1

PM-1 3 1/2"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

PRECISIONSM RESOLUTION TARGETS

PIONEERS IN METHYLENE BLUE TESTING SINCE 1974

1800 COUNTY ROAD 6, BURNING WOOD, OH 44027, USA
TEL: 812 426 7907 FAX: 812 426 7907 TUL: 812 426 7907

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file - Votre référence

Our file - Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

**A COHERENT PLANNING MODEL FOR THE DEVELOPMENT OF
INTEGRATED MATERIALS PLANS AND PRODUCTION
SCHEDULES**

by

PETER BISHOP, B. ENG.

A thesis submitted to the Faculty of Graduate Studies and
Research in partial fulfilment of the requirements for the
degree of

Master of Engineering

Department of Electronics

**Carleton University
Ottawa, Ontario
August, 1992**

**© Copyright 1992
Peter Bishop**

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Author: Autor responsable

Author: Auteur responsable

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-79784-3

Canada

The undersigned recommend to the Faculty of Graduate Studies and
Research acceptance of the thesis

**A COHERENT PLANNING MODEL FOR THE DEVELOPMENT OF
INTEGRATED MATERIALS PLANS AND PRODUCTION
SCHEDULES**

submitted by

PETER BISHOP, B. ENG.

in partial fulfilment of the requirements for the degree of Master of
Engineering

Thesis Supervisor

Thesis Supervisor

Carleton University
July, 1992

Acknowledgments

I would like to acknowledge the contributions of Dr. John Knight and Dr. John Chinneck who, through many hours of discussion and review, helped focus and direct this research.

Abstract

Many large job-shop manufacturing operations suffer from infeasibilities and cost inefficiencies in their production plans which arise from the use of separate materials planning and shop scheduling processes. The assumptions of infinite capacity to manufacture and infinite material availability used in the respective processes are often unrealistic.

This thesis presents an integrated approach to the development of materials plans and shop schedules which considers the actual manufacturing capacity and material availability in a plant. The simultaneous development of the production plans eliminates infeasibilities and reduces cost.

The algorithm is practical for existing manufacturing operations. A proof-of-concept prototype is presented which compares favourably against the results from a real manufacturing planning system.

Extensions to the base algorithm which extend its practical utility for the general manufacturing planning problem are also presented.

Table of Contents

1. Introduction	1-1
2. An Introduction to Materials & Production Planning Methods.....	2-1
2.1. Introduction	2-1
2.2. Inventories in a Manufacturing Environment	2-2
2.2.1. Manufacturing vs. Distribution Inventory	2-3
2.2.2. Multiechelon Inventory	2-3
2.3. Materials Planning Processes	2-4
2.3.1. Master Production Scheduling (MPS).....	2-4
2.3.2. Material Requirements Planning (MRP)	2-5
2.3.2.1. Introduction	2-5
2.3.2.2. Netting of Demand.....	2-6
2.3.2.3. Explosion.....	2-7
2.3.2.4. Level-by-level iteration.....	2-7
2.3.3. Capacity Requirements Planning (CRP)	2-10
2.4. Production Scheduling	2-13
3. A Review of Current Production Planning Algorithms and Research	3-1
3.1. Algorithms for Integrated Production Planning	3-1
3.1.1. Commercial Scheduling Products	3-1
3.1.2. Research Results.....	3-2
3.2. Materials Planning	3-4
3.2.1. Introduction.....	3-4
3.2.2. MRP.....	3-5
3.2.3. Introduction.....	3-5
3.2.4. Lot Sizing Techniques.....	3-6
3.3. Production Scheduling	3-13
3.3.1. Complexity of Scheduling	3-13
3.3.2. Analytic Approaches	3-14
3.3.2.1. Problems with Analytic/Optimum Seeking Approaches.....	3-18
3.3.3. Heuristic Techniques	3-20

3.3.4. Priority rules.....	3-22
3.3.4.1. Scheduling Heuristics.....	3-24
3.3.4.2. Heuristic Techniques vs. Optimum Seeking Methods...	3-26
3.3.5. Interactive	3-28
3.3.6. Artificial Intelligence and Knowledge Based Approaches.....	3-33
3.3.6.1. Introduction	3-33
3.3.6.2. Directions in AI Based Scheduling.....	3-33
3.5. Practical Implementations.....	3-35
3.6. Due Dates in Job Shop Scheduling.....	3-36
4. Problems with Existing Materials Planning & Production Scheduling Systems	4-1
4.1. Problems in Materials Planning.....	4-2
4.2. Problems in Shop Scheduling.....	4-4
4.3. Practical Implications of the MRP/Shop Schedule Skew	4-5
4.4. Objectives of this Research	4-7
4.5. Measures of Schedule Quality.....	4-8
5. Elements of an Algorithm for Creating Integrated Material Plans and Production Schedules	5-1
5.1. Introduction	5-1
5.2. A Conceptual Model for Coherent Production Planning.....	5-1
5.2.1. Time and Product Structure Traversal	5-1
5.2.2. Lead Time Estimation	5-6
5.2.3. Materials Planning.....	5-8
5.2.4. Shop Scheduling.....	5-11
5.2.5. Infeasibilities in Plan Synchronization and Due Dates	5-14
5.2.6. A Simple Example of Coherent Production Planning.....	5-14
5.2.7. Assessment of Schedule Quality	5-30
6. A Software Prototype.....	6-1
6.1. Introduction	6-1
6.2. Development Platform	6-1

6.3. The Software Structure	6-1
6.3.1. Comparison Baseline	6-1
6.3.1.1. Overview	6-1
6.3.1.2. Materials Planning in the Baseline Algorithm	6-4
6.3.1.3. Shop Scheduling in the Baseline Algorithm.	6-6
6.3.2. Coherent Planning Model	6-8
6.3.2.1. Overview	6-8
6.3.2.2. Materials Planning in the Coherent Model	6-11
6.3.2.3. Shop Scheduling in the Coherent Model	6-14
6.3.3. Results Analysis	6-16
6.3.4. Input Data Files	6-18
 7. Experimentation & Results.....	7-1
7.1. Source of the Data	7-1
7.2. Modifications in the Data	7-2
7.3. Results of Experimentation	7-4
7.3.1. Analysis of a Shop with a Bottleneck	7-4
7.3.2. Analysis of an Evenly Loaded Shop	7-8
7.3.3. Analysis of Modified Cost Criteria	7-11
7.3.4. Analysis of Modified Order Policies.....	7-13
7.4. Experimentation with Due Date Setting Algorithms.....	7-16
 8. Discussion.....	8-1
8.1. The Significance of Due Dates.....	8-1
8.2. Extensions for Practical Use.....	8-2
8.2.1. Inventories.....	8-2
8.2.2. Work in Process.....	8-3
8.2.3. Schedule Stabilization.....	8-4
8.2.4. Demand Pegging.....	8-4
8.3 Complexity of the Algorithm	8-5
 9. Summary	9-1

9.1. Benefits of the Algorithm.....	9-1
9.1.1. Reduction of Costs and Removal of Infeasibilities.....	9-1
9.1.2. Applicability to the General Industrial Problem.....	9-2
9.2. Conclusions	9-4
9.3. Directions for Future Research.....	9-5
9.4. Contributions	9-6
9.4.1. Summary of Contributions	9-6
10. References	10-1

Appendices

A1. Glossary of Terms	A.1
A2. Sample Problems from Chapter 3	A.6
A3. Heuristic Performance Measures.....	A.8
A4. Priority Rule Definitions.....	A.9
A5. Dataset #1 from Chapter 7	A.11
A6. Dataset #2 from Chapter 7	A.12
A7. Dataset #3 from Chapter 7	A.13
A8. Dataset #4 from Chapter 7	A.14
A9. Due Date Experimentation Data from Chapter 7	A.15
A10.Speed of Computation	A.17

List of Tables

Table 3-1: Comparison of optimizing shop scheduling algorithms.	3-16
Table 5-1: Step 1 Materials Plan	5-16
Table 5-2: Step 2 Materials Plan	5-16
Table 5-3: Steps 3 & 4 Materials Plan.....	5-17
Table 5-4: Jobs queued after step 4	5-18
Tables 5-6, 7, 8: Queued jobs after scheduling in step 4.....	5-19
Table 5-9: Materials Plan Step 5	5-19
Table 5-10: Queued jobs after step 5.....	5-20
Tables 5-11,12,13,14,15: Queued jobs after scheduling in step 5	5-21
Table 5-16: Materials Plan Step 6	5-22
Table 5-17: Materials Plan after complete explosion.....	5-23
Table 5-18: Queued jobs after step 6.....	5-23
Table 5-19,20: Queued jobs after step 6.....	5-24
Table 5-21: Materials Plan Step 7	5-24&25
Table 5-22: Queued jobs after step 7.....	5-25
Table 5-23 -> 29: Queued jobs after step 7	5-26
Table 5-30: Final Materials Plan	5-27
Tables 5-31 -> 37: Final machine schedules.....	5-28&29
Table 5-38: Carrying Costs	5-30
Table 5-39: Final materials plan with inventory calculation	5-31
Table 5-40: Operation setup costs	5-32
Tables 5-31 & 34 (repeated with highlights)	5-32
Tables 5-35,36,37 (repeated with highlights)	5-33

List of Figures

Example 2-1: Master Production Scheduling (MPS).....	2-5
Example 2-2: Materials Requirements Planning (MRP).....	2-8
Example 2-3: Capacity Requirements Planning (CRP).....	2-11
Example 5-1: An illustration of the time and product structure transversal under a reverse-time demand processing model.	5-4
Example 5-2: An illustration of the problem of lead-time overlapping as may occur against a part having parent dependent lead times.	5-5
Example 5-3: A 'Bill of Manufacture' notation as derived from the traditional product structure and shop routing diagrams.	5-7
Example 5-4: START and END operations in routing to provide flow control point for jobs entering and leaving the shop scheduling process.	5-13
Figure 5-1: A Bill of manufacture for a simple example of coherent planning.	5-14
Figure 7-1: An example of the manner in which complete sub-branches of a product structure can be clipped from the whole product structure.	7-1
Figure 7-2: Carrying cost comparison under bottleneck conditions.	7-5
Figures 7-3: Machine setup costs under bottleneck conditions.	7-6
Figures 7-4: Order overhead costs under bottleneck conditions.	7-6
Figure 7-5: Total costs under bottleneck conditions (sum of 7-2,3 & 4).....	7-7
Figure 7-6: Number of infeasibilities occurring in the baseline plans generated	7-8
Figure 7-7: Carrying costs under even shop load.	7-9
Figures 7-8 : Machine setupcosts under even loading	7-9

1. Introduction

The work embodied in this thesis was motivated by a need for better systems to perform the planning of material requirements and the scheduling of manufacturing operations for large, job-shop* manufacturers. This need was identified through examination of the systems in use at a number of North America's largest manufacturers. In studying the underlying logic of such systems it is clear that a number of fundamental assumptions used in production planning are not valid. In fact, most planners responsible for these systems are well aware of the inconsistencies arising from these assumptions.

The objective of this thesis is to present a practical solution to some of the problems in current methods of manufacturing planning. In removing the erroneous assumptions in the planning systems it is possible to eliminate a number of the day-to-day problems that arise on the manufacturing shop floor due to infeasible plans. Furthermore, the removal of the problems will allow for the elimination of the safety factors incorporated into production plans to avoid such occurrences. Aside from the intangible costs that shop floor problems represent, the cost of attempting to avoid them is significant in large manufacturing facilities.

The complexity of the manufacturing planning problem as a whole in the job-shop environment is such that it has been under intense study since the early 1900s. With each technological advance in manufacturing processes there has arisen a need to understand better the means of operating and supporting these processes while minimizing overhead costs. The study of manufacturing planning is largely focused on two areas, the planning of material requirements and the scheduling of manufacturing operations on the shop floor. Even with this decomposition of the overall problem into two substantial sub-problems, the solutions to either part have proven to be tractable only to the extent of achieving feasible results with some local improvements. In general, the generation of optimal solutions to the materials planning problem is considered to be impractical. The shop scheduling problem, by contrast, has been shown to be NP-complete. The

* A "job-shop" refers to an operation characterized by many end products having some degree of customization. Manufacturers of aircraft, telecommunications equipment, industrial machinery etc. fall into this category.

complexity of these problems and the lack of success in addressing them as they stand today continues to attract researchers, and to elude a definitive solution.

In the 1960s a model emerged for the development of materials plans which addressed a large number of the issues of interest to manufacturers at that time. This model, Material Requirements Planning (MRP), was widely adopted by North American and European manufacturers. The simplifying assumption of MRP which allowed it to be decoupled from scheduling decisions in the factory was that of infinite capacity to manufacture. Research into the shop scheduling problem continued, but attractive solutions proved much more elusive. The similarity that most scheduling systems share with MRP is an underlying assumption that allows them to discount material planning issues: the assumption of infinite material supply.

The algorithm presented in this thesis will show how these assumptions can be eliminated by combining the materials planning and shop scheduling processes into one 'coherent' production planning system. The algorithm is structured as a framework which can be adapted to the specifics of existing planning systems. Within this framework, techniques from both materials planning and shop scheduling can be incorporated to reflect the unique flavour of each company's operations. In this sense this algorithm is not intended to be a revolution in materials planning or shop scheduling techniques, but a framework for the integration of the two. One of the principal objectives in developing a solution to this problem has been that of practicality. No matter how complex the problem statement, or how dramatic the results, the algorithm will be of no value if it does not address the real industrial problem in a manner that can be tailored to systems in use today.

As part of the analysis of this algorithm, a prototype was developed and tested using data taken directly from a large, job-shop manufacturing operation. The production plans generated by the prototype are compared against results generated using the same data and a planning system representative of those in use in industry today. Many of the performance criteria commonly used by manufacturers in terms of overhead costs and carrying costs were applied to the two sets of results to provide a quantitative contrast. In all cases, the 'coherent' planning algorithm was shown to generate synchronized production plans with consideration for shop capacity and material availability. This algorithm also consistently demonstrated a reduction in costs ranging from 26% to 79%

when compared to the baseline system.

In the final analysis, the characteristics of the algorithm developed will be discussed as they relate to the original problem. The applicability of the algorithm to the general industrial problem will also be examined through a discussion of the aspects of existing systems not explicitly incorporated in the algorithm. Although it is difficult to make a definitive statement about the potential success of the algorithm in an industrial implementation, there is sufficient evidence to support further analysis and small scale industrial trials.

2. An Introduction to Materials & Production Planning Methods

2.1. Introduction

As part of the continued industrialization of North America, brought on by such events as the world wars, a movement came about to increase the efficiency and productivity of assembly line manufacturing operations. The need for a greater variety of complex equipment to support the war efforts drove manufacturers and military logisticians to analyze then current manufacturing processes and planning. Their objective was to eliminate the ad hoc planning methods in use for determining what materials were required at what times and on which machines. Instead they would propose a series of planning and prediction algorithms which would allow manufacturers to forecast their material requirements and to predefine the assignment of jobs to machines to meet a given delivery date. In implementing these algorithms the manufacturer would be able to produce more end product with a reduced requirement for large in-process inventories and, hence, overhead costs.

Since there was relatively low variability in the end goods produced, effort was focused on the optimization of an assembly-line type environment. In this situation, the predictability of requirements for the products was high and the configuration of the factory in which they were to be produced was known. This form of problem lent itself well to solution through mathematical means and resulted in fixed routings in the shop for the production of goods and fixed order quantities for raw materials which were consumed at a known rate.

With the return to the production of consumer goods it was discovered that the assumptions underlying this planning method no longer held true. The actual demand from consumers could not be predicted with a high degree of certainty into the future. Furthermore, the move to better customer service and satisfaction lead to variability in the end products being made as customers were given the choice of several options on a base product. This new format of supply and demand led to the characterization of the make-to-order shop environment as distinct from the make-to-stock requirements of the

military. This situation is also described as the discrete-batch, job-shop environment.

Research into production planning in the discrete-batch job-shop has been largely split into the investigation of materials planning and shop scheduling algorithms. The purpose of the materials planning logic is to ensure that the right quantities of the materials required to build all products are available when needed while minimizing the overhead cost of inventory that is carried to meet these needs. The purpose of the shop scheduling algorithms is to arrive at an efficient assignment of jobs to machinery available in the shop such that the machines have a high degree of utilization and the jobs are completed on time.

In the large majority of manufacturing organizations which fit the discrete-batch, job-shop characteristics the materials planning process is implemented using a methodology known as Materials Requirements Planning or MRP. A later enhancement to the MRP philosophy encompasses a broader range of production planning issues. Manufacturing resource planning (MRP II) includes three principal processes, they are Master Production Scheduling (MPS), traditional MRP, and Capacity Requirements Planning (CRP).

The job-shop scheduling problem has proven to be much more difficult and is, in fact, largely considered to be unsolved for a machine/job combination of any reasonable size. As a result, most manufacturers have implemented proprietary shop scheduling algorithms or have resorted to manual planning.

The material which follows in this section is a brief introduction to the processes described above as they are central to the subject of this thesis.

2.2. Inventories in a Manufacturing Environment

There are several characteristics of manufacturing inventory in the job-shop environment which afford it a specific niche among general inventory management problems. These characteristics are introduced below and have formed the basis for the development of the materials management techniques which are the subject of this section.

2.2.1. Manufacturing vs. Distribution Inventory

Distribution inventory is produced and held to directly satisfy customer demands. In the job-shop, customer demand is typically forecast based on market predictions and previous experience. Instability in the market place leads to variations between forecast and actual demand and hence distribution inventory is subject to a degree of volatility. Manufacturing inventory, by contrast, is held to support the manufacturing process. Starting with the raw materials, manufacturing inventory is eventually transformed into distribution inventory. The requirements for manufacturing inventory are derived from the end requirements for distribution inventory. Given a forecast, the demand for manufacturing inventory is known and stable.

The objective in holding distribution inventory is to provide good customer service in the form of product availability. The cost of providing a particular level of service is balanced by the costs incurred by holding inventory in its finished form. Since actual demand for the end product cannot be known with certainty, it is necessary to hold a high level of inventory to provide a high level of customer service. Conversely, since the demand for manufacturing inventory is known and assumed stable, there is no benefit to be derived in holding more inventory than necessary to meet the production plans. In determining the manufacturing inventory level, therefore, the objective is to minimize the overhead costs incurred from holding inventory given a known set of parameters.

2.2.2. Multiechelon Inventory

Systems in which one 'level' of inventory depends in some manner on preceding or succeeding levels are referred to as multiechelon inventory systems. A simple example of one such system in the distribution environment is the hierarchy of manufacturer's inventory, distributor's inventory and retailer's inventory. Each level in this network is dependent on the levels preceding and succeeding them. In the job-shop environment one is typically concerned with the production of complex products which can be broken down into several levels of sub-assembly and component. Items at any level of the product structure may be held in inventory. When considering inventory, therefore, one must consider the fact that an assembly in inventory includes its constituent sub-assemblies and raw materials. Furthermore, one must recognize the existence of

'dependent' demand where the requirements for a lower level assembly are derived only from the requirements for the parent part. This relationship causes a dramatic ripple effect where small changes in demand for the parents results in completely new demand profiles for children several levels lower in the bill of material.

In planning the inventory to meet requirements and minimize costs it is necessary to consider these hierarchical interdependencies to arrive at a good solution.

2.3. Materials Planning Processes

2.3.1. Master Production Scheduling (MPS)

MPS is the first tool used in the materials planning process and addresses the production schedule at the forecast level of planning. The input to MPS is a series of projected orders or desired production plans stated in vague terms which correspond to the level of detail necessary for budgetary planning (such as 1000 cars per quarter). This input reflects the companies goals and business outlook for the near future.

This high-level input information is then processed using a number of pre-specified policies for distributing the production over the forecast period (e.g. evenly by week, daily, fixed quantity batches etc.) and the actual mix of models and features implied by the 'average unit' (e.g. 30% Blue, 20% Red, 50% Black; 33% automatic transmissions, 80% power brakes etc.). The first result is a forecast for actual buildable products spread out over the MPS horizon. As time progresses and customer orders are received, the MPS process is re-run to reconcile the original forecast against the actual demand. Generally, the closer the time period under consideration is to the actual date, the more weight is given to known customer demand. In any case, the output of MPS is a detailed schedule stating what the factory will produce during specified time periods for the current period and several periods into the future.

It is important to note that the MPS process does not explicitly include consideration for the ability of the factory to actually meet the proposed schedule in terms of human or machine resources. Any consideration for this factor is external to the MPS.

The following is a brief example of the MPS logic:

Example 2-1: Master Production Scheduling (MPS)

MPS input:

Period	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter
Units Forecast	100	150	100	150

Policies: 1) Demand is distributed evenly over the period on a weekly basis (to the nearest whole unit).

2) Typical units ordered are 40% Red, 60% Black

Sample Calculation: Requirements for red units in week 1:

$$(100 \text{ (Q1 forecast)} / 13 \text{ Weeks/Quarter}) \times 40\% \text{ Red Units} = 3$$

MPS output:

1st Quarter

Week of Quarter	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6...	..Wk 12	Wk 13
Red Units	3	3	3	3	3	3 ...	...3	4
Black Units	4	5	4	5	4	5...	...5	4

Note: production can only build whole units hence demand is distributed 'evenly' by WHOLE numbers.

2.3.2. Material Requirements Planning (MRP)

2.3.2.1. Introduction

According to Joseph Orlicky [83], one of the initial developers of MRP, it can be

described as

"A set of logically related procedures, decision rules, and records (alternatively records may be viewed as inputs to the system) designed to translate a master production schedule into time-phased net requirements, and the planned coverage of such requirements, for each component inventory item needed to implement this schedule." pg. 21, MRP

MRP is the process used to determine what components will be needed, and when they will be needed to meet the master production schedule. This process takes each assembly in the product line and breaks it down into its constituent parts using product structure information called a bill of materials. For each assembly or component the system then determines the net requirements and generates shop or purchase orders to satisfy the outstanding demand.

2.3.2.2. Netting of Demand

The MRP process begins with the top level items in the product structure and proceeds one level at a time. Typically the top (or end) items are those delivered to a customer such as a car or telephone switch. The first step in the process for each level of the product structure is to determine the net demand against a part. If the item is a top level assembly then the gross demand is usually derived from actual customer orders or forecast and is called 'independent' demand. The calculation of net demand takes the gross demand for the assembly and subtracts from it the quantity currently available in inventory or projected to be available through pending orders. This calculation is called 'netting'. Once the series of net demands has been determined a lot sizing algorithm is applied to group the demands into appropriate order quantities. The rules used in lot sizing vary widely and often trade off the costs of carrying inventory against the cost of creating new orders. Further lot size modifiers are also considered at this stage such as the minimum or maximum lot sizes allowed or the multiple in which orders must be placed (e.g. minimum of 10 with increments of 7 means 10, 14, 21, 28, ...).

Once the lot sizes are determined these groupings of net requirements are referred to as planned orders. If the item being planned is purchased then an order is placed with the

supplier with the understanding that a certain amount of time will be required before the order can be filled (the purchase lead time). If, however, the component is a manufactured part then its orders are propagated further down through the bill of materials to its children. The demand created at the child level is referred to as 'dependent demand' since it is derived from (or depends on) a demand for the parent.

2.3.2.3. Explosion

The process of propagating demand from parent to child through the bill of materials is referred to as explosion. This term connotes the action of breaking the parent assembly into its many constituent pieces in order to determine what components or sub-assemblies are required to create more of the parent.

During the explosion process an adjustment is made to allow for the manufacturing time required to make the needed components. To achieve this offset, the due date for the child components are set earlier than the due date for the parent. This back off of due dates is based on a manufacturing 'lead time' which is a statistically generated estimate of the time required to manufacture an average lot of the component. In general this lead time is fixed, irrespective of the lot size. Some systems, however, calculate lead times based on the number of components requested in an actual order. It is extremely important to notice that the lead time backoff feature is an estimate and does not explicitly consider the actual capacity available in the shop to manufacture the component during the specified time period. To accommodate the variability in demand and the ability of the factory to manufacture, lead times are typically inflated to leave slack in the plans. The 'padding' of lead times, however, conflicts with the goal of inventory reduction since it causes materials to be built or ordered in advance of their actual need dates. It is also important to realize that it is the use of estimated lead times which allows materials planning to be decoupled from shop scheduling.

2.3.2.4. Level-by-level iteration

The iteration between the explosion of requirements and the netting of demand is repeated through the levels of the product structure until all parts with demand have been processed. The end result of the MRP process is a set of schedules and orders for each

component in the end product from the shippable entity itself down to the raw steel or nuts and bolts used to make it. The logic of this process is designed to make the necessary materials available at the time they are needed without carrying large inventories.

A further function of MRP is to update the current schedules and orders based on changes in the forecast or customer orders. Toward this end, most companies run MRP on a weekly or even daily basis, incorporating the transactions that have occurred on the inventory, product structure and order databases in the mean time. As a result, MRP may recommend that existing shop orders or purchase orders be expedited, de-expedited or canceled in order to keep the material plans consistent with the 'Just-In-Time' philosophy of material availability. The following is a brief example of the MRP process.

Example 2-2: Materials Requirements Planning (MRP)

The preceding diagram shows the structure of an arbitrary product A which is made by assembling one B and two C's. One B, in turn, is made from a D and two E's and each C is made from one E. In the language of materials planning, A is a parent of B and C; B and C are children of A. Similarly, B is a parent of D and E and C is a parent of E. D is a child of B only and E is a child of both B and C. Associated with each component in the diagram is the manufacturing lead time for that part.

Sample Calculation: C's required for one A

Quantity: 1 A x 2 C's per A = 2 C's

C Due Date: wk 7 - 1 wk for A = wk 6

C Start Date: wk 7 - 1 wk for A - 2 wks for C = wk 4

Therefore, if the input from MPS was an order for 100 A's due at the end of week 7, MRP would generate the following materials plan schedule assuming no inventories or pending orders:

		Quantity in Week							Lead
		1	2	3	4	5	6	7	
A	Due date							100	1 wk
	Start date						100		
B	Due date						100		2wks
	Start date				100				
C	Due date						200		2wks
	Start date				200				
D	Due date				100				3wks
	Start date	100							
E	Due date				400				1 wk
	Start date			400					

As an extension to the above example to illustrate the concept of lot sizing consider the following table which shows the net requirements for component D as exploded from all of its parents (remembering that it may be used in assemblies other than A).

Net Requirements for 'D'														
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Requirement	100		400		50	325			90		150			225

Under a DISCRETE lot sizing (order) policy, the requirements for D would be ordered in quantities which satisfy its demand for a specified number of weeks. The table below illustrates the application of the DISCRETE policy using a period of 3 weeks. The quantity on hand for each period, n , is calculated as follows:

$$\text{Onhand}_n = \text{Onhand}_{n-1} + \text{Planned}_n - \text{Requirement}_n$$

Planned Orders Against D Using a 3 week Discrete Order Policy

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Requirement	100		400		50	325			90		150			225
Planned Order	500				375				240					225
Qty Onhand	400	400	0	0	325	0	0	0	150	150	0	0	0	0

Similarly, under a FIXED order policy, the requirements for D would be satisfied by orders for a fixed quantity, such as 200, with a fixed increment, again 200. The table below illustrates the implementation of this policy.

Planned Orders Against D Using a Fixed Order Policy (Qty 200)

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Requirement	100		400		50	325			90		150			225
Planned Order	200		400			400			200					200
Qty Onhand	100	100	100	100	50	75	75	75	185	185	35	35	35	10

2.3.3. Capacity Requirements Planning (CRP)

CRP is the final stage of the materials planning process. In this phase the load placed on the factory's equipment and labour force is determined. This process takes the materials plans from the MRP stage and uses a sequence of operations (routing) for each part to

determine which machines or classes of labour will be required. Since the purpose of this process is to get an idea of the capacity required some simplifications are often applied to the calculation. Common implementations of CRP only calculate the requirements against key machines, classes of labour, or work groups to develop a rough capacity picture.

There are two general classes of CRP system. Those systems which determine the shop load based on the material due dates established by MRP without regard for the ability of the shop to meet these dates are classified as 'infinite' capacity planning systems. The more complex system of automatically limiting the shop load to be within its actual capacity (finite capacity planning) is much less commonly used. In either case, the results of CRP are used only as information for the people actually trying to run the shop. These results are not used to automatically realign the materials plan to reflect reality. If the capacity requirements are too onerous some effort may be made to start back at the MPS level and redistribute the load.

The following is a brief example of the logic used in CRP.

Example 2-3: Capacity Requirements Planning (CRP)

Series of Operations for Items A & B

Consider the sequence of key operations for CRP for two parts as illustrated on the preceding page.

The requirements from MRP are as follows:

15 units of A - due on day 4, start on day 1 (inclusive)

10 Units of B - Due on day 4, start on day 1 (inclusive)

Based on these requirements, the CRP process would generate the following schedules:

Infinite capacity plan

		day 1	day 2	day 3	day 4
M1	units processed	15			
M2	units processed		15		
M3	units processed			25*	
M4	units processed				15
M5	units processed	10			
M6	units processed		10		
M7	units processed				10

** This requirement exceeds the capacity of M3 and will therefore not be met.*

Finite capacity plan

		day 1	day 2	day 3	day 4	day 5
M1	units of A	15				
M2	units of A		15			
M3	units of A			15		
	units of B			5	5*	
M4	units of B				15	
M5	units of B	10				
M6	units of B		10			
M7	units of B				9	1

** The capacity limit on M3 requires that the remaining 5 B's be built in the first quarter of day 4. This leaves 3/4 of day 4 to process B's on M7. The priority assigned to the orders in this example is arbitrary.*

2.4. Production Scheduling

The process of assigning jobs to machines in an efficient way varies greatly from company to company. Unlike MRP, there is no generally accepted means for arriving at this assignment, probably as a result of the overall complexity of the problem. There are, however, a number of underlying characteristics which are common to most shop scheduling systems.

The characteristics of the shop are contained in the description of the machinery in the shop and the hours that each machine is available in one working period, usually one day. Planning parameters associated with the machines often include required maintenance time, cost of using the machine per unit of time, and a variety of relationships tying the machine to higher level aggregations in the factory such as work groups, departments, or divisions. In fact, many shop scheduling algorithms do not schedule at the level of individual machines but may only address work centres which represent a collection of machines capable of performing similar operations.

The link between the shop and the items being produced is defined by the routing for an item. This routing describes the sequence of operations which must be performed on a component or group of components in order to take them from their raw state and incorporate them in a higher level assembly. For each machine that must be visited during this process we have information on the amount of time required to setup the machine to run a batch of components and the time required on the machine per component to perform the operation. The routing may also include alternate machines that may be used in the event that the preferred machine is busy or inoperable.

The input to the shop scheduling process is the shop orders generated through MRP. These orders specify that a particular quantity of a particular item is required on a

particular day. Since the materials planning process works on all levels of the product structure such shop orders are explicitly generated for every component and sub-assembly that would be included in an end product. MRP does not, however, provide the information necessary to link the individual shop orders together (as one depends on the orders for its parent assembly) nor does it provide information indicating which end requirements are to be satisfied by a shop order since the lot sizing process may group several requirements together. In this sense, each shop order from MRP is treated in isolation where the items built to satisfy it are essentially entered into inventory (whether physically or conceptually). The shop order for the parent assembly which incorporates this component then draws on inventory to make the raw materials available at the necessary machines to complete its routing. This process starts with the purchased components and then propagates up the product structure to the deliverable product. The lead time backoffs applied during MRP provides the staging such that component assemblies are scheduled to be built before they are required in the parent assembly.

It is important to notice in this explanation that there is no explicit consideration given to the availability of material resources in the shop scheduling process as a result of the inventory buffering concept. This assumption of infinite material availability parallels the assumption of infinite capacity used in materials planning and allows shop scheduling to be decoupled and simplified by discounting material considerations. While this is generally true for shop scheduling systems it is not correct in the few cases where the scheduler incorporates a breakdown of the source of requirements for all shop orders allowing them to be tied back to the higher level order which generated them.

Since the purpose of the scheduling tool is to generate a feasible shop schedule, the large majority of algorithms work on the basis of finite capacity. That is, a given machine or work centre in the shop can not be assigned more work in any given time period than it is known to be able to accommodate. Working from either the due date or start date passed from MRP, the shop scheduling algorithm selects the highest priority job for assignment to the machine and then blocks out the required amount of time on that machine by summing the total run time for the batch with the setup time. If the algorithm begins with the first available slot on the machine near the due date established by MRP and works backward in time it is referred to as a 'backward loading'. The alternative is to

use 'forward loading' by beginning the job on or near the MRP start date and allocating time on the machine into the future.

The process of assigning priority to jobs which conflict for machine resources is generally handled by a priority heuristic which considers any of a large number of decision criteria such as the due date of each job or their required time on the machine. Further heuristics are applied to handle the problems of determining which machines to schedule first (since many are linked by several item's routings), when jobs can be combined to reduce setups or split to shorten processing times and a wide variety of other scheduling issues which can affect the efficiency of the scheduling algorithm.

The end result of the shop scheduling process is a machine by machine schedule of which operations are to be performed on which items in order to produce the end products as requested through MRP.

If one considers the example presented for the process of CRP under strict finite capacity conditions and with all operations considered on all machines at the lowest level then it will serve to illustrate the basic process of shop scheduling.

3. A Review of Current Production Planning Algorithms and Research

3.1. Algorithms for Integrated Production Planning

The principal focus of this thesis is the development of an integrated approach to materials planning and production scheduling. The need for such an algorithm was determined through examination of several large scale manufacturing operations and their attendant materials planning and production scheduling systems. The assumptions underlying the planning processes gives rise to a lack of coordinated results in the final plans and, hence, schedule inefficiencies and infeasibilities.

The problems that this lack of coordination represents from a practical standpoint are widely recognized and are handled on a weekly, if not daily basis, in most large job-shop manufacturing facilities. That the problem should be so well known and understood, however, and yet so little reported or researched is surprising. There is some documentation, arising primarily from industrial trade show presentations and trade magazines, to support the existence of the problem. A paper by Savage & Mikurak[90] examines the problem of marrying these two planning systems together but makes no firm suggestion as to how this might be accomplished. Instead the paper concludes that if such an integration were to take place the results would be significant for the production planning field. A further paper by Toye[103] address the shortcomings of materials planning in terms of scheduling and shop capacity issues while a paper by Connor[17] addresses the shortcomings of finite scheduling programs in handling material availability issues.

3.1.1. Commercial Scheduling Products

With the resurgence of finite scheduling software products in the manufacturing industry there is further evidence of the need for this process to be tightly integrated with the materials planning process. Recognizing that the input to such programs will be the output of the MRP process, most vendors of scheduling software provide interfaces to a variety of commercial materials planning systems. They further recognize, however, that the scheduling process is likely to schedule jobs outside of the time frame allocated by MRP through standard lead-time backoffs and therefore suggest feeding the results of the

scheduling run back into the materials planning system. To avoid excessive oscillation of the materials plan they propose that these modified order dates be fed in as 'firm' orders which the materials planning system may not regroup or reschedule.

This suggestion for a single-loop feedback from the scheduling program to the materials planning system fails to recognize the purpose of lot sizing algorithms in MRP or that the availability of raw materials may be restricted. Most MRP systems group requirements at each level of the product structure to take advantage of order quantity and order processing efficiencies. In this process the lot-sizing algorithm weighs the cost of carrying material in inventory against the cost of creating another order. By assigning new need dates to the requirements grouped in an order and yet forcing the order quantity to stand, the shop scheduling algorithm frequently negates any cost reduction obtained through lot sizing. While this suggested approach may cause the materials plans and production schedule to be synchronized it only addresses cost efficiency of the resulting plans from the shop standpoint.

3.1.2. Research Results

In spite of the apparent disinterest in this subject area in research circles there has been one significant, long-term study directly relating to this topic. In a series of publications from 1982 through 1985 Marshall[70], Hastings & Marshall[47], Hastings, Marshall & Willis[46], and Hastings and Packham [48] detail the development, implementation and analysis of an integrated material planning and production scheduling system referred to as JOBCODE which implements their 'Schedule-Based' MRP algorithm. This system uses a subset of the problem definition and research objectives as defined in this text but is clearly a full, working implementation of an integrated production planning system.

This approach to the problem is targeted at small to medium scale manufacturing organizations and is considered to be appropriate for handling several hundred to several thousand jobs. One of the key aspects of this system is its approach to integrating the bill of materials structure used in materials planning and the production routing information used in shop scheduling into a single data structure referred to as a 'bill of manufacture'. In so doing, the developers claim to have created a coherent, logical structure representing the product in its entirety. Using this structure, the lead time estimations

present in MRP are replaced by the scheduling of the actual operations required to complete a component of an assembly and in so doing arrive at the actual job completion date as normally determined by a shop scheduler. This information is then used as a firm basis upon which to calculate need dates for other components in the product structure with the confidence that the capacity assumptions of materials planning have been removed.

The implementation of this solution is based on the interactive philosophy of job shop scheduling and production planning. Specifically, the algorithm does not provide for the automatic generation of planned orders or the rescheduling of existing orders as defined in the original MRP model. Marshall[70] describes this philosophy with

"It is assumed that Re-Order Levels (RL) and Re-Order Quantities (RQ) have been set so that the normal situation is that the Projected Net Level remains between the Re-Order Level and the sum of the Re-Order Level and Re-Order Quantity.

When this is not the case, a warning message is printed out beside the transaction which causes the stock level of the item concerned to move outside these boundaries. The messages are:

- i. $STOCK > RL + RQ$ (overstocked according to the set parameters).*
- ii. $STOCK < RL + RQ$ (indicating that a replenishment of stock is required).*
- iii. OUT OF STOCK.*

Nothing automatic is done when any of these conditions show in the time-phased net level of stock, e.g. there is no automatic placement of orders to cover situations (ii) and (iii) above. This is in line with the design philosophy that keeps the computer-based manufacturing system in the role of providing timely, accurate, relevant and intelligible information upon which the Production Administrator can make decisions."

This approach requires that a human resolve the material availability problems before the production planning process can be completed. In essence the person must make material ordering decisions as each level of the product structure is processed.

The developers of this system have demonstrated its usefulness through industrial trials over a period of several years. An analysis of the results of these trials show the project to have been a success in meeting the original objectives.

Although the basic problem definition as stated by Marshall is directly applicable to the problem under consideration and the proposed solution has been shown to be viable in a production environment, the underlying philosophy of direct and frequent human interaction makes it unsuitable for large-scale, job-shop manufacturing. In the environments of interest it is extremely unusual to encounter MRP implementations that can generate a full materials plan in less than four hours. This processing load is largely a result of the sheer volume of data which must be processed since MRP itself is not computationally intense. To suggest that human intervention would be a viable option under these conditions is not sensible. In dismissing this solution for the problem of interest it should be noted that nowhere do Marshall et al. suggest that their approach is intended to address this magnitude of problem. A further compounding of this issue, however, is that the simplification introduced by the use of human decision making dramatically diminishes the value of the schedule-based MRP approach as a whole when addressing the larger-scale problem.

3.2. Materials Planning

3.2.1. Introduction

One of the key objectives in the development of this thesis is the development of a practical solution to the problem of integrating materials planning and production scheduling. One must consider that the environment of choice for its implementation is large job-shop manufacturing operations and that such companies have long ago implemented production planning systems at the expense of many man-years of effort and millions of dollars. It therefore seems reasonable to conclude that the most palatable algorithm from a corporate standpoint will be one which protects the company's current

investment in planning software with a minimum incremental cost. With this in mind, this literature review has been largely restricted to those systems currently in use in industry or those under development which share the same underlying objective of practicality.

There are several key aspects with regard to the manner in which materials planning systems generate shop orders and the manner in which shop schedulers attempt to assign their attendant jobs to machines. Those aspects of interest cause the two processes to be largely incompatible from the standpoint of an integrated calculation. In recognizing and proposing alternatives for the incompatible aspects of the algorithms the integration of the two processes becomes natural. This review of the current state-of-the-art in materials planning and production scheduling is therefore not exhaustive but instead focuses on the salient aspects of the production planning process.

3.2.2. MRP

3.2.3. Introduction

Materials planning using the MRP model is considered to be well understood and, as a reflection of this, is widely implemented in manufacturing operations of all sizes throughout the world. The fundamentals of the model and the popularization of the technique are widely attributed to the works of Oliver Wight, Joseph Orlicky and George Plossl. The American Production and Inventory Control Society (APICS) is an association of manufacturers and manufacturing professionals who espouse the MRP philosophy and promote its use and development in manufacturing organizations throughout the world. As a result of the immense popularity of the method and the extent to which it has been adopted in manufacturing environments there are innumerable texts on the subject. As a further introduction to the topic beyond that provided in the background material one could refer to the works of the 'founders' such as Wight [109] or Orlicky[83] or such general manufacturing planning texts as those of Chase & Aquilano [16] and Greene[44].

One area of MRP which has received ongoing attention and which is particularly relevant to this work is that of the lot sizing techniques applied to net requirements. An

examination of the state of the art in this area is provided below.

3.2.4. Lot Sizing Techniques

Early in the development of MRP systems it was recognized that the selection of a lot sizing technique (or techniques) would have a significant impact on the effectiveness of the system in minimizing inventory carrying costs and exploiting economies of scale in manufacturing batches. Typical MRP systems have been implemented using simple lot-sizing algorithms such as lot-for-lot, and fixed quantity for the purposes of computational simplicity and ease of comprehension. These techniques are typically applied on a part by part basis through all levels of the product structure, thereby treating all items in isolation.

Considerable research has been devoted to the issue of lot sizing in both statistically controlled and MRP based inventory systems. In addressing this issue, researchers have tended to develop algorithms which attempt to minimize the overall inventory and manufacturing costs by balancing the costs of lot sizing decisions in these two areas.

A general formulation of the problem as a mathematical program is not unreasonable and the first widely recognized approach was proposed by Wagner and Whitin in 1958 [104]. They formulated the problem as a dynamic program balancing inventory carrying costs against operation setup costs in the factory. The input to their formulation is a set of discrete demand quantities for an individual item assigned to discrete time periods, a value for the inventory carrying cost per part per day and a value for the manufacturing setup cost per batch run. The output of the algorithm is a set of dates and quantities representing the inventory to be received to meet the specified demand. This approach is guaranteed to give an optimal solution to the problem. This algorithm has long been considered the benchmark against which all single-level, single-item lot sizing techniques are measured.

There are two aspects of the basic Wagner-Whitin algorithm which cause it to fall short of the optimal solution to the materials planning problem as a whole. The first shortcoming is that it fails to recognize the interdependencies between parts in an assembly type (multi-echelon) product structure. Since the lot size specified for a parent

generates the lot size required for its children on explosion it is not possible to consider the two in isolation when seeking an optimum solution. Several researchers have recognized this failure and proposed formulations of the problem which consider the hierarchical dependencies in generating optimal lot size solutions. Steinberg & Napier [100] developed a network model for the assembled product lot-sizing problem which addressed the multi-echelon planning problem resulting from dependent parts. This model considered inventory carrying costs, setup and order costs and initial inventory to produce a set of optimal lot-size decisions for all parts in a product structure given demand for the top level parent part.

Baker, Dixon, Magazine, & Silver [6] and Billington, McClain & Thomas [9] considered a second problem which was ignored by both Wagner & Whitin and Steinberg & Napier which is the recognition of available capacity to produce the assemblies and sub-assemblies under consideration. The latter group proposed a mixed, integer-linear formulation of the problem with capacity constraints. As with most optimization algorithms addressing combinatorially explosive problem sets, these authors recognized that the problem, although tractable, would require a significant amount of time to solve for product structures of any significant depth. In conjunction with their algorithm, therefore, they further postulated a heuristic technique for compressing the product structure through elimination of components or subassemblies ordered on a lot-for-lot basis. This assumption necessarily negates the optimality of the lot-sizing across all levels of the product in return for the ability to generate solutions faster or for larger product structures. The group of Baker et al. proposed a branch and bound formulation but recognized the same computational deficiency as Billington et al.. Baker notes a further deficiency in both algorithms in that they assume a factory which produces one end product, that is, there is no competition for machinery with other end products.

These formulations are representative of optimal lot-sizing techniques which tend to address a subset of the whole materials planning problem in return for a computationally feasible solution. Even with the simplifying assumptions described, the computer processing power required to run these algorithms on a part or product basis has rendered them impractical for MRP systems which must plan inventories ranging from 1,000 to 500,000 individual items. A principal contributor to this heavy computing load is the

requirement to consider and plan all demands and orders in one massive calculation. Most practical MRP systems perform lot sizing on an order by order basis with some minimal look forward or back. A further failing of optimizing techniques is their high degree of sensitivity to changes in input data which can cause the original plan to be completely modified. This characteristic is referred to as 'nervousness' and is an unacceptable quality in material planning systems since requirements change frequently and vendors or factories require a stabilized plan reaching several weeks into the future, regardless of its optimality.

The second and broader research front into lot-sizing techniques has focused on fast heuristics which attempt to duplicate the results of such algorithms as Wagner and Whitin while minimizing the computational overhead. With few exceptions, the development of these heuristics is based on the same single-level, uncapacitated assumptions as Wagner-Whitin for the purposes of achieving computational efficiency. Most of these algorithms determine lot sizes by processing demands individually beginning with the first demand in time. As each new demand is presented to the algorithm it is either grouped with the order currently in progress or the current order is released and a new order is begun. Among the many such algorithms is a widely recognized set of cost based heuristics which deal with one or both of inventory carrying cost, manufacturing setup and order costs. These are:

- EOQ (economic order quantity) - a fixed lot size technique where the lot size is derived using the traditional EOQ formula and forecast demand. The principal shortcoming of this technique is that it is based on a constant rate of inventory consumption and fails to recognize the lumpy nature of discrete demand.

$$EOQ = \sqrt{2US/IC}$$

where U = number of units consumed in one year.

S = the setup cost for a single batch order (or production run).

I = the unit carrying cost per day.

C = the unit cost.

- **POQ (period order quantity)** - this is an inversion of the EOQ technique where the actual demand for a fixed number of days on a part is grouped together and ordered as a single batch. The number of days is derived by taking the EOQ and dividing it into the total yearly demand to determine the average number of days the EOQ is to cover. This technique can compensate for the failings of the EOQ method under discrete demand however it still performs poorly under large demand fluctuations.

$$\text{POQ} = (\text{EOQ}/U) * \# \text{ of inventory days/year}$$

- **LUC (least unit cost)** - this is a dynamic technique which considers each discrete requirement in isolation. Based on the principle that a new order is planned to meet the first requirement after a stock out, this method groups requirements together with the lot required for the first shortage until a requirement causes the sum of inventory carrying costs + setup cost to increase on a per unit basis. For successive requirements the formulae are:

$$\text{First requirement : } \text{Unit Cost}_1 = S/Rq_1$$

$$\text{Second requirement: } \text{Unit Cost}_2 = \frac{S + IRq_1D_1}{Rq_1 + Rq_2}$$

$$\text{Third requirement: } \text{Unit Cost}_3 = \frac{S + IRq_1D_1 + IRq_2D_2}{Rq_1 + Rq_2 + Rq_3}$$

etc.....

$$\text{repeat until: } \text{Unit Cost}_n < \text{Unit Cost}_{n+1}$$

where: Rq represents the quantity of the requirement denoted by the subscript.

D represents the number of days a requirement is carried in inventory

- **LTC (least total cost) / PPB (part period balancing with no lookahead/lookback)** - the concept of the LTC/PPB approach is taken from an observation of the results of the Wagner-Whitin algorithm. In the WW method the inventory carrying costs per lot ordered is nearly equal to the fixed setup cost. This method applies this result by grouping requirements together until the total carrying costs for the lot is greater

than or equal to the order setup cost.

This algorithm groups together n requirements such that:

$$\sum_n IRq_n D_n \leq S$$

- Silver-Meal (least period cost) - the Silver-Meal (Silver & Peterson [97], Silver & Miltenburg [96]) heuristic attempts to minimize the sum of setup and carrying costs per unit of time covered by a particular order. Like the LUC and LTC approaches, this method takes a myopic approach to lot sizing, considering each requirement in time order and deciding whether to begin a new order with it or group it with the previous lot.

Where the per period cost up to requirement n is defined as:

$$(S + \sum_n IRq_n D_n) / D_n$$

This algorithm groups requirements until: per period cost _{$n+1$} > per period cost _{n}

An algorithm closely resembling Silver-Meal is presented by Kicks & Donaldson [60] where the inventory carrying cost is calculated by assuming that the quantity ordered in a lot is consumed at a fixed rate over the intervening time before the next requirement. In the Silver-Meal heuristic the carrying cost is calculated by assuming that each discrete requirement forming the lot is completely consumed on the day it is was required.

The introduction of heuristics to the materials planning problem was meant to address the computational complexity of optimizing techniques. The second criticism of optimizing techniques, however, was that of system nervousness resulting from changes in the input data. While this has not typically been a consideration in the development of heuristics, due to their myopic nature they tend not to exhibit this characteristic to the same extent as the optimizing techniques. A study by Blackburn & Millen [10] examined the use of the Wagner-Whitin, two versions of Silver-Meal and the PPB techniques in a rolling schedule environment typical of materials planning systems (wherein a plan is generated for periods 1 through n , period 1 is implemented, a new plan is generated for periods 2

through $n+1$ etc.). The results of this study show that, although PPB performed poorly overall,

“when a firm has limited information about future demand, either the unmodified Silver-Meal heuristic or the modified Silver-Meal method (a new procedure developed in this research) should be used rather than the Wagner-Whitin algorithm for reasons of cost effectiveness. The finding is particularly interesting in that it contradicts conventional wisdom which holds that, by using one of the simpler lot-sizing heuristics, schedule stability and computational savings are achieved at the expense of poorer cost performance.”

Another set of researchers has addressed the problem from a different perspective in proposing modifications to these lot-sizing techniques to accommodate system nervousness. In their first paper, Carlson, Jucker & Kropp [14] modify the Wagner-Whitin algorithm to incorporate a cost for changes in the schedule given a modified set of input data. The schedule change cost function is oriented to penalizing the addition of new orders to the plan on a time varying basis. The penalty for additions to the near future are extremely high while the cost for changing plans in the distant future is nil. The value of this modified algorithm is not given explicitly but it is examined in context when compared to modified versions of the PPB and Silver-Meal heuristics developed in a later paper ([61]). In the case of the heuristics the authors incorporate the cost of schedule changes in the setup costs normally used in the algorithms, thereby avoiding explicit use of the original schedule in the generation of a new plan. The results of this study support the finding of Blackburn and Millen in showing that the PPB method is unsuitable in the case of changing requirements. A comparison of the modified Wagner-Whitin and Silver-Meal approaches, however, found each to perform well under certain cases of input modifications and planning horizons.

Although system nervousness has been identified as a problem in the case of both optimizing and, to a lesser extent, heuristic techniques, Carlson et al. have shown that the objective of minimizing nervousness can be included in the planning system. There remain, however, several unresolved problems which make both approaches deficient for

the real job-shop problem, these are:

- capacity consideration - although some authors have proposed optimal seeking algorithms which recognize finite capacity, they tend to require other simplifying assumptions to allow a reasonable formulation of the problem. The heuristic research has largely ignored this restriction.
- multi-echelon inventory - as with capacity, most algorithms do not consider the interdependencies of components in an assembly. In the case of heuristics this is a typical simplifying assumption which is introduced to allow isolation of components and hence faster solutions.
- lead time assumptions - one of the most common fictions in all algorithms which consider inventory carrying costs is the use of a standard lead time, independent of lot size. Unless the materials requirements are based on actual shop capacity and the number of pieces to be manufactured in the lot of interest, the number of days the inventory will be carried is no more than a poor estimate.
- consumption rates - in conjunction with the lead time issue is the common assumption that a lot of inventory is consumed on the day it is required representing an infinite rate of consumption. In fact, the operations required to transform one component into another or to incorporate it in an assembly may take place over several days, again depending on the lot size. As mentioned previously, Kicks & Donaldson [60] have proposed an algorithm which incorporates a consumption rate estimation however this method assumes continuous demand for an item. If there is no demand for a particular component over a period of time, they will greatly underestimate the consumption rate by including the zero demand period in the calculation of holding costs.

3.3. Production Scheduling

3.3.1. Complexity of Scheduling

In order to develop a proper perspective for evaluating various approaches to the job shop scheduling problem, it is important to consider some results from algorithmic complexity theory.

The first published studies of the shop scheduling problem were undertaken in the mid 1950s and concentrated on finding optimal solutions through methods such as linear programming and disjunctive graphs. Further study and developments in algorithmic theory led to branch and bound, integer programming, and mixed integer-linear programming attempts. These approaches met with limited success when applied to problems containing more than two machines. In their 1967 book, Theory of Scheduling, Conway, Maxwell, and Miller[18] surveyed the current state of research on the n job/ m machine scheduling problem and opened with the following, oft-quoted statement,

"The general job-shop problem is a fascinating challenge. Although it is easy to state and to visualize what is required, it is extremely difficult to make any progress whatever toward a solution. Many proficient people have considered the problem, and all have come away essentially empty-handed. Since this frustration is not reported in the literature, the problem continues to attract investigators, who just cannot believe that a problem so simply structured can be so difficult until they have tried it."

In this same time frame, advances in algorithmic complexity theory established the existence of a class of problems referred to as NP-complete. Simply put, the worst case of the solution to problems in this class can not be evaluated in a number of steps definable by a polynomial expressed in terms of the size of the inputs to the problem. In a 1972 paper, Karp[58] showed that a known NP-complete problem could be reduced to a set of 25 computationally difficult problems; among these was the general job shop scheduling problem under known execution times, due date constraints and penalties.

This criteria was sufficient to prove that the scheduling problem itself is NP-complete.

Florian, Lenstra, and Rinnooy Kan [31] further developed this result by showing that another known NP-complete problem was reducible to several special cases of the job shop scheduling problem under varying cost and capacity constraints. In an appendix, they conclude that

"...the NP-complete problems are equivalent in that none of them have been well solved, and that, if one of them would be well solved, then the same would be true for all of them. Since all the classical problems that are notorious for their computational intractability, such as the traveling salesman, job shop scheduling and integer programming problems are known to be NP-complete, the polynomial-time solution of such a problem would be very surprising indeed. For practical purposes, this implies that in solving those problems one may just as well accept the inevitability of a bad (super polynomial) optimization algorithm or resort to using a good (polynomial) approximation algorithm."

Given this assessment of the problem, it is not surprising that Conway, Maxwell and Miller [18] arrived at their conclusion nor that this same statement seems almost prophetic in terms of today's research into optimal solutions to job shop scheduling problems.

3.3.2. Analytic Approaches

There can be no question that for every feasible scheduling problem with known objectives and constraints there is a set of optimal solutions. With this in mind, there has been a great deal of research devoted to the discovery of the method which arrives at these solutions most quickly. Given the complexity of the problem, however, it is no surprise that researchers soon discovered that this task was all but impossible for any problem of practical significance. The following is a brief survey of some of the optimizing approaches and techniques that have been applied to the shop scheduling problem.

Early studies saw the general n job $\times$ m machine scheduling problem formulated as an

integer programming problem. Some of the first formulations published were those of Bowman [12], Wagner [104], and Manne [69]. Each of these formulations used binary decision variables for assigning a particular item to an operation in a time slot. Manne claimed the most efficient formulation of the three requiring fewer decision variables and equations than his two predecessors. None of the formulations was particularly compact, however, since even that of Manne would require 31 variables and 94 constraints for a 3 job x 4 machine problem. Bowman summarized the state of such formulations in his paper with

"...no formal restriction including the size of the problem to be solved are inherent in the following method. However, for even a problem of modest size, the computation required will be quite large, and therefore the phrase is recalled - such solutions are very little solved. It is hoped that mechanical and conceptual advances might help to overcome this roadblock."

Story & Wagner [101] experimented with a further formulation using integer programming and attempted to improve on the solution time through the use of several cutting plane techniques. They concluded that the available integer programming techniques were insufficient to match the problem and suggested further study to tailor the algorithms to the problem.

Giffler & Thompson [38] approached the problem from two different directions. Their first study involved an explicit enumeration technique limited to the generation of 'active schedules.' Under the minimum makespan criteria, the set of active schedules is known to contain all optimal schedules as a subset. In spite of the limitation to active schedules, they discover in a further paper (with Van Ness [39]) that this technique is impractical, requiring the generation of 84,802 active schedules for a 6x6 problem with 5 operations per job. Instead they propose a Monte Carlo technique using random and biased random selection criteria for resolving resource conflicts when generating schedules. They conclude that this approach holds more promise in that the time required to generate a good (non-optimal) solution is linearly proportional to the dimensions of the problem. They suggest however, that further study be conducted on the development of rules other

than random selection.

The next major thrust was the application of branch and bound techniques using a variety of all integer and mixed integer/linear approaches. Greenberg [43] presented one of the first such approaches in which the bounding value was determined through the solution of a linear program. The computational effort required for this approach did not seem to justify the results. Subsequent studies focused on a disjunctive graph model with bounding based on minimum paths through the graph. Balas [7], Charlton & Death [15], Schrage [92] and Florian, Trepant & McMahon [33] all presented formulations on this basis, each tending to claim performance improvements over the preceding formulations. The table below is taken from Florian et al. showing comparative results of the latter three against a problem set proposed by Fisher & Thompson [30] (included in the appendix).

Table 3-1: Comparison of optimizing shop scheduling algorithms.

Approach	Problem	Number of Machines	Number of Operations	Value of First Solution	Value of Best Solution	Iterations to Best Solution	Iterations to Optimal
Florian...	I	6	36	59	55	233	241
	II	5	100	1252	1222	124	*2000
	III	10	100	1041	1041	100	*2000
Schrage	I	6	36	55	55	36	186
	II	5	100	1306	1302	1539	*4000
	III	10	100	1193	1156	923	*2000
Balas	I	6	36	152	55	49	*200
	II	5	100	3218	1231	81	*200
	III	10	100	3394	1177	156	*200

**denotes calculations terminated without reaching the optimum.*

In spite of the successive improvements in these formulations, none of them was successful in solving problems more complex than 6 jobs x 6 machines with 6 operations each.

A further branch and bound approach was studied by Lagweg, Lenstra & Rinnooy Kan [63] in which branches were created through a limited enumeration of active schedules and bounds were established by relaxing capacity constraints. In this paper they examined several variations of their formulation and investigated the use of heuristics to improve the solution time. The results of their studies are such that they conclude the paper by confirming Conway's et al. comment on the success of investigation into job shop scheduling.

More recent research on pure analytical techniques for scheduling the job shop has been conducted (Ku & Karimi [62], Kang & Hong [57]) however the motivation for the studies has changed. In proposing algorithms of this sort, researchers are aware of the limitations of the approach and make no pretense as to their use for large scale problems. In presenting their mixed integer/linear formulation Ku & Karimi state that, in spite of the limitations of the formal analytic technique they have developed it in order to discover useful heuristics for the solution of more complex versions of the problem.

The bulk of current research into analytic techniques has grown out of the optimal seeking approaches and focused on the generation of good, sub-optimal solutions in a reasonable amount of time. These approaches tend to relax some of the constraints present in the true statement of the problems and to use hybrid analytic/heuristic approaches for their solution. Some representative examples of these approaches follow:

- McCormick et al [73] formulated the problem as a network of flows and analyzed it as a minimum flow/maximum cut. They then introduced a heuristic to reduce the complexity of the problem which attempts to guide the development of better schedules through the minimization of machine idle time. Under analysis, the heuristic approach produced better results than a purely random generation and consumed less processing time than both enumeration and random generation.
- Libosvar & Proth [66] present an analytic approach to develop a set of good feasible schedules and then apply dynamic programming to select the best single schedule from the set (with the objective of reducing Work in Process, WIP, inventory).
- Sawik [91] decomposes the flow shop problem into a series of integer programming

problems, one for each stage of the process. The optimum solution is determined for each stage independently, starting with the last, using the required output levels as constraints. This approach requires the solution of a number of small integer programs rather than one large problem thereby reducing the computation required. The solution will not be globally optimal however the formulation is flexible in allowing different objectives and constraints at each stage.

- Gavish & Johnson [36] propose an approach in which the planning horizon over which the problem is examined is decomposed into a number of smaller periods and each one is solved independently considering the inputs resulting from the previous stage. In the same vein as the approach of Sawik, this allows for the solution of several smaller problems thereby reducing complexity but does not result in a globally optimal result. This approach is similar to a study conducted by Baker & Peterson [4,5] in which he proposed and evaluated the effectiveness of 'rolling schedules.' A rolling schedule is formed by solving a multi-period problem and implementing only the first period's decisions. In the following period the model is updated and the problem re-solved over the next span. Consider a problem with m periods; this approach would create the following schedules:

$(1 \rightarrow n), (2 \rightarrow n+1), (3 \rightarrow n+2), \dots, ((m-n) \rightarrow m)$

A further advantage of these approaches is that the uncertainty in the requirements for the longer term is given less weight in the solution of the near terms.

These examples are representative of some of the types of approaches being taken to various aspects of the shop scheduling problem. The extent of the research being conducted is such that current work tends to focus on very specific aspects of the general problem the enumeration of which is beyond the scope of this survey.

3.3.2.1. Problems with Analytic/Optimum Seeking Approaches

Pure analytic approaches to the creation of shop schedules suffer from a number of shortcomings which make them unsuitable for application in industry. The principal problems with these approaches are:

- **Simplifying assumptions:** in order to reduce the problem to one that can be modeled mathematically researchers tend to incorporate a number of simplifying assumptions. These assumptions often remove essential elements of the problem from a practical point of view to the extent that the results are not useful in industry. Some of the typical assumptions in these approaches (as reported by Ramasesh [89]) are:
 - a component may not visit a particular machine more than once (no backpassing).
 - a job can not be preempted once it has begun.
 - no alternate routings for components are considered/allowed.
 - machines are assumed to be 100% reliable.
 - there are no assembly operations, that is, all end products are formed from raw material through a series of machining operations only.
 - set-up, queue, wait, and move times are negligible.
- **Problem set:** the input problem set is typically static. No new jobs are admitted to the shop or allowed for in planning beyond the initial set which conflicts with the continuous arrival nature of true job shops. This causes a high degree of volatility or 'nervousness' in the schedule when a slightly modified problem set is solved. According to Graves [42], when

"...schedule plans for future periods are repeatedly changed; (the) instability can be quite costly if the schedule is used as a basis for manpower planning and procurement decisions."
- **Performance criteria:** by far the majority of research into optimal solutions has been conducted under the objective of minimizing the product makespan, or , overall production time from initiation to completion. In practice the objective is typically a combination of minimizing the average lateness while maintaining minimum inventory and incurring minimum extraordinary expenses such as overtime or subcontracting. The majority of the practical considerations are focused on delivering products near their due dates while working within the actual capacity of

the factory (Marshall [70], Conway et al. [18]).

- **Complexity and computation:** the final, and potentially most damaging objection to pure analytic techniques is their combinatorial explosion in complexity and, hence, computation time required as the problem grows. In a similar manner to analytic approaches to lot sizing, the attempt to solve the entire problem in one massive calculation causes the computation to become intractable. Few studies have demonstrated algorithms which arrive at optimal solutions for problems containing more than 6 jobs and 6 machines. In practice, a problem of this size would represent an exceedingly small factory and, in fact, true factories of interest tend to have thousands of jobs to be sequenced on hundreds of machines. The enormity of the computational effort required to generate an optimal solution conflicts with the reality that the problem must be frequently solved and re-solved in industry to accommodate the variability in requirements and furthermore, that it should allow for interactive 'what if' analysis to preempt the surprises introduced by these variations.

3.3.3. Heuristic Techniques

The success, or lack thereof, obtained through the study of optimal seeking approaches led to a branch in the directions of research on the shop scheduling problem. As a further parallel to the work on lot sizing techniques in materials planning, researchers realized that optimal seeking approaches would not be useful in a real job-shop environment and turned to the development of heuristic techniques. By eliminating the requirement for an optimal solution, practical methods were developed which addressed a number of the shortcomings in pure analytic methods, particularly the large amount of computation time required, at the expense of the overall schedule 'cost'. Dannenbring [21] summarized the basic concept with

"In general heuristic methods are relatively economical in their utilization of computational resources when compared to optimizing techniques. The question of which heuristic is most economical is normally overshadowed by the question of how 'good' the solutions are."

Most heuristic approaches are applied within a common framework where the jobs to be scheduled are considered on a machine by machine basis. Time is allocated on the current machine starting with the highest priority job as determined by a priority heuristic. Where the interdependence between components and their parent assembly is tracked (as opposed to relying strictly on MRP lead time backoffs) the required operations for the lowest level component would be scheduled first. Upon completion of the required components, the sequence of operations to complete the parent assembly are triggered and so on up the product structure to the deliverable entity. In this manner most job assignments are made in a forward time order from the jobs which must be processed today to those in the future.

The momentum behind heuristic approaches to job-shop scheduling was such that it dominated the research through the 1970s and into the 1980s. By 1977, Panwalker and Iskander[86] were able to document over 100 job-shop priority rules and sequencing heuristics through the examination of some 36 papers published on the topic.

The extent of the research into this method of job shop scheduling is sufficiently broad that it has been the subject of several comprehensive survey papers beyond that of Panwalker & Iskander mentioned above. To produce a further survey of this sort would be redundant and far beyond the scope of this document. The approach taken here is to overview the research through reference to the surveys and to refer the reader to these papers for further detailed information.

One of the early overviews of research in this area was provided by Gere [37] who provided some of the definitions and underlying principles used in the studies. Three particularly significant concepts were defined which will provide the basis for the development of this section, they were a priority rule, scheduling rule, and a heuristic. A liberal interpretation of Gere's definitions is as follows:

Priority rule: the process by which a number is assigned to each job in queue at a machine indicating their relative importance.

Heuristic: a "rule of thumb" representing intuition as to the best way to sequence jobs.

Scheduling rule: the combination of priority rules and heuristics.

3.3.4. Priority rules

The variety of priority rules developed in this area gives rise to the need for a classification mechanism which describes their assumptions and characteristics. Ramasesh [89] provides the following broadly applicable classification scheme:

A) Dependence on the time in the shop.

- i) Static - a fixed priority assigned as the job enters the shop.
- ii) Dynamic - a priority which may change as the job moves from operation to operation.

B) Dependence on the state of the shop.

- i) Dependent
- ii) Independent

C) Structure of the rule.

- i) Simple - generally based on one parameter related to the job.
- ii) Combinatorial - based on several parameters related to the job or environment.
- iii) Truncated - a set of rules, one of which is applied depending on parameter values.
- iv) Weighted Index - a weighted combination of rules.
- v) Heuristic

D) Information content of the rules

- i) Arrival time.
- ii) Processing time.
- iii) Due date based
 - a) allowance based
 - b) slack based

- c) ratio based
- iv) Cost or value added.
- v) Combination of one or more of the above.

Further classification of priority rules is provided by Ramasesh [89] on the basis of their intended performance criteria. A list of these classifications is provided below with examples of rules designed to perform well under the stated criteria. A breakdown of the classes and specific definitions of the rules mentioned are provided in the appendix.

- A) Time based measures - Under this performance measure, rules such as shortest operation (SPT), slack time per remaining operation (S/OPN), and critical ratio (CRR) are typically used.
- B) Work-in-process (WIP) measures - Rules such as SPT, mentioned above, fewest operations remaining (FOPNR), no setup time (NSUT), least work remaining (LWKR), and least operational slack (OSLK) as well as composite, value based rules, are oriented to minimizing in process inventory.
- C) Due date related measures - SPT, number of remaining operations (NOP), and total processing time (TWK) rules are often applied where the objective is related to the due date.
- D) Cost based measures - Special purpose rules such as the composite-cost oriented scheduling rule (COST) and the delay cost over remaining time (COVERT) rules are designed to address cost based performance criteria. Ramasesh cites several studies, however, that have shown that such simple rules as SPT and S/OPN often outperform cost based criteria since their objectives are often the target of the cost penalties.

Numerous studies have been undertaken to compare the relative performance of priority rules on the basis of a particular criterion or across a set of criteria. In most studies the rules are applied to a set of scheduling problems and their performance measured in absolute (where an optimal solution is known) or relative terms. Muhlemann, Lockett & Farn [80] conducted one such study comparing the RAN, FCFS, EDD, SPT, LWKR, SPT-T, S/OPN, ODD, LSTASRT, CRR, OSLK, and COM rules under varying

conditions of uncertainty in machine availability and rescheduling period. Their performance was then ranked using the flow time/process time, mean queue time, mean lateness, mean tardiness, percentage of jobs late, conditional mean tardiness, and CPU consumption measures. Giving each performance criterion equal weight, the evaluation showed that the SPT and SPT-T rules were the best overall performers with the exception of an unspecified composite priority function rule.

Blackstone, Phillips & Hogg [11] provide an overview of significant studies of this sort conducted and ranked the SPT, S/OPN, SPT-T, EDD, Slack, CRR, FIFO, and COVERT rules. They too determined from this literature survey that the SPT and SPT-T rules lead the others across a variety of performance criteria. These results were confirmed as recently as 1990 by Ramasesh [89] who concluded that SPT/SPT-T are the best rules under a variety of circumstances.

3.3.4.1. Scheduling Heuristics

The use of such myopic priority rules as the sole means for determining shop schedules was recognized as a limited approach when the addition of heuristics might well improve their performance. In his survey, Gere [37] examines seven heuristics which he applied in conjunction with a variety of the priority rules (with reference to an eighth, ad hoc, approach). These are:

- Alternate operation - schedule operations strictly according to the priority rule selected until the scheduling of one operation causes another to become 'critical' (i.e. late). If so, remove the job just scheduled and replace it with the one having the next highest priority; repeat this process until no critical jobs are created.
- Look ahead - in slack based rules there is an attempt made at speeding the processing of a job through the shop as its time runs out. In scheduling a job on a machine, it is possible that a critical job may arrive at the machine while the current processing is under way. This heuristic proposes looking at jobs whose next stop is the queue of the current machine and inserting an idle time to allow them to be processed as soon as they arrive in queue.

- **Insert** - once an idle time has been created in the schedule using the look ahead heuristic it may be possible to complete a job currently in queue during the idle period. This heuristic dictates that the longest job currently in queue which fits in the idle period should be processed.
- **Time-transcending schedule** - the heuristics so far have been based on scheduling the jobs in queue at a particular machine. This heuristic proposes the scheduling of jobs based on priority alone, across all machines. Therefore, the job with the highest priority will have its next operation scheduled first, regardless of the time at which that operation will occur, or the machine on which it is to be performed. This approach contravenes the typical forward time progression of most scheduling algorithms.
- **Subset of critical jobs** - under the general concept of time-transcending approaches, this method proposes scheduling all operations of a subset of critical jobs. The remaining jobs are then scheduled around them hence giving the initial subset highest priority. Gere points out that one flaw with this approach is that jobs will not necessarily remain critical across their entire schedule horizon.
- **Re-do with adjusted due dates** - this method suggests an iterative approach where jobs deemed late in the first iteration of the schedule are assigned a higher priority in the next pass by giving them earlier due dates. The authors suggest a limit of two schedule generations to limit thrashing as the set of late jobs will probably oscillate.
- **Flexibility** - this approach suggests violating the strict priority rules to fill idle time where the existing gap is almost large enough to accommodate the intervening jobs. This method is only applicable in a time-transcending scheduling approach since time progression methods will tend to pack jobs tightly together and do not support look ahead/look back type visibility.

Clearly there are as many possible heuristic approaches to the problem as there are priority rules. Gere's study however reveals some interesting insight into the use of heuristics in general. In his conclusions he states that:

1. When the performance criterion is based on due date, the selection of a priority rule is not as important as the selection of heuristics to apply it.
2. Since the heuristics selected dominate the performance of the priority rules one should select a computationally efficient rule to minimize its impact on processing time.
3. Heuristics which anticipate the future performance of a schedule such as look ahead and alternate operation provide significant improvements. The incremental computing costs incurred by these approaches are well balanced by the performance improvement achieved.

3.3.4.2. Heuristic Techniques versus Optimum Seeking Methods

As with many problems prone to combinatorial explosion, priority rules and heuristics were introduced to the shop scheduling problem as a result of the failure of optimum seeking approaches to produce satisfactory results. In fact, heuristics were successful in addressing some of the problems discovered with optimizing approaches but they also exhibited undesirable characteristics of their own. The following is a brief overview of the key differences and results between heuristic approaches and optimizing techniques:

- **Processing time/problem size:** The principle advantage in the heuristic approach is that it allowed researchers to address problems with practical significance to real job-shops. The reduction in processing time over optimizing techniques was sufficient to allow study of problem sets involving tens of thousands of operations versus problem sets on the order of tens of operations for optimizing techniques.
- **Method of approach:** The nature of optimizing techniques is such that the problem set must be completely known before the scheduling run. This was cited as one of the key problems with optimizing approaches as the true job-shop environment is dynamic and prone to change. This characteristic leads to a high degree of sensitivity to uncertainty in the input data and, hence, schedule nervousness. Conversely, the relatively myopic nature of scheduling rules accommodates a the

variability in the input job set. In fact most studies in this area were based on simulation techniques where the input problem set was dynamically generated from a probability distribution.

- **Quality of results:** Although there is little agreement on whether one can, let alone should, generate optimal schedules, by their nature heuristic approaches are likely to produce 'worse' schedules than optimizing techniques given rigidly defined performance criteria. An interesting defense to this criticism was provided by Dannenbring [21] through a sensitivity analysis of flow shop sequencing heuristics under variability in the input data. He found that

"...performances were not greatly affected by the data errors. Even at 20% standard error, none of the methods had as much as a 5% increase in relative error.

Another significant conclusion is that the differential among the various methods decreases as the standard error increases. ...

This result has great significance with regard to implementation. It is not unreasonable to expect estimation errors in the range of 10 to 15% in practice. In these situations good heuristics offer performance similar to that achievable by an optimizer, but at much less cost."

- **Simplifying Assumptions:** As with optimal seeking approaches, the problems were usually modified by a set of simplifying assumptions. In this instance, however, the assumptions tended to be less general thereby affording a more realistic formulation of the problem.
- **Load Sensitivity:** One aspect of the problem which was not an issue in optimizing techniques was the overall shop load represented by the problem set under consideration. Given the fixed algorithmic nature of these techniques, sensitivity to this parameter would only manifest itself as an increase in solution time required. Under heuristic approaches, however, the shop load was shown to materially affect the quality of the schedule produced by a particular heuristic. Ramasesh [89] cites several studies conducted under different levels of shop load in which the 'best'

heuristic was a function of the load level selected.

- **Assembly Environment:** The majority of research into heuristic techniques did not consider a manufacturing process which involved assemblies and subassemblies. Consequently, the 'end products' under study were typically raw material upon which several operations had been performed. Clearly this assumption is unrealistic in a practical situation where most end products are composed of individually manufactured pieces and then assembled. Sculli [94, 95] was one researcher to recognize this flaw and to study several priority rules under a problem set representing composite products. In this study the SPT rule was compared against several, assembly oriented rules such as JDD (priority goes to the job whose parent has the earliest due date), TJT (priority goes to the job whose parent has the smallest total operation time including completed operations) and NUJOB (priority goes to the job whose parent has the smallest number of uncompleted parts). The results of this study showed that the SPT rule is unsuited to application in an assembly environment and, more importantly, that serially produced product studies do not generalize to the assembly shop case.

3.3.5. Interactive

All computerized information systems can be regarded as tools for the end user. The degree of processing and level at which the information is presented should be a function of the level at which it is most effective for the user to make decisions. In the case of job shop scheduling there is a growing school of thought which proposes that an experienced scheduler should be provided with information at a low level and retain the task of making all decisions. In this framework the system performs the data management functions and implements the scheduler's decisions, providing immediate feedback on their ramifications. This concept is captured by Marshall [70] in describing an interactive system with

"nothing automatic is done when any of these conditions show... This is in line with the design philosophy that keeps the computer-based manufacturing system in the role of providing timely, accurate, relevant and

intelligible information upon which the Production Administrator can make (or investigate) decisions."

Proponents of the interactive approach cite several problems with shop-scheduling which makes it unsuitable for automation. The principal such characteristic is the complexity of the problem itself. In suggesting that the automation of the scheduling function is inappropriate, McKay [74] refers to a series of interviews conducted with a wide group of schedulers who rarely claimed schedule stability lasting longer than fifteen minutes. Using this background information he further supports the interactive approach to shop scheduling by stating that

"Manufacturing plants are complex, random, and dynamic places. Scheduling methods that explicitly seek to incorporate all of the shop floor variables are doomed to failure. The problem is simply too complex. However, almost all good production schedulers have intimate knowledge of the factory floor and most of its uncertainty. The solution to the scheduling problem is to harness the knowledge of the scheduler."

The principal criticisms put forward by proponents of interactive scheduling are:

- for automation to be most effective it is necessary to have a good model. The large amount of intangible information which is present in the shop makes that particular environment resistant to accurate characterization. Given a sufficiently experienced scheduler in the interactive environment, however, the intangibles could be accommodated.
- as outlined in the study of analytic approaches to shop scheduling another significant weakness is data variability. As the algorithms become more complex and automated, the percentage of error in the input data set has a wider effect on the schedules created. Once again, in an interactive mode it is claimed that the scheduler would have a feel for the slack in the data and be able to accommodate it in generating the schedule.
- the final criticism of automated scheduling systems is common to all automated

processes, that being the loss of flexibility when procedures are hard-coded in the system. To address the other points detracting from analytic approaches such as model quality it is necessary to embed more and more functionality in the system. In so doing one loses the ability to change these functions and hence the overall flexibility of the system is diminished.

The lack of success achieved through analytic approaches to shop scheduling has fueled the research thrust in this area. In order to make models more accurate it is necessary to incorporate more detail which, in turn, subjects the model to more sources of error and performance degradation. Initial forays into 'computer assisted' shop scheduling began in the late 1960's with demonstrations of systems which supported, but did not replace, the scheduler's decision making process. Haider, Moodie & Buck [45] provide several historical references for interactive scheduling studies which provided the impetus for their comparative study of interactive and heuristic scheduling techniques. In this study Haider et al. presented a group of six people, with no scheduling background, a set of eight scheduling problems. These schedulers were provided with a computer based system to assist them in creating schedules for the problem sets. A parallel system was developed which applied the Slack-per-Remaining-Operation (S/OPN) rule to the same problem set to create schedules. The results of this study showed that, for six of the eight problem sets the interactively generated schedules were measurably better than the automatically generated schedules based on an average job tardiness performance criteria. Specifically the interactive approach was found to solve problems with a high degree of operation time variability better than the automatic technique. The converse, however, was also found to be true in that for cases of low operation time variability the heuristic approach performed better.

Many authors proposing interactive systems have gone to the extent of implementing and installing them in industrial environments. In a series of papers Hastings, Marshall, Packham and Willis [46, 47, 48, 70] describe their implementation of JOBCODE, a 'combined' materials planning and shop scheduling system. For the decision making processes of both aspects of this system they have applied the user driven philosophy. The system neither recommends material orders in the manner of typical materials planning systems, nor does it impose any scheduling logic on the jobs to be placed on the

shop. Instead the system serves as an information storage and processing base, indicating to the user the ramifications of his decisions but making no suggestions.

A further implementation of an interactive scheduling system is the subject of most papers by McKay [74, 75]. The stated objective in the development of his WATPASS package is to provide tools to support the expert scheduler. An extensive survey of the scheduling community mentioned previously led to the conclusion that these tools must be:

- understandable and predictable.
- flexible.
- able to communicate effectively with the user.
- able to support interactive changes and results processing.

Many of which conflict when one attempts to automate the decision making process.

The fundamental precept of the interactive approach to shop scheduling is the assumption that the scheduler is well equipped to perform the job presented him, much more so than any automated system. The criticisms directed at the automated approaches are largely supported on the interactive scheduling side by claiming that the scheduler is prepared to undertake the jobs himself. The following addresses these points specifically and identifies the weaknesses in the arguments and, hence, in the interactive scheduling approach:

- **problem complexity** - in spite of the fact that it is extremely difficult to address shop scheduling problems from a pure analytic perspective it is not reasonable to suggest that a human can do better on the basis of experience alone. The complexity of the problem stems from the combinatorial explosion of possibilities which occurs as the size of the problem grows. A human is arguably less well equipped to analyze a large number of alternatives than a computer although they may be able to apply broader decision criteria to fewer possibilities. One could further argue that any heuristics that the human might use to reduce the number of possibilities considered can be incorporated in the computer algorithm.
- **model accuracy** - the number of factors describing the problem and the environment

has a direct effect on the outcome of the processing. The more closely the model represents reality, the better the solution. Since the shop scheduling problem is complex there are a large number of parameters which might be incorporated in the system but, for reasons of program complexity, they are not. Alternatively, the human scheduler is able to consider more intangible information in his 'modeling' process and apply this knowledge to his advantage in schedule generation. Humans are not, however, capable of analyzing the possibilities presented through this knowledge in any depth. The trade-off, therefore, is the balancing of a few principal model parameters examined in depth versus the incorporation of many more parameters considered very superficially.

- **data variability** - while it is entirely true that the majority of data available in manufacturing environments is inaccurate it is not equally true that this skews all automated decision processes to the same degree. Optimizing techniques have been shown to be particularly sensitive to data errors however automated heuristics have been shown to be considerably less so (Dannenbring [21]). The ability of a scheduler to apply a second guess to the data presented to him is valid, however his ability to benefit from this knowledge is unclear. It is entirely likely that the scheduler's attempts to compensate for errors in the input data can introduce as many errors as they correct.
- **problem size** - one of the key failings in an interactive scheduling approach is the size of the problem under consideration. Interactive scheduling systems are rarely recommended for shops with more than two or three thousand operations to be scheduled. In reality, the bigger the shop, the bigger the scheduling problem and the greater the need for scheduling assistance. As mentioned previously, humans are far less capable of processing large amounts of information and correlating the different pieces of that information than are computers. While one might argue that a large shop could be handled by several expert schedulers who interact when necessary, it is precisely this type of loosely distributed structure for the management of a single function that is a prime candidate for automation in an industrial setting (as with materials and distribution planning systems).

3.3.6. Artificial Intelligence and Knowledge Based Approaches

3.3.6.1. Introduction

Knowledge based systems have been applied to a broad range of problems where the knowledge of a specific individual or group of individuals is crucial to identifying a solution. The underlying mechanisms for the acquisition, storage and manipulation of the necessary knowledge have evolved over the last fifteen to twenty years to the point where there are products available commercially which provide a comprehensive tool set for the development of such systems.

The subject of job shop scheduling is considered by many to be the province of domain experts whose ability to generate reasonable solutions to a given problem set is derived from an intimate knowledge of the problem and system characteristics. This underlying premise is at the heart of the argument for interactive scheduling systems and is accommodated by those who would automate the application of this knowledge through artificial intelligence.

3.3.6.2. Directions in AI Based Scheduling

The many researchers studying the application of artificial intelligence to job shop scheduling have not only taken different approaches to the problem but have positioned the AI aspect to achieve a variety of goals. The following paragraphs describe a number of projects conducted in this vein.

Smith [98] takes the view that traditional scheduling systems have focused too heavily on the predictive aspect of scheduling, that is, generating an initial schedule. He has developed a system which, given an initial schedule and a set of changes to that schedule, will 'react' to the changes and suggest a new production schedule to accommodate them. The underlying function of the system is based on a set of constraints derived from the initial description of the system and the current schedule. Changes to the schedule give rise to inconsistencies in the original schedule (constraint violations) or opportunities (for improvement). The system relies on a constraint directed search algorithm using the

Beam heuristic to arrive at a feasible schedule incorporating the specified changes. It is possible that, in accommodating these changes the original schedule may be completely changed, an undesirable characteristic from a user viewpoint. Smith suggests in conclusion that predictive scheduling mechanisms could be improved by allowing them to specify a flexible window in which scheduled events can occur thereby allowing greater visibility of opportunities to change the schedule when changes do arise.

There appears to have been a significant amount of effort applied to a particular project documented in papers by Ow, Smith & Howie [84]; Weisser & Howie [106]; and Weisser and Cochran [107]. The system under development is called the Cooperative Scheduling System (CSS) and has directed the use of AI to the distribution of the scheduling task across many processors while maintaining global schedule objectives. The system is physically distributed and decomposed into one Work Order Manager (WOM) task and many Resource Broker (RB) tasks. The WOM formulates requests for shop operations to be performed and 'calls' for the RB's to submit 'bids' on the job. Each RB then considers the opportunity to take on the work with consideration for its available capacity, the time frame requested for completion, and the global objectives of completing work on time (local objectives can be accommodated). The WOM then considers the bids received from the RBs and awards the job based on the bid which best suits the original criteria. The objective in the development of this system is the ability to generate schedules for large manufacturing organizations while considering the characteristics and constraints of each work cell. The structure allows the RBs to represent the nature objectives of the local work cell while still addressing the goal of global schedule efficiency. Decomposition of the problem into communicating sub-tasks also allows for the application of more computing power to address the issue of schedule generation speed. Further work in the area of distributed scheduling systems is documented in the papers by Atabakhsh [1] and Sycara, Roth, Sadeh & Fox [102].

One further direction for the application of AI to shop scheduling has developed with the advent of constraint programming languages. In many implementations of shop scheduling the method of solution is a constraint directed search (as referenced in Smith [98]). At least two groups have identified shop scheduling as a candidate application for such languages. Meng & Sullivan [77] describe a system and language for the

specification of 'operations management' problems called LOGOS with the specific example of the reactive development of schedules. The scheduling problem is defined as a set of constraints expressing the relationship between jobs, due dates and other jobs. Reaction to changes in the scheduling parameters is accommodated by incrementally adding or retracting constraints. The system employs internal constraint directed search algorithms to arrive at feasible solutions to the problem. A further implementation of this approach was described by Van Hentenryck at a workshop of the AAAI special interest group in manufacturing (SIGMAN) and documented by Kempf et.al. [59]. His approach is very similar and involves the expression of the scheduling problem in a constraint programming language called CHIP. Reaction to changes in the schedule is handled through incremental changes in the constraints specifying the problem. This system has the added advantage, however, of allowing the programmer to select the method of search to be used in arriving at a solution such as tree search, problem modification and user interaction. The principal advantage of constraint programming languages advocated by the authors is the flexibility they provide and the range of problems to which they can be applied.

3.5. Practical Implementations

There have been several implementations of AI based schedulers in actual manufacturing facilities which demonstrates a strong belief that such systems can be made to work in a real environment. A recent meeting of the AAAI SIGMAN [59] made reference to several projects such as those at General Motors and Westinghouse which have made use of commercially available expert system shells from Intellicorp and Inference to create scheduling systems for flexible manufacturing environments. Texas Instruments has expanded the scope of the typical shop scheduling problem to consider the availability of material resources as constraints when generating a shop schedule. This aspect brings the system much closer to addressing the problem area of this thesis however the TI system does not simultaneously generate plans for the material requirements. The Cooperative Scheduling System described previously is currently under development in cooperation with people at United Technologies and is scheduled for implementation at one of their manufacturing facilities.

3.6. Due Dates in Job Shop Scheduling

Of the many studies into the problem of job shop scheduling there are relatively few which measure the performance of the proposed scheduling methodology against the original due date of the job. It is, however, fairly widely recognized that, from a practical standpoint, this criterion is one of the most important measurements of the quality of a schedule in a real industrial environment. In his 1984 thesis Marshall[70] states

" The key parameter in deciding the sequence of loading the production facilities up to capacity is the due date of a job. Again, from extensive fieldwork, the meeting of due dates was found to be the key criterion in judging the schedule, outside of schedule feasibility."

There are two general approaches employed to address the due date problem in shop scheduling. The first, and most common, approach is to use the given job due dates as guidelines for scheduling with the end result that the delivery date produced by the scheduler may not coincide with the date specified. This condition most frequently arises when the specified due date is not feasible, a condition which is often correlated to the load on the shop during the scheduling period of interest. The second approach recognizes the fact that the due dates specified as part of the input data set may not have been assigned with consideration for the shop load. In this instance a first pass due date assignment procedure is executed which considers several criteria with regard to the requirements of each job or the conditions existing in the shop and attempts to adjust the specified due dates to reflect these conditions accordingly. This approach recognizes the fact that the quality of the shop schedule produced by the scheduling mechanism and the algorithms employed in that process are affected by the feasibility of the due dates used (as demonstrated by Conway, Maxwell & Miller[18], Weeks[105] and others).

In a survey paper which categorizes some of the previous research into due date assignment procedures, Ragatz and Mabert[88] identify four categories of algorithm for due date assignment, these are:

- Naive Models - algorithms which give no explicit consideration to the requirements

of the job to be scheduled or the conditions in the shop. Methods such as 'Random' (RAN) and 'Constant' (CON) which are based on the random assignment of completion times and the assignment of a constant number of days to complete a job respectively fall in this category.

- Models using information on the arriving job - algorithms which consider parameters such as the quantity ordered, the number of operations required or the total number of machine or labour hours required for a particular job. The logic of this form of due date assignment procedure is reflected in some of the priority heuristics for shop scheduling such as TWK (total work content) and NOP (number of operations) as discussed previously in this section. Algorithms of this sort tend to use these parameters as a weighting factor for modifying an average number of days required for a job.
- Models using information on jobs in the system - algorithms which consider information with regard to the current status of the shop such as the total work load on the shop, the number of jobs in the shop, and the available capacity. There are two primary forms of algorithm applied under this category. Researchers such as Johri [55, 56] and Wiede, Kuriyan & Reklaitis [108] suggest algorithms which analyze the current state of the shop through mathematical approaches to determine the capacity remaining for new jobs or the capacity currently allocated. This result is then used as a weighting factor for the assignment of a due date to the new job to be scheduled. Others such as Dumond, and Mabert[27] advocate iterative, trial and error approaches in which the new job is added to the shop schedule and its resulting completion date is determined. This completion date is then used as an indication of a realistic due date for the job in subsequent scheduling passes. This approach implicitly considers the current shop conditions by actually scheduling the job in the existing environment.
- Models using information on future jobs - algorithms which attempt to allow consideration for jobs not yet loaded in the shop when assigning due dates. Although little used, these algorithms recognize the dynamic environment existing in the shop and attempt to compensate for potential future changes in the shop schedule, a

consideration which is overlooked by methods in the three previous categories.

As an example of due date setting procedures consider the paper by Dumond & Mabert[27] referred to previously. In this paper they describe, implement and compare the results of four methods which span all of the categories they had defined previously. These methods (or rules) are:

Mean Flow Time Due Date Rule (FLOW): This rule is essentially the CON due date rule (under the naive model) in which the due date for a job is determined by adding the statistical mean number of days required for previous jobs of this type to the job arrival date to determine the completion date. This method reflects the lead time allocation procedure in MRP which allocates a fixed amount of time to a job without regard to shop conditions or job characteristics.

Number of Activities Due Date Rule (NUMACT): This rule determines the completion date for a job by allowing a constant number of days for each operation required in the job beginning with the job arrival date. This rule falls under the category of models using information on the arriving job.

Critical Path Time Due Date Rule (CPTIME): The critical path lead time for the job is multiplied by an anticipated delay factor to determine the number of days required. This number of days is then added to the current (arrival) date to determine the job completion date. The consideration of the length of the job's critical path and the anticipated delay due to shop loading causes this rule to fall under both the models which consider jobs in the system and models using information on the arriving jobs categories.

Scheduled Finish Time Due Date Rule (SFT): The implementation of this rule is essentially a first pass attempt at scheduling the job in the shop using the assigned due date. The actual completion date for the job is then taken as the current date plus the number of days required during the initial scheduling attempt multiplied by an anticipated delay factor. The delay factor is similar to that used in the CPTIME rule and is intended to allow for external consideration of future jobs. This rule encompasses aspects of the last three categories of models by considering the current job, the shop status and future jobs.

The use of these rules is analyzed in conjunction with scheduling systems operating under a variety of due date and non-due date based priority rules. The general conclusion of this analysis is that the SFT rule performed best overall in terms job lateness variability and tardiness by a significant margin. The authors do recognize, however, that this mechanism places a higher computational load on the overall production scheduling process than other mechanisms and may not be useful for large scheduling problems.

Although the issue of assigning realistic due dates, and its impact on shop scheduling algorithms has long been recognized as being significant in production planning it is not considered to be solved nor does it appear to receive the same level of attention as the shop scheduling problem itself. It is my impression through exposure to a number of large scale manufacturing environments and production planning systems that this problem is not considered to be solved in a practical sense either, nor is there a uniform approach to addressing it. Manufacturers appear to use some form of constant lead time mechanism when assigning due dates to orders sufficiently far out in their planning horizon. Should they receive a request falling in or near the average lead time for the product, the decisions to accept, reject, or re-negotiate the due dates are handled on an ad-hoc basis. Since, in many instances, penalty costs for failing to deliver on the specified due date are intangible, the trade-off between providing the customer with a firm, realistic completion time and failing to receive the order as a result of quoting long lead times is often heavily weighted toward generating revenue.

4. Problems with Existing Materials Planning and Production Scheduling Systems

In the material presented thus far there have been two principal assumptions upon which materials planning and production scheduling have been based. These assumptions have allowed the processes to be decoupled and treated as independent problems with a minimal amount of data transferred between them. In the materials planning process, particularly those based on the MRP model, there is an implicit assumption of infinite capacity in the factory. Given this assumption, the materials planning problem is reduced to the process of exploding demand, subtracting inventory and generating new orders for the outstanding balance. The corollary to the assumption used in materials planning is the shop scheduling assumption of unlimited materials. That is, an infinite inventory of all necessary raw materials and sub-assemblies is available to the machines used to perform the machining or assembly operations in order to create the end product. The complexity of the scheduling problem alone may be one of the key factors which contributes to the propagation of this assumption. Even when this assumption is present, most researchers focus only on the generation of feasible, not optimal schedules.

It has been demonstrated in many industrial production planning systems (see typical industry presentation papers such as Savage & Mikurak[90], Connor[17], and Towe[103]) that the two assumptions which underlie most industrial planning research are invalid. While it is theoretically possible for an organization to maintain a large staff and extensive production facilities, this is not economically feasible. Furthermore, while it may be possible to purchase and retain a large store of inventory, this is undesirable since the company must support the carrying cost of the inventory. Assuming that the objective of an organization is to maximize profit or minimize cost, the goal is to maintain the minimum possible production facility necessary to meet customer demand while carrying the minimum possible inventory to support the manufacturing process. Since these objectives contradict the underlying assumptions of the materials planning and shop scheduling processes, the conflict manifests itself as materials plans which put the factory in an overload situation and a shop schedule which proposes to build products from raw materials and sub-assemblies that are unavailable.

The following text highlights the specific problems in materials planning and shop

scheduling by addressing the most common models for these processes. The practical implications of these issues are then examined with reference to actual situations arising in manufacturing job-shops and their impact on the manufacturing operation. This section concludes with a specific statement of the objectives of the algorithm to be developed in this thesis to address these problems.

4.1. Problems in Materials Planning

The Assumption of Infinite Capacity to Manufacture

Most materials planning systems process demand on a level-by-level basis down through the product structure. The process begins with the earliest demand in time for the top level (deliverable) products. The entire planning horizon for each product is planned moving forward in time until no further demands exist and all net requirements have been satisfied. Those demands not satisfied through inventory are satisfied by the creation of planned orders. For each component or subassembly incorporated in the product some manufacturing operation is required. As a result, new planned orders for manufactured parts place additional load on the factory. These planned orders are created to cover the demand by the date specified without regard for the current or incremental load on the factory. At no time during typical MRP processing is any consideration given to the ability of the factory to meet the load placed on it.

Time Transversal and the Explosion of Demand

The infinite capacity assumption not only allows the processing of materials planning to be decoupled from shop scheduling but it allows each item in the product structure to be planned, in its entirety, without regard for demands placed on its components. When a planned order is created at a parent level, dependent demand is placed against its components. Demand at the child level is offset in time from the due date of the parent. This offset is a fixed estimate of the time required for the operations necessary to incorporate the child into the parent (the manufacturing lead time). Since the children must be ready for processing before the parent due date, this lead time moves the demand against the child into the past. The MRP model processes demand against a part in forward time sequence, hence this explosion of demand into the past represents an

aberration in the forward progression of time and is acceptable only because the levels of the product structure are decoupled. The constraints placed on parent assemblies through their children in the shop scheduling process can not accommodate this aberration. Since the child demand moves backward in time it requires that a decision has been made previously while scheduling at the child level to accommodate the parent's current need; a prediction of sorts. While this characteristic is not a problem within the confines of MRP it is a barrier to the integration of the MRP and shop scheduling processes.

Lot Sizing Under Uncertainty

Given the preceding assumptions and processes, the last step for materials planning at each stage of the product structure is to perform lot sizing, that is, to group the net demands into order quantities to achieve economies of scale. Lot sizing is done dynamically. As the materials planning progresses the lot size algorithm considers the grouping of each new, unsatisfied demand with the current order being planned. Most lot sizing algorithms, as discussed in chapter 3, are based in some way on the date upon which the demand occurs. The decision to group a particular demand with an order quantity being created is usually made with consideration for the number of days the excess material will be carried in inventory before it is consumed. The drive toward minimum inventory is in keeping with the much heralded 'Just-In-Time' philosophy of materials management. This method of determining lot sizes fails to recognize that the requirement dates are based on estimates of the time required to build the product in the shop (lead time). To accommodate the uncertainty in manufacturing, most systems use heavily padded lead times. The end result is that the grouping of requirements into order quantities to achieve economies of scale and reduce administrative overhead is fruitless since the key parameters used in the algorithm are known to be unrealistic estimates. The application of lot-sizing algorithms is an important opportunity for cost improvement however the time estimates must be removed.

The fundamentals of materials planning in general and the MRP model in particular as described above are well covered in the texts by Orlicky[83], Wight[109], and Marshall[70].

4.2. Problems in Shop Scheduling

The Assumption of Infinite Material Availability

The input to a shop scheduling system is the open, firm, and planned orders generated by MRP. These orders identify each item that must be produced in the shop, the quantity of the item required, and the time frame in which the job should be scheduled. Shop scheduling algorithms fed from MRP treat each of these jobs in isolation without regard for the constraints introduced by the materials required before a job can begin. While the assumption of infinite material availability is rarely stated explicitly in shop scheduling studies, the lack of reference to this issue in any of the papers reviewed in chapter 3 only serves to underscore its existence.

Loss of Hierarchical Relationships

The MRP process performs all of the time phased material requirements planning for all levels of the product structure. In so doing, it eliminates the need for the hierarchical connection between the material requirements and the higher level assemblies in which they are used. Most shop scheduling algorithms, therefore, view inventory as a buffer, retrieving the necessary components from stock and issuing assembled products to stock. Provided that the jobs required to create the component assemblies for any given product were scheduled within the fixed window allowed by the manufacturing lead time used in MRP, the inventory buffer concept holds true. If, however, the components were not scheduled and completed within the allowed lead time then the parent job may attempt to begin when the store room is empty.

This problem is further compounded by the application of lot sizing logic to group requirements from one or more parent assemblies without retaining the parentage information. Once again, this information should not be needed since the materials planning system has allowed for the schedule offsets of the components. The absence of this information restricts some of the scheduling opportunities which might exist if the order breakdown was known.

The principles and problems of shop scheduling as outlined above can be found in the

texts by Conway, Maxwell & Miller[18], Muth & Thompson[81], and Marshall[70].

4.3. Practical Implications of the MRP/Shop Schedule Skew

Material Shortages and Idle Machinery

There are synchronization problems which arise when the underlying assumptions of materials planning and production scheduling prove to be incorrect. If there is insufficient capacity in the shop to meet the due dates as used by MRP then the due dates will be pushed out by the shop scheduler. The two conditions which arise from such changes depend on the level of the affected item in the product structure.

If the item rescheduled is an assembly then its components may be prepared before they are needed. While this is not catastrophic, it does affect inventory overhead costs. Since the material plans have been generated based on the original due dates, raw materials may be purchased and held in inventory for a longer period of time than necessary thereby incurring carrying costs. Furthermore, since the parentage of the lower level assemblies has been filtered out of the information used in the shop scheduler, those assemblies which are not capacity constrained will continue to be produced for the target due date causing more valuable products to be held in inventory until the necessary machinery is available. The costs arising under this circumstance are explicitly considered in the objective function stated in section 4.4.

If the item rescheduled is a sub-assembly then the higher-level assemblies in which it is to be included may attempt to begin processing before the necessary materials are available. This condition is much harder to quantify in dollar terms and may be much more severe. If the time allocated to the operations on the parent assembly can not be occupied by substitution of another job then the machinery and workforce will be idle. As a result of the interdependencies of the many jobs to be run, the failure of one job to run in its scheduled time slot will produce a ripple effect on subsequent work centres and products. In an extreme case, such a delay may well shut down several machines until the component delay is resolved. Due to the varied nature of this type of problem, its cost can not be explicitly incorporated in an objective function.

Increased Overhead Cost

The manufacturing industry has long experienced the material shortages and machine idle time due to this skew between material plans and production schedules. To avoid the costs resulting from such problems, many safety factors have been incorporated into both the materials planning and production scheduling processes. Material order quantities are inflated to protect against stock outs. Manufacturing lead times are padded to account for product delays. These safety factors contribute directly to the inventory carrying costs, order setup costs and machine and labour costs. Therefore, in an effort to minimize the intangible costs of the synchronization problems manufacturers have substituted very tangible overhead costs.

In identifying the underlying synchronization problems one can envision the opportunity to improve the planning, and hence manufacturing, processes. By eliminating these problems one eliminates both the tangible and intangible costs associated with them. Furthermore, in the absence of such problems, there is no need for safety factors to guard against them nor the overhead costs they incur. This opportunity is the purpose of, and motivation for this thesis.

4.4. Objectives of this Research

The central purpose of this thesis is to develop an algorithm which better meets the needs of large, job-shop manufacturers for the generation of materials plans and production schedules. A 'better' algorithm can be strictly defined as one which:

- **produces feasible, synchronized materials plans and production schedules.**
- **incurs less cost than current systems in industry.**
- **is practical and can be implemented in existing manufacturing operations.**

It is interesting to note that Marshall [70] is one of the very few persons to have confirmed the existence of the synchronization problem and that he has done so as a result of practical manufacturing experience. While Marshall's solution is targeted toward a medium scale, interactive planning environment and is not suitable for larger manufacturers, his statement of the problem is directly applicable. Similar to Marshall, the foregoing formulation of the problem has arisen as a result of exposure to a number of large manufacturing planning systems. The objective of putting forward a solution to the problem is therefore not sufficient unless that solution is practical in industry. As Marshall states, there is no value in solving a problem which does not satisfy a real need.

The manufacturing environment of interest for the purposes of this thesis can be described as follows:

- a job-shop environment producing highly configurable, highly complex products such as aircraft engines or telephone switches.
- several end products which may share common assemblies and components and which may compete for the manufacturing resources in the factory.
- a limited ability to manufacture, that is, a capacity limited production facility.
- a limited ability to acquire raw materials.

- incurs indeterminate, finite costs in the event of material shortages or idle machinery.
- an objective of maximizing profit through the minimization of manufacturing overhead costs, specifically, inventory carrying costs, machine setup costs and order generation costs. The factory is assumed to operate under an intangible objective of delivering products on time.

In developing an algorithm that is practical for industrial use there are several system level objectives that must be considered, these are:

- must be able to easily accommodate changes in the input data.
- must be fast enough to support repeated planning runs and, preferably, to allow interactive, "what if" evaluations of changes to the input data or parameters.
- must allow for the stabilization of data in the near term schedule resulting from relatively minor changes in the input data set.
- must support the planning of factories having databases on the order of 10,000 items and several thousand end item demands.

4.5. Measures of Schedule Quality

The means for measuring the quality of the plans generated are many and varied. These measures are incorporated in analytic lot-sizing approaches such as the Wagner-Whitin algorithm which uses inventory carrying costs and order setup costs as the principle cost assessment criteria.

Ramasesh [89] cites a wide variety of performance measures commonly used in shop scheduling analyses. In selecting the criteria to be used in assessing the performance of this algorithm these criteria were considered. A heavier weighting was given to those criteria found in existing industrial planning systems. Of these criteria the primary, tangible, measures in use are due-date performance, inventory carrying costs (including

value of work-in-process or in queue), and order setup costs (administrative overhead, receiving, ...). One further cost-based parameter that relates closely to shop scheduling and materials planning performance is that of machine setup time or costs. Although this is not typically assigned a dollar value in industrial scheduling, it is often an element of the objective function in shop scheduling algorithms.

These criteria address the quantifiable costs outlined in the explanation of section 4.3. It is not reasonably possible to quantify all of the costs resulting from material shortages or machine idle time. These conditions arise as a result of synchronization infeasibilities between the material plans and shop schedules. In order to properly account for, and remove these costs, a requirement of feasibility is imposed on the objective.

Therefore, a simple statement of an objective function incorporating these criteria is:

Produce a feasible production schedule which minimizes

Σ [order costs, carrying costs, overhead costs]

while delivering all end products on the specified due date.

5. Elements of an Algorithm for Creating Integrated Material Plans and Production Schedules

5.1. Introduction

In developing an algorithm to address the problems outlined in chapter 4 one must identify and focus on the key aspects of the materials planning and production scheduling process which cause them to be difficult to integrate. One view of a solution to this problem is to develop a single process which performs the material planning function at the same time as the shop scheduling. An objective in creating this single process has been to accommodate current planning methodologies with minimal distortion. This objective has been one of the guiding principles in the development of this algorithm.

The first several sections of this chapter address the key issues in bringing the materials planning and production scheduling processes together. The resolution of these issues is then presented through the use of a simple example. Finally, the criteria used in analyzing the result of the schedules produced by this plan are presented with a discussion.

An example of the integration of these principles into a prototype is deferred to chapter 6. This format has been selected since the actual details of the implementation of the principles is not essential to understanding their purpose and application.

5.2. A Conceptual Model for Coherent Production Planning

5.2.1. Time and Product Structure Traversal

Traditional materials planning and shop scheduling processes are based upon a two dimensional iterative algorithm. In materials planning the 'outer' loop steps through the levels of the product structure from top-level assemblies to purchased components. The inner loop traverses time from the earliest relevant date to the end of the planning horizon for each item being planned. In shop scheduling, however, the progression through the product structure is inextricably linked with the progression through time. Shop scheduling algorithms tend to move forward through time beginning with the earliest jobs to be scheduled and through the product structure from the lowest level

components to the top-level assemblies. During this process, however, there may be overlapping traversals through the product structure over time.

In order to bring the shop scheduling and materials planning processes together it is necessary to impose a consistent method of dealing with the progression through the product structure and through time that satisfies the requirements of both processes. From the materials planning standpoint, these requirements are as follows:

- i) They must begin with top level items since customers order end products and the bill of materials is unidirectional.
- ii) The hierarchy of assemblies, sub-assemblies and components must be maintained (components are built into the proper assemblies).
- iii) They must process demands in sequential time order (earliest to latest or vice versa) to be compatible with most lot-sizing algorithms.

The shop scheduling algorithm imposes the following restrictions:

- iv) The hierarchy of jobs must be maintained (finish of one job triggers the start of the next).
- v) The hierarchy of assemblies, sub-assemblies and components must be maintained.

If one considers the restrictions presented by shop scheduling methods we see that we can process the jobs either forward or backward in time (earliest job first/ latest job first). Either traversal through time satisfies requirement (iv) since one can either use the end date of child jobs to set the start date of parents (forward time) or the start date of parent jobs to set the end dates of the children (backward time). Furthermore, we can schedule the jobs by following a model of assembling the final product (bottom-up) or by dismantling the product (top-down) to satisfy requirement (v).

The materials planning restrictions state that we must move through the product structure in a top-down fashion only. This requirement (i) arises from the fact that independent demand is placed against the top-level products. The dependent demand on the children can only be determined by traversing down the bill of material. If we remove the

assumption of a fixed lead time used in the traditional MRP model, as will be discussed in the following section, then we lose the advantage it presents for decoupling the levels of the product structure. The only mechanism available to ensure that we move strictly down the product structure when processing demands is to start with the last demand in the planning horizon and move backward in time. This process recognizes the time-phased nature of manufacturing in that the last jobs performed or demands processed must be for top-level (deliverable) products. Conversely, if we moved forward through time, we would process demands in the order in which the products are actually assembled, bottom-up, thereby violating requirement (i). Consider example 5-1 which illustrates the implicit top-down traversal of the product structure when moving backward through time.

The second requirement imposed by the materials planning process, that of maintaining the hierarchy of parts is satisfied by this time/product structure traversal sequence as it was for shop scheduling.

The remaining requirement (iii) further complicates the algorithm. In moving backward through time and processing demand against parts in time order, latest to earliest, we are moving towards the traditional shop scheduling model of product structure traversal. That is, we will move down through various branches of the product structure several times (as shown in Ex. 5-1) as we progress from the end to the start the planning horizon.

Example 5-1: An illustration of the time and product structure transversal under a reverse-time demand processing model.

Product Structure

* one child per parent; one week offsets

Requirements Schedule

In the requirements schedule we see that processing the demands in reverse order causes two top-down traversals of the product structure, one for each order on A, since no lot sizing is applied. The arrows in the schedule indicate the dependent demand relationships created by explosion through the bill of materials. In all cases the direction of time transversal coincides with the arrows indicating the direction of traversal through the bill of materials. If we consider the forward time case, the direction of time transversal would be reversed and would oppose the desired traversal sequence through the product structure.

Since a component may have several parent assemblies with varying lead times, it is possible that a parent with demand in some week, n , may place dependent demand on the component in week x (where $x < n$). A second parent with a longer lead time may have a demand in week $n+1$ but not place demand on the child until week $x-1$. One can see that if the demands are exploded in time order (latest first), the demands presented at the child level may not arrive in this sequence and therefore violate requirement (iii). The explanation of the manner in which this restriction is satisfied addresses the shop scheduling philosophy and is therefore deferred to section 5.2.5. This problem is illustrated in Example 5-2.

From the product structure diagram of Example 5-2 we see that C is a child of both A and B. If we process demands from latest in time to earliest, we first process the demand for 2 A's in week 6. This requirement places a dependent demand of 2 on C in week 2. We next process the demand for 1 B in week 5 which places a dependent demand for 1 C in week 3. If the explosion occurred when the parent demand was processed then item C receives the demand for 2 in week 2 (from A) and then the demand for 1 in week 3 (from B). Remembering that we want to process demands from latest in time to earliest, we see that the arrival of demands at C violates rule (iii).

Example 5-2: An illustration of the problem of lead-time overlapping as may occur against a part having parent dependent lead times.

5.2.2. Lead Time Estimation

As has been discussed previously, one of the key assumptions underlying MRP is that of a fixed manufacturing lead time. The actual time variability experienced in the shop as a result of capacity limitations or lot sizes when manufacturing a group of components, however, is not accommodated by this model. In synchronizing the materials plans and production schedule it is necessary to eliminate this estimate and instead substitute the actual manufacturing time that the job can expect to require in the shop.

Marshall[70] recognized that the most accurate measure of the manufacturing time required for a job is taken from the time allotted to its job sequence on the actual shop schedule. To determine this period one must work from one end of the job sequence and schedule all intervening jobs to arrive at the opposite end. In performing the shop scheduling of the required jobs to determine the start and end dates we eliminate the estimation process entirely in favour of the actual, firm, scheduled dates.

The framework established for the progression through time and the product structure in this algorithm has been structured to accommodate this mechanism. In this algorithm, moving backward through time, the due date of each job is taken as being fixed and the operations are scheduled in the opposite order to that in which they will actually be conducted; last operation first. The scheduling is conducted in the manner of dismantling the end product instead of assembling it. The planned orders against the parents are put through a shop scheduling algorithm, allocating time backward from the fixed due date to determine their start dates. This start date is then used as the required due date for the child components to be used in building the parent. At each level of the product structure this process is repeated in a manner similar to the level-by-level explosion and netting process of MRP.

Marshall [70] provides a convenient notation for the illustration of a combined product structure/operation sequence diagram which he refers to as a 'bill of manufacture'. The structure of the diagram itself suggests the sequence of steps the planning process must conduct when traversing down the product structure. An adaptation of this notation is provided in example 5-3 and is compared to the traditional diagrams.

Example 5-3: A 'Bill of Manufacture' notation as derived from the traditional product structure and shop routing diagrams.

Traditional Product Structure and Shop Routing diagrams

Equivalent 'Bill of Manufacture' Diagram

5.2.3. Materials Planning

In the background material on MRP provided in chapter 2 there were two key processes described. The process of *netting* refers to the balancing of supply against demand for an item when determining if new planned orders are required. The explosion process takes the planned orders created in the netting process and passes them down through the product structure to the child assemblies of the current item (if any). A level-by-level iteration takes place with netting and explosion performed for each part on the current level of the product structure.

The same concepts apply to the algorithm presented here however the processes have been adapted somewhat to accommodate the new time/product structure traversal mechanism. The traversal sequence in use in this algorithm moves backward in time making several passes down through the product structure. Since we can no longer use the complete level-by-level, part-by-part planning approach of MRP, the netting and explosion procedures are done in a dynamic, piece wise fashion. The advantage of the backward traversal through time is that for each demand processed, all demands for all parts which appear later in time (and, hence have already been processed) are known with certainty. This allows the simultaneous planning of all active parts in the system since no lot-sizing decisions can affect the demand grouping assigned previously.

The basis of the netting logic will be presented with the assumption that the system contains no inventory or open orders. This assumption is not a necessary restriction of this algorithm. The use of this assumption is to simplify the explanation of the algorithm; an explanation of the accommodation of non-computer generated* supplies is provided in section 8.2.3. With the elimination of supplies from the problem, the process of netting is reduced to the generation of new planned orders to satisfy net demand. As the process moves through time, demands are generated for each part. In section 5.2.1 it was stated that a requirement of most lot-sizing algorithms is that demands be received sorted in time sequence, either forward or backward. This requirement is satisfied, as explained in the following section. The only remaining problem is that of developing a

* computer generated supplies or orders refers to purchase or shop orders that are suggested by the materials planning algorithm, independent of any human input.
Peter Bishop

lot-sizing procedure that can process demands backwards. Section 3.2.4 presented many lot sizing procedures which either operate on the whole planning horizon for a part at once or, dynamically by considering each demand forward in time. Consider the reverse-time formulations of two cost-based lot-sizing algorithms presented previously:

The formulations presented below use the following notation:

- **R_q** represents the quantity of the requirement denoted by the subscript.
 - **S** represents the set up cost for each lot processed.
 - **D** represents the number of days a requirement is carried in inventory.
 - **I** represents the per part, per day inventory carrying cost.
 - **D_n** represents the number of days requirement n must be carried in inventory if it is grouped with the current requirement.
- 1) LUC (least unit cost) - the reformulation of this method groups requirements together until adding another requirement causes the sum of inventory carrying costs + setup cost to increase on a per unit basis. For successive requirements the formulae are:

First requirement, n:	no calculation
test for second requirement, n-1:	$S/R_{q_{n-1}} \leq \frac{S + IR_{q_{n-1}}D_{n-1}}{R_{q_{n-1}} + R_{q_n}}$
test for third requirement, n-2:	$S/R_{q_{n-2}} \leq \frac{S + IR_{q_{n-2}}D_{n-2} + IR_{q_{n-1}}D_{n-1}}{R_{q_{n-2}} + R_{q_{n-1}} + R_{q_n}}$
etc.....	

The first requirement meeting the specified condition begins the next new order.

- 2) LTC (least total cost) / PPB (part period balancing with no lookahead/lookback) - the reformulation of this method groups requirements together until the total carrying costs for the lot is greater than or equal to the order setup cost. The

reformulation of this method is very similar to the LUC method presented above.

First requirement, n:	no calculation
test for second requirement, n-1:	$IRq_n D_n \leq S$
test for third requirement, n-2:	$IRq_{n-1} D_{n-1} + IRq_n D_n \leq S$

The first requirement causing the specified test to fail begins the next new order.

It is not difficult to reformulate other cost based lot-sizing algorithms to operate on demands received in reverse time order. Many other non cost-based lot-sizing algorithms are also easily adapted to use in reverse time sequence. Algorithms such as DISCRETE which groups demands for a specified period of time (days, weeks, ...) require no modifications to operated in reverse.

Interestingly, some of the simplest algorithms do not lend themselves to reversal. The characteristic which identifies such algorithms is a tendency to myopically over-order on requirements. An algorithm such as FIXED which simply generates an order for a fixed quantity whenever the current stock level is too low will tend to order more than is required in the current time period. This presents a problem since excess material would normally be carried forward into the next period, however this period has already been planned due to the backward traversal through time.

A reverse time formulation of the PPB lot-sizing algorithm was used in this thesis. The implementation and results of this decision are described in chapters 6 and 7.

The second phase of the materials planning process is the explosion of net demand to child assemblies. This process is implemented in essentially the same manner as it is in materials planning systems today. The one exception is that the lead time backoff mechanism has been replaced with a shop scheduling algorithm. When the materials planning logic releases a new planned order, that order is queued up and scheduled in the shop. When child components are required, dependent demand is placed on the child using the scheduled start date of the job and the parent order quantity times the child-per-parent multiplier.

One advantage of a coherent production planning model is the accessibility to information that does not normally get passed between MRP and a shop scheduling algorithm. In MRP individual requirements grouped into an order by a lot-sizing algorithm typically lose their identity and are only exploded further as one lump sum. In this model the detailed information making up a lot is retained. Only the original due date of each sub-lot is changed to reflect its new due date. When the individual quantities are exploded to the child level the opportunity exists to make more intelligent lot-sizing and shop scheduling decisions. Specifically, shop scheduling techniques such as lot-splitting and overlapping can be applied in a much more efficient manner. The advantages provided by a detailed explosion of demand are discussed further in chapters 6 and 8.

5.2.4. Shop Scheduling

The central purpose of this thesis is not to develop a better shop scheduling or materials planning algorithm per se, but to create a framework which allows methods currently in use to be integrated and to produce synchronized plans. With this in mind, the following adaptation of a simple commonly used shop scheduling model was adopted.

When planned orders are released from the materials planning logic for any manufactured part in the system the operations required to manufacture that part are queued at the appropriate work centres with the necessary run times and setup times calculated. The sequence of operations is maintained such that only the last operation in the routing is marked as 'ready to run.' When each job in sequence has been scheduled and assigned firm end and start dates the prior job is triggered using the start date of the current job as the due date for the preceding job (the earlier job in time). In assigning the jobs to the necessary work centres a strict backward load to finite capacity was used. This is essentially a mirror of the widely used forward load to finite capacity algorithm described by Hastings & Marshall et al. [46, 47, 48, 70] and McKay et al. [74, 75]. The details of the implementation of the backward loading algorithm are given with the description of the software prototype in chapter 6.

One outstanding issue from previous explanations (section 5.2.1) is the manner of ensuring that dependent demands arrive at an item in reverse time sequence only (latest demand in time first). The example provided with the description of the problem illustrates how a demand at one parent can easily propagate through the shop scheduling algorithm and arrive at the child before a later demand against a second parent has finished being scheduled. The end result is that a child is presented with the demands out of sequence which violates requirement (iii) specified for materials planning in section 5.2.1. To compensate for this potential time skew a modification has been made to the routings. Without changing the actual sequence of jobs in the routing two new jobs have been added. The first 'machine' on which each job must be scheduled is the 'END' machine (remembering that the last job is scheduled first). The last 'machine' on which each job must be scheduled is the 'START' machine. All manufacturing jobs are forced through these common operations which serve as flow control points for jobs entering and leaving the shop. So as not to alter the flow time of the job through the shop, both of these machines have infinite capacity and the actual run and setup times for the jobs is zero. This means that once a job is selected for processing on either the START or END operations, that job is instantaneously completed and passed on to the next operation. The job priority selection criteria applied to these machines allows one to control the order in which the jobs enter or leave the shop and, hence, the order in which dependent demand is passed on to child components. To address this specific problem, therefore, a strict policy of scheduling the START machine by due date (latest due date first) was used. By maintaining a *scheduling window* size that roughly corresponds to the longest estimated lead time in the system, the jobs leaving the START operation and creating dependent demand can be maintained in strict reverse time sequence across all parts. Example 5-4 illustrates the modified routing concept.

Example 5-4: START and END operations in routing to provide flow control point for jobs entering and leaving the shop scheduling process.

**Routings for Arbitrary Items A, B and C
(forward time traversal)**

**Modified Routings for Items A, B and C
(reverse time traversal)**

5.2.5. Infeasibilities in Plan Synchronization and Due Dates

In the presentation of the problem in chapter 4, one of the stated objectives is the generation of feasible materials plans and production schedules. There are two common forms of infeasibility which are exhibited in manufacturing planning.

The first type of infeasibility is characterized by a lack of synchronization between the materials plans and production schedules. This problem was discussed in section 4.3 and has been eliminated from the plans produced through the use of the coherent planning algorithm. This lack of synchronization is one of the fundamental problems with existing manufacturing planning systems.

The second form of infeasibility manifests itself as requirements for material on a date before the present. The explosion of demand backward in time from parent to child during materials planning may result in dependent demand arriving against a child in a 'past due' time period. Traditional MRP systems often exhibit this problem. In fact, anyone who has had exposure to actual MRP installations of any reasonable size will know that the 'past due' time bucket is never empty. This condition arises when the due date set for a demand at the parent level is unrealistic. The issue of due date setting and the research being conducted into this topic was presented in chapter 3.

The incorporation of the backward explosion of demand used in MRP may cause the

coherent planning algorithm to exhibit this same problem. If an independent demand is requested on an infeasible due date, the algorithm will cause operations to be scheduled before 'today'. Similar to MRP systems, the coherent algorithm, as it is presented in this thesis, is predicated on the assumption of feasible due dates during each separate planning pass. Unlike MRP a mechanism for iteratively adjusting the due dates to ensure feasibility has been explored. The material presented in chapters 7 and 8 will more fully address the issue of due date setting and the investigation which has been undertaken to ensure that it is adequately accommodated in this algorithm.

5.2.6. A Simple Example of Coherent Production Planning

An example of the integration of the principles of this algorithm are presented below. It should be noted that this example is trivial in its complexity when compared to the magnitude of the problems of interest. One example of the manner in which these principles might be integrated into a planning system is presented in chapter 6.

Figure 5-1: A Bill of manufacture for a simple example of coherent planning.

In this example the following conventions shall be used:

- requirements schedules will be titled Materials Plan.
- the current demand under consideration for processing will be shown in bold on the materials plan.
- order quantities under consideration will be shown on the order line in italics.
- orders shown in normal text are considered to have been released to the vendor or the shop as appropriate.
- an order policy of DISCRETE with a 2 day interval will be used for all parts (see table 5-16).
- subscripts on the orders are order numbers used for tracking demand (see table 5-1).
- subscripts on requirements indicates the parent order from which the demand originated (see table 5-16).
- the operations have the following daily capacities: START = inf; END = inf; Paint = 15; Assemble = 30; Polish = 35; Mill = 50; Join = 80.
- a first come first served (FCFS) priority rule will be adopted in selecting jobs for scheduling.
- jobs will be scheduled as soon as they are available. No consideration will be given to jobs which may arrive earlier in time.

Step 1: Process latest demand in the planning horizon. Current date = day 15.

Table 5-1: Step 1 Materials Plan

		Materials Plan									
		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt							4	7		5
	Order										5 ₁
G	Reqmt						6			10	
	Order										

Since no orders have been released in the first step and there are no jobs waiting to be scheduled in the shop we proceed with the second demand.

Step 2: Process next demand in horizon (moving backward in time). Current date = day 14.

Table 5-2: Step 2 Materials Plan

		Materials Plan									
		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt							4	7		5
	Order										5 ₁
G	Reqmt						6			10	
	Order									10 ₁	

Once again, no orders have been released and no jobs are waiting to be scheduled. This situation continues until day 12 is considered therefore steps 3 & 4 have been combined in the next materials plan.

Steps 3 & 4: current date = day 12.

- group demand on day 13 with demand on day 15 since they are within 2 days of each other.
- do not group requirement for day 12 with order on day 13 since the order would then cover 3 days.
- release order on day 13, start new order on day 12.
- queue jobs resulting from released order on appropriate machines.

Table 5-3: Steps 3 & 4 Materials Plan

		Materials Plan									
		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt							4	7		5
	Order							4,	12,		
G	Reqmt						6			10	
	Order									10,	

To determine the time required for a queued job on a particular operation the following formula is used:

Operation Time = (order quantity x run time per part) + batch setup time.

19000 COUNTY ROAD 6, SUDBURY, ON N3B 6B9 USA
TEL 512 438 7007 FAX 512 438 7007 TLM 810000000

Table 5-4: Jobs queued after step 4

Queued Jobs				
Item	Operation	Order	Due Day	Time
A*	END	1	13	0
A	Paint	1		29**
A	Assemble	1		86
A	START	1		0

* Operations in bold are ready to be scheduled.

** time = (qty x run time) + set up time = (12 x 2) + 5 = 29

In selecting jobs to be scheduled they must meet two criteria. First the jobs which succeed them in time must have been completed (since jobs later in time are scheduled first) and their due date must be later than the current date. When performing scheduling the process is repeated until no further jobs are available to be scheduled. Since at least one job meets the scheduling criteria we will initiate a scheduling pass. The scheduling logic proceeds as follows:

- select the highest priority job to be scheduled on the current machine (in table 5-4 the only ready job is for item A on the END operation).
- locate a sufficiently large time slot on the machine on or before the job due date.
- allocate this time on the machine.
- update the waiting job queue if there were any dependent jobs.
- check the job queue to determine if any other jobs are ready to be scheduled; if yes repeat these steps.

At the end of the scheduling pass the job queue and machine schedules are as shown below.

Tables 5-6, 7, 8: Queued jobs after scheduling in step 4**Queued Jobs**

Item	Operation	Order	Due Day	Time
A	Assemble	1	11	86
A	START	1		0

End Operation: capacity = inf

Item	Order	Date	Time
A	1	13	0

Paint Operation: capacity = 15

Item	Order	Date	Time
A	1	13	15
A	1	12	14

Step 5: Process demand on day 11. Current date = 11.

Table 5-9: Materials Plan Step 5**Materials Plan**

		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt							4	7		5
	Order							4,	12,		
G	Reqmt						6			10	
	Order						6,			10,	

The requirement for day 11 against item G is three days earlier than that on day 14 therefore it is not grouped in with the current order, order 2. A new order is begun, 4, and order 2 is released. The jobs required for order 2 are queued as a result of its release.

Since, however, there are no remaining demands against items A or G then their pending orders are flushed out and their corresponding jobs are queued as well. The resulting job queue is shown in table 5-10.

Table 5-10: Queued jobs after step 5

Queued Jobs				
Item	Operation	Order	Due Day	Time
G	END	4	11	0
G	Assemble	4		29
G	START	4		0
A	END	3	12	0
A	Paint	3		13
A	Assemble	3		30
A	START	3		0
G	END	2	14	0
G	Assemble	2		45
G	START	2		0
A	Assemble	1	11	86
A	START	1		0

The many queued jobs initiate a scheduling pass which will continue until a job sequence completes or there are no further jobs in the queue which meet the scheduling criteria. Since all independent demands in the system have already been processed the system will iterate between scheduling jobs and exploding demands on completion of a job sequence until all planning is complete. During this process, the current date for planning is moved back one day at a time. If the scheduling process runs until the first job sequence

completes the following job queue and machine schedules will result:

Tables 5-11,12,13,14,15: Queued jobs after scheduling in step 5

Queued Jobs

Item	Operation	Order	Due Day	Time
G	START	4	10	0
A	Assemble	3	10	30
A	START	3		0
A	START	1	8	0

Paint Operation: capacity = 15

Item	Order	Date	Time
A	1	13	15
A	1	12	14
A	3	11	13

START Operation: capacity = inf

Item	Order	Date	Time
G	2	13	0

Assemble Operation: capacity = 30

Item	Order	Date	Time
G	2	14	30
G	2	13	15
G	4	12	29
A	1	11	1
A	1	10	30
A	1	9	30
A	1	8	25

End Operation: capacity = inf

Item	Order	Date	Time
G	2	14	0
A	1	13	0
A	3	12	0
G	4	11	0

Step 6: Explode demand to child assembly

With the completion of the job sequence for order 2 on item G dependent demand can be placed on its children. This new demand start orders for both items C and H as shown below.

Table 5-16: Materials Plan Step 6

		Materials Plan									
		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt							4	7		5
	Order							4,	12,		
G	Reqmt						6			10	
	Order						6,			10,	
H	Reqmt								10,		
	Order								10,		
C	Reqmt								20,		
	Order								20,		

Step 6: Schedule and Explode until a child order is released.

The scheduling process continues until the next job sequence completes at which time another explosion of demand to the appropriate children takes place. The tables below show the state of the system once the first orders at the second level of the product structure have been released.

Table 5-17: Materials Plan after complete explosion

		Materials Plan									
		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt							4	7		5
	Order							4 ₁	12 ₁		
G	Reqmt						6			10	
	Order						6 ₁			10 ₁	
H	Reqmt					6 ₁			10 ₁		
	Order					6 ₁			10 ₁		
C	Reqmt					12 ₁			20 ₁		
	Order					12 ₁			20 ₁		

Table 5-18: Queued jobs after step 6

Queued Jobs				
Item	Operation	Order	Due Day	Time
C	END	6	13	0
C	Mill	6		123
C	Join	6		125
C	START	6		0
A	START	3	7	0
A	START	1	8	0

Note that no jobs are queued for item H as it is a purchased part.

Tables 5-19,20: Queued jobs after step 6

Assemble Operation: capacity = 30

Item	Order	Date	Time
G	2	14	30
G	2	13	15
G	4	11	29
A	1	11	1
A	1	10	30
A	1	9	30
A	1	8	25
A	3	7	30

START Operation: capacity = inf

Item	Order	Date	Time
G	2	13	0
G	4	10	0

Only those operations that had new jobs scheduled are shown.

Step 7: Iterate.

Once again the system would continue to iterate between scheduling of jobs and exploding of demand at the completion of a job sequence. This step is a snapshot of the state of the system once all manufactured items have received all of their demands.

Table 5-21: Materials Plan Step 7

		Materials Plan									
		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt							4	7		5
	Order							4,	12,		
G	Reqmt						6			10	
	Order						6,			10,	
C	Reqmt		4,	12,		12,			20,		
	Order		4,,	24,					20,		

Table 5-21 (cont'd)

		Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
B	Reqmt		12,	36,							
	Order		48,,								
H	Reqmt					6,			10,		
	Order					6,			10,		
F	Reqmt				40,						
	Order				40,						
E	Reqmt				20,						
	Order				20,,						

Table 5-22: Queued jobs after step 7

Queued Jobs				
Item	Operation	Order	Due Day	Time
B	END	11	7	0
B	Polish	11		250
B	Join	11		396
B	START	11		0
C	END	12	7	0
C	Mill	12		27
C	Join	12		29
C	START	12		0
C	END	8	8	0
C	Mill	8		147
C	Join	8		149
C	START	8		0

Table 5-23->29: Queued jobs after step 7**Assemble Operation: capacity = 30**

Item	Order	Date	Time
G	2	14	30
G	2	13	15
G	4	11	29
A	1	11	1
A	1	10	30
A	1	9	30
A	1	8	25
A	3	7	30

START Operation: capacity = inf

Item	Order	Date	Time
G	2	13	0
G	4	10	0
C	6	10	0
A	1	8	0
A	3	7	0

End Operation: capacity = inf

Item	Order	Date	Time
G	2	14	0
A	1	13	0
C	6	13	0
A	3	12	0
G	4	11	0

Paint Operation: capacity = 15

Item	Order	Date	Time
A	1	13	15
A	1	12	14
A	3	11	13

Mill Operation: capacity = 50

Item	Order	Date	Time
C	6	13	50
C	6	12	50
C	6	11	23

Join Operation: capacity = 100

Item	Order	Date	Time
C	6	11	100
C	6	10	25

Step 8: Flush outstanding demands and jobs.

The system continues to iterate until all demand has been exploded and jobs scheduled.

The Final materials plan and operation schedules are shown below.

Table 5-30: Final Materials Plan

		Final Materials Plan											
		Day 0	Day 4	Day 5	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt									4	7		5
	Order									4,	12,		
G	Reqmt								6			10	
	Order								6,			10,	
C	Reqmt				4,	12,		12,			20,		
	Order				4,,	24,					20,		
B	Reqmt				12,	36,							
	Order				48,,								
H	Reqmt							6,			10,		
	Order							6,			10,		
F	Reqmt		48,	8,			40,						
	Order		56,,				40,						
E	Reqmt		24,	4,			20,						
	Order		28,,				20,,						
D	Reqmt	144,,											
	Order	144,,											

Tables 5-31 through 37: Final machine schedules**Assemble Operation: capacity = 30**

Item	Order	Date	Time
G	2	14	30
G	2	13	15
G	4	11	29
A	1	11	1
A	1	10	30
A	1	9	30
A	1	8	25
A	3	7	30

START Operation: capacity = inf

Item	Order	Date	Time
G	2	13	0
G	4	10	0
C	6	10	0
A	1	8	0
A	3	7	0
C	12	5	0
C	8	4	0
B	11	1	0

END Operation: capacity = inf

Item	Order	Date	Time
G	2	14	0
A	1	13	0
C	6	13	0
A	3	12	0
G	4	11	0
C	8	8	0
B	11	7	0
C	12	7	0

Mill Operation: capacity = 50

Item	Order	Date	Time
C	6	13	50
C	6	12	50
C	6	11	23
C	8	8	50
C	8	7	50
C	8	6	47
C	12	6	3
C	12	5	24

Paint Operation: capacity = 15

Item	Order	Date	Time
A	1	13	15
A	1	12	14
A	3	11	13

Polish Operation: capacity = 100

Item	Order	Date	Time
B	12	7	100
B	12	6	100
B	12	5	50

Join Operation: capacity = 100

Item	Order	Date	Time
C	6	11	100
C	6	10	25
C	12	5	29
C	8	5	81
C	8	4	68
B	11	4	32
B	11	3	100
B	11	2	100
B	11	1	100
B	11	0	64

The foregoing material has presented a simple example of the use of the algorithmic concepts outlined in the first sections of this chapter. In reviewing the resulting schedules one can see that the materials plan and shop schedule are entirely synchronized. That is, no operations are scheduled to begin when materials are not available, nor are materials made available long before they are required unless the materials lot-sizing algorithm has purposely done so.

5.2.7. Assessment of Schedule Quality

As an example of the application of the objective function from section 4.5 we will determine the 'cost' of the production schedule created in the example of section 5.2.6. For this purpose the following costing information will be used:

Order Overhead Costs:

From the final materials generated in the example we have a total of 15 planned orders at a cost of \$28 each, therefore,

$$\text{Total order cost} = 15 \text{ planned orders} \times \$28.00/\text{order} = \$420.00$$

Inventory Carrying Costs:

The table below is derived from the accompanying materials plan. The length of time material is carried in inventory includes the day on which the order was received and excludes the day the material was consumed.

Table 5-38: Carrying Costs

Inventory Carrying Costs/Part/Day								
	Item A	Item B	Item C	Item D	Item E	Item F	Item G	Item H
Carrying Cost	\$0.20	\$0.16	\$0.11	\$0.06	\$0.03	0.04	0.10	0.05
Part-Days Carried	10	36	24	0	4	8	0	0
Total	\$2.00	\$5.76	\$2.64	0	\$0.12	\$0.32	0	0

The total inventory carrying costs are \$10.84.

Table 5-39: Final materials plan with inventory calculation

		Final Materials Plan											
		Day 0	Day 4	Day 5	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15
A	Reqmt									4	7		5
	Order									4 ₁	12 ₁		
	Inven.	0	0	0	0	0	0	0	0	0	5	5	0
G	Reqmt								6			10	
	Order								6 ₁			10 ₂	
	Inven.	0	0	0	0	0	0	0	0	0	0	0	0
C	Reqmt				4 ₁	12 ₁		12 ₄			20 ₂		
	Order				4 ₁₂	24 ₁					20 ₆		
	Inven.	0	0	0	0	12	12	0	0	0	0	0	0
B	Reqmt				12 ₁	36 ₁							
	Order				48 ₁₁								
	Inven.	0	0	0	36	0	0	0	0	0	0	0	0
H	Reqmt							6 ₄			10 ₂		
	Order							6 ₇			10 ₅		
	Inven.	0	0	0	0	0	0	0	0	0	0	0	0
F	Reqmt		48 ₈	8 ₁₂			40 ₆						
	Order		56 ₁₃				40 ₉						
	Inven.	0	8	0	0	0	0	0	0	0	0	0	0
E	Reqmt		24 ₁	4 ₁₂			20 ₆						
	Order		28 ₁₄				20 ₁₀						
	Inven.	0	4	0	0	0	0	0	0	0	0	0	0
D	Reqmt	144 ₁₁											
	Order	144 ₁₃											
	Inven.	0	0	0	0	0	0	0	0	0	0	0	0

Machine Setup Costs:

The opportunity to combine setups for like items during processing and thereby reduce setup costs does exist. This fact is usually taken into consideration in the development of shop scheduling algorithms. This device has not been applied in the coherent planning algorithm since this algorithm does not materially affect the opportunity to make such grouping decisions. It is expected that this mechanism would be integrated with the coherent algorithm as part of the scheduling logic drawn from the existing system.

Each separate job that was scheduled to run on an operation incurs a setup cost. Table 5-40 summarizes these costs for the five real operations (excluding the START and END operations). On the machine schedules shown in table 5-40, the days requiring setup operations are shown in bold.

Table 5-40: Operation setup costs

Operation Setup Cost/Batch					
	Paint Operation	Assemble Operation	Polish Operation	Mill Operation	Join Operation
Setup Cost	\$12.50	\$22.00	\$8.00	\$16.00	\$18.00
# of Setups	2	4	1	3	4
Total	\$50.00	\$88.00	\$8.00	\$48.00	\$72.00

The total Machine setup cost for these schedules is **\$266.00**

Tables 5-31 & 34 (repeated with highlights)

Assemble Operation Schedule			
Item	Order	Date	Time
G	2	14	30
G	2	13	15
G	4	11	29
A	1	11	1
A	1	10	30
A	1	9	30
A	1	8	25
A	3	7	30

Mill Operation Schedule			
Item	Order	Date	Time
C	6	13	50
C	6	12	50
C	6	11	23
C	8	8	50
C	8	7	50
C	8	6	47
C	12	6	3
C	12	5	24

Tables 5-35, 36, 37 (repeated with highlights)**Paint Operation Schedule**

Item	Order	Date	Time
A	1	13	15
A	1	12	14
A	3	11	13

Polish Operation Schedule

Item	Order	Date	Time
B	12	7	100
B	12	6	100
B	12	5	50

Join Operation Schedule

Item	Order	Date	Time
C	6	11	100
C	6	10	25
C	12	5	29
C	8	5	81
C	8	4	68
B	11	4	32
B	11	3	100
B	11	2	100
B	11	1	100
B	11	0	64

The total cost of the production plans produced in the example are $(420.00 + 10.84 + 266.00 =) \696.84 .

6. A Software Prototype

6.1. Introduction

The algorithm for coherent production planning developed in this thesis was fully implemented and tested. The validity of the approach was evaluated through comparison to the results produced by a typical industrial production planning system in use today. This section will describe some of the specifics of the implementation of the algorithm itself and the system used for the comparison of results. The structure of the software and the data used to drive it will be presented. Details of the algorithms implemented in the materials planning and shop scheduling systems will be given and will be shown to be representative of an industrial implementation. Finally, the logic used to apply the cost analysis measures detailed in chapter 5 will be described and the justification for the manner of applying them will be given.

6.2. Development Platform

All software developed for the analysis of the coherent planning model was created on an IBM Risc System 6000 model 530. The code was developed under the IBM AIX version 3.1.5 of the UNIX operating system using the C programming language.

6.3. The Software Structure

6.3.1. Comparison Baseline

6.3.1.1. Overview

The system created as a basis for comparison of the algorithm developed in this thesis was a composition of two pieces of software. The materials planning system was taken from an implementation of an industrial MRP system in use at a large job-shop manufacturing operation (to be referred to as The Company) which exactly meets the target environment for this algorithm. The basic software for the MRP system was

developed by CSI/CARP Systems International of Kanata, Ontario and is currently in use by The Company as a high-speed emulation mechanism for their existing host MRP and CRP system. The results produced by the CSI system have been verified to match those produced by The Company's host MRP system with negligible variations. With the exception of the addition of a level-by-level order tracking mechanism, the CSI implementation of the MRP system was used without modification in this research.

The shop scheduling method presently in use at The Company was not made available for the purposes of this study. Instead, an algorithm was coded as part of the thesis research and is driven by data from the output of The Company's MRP system. This algorithm uses common shop scheduling techniques as found in such commercial scheduling products as WATPASS from Waterloo Engineering Software and RSS from Northern Computing Systems. The custom code written for the shop scheduling component of the baseline comprises some 3000 lines of C.

The pseudo-code for the baseline algorithm is given below, note that the demand file contains all independent and dependent demands passed from MRP, sorted by start date in ascending time order. The demands in this file have already had lot-sizing and lead time backoffs applied during the MRP process. A detailed breakdown of the underlying logic will be presented after the overview along with the relevant lines of code. The line numbers have been preceded with a B to indicate baseline code.

```

B1.  initialize data structures:
B2.  load items
B3.  load machines
B4.  load routings
B5.  load bill of material
B6.  load order breakdown

B7.  set the max_date to the earliest job date + the
      scheduling window size
B8.  read a demand from the demand file

B9.  while (not at the end of the demand file)
B10. while (the demand due date < max_date)

```

```
B11.      enqueue the jobs required for this order on the
          machines in the shop.
B12.      flag the first operation of this job sequence
          as ready to schedule.
B13.      read the next demand from the demand file

B14.      endwhile

B15.      initiate a shop scheduling pass:
B16.      continue_scheduling = TRUE
B17.      while (continue_scheduling = TRUE)
B18.          continue_scheduling = FALSE
B19.          for (each machine in the database)
B20.              while (there is at least one job ready to be
                  scheduled on this machine)

B21.                  search the waiting queue for the highest
                  priority job ready to schedule
B22.                  search the scheduled queue of the machine
                  for the first open time slot on or after
                  the job start date
B23.                  assign this job to the selected slot
B24.                  remove this job from the machine wait
                  queue
B25.                  continue_scheduling = TRUE

B26.                  if (this job has a successor)
B27.                      flag the succeeding job as ready to run
B28.                  else
B29.                      if (this item has any parents)
B30.                          find the parent order which created the
                          child demand and flag as read to run
```

```

B31.          if (the job end date > parent start date
                from MRP) then
B32.          push the parent start date (*normally
                shows up as a stockout)
33.          endif
34.          endif
35.          endif
36.          endwhile
37.          endfor
38.          endwhile
39.          add the scheduling window size to the max_date
40.          endwhile

41. schedule any remaining jobs and explode dependent
    demands
42. while (any machine has jobs remaining to be scheduled)
43.     initiate a scheduling pass
44. endwhile

45. assign a cost to the plans generated:
46. determine the number of orders and their cost.
47. determine the number and cost of part-inventory days
    carried
48. determine the number and cost of machine setups

```

6.3.1.2. Materials Planning in the Baseline Algorithm

As stated earlier, the materials planning for the comparison baseline was performed by a software package presently in use at The Company. The simplest means of providing the flavour of this package is to give a general list of the materials planning features it supports. These are:

- independent and dependent demand.

- inventory
- computer planned orders, firm planned orders, open/released orders.
- alternate part substitution.
- lot-sizing policies:
 - part period balancing (PPB) with look-ahead and look-back.
 - discrete
 - fixed
- scrap/shrinkage factors
- minimum, maximum and multiple lot size modifiers.
- fixed lead time backoff.
- safety stock level.
- order action reporting (cancel, reschedule, ...)

This system included a CRP module for the determination of infinite capacity loading of the work centres and labour classes in the factory. This aspect of the implementation is not relevant to this thesis.

The complete set of independent and dependent demand generated by the MRP system and the planned orders against all parts in the database was passed to the shop scheduling module. From the combined demand and order file a table was created which associated the individual demands at each level of the product structure with the planned order which had been placed to satisfy them. This grouping information was used in the shop scheduling module.

A file containing only planned orders was used to drive the shop scheduler. This mode of operation is the traditional approach to the MRP/shop scheduling sequence. In this approach the planned orders passed from MRP are treated as independent jobs in the shop and are scheduled to start on the MRP specified date. The detailed handling of the demand will be given in the explanation of the shop scheduling logic.

6.3.1.3. Shop Scheduling in the Baseline Algorithm.

The shop scheduling model used for the comparison baseline software is intended to represent commonly used shop scheduling algorithms in use in industry today. The basis for these algorithms comes from analysis of current research into the shop scheduling problem and commercial products available for this purpose. The algorithms implemented are described in this section with reference to the pseudo-code given in section 6.3.1.1.

Scheduling Window

**B7. set the max_date to the earliest job date + the
 scheduling window size**

A scheduling window was implemented in the shop scheduling algorithm to allow proximate jobs to compete for machine resources. This technique is commonly used to build up a waiting queue at the job. The system iterates between enqueueing new demand falling within the current window and attempting to schedule those jobs enqueued.

Job Priority Heuristic

**B20. while (there is at least one job ready to be
 scheduled on this machine)**
**B21. search the waiting queue for the highest priority
 job ready to schedule**

In line B20, a job is considered to be ready to schedule if its preceding job has been completed (and hence this job's start date is known) and if the start date for the job is greater than or equal to the current max-date in the scheduling routine. Line B21 is invoked when more than one job in the queue for a machine is ready to be scheduled. In section 3.5.3 an overview of the research conducted into the development and use of priority rules in shop scheduling was given. For the purposes of this implementation a two level priority mechanism was used. The Earliest-Due-Date (EDD) priority rule was first applied in an effort to give priority to the meeting of delivery dates. If this rule did

not produce a preferred job then the Shortest-Processing-Time (SPT) rule was invoked. This rule gave preference to the job requiring the least amount of time on the machine. Justification for the use of these heuristics as being representative priority rules can be found in the papers by Blackstone, Phillips & Hogg[11] and Muhlemann, Lockett & Farn [70]. These priority mechanisms can also be found in commercial scheduling products such as RSS from Northern Computing Systems, Parry Sound, Ontario.

Machine Loading Algorithm

- B22. search the scheduled queue of the machine for the
 first open time slot on or after the job start date**
- B23. assign this job to the selected slot**

The implementation of scheduling algorithms in industry, and the current research work, seems to be moving in the direction of more simplistic scheduling algorithms in favour of execution speed and schedule feasibility. One of the simplest machine loading algorithms is that of 'forward load to finite capacity'. This mechanism simply says that a job is assigned to the first available time slot which is large enough to accommodate the job. This time slot must be after the current date and on or after the job start date. Each machine was loaded up to its specified capacity for each day scheduled. A significant amount of literature exists to support the use of this mechanism such as the papers by Hastings, Marshall et al. [46, 47, 48, 70], McKay et al. [74, 75]. The majority of work referenced from these authors is based on industrial implementations of shop schedulers and therefore can be taken as evidence of the validity of this algorithm for comparison purposes.

Job Synchronization

- B31. if (the job end date > parent start date from MRP)
 then**
- B32. push the parent start date (*normally shows up as
 a stockout)**

The infeasibilities which arise from the lack of synchronization between the materials plans and the shop schedule was described as part of the problem statement and algorithm description. The baseline scheduling module was written to count the number

of such synchronization infeasibilities which arose. It was also fed with the information necessary to avoid them. This feature was implemented to allow the scheduler to more closely simulate what would actually occur in a job-shop. When materials are unavailable the succeeding jobs get pushed out in time while waiting for the necessary components. To allow a fair comparison of the inventory carrying costs between the baseline and the coherent planning model it was important to closely model the actual execution in the shop.

6.3.2. Coherent Planning Model.

6.3.2.1. Overview

The system created to run the coherent planning algorithm also consists of two parts. The basic logic implemented in the 'real' MRP system described above was used to pre-process the demands passed to the algorithm. This pre-processing was used to accommodate the existence of inventory and planned orders in the test data set used as well as to demonstrate the ability of the algorithm to accommodate these factors. The pre-processing pass will be described in chapter 8.

The coherent planning algorithm was implemented in full and is driven by the data produced in the MRP pre-processing pass. The cost-assessment logic implemented in the comparison baseline version of the software was duplicated in this piece of code for consistency in analyzing the resulting production plans. This component was developed using the shop scheduling code of the baseline system as a starting point. Similar to the baseline implementation, this part of the system required approximately 3000 lines of C code of which some 40% was different from that used in the baseline.

A high-level, pseudo-code overview of the structure of the software is given below. The C preceding the line numbers denotes code for the Coherent algorithm. Note that the demand file contains **only the net independent demands** sorted by due date in **descending time order**:

```
C1.  initialize data structures:
C2.  load items
```

```
C3.  load machines
C4.  load routings
C5.  load bill of material

C6.  set the min_date to the latest job date - the schedule
     window size
C7.  read a demand from the demand file

C8.  while (not at the end of the demand file)
C9.      while (the demand due date > min_date)

C10.     pass the demand to the materials planning logic for
         the item
C11.     if (the demand can't be grouped with the current
         order) then
C12.         release the current order
C13.         enqueue the jobs required for this order on the
         machines in the shop.
C14.         flag the END operation of this job sequence as
         ready to schedule.
C15.         use the current demand to start a new order
C16.     else
C17.         group this demand with the current order
C18.     endif

C19.     read the next demand from the demand file
C20. endwhile

C21. initiate a shop scheduling pass
C22. continue_scheduling = TRUE
C23. while (continue_scheduling = TRUE)
C24.     continue_scheduling = FALSE
C25.     for (each machine in the database)
```

```

C26.      while (there is at least one job ready to be
           scheduled on this machine)

C27.      search the waiting queue for the highest
           priority job ready to schedule
C28.      search the scheduled queue of the machine for
           the first open time slot on or before the job
           due date
C29.      assign this job to the selected slot
C30.      remove this job from the machine wait queue
C31.      continue_scheduling = TRUE

C32.      if (this job has a successor)
C33.          flag the succeeding job as ready to run, due
              on the scheduled job's start date
C34.      else
C35.          if (this item has any children)
C36.              explode the dependent demand to the child
                  item(s)
C37.              set the child due dates to this job's
                  start date
C38.              pass the child demands to the materials
                  planning logic
C39.          endif
C40.      endif
C41.      endwhile
C42.  endfor
C43. endwhile
C44. subtract the planning window size from the min_date
C45. endwhile

C46. flush remaining orders from materials planning:
C47. for (each item)
C48. if (an order is in progress) then

```

```

C49.   enqueue the jobs required for this order on the
        machines in the shop.
C50.   flag the END operation of this job sequence as ready
        to      schedule.
C51. endif
C52. endfor

C53. schedule any remaining jobs and explode dependent
        demands:
C54. while (any machine has jobs remaining to be scheduled)
C55.   initiate a scheduling pass
C56. endwhile

C57. assign a cost to the plans generated:
C58. determine the number of orders and their cost.
C59. determine the number and cost of part-inventory days
        carried
C60. determine the number and cost of machine setups

```

6.3.2.2. Materials Planning in the Coherent Model

The materials planning logic implemented in this system was put in place as a single routine which was called for the processing of independent and dependent demands. This routine is accessed from two places. The first occurs during the input and processing of independent demand from the input demand file and the second during the explosion of demand from a parent item. The details of the algorithm are given with references to the corresponding lines of pseudo-code.

Lot Sizing

```

C11.   if (the demand can't be grouped with the current
        order) then
C12.     release the current order
        .

```

```

•
•
C16.    else
C17.        group this demand with the current order
C18.    endif

```

Line C11 of the pseudo-code refers to the execution of the lot-sizing logic used to determine if a new requirement should be grouped with an order currently being planned. For this thesis a cost-based lot-sizing algorithm was deemed to be appropriate and interesting. The reverse time Part-Period Balancing algorithm (PPB) as discussed in section 5.2.4 was used since the forward time PPB algorithm is commonly found in industry and was used in The Company's MRP system.

This algorithm compares the cost of carrying the number of items currently grouped in the order quantity in inventory back to the date of the new requirement. Since the new requirement defines the order due date (if included in the group) it does not incur any carrying costs. The total order cost is compared to the administrative overhead cost of placing a new order. New requirements continue to be grouped with the order, thereby moving the due date back in time, until the total carrying cost for that order exceeds its overhead cost.

Requirements satisfying the lot-sizing conditions are grouped with the current order (line C17) otherwise the current order is released and a new order begun (lines C12 through C15).

Explosion

```

C36.    explode the dependent demand to the child item(s)
C37.    set the child due dates to the parent job's start
        date
C38.    pass the child demands to the materials planning
        logic

```

The explosion process is a simple multiplication of the parent order quantity by the child per parent multiplier with one small exception. In order to better serve the lot-sizing and shop scheduling logic, the individual requirements grouped into the parent order quantity

are exploded to the child. This information is normally filtered out of the data passed from MRP to the scheduler. The due dates of the demands are modified to reflect the parent order start date as determined through scheduling in the shop. This detailed breakdown of the requirements allows the lot-sizing logic to satisfy the parent order quantity in smaller lots if this is profitable. Furthermore, this information allows the scheduling of the jobs in the shop to intelligently use such scheduling techniques as lot-splitting or overlapping* .

The technique of exploding individual demands incorporated in an order also allows a level-by-level pegging of the source of requirements. In most MRP systems this feature is exceedingly difficult to implement and invaluable where it does exist.

Final Order Completion

C46. flush remaining orders from materials planning:

C47. for (each item)

-
-
-

C52. endfor

MRP systems typically make provision for flushing out any orders currently in the planning stage when the end of the horizon has been reached and grouping them into one final order. Since this module incorporates the typical MRP order planning logic this function is implemented as well. Once the last independent demands are processed against a part any outstanding orders being planned are released for scheduling in the shop.

* **Lot-splitting** is the process by which a batch is broken up to allow faster processing, usually on parallel machines. **Overlapping** involves the moving of partial batches to the next operation in sequence while the current operation continues.

6.3.2.3. Shop Scheduling in the Coherent Model

The principles outlined for the coherent planning model are intended to form a framework which can accommodate a variety of shop scheduling techniques in a manner that can be integrated with materials planning. This section details some of the implementation of the shop scheduler in the coherent planning model, with reference to the pseudo-code from section 6.3.2.

Scheduling Window

C6. set the min_date to the latest job date - the schedule window size

In a similar manner to the scheduling window used in the baseline module, a window was applied here to control the iteration between the processing of demands and passes through the shop scheduler. In section 5.2.4 the problem of sequencing the demands passed to the materials planning logic required that the schedule window size be as large as the largest level-to-level lead time. For the database used in this testing the schedule window size was set to seven days. Some experimentation was conducted with larger window sizes with no conclusive results.

Job Enqueuing

C13. enqueue the jobs required for this order on the machines in the shop.

C14. flag the END operation of this job sequence as ready to schedule.

The problem of selecting the schedule window size is associated with the method of enqueuing jobs. In order to control the sequence in which jobs leave the shop and are passed to the materials planning logic a START operation was added to the beginning of each item's routing and an END job added to the end. The use of the END job is not essential for the algorithm and was included due to the selected implementation approach. The START operation, however, provides a control point for all demand leaving the shop (remembering that scheduling is done in reverse time sequence). A priority rule of latest due date was applied to this operation which, in conjunction with

the schedule window size, satisfied the sequencing requirement of demand passed to the materials planning logic.

Job Priority Mechanism

C26. while (there is at least one job ready to be scheduled
on this machine)

C27. search the waiting queue for the highest priority
job ready to schedule

In the final results analysis, the same scheduling priority rules were implemented in the coherent planning module as in the baseline comparison module. This was a two level, Earliest-Due-Date (EDD)/ Shortest-Processing-Time (SPT) combination with a basis in several studies and commercial packages. Some experimentation was conducted with other priority mechanisms such as Total-Work (TWK) to determine their effect on the plans produced. Once again, these side experiments were inconclusive and deemed to be sufficiently outside of the scope of the thesis to warrant their discontinuation. Chapter 9 discusses this, and other areas for further experimentation.

Machine Loading Algorithm

C28. search the scheduled queue of the machine for the first
open slot on or before the job due date.

C29. assign this job to the selected slot

The machine loading algorithm used in the coherent planning model is a reverse time adaptation of the 'forward load to finite capacity' mechanism described previously. In this implementation jobs were loaded onto the machines in the first time slot on or before the job due date, moving backward in time. The load assigned to any particular day on the machine was limited to the machine capacity. This mechanism is very similar in concept to the 'backward loading' technique described by Marshall [70] and implemented as optional loading techniques in WATPASS and RSS. The traditional backward loading method refers, however, to the practice of assigning time moving from the MRP assigned job due date backward. This mechanism differs from the coherent planning implementation in that the systems cited above proceed to schedule the next job in time, that is, the succeeding job as opposed to the preceding job as in this model.

Job Synchronization

37. set the child due dates to the parent job's start date

The synchronization of successive jobs in the coherent planning model is one of the objectives and significant advantages of this algorithm. In maintaining the information required to associate jobs, not just within one item's sequence but across levels of the product structure one eliminates the infeasibilities described previously between the material plans and production schedules. Furthermore, this tight synchronization eliminates the early ordering of child components and reduces carrying costs resulting from the MRP/shop scheduling decoupling which uses inventory as a buffer. The heavier the load on the shop in the traditional planning approach, the greater the number of infeasibilities and the greater the inventory carrying costs. The details of the problem addressed here were given in section 4.2

6.3.3. Results Analysis

The last phase of processing for both the baseline module and the coherent planning module produced a cost analysis of the plans generated using the cost criteria specified in section 5.2.7. This section will briefly describe the details of how these cost criteria were applied to the resulting schedules. This explanation should provide a better basis for understanding the results presented in chapter 7.

Order Overhead Costs

C58. determine the number of orders and their cost.

The total administrative costs incurred for the planned orders generated by the planning algorithms is determined through a simple multiplication. The total number of planned orders created is multiplied by the overhead cost per order to give the total overhead cost. Planned orders for items at all levels of the product structure were included in the order count, whether for manufactured items or purchased components.

Inventory Carrying Costs

C59. determine the number and cost of part-inventory days carried

Inventory carrying costs were assessed against all items held in inventory for one full day or more. Once again, these costs were applied to items at all levels of the product structure (manufactured or purchased) since both represent an investment in materials, labour, and storage space. This practice of applying carrying costs to all items is standard in MRP systems and industry in general.

The number of days an item is carried in inventory is calculated as the number of days the item exists in the shop in its raw state (not assembled in another item) **exclusive** of the day it was received in the shop and **exclusive** of the day on which it begins processing for inclusion in a higher level assembly. This definition of carrying days includes the time the item itself spends in assembly and therefore covers the cost of work-in-process inventory. There are two slightly different calculations that apply to the determination of carrying days, these are:

purchased component :

$$\# \text{ days} = \sum_{\text{purchase orders}} (\text{Parent item start date} - \text{MRP due date (i.e. date received)} - 1)$$

manufactured component:

$$\# \text{ days} = \sum_{\text{shop orders}} (\text{Parent order start date} - \text{start date of first operation for this order} - 1)$$

This determination of the number of days an item is carried gives a heavier weighting to the higher level assemblies in the system since the charge is transferred to the higher level assembly on the day the operation sequence to build it begins. The application of this rule is consistent across both the baseline and coherent planning modules.

The final cost of carrying an item in inventory is calculated by multiplying the item's per part per day inventory carrying cost against the total number of inventory-part-days charged to it.

Machine Setup Costs

60. determine the number and cost of machine setups

Machine setup costs represent the processing time lost and labour costs incurred to re-configure a piece of equipment to process a new batch of material. Many shop scheduling algorithms attempt to sequence like operations in succession or combine batches of the same item to reduce setup costs. The number of setup costs incurred is greatly affected by the materials lot-sizing algorithm since the selected policy can decrease the number of individual batches processed in the shop.

For the purposes of this study a machine setup cost was charged for each batch processed at an operation. This application of the cost does not recognize the fact that like operations in succession may not incur an additional setup cost. This approach was selected for simplicity in the shop scheduling algorithm and results analysis and was applied uniformly across both planning modules.

6.3.4. Input Data Files

There were five data files loaded to drive the baseline scheduling module, four of which were used in the coherent planning module. With reference to the pseudo-code, the record structure of these relational format files is given below.

B/C2. load items

record structure: ITEM NAME, CARRYING COST, ORDER COST

sort keys: None

B/C3. load machines

record structure: MACHINE NAME, SETUP COST, CAPACITY/DAY

sort keys: None

B/C4. load routings

record structure: ITEM NAME, MACHINE NAME, SETUP TIME, RUN
TIME/ITEM, OPERATION#

sort keys: ITEM NAME, OPERATION#

B/C5. load bill of material

record structure: PARENT ITEM NAME, CHILD ITEM NAME, CHILD,
QUANTITY/PARENT, CHILD LEAD TIME

sort keys: PARENT ITEM NAME

B/C6. load order breakdown*

record structure: ITEM NAME, ORDER NUMBER, DEMAND ORDER NUMBER(S)
(from parent orders)

sort keys: ITEM NAME

*the order breakdown file was used for demand pegging in the baseline module only.
Pegging in the coherent planning module was done dynamically by the algorithm.

7. Experimentation & Results

7.1. Source of the Data

The data used in this study was extracted from the live materials planning and shop scheduling databases of The Company. The full database in use by The Company contains more than 30,000 active assemblies, sub-assemblies and components. From this database several slices comprising twenty-seven items were selected. These slices represent several lower 'branches' of the product structure and were deemed to be representative of the many assemblies in the tree. The average depth of the branches selected is four levels. The diagram below represents a conceptual view of how branches can be clipped from the product structure and still maintain usable data. All data associated with each item was extracted from the host database.

Figure 7-1: An example of the manner in which complete sub-branches of a product structure can be clipped from the whole product structure.

To create an interesting problem from the point of view of the coherent planning algorithm it is desirable to have various items competing for the factory resources. The branches selected from the product structure therefore contain items with overlapping manufacturing operation sequences. The average number of operations in the job sequence (or routing) for the selected items is eighteen, with some items visiting the same operation more than once. The complete routings for the twenty seven items selected required the extraction of 117 machines from the database. Given the average of eighteen operations per routing this means that each machine is used in $(18 * 27 / 117 =)$

4 routings on average.

The system was driven by 237 independent demands spanning a period of approximately twenty-four months. These independent demands were against top level assemblies and some sub-assemblies as is typical of this type of manufacturing environment. The planning horizon is rather long for typical shop scheduling algorithms. Materials planning, however, frequently addresses periods between one and two years in length. From this set of independent demands, approximately 215 dependent planned orders are generated. The specific number of dependent orders created varies from run to run depending on the planning parameters such as carrying costs and setup costs. Ten of the items selected from the database were assigned a lot-sizing policy of lot-for-lot DISCRETE on the host MRP system, that is, for each demand received against the item a planned order was generated for the same quantity. The balance of seventeen items were ordered on a part-period balancing (PPB) policy.

7.2. Modifications in the Data

In the interest of maintaining a dataset representative of actual industrial conditions, very few changes were made to the extracted data. There were two significant alterations to the data to make the problem more interesting from an algorithmic point of view. This section overviews these changes and the reasons for them.

Inventory Carrying Costs

The host database used a single figure for the per part per day inventory carrying costs for all items. It is unclear whether this was implemented as a result of a failure to collect and enter the necessary data, an effort to manipulate the MRP logic, or as being representative of The Company's accounting policies. Traditional inventory carrying costs represent the opportunity cost of capital invested in inventory as well as the cost of maintaining warehouse space to store the material. The Company's fixed carrying cost acknowledges the warehousing costs of inventory but it does not acknowledge the difference in value of the various materials which might exist in inventory.

As an example of the misconception represented by a fixed carrying cost, consider an assembly, A which is built from two components, B and C. For each day B or C is carried in inventory a cost of η is incurred. When B and C are assembled to create an A, however, A's carrying cost is still only assessed as η . This fails to recognize the fact that both a B and a C are still being carried in inventory (in A) as well as the added value of the labour or machine time required to make A.

In an effort to make the carryings costs more representative of those in manufacturing operations, the fixed cost was modified for each item in the dataset by multiplying the cost by the number of levels the item is from the bottom of the product structure. That is, if an item was purchased it was assigned the fixed cost. If an item was a first level assembly (built from purchased components) it was assigned twice the fixed cost. If the item was built from second or lower level assemblies it was assigned three times the fixed cost, and so on. This method of assignment of carrying costs is not a restriction of the algorithm but merely a convenience used to arrive at more representative carrying costs.

This change to the inventory carrying costs was implemented in both the baseline MRP system being used for comparison purposes and the coherent planning module so as not to skew the results.

Machine Capacity Variation

The daily capacity of the machines in the factory was extracted as part of the test dataset. This dataset represents a very small portion of the actual products built in the factory and a small portion of the load on it. By extracting only a portion of the demand in the factory, the load from other products has been eliminated and a dramatic over capacity situation exists.

The real problem under study in this thesis is the inability of the shop scheduling algorithm to modify the materials plans in response to shop load. The heavier the load on the shop the more likely it is for the problems described in chapter 4 to manifest themselves.

Given the preceding explanation, the overall capacity of the factory was controlled in the analysis of the performance of the algorithms. This was accomplished by uniformly

setting all machine capacities at some percentage (< 100) of their actual capacity for each planning run. Recognizing the inherent over-capacity situation arising from the extraction of data, the shop algorithms were tested using shop capacities ranging from 2% to 50% of actual capacity.

The items selected for experimentation were chosen partially on the basis of their use of common machinery. Although only (27/30,000) 0.1% of the items in the entire database were selected, more than 0.1% of the demand on this set of machines is preserved since these items were specifically chosen for their use of common machinery. The extremes of the capacity scale were selected through experimentation. At 2% of true capacity the shop was placed in an extreme under capacity situation. At 50% of the true capacity there were few scheduling conflicts or jobs completed past their due dates. This suggests that some value between 2 and 50% of the actual load on the selected set of machines is derived through the items selected for experimentation.

7.3. Results of Experimentation

The analysis of the coherent planning algorithm was conducted under several different scenarios of demand and settings of the planning parameters. The quantitative results of this analysis are presented in the following sections. The data corresponding to the many graphs in this section can be found in the appendix.

7.3.1. Analysis of a Shop with a Bottleneck

The first data scenario examined was inherent in the dataset used for testing. One particular branch of the product structure was heavily loaded compared to the other branches. One of the machines required for an item in this branch was also in high demand for items in other branches of the product. The end result being that the overloaded machine was scheduled to run constantly while many jobs were pushed out due to an inability to acquire the needed time slot on the machine. The tables and graphs in this section represent the output of the planning runs under these conditions.

Note that in figures 7-2 through 7-22, 'Baseline' refers to the comparison baseline algorithm while 'Coherent' refers to the coherent planning model.

Figure 7-2: Carrying cost comparison under bottleneck conditions.

The difference in carrying costs illustrated above underscores the ability of the coherent planning algorithm to respond to and adjust for changes in factory capacity. The baseline result must work from fixed material plans which were based on assumptions about the factory's ability to build the product (fixed lead times). As the capacity in the factory increases we see that those assumptions approach reality and the coherent planning algorithm loses its advantage.

The fixed lead times used in the baseline algorithm are set by The Company with some consideration for the normal load in the shop. In that sense one might argue that the process of selecting only a slice of the database for experimentation causes them to be overly inflated. This inflation may well contribute to the higher inventory carrying costs exhibited by the baseline algorithm. Since it is not known how the fixed lead times are derived and what portion of them can be attributed to capacity considerations it is not possible to make explicit adjustments for this potential skew. This issue will be further addressed in the discussion of the total cost results for this scenario.

Figures 7-3,4: Machine setup and order overhead costs under bottleneck conditions.

The order overhead and machine setup costs are closely related in this analysis since, for each batch processed through the factory (each shop order) a fixed number of machine setup costs are incurred per item. The coherent planning algorithm gains an advantage from its ability to adapt to the actual load conditions in the factory. The due date backoffs used in positioning requirements in the horizon for the baseline algorithm are heavily padded, however, (as discussed in chapter 4) and the opportunity to group requirements more efficiently is missed. The coherent planning algorithm uses the actual requirement dates as determined by scheduling the job in the shop and can therefore

group requirements without the extra lead time padding. As the factory capacity increases the advantage of the coherent algorithm diminishes and the results converge.

Figure 7-5: Total costs under bottleneck conditions (sum of 7-2,3 & 4).

Under this scenario the coherent planning algorithm offers an average cost reduction of 26% across the factory load levels. This reduction is composed of two relatively fixed cost components from order and machine setups and a varying component from carrying costs. The cost reduction contributed by the carrying cost component tends to go to zero with higher capacity as shown in figure 7-2. If, in the extreme case, we ignore the carrying cost reduction we are still left with a relatively constant cost improvement of approximately 23%. As discussed previously, there may exist some minor skew in the carrying cost component of the results arising from the use of the fixed lead times assigned for the full database. If we remove any possibility of this skew affecting the results we still see that the cost improvement of the coherent algorithm is significant. This same analysis can be applied to subsequent data scenarios.

Figure 7-6: Number of plan synchronization infeasibilities occurring in the baseline plans generated

As an example of the number of synchronization infeasibilities which arise between materials planning and shop scheduling in a traditional, two-phase environment this data was recorded during the planning runs for this scenario. For this purpose a synchronization infeasibility is defined as that situation which arises when a job is scheduled to begin in the shop but the necessary materials have not been planned to be available. Figure 7-6 illustrates the number of such occurrences at different levels of shop loading (through the adjustment of capacity). The graph shows only the results of the baseline algorithm since the coherent algorithm does not generate synchronization infeasibilities.

7.3.2. Analysis of an Evenly Loaded Shop

The demand data used in the preceding analysis was modified to lighten some of the load on the bottleneck work centre. This lightening of the load required actual manipulation of the independent demands in the shop and does not represent the exact situation that existed at The Company's operation, it is however a scenario of interest. The tables and graphs in this section represent the output of the planning runs under these conditions.

Figure 7-7: Carrying costs under even shop load.

Under even loading the relative shape of the curves is very similar to that under bottleneck conditions. In terms of inventory management under even loading across items the coherent algorithm exhibits the same desirable adaptability as in the previous scenario.

Figures 7-8 : Machine setup costs under even loading

Figures 7-9 : Order overhead costs under even loading

In terms of machine setup costs and order overhead costs the coherent planning algorithm exhibits essentially the same behavior as under bottleneck conditions.

Figure 7-10: Total costs under even loading (sum of figures 7-7,8 & 9).

The overall performance of the coherent planning algorithm is essentially the same under even shop loading conditions as under bottleneck conditions. The average cost reduction achieved by the coherent algorithm in this scenario is 33%. This represents an improvement of 7% over the bottleneck scenario of section 7.3.2. The difference in results is not dramatic since the bottleneck in the previous scenario affected only a few of the many parallel product lines.

7.3.3. Analysis of Modified Cost Criteria

A review of the results in the preceding two sections revealed that the bulk of the improvements achieved by the coherent planning algorithm were based on a dynamic regrouping of material order quantities in response to changes in the shop load level. The basis for these grouping decisions is largely a trade-off of inventory carrying costs against order overhead costs. The dataset used for the next evaluation altered these cost criteria, doubling the inventory carrying costs and reducing the order overhead costs by a factor of four to determine the dependency of the results on the relative magnitudes of these parameters.

Figure 7-11: Carrying costs under modified cost criteria

The change in the magnitude of the carrying costs results in a magnification of the difference between the baseline results and the coherent planning algorithm. The relative shape of the curves has not changed significantly and still illustrates the ability of the coherent algorithm to take advantage of more accurate information when making inventory decisions.

Figures 7-12: Machine setup costs under modified cost criteria.

Figures 7-13: Overhead costs under modified cost criteria.

In a similar vein to the changes in the carrying costs, the change in magnitude of the absolute per order overhead costs is reflected in the relative difference between the total overhead costs of the two algorithms. Once again the characteristic shapes and relative positions of the curves has not changed significantly.

Figure 7-14: Total costs under modified cost criteria (sum of figures 7-11,12 & 13).

The average reduction in cost achieved by the coherent planning algorithm under the modified cost criteria is 32%. It is interesting to note that this cost improvement is very similar to that obtained under the previous scenarios despite the uneven modification of the underlying cost criteria (double carrying costs, one-quarter overhead costs). The large advantage obtained in the first two scenarios on order overhead costs has been eroded due to the decreased magnitude of this component, however, that advantage is recaptured from the inventory carrying costs.

7.3.4. Analysis of Modified Order Policies

In the dataset used for these analyses only seventeen of the twenty-seven items in the are ordered using a cost-based lot-sizing policy. For some reason a decision had been made to order the remaining ten items on a lot-for-lot discrete basis; a decision which used

criteria outside of the scope of traditional material planning logic. In the previous scenarios the materials planning logic implemented the same lot-sizing policies (in reverse time) as assigned in the host system to allow for a completely unprejudiced comparison. The dataset used in this scenario has been modified to order all items on the reverse-time PPB policy for coherent planning as compared to the host assigned policies of the baseline.

Figure 7-15: Carrying costs when applying cost-based lot-sizing to all items.

Figures 7-16: Machine setup costs under modified order policies

The carrying cost performance under pure cost-based lot-sizing, as shown in figure 7-15, exhibits two interesting characteristics not seen in the previous scenarios. First, the coherent algorithm is largely unaffected by the shop load conditions, adapting the order grouping of all items across the various levels of shop load. While this is generally true when compared to the baseline algorithm in other scenarios, it is also more level under heavy shop load here when compared to its own performance in other scenarios. Second, as the shop capacity increases the coherent algorithm no longer converges with the baseline but incurs higher carrying costs instead. The explanation for this second aberration must be given when considering the order overhead costs.

Figures 7-17: Order overhead costs under modified order policies

The cost of the external decision to use discrete lot-sizing becomes apparent when one examines the machine setup and order overhead costs incurred as a result. The ability of the coherent planning algorithm group orders and make informed inventory decisions has been increased by applying these rules to all items. The results from figures 7-3 and 7-4 are included as reference points to highlight the improvements. In figure 7-15 we saw that the coherent algorithm actually incurred higher carrying costs in order to perform more extensive order grouping than the baseline. In these graphs we see the benefit of that action in terms of dramatic reductions in the setup and overhead costs. The key

source of the improvement is the ability of the coherent algorithm to make decisions while considering the actual shop conditions while the baseline algorithm must plan using a set of external, confining decisions.

Figure 7-18: Total costs under modified order policies (sum of 7-15,16 & 17).

The dramatic improvements in the setup and overhead cost components of the analysis are clearly reflected in the total. The trade-off of higher carrying costs under circumstances appears to have been well founded in achieving significant improvements across the range of shop load conditions tested. The average cost reduction achieved under these circumstances is 79%.

7.4. Experimentation with Due Date Setting Algorithms

The coherent planning algorithm is predicated on the use of feasible due dates for the independent demand placed on the system. It is not possible, however, to determine if the due dates are feasible before the scheduling has taken place unless an extreme under or over capacity situation exists.

As discussed in chapter 3, the problem of due date setting and predictability is under study by several researchers and has clearly been identified as a problem area of interest. Several algorithms have been put forward for assessing due dates however the extent of their use in industry is unknown. It is clear that the problem exists in industry in that the 'past due' time bucket for a number of items in any large manufacturing organization contains actions that had to have been performed in the past to meet the current delivery schedules. Given the reliance of the coherent planning algorithm on this criteria some experimentation was conducted on the setting of feasible due dates.

The model selected for due date setting was taken from a paper by Dumond & Mabert [27] which examined four methods of due date setting. Their results suggest that their Scheduled Finish Time (SFT) algorithm produces the best results in terms of job lateness variability and job tardiness. The basis of this algorithm is the use of a first pass scheduling mechanism which goes through the entire shop scheduling process to assign due dates to the jobs. This method makes use of the most accurate information regarding the shop since it creates a fully detailed shop schedule. The delivery dates resulting from this scheduling pass are then fed back into the materials planning algorithm as the new due dates and the process is repeated. This mechanism closely resembles the suggested approach for integrating materials planning and shop scheduling as suggested by many vendors of finite scheduling software (and as discussed in section 3.1.1) with the exception of the iteration back through the scheduling process.

In testing this algorithm an extension to the work of Dumond & Mabert was added. The inflexibility of lead times in materials planning is another source of error and skew between the materials plans and shop schedule. In the 'due date' scheduling pass the full shop schedule is developed down to the detail level and the actual elapsed time required from the start of a job to its completion can be determined. From this same scheduling operation, therefore, the mean lead time required to manufacture a batch of an item was extracted and re-loaded into the materials planning process along with the adjusted due dates.

In fact, a third version of the software was created specifically for the evaluation of due date setting algorithms. This version referred to as the 'Date' module was largely built from the baseline software with two modifications:

- the priority rule applied in selecting jobs for shop scheduling was the Total Work Content (TWK) rule as described in the appendix, section A4.
- succeeding jobs were flagged to begin on the day following the completion of the child components. The jobs were not made to wait until the MRP scheduled start date as in the baseline scheduling algorithm.

The Date module is a fully functional version of the software, producing final schedules and plans similar to those of the baseline. It also produces modified due dates and mean order lead times which are then fed back into the materials planning processes to generate the new schedules based on these date modifications.

The results of some of the experimentation conducted into this aspect of the problem are shown in figures 7-19 through 7-22 below. The schedule results from the Date module are shown for comparison purposes only. The data corresponding to these figures can be found in the appendix.

Figures 7-19,20: Days late and total costs with no due date adjustments.

In figures 7-19 through 7-22, the total days late was calculated differently depending on the algorithm. For the baseline and date planning algorithms, the number of days late was taken as the number of days each independent demand was satisfied after its MRP specified due date. Since, however, in the coherent algorithm the completion date is guaranteed to meet the due date, the number of days late is measured as the total number of days each demand was scheduled to start before the 'today' date. That is, the number of days late for the coherent algorithm was measured as the sum of the number of job-days scheduled in the past.

Figures 7-21,22: Days late and total costs with due dates adjusted.

The total cost diagrams (figures 7-20 & 7-22) for the baseline and date algorithms exhibit similar behavior. Furthermore, between the two data scenarios (no due date setting and after adjustments) there is little change in the total cost curves between the baseline and coherent algorithms. This observation is intended to illustrate the utility of the coherent algorithm in an environment that supports due date setting. This example should also support the suggestion that the results presented in this chapter can be considered representative of the algorithm's behavior in such an environment.

The material presented here belies the actual amount of work applied to the problem of due date setting. After some further effort it was concluded that the coherent planning algorithm itself is a useful tool, whether it is coupled to a due date setting algorithm or not. The problem of developing and testing appropriate due date setting mechanisms is sufficiently complex to merit a separate, in-depth study.

8. Discussion

8.1. The Significance of Due Dates

In several places throughout this thesis, reference has been made to the issue of due dates on independent demands. In the initial explanation of the algorithm it was stated that there is an implicit assumption of feasible due dates. As with any manufacturing planning tools which work backward from due dates, be they material planning tools such as MRP or specific implementations of shop schedulers, there is a possibility that some planning will be done in the past. Superficially, therefore, it is quite easy to dismiss this algorithm as worthless since there is no guarantee of due date feasibility in the input data, particularly in complex job-shop environments.

In support of the viability of this algorithm, and to put the due date issue in perspective, a review of current research into this problem was presented. This material highlighted several research efforts which address the due date problem specifically and have demonstrated a number of viable algorithms for assessing feasible due dates. As part of this research, the coupling of such due date setting algorithms with the coherent planning algorithm was tested. The results of these tests were given in section 7.4 and show that the behaviour of the coherent planning algorithm is roughly equivalent under modified due dates. While these tests were not exhaustive they were conducted under conditions of 'real' data, planning parameters, and cost analysis criteria. In terms of due date performance, the coherent planning algorithm was shown to far outperform the baseline algorithm and, in fact, to avoid 'past due' planning (as discussed in section 5.2.6) entirely, except under extreme shop loading conditions.

It is important to remember in the evaluation of this algorithm that one of its principal objectives is to remove the synchronization infeasibilities that arise in and between materials planning and shop scheduling procedures. In all cases studied the algorithm was shown to meet this objective and to do so at an average of 30% less cost. In the worst case, it will be necessary to couple the coherent planning algorithm with a due date setting procedure to remove the 'past due' infeasibilities, assuming that one is not already in use in the target manufacturing environment. Given the logic and interaction of materials planning and shop scheduling systems in use today, there is no reasonable way

to remove the infeasibilities they generate. In this sense, therefore, the alternatives are to accept and deal with the current planning systems, their costs and infeasibilities, or to implement an algorithm such as that presented in this thesis.

8.2. Extensions for Practical Use

Throughout the development of this thesis there have been many issues which have been skirted in discussing the specifics of the algorithm in favour of addressing them here. These issues are common features of materials planning and shop scheduling systems and are essential capabilities in any new system one might propose.

The full implementation and test of these features is beyond the scope of this thesis. While the algorithm has been constructed to accommodate these issues, they have not all been fully implemented and tested as part of this thesis.

The material presented in this section, therefore, is meant to provide clarification of the issues and the manner in which the algorithm can be structured to accommodate them. In addressing a range of topics associated with the core production planning problems one can better see the practicality of the algorithm.

8.2.1. Inventories

No manufacturing facility operates without inventories. Whether they exist in storerooms, warehouses, or in bins on the shop floor, all manufacturers must maintain some minimum quantity of the raw and intermediate materials they require to produce their end products. During the development of this thesis the issue of inventories was deferred for clarity of the overall concept.

As part of the work conducted in this research, a pre-processing materials planning pass was implemented and tested for the purposes of consuming inventory and open orders. This pass takes the form of a 'mini-MRP' in which enough forward time processing of the material requirements is performed to consume all available materials. This aspect of the materials planning problem does not fit the coherent planning algorithm directly due to the well founded assumption that a manufacturer will plan to consume all inventory of an item before ordering more. This assumption goes to the issue of cost effectiveness

(maximization of profit through minimization of cost) since there are overhead costs associated with carrying inventories. In coherent planning the materials plans are developed working backward in time without complete knowledge of demands which may arise earlier in time. One can therefore not determine the point at which to stop ordering more materials and begin to consume inventory since the requirements are unknown. The pre-processing of demands down to the net demand only eliminates this problem.

As was done in this implementation, it is entirely reasonable to use the existing MRP system for this pre-processing pass. By applying a lot-for-lot, non-intervalized (i.e. lead times of zero) MRP run one is left with net demands. The implementation of this enhancement to the algorithm requires some additional logic for the management of the pre-exploded independent and dependent demands and their interrelations. This additional effort was not significant and should not be prohibitive.

8.2.2. Work in Process

Manufacturing planning is done on a repetitive basis. Manufacturers do not generate a production plan for a period of time, execute the entire plan, and then generate a new one. Instead they implement the mechanism of 'rolling schedules' as discussed by Baker [3], Baker & Peterson [5], and Blackburn & Millen[10] in which a plan is generated for periods one through n . The plan is executed for period one and a new plan is generated for periods 2 through $n + 1$. This process implies that production planning systems can not begin with an empty shop and empty storerooms, instead they must plan around a large amount of work-in-process.

The algorithm presented in this thesis has been presented from the standpoint of an empty shop and no inventories (with the exception of the preceding inventory discussion). The reality is that one must take the existing factory status into consideration when generating new plans and allow for the machines already in use or the materials already manufactured. The accommodation of work-in-process in the coherent planning algorithm is quite easy and natural. As part of the dynamic data structures used in scheduling the shop one must maintain a current schedule for each machine or operation in it. As new jobs are assigned to the equipment, the block of time allotted to them is removed from the list of available time slots. In addressing the issue

of work-in-process, one must simply pre-load the machine schedules with those jobs, or blocks of time, already allocated and thereby avoid the creation of conflicts with new work.

This approach to work-in-process is very common in shop scheduling algorithms today as is the parallel of inventories and open orders in materials planning.

8.2.3. Schedule Stabilization

In the discussion of optimizing solutions to the materials planning and shop scheduling problems in general and during the analysis of some lot-sizing techniques in chapter 3, the issue of schedule stabilization was addressed. This issue was presented more from the perspective of schedule 'nervousness' in that minor changes to the input data frequently results in major changes in the resulting plans. In the rolling schedule planning environment this problem is aggravated by the frequent regeneration of production plans in an effort to stay up to date.

To overcome this problem, many materials planning and shop scheduling system use a feature referred to as a stabilized horizon. Simply put, the stabilized horizon identifies a period of time from the current date in which previous production planning decisions are not to be changed. This feature is easily implemented in the coherent planning algorithm in the same way that work-in-process is accommodated. Any fixed materials decisions can be treated as inventory or open orders and can be consumed during a pre-processing pass. Shop scheduling decisions should also be considered to be foregone conclusions and their corresponding time allocated on the machines before any further shop scheduling begins.

8.2.4. Demand Pegging

One feature found in many materials planning systems at various levels of functionality is a demand pegging capability. The underlying principles of the MRP model cause the source of demands to be masked as they propagate down the product structure. After several levels of explosion and demand groupings, the top-level source of demand on a purchased component is not usually identifiable. The ability to tie demands back to their source is valuable in a job-shop environment due to the volatility of customer orders. If a customer requests any sort of change to an order it is virtually impossible to determine

the impact of that change on lower level components without the ability to peg demand or else regenerating the production plans.

The tight coupling of the materials planning and shop scheduling functions in the coherent planning algorithm supports, and in some cases requires, a level-by-level pegging capability. To provide flexible lot-sizing and production scheduling decisions at the lower levels of the product structure, the parent level source information is retained against the demand. With the implementation of a tool to scan through the levels of pegged demand one could easily follow requirements from their source to their bottom level supply or back.

8.3. Complexity of the Algorithm

A thorough examination of the complexity of the coherent algorithm has not been conducted. It is not known with certainty, therefore, how its execution time will increase with an increase in the size of the database to which it is applied. Some cursory analysis of the actual execution times of the prototype software was conducted and is detailed in appendix A10.

An examination of the structure of the prototype software does give some indication of the sensitivity of the algorithm's performance to the relative size of the input database segments. The software is structured to iterate between the planning of material requirements and the generation of shop schedules, each contributing to the computing load in isolation.

The materials planning logic is invoked once for each demand in the system, independent or dependent. The total number of demands is a function of the number of independent demands in the database. The number of dependent demands tends to increase with the number of bill of material relationships in the database. This effect is due to the explosion of independent demands through the product structure which has a multiplying effect as one parent may explode to several children. The number of items in the database tends to decrease the total number of demands by reducing the average number of bill relationships per item and thereby simplifying the product structure.

The shop scheduling logic is applied once for each operation which must be scheduled in the shop. The total number of operations to be scheduled is a function of the total

number of demands in the system as described for the materials planning logic. Each demand must be processed through each work centre on the routing for the item required. The number of machines on the routing for an item therefore has a multiplicative effect similar to the number of bill relationships in that the larger the routing, the more operations per demand. Extending this analogy, the greater the number of items in the database, the lower the average number of routing steps per item and hence the lower the total number of operations.

The complexity of the materials planning logic process itself in this algorithm is minimal. It involves only the lot-sizing of demands since the explosion of demands is incorporated in the shop scheduling logic. The application of most lot-sizing techniques (such as those described in chapter 3) is trivial. The shop scheduling logic is significantly more complex than that of materials planning since it must apply a number of searching, sorting, and prioritization algorithms to queued and scheduled jobs for each new job-machine combination. Even in its simplest form, the computational load of the shop scheduler far outweighs that of materials planning on a per application basis.

This brief analysis suggests that the complexity of the shop scheduling component of the overall algorithm is much more significant than that of the materials planning component. Given that materials planning is relatively well understood while the shop scheduling problem has been shown to be NP-complete, this result is not surprising. The computational performance of the coherent algorithm therefore depends heavily on the characteristics of the shop scheduler used with it. In order to achieve reasonable performance on a large database it will be necessary to use fast shop scheduling techniques.

9. Summary

9.1. Benefits of the Algorithm

The original objective of this thesis was to develop, implement and test a practical framework which would better meet the needs of existing manufacturers. There were three principal objectives in suggesting such a framework, they were:

- the elimination of synchronization infeasibilities in the material plans and shop schedules arising from the underlying assumptions in these processes.
- the reduction of costs associated with these assumptions and the inefficiencies they create.
- the development of an algorithm which will be of interest to industry, that is, a practical solution.

9.1.1. Reduction of Costs and Removal of Infeasibilities

The material presented in the chapter 7 supports the attainment of these objectives. In contrast to the baseline planning system, the coherent planning algorithm is guaranteed to generate feasible production plans under all circumstances, given feasible due dates. In the absence of feasible due dates, the algorithm may suffer from 'past due' infeasibilities which can be removed by iteratively adjusting the due dates of the independent demand. Furthermore, the algorithm was shown to generate production plans for a slice of the host database with a minimum of 26% reduction in the cost criteria analyzed and a maximum of 79% when using 2 to 50% of the actual shop capacity. In the manufacturing environments of interest the minimum cost improvement would represent tens to hundreds of thousands of dollars per year without consideration for the intangible benefits of eliminating synchronization infeasibilities.

The use of real data and real planning parameters and constraints reinforces the value of this algorithm as a practical solution to these problems. While the only real evidence that this algorithm fully meets these objectives in an industrial environment would be a full-

scale implementation and long term test, the evidence presented here provides sufficient support for such testing to be undertaken.

A number of lesser benefits can be derived from the implementation of the coherent planning algorithm. By tightly coupling the planning processes, the full spectrum of information underlying both areas is available during decision making. This information is normally filtered out during the transfer of data from the materials planning process to the shop scheduling process. Its availability makes it possible to apply more intelligent decision criteria to such issues as lot-sizing for multi-echelon inventory environments and lot-splitting and overlapping in shop scheduling.

9.1.2. Applicability to the General Industrial Problem

In spite of any benefits that might be claimed in the development of new algorithms for production planning, the likelihood that such algorithms will be adopted in industry lies more in the ease with which these algorithms may be integrated into the existing planning systems. In attempting to address problems experienced in the largest, most complex manufacturing environments one also invites the problem of overcoming the inertia which this complexity creates. This problem is compounded by the intangibility of some of the purported benefits. While it is not difficult to quantify overhead costs and carrying costs in manufacturing, it is extremely difficult to assign a value to the problems caused by a stock out situation or schedule slip.

With this in mind, the algorithm developed in this thesis is couched in terms of "principles" or a framework that should be used to guide the development or modification of a production planning system rather than a hard-and-fast algorithm. This philosophy is in support of the objective of practicality and is intended to protect as much of the existing system as possible. Throughout the thesis where the specifics of a particular algorithm were not dictated by the needs of the coherent planning framework, every effort was made to demonstrate the use of conventional materials planning or shop scheduling algorithms, techniques and structure. This approach was used in the development of lot-sizing algorithms, the interaction and relation of independent and dependent demands, and the explosion of demands in the materials planning process. 'Standard' techniques were selected for priority rules and heuristics and machine loading during shop scheduling. The demonstration that such existing logic and algorithms can

easily be accommodated by the framework eases the impact of the change that would be brought on by the implementation of this algorithm.

The principles themselves more directly address the flow of the planning process and the modifications required to support this change in flow. In the case of materials planning there were two key aspects which required modification. In lot-sizing, while it was not possible to accommodate the current algorithms directly it was shown that many could be easily adapted to fit the time transversal model. The second area of materials planning requiring modification, that of lead-time determination, represents the underlying assumption of materials planning which is at the core of the synchronization problem. In shop scheduling the issue was one of creating a proper decomposition of the scheduling process to allow the interspersing of materials planning stages and the proper consideration for material availability. This too, addresses the underlying assumption of shop scheduling which goes directly to the source of the synchronization issue.

Although not explicitly stated in the explanation of the prototype implementation, the database structure used in testing is very similar to that used by The Company. For the purposes of the prototype the organization of the data was not important since the volume was small. In industry, however, such databases require several hundreds of megabytes of storage and a significant amount of maintenance effort. In structuring the algorithm to work from existing database structures, the need for massive change in this critical and sensitive area is eliminated.

Even with this level of consideration for the preservation of the existing core planning logic and system structure, the implementation of this algorithm will not be trivial. The original objective of minimizing the impact of the required change is difficult to evaluate without actually attempting it. The only mitigating factor in this uncertainty is that it is unlikely that the benefits of the coherent planning algorithm could be realized with lesser changes to the planning systems.

9.2. Conclusions

From the material presented in this thesis a number of conclusions can be drawn, these are:

- the assumptions used in materials planning and shop scheduling systems today are frequently invalid and may result in increased costs to the manufacturer as shown in chapters 2, 4 and 6.
- little research into the resolution of these problems for large job-shop operations appears to exist (chapter 3).
- the solution proposed by Marshall [70] for this problem in smaller manufacturing facilities is not applicable to larger operations (chapter 3).
- the prototype software developed and tested against a 'real' system in this thesis has demonstrated the feasibility of implementing the coherent planning algorithm using existing production planning logic and data.
- the coherent planning algorithm will not only generate production plans free from synchronization infeasibilities, but may reduce the overhead and carrying costs of the manufacturing operations as shown in chapter 7.
- the coherent planning algorithm developed in this thesis and detailed in chapters 5 and 6 appears to offer a viable solution to these problems.
- the coherent planning algorithm meets the original objectives of this research and offers sufficient promise to merit further study and industrial trials.

9.3. Directions for Future Research

Many aspects of the coherent planning algorithm have not been fully explored in this thesis. Those issues left unresolved offer significant opportunities for further study and experimentation. These areas are:

- the development of and experimentation with reverse time lot-sizing algorithms.
- the development of and experimentation with scheduling mechanisms best suited to a reverse time, rolling schedule environment.
- the analysis of a mechanism for selecting the window size to be used during the planning process which can accommodate the necessary lead time restrictions and provide good scheduling and execution performance.
- the analysis of the performance of the coherent planning algorithm using different data sets and performance criteria. Further analysis under different degrees of freedom such as database size and product structure complexity is required.
- the analysis of and strategies for the implementation of this algorithm in existing production planning systems.

9.4. Contributions

9.4.1. Summary of Contributions

The work embodied in this thesis has presented a number of new concepts and techniques not previously reported in manufacturing planning literature, these are:

- the ability to automatically generate fully synchronized and detailed material plans and shop schedules simultaneously.
 - the processing of material demands backward in time (from last to first) using the principles of lot-sizing techniques currently in use in industry.
 - the development of shop schedules by conceptually decomposing an end product into its raw components rather than assembling the components to build the end product.
- the ability to reduce overhead and carrying costs in manufacturing operations through tightly integrating the materials planning and shop scheduling processes.
- the demonstration of a prototype system which generates such plans using manufacturing data and production planning logic currently in use at a large manufacturing operation.

10. References

1. Atabakhsh, H. "A survey of constraint based scheduling systems using an artificial intelligence approach" *Artificial Intelligence in Engineering*, Vol. 6, #2; 1991, pg. 58
2. Ashour, S. "An experimental investigation and comparative evaluation of flow shop scheduling techniques" *Operations Research*, Vol. 18, #3; 1970, pg. 541
3. Baker, K. "A comparative study of flow-shop algorithms" *Operations Research*, Vol. 23, # 1; 1973, pg. 62
4. Baker, K. "An experimental study of the effectiveness of rolling schedules in production planning" *Decision Sciences*, Vol. 8, #1; 1977, pg. 19
5. Baker, K.; Peterson, D. "An analytic framework for evaluating rolling schedules" *Management Sciences*, Vol. 25, #4; 1979, pg. 341
6. Baker, Dixon, Magazine, and Silver "An algorithm for the dynamic lot-size problem with time-varying production capacity constraints" *Management Sciences*, Vol. 24, #16; 1978, pg. 1710
7. Balas, E. "Machine sequencing by disjunctive graphs: an implicit enumeration algorithm" *Operations Research*, Vol. 17; 1969, pg. 941
8. Biggs, J.; Goodman, S. & Hardy "Lot sizing rules in a hierarchical multi-stage inventory system" *Production and Inventory Management*, Vol. 18 #1; 1977, pg. 104
9. Billington, P.; McClain & Thomas "Mathematical programming approaches to capacity constrained MRP systems" *Management Sciences*, Vol. 29, #10; 1983, p. 1126
10. Blackburn, J. & Millen, R. "Heuristic lot-sizing performance in a rolling-schedule environment" *Decision Sciences*, Vol. 11, #4; 1980, p. 691
11. Blackstone, J., Phillips, D. & Hogg "A state of the art survey of dispatching rules for manufacturing job-shop operations" *International Journal of Production Research*, Vol. 20, #1; 1982, p.27
12. Bowman, E. "The schedule sequencing problem" *Operations Research*, Vol. 7, #5; 1959, p.621
13. Brooks, G. & White, C. "An algorithm for finding optimal or near optimal solutions to the production scheduling problem" *Journal of Industrial Engineering*, Vol. 16, #1; 1965, p. 34
14. Carlson, R.; Jucker, J. & Kropp "Less nervous MRP systems; a dynamic economic lot-sizing approach" *Management Sciences*, Vol. 25, #8; 1979, p.754

15. Charlton, J. & Death, C. "A generalized machine scheduling algorithm"
Operations Research Quarterly, Vol. 21, #1; 1970, p.127
16. Chase, R., Aquilano, N. Production and Operations Management
Richard D. Irwin, Inc.: Homewood, Illinois, 1981
17. Connor, J. "Finite scheduling - it works if....."
Manufacturing Systems, Vol. 5, #8; 1987, p.39
18. Conway, R.; Maxwell, W. & Miller Theory of Scheduling
Addison Wesley : New York, N.Y., 1967
19. Cooper, D. "Heuristics for scheduling resource-constrained projects: An experimental investigation" Management Sciences, Vol. 22, #11; 1976, p.1186
20. Czajkiewicz, Z. & Suer, G. "Inventory and production decisions in multi-item, multi-stage manufacturing systems" Computers in Engineering, Proceedings 1988; 1988, p.209
21. Dannenbring, D. "An evaluation of flowshop scheduling heuristics"
Management Sciences, Vol.23, #11; 1977, p. 1174
22. Dannenbring, D. "Procedures for estimating optimal solution values for large combinatorial problems" Management Sciences, Vol. 23, #11; 1977, P.1273
23. Das, D. & Dowsland, W. "A model to examine the effect of loading conditions on scheduling rules in a job-shop environment" International Journal of Production Research Vol.19, #5; 1981, p.577
24. David, H., Lefevre, D. "Finite capacity scheduling with OPT"
Proceedings of the 24th APICS Conference, 1981, p.178
25. Davis, E. "Project scheduling under resource constraints - historical review and categorization procedures" AIIE Transactions, Vol. 5, #4; 1973, p.297
26. Day, J.; Hottenstein, M. "Review of sequencing research"
Naval Research Logistics Quarterly, Vol. 17, #1; 1970, p. 11
27. Dumond, J.; Mabert, V. "Evaluating project scheduling and due date assignment procedures: an experimental analysis" Management Sciences, Vol. 34, #1; 1988, p. 101

28. Durand, C; Silver, E. "New attention for computer scheduling of multi-product plants working batchwise under constraints" *International Journal of Production Research*, Vol. 21, #4; 1983, p.499
29. Elion, S.; Chowdhury, I. "Due dates in job shop scheduling" *International Journal of Production Research*, Vol. 14, #2; 1976, p. 223
30. Fisher, H.; Thompson, G. "Probabilistic learning combinations of local job-shop scheduling rules" in *Industrial Scheduling*, Prentice Hall: Englewood Cliffs, N.J., 1963, p. 225
31. Florian, Lenstra & Rinnooy Kan "Deterministic production planning: algorithms & complexity" *Management Sciences*, Vol. 26, 1980, p. 669
32. Florian, M.; Klein, M. "Deterministic production planning with concave costs and capacity constraints" *Management Sciences*, Vol. 18, 1971, p.12
33. Florian, M.; Trepant, R., McMahon "An implicit enumeration algorithm for the machine sequencing problem" *Management Sciences*, Vol. 17, #12; 1971, p.782
34. Fry, T.; Markland, R. & Phillipboom "Dispatching in a multi-stage job-shop where machine capacities are unbalanced" *International Journal of Production Research*, Vol. 26, #3; 1988, p.1193
35. Garey, R.; Graham, R. & Johnson "Performance guarantees for scheduling algorithms" *Operations Research*, Vol. 26, #1; 1978, p. 3
36. Gavish, B.; Johnson, R. "Fully polynomial approximation scheme for single-product scheduling in a finite capacity facility" *Operations Research*, Vol. 38, #1; 1990, p.70
37. Gere, W. "Heuristics in job shop scheduling" *Management Sciences*, Vol. 13, #3; 1966, p. 167
38. Giffler, B.; Thompson, G. "Algorithms for solving production problems" *Operations Research*, Vol. 8, #4; 1960, p. 487
39. Giffler, B.; Thompson, G. & Van Ness "Numerical experience with the linear and Monte Carlo algorithms for solving production scheduling problems" in *Industrial Scheduling*, 1963, p. 21
40. Giglio, R.; Wagner, H. "Approximate solutions to the three machine scheduling problem" *Operations Research*, Vol. 12, #2; 1964, p. 305
41. Goodwin, J.S.; Goodwin, J.W. "Operating policies for scheduling assembled parts" *Decision Sciences*, Vol. 13, #4; 1982, p. 585

42. Graves, S. "A review of production scheduling"
Operations Research, Vol. 29, #4; 1981, p.646
43. Greenberg, H. "A branch & bound solution to the general scheduling problem"
Operations Research, Vol. 16; 1968, p. 353
44. Greene, H. ed. Production & Inventory Control Handbook
McGraw Hill: New York, N.Y., 1987
45. Haider, W.; Moodie, C.; & Buck, J. "An investigation of the advantages of using a man-computer interactive scheduling methodology" International Journal of Production Research, Vol. 19, #4; 1981, p. 381
46. Hastings, N.; Marshall, P. & Willis "Schedule-based MRP: an integrated approach to production scheduling and MRP" Journal of the Operations Research Society, Vol. 33, #11; 1982, p. 1021
47. Hastings, N.; Marshall, P. "JOB CODE: Schedule-based materials requirements planning" TIMS & ORSA National Conference Proceedings, 1984
48. Hastings, N.; Packham, B. "Developments in schedule-based MRP"
Proceedings of the 7th Conference of the Australian Operations Research Society; 1985
49. Hastings, N.; Yeh, C. "Job oriented production scheduling"
European Journal of the Operational Research Society, Vol. 47, #5; 1990, p. 35
50. Hax, A.; Meal, H. "Hierarchical integration of production planning and scheduling"
Studies in Management Science, Vol. 1; 1975
51. Ho, Y. "A new approach to the analysis of discrete event dynamic systems"
Automatica, Vol. 19; 1983, p. 149
52. Hoffman, T.; Scudder, G. "Priority scheduling with cost considerations"
International Journal of Production Research, Vol. 21, #6; 1983, p.881
53. Hon, K. "Sequence planning systems for dynamic manufacturing environment"
Proceedings of the 3rd Conference on Manufacturing Engineering, 1986, p. 123
54. Jefferson, D. "Virtual Time"
ACM Transactions on Programming Languages, Vol. 7, #3; 1985, p. 404
55. Johri, P. "Linear programming approach to capacity estimation"
International Journal of Production Research, Vol. 25, #6; 1987, p. 851

56. Johri, P. "Iterative algorithms for estimating the capacity of automated production lines with finite buffers" *International Journal of Production Research*, Vol. 27, #34; 1989, p. 687
57. Kang, S.; Hong, S. "Study on the facility allocation to the multi-product production system" *Proceedings of the 8th International Conference on Production Research*, 1985, p. 305
58. Karp, R. "Reducibility among combinatorial problems" *Complexity of Computer Communications*, 1972, p. 85
59. Kempf, C., LePape, C., Smith, S., Fox, B. "Issues in the Design of AI Based Schedulers: A Workshop Report", *AI Magazine*, Vol. 11, #5; 1991, p. 37
60. Kicks, P.; Donaldson, W. "Irregular demand: assessing a rough and ready lot size formula" *Journal of the Operations Research Society*, Vol. 31, #8; 1980, p. 725
61. Kropp, D.; Carlson, R. & Jucker "Heuristic lot-sizing approaches for dealing with MRP system nervousness" *Decision Sciences*, Vol. 14, #2; 1983, p. 156
62. Ku, H.; Karimi, I. "Scheduling in serial multi-product batch processes with finite interstage storage; a mixed integer-linear programming formulation" *Industrial Engineering Chemistry Research*, Vol. 27, #10; 1988, p. 1840
63. Lagweg, L; Lenstra, J. & Rinooy Kan "Job-shop scheduling by implicit enumeration" *Management Sciences*, Vol. 24, #4; 1977, p. 441
64. Leary, E. "Decision support systems aid in management of operations" *Industrial Engineering*, #9; 1985, p. 26
65. Lenstra, J.; Rinooy Kan & Brucker "Complexity of machine scheduling problems" *Annals of Discrete Mathematics*, Vol. 7, #3; 1977, p. 343
66. Libosar, C.; Proth, J. "Production flow control under finite capacity" *Proceedings of the 27th IEEE Conference on Decision and Control*, 1988, p. 1858
67. Love, S. Inventory Control
McGraw Hill: New York, N.Y., 1979
68. Mamcheck, A.; McClelland, M. "Strategic Issues in make to order manufacturing" *Production and Inventory Management*, Vol. 2; 1986, p. 82
69. Manne, A. "On the job-shop scheduling problem" *Operations Research*, Vol. 8, #2; 1960, p. 219

70. Marshall, P. "Schedule-based materials requirements planning"
Ph.D. Thesis, Monash University, Aus., 1984
71. McCarthy, S.; Barber, K. "Medium to short term finite capacity scheduling"
Engineering Costs and Production Economics, Vol. 19, #1; 1990, p. 189
72. McClain, Maxwell, Muckstadt et al. "On MRP lot-sizing"
Management Sciences, Vol. 28, #5; 1982, p.582
73. McCormick, Pinedo, Shenker & Wolf "Sequencing in an assembly line with blocking to minimize cycle time" Operations Research, Vol. 37, #6; 1989, p. 925
74. McKay, K.; Buzacott, J. & Safayeni "The scheduler's desk - can it be automated"
IFAC '89 - Genoa, Italy, 1989, p. 155
75. McKay, K.; Buzacott, J. & Safayeni "The scheduler's information system: what is going on? Insights for automated environments" INCOM '89 - Madrid, Spain, 1989, p. 405
76. McLeavey, D. & Narasimhan, S. Production Planning and Inventory Control
Allyn & Bacon Inc.: Newton, Mass., 1985
77. Meng, A.; Sullivan, M. "Logos - a constraint directed reasoning shell for operations management" IEEE Expert, February 1991, p. 29
78. Miller, L.; Ginsberg, A.; Maxwell "An experimental investigation of priority dispatching in a simple assembly shop" Logistics, Vol. 1; 1975, p. 161
79. Misra, J. "Distributed discrete event simulation"
ACM Computing Surveys, Vol. 18, #1; 1986, p. 39
80. Muhlemann, A.; Lockett, A. & Fam "Job shop scheduling heuristics and frequency of scheduling" International Journal of Production Research, Vol. 20, #2; 1982, p. 227
81. Muth, J. & Thompson, G. Industrial Scheduling
Prentice Hall: Englewood Cliffs, N.J., 1963
82. New, C. "Lot-sizing in multi-level requirements planning systems"
Production and Inventory Management, Vol. 15, #4; 1974, p. 57
83. Orlicky, J. Materials Requirements Planning
McGraw Hill: New York, N.Y., 1975
84. Ow, P.; Smith, S. & Howie, R. "A cooperative scheduling system"
Expert Systems & Intelligent Manufacturing, Vol. ?; 1988

85. Owens, T. "Hybrid optimization and knowledge-based approach to job-shop scheduling" 2nd International CIM Conference, Troy, N.Y.; May 21-23, 1990
86. Panwalkar, s.; Iskander, W. "A survey of scheduling rules" Operations Research, Vol. 25, #1; 1977, p. 45
87. Park, Y.; Peyden, C.; Enscoe, E. "A survey and evaluation of static flowshop scheduling heuristics" International Journal of Production Research, Vol. 22, #1; 1984, p. 18
88. Ragatz, G.; Mabert, V. "A framework for the study of due date management in job-shops" International Journal of Production Research, Vol. 22, #4; 1984, p. 685
89. Ramasesh, R. "Dynamic job-shop scheduling: a survey of simulation research" Omega International Journal of Management Science, Vol. 18, #1; 1990, p. 43
90. Savage, E.; Mikurak, M. "Finite scheduling - staging a comeback" CIM Technology, Vol. 5, #1; 1986, p. 26
91. Sawik, T. "Hierarchical scheduling flowshops with parallel machines and finite in-process buffers" Arch. Automat. Telemech., Vol. 33, #3; 1988, p.403
92. Schrage, L. "Solving resource constrained network problems by implicit enumeration - non-preemptive case" Operations Research, Vol 18; 1970, p. 263
93. Scudder, G. "Scheduling and labour assignment policies for a dual constrained shop" International Journal of Production Research, Vol. 26, #3; 1986, p. 623
94. Sculli, D. "Priority dispatching rules in job-shops with assembly operations and random delays" Omega, Vol. 8, #2; 1980, p. 227
95. Sculli, D. "Priority dispatching rules in an assembly shop" Omega, Vol. 15, #1; 1984, p. 49
96. Silver, E.; Miltenburg, G. "Two modifications of the Silver-Meal lot sizing heuristic" INFOR, Vol. 22, #1; 1984, p.56
97. Silver, E.; Peterson, R. Decision Systems for Inventory Management and Production Control John Wiley & Sons Inc.: New York, N.Y., 1979
98. Smith, S. "A constraint based framework for reactive management of factory schedules" Expert Systems and Intelligent Manufacturing, Elsevier Science Publishing Company: 1987, p. 113

99. Smith, S.; Fox, M.; Ow, P. "Constructing and maintaining detailed production plans" *AI Magazine*, Vol. 7, #4; 1991, p. 29
100. Steinberg, E.; Napier, H. "Optimal multi-level lot sizing for requirements planning systems" *Management Science*, Vol. 26, #12; 1980, p. 1258
101. Story, A.; Wagner, H. "Computational experience with integer programming for job shop scheduling" in Industrial Scheduling, Prentice Hall: Englewood Cliffs, N.J., 1963
102. Sycara, Roth, Sadeh, & Fox "Resource allocation in distributed factory scheduling" *IEEE Expert*, February, 1991, p. 29
103. Toye, C. "Let's update capacity requirement planning logic" *1990 APICS Conference Proceedings*, 1990, p. 215
104. Wagner, W. "An integer linear programming model for machine scheduling" *Naval Research Logistics Quarterly*, Vol. 6, #2; 1959, p. 131
105. Weeks, J. "A simulation study of predictable due dates" *Management Sciences*, Vol. 25, #4; 1979, p. 363
106. Weisser, P.; Howie, R. "Distributed scheduling and temporal reasoning" *7th AIAA Computers in Aerospace*, Monterey Calif., Oct. 1989
107. Weisser, P.; Cochran, S. "Modeling temporal relations in task hierarchies for job shop scheduling" *Proceedings of the 2nd International CIM Conference*, May 21-23, 1990
108. Wiede, W.; Kuriyan, K. & Reklaitis "Determination of completion times for serial multiproduct processes" *Computers in Chemical Engineering*, Vol. 11, #4; 1987, p. 337
109. Wight, O. Production and Inventory Management in the Computer Age Cahnern Publishing: Boston, Mass., 1974
110. Wortman, D. "Simulation for the design and control of manufacturing operations" *SME Technical Paper*, 1989, p. 728

APPENDIX

A1. Glossary of Terms

assembly

Any object composed of two or more components (syn: parent).

backward loading/scheduling

The act of allocating time for an operation beginning with the due date (or latest finish) and moving backward in time.

bill of materials

A specification of the children included in an assembly.

Capacity Requirements Planning (CRP)

A process by which the machine or labour load required to meet a given material plan is determined. The results are typically stated in terms of hours per machine or labour class.

child

Any object which is used to create an assembly (syn: component, sub-assembly).

component

Any object which is used to create an assembly (syn: child, sub-assembly).

computer planned order

A planned order generated by the computer program used to run the MRP process. This form of order has had no human assessment applied to it.

consumption rate

The number of units of a given item removed from inventory per unit of time.

dependent demand

Demand for children which is derived only from demand against a parent.

discrete-batch

A fixed number of a given item processed as one group.

discrete order policy

A lot sizing technique in which an order is placed for each individual requirements. No requirements are grouped for ordering purposes.

distribution inventory

Finished goods inventory held to support the sales process.

explosion

The process of identifying the children of an assembly by decomposition through a bill of materials.

finite capacity

A limited ability to manufacture. Usually used to refer to the ability of an algorithm to consider this limit.

forecast

Demand which is predicted to occur based on market research and statistical analysis.

forward loading/scheduling

The act of allocating time for an operation beginning with the earliest start date and moving forward in time.

independent demand

Demand derived through forecast or customer demand.

infinite capacity

Unlimited ability to manufacture. Usually used to refer to the inability of an algorithm to consider limits on capacity

item

Any assembly, sub-assembly or component.

job-shop

A manufacturing environment which produces a variety of end items. Each batch of product manufactured may differ from previous batches.

lead time

An estimate of the amount of time required for a child to be assembled in to its parent.

lot size

The number of items selected for manufacturing or purchasing in a single order.

lot splitting

The act of separating a batch of items and processing each sub-batch on separate machinery. This process is usually employed to reduce the elapsed batch processing time.

lot-for-lot

A lot-sizing policy where no grouping of requirements is employed. A new order is created for each demand against the item.

manufacturing inventory

Inventory held to support the manufacturing process. Purchased materials, sub-assemblies and components can be held as manufacturing inventory.

manufacturing planning

Some or all of MRP, MPS, CRP or shop scheduling (syn: production planning).

Master Production Scheduling (MPS)

A process which converts a high-level production forecast into a specific finished goods delivery schedule.

Material Requirements Planning (MRP)

A process which converts a finished goods delivery schedule into net material requirements expressed as shop orders and purchase orders.

multi-echelon inventory

A system in which one level of inventory depends on other levels. Inventory of finished automobiles, engines, transmissions, carburetors and gears represent a multi-echelon inventory system since several levels of stockable item depend on (or incorporate) other stockable items.

nervousness

Small, frequent and unpredictable variations such as occur in near term planned orders in an MRP environment.

netting

The process of determining net demand given gross demand and supply information.

open order

Supply of an item which has been requested from a supplier but not received in full.

overlapping

The act of sending a batch of an item on to the next operation in its routing for processing before the current batch is complete. This allows the next operation to begin on part of the batch while the previous operation completes.

parent

Any object composed of two or more components (syn: assembly).

part

Any assembly, sub-assembly or component (syn: item)

part-period balancing (PPB)

A lot sizing algorithm which balances order overhead costs against inventory carrying costs.

planned order

A tentative or suggested supply for an item. Planned orders typically have not been release to the shop or supplier.

production planning

Any one or more of MRP, CRP, MPS or shop scheduling (syn: manufacturing planning).

routing

The sequence of operations required to transform an item or group of items into its manufactured state or parent assembly.

scrap

That portion of manufactured product which does not meet finished specifications and is discarded or reworked.

scheduling window

The maximum horizon over which jobs are considered for scheduling. A scheduler will hold all jobs falling in the same window before applying a priority mechanism to determine which has highest priority.

shop scheduling

The process of assigning manufacturing operations to machines in specific quantities and time slots.

shrinkage

That portion of purchased items which do not meet specification through in-coming inspection and are discarded.

sub-assembly

An assembly which is also a component.

A2. Sample Problems from Chapter 3

Sample problems taken from Fisher & Johnson

A) 6 Job x 6 Machines x 6 Operations

Opn #	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6
1	3/1	2/8	3/5	2/5	3/9	2/3
2	1/3	3/5	4/4	1/5	2/3	4/3
3	2/6	5/10	6/8	3/5	5/5	6/9
4	4/7	6/10	1/9	4/3	6/4	1/10
5	6/3	1/10	2/1	5/8	1/3	5/4
6	5/6	4/4	5/7	6/9	4/1	3/1

* X/Y denotes a requirement of Y hours on machine X

B) 10 x 10 x 10 Problem

Opn #	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9	Job 10
1	1/29	1/43	2/91	2/81	3/14	3/84	2/46	3/31	1/7	2/85
2	2/78	3/90	1/85	3/95	1/6	2/2	1/37	1/86	2/69	1/13
3	3/9	5/75	4/39	1/71	2/22	6/52	4/61	2/46	4/76	3/61
4	4/36	10/11	3/74	5/99	6/61	4/95	3/13	6/74	6/51	7/7
5	5/49	4/69	9/90	7/9	4/26	9/48	7/32	5/32	3/85	9/64
6	6/11	2/28	6/10	9/52	5/69	10/72	6/21	7/88	10/11	10/76
7	7/62	7/46	8/12	8/85	9/21	1/47	10/32	9/19	7/40	6/47
8	8/56	6/46	7/89	4/98	8/49	7/65	9/89	10/48	8/89	4/52
9	9/44	8/72	10/45	10/22	10/72	5/6	8/30	8/36	5/26	5/90
10	10/21	9/30	5/33	6/43	7/53	8/25	5/55	4/79	9/74	8/45

C). 20 x 5 x 5 Problem

Opn #	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job 7	Job 8	Job 9	Job 10
1	1/29	1/43	2/91	2/81	3/14	3/84	2/46	3/31	1/76	2/85
2	2/9	2/75	1/39	1/71	2/22	2/52	1/61	2/46	4/76	3/61
3	3/49	4/69	3/90	5/9	1/26	5/48	3/32	1/32	3/85	1/64
4	4/62	3/46	5/12	3/85	4/21	1/47	4/32	4/19	2/40	4/47
5	5/44	5/72	4/45	4/22	5/72	4/6	5/30	5/36	5/26	5/90
Opn #	Job 11	Job 12	Job 13	Job 14	Job 15	Job 16	Job 17	Job 18	Job 19	Job 20
1	2/78	3/90	1/85	3/95	1/6	2/2	1/37	1/86	2/69	1/13
2	4/36	1/11	3/74	1/99	2/61	1/95	3/13	2/74	3/51	2/7
3	1/11	2/28	2/10	2/52	5/69	4/72	2/21	5/88	1/11	3/76
4	5/56	4/46	4/89	4/98	3/49	5/65	4/89	3/48	4/89	4/52
5	3/21	5/30	5/33	5/43	4/53	3/25	5/55	4/79	5/74	5/45

A3. Heuristic Performance Measures (Referenced in Chapter 3)

The following is a list of performance measures used in evaluating shop scheduling heuristics. The list has been taken from a paper by Ramasesh[89].

A) Time based measures.

- i) Mean and variance of flow time per job.
- ii) Mean and variance of flow time per operation.
- iii) Mean waiting time.
- iv) Machine idle time.

B) Work-in-process (WIP) measures.

- i) Average number of jobs in queue.
- ii) Value of work in process.

C) Due date related measures

- i) Mean tardiness.
- ii) Conditional mean tardiness.
- iii) Proportion of jobs tardy.
- iv) Mean lateness.
- v) Number of jobs tardy.
- vi) Maximum lateness.

D) Cost based measures.

- i) Cost of idle machines.
- ii) Cost of wrong promises.
- iii) Cost of WIP inventory.
- iv) Total cost per job.
- v) Average value added in queue.

A4. Priority Rule Definitions (referenced in chapter 3)

The textual explanation of the priority rules stated below describes the job to be given the highest priority from the set of jobs in the queue under consideration.

COM (Composite): the job with the lowest value of a composite priority function.

COST : the job with the highest value of a composite, cost - oriented, priority function.

COVERT (Cost OVER Time): the job with the highest delay cost over the remaining time to its due date.

CRR (CRITICAL Ratio): the job with the lowest ratio of remaining slack time over processing time.

EDD (Earliest Due Date): the job with the earliest due date.

FCFS (First-Come-First-Served): same as FIFO .

FIFO (First-In-First-Out): the job which has been in the queue the longest .

FOPNR (Fewest OPERatioNs Remaining): the job with the fewest number of operations remaining.

JDD (Job Due Date): the job whose parent has the earliest due date.

LSTART (Latest START): the job with the earliest latest start time.

LWKR (Least Work Remaining): the job with the least total amount of processing time remaining.

NSUT: the job with the lowest set up time for the current operation.

NUJOB (NUMBER of JOBS): the job whose parent has the smallest number of uncompleted parts.

ODD (Operational Due Date): the job with the earliest operational due date (where due dates are assigned to each operation as the job enters the shop).

OSLK (Operational SLack): the job with the least operational slack (operational due date - processing time).

RAN (RANDOM): select a job at random.

SLACK: the job with the least remaining slack.

S/OPN (Slack per remaining OPerationS): the job with the lowest ratio of remaining slack time to the number of remaining operations.

SPT (Shortest Processing Time): the job with the shortest processing time on this machine.

SPT-T (truncated SPT): the job with the shortest processing time or the job which has exceeded the maximum wait time in this queue.

TJT (Total Job Time): the job whose parent has the smallest total operation time including completed operations.

TWK (Total Work): the job with the least total work (i.e. processing time required).

A5. Dataset #1 from Chapter 7

Scenario 1 - Shop with a Bottleneck Machine

Carrying Costs

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	11375	4334	2731	2138	1526	1041	903
Coherent Model	3200	1245	1076	1022	928	742	898

Order Setup Costs

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	27280	27280	27280	27280	27280	27280	27280
Coherent Model	20880	20880	20960	20880	20880	20880	20880

Machine Setup Costs

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	7995	7995	7995	7995	7995	7995	7995
Coherent Model	6337	6312	6333	6312	6315	6312	6294

Total Costs Less Late Penalty

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	46650	39609	38006	37413	36801	36316	36178
Coherent Model	30417	28437	28369	28214	28123	27934	28072
% Cost Reduction	35%	28%	25%	25%	24%	23%	22%

Infeasibilities in Generated Plans

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	137	57	18	9	4	2	1

A6. Dataset #2 from Chapter 7

Scenario 2 - Even Shop Loading

Carrying Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	10610	4280	2690	2098	1485	1001	863
Coherent Model	2636	1200	1139	1075	945	837	896

Order Overhead Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	25360	25360	25360	25360	25360	25360	25360
Coherent Model	17280	17200	17280	17200	17280	17360	17200

Machine Setup Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	7215	7215	7215	7215	7215	7215	7215
Coherent Model	5311	5268	5286	5265	5289	5307	5268

Total Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	43185	36855	35265	34673	34060	33576	33438
Coherent Model	25227	23668	23705	23540	23514	23504	23364
% Cost Reduction	42%	36%	33%	32%	31%	30%	30%

A7. Dataset #3 from Chapter 7

Scenario 3 - Modified Order Overhead and Carrying Costs

Carrying Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	22750	8668	5462	4726	3051	2081	1806
Coherent Model	4884	1628	1043	893	758	614	726

Order Overhead Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	6820	6820	6820	6820	6820	6820	6820
Coherent Model	5760	5660	5680	5700	5680	5680	5620

Machine Setup Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	7995	7995	7995	7995	7995	7995	7995
Coherent Model	6881	6760	6781	6803	6779	6779	6722

Total Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	37565	23483	20277	19541	17866	16896	16621
Coherent Model	17525	14048	13504	13396	13217	13073	13068
% Cost Reduction	53%	40%	33%	31%	26%	23%	21%

A8. Dataset #4 from Chapter 7

Scenario 4 - Even Shop Load, All Items Ordered on PPB

Carrying Costs

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	11375	4334	2731	2138	1526	1041	903
Coherent Model	3906	4079	3590	3655	3325	3237	3236

Order Setup Costs

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	27280	27280	27280	27280	27280	27280	27280
Coherent Model	3440	3760	3520	3520	3680	3760	3760

Machine Setup Costs

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	7995	7995	7995	7995	7995	7995	7995
Coherent Model	855	1059	971	971	1013	1059	1059

Total Costs Less Late Penalty

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	46650	39609	38006	37413	36801	36316	36178
Coherent Model	8201	8898	8081	8146	8018	8056	8055
% Cost Reduction	82%	78%	79%	78%	78%	78%	78%

A9. Due Date Experimentation Data from Chapter 7

Scenario 5 - Costs from Scenario 3 with Late Days

No Date Adjustments

Carrying Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	22750	8668	5462	4726	3051	2081	1806
Date	22508	8621	5417	4239	3046	2070	1806
Coherent Model	4884	1628	1043	893	758	614	726

Order Overhead Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	6820	6820	6820	6820	6820	6820	6820
Date	6820	6820	6820	6820	6820	6820	6820
Coherent Model	5760	5660	5680	5700	5680	5680	5620

Machine Setup Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	7995	7995	7995	7995	7995	7995	7995
Date	7995	7995	7995	7995	7995	7995	7995
Coherent Model	6881	6760	6781	6803	6779	6779	6722

Order Days Late

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	17626	3325	1200	734	307	39	0
Date	15672	2474	699	383	166	18	0
Coherent Model	6792	249	0	0	0	0	0

Total Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	37565	23483	20277	19541	17866	16896	16621
Date	37323	23436	20232	19054	17861	16885	16621
Coherent Model	17525	14048	13504	13396	13217	13073	13068

One Pass Due Date Adjustments

Carrying Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	16537	7710	5413	4273	4042	3958	3954
Date	16034	7674	5385	4265	4035	3960	3954
Coherent Model	5264	4489	4255	4152	4196	4155	4178

Order Overhead Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	6820	6820	6820	6820	6820	6820	6820
Date	6820	6820	6820	6820	6820	6820	6820
Coherent Model	5820	5780	5680	5860	5780	5680	5660

Machine Setup Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	7995	7995	7995	7995	7995	7995	7995
Date	7995	7995	7995	7995	7995	7995	7995
Coherent Model	5165	5053	5071	5074	5053	5071	5053

Order Days Late

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	10044	1201	399	7	0	0	0
Date	6602	617	201	5	0	0	0
Coherent Model	6224	237	0	0	0	0	0

Total Costs (\$)

Scheduling Method	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Baseline	31352	22525	20228	19088	18857	18773	18769
Date	30849	22489	20200	19080	18850	18775	18769
Coherent Model	16249	15322	15006	15086	15029	14906	14891

A10.Speed of Computation

One of the original objectives was to develop an algorithm that would be practical for implementation in an industrial environment. As part of this criteria, the speed of execution of the algorithm is of interest. The ability of a manufacturer to respond to change is one of the key mechanisms used to gain a competitive advantage in industry today. As part of the ability to respond, manufacturers need information systems that allow them to analyze the implication of manufacturing decisions in terms of feasibility and cost impacts. MRP runs alone typically require four to twenty-four hours to generate a complete set of plans today in the manufacturing environments of interest. These plans must then be processed by the shop scheduling system to get an overall view of any suggested changes to the end product delivery schedule.

Some researches have recognized this requirement and are trying to address the problem of 'reactive' production planning as opposed to predictive planning. Smith, Fox & Ow have made some strides in this area using artificial intelligence approaches as outlined in two papers [98, 99].

The thrust of this discussion is that it is worthwhile to consider the execution speed of the coherent planning algorithm. The results of this analysis are given here without drawing any conclusions. To be able to make a definitive statement about the feasibility of this approach it would be necessary to analyze the execution speed on a variety of database sizes as well as shop load conditions. Since only one database was available for this study the performance was only considered with regard to load conditions. The table and graph below present the results of this analysis. This data was gathered using the problem scenario of section 7.3.2.

Execution times of the coherent planning algorithm

	Fraction of Factory Capacity						
	0.02	0.04	0.06	0.08	0.125	0.25	0.5
Elapsed Time (s)	1.54	1.07	0.94	0.93	0.84	0.80	0.77
CPU Time (s)	1.47	0.96	0.86	0.84	0.78	0.75	0.74

Figure A-1: Execution times of the coherent planning algorithm.

END

24 | 05 | 93

FIN