, 1999] can be used for this purpose. Also, [So and Pinault, 1988] studies the allocation of buffer storages in a pull system.

4. **Hedging times**

Prior to the simulation studies, we had hoped to derive some approximate expressions for the hedging times in the system. However, the behavior of the hedging times depends on many parameters in the system. The effect of the demand lead time and the hedging times at the upstream machines were realized during the course of the experiments. The *buffer-transfer effect* of the hedging times was also observed.

We also struggled with the definition of an *"optimal"* hedging time. Should it be the time which minimizes the probability of starving or blockage for the machine? Or should it be a function of the average buffer levels (using the *buffer-transfer effect*)?

5. **Interdependence of hedging times**

The results of Experiment 5 need to be studied further. [Gershwin, 1992] can serve as a starting-point for the calculation.

6. **Resource contention**

During the study of *two part-type* systems and the comparison studies, we found there was a limit to the best performance that could be achieved for the second ranked part-type in the system. Beyond a certain limit, increasing its hedging times and the buffer sizes did not make improvements in its service rate and lateness. This effect was exaggerated when the hedging times and buffer sizes are large for the first ranked part-type. It indicates that with a fixed ranking, the capacity (and the variability) as seen by the second (and thus the third and so forth) part-types is different. Therefore, the effect of resource contention needs to be investigated further for multiple part-types.

This chapter serves as a pointer to further theoretical research for the CPP. It does not answer many questions, but it does raise many important questions. The next chapter of the document concludes the thesis.

Chapter 7

Conclusions

This thesis proposes a new scheduling policy for a factory. The policy comes in all the three versions discussed; the time-based version is treated in detail in this document. There are three main divisions associated with the material presented in this document: the theoretical research of the control point policy, its implementability in a factory, and the implications for the Lean Aerospace Initiative (LAI). All three of them are important, and the issues associated with each of them are diverse. In this chapter, we provide the important conclusions and directions for further research for each of them.

7.1 Conclusions for CPP theoretical research

This is the first time research has dealt with a time-based version of the hedging point theory. All previous research has been focussed on the token-based or the surplus-based policies. The only theoretical treatment of the policy are the simulation results presented in Chapter 6. Introducing the element of "time" in scheduling raises many important issues which are not completely understood at this incipient stage of research. The conclusions of the same chapter (Section 6.10) provide detailed results and directions for further research. Here, we briefly outline the major issues for further research.

1. It is premature to say that the three versions of the policy are identical and that the only difference is in the way each of them are implemented. We observe the following:

a. The three policies can be identical only if the demand process and the due date allocation process in the system are deterministic. If there is a variability in either of the processes, then the three versions are different. The non-satisfaction of the Equation 6.6 (Section 6.5.5) indicates this. This needs to be investigated further.

b. The second issue is in the applicability to different manufacturing environments. We claim that the time-based version of the policy is more suitable for the make-to-order systems, while the token-based version is suitable only for make-to-stock systems. Preliminary discussions[1] confirm this claim. However, this needs to be verified. The difference for the two manufacturing environments lies in the information necessary for the scheduling. [Karaesmen et al., 1999] discuss tokens for d_n with a vector of information (t_n, q_n, δ_n), which can be used as a token system for scheduling make-to-order systems.

2. The importance of the release process and the due date allocation process is very important for the performance of the control point policy (or any time-based policy). It is impossible to isolate the scheduling inside the factory from these factors and vice-versa. Therefore, further research in these areas needs to be unified.

3. The simulation studies are a prelude to the further research for the policy; they are not as comprehensive as the decomposition techniques developed in [Gershwin, 1994]. Readers may refer to [Gershwin, 1999] for the details on further research in the decomposition techniques for the token-based version of the CPP. However, the simulations studies provide very interesting insights into the performance of the policy. More sophisticated simulation techniques are needed for a detailed analysis. The simulation program used here, though very flexible, has some shortcomings[2]. The future use of appropriate simulation techniques needs to be decided.

4. An additional concept which came up during the experiment with Boeing was the selection of control points: their location and number. This new feature needs to be added to the

[1]With Dr. Stanley Gershwin (*MIT*), Dr. Yves Dallery (*Université Pierre et Marie Curie*) and Gary Collison (*Boeing, Portland*).

[2]Readers may refer to *Appendix B* for an assessment of the simulation program.

further research discussed in [Gershwin, 1999].

7.2 Conclusions for CPP implementation

The Boeing experiment is the first time the policy was implemented in a factory. Many new issues were realized during the course of the experiment, for example, the selection of control points. The conclusions of the experiment (Section 5.10) cover the details of the achievement and lessons from the experiment. In this section, we describe the important issues for future implementations of the policy.

1. The conclusions in Section 5.10 and the discussions of Chapter 4 are heavily influenced by the observations of the Boeing experiment. They may not apply to other factories. More complete observations can be made by getting feedback from different types of factories. The only way to achieve this is by conducting similar experiments at other factory sites.

2. At this point in time, we can present only broad guidelines for selecting the parameters of the policy. We rely on the user's knowledge and intuition about the factory system for selecting the parameters. This approach, more often than not, is the best way for conjecturing the accurate parameters for the system. The feedback for the policy (*Appendix A*) indicates that most users see the CPP as a very good tool for inventory reduction and smoother flows in the factory. At the same time, the top ranking complaint about the policy is the need for more precise estimates for the hedging times and the buffer sizes. Since the optimization techniques for these parameters are going to take some time[3], more elaborate rules for conjecturing the parameters need to be developed in the mean time.

3. Given that it will take some time before a totally independent software for the CPP is developed, attention needs to be paid to how it can easily incorporated into a company's existing system. To a large extent, the experiment with Boeing was helped by the availability of expert computer programmers in the company. They were easily able to translate the CPP algorithm into computer code. The *subscription mode* (Section 4.3.2)

[3]Depending on how further research in this area progresses.

seems to be a very convenient architecture. If possible a generic system-independent program should be written which can be used in all future experiments.

4. Our interaction with the MRP II system in Boeing and the literature conclude that the criticism of MRP II as a scheduling system is unjust to a certain extent. In many cases, it is the only scheduling system which can be used in the factory. Improvements, though not adequate (Section 2.7.4), have been made to correct the flawed assumptions associated with the original MRP system. Improper use and user indiscipline exacerbates the shortcomings in the system to a large extent.

7.3 Conclusions for LAI

Most of the companies affiliated with the LAI program have activities in the defence, aeronautics or space industry. The typical characteristics for a company in one of these industries are as follows:

1. **Very complex final products**: The bill of material (BOM) for the company has many levels and thousands of components. Each of the components is procured or manufactured from sources which are spread over very long distances and times.

2. **Very small number of orders**: The requirements for each order are, in general, unique and customer-specific. The longest series rarely exceeds more than a hundred units per product and the orders arrive at a very low rate. Even if it is the same product, engineering changes occur very frequently. They are imposed either by the customers or by new technologies.

3. **Very long and uncertain lead times**: The manufacturing leads times for the products and, thus, the demand lead times quoted to the customer are very long (up to years) and uncertain. Frequent engineering changes, coordination of many uncertain activities, and new and unreliable processes contribute to the problems.

4. **Very complex production process**: In addition to very technologically intensive material and manufacturing requirements, the process flows are very complex. Also the processes themselves are long and failure-prone.

In this respect, most of the factories in these industries lie in the right half of the spectrum for manufacturing system environments (Section 2.5). All of them lie close to a make-to-order system, while some (space related industries) can, in fact, be classified as engineer-to-order systems. Scheduling in these types of systems is the most difficult task [Hatchuel et al., 1997]; mainly because of uncertain manufacturing lead times and complex end-products. Also, simple binary token systems can not provide sufficient information for the scheduling of these systems. Sophisticated time-based policies are needed.

Most of the factories in these industries use MRP II for the coordination of production. Unfortunately, the system operates, in most cases, as a complexity control tool for the management of the component database, rather than as a production scheduling tool. Its assumptions about infinite capacity and fixed lead times are exactly contrary to the typical characteristics of the factories. Therefore, most of the factories are ridden with the symptoms of poor scheduling discussed in Section 2.3. The factory managers begin to set exaggerated due dates in order to have enough time to finish the production in time. Since the BOM consists of many assembly items, it results in very long lead times for the end-product. One can easily imagine the resulting bad performance in such a system.

The companies in these industries need to look at more sophisticated scheduling techniques to achieve lean improvements. Another feature, due to the highly capital intensive nature of products and processes, is that it is very difficult to affect radical changes in this type of an environment. The control point policy, proposed in this document, is a possible simple approach to achieve better scheduling. Though it is not a complete solution as yet, it does provide a very promising framework to achieve lean improvements for a factory.

Appendix A

CPP Survey

A.1 General questions

1. Have you observed any problems with MRP?

A. Inaccurate data?

B. Inflexible?

C. Actual shop-floor information different?

D. Not responsive to shop-floor improvement?

E. Can not tolerate the variability?

F. any other?

Please rank them if possible.

USER 1

A. Yes, but it is correctable.

B. No.

C. It's close.

D. You have to devise ways to go around it sometimes, but usually you can come up with a way to use both.

E. Learn to deal with it.

F. If you understand how MRP works, you can deal with it.

USER 2

Our problems with MRP are really a manifestation of lack of reliable methods on the shop-floor and poor understanding of the system.

USER 3

I. A

II. C

III. D

IV. B

V. E

VI. If parts are not logged into the system the list will be inaccurate.

USER 4

Yes, not very accurate.

USER 5

I. A

II. B

I have limited experience with the MRP system here.

USER 6

Though there is inaccuracy of data problems, the system is not inflexible. Other problems faced are that it is not responsive to changing shop-floor situations and problem of variability tolerance.

USER 7

MRP is not a flexible system when working to improve the process flow. The system has a lot of maintenance and usually the changes on the floor are not corrected in the MRP system to show true schedule position. All high in the rankings.

2. **Has there been an attempt towards establishing a cellular flow in the factory? YES or NO? Please describe.**

USER 1

Yep, but you generally have to go to a visual system inside the cell.

USER 2

Yes, ignorant people here move machines 'close' together in an attempt to create a cell. As you know a 'cell' is much more than machines proximity.

USER 3

Yes, by keeping all the processes in the same area (if all shot run in same area).

USER 4

Yes,

USER 5

No. This is limited to the time I have been here (approximately: 1 year)

USER 6

Yes, factory was originally designed in a cellular fashion. Some work in assembly.

USER 7

Yes. The 777 factory floor was setup for a cellular flow. The parts are introduced into the machining area on a set date on a CONWIP system. The flow was given the right amount of buffers and true flow times were adjusted so that the parts flow through the system to meet the rate demand at the end item ship.

3. **Has there been an attempt to establish single-part flow in the factory? YES or NO? Please describe.**

USER 1

Yes, for some parts that are manufactured in a way that it is possible due to the machines that they are made on. However, there is always a problem with batching in some areas.

USER 2

Yes. Our business team runs single part shop order for all our parts.

USER 3

Yes, covering system has been put in place in 777-flap support area.

USER 4

Yes, Machining has single part orders now.

USER 5

Yes, our raw material is delivered in 'single ships' set quantities. Parts are then controlled and tracked with individual orders from initial cut until delivery.

USER 6

Yes, done in order quantity. We use sequencing. Shot peen and plating have capacity constraints and need batches.

USER 7

99% of all parts are single part orders. There are a few parts that are ran outside of our flow that has different lot sizes due to the different metals and machine loads. Also the use of carts that control the amount of parts in the assembly area, etc.

4. **Have there been efforts to reduce the setup-time and setup-costs? YES or NO? Please describe.**

USER 1

Always, some of them actually worked

USER 2

Yes, a few but not many. Our CNC mills are off-line set-ups, (i.e. set up the job while another is running).

USER 3

Yes, we have had ongoing efforts to reduce set up time and costs.

USER 4

Yes, running ship-set quantities.

USER 5

No. Again this is limited to the year I have been here.

USER 6

Yes, but not to any great degree in the traditional SHINGO manner as most machines have external set-up.

USER 7

Yes. There has been new fixture designed, programming changes, fixtures designed on the boring mills to run both even and odd dash numbers on the parts, etc.

5. Are there any advantages in having finite buffers at all/few places in the factory? YES or NO? Please describe.

USER 1

You bet. It can help with quality problems or batching. Now figuring out where they should be and what the quantities should be is the hard part.

USER 2

No, I don't believe there is. MRP feeds our production system at the same rate as that our customer uses our finished goods. A more critical factor is through put velocity. If you introduce the parts into the system at the rate your customer uses them, the only problem that remains is ensuring the parts flow through the system at the required speed.

USER 3

Yes, it allows you to see what you need and if you are in need of something.

USER 4

Yes.

USER 5

Yes.

a. Reduces the cost of production inventory.

b. Minimizes the impact of defects.

c. Can illustrate excess flows and inventory.

USER 6

Due to FIFO sequencing, buffers reduce probability of improper inventory as well.

USER 7

Yes. Through out the flow of the product on the factory floor there are certain cost centers that the 777 competes for machine times. These buffers allow for the scheduling of the machines and any machine down times that may happen.

6. Have there been efforts to reduce the lead times in the factory? YES or NO? Please describe.

USER 1

In some areas, but it never worked. It was never done very well or thought out completely, it's depressing.

USER 2

Yes, very many. But reducing flow times in a shop with poor quality doesn't do any good.

USER 3

Yes, by taking out extra time in flows after following parts through the process.

USER 4

Yes, setting up buffers.

USER 5

No. Again this is limited to the year I have been here.

USER 6

Yes, 60% reduction.

USER 7

Yes. The shop has taken steps to reduce lead times at the front end of the parts flow. The use of the CONWIP allows the material to be delivered on time and with the right amount of buffers reduces the lead time requirements.

A.2 Before CPP implementation data

In your opinion, what were the major problems associated with the factory before the CPP implementation was tried?

7. Excess Inventory? YES or NO? Please describe.

USER 1

Yeah, and excess WIP.

USER 2

No, we have reduced our flow times and increased our through put to the point that excess inventory on the shop floor is no longer an issue.

USER 3

No, we were running what we needed but not watching the parts in stores. Sometimes the stores would have more than needed of one thing and none of another.

USER 4

Structure was not there but is great now.

USER 5

Yes.

USER 6

Yes, all parts regardless of costs move at the same velocity.

USER 7

All parts issued into the machining area are one part orders. The break down comes when the parts flow through other cost centers that compete for manpower and machine time. The part tend to get lumped together as like part numbers. This causes the wrong parts to be received into the end item stores. A lot of one part-number/and shortages of another

8. Lack of knowledge of the location of inventory? YES or NO? Please describe.

USER 1

Yes. Clueless.

USER 2

No, MRP does a great job of giving us visibility.

USER 3

No, most know where the inventory is located.

USER 4

Not much of a problem.

USER 5

Yes, since no definitive tools were available to identify the location.

USER 6

No.

USER 7

Yes. We knew that the shop had a certain amount of inventory on the shop floor because of the introduction of CONWIP, knowing the flow times and the buffers, but the location of the inventory was not always known.

9. Due date performance not good? YES or NO? Please describe.

USER 1

Not only bad, but also usually the wrong due date too.

USER 2

Yes, we use a FIFO system of prioritizing work at the various machines. This system has no correlation to the due date, if there is any hick up in the part flow.

USER 3

Yes, depends on the product. Flap supports were pretty much on schedule. It was the parts not put into CPP that got over-looked.

USER 4

Yes. Always been difficult.

USER 5

No.

USER 6

Yes, but predominantly this is/was an issue of motivation and management behavior/performance.

USER 7

Yes. Getting the right parts in the right sequence does effect the delivery dates. You will always have to go up stream

in your product flow to ship on time.

10. Manual monitoring/scheduling of jobs needed? YES or NO? Please describe.

USER 1

Yes. Everybody was an expediter.

USER 2

Yes, before FIFO we used MRP dispatch lists, with very poor results. The MRP dispatch lists was not a bad tool, we just used it incorrectly, which exacerbated the fundamental shortcomings of MRP.

USER 3

Yes, parts in short run were run in batches and mixed loads. What could be run was run without worrying about inventory stock.

USER 4

Visual control board - very simple.

USER 5

Yes, shop supervisor in our assembly area would chase specific parts to accelerate schedule.

USER 6

Yes, User 7 spends a massive amount of time expediting.

USER 7

Yes. All parts are currently monitored to deliver the product to the customer on time, but there is extra work when you have shortages because the shops don't work the correct part in the right sequence.

11. **Part availability i.e. Were parts always available when needed? YES or NO? Please describe.**

USER 1

Yeah, sure[1]. That's why we keep changing things.

USER 2

Yes, FIFO needs correlation to due dates more often.

USER 3

Yes, sometimes not what was needed, but there were always parts.

USER 4

No, not most of the time.

USER 5

No.

[1] That's sarcasm for you.

USER 6

No, some variability reflected in levels.

USER 7

No. Parts were not always available.

12. **Sensitivity to crisis i.e. How much time and effort did it take to respond to a major production disruption? Please describe.**

USER 1

Things happened right away, everybody runs around like crazy and over reacts.

USER 2

NA.

USER 3

Depends on the part and how far along in the process it was. It could be a day to three days.

USER 4

NA.

USER 5

From my perspective, our shop supervisor wouldn't spend much more time on a major disruption, as he would for normal production. However, to qualify this, our supervisor devotes significant amount of time to support normal production.

USER 6

Significant due to lack of information.

USER 7

This is hard to quantify. A lot.

13. **MRP dispatch list accuracy i.e. Did you follow the sequence given by the MRP? YES or NO? Please describe. (If NO, what were the calculations you performed in your mind?):**

USER 1

Yes, if MRP is used correctly, it's a good tool.

USER 2

Yes, but it constantly (twice per day) changed.

USER 3

No, in shot seen batches were run. If your first part could be run with others, it didn't matter what sequence it was at, just so you could get a full load.

USER 4

No, never.

USER 5

MRP dispatch lists is not used in production control. Production was typically governed by either a visual control, FIFO system or assembly supervisor.

USER 6

No. FIFO.

USER 7

No. By knowing the end item demand and parts that have a history of being late or maybe high rejection rates.

14. Was there a need for manual expediting? YES or NO? Please describe.

USER 1

Yes, due to the nature of our system, or lack of a system in certain areas it's needed. Some of the people aren't very well disciplined

USER 2

Yes, we constantly expedite because parts flow erratically.

USER 3

Yes, parts that were behind or to push parts that had been scrapped.

USER 4

Yes, always.

USER 5

Yes.

USER 6

Once. In assigning sequences.

USER 7

Yes. Manual expedite was required to help schedule priority work through cost centers that have other products in them.

A.3 Post CPP impressions

Were you able to detect (or see the scope for) any improvements in the parameters?

15. Reduction of Inventory? YES or NO? Please describe.

USER 1

Things became more visible so unneeded inventory stood out.

USER 2

No. Refer to response 7.

USER 3

Yes, when parts are flowing properly. You do not have the bottlenecks that you would get when there are more parts then there should be in the system.

USER 4

Yes, 2 ship-set buffer.

USER 5

Yes, by tracking the parts that are held by the CPP, and the duration of time they are held, I felt we could confidently reduce inventory.

USER 6

No, policy wasn't utilized long enough.

USER 7

No. Due to the changes on the factory floor, there was no reduction in the work in process. But I know the tool would drive out flow time improvements and reduction of inventory.

16. More knowledge on the location of inventory? YES or NO? Please describe.

USER 1

Yes because we had to think all of this through to implement CPP

USER 2

No, it doesn't get better than our MRP system.

USER 3

Yes, you would be able to see if you were in need of a certain part and would make you locate it.

USER 4

Yes, visual controls.

USER 5

Yes, the CPP would quantitatively identify where the access inventory is.

USER 6

Yes, when used.

USER 7

Yes. During the test period.

17. Improved due date performance? YES or NO? Please describe.

USER 1

Yeah, it worked

USER 2

NA.

USER 3

Yes, by getting the details worked that were needed.

USER 4

Yes, very much so.

USER 5

Yes, the CPP would help assure that only the parts that are needed would be worked on. Thereby, freeing up machinery to work on parts that support an end item.

USER 6

Refer to 15.

USER 7

No. Process was tested for a short period of time. Parts did flow through the shot peen area. Didn't have time to change flow times and buffers.

18. Lesser need for manual monitoring/scheduling of jobs? YES or NO? Please describe.

USER 1

It made things really simple during times of large batches of work going through the area. Everybody looked at the same information.

USER 2

No.

USER 3

Yes, I would believe so.

USER 4

Yes.

USER 5

I don't think I can answer this question yet with the limited experience I have had.

USER 6

Refer to 15.

USER 7

Yes. During the test period.

19. Part availability improvement? YES or NO? Please describe.

USER 1

More like the right parts should be there on time.

USER 2

NA.

USER 3

Yes, by making parts needed and not the parts you already have.

USER 4

Yes.

USER 5

Yes.

USER 6

Refer to 15.

USER 7

Yes. During test period.

20. Reduction in sensitivity to crisis? YES or NO? Please describe.

USER 1

Well, this place live for crisis, we'll see.

USER 2

NA.

USER 3

No, we always have crisis.

USER 4

Yes.

USER 5

Yes.

USER 6

Yes, it gave shot peen a tool.

USER 7

Yes. Except for machine down time. Parts then are transported to the other building for processing. All expediting.

21. More accuracy over the MRP dispatch lists? YES or NO? Please describe.

USER 1

Yes, the MRP dispatch lists would hop around from day to day too much and put less expensive parts ahead.

USER 2

NA.

USER 3

No, In order for CPP to work you have to log the parts in the shop. All the same.

USER 4

Yes.

USER 5

Yes.

USER 6

Yes, recognized factory status.

USER 7

Yes. True requirements. MRP dispatch lists has not been maintained in the 777 flap support area for a few years.

22. Reduction in need for manual expediting? YES or NO? Please describe.

USER 1

A reduction, but not an elimination.

USER 2

NA.

USER 3

No, someone is always looking to see where parts are not to expedite but more to follow part that have been reworked or parts that have been made to replace other parts.

USER 4

Yes, most of the time.

USER 5

Yes.

USER 6

Refer to 15.

USER 7

Yes. Operators know what to work.

23. Where do you see the best utility of the CPP? Please rank them.

A. As a re-sequencing mechanism?

B. As a means of making decisions more transparent and logical to the shop-floor personal?

C. As a tool to reduce inventory?

D. As a crisis solver?

E. As a means to improve the flow-times through the factory?

USER 1

I. B

II. A

III. D

IV. E

V. C

USER 2

I. C

II. A

USER 3

I. A

II. E

USER 4

I. E - As a means to improve the flow-times through the factory. Visual Controls: Boards and FIFO.

USER 5

I. C

II. A

III. E

IV. B

V. D

USER 6

I. A

II. C

III. E

USER 7

I. E

II. C

III. A

IV. B

V. D

24. What are your complaints with the CPP? Please rank them.

A. Not very different from the MRP dispatch list?

B. Hedging times and buffers sizes used need to be more accurate?

C. Ranking needs to be based differently?

D. Difficult to have finite and homogenous buffers inside the factory?

E. Any other?

USER 1

I. B

II. C

III. D

IV. A

V. Big ugly Process Engineers

USER 2

My only problem with CPP if that it is an optimization tool. We are far from needing refinement and optimization. We still haven't done the basics.

USER 3

We run many details for different airplane models. If you have only one product on CPP the rest of the details get over-looked. All of the details of all the models need to be on the same system in order to work properly and effectively. IF not, then you are better off with a MRP dispatch lists.

USER 4

I. D - Difficult to have finite and homogenous buffers inside the factory.

USER 5

CPP requires very specific buffers. However, this is fairly easy to change, and does force the factory to monitor its buffers and change them accordingly.

USER 6

II. B - Hedging times and buffers sizes used need.

USER 7

After watching the start up of the test. I feel that the biggest change has to come from the factory floor and to adhere to the list and buffers that are put into the system. The common tendency is to run parts if they are in the shops, regardless of the schedule. I have no complaints about CPP. Although we struggled for the start up and test, a longer test period would have prepared me in giving a better response.

Appendix B

Critique on SIMUL8 software

The SIMUL8[1] software program was used for the simulation studies of Chapter 6. This was the first time a commercial software was used to do simulations; all previous studies were done using in-house unix-based simulation codes. Here, we present an appraisal of the SIMUL8 program.

B.1 Pros

1. Available on MS Windows platform. Can be interfaced with MS Excel.
2. External code modules can be added to the software program. Thus, it can be customized.
3. Very convenient GUI interface, which allows the user to create different types of manufacturing systems, flows and processes easily.
4. The consultant team is ready to help with the problems. They are also ready to help you with new versions and ideas.

B.2 Cons

1. Available on MS Windows platform. Therefore, the usual problems with MS Windows persist.

[1]Copyright ©1998 Visual Thinking Int. Ltd.

2. Lot slower and bigger in memory-size as compared to the unix-based simulations.

3. The software language is *very* week. Coding and debugging the software modules is a very *painstaking* task.

4. Another difficulty is with data collection. You cannot collect all the data from the run and then analyze it at the end. Internal spread sheets, which are used for the data collection during the run, are only of a limited size. Therefore, the external modules have to see that the sizes of the spread sheets do not exceed this limit. Else the program crashes. This has to be done at real-time (during the simulation run). This makes the program very slow. Also, the external modules need to do the averaging and counting etc. during the run.

5. During a simulation run, the program slows down all other applications on the computer. Therefore, the computer cannot be used for any other purpose when the simulation is running.

6. The simulations cannot be distributed over the NT network, like unix-based simulations. You have to physically start it at different NT servers.

The software can be useful depending on the type of simulations being conducted. If there are complex routings and processes through the factory, then the program is very useful. However, if simple manufacturing lines need to be simulated, then the unix-based simulations are faster and more efficient.

There are a lot of ongoing improvements in the software. However, the first improvement necessary is to make the coding language more powerful. Also, an advanced debugger and coding interface is needed. Currently, these problems make the software “inadequately equipped” for the requirements of the kind of simulations performed in Chapter 6. If these improvements are done, then the software will be of great utility for future simulations. Please contact Mark Elder <mark.e@VisualT.com> for the latest changes in the software. He is the CEO of the company and is very helpful.

Appendix C

Details of strategy for each policy in Section 6.7

We describe **Step 1** and **Step 2** and **Step 3b** for each policy in the strategy used for the comparison studies in Section 6.7. We conjecture that these step are the most-logical ways for solving the problem, other methods may be possible.

1. **KANBAN**

Step 1. Keep the buffer sizes i.e. number of KANBAN tokens for *Product 1* at 2 at each stage. Vary the number of KANBAN tokens *uniformly* for Product 2 at each stage. Get the best possible performance for *Product 2*. The service rate for Product 2 stops improving beyond a certain number of KANBAN tokens. Choose this value for Product 2.

Step 2. Keep this number of KANBAN tokens for *Product 2* constant. Vary the number of KANBAN tokens *uniformly* for *Product 1* till you get the best possible performance.

Step 3b. Change the number of KANBAN tokens for Product 1.

2. **Basestock**

No strategy required. Allow infinite backlogging for both products.

3. **CONWIP**

Step 1. Keep the size of CONWIP loop for *Product 1* at 3[1] at each stage. Vary the size of CONWIP loop for Product 2. Get the best possible performance for Product 2. The service rate for Product 2 stops improving beyond a certain size of the CONWIP loop. Choose this value for Product 2.

Step 2. Keep this size of the CONWIP loop for *Product 2* constant. Vary the size of the CONWIP loop for *Product 1* till you get the best possible performance.

Step 3b. Change the size of the CONWIP loop for Product 1.

4. **EDD**

No strategy required.

5. **LSP**

Step 1. Keep the *time remaining in the system* (ζ_{s1}) for *Product 1* at each stage s as simply the sum of the remaining processing times downstream of the stage. Vary time remaining in the system (ζ_{s2}) for Product 2 at each stage s : Start with sum of the remaining processing times downstream of the stage, and step-up each of them by a constant value. Get best possible performance for Product 2.

Step 2. Keep these ζ_{s2} for Product 2 constant. Step-up ζ_{s1} for *Product 1* by a uniform constant till you get the best possible performance.

Step 3b. Change ζ_{s1} for Product 1.

6. **CPP**

Step 1. All the machines are control points. Give higher ranking to Product 1. All buffers in the system are constant at size 10 each. Keep the hedging times for *Product 1* at each stage s ($HT_{s1} = 1$). Vary hedging times at each stage for Product 2 (HT_{s2}). Increase them uniformly up to ($A_2 - A'_2 = 20$). Get best possible performance for Product 2.

[1] That is the minimum possible size, any smaller size leads to an unstable system.

Step 2. Keep these hedging times for Product 2 constant. Vary the hedging times for *Product 1* (HT_{s1}) up to ($A_1 - A'_1 = 10$) *uniformly* till you get the best possible performance

Step 3b. Change hedging times HT_{s1} for *Product 1*.

Appendix D

Acknowledgments

First, and most important, I wish to thank my wonderful family. To my mom and dad, who never cease to encourage me to achieve everything I have desired. They are always there for me unconditionally; always ensuring that I get the best of everything in the world. To my sister, the simple and uncomplicated Bahar, the sweetest girl in the world. And not to forget my pet-dog, Goofy - he may be gone, but he still lives for me in all the vivid memories. To all of them, and the best home in the whole world - 4 Akar, this thesis is dedicated.

This thesis was supported by many sources. From MIT academia, primarily, I would like to thank my advisor Dr. Stanley Gershwin, who supported me technically, financially, and emotionally during my entire two-year stay at MIT. He has always been an un-ending source of ideas and encouragement. I will never forget his words of wisdom - be it in research, or in life and beyond. Stan is *just* Stan - there is no one else like him.

No word of thanks will suffice Gary Collison and his entire 777 Flap Support team at Boeing, Portland. I learned a lot of things from him during the course of the Boeing experiment: professionally, personally and fishing-wise. This thesis would not have been possible without the 777 Flap Support team.

I would like to thank Tom Shields and the Lean Aerospace Initiative for the financial support and the chance to work with Boeing. Tom's advice and guidance helped me keep things in perspective at all times.

Finally, I would like to thank all my friends. Starting with the India Club, all the members - you make me feel proud. I thank them for being such great friends, for sticking with me in all times, and for taking me the way I am. Also The Group - Anand, Diego, Joe, Omar, Young. They have always helped me beyond the ordinary. Thanks a lot - the Numero Uno position is up for grabs again.

Bibliography

[Akella and Kumar, 1986] Akella, R. and Kumar, P. R. (1986). Optimal control of production rate in a failure prone manufacturing system. *IEEE Transactions on Automatic Control*, AC-31(2):116–126.

[Bai, 1991] Bai, S. X. (1991). *Scheduling manufacturing systems with work-in-progress inventory control*. PhD thesis, Massachusetts Institute of Technology.

[Baker, 1984] Baker, K. R. (1984). The effects of input control in a simple scheduling model. *Journal of Operations Management*, 4(2):99–112.

[Baudin, 1990] Baudin, M. (1990). *Manufacturing Systems Analysis With Application To Production Scheduling*. Yourdon Press, Prentice Hall Building, Englewood Cliffs, New Jersey 07632.

[Bergamaschi et al., 1997] Bergamaschi, D., Cigolini, R., Perona, M., and Portioli, A. (1997). Order review and release strategies in a job shop environment: a review and classification. *International Journal of Production Research*, 35(2):339–420.

[Berkley, 1991] Berkley, B. J. (1991). Tandem queues and kanban-controlled lines. *International Journal of Production Research*, 29(10):2057–2081.

[Berkley, 1992] Berkley, B. J. (1992). A review of kanban production control research literature. *Production and Operations Management*, 1(4):393–411.

[Bertrand, 1983] Bertrand, J. W. M. (1983). The use of workload information to control job lateness in controlled and uncontrolled release production systems. *Journal of Operations Management*, 3(2):79–92.

[Bielecki and Kumar, 1988] Bielecki, T. and Kumar, P. R. (1988). Optimality of zero-inventory policies for unreliable manufacturing systems. *Operations Research*, 36(4):532–541.

[Blackstone et al., 1982] Blackstone, J. H., Phillips, J. H., and Hogg, G. L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, 20(1):27–45.

[Bonvik, 1996] Bonvik, A. M. (1996). *Performance analysis of manufacturing systems under hybrid control policies.* PhD thesis, Massachusetts Institute of Technology.

[Buzacott and Shanthikumar, 1993] Buzacott, J. A. and Shanthikumar, J. G. (1993). *Stochastic Models of Manufacturing Systems.* Prentice Hall, Englewood Cliffs, New Jersey.

[Cheng and Gupta, 1989] Cheng, T. C. E. and Gupta, M. C. (1989). Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research*, 38(1):156–166.

[De'Mello et al., 1999] De'Mello, T. H., Shapiro, A., and Spearman, M. L. (1999). Finding optimal material release times using simulation-based optimization. *Management Science*, 45(1):86–102.

[Dietrich, 1991] Dietrich, B. L. (1991). A taxonomy of discrete manufacturing systems. *Operations Research*, 39(6):886–902.

[Duenyas and Hopp, 1995] Duenyas, I. and Hopp, W. J. (1995). Quoting customer lead times. *Management Science*, 41(1):43–57.

[Fine, 1998] Fine, C. H. (1998). *Clockspeed : Winning Industry Control in the Age of Temporary Advantage.* Perseus Books.

[Gershwin, 1992] Gershwin, S. B. (1992). Variance of output of a tandem production system. In *Queuing Networks with Finite Capacity, Proceedings of the Second International Workshop held in Research Triangle Park, North Carolina, May 28-29, 1992.* Edited by R. Onvural and I. Akyldiz, Elsevier, 1993.

[Gershwin, 1994] Gershwin, S. B. (1994). *Manufacturing Systems Engineering.* Prentice Hall, Englewood Cliffs, New Jersey.

[Gershwin, 1997] Gershwin, S. B. (1997). Design and operation of manufacturing systems - control- and system- theoretical models and issues. pages 1909–1913.

[Gershwin, 1999] Gershwin, S. B. (1999). System analysis, design, and control: Unification and decomposition. In *Second Aegean International Conference On Analysis And Modeling Of Manufacturing Systems, Tinos Island, Greece.*

[Glasserman and Wang, 1998] Glasserman, P. and Wang, Y. (1998). Leadtime-inventory trade-offs in assemble-to-order systems. *Operations Research*, 46(6):858–871.

[Goldratt and Fox, 1984] Goldratt, E. and Fox, J. (1984). *The Goal.* North River Press, P.O. Box 567, Great Barrington, MA 01230.

[Goldratt, 1994] Goldratt, E. M. (1994). *It's Not Luck.* North River Press, P.O. Box 567, Great Barrington, MA 01230.

[Gong et al., 1994] Gong, L., Kok, T., and Ding, J. (1994). Optimal leadtimes planning in a serial production system. *Management Science*, 40(5):629–632.

[Gutzmann and Wysk, 1986] Gutzmann, K. M. and Wysk, R. A. (1986). Capacity control polcies for MRP systems. *International Journal of Production Research*, 24(2):359–374.

[Hariharan and Zipkin, 1995] Hariharan, R. and Zipkin, P. (1995). Customer-order information, leadtimes, and inventories. *Management Science*, 41(10):1599–1607.

[Hatchuel et al., 1997] Hatchuel, A., Saidi-Kabeche, D., and Sardas, J. C. (1997). Towards a new planning and scheduling apporach for multistage production systems. *International Journal of Production Research*, 35(3):867–886.

[Hendricks and Singhal, 1997] Hendricks, K. B. and Singhal, V. R. (1997). Delays in new product introductions and market value of the firm : The consequences of being late to the market. *Management Science*, 43(1):422–436.

[Hopp and Roof, 1998] Hopp, W. J. and Roof, M. L. (1998). Setting wip levels with statistical throughput control (stc) in conwip production lines. *International Journal of Production Research*, 36(4):867–882.

[Hopp and Spearman, 1996] Hopp, W. J. and Spearman, M. L. (1996). *Factory Physics: Foundations of Manufacturing Management*. Irwin.

[Karaesmen et al., 1999] Karaesmen, F., Buzacott, J. A., and Dallery, Y. (1999). Integrating advance information in pull type control mechanisms for multi-stage production. In *Second Aegean International Conference On Analysis And Modeling Of Manufacturing Systems, Tinos Island, Greece.*

[Kimball, 1988] Kimball, G. (1988). General principles of inventory control. *Journal of Manufacturing and Operations Management*, 1(1):119–130.

[Kimemia and Gershwin, 1983] Kimemia, J. G. and Gershwin, S. B. (1983). An algorithm for the computer control of production in a flexible manufacturing system. *IIE Transactions*, 15(4):353–362. Reprinted in Modeling and Control of Automated Manufacturing Systems by Alan A. Desrochers, IEEE Computer Society Press Tutorial, 1990.

[Liberopoulos and Dallery, 1998] Liberopoulos, G. and Dallery, Y. (1998). A unified framework for pull control mechanisms in multi-stage manufacturing systems. Technical Report LIP6-CNRS, Laboratoire d'Informatique de Paris.

[Little, 1961] Little, J. D. C. (1961). A proof of the queueing formula $L = \lambda W$. *Operations Research*, 9:383–387.

[Lou and Kager, 1989] Lou, S. X. C. and Kager, P. (1989). A robust production control policy for VLSI wafer fabrication. *IEEE Transaction on Semiconductor Manufacturing*, 2:159–164.

[Lou and Ryzin, 1989] Lou, S. X. C. and Ryzin, G. J. V. (1989). Optimal control rules for scheduling job shops. *Annals of O.R.*, 17:233–248.

[Lu and Kumar, 1991] Lu, S. H. and Kumar, P. R. (1991). Distributed scheduling based on due dates and buffer prioritization. *IEEE Transactions on Automatic Control*.

[Melynk and Ragatz, 1988] Melynk, S. A. and Ragatz, G. L. (1988). Order review/release and its impact on the shop floor. *Production and Inventory Management Journal*, 29(2):13–17.

[Merrle, 1997] Merrle, T. (1997). Emerging technology: Production scheduling. *IIE Solutions*, 29(1):24–29.

[Miltenburg, 1997] Miltenburg, J. (1997). Comparing JIT, MRP and TOC, and embedding TOC into MRP. *International Journal of Production Research*, 35(4):1147–1169.

[Monden, 1998] Monden, Y. (1998). *Toyota Production System : An Integrated Approach to Just-In-Time*. Engineering and Management Press.

[Nemec, 1999] Nemec, J. E. (1999). *Diffusion and decomposition approximations of stochastic models of multiclass processing networks*. PhD thesis, Massachusetts Institute of Technology.

[Pinedo, 1995] Pinedo, M. (1995). *Scheduling: Theory, Algorithms and Systems*. Prentice Hall, Englewood Cliffs, New Jersey 07632.

[Ragatz and Mabert, 1988] Ragatz, G. L. and Mabert, V. A. (1988). An evaluation of order release mechanisms in a job-shop environment. *Decision Sciences*, 19(1):167–189.

[Rishel, 1975] Rishel, R. (1975). Dynamic programming and minimum principles for systems with jump markov disturbances. *SIAM Journal on Control*, 13(2).

[Ryzin, 1987] Ryzin, G. J. V. (1987). Control of manufacturing systems with delay. Master's thesis, Massachusetts Institute of Technology.

[Ryzin et al., 1993] Ryzin, G. J. V., Lou, X. C., and Gershwin, S. B. (1993). Production control for a tandem two-machine system. *IIE Transactions*, 25(5):5–20.

[Salegna, 1996] Salegna, G. (1996). Integrating the planning and scheduling systems in a job shop. *Production and Inventory Management Journal*, 37(4):1–7.

[Schor, 1995] Schor, J. (1995). Efficient algorithms for buffer allocation. Master's thesis, Massachusetts Institute of Technology.

[Sethi and Zhang, 1994] Sethi, S. P. and Zhang, Q. (1994). *Hierarchical Decision Making in Stochastic Manufacturing Systems*. Birkhaeuser.

[So and Pinault, 1988] So, K. C. and Pinault, S. C. (1988). Allocating buffer storages in a pull system. *International Journal of Production Research*, 26(12):1959–1980.

[Song and Zipkin, 1996] Song, J. and Zipkin, P. H. (1996). Inventory control with information about the supply conditions. *Management Science*, 42(10):1409–1419.

[Spearman et al., 1990] Spearman, M. L., Woodruff, D., and Hopp, W. J. (1990). CONWIP: a pull alternative to kanban. *International Journal of Production Research*, 28(5):879–894.

[Spearman and Zhang, 1999] Spearman, M. L. and Zhang, R. Q. (1999). Optimal lead time policies. *Management Science*, 45(2):290–295.

[Srivatsan and Dallery, 1998] Srivatsan, N. and Dallery, Y. (1998). Partial characterization of optimal hedging point policies in unreliable two-part-type manufacturing systems. *Operations Research*, 46(1):36–45.

[Tabe et al., 1980] Tabe, T., Muramatsu, R., and Tanaka, Y. (1980). Analysis of production ordering quantities and inventory variations in a multi-stage production ordering system. *International Journal of Production Research*, 18(4):393–411.

[Tetzlaff, 1998] Tetzlaff, U. A. W. (1998). Optimal order release decisions for asynchronous production systems. *IIE Transactions*, 30(11):1001–1008.

[Tsai et al., 1997] Tsai, C. H., Chang, G. T., and Li, R. K. (1997). Integrating order release control with due-date assignment rules. *International Journal of Production Research*, 35(12):3379–3392.

[Turbide, 1995] Turbide, D. A. (1995). MRP II: Still number one! *IIE Solutions*, 27(7):28–31.

[Vollmann et al., 1992] Vollmann, T. E., Berry, W. L., and Whybark, D. C. (1992). *Manufacturing Planning and Control Systems*. Irvin, Burr Ridge, IL.

[Wagner and Whitin, 1958] Wagner, H. M. and Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96.

[Wein, 1988] Wein, L. M. (1988). Scheduling semiconductor wafer fabrication. *IEEE Transactions on semiconductor manufacturing*, 1(3):115–130.

[Wisner, 1995] Wisner, J. D. (1995). A review of order release policy research. *International Journal of Operations and Production Management*, 15(6):25–40.

[Yan et al., 1996] Yan, H., Lou, S. X. C., Sethi, S. P., Gardel, A., and Deosthali, P. (1996). Testing the robustness of two-boundary control policies in semiconductor manufacturing. *IEEE Transactions in Semiconductor Manufacturing*, 9(2):285–288.

[Yeung et al., 1998] Yeung, J. H. Y., Wong, W. C. K., and Ma, L. (1998). Parameters affecting the effectiveness of MRP systems: a review. *International Journal of Production Research*, 36(2):313–331.