University of Alberta

Design and Development of a Decision Support System for Scheduling Steel Fabrication Projects

by

K.N. Gopalakrishnan ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**

The Department of Mechanical Engineering

University of Alberta

Edmonton, Alberta, Canada

Spring 1998

0-612-29041-7

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Design and Development of a Decision Support System for Scheduling Steel Fabrication Projects** submitted by **K.N. Gopalakrishnan** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dr. J. Whittaker

Dr. S.AbouRizk

Dr. J. Sprague

Prof. P. Dozzi

Dr. D. Halpin

06 February, 1998

Abstract

One of the most important and difficult problems faced by the steel fabrication industry is the planning and scheduling of shop activities. Competitive pressures force fabricators to disrupt schedules in progress to accommodate frequent requests from key customers for design changes and/or delivery schedules. Without a proper shop floor control system, organizations will experience many productivity problems. Resolution of these difficulties can translate into considerable direct and indirect savings.

The analysis carried out in this research describes the development of a decision support system for scheduling steel fabrication projects that is geared towards (1) providing a real-time shop floor status monitoring system, (2) providing predictive schedules, analysis and consequences of managerial priorities on schedule performance, (3) providing local decision making capabilities, and (4) optimizing resource utilization. The DSS has been designed to handle many shop configurations; it takes into account real-life constraints. The purpose of designing the DSS is to close the gap between scheduling models in research and real scheduling problems faced by the steel fabrication industry.

The need for both short term planning and an efficient progress reporting and feedback mechanism in situations where each project is unique and virtually made to order has long been recognized by steel fabricators. The paucity of process information and timely shop floor status updates places a burden on schedulers who have to manually sort through the various permutations and combinations that are possible in scheduling. In this scenario, where flexibility and rapid response to changing customer requirements are key success

factors, the role of a computerized decision support system to aid the shop personnel cannot be over-emphasized.

The DSS prototype is being implemented in a real-life environment, using event driven programming and a relational database. The results to date are very encouraging.

Acknowledgement

The author wishes to express his sincere thanks to his two academic advisors, Dr. J.D. Whittaker, and Dr. S.M. AbouRizk for their guidance and encouragement in the preparation of this thesis.

Special thanks are due to my wife Narayani for her unflinching support and understanding during the years of this research.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Introduction

The symptoms of low competitiveness include: low productivity (despite larger plant facility and sales volume), high work-in-process (WIP) levels, low profit margins (despite price increases), slow deliveries (lateness of customer orders), and slow response to a rapidly changing environment of short product life cycle and frequent design changes.

To gain a competitive advantage, manufacturers must carefully plan and invest in sometimes costly plants and equipment. The performance of a manufacturing/fabrication system is affected by numerous (and sometimes conflicting) needs such as: material requirements, capacity allocation, facility and material handling capacity, inventory control and tool management, scheduling, and quality control.

1.2 The Problem

In 1976 Joseph Orlicky wrote a text on Materials Requirement Planning (MRP) that has become, to date, the single most important resource on materials requirements planning. MRP has evolved from a Bill-of-Materials (BOM) explosion technique to a comprehensive manufacturing information system. Orlicky's text was instrumental in providing the knowledge required to promote development and growth of these concepts. Similar progress however has not been made in the understanding of capacity management problems in manufacturing/fabrication. Capacity management is a far more complex problem and is key to proper management of manufacturing/fabrication activities. With reduced inventory levels, effective use of system capacity can provide the competitive edge needed to increase overall plant output.

Scheduling is one of the most important issues in the planning and operation of manufacturing/fabrication systems. It arises in situations where limited resources are to be assigned over time to perform a set of activities. Many productivity problems are associated with scheduling problems such as not having the right item when needed, not having the right equipment when needed, using excess inventory to hide problems, and inflexibility and lack of responsiveness (Huang and Tang, 1990). Resolution of these difficulties can translate into considerable direct and indirect savings. Applications for scheduling can be found in production planning, time-tabling, or real-time system controlling. Scheduling is implemented as part of production planning and control functions. Schedules serve as a guide for production/fabrication and establishing manufacturing resource requirements in terms of manpower, machines, toolings, fixtures, materials, and possibly other specific facilities (Newman, 1988). An introduction to scheduling can be found in (Conway et al., 1967), (Baker, 1974), (Rinnooy Kan, 1976), (Pinedo, 1995).

Dudek et al. (1992) lament the lack of application of flow shop scheduling research results, saying, "we have never been approached by anyone claiming to have a need for solving a problem having the characteristics assumed by flow shop researchers." They go on to suggest a number of issues affecting the implementation of scheduling algorithms, particularly those that try to minimize makespan, viz.:

- The problem as described seldom or never exists.
- The problem may exist but with multiple criteria.
- The assumptions used in algorithm development may be overly restrictive.
- The algorithm may be inflexible and ineffective in handling any situations other than under the assumed conditions.

They add, "at this time, it appears that one research paper sets a wave of research in motion that devours scores of person years of research time on intractable problems of little practical consequence."

Reisman et al. (1997) surveyed flow shop scheduling literature from 1952 to the present. The primary objective of their study was to evaluate the nature of research in the shop scheduling area. They found that 91% of all the articles they surveyed fell under what they call the "untested theory" category (as opposed to 3% that fall in "pure application" category). They cite the need for doing "missionary" work in addressing and solving real problems of consequence in the work place. (Specifically, problems that do not make common assumptions that machines are continuously available, jobs can be processed at one machine only, or the restrictive assumption that one machine can handle only one job at a time).

The problem of scheduling is more acute in the steel fabrication industry where the prevailing methods are fragmented and fail to address the unique nature of steel projects. A typical steel project shop, which has the complexity and uncertainty of the manufacturing environment, is a collection of machines and individuals capable of performing a variety of tasks towards the completion of jobs.

Scheduling is one of the most important issues in the planning and operation of steel projects. Projects typically span a long time interval (even 2 or more years) and are generally of a non-repetitive nature. This rules out traditional economies of scale approaches of batch production. Moreover, since 'one of a kind' projects are not rare, the data collection methods employed by the industry are not ideal. This presents considerable challenges to be overcome through innovative and advanced data-gathering techniques and development of decision support infrastructures that will promote data based decision making. In reality, scheduling within the steel industry requires a more involved procedure, which can account for prioritization of the given work based on varying criteria such as who the customer is, the impact on other projects and effects of changes on overall company policy, and overall project performance. In a typical steel shop, jobs are often expedited, pre-empted, and postponed. Job expedition and pre-emption of live orders occurs because of unrealistic promise dates and lack of sufficient system capacity. Postponement can occur due to pre-emption of other orders or lack of materials, tools, jigs, or manpower. The scheduling system must be able to operate in near real-time, accommodate multiple performance measures and must be easily understood by the shop floor personnel. Therefore, a good and simple user interface is essential. The timing factor, to a large extent, eliminates simulation and all forms of mathematical programming approaches to this problem.

Existing algorithmic techniques for factory scheduling are capable of incorporating only a small fraction of scheduling knowledge, and result in schedules that bear little resemblance to actual schedules. At best, they only provide a framework for developing detailed schedules. As a result, many manufacturing/fabrication policies are not consistent with the firm's objectives and market needs. This problem is further complicated by the inherently unpredictable nature of job shop operations. Consequently, decisions are made which focus on the short term planning horizon and ignore the more important issues. Hence, it is necessary to provide management with the ability to make sound short term data-based decisions. In order to address the competitive factors mentioned above, organizations need a very flexible and quick method that will be useful for analysis of various strategies (and rules) for fulfilling customer requirements. An organization that has several concurrent projects (or products being produced) that share common resources, will have a need for:

- A comprehensive prioritization scheme.

- Resource leveling features.
- Company wide information systems for updating/revising customer requirements and actual shop status.

The study of competitiveness at the firm level is significant because some type of hierarchy is always involved in the decision making process, thus negating single stage optimization techniques. There is a great need for high level research on the subject, specifically a sound theoretical and scientific foundation for efficient solutions of future practical problems. A proper framework for the design and development of a comprehensive decision support system for scheduling steel projects that can be applied to the steel fabrication/construction sector will be a valuable addition to the body of academic knowledge and will help to organizations to better understand the dynamics involved.

1.3 Research Objectives:

The primary objective of this research is to provide solutions and improvements for the current inadequacies of scheduling practices in the construction /steel fabrication industry. This will be accomplished through the design and development of a Decision Support System (DSS) that organizations can use for finding appropriate solutions to questions such as: for a fixed plant and set of equipment,

1. Given a set of customer orders (firm and/or planned) with different priorities, that have varying resource requirements, what is the combination (schedule of orders/operations) that will satisfy a set of goals (meeting due dates, maximize resource utilization, maximize revenues etc.;)?
2. What are the potential bottlenecks for a given demand forecast? Are there any supporting resources that restrict throughput? How does the system perform for different resource levels and for different user specified rules of sequencing? What is

the value of WIP? What is the utilization of resources? What is the effect of a machine breakdown? What is the effect of adding one more machine, shift or overtime?

3. A new rush order comes in a specified time and shop condition. Given the existing shop load, should the "new rush order" be accepted or rejected? What should be the price for the new order? If the “new rush order” must be accepted, what is the basis for delaying or rejecting one of the existing orders? What is the cost of this action?
4. What is the effect on system performance when an order being currently worked on is changed (minor change like order quantity, or major change(s) like design change (process change)) and will effectively block the machine (resource) for a known/unknown period of time?

Other benefits from finding the appropriate actions to the above questions will be the ability to:

- Provide a real-time shop floor status monitoring system.
- Provide predictive schedules, analysis, consequences and pro-active scheduling ability.
- Provide local decision making and analysis capabilities.
- Project the factory-wide consequences of alternative manufacturing decisions being considered at a given work center.
- Reduce WIP and optimize resource utilization.
- Understand tradeoffs between policies.
- Provide an active user interface.

1.4 Overview of Scheduling and Motivation for Research

The factory scheduling problem concerns the allocation of resources to manufacturing activities over time so that the order for parts received by the factory are produced in a timely and cost-effective fashion. This scheduling problem is in essence very simple to state:

*A set of '**n**' jobs with different priorities have to pass through '**m**' processes in such a way that some objective function(s) is / are optimized.*

The simplicity of this statement, however, belies the complexity of the problem in the practical situation. The unpredictability of actual shop floor operations further complicates the problem in actual manufacturing environments. Three difficulties immediately emerge:

- The combinatorial nature of the problem.
- The difficulty in specifying the objective function.
- The need to accommodate change.

Thus, a realistic solution requires integration of two broad capabilities, (i) generation of production schedules that accurately reflect the constraints and objectives of the actual manufacturing environment, and (ii) incremental revision of these schedules as unanticipated events on the shop floor force deviations from the planned actions/ schedule.

This simple scheduling problem which has been the interest of both academic and industrial professionals, has proven to be too complex to treat mathematically. The scheduling problem has many solutions, thus, the question of optimality and the determination of a single objective function arises. Various attempts to use exact, analytic models for practical scheduling problems usually lead to many difficulties. On the one hand the size of the real scheduling task is comparatively large. On the other hand, the validity of the analytical model is usually conditioned by a whole list of various assumptions and constraints. Although many of the constraints encountered in practice can be represented mathematically, with only a few of the situations considered, the problem soon becomes intractable. Hence much of the research using these methods involves highly simplified and not very applicable versions of the actual problem. For this reason, a heuristic approach has been used to make the scheduling model workable in real-life systems (Hastings and Yeh, 1990). Although a heuristic approach may lead to only sub-optimal solutions, in practice such solutions are usually sufficient, as feasibility of the schedule and speed of generation are more critical than theoretical optimality in an environment where the objective function often changes.

Currently, a scheduler or production controller usually uses knowledge and experience to carry out scheduling. Any appropriate computer model must be able to accommodate the controller's knowledge and mate it with the combinatorial power of the computer. Many requirements must be met by the scheduling system. The most important of them are:

- The schedule must meet due dates.
- The actual state of the production system must be taken into account.
- The schedule must be immediately performable.
- The scheduling system must not only be capable of creating a schedule, but also must be able to survive various disturbances leading to the invalidation of the schedule.
- Various specific criteria and conditions must be respected according to the particular production system.

A case of using only one priority rule in the scheduling of operations is less probable in practice. Usually, despite the complexity of the decision process, the scheduled operation must be chosen according to several, often inconsistent criteria. Thus the problem is one of multi-criteria decision making. The scheduling system must have the capability to:

- Minimize idle times.
- Ensure some overlapping of operations.
- Handle high priority jobs.
- Give preference to jobs according to their stage of completion.
- Minimize work-in-process.
- Meet various technological and other constraints in the system.
- Consider availability to multiple machines, tools and other resources.

Generally, scheduling rules are highly inter-related and can be conflicting as they cannot be achieved concurrently. In order to cope with them realistically, decisions should be made upon mixed but hierarchical scheduling goals, viz.:

- Satisfy all rush orders 100% of the time.
- Maximize utilization of bottlenecked machines.
- Minimize setups.

Special care should be taken as:

- Processes can be linear (where operation **'n-1'** is performed before operation **'n'**) or they can be non-linear (network having several predecessors).
- Batches can be of varying sizes.
- Many operations are performed one part at a time but some, like annealing and cleaning are done on batches at the same time. (Therefore, the system must be intelligent enough to realize that the time taken to perform the operation can sometimes be independent of the number of pieces in the batch).

- Fixtures, tools and operators may have availability patterns different from the machine availability pattern.
- There may be changeover costs, especially in sequence-dependent set-ups.
- There may be sequence dependent segments (tear down or cleanup for example).
- Intangibles are what make a system acceptable to the user. e.g.: user defined priority for a customer order that has been pre-paid, or assigning a setup operation between 8 and 12 am because the manager wants to look it over.

> In his 1990 article on CIM, Lee states the following:
> "There are a lot of work and effort done in the job shop scheduling and control. However, much of this work has not found practical application because of the difference between theory and the practice. In addition, the job shop scheduling and control problem has been studied as several separated problems such as sequencing, dispatching, load balancing or lot sizing problem with composite measures. There is a real need for an integrated and computerized approach to solve the real-life scheduling and control problems rather than the concern for optimality."

Due to the short lead-time, complexity, volume of information and lack of planning tools, shop foremen are forced to resolve the problem with little more than intuition and experience (Svetka and Jiang, 1990). Thus the real-time control of shop floor planning and scheduling for construction projects is an area of research with very high potential. The high cost of inventories and the great variety of product options demanded by the market make its importance clear. In this environment, manufacturers are forced to carry no finished inventory and operate in a mode where small batches or even single units are virtually made to order.

1.5 Research Methodology

The problem is a prime candidate for application using computers and can form the basis of a Decision Support System (DSS) because:

- Scheduling is a decision-making function.
- It is a knowledge intense activity.
- The number of permutations and combinations easily overburdens human schedulers.
- Existing computer based algorithms and heuristics incorporate only a fraction of a scheduler's knowledge.

DSS is an interactive information technology (IT) based system that is intended to help decision-makers utilize data and models in a semi-structured decision situation. DSS's are meant to be an adjunct to decision-makers, i.e.: to extend their capabilities but not to replace their judgment. The objective of the DSS is not only to obtain a short-term detailed schedule of the operations, but also to make the system capable of rapid response to various disturbances and dynamic changes to the system. Thus, a DSS supports complex decision-making and increases its effectiveness. Little (1970), states that in order to be successful, a DSS must be:

1. Simple.
2. Robust.
3. Easy to control.
4. Adaptive.
5. Complete on issues.
6. Easy to communicate with.

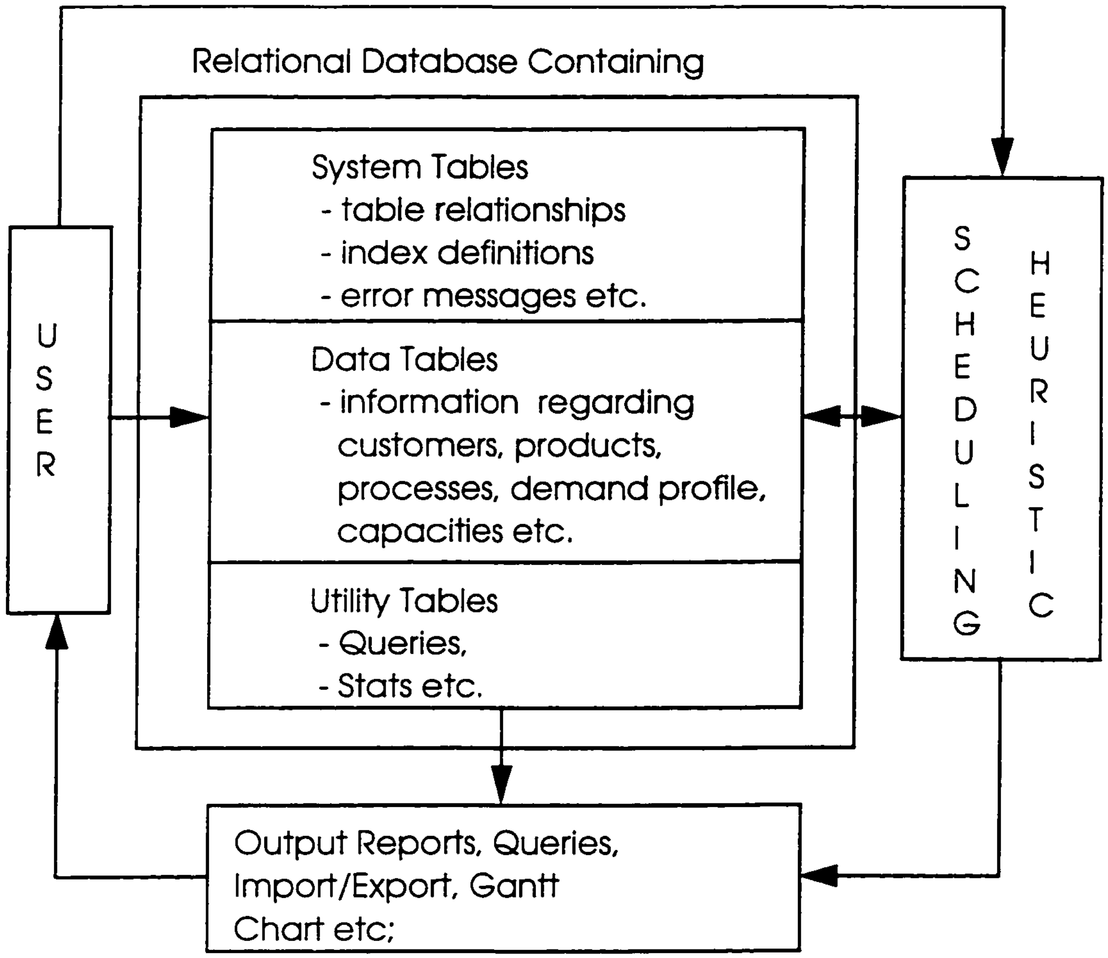


Figure 1-1: The Framework for the Proposed Decision Support System

The development of the DSS as shown in Figure 1-1 involves the following activities:

- Requirement identification and data structure design.
- Design and development of a relational data base that will contain all of the relevant information regarding the customer orders, the product structure and the process information, capacity (work centers, machines, operators and other resources), and the scheduling parameters (rules, and schedule horizon).
- Development and testing of a finite capacity based, resource constrained scheduling heuristic.

- Design of user interfaces for input/output, screens and menu design, reports and queries.

The data base structure has been designed to answer a wide variety of user requirements in an efficient manner and is shown below in Figure 1-2.

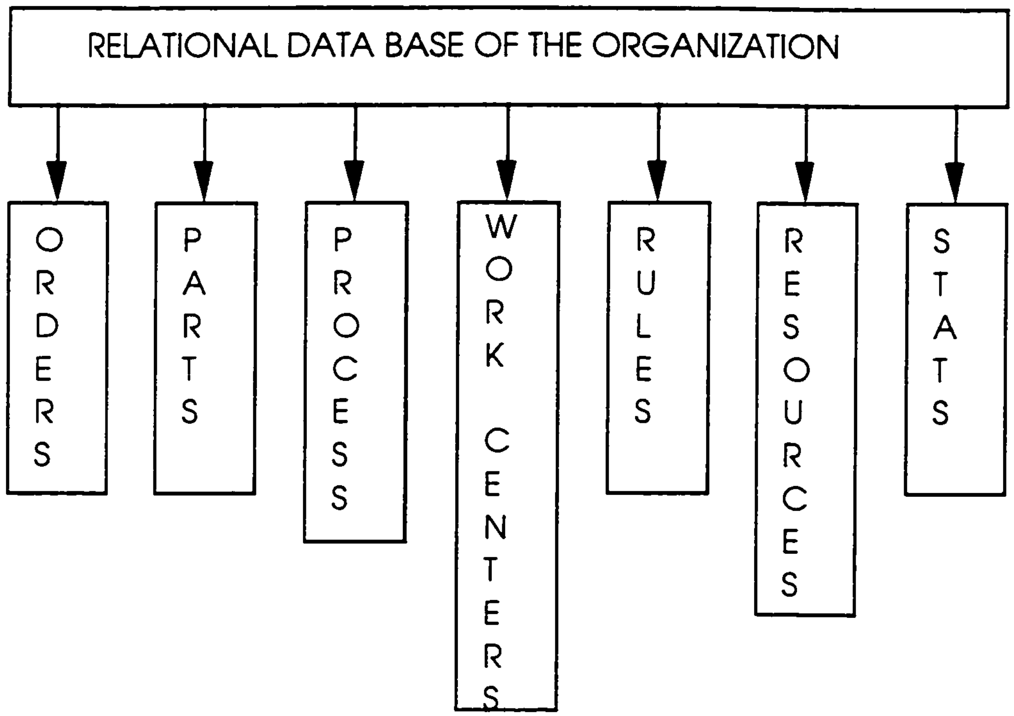


Figure 1-2: Overview of the Database Information

As a demonstration of this research, a DSS will be designed and developed to deal with a specific scheduling problem for planning and control of steel fabrication projects. It will be shown that a DSS approach to scheduling problems can be a useful in filling the gap between theory and practice. The DSS can be a very valuable tool for operational scheduling situations, particularly since the DSS supports all phases of the decision process from the initial data preparation phase to the scheduling and management presentation phase. The strength of the system is its ability to handle virtually all real-life situations in a realistic manner and quickly generate feasible solutions, then improve them if possible. In industrial production scheduling we can have several hundred jobs which are to be scheduled. Each job involves a number of operations - usually 5 to 10. An operation consists of a visit to a work group at which the job is processed in some way, typically by performing one or more segments of the operation. The capacity of each work group may vary in regards to the number of parallel machines and the hours of work. The shop may involve assembly job routings in which parts and assemblies have to be produced. Job precedence constraints may be defined in which each job may have any number of immediate predecessor and successors. Operation precedences within a job are not necessarily sequential; some other relationships, such as parallel, free start, simultaneous start, direct follows, lagged start, and time-sharing of facilities, are valid. Also, there may be transport time between operations.

The scheduling heuristic developed for this thesis is a job oriented heuristic (JOH). It will always create a realistic schedule as it schedules jobs currently in progress and planned jobs to actual capacity. The schedule results will be the scheduled start and completion times of each production operation, calculated to the nearest minute. Various printed and screen displays and reports can be generated as required. The schedule output can be saved as a scenario and so 'what-if' type of analysis can be easily done by changing the scenario name and comparing the different scenarios on critical areas of interest. (e.g.: utilization, average queue time, number of operations scheduled, and number of orders late if one runs 1 shift, 2 shifts, or 3 shifts).

Studies of individual decision making have found that, when faced with a complex, un-programmed situation, the decision-maker is very constrained by his/her own cognitive abilities. As a result, the decision-maker seeks to simplify the situation - to split the big decision into sub-decisions, and to reduce these into elements sufficiently simplified that familiar, general, interchangeable sets of procedures can apply. Hence, the decision-maker deals with unstructured issues by changing them into familiar, structurable elements. Furthermore, the individual decision-maker uses a number of problem solving habits, such as seeking solutions that are satisfactory instead of optimal, not looking too far ahead, and reducing a complex environment to a series of simplified conceptual models. For the DSS to answer questions such as which existing customer order should be delayed to accommodate a new rush order, a multi-criteria-decision making methodology on the lines Multi-Attribute Utility function theory taking into account the multiple objectives and their relative importance to the decision maker has been developed.

1.6 Research Contribution

The purpose of this research is to investigate the special requirements of the steel fabrication industry and to design and develop a DSS for scheduling that will enable us to close the gap between scheduling models in research and real scheduling problems faced by the industry. The DSS can take practical information into account and has the capability to solve complicated scheduling problems.

Job shop scheduling problems are important in production planning. The job shop represents a common manufacturing environment, in which a set of machines is used to process streams of jobs arriving randomly. The stochastic arrival pattern of the jobs, their processing sequence and processing times render the scheduling of jobs in a job shop a complex and difficult problem. Furthermore, the scheduling task involves multiple and conflicting objectives such as minimizing flow times, lateness, number of jobs being processed and so on. This research will enable the development of a framework for the efficient scheduling of jobs in the presence of these multiple objectives.

An attempt to bridge the gap between theory and practice has been made in this research. The solution sought is to come up with a tool that provides good, feasible schedules in real-time taking into account all real-life scheduling constraints. The objective of the DSS is not only to obtain a short-term detailed schedule of the operations, but also to make the control system capable of rapid response to various disturbances and dynamic changes to the production system. The system uses JOH, which are computationally efficient for large-scale problems. It also uses an improvement heuristics, which improves upon the JOH where possible. The supported decision situations include several real-life features, which are disregarded by conventional scheduling models such as time-sharing of facilities and scheduling job families. The system supports the planner in constructing efficient and effective schedules. Several features, including operation splitting, stay-on-line, minimize moves, JIT, and minimize setups have been incorporated into the DSS. These procedures will enable decision makers on the shop floor to manage problems involving due dates, rush jobs, customer inquiries, and quotation of due dates.

Furthermore, with tools like the scheduling DSS, a decision-maker will be in a better position to study the other aspects of the scheduling problem, such as the problem of which job to activate from the backlog. Since the problem of determining which jobs to activate and the creation of the operational schedule are directly related, their performance should be jointly assessed. To begin this task, the order release method should use some intelligent means to select jobs to be activated.

1.7 Organization of Thesis

The thesis is organized as follows. In chapter 2, a brief introduction to scheduling, heuristics, Decision Support Systems (DSS) and research focus is presented. In chapter 3, the main scheduling heuristic procedure is discussed. In chapter 4, the design and development of the DSS for the operational scheduling problem is presented. In chapter 5, a plan for implementation is presented through a case study approach. Finally, chapter 6 highlights the lessons learned and identifies and recommends areas for future research.

A detailed survey of the literature on scheduling theory can be found in Appendix 1. The details of the database structure, relationships and fields are presented in Appendix 2.

2. Overview of Scheduling, Heuristics and Decision Support Systems

2.1 Scheduling

Scheduling problems arise in situations where a set of activities must be processed using a limited number of resources. Scheduling refers to the selection of activities to perform (if more than one exists) and the assignment of actual times and resources. Scheduling is the process of sequencing a known set of goals, whereas planning achieves higher-level goals by generating sub-goals. When scheduling steel fabrication/construction projects, one must consider jobs (projects) content, resources and constraints. A comprehensive introduction to scheduling can be found in Pinedo (1995). As mentioned earlier, scheduling is implemented as part of production planning and control functions. Schedules serve as a guide for production and establishing manufacturing resource requirements in terms of manpower, machines, toolings, fixtures, materials, and possibly other specific facilities (Newman, 1988). A survey of 40 practicing schedulers conducted by McKay, Safeyeni and Buzacott (1988) reveals that one of the reasons scheduling research has no impact on real-world situations is the fact that most scheduling studies simulate unrealistic environments and make impractical assumptions.

2.1.1 The Job Shop Process

Job shop scheduling (JSS) is an important model in scheduling theory; it serves as the test bed for new algorithmic ideas and provides a starting point for more complicated and practically relevant scheduling models. Much of the research literature on sequencing refers to the job shop-scheduling problem and uses the terminology of manufacturing: job, machine, operation, routing and processing-time. The basic unit of the job-shop process is the operation. One can envision an operation as an elemental task to be performed. An operation usually has three attributes: one pertaining to the job, another pertaining to the machine where the operation is to be performed, and the last pertaining to the duration of the operation (the processing time). A machine is intuitively a device or facility capable of performing whatever it is that has to be done in an operation, but abstractly, a machine is just a time scale with certain intervals available. A job shop is a set of all machines that are identified with a particular set of operations. A job shop process consists of machines, jobs (operations), and a statement of the disciplines that restrict the manner in which operations can be assigned to specific points on the time scale of the appropriate machine. Scheduling the job shop process is the task of assigning each operation to a specific position on the time scale of the specified machine.

The general job shop problem is a fascinating challenge. This scheduling problem is in essence very simple to state: "*a set of '**n**' jobs with different priorities must pass through '**m**' processes in such a way that some objective function(s) is (are) optimized.*" Although it is easy to state and visualize what is required, it is extremely difficult to make any progress whatsoever towards an optimal solution. The following discussion presented by Biegel and Davern (1990) characterizes just how quickly the problem becomes too large to solve.

> "Assume that the world is 20 billion years old. This means that the world is 20*10^ 9*365.24 * 60*60 = 6.31152*10^ 17 seconds old, or 6.3*10^ 23 micro second old. If we had a computer system capable of producing and evaluating a job shop schedule every microsecond since the beginning of time, we could have produced only 24! or 6.2*10^23 schedules. In terms of a one machine scenario, this means that since the beginning of time we could have optimally scheduled only 24 jobs in the job queue waiting for processing on our single machine."

Certainly the above discussion suggests just how big the JSS problem is, especially when we are faced with the '**n**' task, '**m**' machine scenario. Add alternate routes, transfer lots, and resource constraints, and the complexity increases.

2.1.2 Approaches to Scheduling

From the point of view of scheduling research, the nature of the problem studied can be categorized in a number of ways, viz.:

1. Number of machines (single machine case, two-machine case, multiple machine case).
2. Finite capacity or infinite capacity (and resource constraints).
3. Nature of the shop (flow shop or generalized job shop.
4. Nature of jobs and job related information processing (deterministic, static, stochastic, or dynamic).
5. Performance criterion (minimize tardiness, minimize mean flow time, or some other criteria).

We can group scheduling approaches into three broad categories, viz., the conventional, knowledge based and distributed approaches as shown below in Figure 2-1.

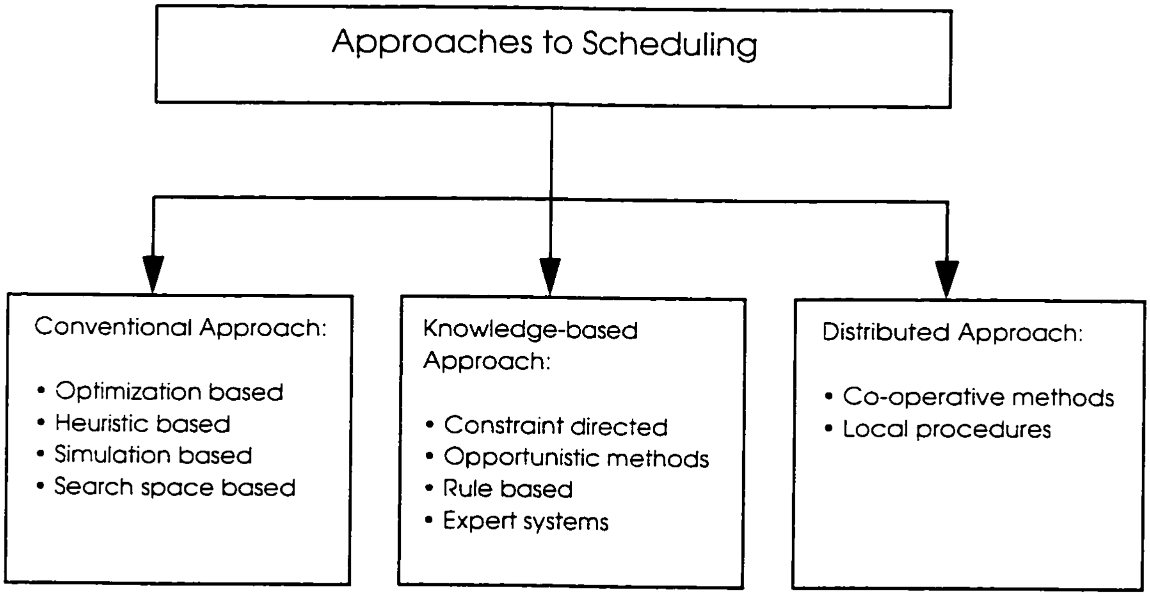


Figure 2-1: Approaches for Scheduling

A detailed review can be found in Appendix 1. The scope for mathematical programming is limited because of the complexity of the scheduling environment. Although many of the constraints can be represented mathematically, soon, with only a few of the situations considered, the problem becomes intractable. Hence much of the research using these methods involves highly simplified versions of the actual problem (King, 1976). All deterministic scheduling problems are combinatorial optimization problems. There are four major classes of solution methods for combinatorial optimization problems: complete enumeration, exponential time algorithms, polynomial time algorithms, and approximation algorithms. Each class of solution method yields a different level of efficiency and accuracy. One useful technique for scheduling problems has been heuristics. Heuristics have the advantage over branch and bound in terms of efficiency, and the advantage over polynomial time approximation in terms of widespread applicability. In general, however, heuristics offer no guarantee that the solution is within an acceptable margin of error when compared to the optimal solution.

2.2 Heuristics

Optimal solutions to some complex decision problems could involve a prohibitive amount of time and cost, or the task itself may even be impossible. Alternatively, simulation approaches may be lengthy, complex and even inaccurate. In such situations, it is sometimes possible to arrive at 'satisfactory' solutions more quickly and less expensively by using heuristics. While heuristics are used primarily for solving ill-structured problems, they can also be used to provide satisfactory solutions to certain complex, well-structured problems. The main difficulty in using heuristics is that they may result in a poor solution.

Heuristic thinking does not necessarily proceed in a direct manner. It involves searching, learning, evaluating, judging, and then searching again, re-learning, and re-appraisal as exploring and probing takes place. The knowledge gained from success or failure is fed back at some point and modifies the search process (Rowe, 1988). The heuristic procedure can also be described as finding rules that help to solve intermediate sub-problems in order to discover how to set up these sub-problems for final solution. This is achieved by finding the most promising path in the search for solutions, finding ways to retrieve and interpret information on each experience, and then finding the methods that lead to a computational algorithm or general solution.

According to Pearl (1984), a logical approach to heuristic rules incorporates:

1. A classification scheme that introduces structure into the problem.
2. An analysis of the characteristics of the problem elements.
3. Rules for selecting elements from each category to achieve efficient search strategies.
4. Rules for successive selections, where required.
5. An objective function that is used to test the adequacy of the solution at each stage of selection or search.

2.2.1 Problems in Using Heuristics

Geoffrion and Van Roy (1979) identify the following shortcomings of heuristics:

1. Enumeration heuristics that consider all possible combinations in practical problems can seldom be achieved.
2. Sequential decision choices can fail to anticipate future consequences of each choice.
3. 'Local improvement' can short-circuit the best solution because heuristics, similar to simulation, lacks a global perspective.
4. Interdependencies of one part of a system can sometimes have a profound influence on the whole system.

2.2.2 When to use Heuristics

According to Zanakis and Evans (1981), the use of heuristics instead of optimization is appropriate in the following scenarios:

1. The input data is inexact or limited.

2. Reality is so complex that optimization model is oversimplified.
3. A reliable, exact method is not available.
4. The computation time of optimization is too excessive.
5. It is possible to improve the efficiency of the optimization process (by proving good starting solutions using heuristics).
6. Problems are being solved frequently (and repeatedly), consuming computer time.
7. Complex problems exist that are not economical for optimization or take too long and the heuristic can improve the non-computerized solution.
8. Symbolic rather than numeric processing is involved.

2.2.3 Advantage of Heuristics

The major advantages of heuristics are that they:

1. Are simple to understand and therefore easier to implement.
2. Help in training people to be creative and come up with heuristics for other problems.
3. Save formulation time.
4. Save programming and storage requirements on the computers.
5. Save computer-running time.
6. Produce multiple solutions.

2.2.4 Heuristics for Industrial Scheduling

Heuristics provide a viable approach to job shop scheduling problems of realistic size. Two types of heuristics can be used: (1) operation oriented heuristics, often called dispatching rules, and (2) job oriented heuristics (JOH), in which all the operations of a job are scheduled before the next job is considered.

2.2.4.1 Operation Oriented Heuristics

In this class, scheduling decisions are made at each machine by selecting the next operation from a queue of operations awaiting processing when the machine becomes idle or available. Based on a dispatching rule, priorities are assigned to the operations in the queue, one is selected from the set of schedulable operations based on the priority. There are many heuristic dispatching rules; over 100 are listed in the literature (Panwalkar, 1977). A substantial amount of research has been done in evaluating dispatching rules. However, based on the results of past scheduling research, no dispatching rule dominates all other rules with regard to all criteria. It is, therefore, clear that the proper selection of scheduling rules is a critical activity.

2.2.4.2 Job Oriented Heuristics

Instead of working with individual operations by dispatching rules, job oriented heuristics (JOH) schedule one job at a time, so that all the operations of the job are scheduled before the next job is considered. The list of jobs to be scheduled is first prioritized and the job with the highest priority is scheduled completely. The next job is then scheduled without affecting the previously scheduled jobs and so on. Very little research has been reported on this class of heuristics (Hastings and Yeh, 1990). White (1986) has conducted an extensive experiment designed to examine the overall performance comparison between JOH and operation oriented heuristics. His findings show that JOH is generally superior to the six commonly used dispatching rules in terms of overall quality of the schedule. Another finding reported in his study is that JOH consumes far less computational resources than the equivalent dispatching rules. No evidence indicates that JOH should be eliminated from consideration in either scheduling research or scheduling practice. Indeed, from the research cited, the following conclusions regarding JOH can be stated:

- They are computationally efficient.
- They produce good quality of schedules, as confirmed by experimental testing conducted by white (1986).
- They are intelligible to production personnel, in that they facilitate the control of individual jobs.
- They are well suited to situations involving job precedence constraints arising from products in which components and assemblies have to be made.
- They can easily accommodate changes in job status due to changing priorities, rework or shortage of materials.

It is fair to state that the advantageous features of JOH suggest that it still remains the most promising approach to industrial scheduling, even at the present time.

2.3 Decision Support Systems (DSS)

Scott-Morton first articulated the concepts involved in DSS in the early 1970s under the term 'Management Decision Systems'. He defined such systems as "*interactive computer based systems, which help decision makers utilize data and models to solve unstructured problems*". Another classical definition of DSS provided by Keen and Scott-Morton is "*Decision support systems couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions. It is a computer based support system for management decision makers who deal with semi structured problems*". It should be noted that DSS, like Management Information Systems (MIS) and other Management Support System (MSS) technologies, is content free expression. There is no universally accepted definition of DSS, but Turban (1993) has provided the following useful working description:

"A DSS is an interactive, flexible and adaptable computer based information system, specially developed for supporting the solution of a particular management problem for improved decision making. It utilizes data, it provides easy user-interface, and it allows for the decision maker's own insights. It also utilizes models, it is built by an iterative process, it supports all the phases of the decision making, and it includes a knowledge base" (Turban 1993, page 87).

The DSS description clearly indicates that there are differences between DSS and MIS. Typically, DSS's have the following features:

- A DSS can be used to address ad hoc, unexpected problems.
- A DSS can provide valid representation of the real world system.
- A DSS can provide decision support within a short time frame.
- A DSS can evolve as the decision-maker learns more about the problem.
- Data processing professionals are not necessary for the development.

2.3.1 Decision Making Strategies

Within a decision making process, there are various strategies that a decision-maker can use to organize his or her efforts. The process is affected by the strategy being used to choose an alternative. Among the well-known decision making strategies are:

- Optimizing.
- Satisficing.
- Elimination-by-aspects.
- Incrementalism.
- Mixed scanning.
- First feasible alternative.
- Selection based on the power of proposers and/or supporters.

- Consensus.
- Majority rule.
- Least worst (maxi – min).
- Worst best (maxi – min).
- The analytic hierarchic process (AHP).

A particular decision-maker in a particular decision process may use one of the above methods, a hybrid approach, or an entirely different strategy. A DSS designed to support the use of optimization may be of little help when a satisficing strategy is being used, and vice versa. Every DSS will not be ideally suited to supporting every strategy. Nevertheless, it would be an advantage for a decision-maker to have a DSS sufficiently flexible and adaptable to aid in pursuing whatever the current strategy happens to be. The model base of the scheduling DSS being designed will contain various modules that will fall into optimizing, satisficing, and Heuristic categories.

2.3.2 Characteristics and Capabilities of DSS

There is no agreement on the characteristics and capabilities of DSS because there is no consensus on what a DSS is. However, most DSS's will have some or all of the following features.

1. DSS provides support for decision-makers mainly in semi-structured and unstructured situations by bringing together human judgment and computerized information. Such problems cannot be solved entirely by other computerized systems, such as MIS, or by management science models.
2. Support is suitable for various managerial levels ranging from top executives to line managers.

3. Support is provided to both individuals and groups. Many organizational problems involve group decision making. The less structured problems frequently require the involvement of several individuals from different departments and organizational levels.
4. DSS provides support to several inter-dependent and or sequential decisions.
5. DSS supports all phases of the decision making process: intelligence, design, choice and implementation.
6. DSS is adaptive over time. The decision-maker should be reactive and able to confront changing conditions quickly, adapting the DSS to meet these changes. DSS's are flexible so the user can add, delete, combine, change, or rearrange basic elements (providing fast response to unexpected situations). This capability makes timely and quick ad hoc analyses possible.
7. DSS is easy to use. Users must feel 'at home' with the system. User-friendliness, flexibility, strong graphic capabilities, and an English-like dialog language can greatly increase the effectiveness of DSS. This ease of use implies an interactive mode.
8. DSS attempts to improve the effectiveness of decision-making (accuracy, timeliness, and quality), rather than its efficiency (cost of making the decision, including the charges of computer time).
9. When solving a problem, the decision-maker has complete control over all steps of the decision making process. A DSS specifically aims to support, not replace the decision-maker. The decision-maker can over-ride the computer's recommendation at any time in the process.

10. DSS leads to learning, which leads to new demands and the refinement of the system, which leads to additional learning, and so forth, in a continuous process of developing and improving the DSS.
11. DSS's are relatively easy to construct. End users should be able to construct simple systems by themselves. Larger systems can be built in users' organization with only minor assistance from information system specialists.
12. A DSS usually utilizes models (standards, custom made). The modeling capability enables experimentation with different strategies under different configurations. Such experimentation can provide new insights and learning.
13. Advanced DSS's are equipped with a knowledge component that enables the creation of efficient and effective solutions for very difficult problems.

2.3.4 Components of DSS

A conceptual model of a DSS can be developed on the lines shown in Figure 2-2. It provides a basic understanding of the general structure and components of a DSS. New technology continues to affect the dialog, data, and model components. For example, icon-based or touch screen systems provide new options for directing the system. Relational database technology and, more recently, object-oriented databases, and data warehousing are influencing how data is stored, updated and retrieved.

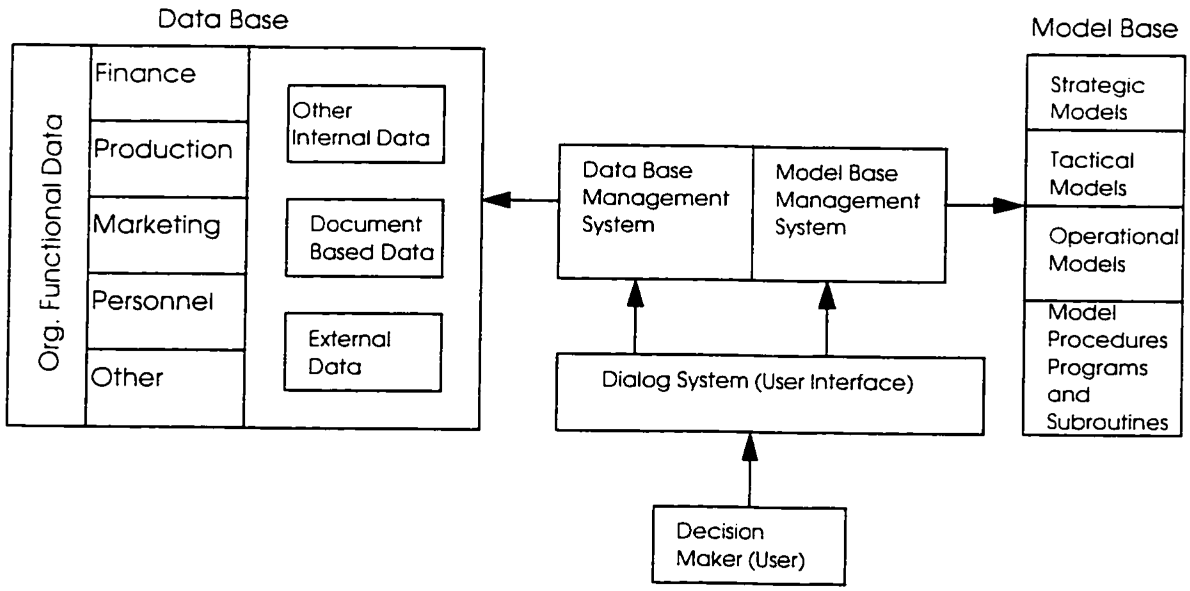


Figure 2-2: A Conceptual Framework for a DSS

1. Data Management: This includes the database(s), which contain relevant data for the situation and is managed by software called 'database management system' (DBMS).
2. Model Management: A software package that includes financial, statistical, management science/OR models, heuristic programs that provide the system's analytical capabilities, and an appropriate software management methodology.
3. Communication Subsystem (Dialog management / user interface): The user can communicate with and command the DSS through this subsystem.

2.4 Observations

In addressing the dynamic scheduling (sequencing) problem, closed-form mathematical and optimization procedures have difficulty capturing the complexities and being solved in a reasonable amount of time. Therefore heuristic approaches to the problem are a preferred solution. The observations based on the survey of literature can be summarized as follows:

- Research using the conventional methods has not established how the best scheduling rule can be selected given a rule set applicable in a case.

- The approach taken by the knowledge-based, distributed problem solving methods have combined both scheduling and controlling and satisfying solutions have been proposed.
- Traditional assumptions such as that one machine can only perform one operation no longer hold true.

Newman (1988) lists the essential characteristics of a good scheduling system as:

- It must be robust enough to handle exceptions and be efficient in terms of meeting due dates and production costs.
- It must be able to use knowledge of constraints, preferences and current information about the environment during schedule generation.
- It must provide enough flexibility to react to disruptions in an efficient and timely manner.
- Scheduling decisions must be based on actual shop conditions in order to increase the efficiency of the system in real-time.

Parunak (1991) has identified five challenges for a scheduling system, viz.:

1. Method of evaluating a schedule (to determine the best one).
2. Accommodating unpredictability.
3. Reducing the computational burden.
4. Handling chaotic behavior.
5. Handling cases that are probably unpredictable.

2. 5 Focus of the Research

It is clear from the review of literature that attempting to solve the real-life scheduling problem is a worthwhile endeavor, especially in the construction industry where virtually everything is made to order and where no "economies of scale" type of approaches can be applied. Scheduling in this environment is a very knowledge intensive activity requiring a vast array of data. A DSS for scheduling is an appropriate means of solving the problem. The general scheduling problem cannot be easily handled by optimization methods, but heuristic approaches can form the first steps in our understanding of the complexities of the shop floor. There are still a large number of valid research issues to be addressed. Specifically:

- There is a clear need for determining a better method for assigning job priorities. In the real world it is a common practice to assign priorities based on class or due-date. A better method for computing the priority of jobs is an area worth pursuing.
- There are multiple solutions to the problem when alternate processes are available; the evaluation of a given schedule in terms of a utility measure to the organization will be a valuable addition to the academic knowledge. On the basis of this utility score, rush jobs can be handled and delays of existing jobs can be justified.

3. The Scheduling Heuristic

3.1 Introduction

In this chapter, the applicability of heuristics for use in the planning and scheduling of steel fabrication/construction projects is studied. The following will be used: a generalization of the job shop scheduling problem in which operations can be processed at several alternate machines and operations requiring machine sets for processing. A Utility theory based scheme is developed for choosing the appropriate schedule based on management's requirements.

3.2 The Scheduling Problem - Revisited

At the operational level of a production plant, the scheduling problem can be simply stated as:

"*A set of* '***n***' *jobs with different priorities have to pass through* '***m***' *processes (machines) in such a way that some objective function(s) is (are) optimized.*"

In other words, a detailed schedule of the production system is to be constructed. At a higher hierarchical planning level, a certain set ('***n***') of jobs is chosen to be scheduled in a specified period, typically a week or a day. The scheduler must decide when to load each job into the system and to which machines ('***m***') the job has to be assigned in order to carry out its activities. The processing sequence in which each machine has to perform the activities must also be chosen as demonstrated in Figure 3-1. Although it is easy to state and to visualize what is required, it is extremely difficult to arrive at an optimal solution.

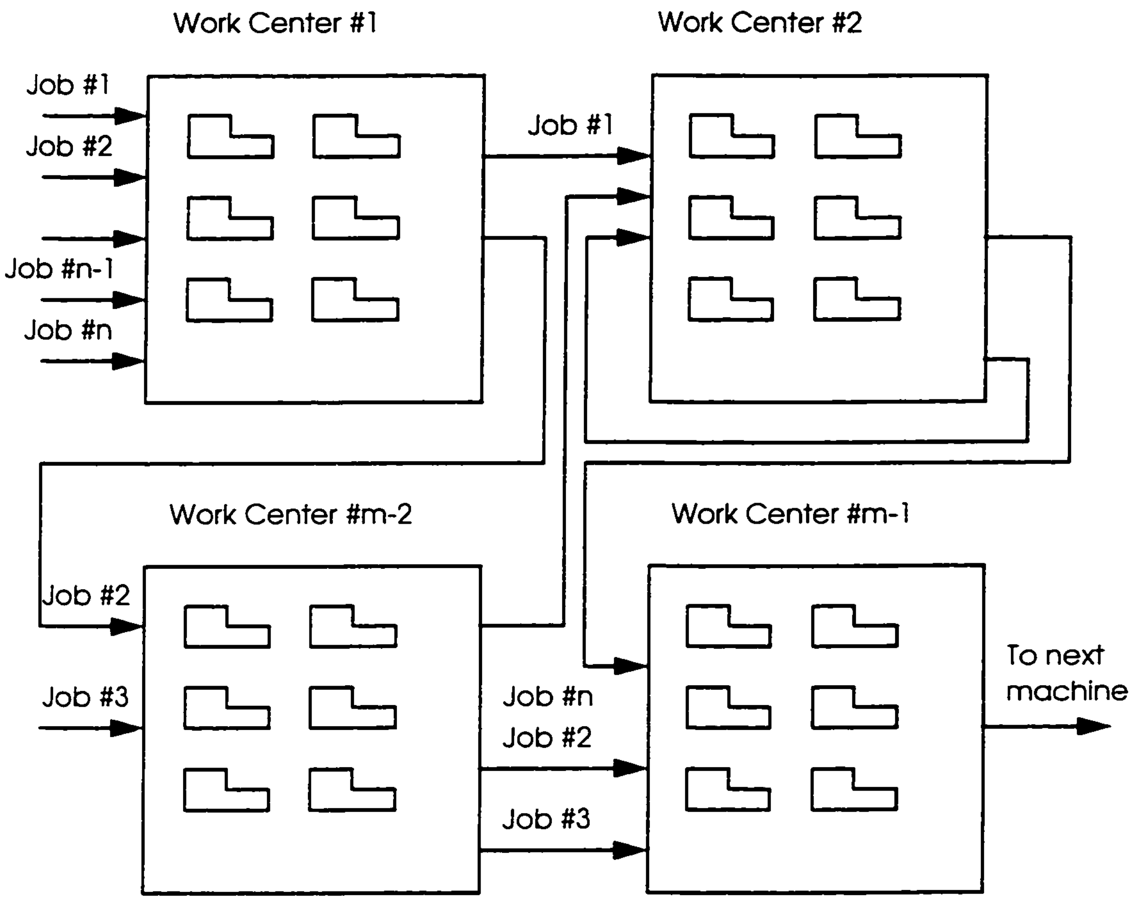


Figure 3-1: Overview of the Scheduling System

The characteristics of the system can be described as follows. An instance of Job shop scheduling problem (JSSP) consists of a set of jobs and a set of machines. Each machine can usually handle at most, one job at a time. Each job (or shop order) consists of a chain of operations (activities) which need to be processed in that order during an uninterrupted (if so desired) time period of a given length on a given machine. Each activity can be further broken into specific tasks or segments (e.g.:preparation, setup, run, clean up, and tear down) which are carried out on one of several alternate machines in a work center. The production system consists of different work centers. Each work center may have one or more machines capable of performing the same set of activities. However, the time taken for the activity (i.e. the processing time), and additional support resources that may be needed, will depend on the particular machine chosen for the activity. Each machine and the additional support resources will normally be available during certain time periods (i.e. they have different shift patterns) and can normally process only one job at a time. There may be special machines (like furnaces or shot blasting machines) which can handle more than one job at the same time.

The data associated with the jobs comes directly from the higher planning level. A release date, due date, priority factor, job class, and number of units to be produced are associated with each job and can be considered to be attributes of the order. There may be batch sizes associated with jobs and they may be different from the transfer batch size (the size for transferring between machines). In addition, a changeover time may be required between two operations. The time required can depend on the preceding operation on the same machine. Finally, there is a fixed sequence for the operations of the job. The problem is to find a schedule (defined as an assignment of the operations to time intervals) so that some objective, such as minimizing the total length of the schedule, is achieved (French, 1982).

There are various performance objectives for operational scheduling problems. One of the more important objectives is the achievement of due dates since the violation of due dates means a late delivery to customers. Other relevant objectives are machine utilization, the reduction of work - in - process, the reduction of lead times, and the reliability of the system with respect to unforeseen events.

3.3 Mathematical Formulation of the Model

Given a set **J** of jobs, a set **M** of machines, and a set **O** of operations; for each operation **a** ∈ **O**, there is a job **j(a)** ∈ **J** to which it belongs, a machine **m(a)** ∈ **M** on which it must be processed and a processing time **d(a)** ∈ **N**. Each machine **M** is available in predetermined time intervals. A release date **r(J)** and a due date **dd(J)** are associated with each job **J**. An objective is to be specified. Usually, it is to minimize the maximum lateness. If due dates are not specified, they can be assumed to be the same for all jobs and the objective becomes the minimization of the maximum completion times.

This problem is a generalization of the classical Job Shop problem and is among the hardest combinatorial optimization problems as it is NP-hard (Garey & Johnson, 1979). It is therefore natural to look for approximation methods that produce an acceptable schedule in a reasonable amount of time. In some special cases the above problem reduces to single machine problems, parallel machine problems, flow shop problems, or pure job shop problems. All of them are well known NP-hard scheduling problems.

A formalization of the problem suggests the presence of three different decision elements: (1) the assignment of operations to specific machines, (2) the determination of precedence constraints between the operations assigned on the same machine, and (3) the timing of the operation start and finish events. Obviously the decision elements are not independent of each other.

There are three general scheduling techniques that can be used in considering these components: one approach is job scheduling, which places jobs on a minimum time line (e.g. PERT / CPM usually used at the planning level to obtain overall schedules). The second method is activity scheduling in which the most important or most constraining jobs are usually scheduled first. The last frequently used method of scheduling is the enumeration of alternatives. Beyond two machines, the job shop problem is inherently intractable in the sense that the existence of a polynomially bounded optimization algorithm is very unlikely. The difficulties of finding an optimal solution are entirely computational, as this is a combinatorial problem and solving for optimality would take an indefinite amount of time. Existing optimum seeking approaches can only solve problems of the order of 10 jobs on 10 machines (Lawler et al. 1982). Due to the complexity of the problem, the computational requirement is unacceptable for optimally solving instances of reasonable size. Therefore, a heuristic solution approach is justified. Two types of heuristics can be used:

1. **Operation Oriented Heuristics** (often called dispatching rules). In this class, scheduling decisions are made at each machine by selecting the next operation from a queue of operations awaiting processing when the machine becomes idle or available. Based on a dispatching rule, priorities are assigned to the operations in the queue and one is selected from the set of schedulable operations based on the priority.
2. **Job Oriented Heuristics (JOH)** (in which all the operations of a job are scheduled before the next job is considered). Instead of working with individual operations by dispatching rules, job oriented heuristics schedule one job at a time, so that all the operations of the job are scheduled before the next job is considered. The list of jobs to be scheduled is first prioritized; the job with the highest priority is scheduled completely. The next job is then scheduled without affecting the previously scheduled jobs, and so on. The ranking and prioritization scheme can include dispatching rules just as in operation oriented heuristics.

3.4 The Scheduling Heuristic

In this section an outline of the general features of practical scheduling problems that are encountered in industry and the description of the JOH for the scheduling system developed is presented. In subsequent sections, more detailed features of the system are discussed. As mentioned earlier, in industrial production scheduling, several hundred jobs are normally scheduled. Each job involves a number of operations, usually 5 to 10. An operation consists of a visit to a work group at which the job is processed in some way, typically by performing one or more segments of the operation. The capacity of each work group may be variable in regard to the number of parallel machines and the hours of work. The shop may involve assembly job routings in which part and assemblies have to be produced (Figure 3-2).

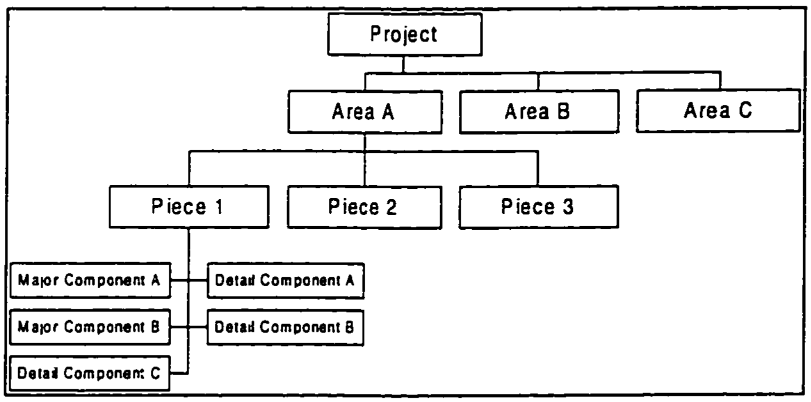


Figure 3-2: Typical Bill-of-material Hierarchy of a job in the construction industry

Job precedence constraints may be defined wherein each job may have any number of immediate predecessors and successors. Operation precedence within a job is not necessarily sequential; some other relationships such as parallel, free start, simultaneous start, directly follows, lagged start, and time-sharing of facilities are valid. There may be transport time between operations.

Scheduling Heuristic Pseudo Code

Phase 1:
Identify scheduling horizon, scheduling rules and schedule filter. These parameters are read from the user provided inputs. For example, 20 days horizon, DUE DATE, CRITICAL RATIO rules and ALL orders to be considered. If there are no orders satisfying the criterion specified, output message and quit.

Phase 2:
Prioritize Orders. Based on the user specified criterion, read all orders from the order master file and rank them according to the chosen scheduling rule. The first step in the application of the heuristic is to develop a ranking for each customer order (job). Each customer order will have a due-date, a priority factor, an order class and other attributes specified. For example, the order class can be defined as OVERDUE, RUSH, PLANNED, HOT, and STOCK. The user can specify an ordering of these factors. The priority factor is another scheme to rank orders and can be a numeric scheme from 1 to 9, 1 being the most important and 9 the least. Thus, a ranking scheme based on Class, Priority and Due-date will sort customer orders by class first, priority next and due-date last.

The next step is to assign ranks to the customer orders. The sorted list is scanned and the first job is picked for scheduling. The first operation for this job and the resources required for the completion of this operation is identified. If the resources are available the operation is scheduled on the resource (set) that completes the job at the earliest time. This procedure is repeated for subsequent operations and if they can be scheduled then the order is given a numeric rank, say 1000. The next job is then selected and the procedure repeated. If it can be scheduled completely it is given a numeric rank, say 2000. This process is repeated for the remaining jobs; they receive ranks of 3000, 4000 etc. Jobs that are not scheduled at all get the highest possible numeric rank 2147483647 (2^{31} - 1).
If a different ranking scheme is desired (say the Critical Ratio (CR) is chosen by the user as the basis for scheduling), then the critical ratio is calculated for each job as:
CR = (Due date - present time)/Processing time remaining for job completion.
These are then sorted in ascending order and the ranking procedure is repeated. Similarly the user can select any combination of rules as the basis for scheduling. Adding user specific decision rules can easily expand the model base. If user wants to view this ordering, it may be presented for him/her to make changes. The default is to proceed without any user intervention.

Phase 3:
Read in Updates to shop status, operations moved by users and other user specified information. In this phase shop progress reported is included to update the schedule. Other requirements like order precedence and networked operations are also verified and read to account for proper technological ordering.

Phase 4:
```
Initialize Availability Pattern for each machine and each supporting resource
Do Until End of Scheduling Horizon
        For Each Machine and supporting resource Do
            Read shift pattern specified for the machine and resource. Create the resource
            availability profile specifying periods when the machine/resource is available, down
            shift periods, blocked by user for maintenance etc.
        Loop until end of all machines and support resources.
Loop
```

Phase 5:
In this phase the actual scheduling of the operations is performed.

```
Do Until End of Scheduling Horizon
    For Each Order Do
        Identify Process to be followed
            For Each Operation Do
                For Each Machine Alternate Do
                    For Each Operation Segment Do
                        Identify when each operation segment can be
                        scheduled taking into account when the
                        segment can be started and the support
                        resource requirements. Select the machine
                        alternate that gives the earliest completion
                        time and schedule the segment on that
                        machine and the supporting resource.

                        Check for special requirements, like
                        maxseggap and maxsegdly to account for
                        breaks in schedules. Check for other features
                        like automatic elimination of setups if
                        previous part same as this part on the
                        machine, minimize moves, JIT, stay on line
                        requests etc.

                        Write out schedule, resource consumed and
                        update machine utilization. If Segment
                        cannot be scheduled within the scheduling
                        horizon, flag reason and exit to next order.
                    Loop Until End of All Segments
                Loop Until End of All Machine alternates
            Loop Until End of All Operations
    Loop Until End of All Orders
Loop
```

The mechanics of the heuristics described above can be stated as follows:

1. Read user inputs to identify the scheduling horizon, scheduling rules to be applied and any scheduling filters to be applied for screening order information. If no order exists, flag message and quit.
2. Identify the jobs (orders) to be scheduled, then prioritize and rank the jobs.
3. Read production progress reported for updating schedule, operations moved by users, order precedence if any, networked operations if any, and other special user inputs.

4. Identify the work centers/machines and support resources, and their shift pattern. Create the availability profile showing periods of availability, down shift, blocked by user for maintenance etc.
5. Identify the processes to be followed for each job.
6. Initialize job availability time, machine availability time, machine schedule, tools schedule, fixture schedule, and TIME the current epoch.
7. Select the highest ranked job.
8. Select the next operation for the job (during the first pass, this is the starting operation).
9. Schedule the operation at the earliest time by checking job availability time, processing route, machine availability, and tool / fixture availability. Select the earliest available machine to process the job operation and determine when to setup machine, then update machine schedule, tool schedule, fixture schedule, machine availability time, and job availability time (for next operation). Check for special conditions like breaks in schedule, stay-on-line criterion, JIT request, and automatic elimination of setup operation if current part is same as previous part. Write operation schedule to file. Update machine utilization time, total remaining processing time, tool schedule and fixture schedule (resources). If operation cannot be scheduled within the scheduling horizon, flag reason in status file, and proceed to step 11.
10. Identify the next operation for the job, and if one exists, return to step 9.
11. Select the next job to be processed. Return to step 8.

12. If all jobs have been considered or if the end of the scheduling horizon is reached for all jobs, compute Work-in-process, Tardiness (lateness) and Utilization, then stop.

See Figure 3-3 for a visual explanation of the above scheme. The scheduling system always creates a realistic production schedule, as it schedules jobs currently in progress and planned jobs to actual capacity. The schedule results are the scheduled start and completion times of each production operation, calculated to the nearest minute. Various printed and screen display reports can be generated as required. The schedule output is saved as a scenario, so 'what-if' analysis can be easily done by changing the scenario name (similar to saving a word processing document under a different name) and comparing the different scenarios on critical areas of interest. For example, utilization, average queue time, number of operations scheduled, and number of orders late if 1 shift, 2 shifts, or 3 shifts are run. Sometimes, special production scheduling problems can be solved separately from the main production schedule.

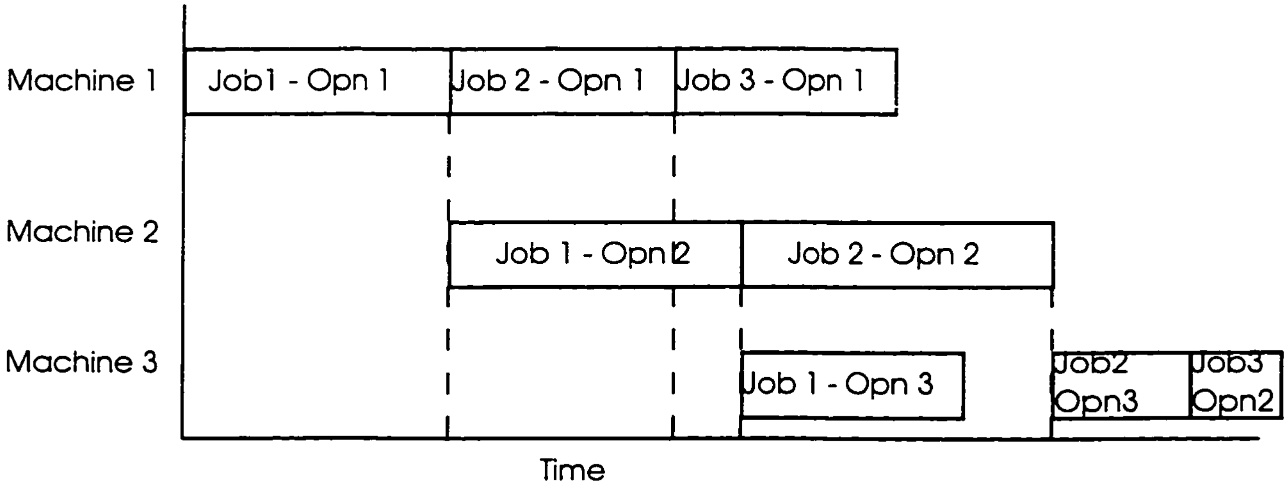


Figure 3-3: A typical Schedule of Jobs

As mentioned earlier, JOH scheduling can be either forward, backward or hybrid. Forward scheduling generates a tight schedule with good resource utilization, but some jobs may finish unnecessarily early, resulting in additional inventory holding costs. Backward scheduling schedules jobs to finish close to their due dates resulting in fewer inventories holding costs. However, gaps of unused time may be created at early stages of the schedule, causing problems in immediate shop floor control. The potential conflict of scheduling objectives between high resource utilization and low inventory costs obviously cannot be resolved by forward or backward scheduling alone. Moreover, jobs to be scheduled in an actual industrial production setting, especially in make-to-order production systems, often have specific scheduling needs. For example, rush jobs always have to be finished as soon as possible. Important jobs have to be finished by the due date, and it may be prudent to avoid risking being late. Expensive jobs, which have high inventory value, should not finish unnecessarily early. Long due date jobs should not be started too early. Figure 3-4-1 to 3-4-4 shows the scheduling approaches for an assembly job.

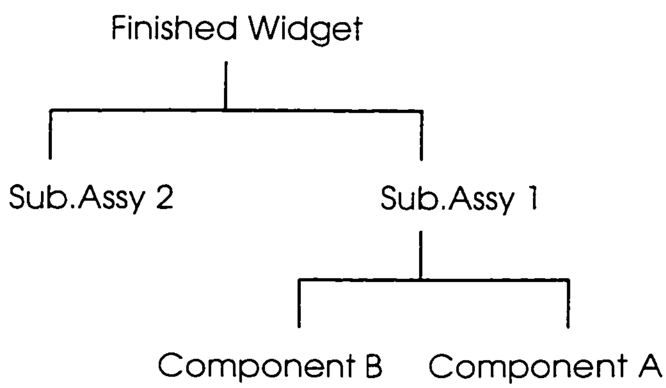


Figure 3-4-1: A Bill of Material structure for a finished product (assembly)

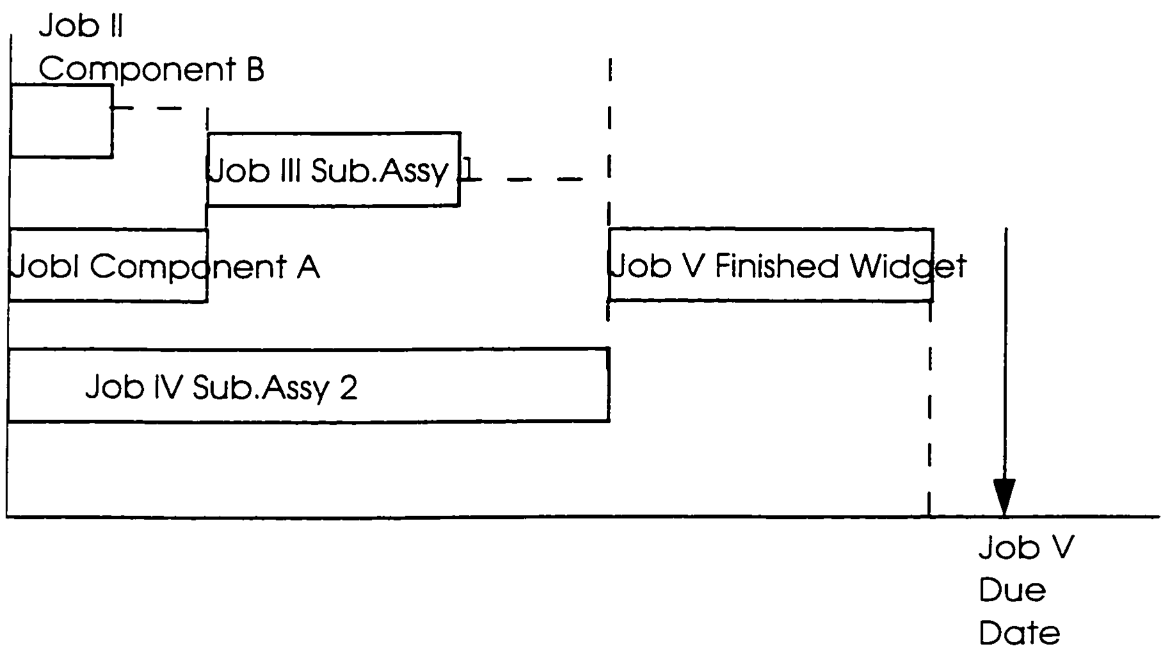


Figure 3-4-2: Forward scheduling method for an Assembly

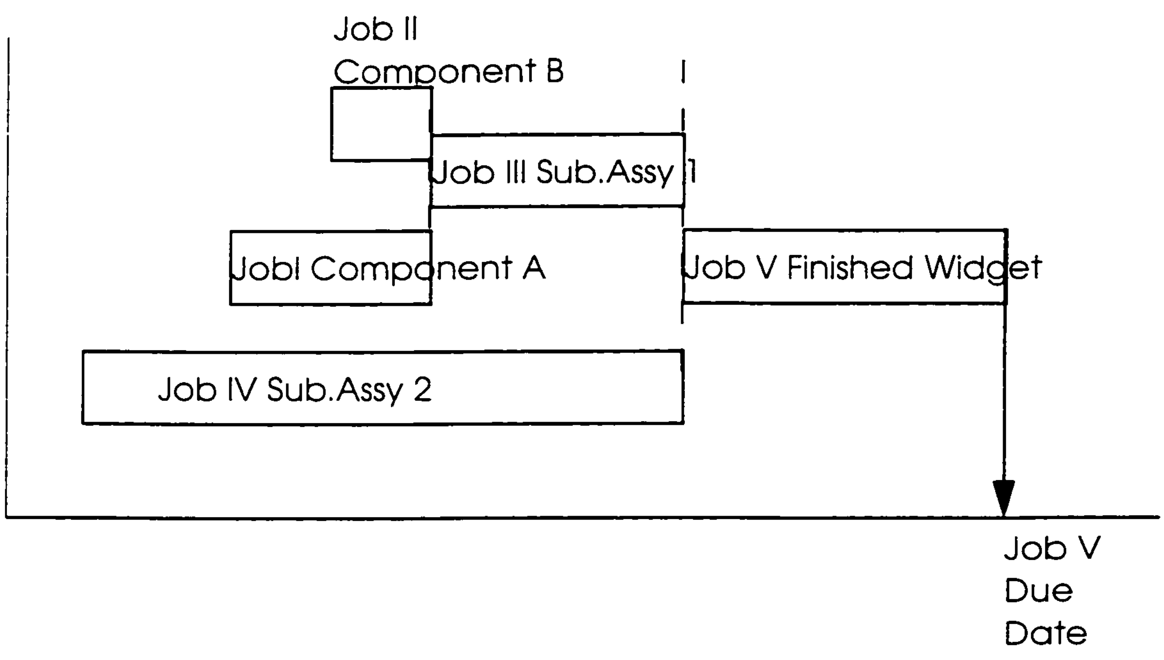


Figure 3-4-3: Backward scheduling method for an Assembly

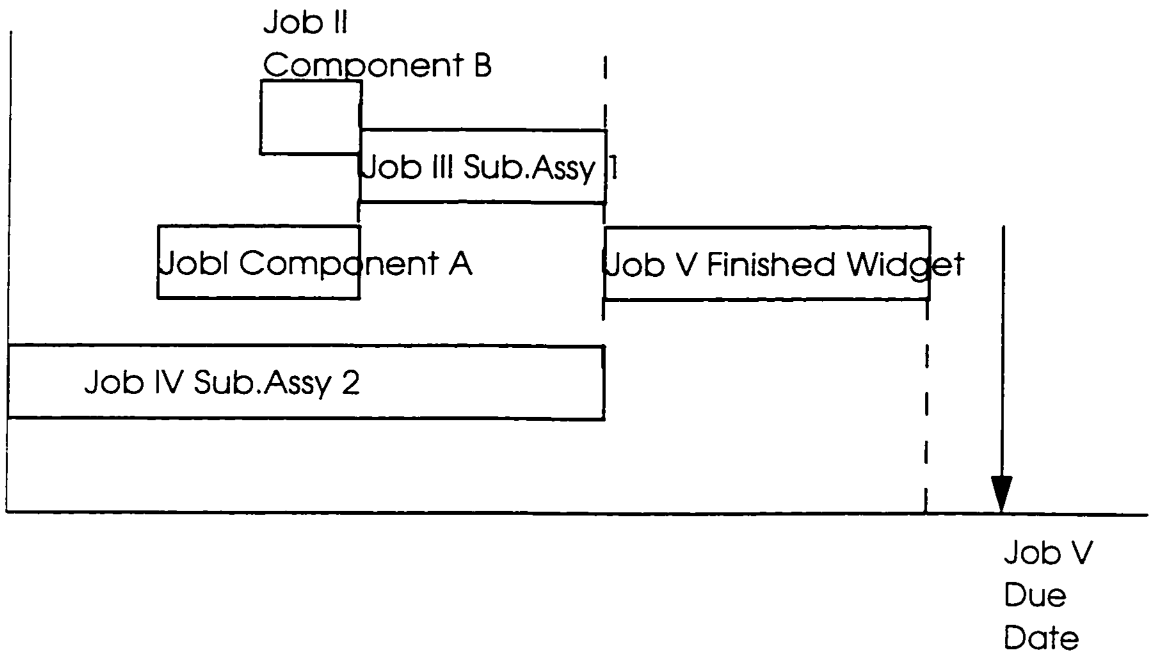


Figure 3-4-4: Hybrid scheduling method for an Assembly

In order to meet particular objectives of individual jobs, a number of scheduling procedures are developed and incorporated in the system. This provides the advantage of versatility in choosing the proper scheduling procedures for each job. Forward scheduling is normally applied to rush jobs and jobs that have relatively short due dates and/or low inventory value. Forward scheduling can also be used for finding out whether the earliest feasible completion time will meet a customer's requirements. Forward scheduling creates a tight schedule at early planning periods. Production managers generally regard it as important to keep resources busy in the short term. Forward scheduling is the default followed by the scheduling heuristic, as keeping the shop resources fully utilized in the short term, combined with frequent re-scheduling requirements make it the preferred alternative.

Both forward and backward scheduling involve a single-pass procedure which assigns operations of a job one by one in a single direction either forwards or backwards on to the first available time slot on the corresponding machines. As a result, forward scheduling always schedules operations of a job towards the first operation, while backward scheduling always schedules towards the last operation. Due to previously scheduled workload this will not always give a minimum in process time for a job. For example, one can modify the schedule shown in Figure 3-2 as shown in Figure 3-5, by delaying Job2 slightly to enable Job3 to finish earlier.

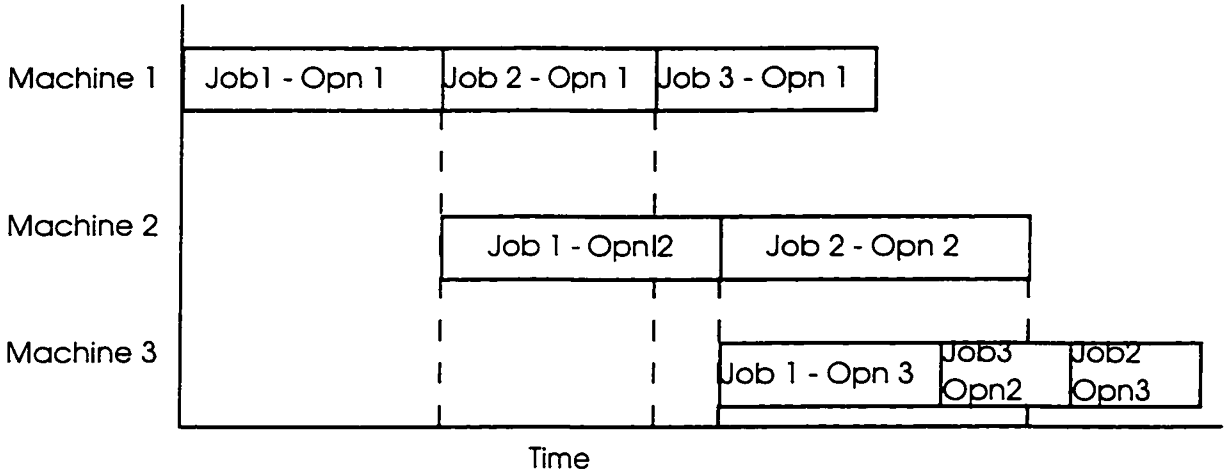


Figure 3-5: Modified schedule of Jobs

3.5 Handling of Network Type Problems

The scheduling heuristic can handle network type problems that are usually handled through, custom tools like Primevera or MS Project. The advantage of the heuristic over these specialized tools is its ability to prioritize different projects and then schedule them based on the chosen priority.

A sample network problem is shown in Figure 3-6 and the relevant information is presented in Table 3-1.

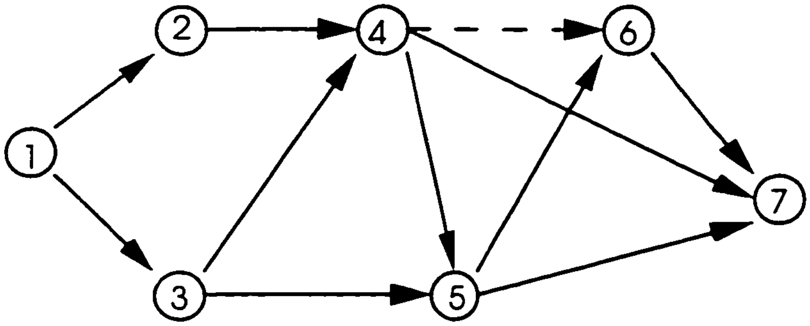


Figure 3-6: A Sample Network Problem

Table 3-1: Data for Sample Network Problem

Activity	From Node	To Node	Duration	Resource Required
1	1	2	3	1
2	1	3	4	2
3	2	4	3	2
4	3	4	4	1
5	3	5	3	1
6	4	5	2	2
7	4	6	0	0
8	4	7	3	0
9	5	6	4	0
10	5	7	7	3
11	6	7	6	0

Based on this information, the solution to the network problem using the Critical Path Method (CPM) technique is as shown in Table 3-2.

Table 3-2: Solution to the Sample Network Problem

Activity	Duration	ES	LS	EF	LF	TF	Critical
1	3	0	2	3	5	2	N
2	4	0	0	4	4	0	Y
3	3	3	5	6	8	2	N
4	4	4	4	8	8	0	Y
5	3	4	7	7	10	3	N
6	2	8	8	10	10	0	Y
7	0	8	14	8	14	6	N
8	3	8	17	11	20	9	N
9	4	10	10	14	14	0	Y
10	7	10	13	17	20	3	N
11	6	14	14	20	20	0	Y

The databases of the DSS is set up in such a way that this problem and others similar to it can be handled in one of two ways:

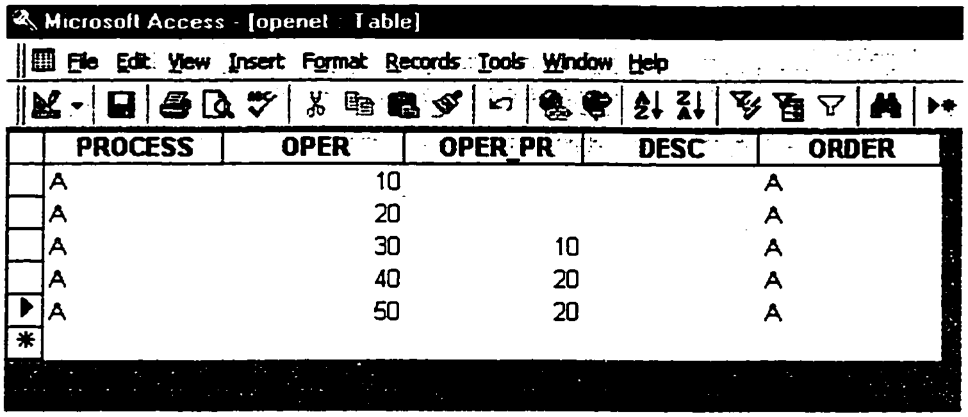
1. **Activities as Operations:** The network can be modeled as operations of a process (say A) for the order (say A) defined in the database and showing the precedence relationships. This information is shown in Tables 3-3 and 3-4 and Figure 3-7.

Table 3-3: Activities modeled as operations

Activity	Operation Number
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
9	90
10	100
11	110

Table 3-4: Showing the precedence relationships

Operation	Previous Operation
10	
20	
30	10
40	20
50	20
60	30
60	40
70	30
70	40
80	30
80	40
90	50
90	60
100	50
100	60
110	70
110	90



PROCESS	OPER	OPER_PR	DESC	ORDER
A	10			A
A	20			A
A	30	10		A
A	40	20		A
A	50	20		A

Figure 3-7: Database Table for Modeling Network as Operations

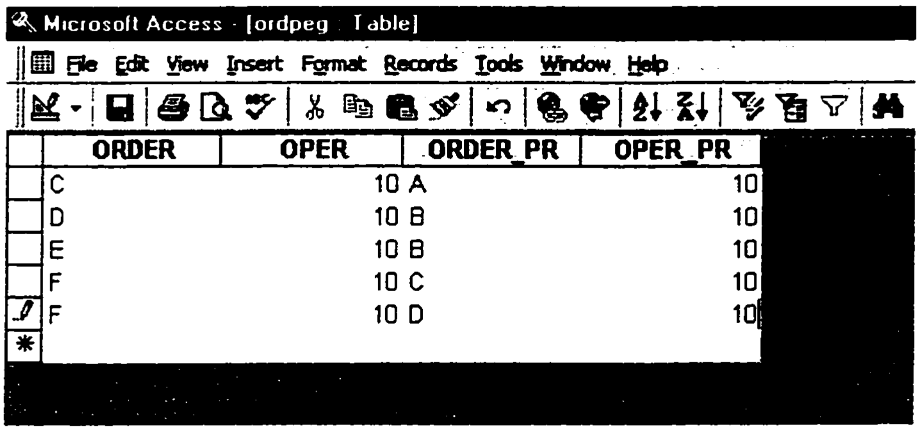
2. **Activities as Orders**: Each activity of the network can be modeled as a separate order (having just one operation) and then the orders can be pegged to each other to account for the precedence relationships. This information is shown in Tables 3-5 and 3-6 and Figure 3-8.

Table 3-5: Activities modeled as orders

Activity	Order Number
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H
9	I
10	J
11	K

Table 3-6: Showing the precedence relationships

Order	Previous Order
A	
B	
C	A
D	B
E	B
F	C,D
G	C,D
H	C,D
I	E,F
J	E,F
K	G



ORDER	OPER	ORDER_PR	OPER_PR
C	10	A	10
D	10	B	10
E	10	B	10
F	10	C	10
F	10	D	10

Figure 3-8: Database Table for Modeling Network as Orders

In either cases, the heuristic will first read-in the information and sort it to form the correct precedence relationships. The next step will identify any errors in network representation – a cycle, for example. The heuristic has a simple but powerful method to check for cycles. The procedure is as follows: First, an adjacency matrix showing the arcs of the network is created. Next, if a column or a row has no entries, the corresponding row and column are removed from the matrix and the process repeated. If there are no more rows or columns, then there is no cycle, otherwise there is a cycle in the network. This procedure is demonstrated below using simple networks shown in Figures 3-9 and 3-10.

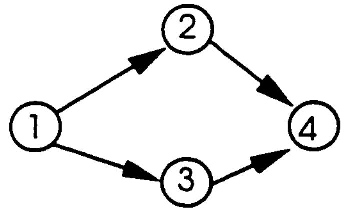


Figure 3-9: Sample network without any cycles

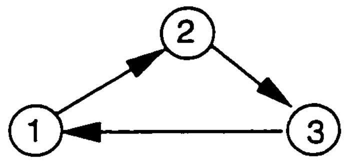


Figure 3-10: Sample network with cycles

Step1: Create the adjacency matrix.

Case 1: Based on Figure 3-9

From/To	1	2	3	4
1		1	1	
2				1
3				1
4				

Case 2: Based on Figure 3-10

From/To	1	2	3
1		1	
2			1
3	1		

Step 2: If a row or column has no entries, delete corresponding row and column.

Case 1: Based on Figure 3-9. Row four has no entries. Removing row 4 and column 4 gives a reduced matrix shown below.

From/To	1	2	3
1		1	1
2			
3			

Repeating the process, eliminate rows 2 and 3 along with columns 2 and 3. This will leave a 1x1 matrix, which can be eliminated too, as there is no entry from 1 to 1. Thus the network shown in Figure 3-9 has no cycles.

Case 2: Based on Figure 3-10. There are no rows or columns having no entries. This suggests that there is a cycle in the network!

The next step is to create a forward and reverse star list and then to perform the activity start and finish calculations. The output of this phase is shown in Tables 3-7 and 3-8.

Table 3-7: Activity Start and Finish Times

Activity	Start Date	Finish Date
1	0	3
2	0	4
3	4	7
4	4	8
5	7	10
6	8	10
7	8	8
8	8	11
9	10	14
10	10	17
11	14	20

As mentioned earlier, the advantage of the scheduling heuristic over standard packages like Primevera and MS Project is that, one can specify priorities to different projects and the heuristic will schedule following these priorities. Apart from handling, network type problems the heuristic is designed to handle a number of situations described below.

Table 3-8: Resource Consumption Profile

From Date	To Date	Resource Consumption
0	3	3
3	4	2
4	7	3
7	8	2
8	8	3
8	10	3
10	11	2
10	11	3
11	14	3
14	17	3
17	20	0

3.6 Handling of a Variety of Scheduling Requirements

To handle specific scheduling needs, a number of requirements must be considered in the design of heuristics. The following are a few of the most important considerations.

3.6.1 Operation Overlapping

Operations of a job can often run parallel to each other, subject to technical constraints. This scheduling procedure allows the user to specify that operations overlap and to indicate any delay caused by minimum quantity sent ahead from the preceding operation, in addition to 'normal' transport time. This approach is useful when attempting to minimize the cycle time of orders.

3.6.2 Operation Splitting

Operation splitting divides a batch operation into sub-batches on available machines. The processing time of the operation is divided according to the size of the sub-batches. The operation setup time required by each sub-batch is multiplied by the number of sub-batches created. While the concept of splitting is usually followed in traditional manufacturing plants where batch sizes are large, the same approach can be used in the construction industry to solve another problem. It is usual for several welders to be working on the same job at the same time, this can be modeled as batch splitting, enabling the total welding operation to be shared by different welders.

3.6.3 Partial Scheduling:

The aim of partial scheduling is to quickly produce a schedule for a limited number of jobs, say all jobs for a specific customer or schedule for the next two days. When preparing a practical work schedule for shop floor control, it is desirable that the scheduling system be run at frequent intervals. This permits the rapid production of work schedules for the following shifts or days, and also provides guidelines for the best course of action to be taken to deal with changes in shop floor conditions. Partial Scheduling limits the number of operations and jobs to be scheduled in 2 ways.

3.6.4 Limited Schedule Horizon

The schedule horizon is restricted to a much shorter period, say, one or two days. The shorter the scheduling horizon given, the lesser the number of potential jobs to be scheduled, and hence less computer processing time is required to produce the schedule. The number of jobs to be scheduled by this method is normally unknown before scheduling. This method is applied to situations where only the schedule results within a certain period are of concern.

3.6.5 Limited Number of Jobs to be scheduled

Only jobs that have relatively high loading sequence are considered. The number of jobs to be scheduled by this method can be worked out before the scheduling procedure takes place. This method is applied to situations where only the schedule results of important jobs are of interest.

3.6.6 Single Job scheduling:

This scheduling technique is designed for two purposes:

1. Scheduling a new job into an existing schedule with a specified loading sequence.
2. Rescheduling a job in an existing schedule by different job parameters and/or scheduling procedures.

Single job scheduling starts from a completed schedule created by an earlier scheduling run. The procedure is as follows: (1) determine the loading sequence of the target job in the existing schedule. (2) de-schedule jobs whose loading sequence is later than that of the target job. (3) schedule the target job, considering only the remaining workload in the existing schedule.

Where there is a need to quickly examine or re-examine new realistic schedule times for a particular job on the basis of an existing schedule, single job scheduling is applicable. Applications arise in relation to sales inquiries, negotiations with customers regarding proposed due-dates and considerations of re-routing or splitting urgent orders.

3.6.7 Preferred Sequence Scheduling

The sequence of work generated by a heuristic procedure may not ideally suit the efficient operation of individual work groups. There are many applications in which work groups have their own preferred sequence for performing various jobs, e.g., light to dark in cases similar to scheduling paint manufacturing, or thin to thick in cases similar to rolling steel. The general problem of efficient scheduling with sequence dependent setup times is a complex one and cannot be optimally solved. To handle this type of scheduling, the system has been designed to accept up to six user defined parameters that will be used to sort and group the jobs according to these user specified functions. The sorting phase alone may not be sufficient to guarantee a preferred sequence and if possible, some other method may be needed to modify the sorted parameters and achieve the desired sequence. The scheduling heuristic has been designed so that, for preferred sequence scheduling, the schedule will be generated and presented to the user before it is written to file. It is up to the user to accept/reject the schedule, as this is a very complex problem for the heuristic to solve on its own.

3.6.8 Time Sharing of Facilities

As mentioned earlier, the system is capable of handling machines, that can process more than one job at a time, like furnaces for example. The system prepares a list of job operations that need to pass through such machines and their availability in terms of their arrival time at the facility and the capacity they consume. (For example, Job A may need 30% of the furnace, Job B may need 45% of the furnace etc.). Once an objective is chosen, say to maximize furnace utilization, the job information can be used to automatically formulate a mathematical model. Bala's 0-1, algorithm can then be triggered to solve this as a bin-packing (knapsack) problem, jobs are grouped and processed as batches and after coming out of such machines proceed normally through other machines.

The next step is to evaluate the schedule and improve it if possible. In this phase, each scheduled job is evaluated in terms of lateness, resource utilization, order value, and customer importance. This phase is required because the heuristic will schedule jobs as long as it is within the scheduling horizon even though it may finish late. Therefore, a search has to be done to identify those jobs that are late and an attempt should be made to move jobs around so that any possible improvements to the schedule can be made.

3.7 Measures of Heuristic Performance

There are a number of ways to evaluate the performance of the scheduling heuristic. Some good performance measures included are as shown in Table 3-9.

Table 3-9: Sample of Performance Measures for Comparing Schedules

	Current Schedule	Previous Schedule	Previous2 Schedule
Number of tardy jobs (tardiness is defined as 0 if the job is early and the time late if the job is late)	4	2	16
Average tardiness	2.3	1.9	0.8
Dollar days late (sum of the number of late days times the dollar value of the job)	23000	19000	8000
Total profit	14000	12000	13970
Mean flow time	16.2	14.8	20.6
Utilization of shop facilities	77%	79%	96%
Number of orders/operations scheduled	14/56	14/56	14/62

A variety of scheduling rules have been built in the system. However, they alone may not resolve all problems. A sample list of order is ranked under different scheduling objectives and the result is presented in Table 3-10.

Table 3-10: Ranking by different scheduling objectives

Order Numbers Ranked According to				
Due Date	**Class, Priority, and Due date**	**Number of Operations**	**Critical Ratio**	**Order Value**
ORDER5	ORDER8	ORDER5	ORDER5	ORDER2
ORDER1	ORDER1	ORDER1	ORDER1	ORDER8
ORDER7	ORDER2	ORDER4	ORDER3	ORDER3
ORDER4	ORDER5	ORDER7	ORDER4	ORDER6
ORDER2	ORDER4	ORDER2	ORDER2	ORDER1
ORDER3	ORDER3	ORDER3	ORDER7	ORDER7
ORDER8	ORDER7	ORDER6	ORDER6	ORDER5
ORDER6	ORDER6	ORDER8	ORDER8	ORDER4

It is evident that, depending on the criteria chosen, the schedule may be different. There is a need to include other factors in the decision making process so that the organizational objectives are fully integrated into any scheduling decision. A Utility theory based approach is proposed to help users identify the proper scheduling rule to be employed.

3.8 Utility Theory based scheme for Evaluating Schedule Effectiveness

3.8.1 Introduction

Modeling management's value system is a critical activity. Decision theory is concerned with giving structure and rationale to the various conditions under which decisions are made. In general, one must choose between a host of alternatives, which scheduling rule to select, for example. These are referred to as actions (or strategies), and each results in a payoff or outcome, number of orders fully scheduled or machine utilization. If the decision-makers knew the payoffs associated with each action, they would be able to choose the action with the largest payoff. Most situations, however, are characterized by incomplete information, so for a given action, it is necessary to enumerate all probable outcomes together with their consequences and probabilities. The degree of information and understanding that the decision-maker has about a particular situation determines how the underlying problem can be approached. Two persons, faced with the same set of alternatives and conditions, are likely to arrive at very different decisions regarding the most appropriate course of action for them. What is optimal for the first person may not even be an alternative for the second. Judgment, risk, and experience work together to influence attitudes and choices. Implicit in any decision making process is the need to construct, either formally or informally, a preference order so that alternatives can be ranked and final choices made. For some problems this may be easy to accomplish. For example, profit maximization rule: in more complex situations, where factors other than profit maximization or cost minimization apply, it may be desirable to explore the decision-maker's preference structure in an explicit fashion and to attempt to construct a preference ordering directly.

In the scheduling system proposed, the effectiveness of any chosen scheduling rules can be analyzed by considering the organizational goals and objectives. These may include (i) order value, (ii) customer importance, (iii) percentage of orders scheduled, (iv) percentage shop utilization, and (v) percentage orders late. The list shown is by no means exhaustive, however, the general approach presented here can be used for a more detailed list. The number of orders scheduled, shop utilization and number of orders late cannot be easily pre-determined before a schedule. Therefore, the DSS will permit the user to select a number of scheduling rules of interest and present comparative figures, so that the user can select the appropriate scheduling rule for the current scheduling run.

This is a multi-criteria decision-making problem. Accepted solution methods vary between pure judgment, weighted scores, and joint analysis system (JAS) – a group evaluation method where each participant ranks the various factors on some scale and the combined scores are used to select an alternative. An important class of techniques that work by eliciting preference information from the decision-maker is predicated on what is known as *utility theory*. This, in turn, is based on the premise that the preference structure can be represented by a real-valued function called a ***utility function***. Once such a function is constructed, selection of the final alternative should be relatively straightforward. In the absence of uncertainty, an alternative with the highest utility would represent the preferred solution. For the case where outcomes are subject to uncertainty, the appropriate choice would correspond to that which attains the highest expected utility. Thus the decision-maker is faced with two basic problems involving judgment: (1) How to quantify (or measure) utility for various payoffs, and (2) How to quantify judgments concerning the probability of the occurrence of each possible outcome or event.

The procedure for structuring an evaluation model consists of four major activities (Shaffer Jr. et al., 1982) (see Figure 3-11):

1. Develop hierarchy of values.
2. Define criteria and scales for measurement.
3. Develop utility functions.
4. Formulate an objective function.

3.8.2 Develop Hierarchy of Values

Decision theory identifies the kinds of information needed to make a decision and the manner in which such information is to be combined and processed so that the results are meaningful for the decision to be made. However, the theory does not provide guidance for identifying and selecting an appropriate set of decision criteria. There is no generally accepted approach for specifying which decision criteria ought to be considered, and there is no objective automatically applicable procedure for determining whether a specified set of criteria is a good one (Shaffer Jr. et al., 1982).

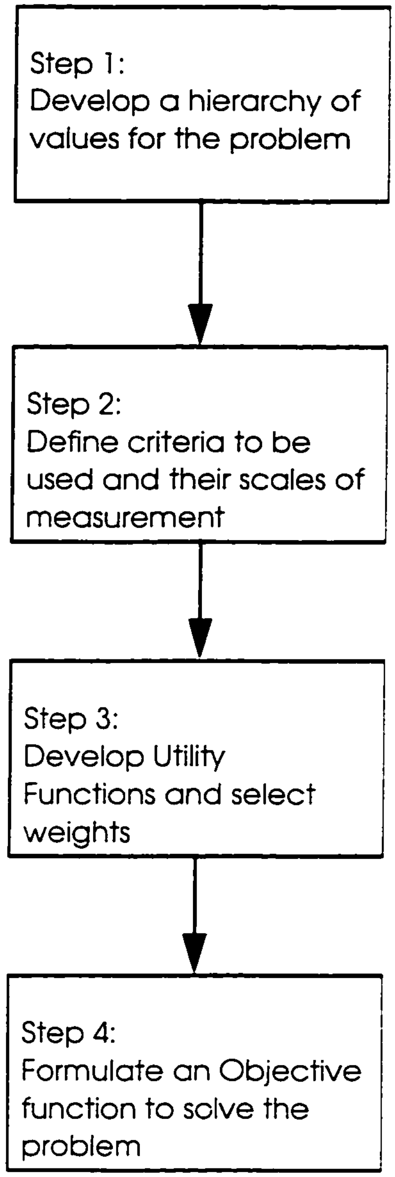


Figure 3-11: A Framework for Structuring and Evaluating Decisions

The approach followed to identify a good set of decision criteria is to structure a hierarchy of values (for scheduling, this could be broken down into major classes pertaining to customer, company objectives, and shop variables). The usefulness of the hierarchy derives from the observation that goals can be analyzed to define the essential factors contributing to their achievement. If necessary these contributing factors can be similarly analyzed. The process of partitioning intermediate goals into their component elements continues until the set of decision criteria is identified. The process provides a complete set of criteria that are consistent since they are derived using the decision-maker's goals and policies.

Structuring the hierarchy is purely judgmental. It requires an iterative search of all germane organizational objectives by people who are knowledgeable about the goals and purpose of the organization and of the decision-maker, for it is the decision-maker's value system that will be used in the actual decision. The tangible output of this effort is a set of explicit decision criteria that are derived from and consistent with the organizational goals and objectives and are, as far as management can determine, complete and significant for the decision to be made. Structuring the hierarchy also provides intangible, yet significant benefits, viz. (1) an in depth understanding of the pertinent value problem, (2) clarification of semantic and conceptual differences that frequently exist among the disparate people who are to work together, and (3) the capability to review, and update the set of criteria as well as the bases for their selection.

For analyzing the scheduling problem, we can consider two main categories, order related and schedule related. For example, order related issues could be broken down into order value and customer importance. Schedule related issues could be broken down into shop utilization, percentage of orders scheduled, and percentage of orders late. Figure 3-12 shows the hierarchical structure and the weight chosen. The values shown are based on discussions with managers and schedulers of the organization where the implementation case study was undertaken.

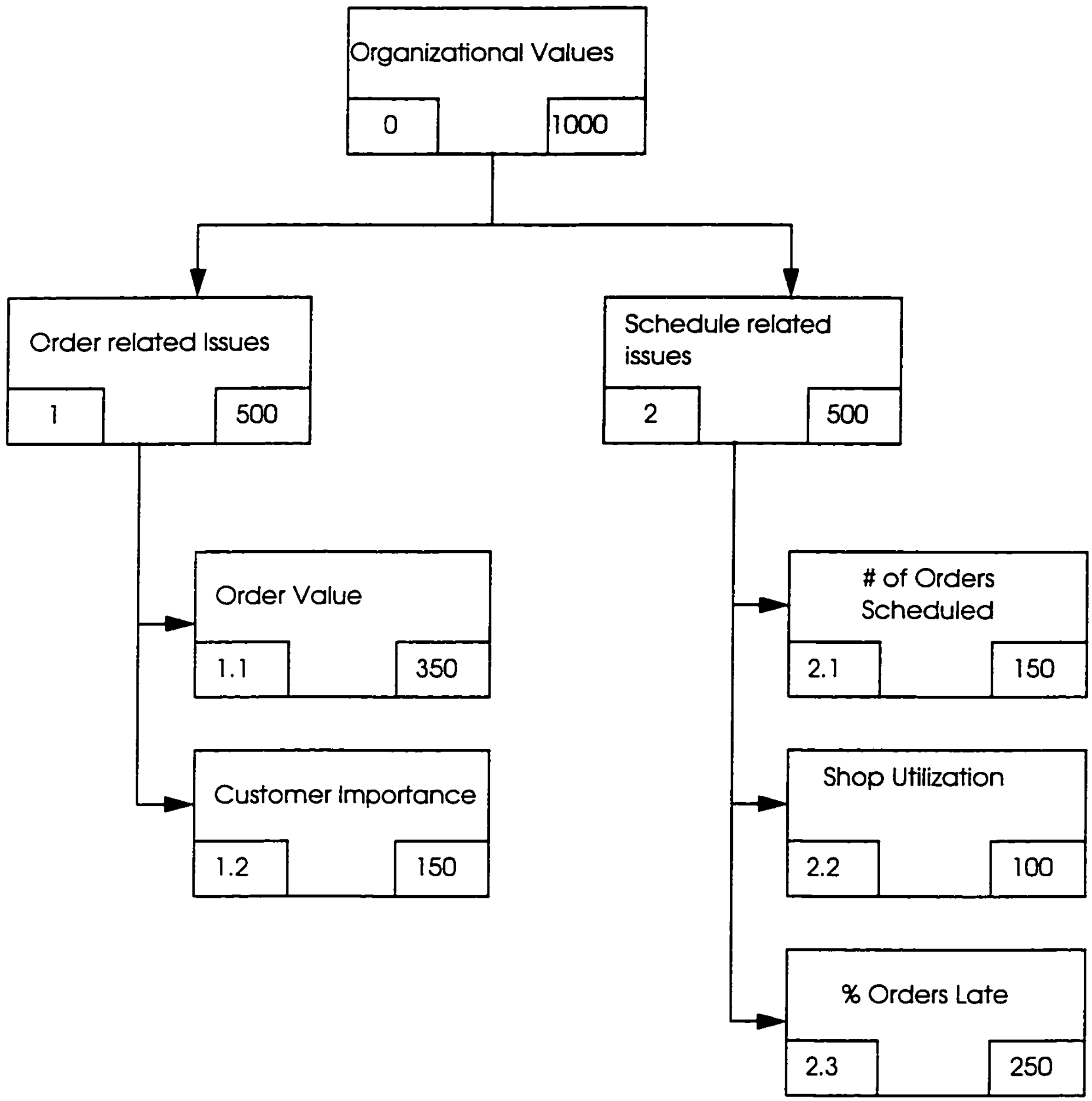


Figure 3-12: Hierarchical framework for sample problem

3.8.3 Define Criteria and Scales of Measurement

It is not sufficient merely to identify the criteria. Since the quantity (amount, level) of each criterion is to be estimated during analysis, and since a utility function is to be formulated for each criterion, the criteria must be unambiguously defined and their measurement scales must be specified. For example, job lateness, if selected as a criterion, can be measured on a scale of dollar days. Once the criteria are unambiguously defined, the decision-maker can delegate the actual estimation of these quantities to appropriate analysis personnel. For some criteria, neither definition nor specification of measurement scales presents a problem. Profits, for example, are well understood by everyone and are normally measured in dollars. Sometimes, however, criteria are not easily defined or their measurement scales are not easily specified. Such criteria (client's image for example) are usually grouped as 'qualitative' or 'intangible.' A measurement scale for an intangible criterion can be devised for a given decision situation. Identify the most favorable situation and assign it an arbitrary value of 100. Identify the least favorable situation and assign it a value of 0. Identify a situation that is midway from the earlier two and assign it a value 50. Repeat the process to refine the scale to include more points. This scale not only provides a measure for the criterion, but also contributes to the clarification of the meaning of the criterion by explicitly illustrating what various numbers on the chosen scale mean.

For the scheduling problem Table 3-11 shows the criteria chosen.

Table 3-11: Criteria Chosen for Consideration

Hierarchy Block	Criterion Designation	Criterion Name	Criterion Scale	Weight
1.1	Y1	Order Value	Dollars	350
1.2	Y2	Customer Importance	0-9 scale	150
2.1	Y3	Percent Orders Completed	Percent	150
2.2	Y4	Shop Utilization	Percent	100
2.3	Y5	Percent Orders Late	Percent	250

3.8.4 Develop Utility Functions

The utility function for the set of defined criteria should satisfy the following requirements:

1. The utility function for a given criterion should represent the decision-maker's preferences for various quantities of that criterion over the range of available choices.
2. The utility function for the set of criteria should represent the decision-maker's preferences for exchanges or trade-offs between the criteria.
3. The utility of the various criteria should be measured on some utility scale so that the expected utilities of individual criteria can be meaningfully combined into a single expected utility of a candidate alternative.

In order to satisfy these requirements, the following procedure can be followed.

Step 1: Specify a range of interest: For each criterion, specify a lower and upper limit of the range of interest.

Step 2:Identify the threshold. Since the range of interest specified in step1 includes both desirable and undesirable quantities of a criterion, it must also include a neutral contribution to success or failure. This neutral point is the threshold.

Step 3: Define utility scales.

Step 4: Develop Utility Functions. Two frequently used relationships are the straight line and exponential forms. Sample utility functional forms chosen for the sample problem are shown in Figure 3-13. A spreadsheet template is used to elicit user input, then the appropriate utility profiles are created.

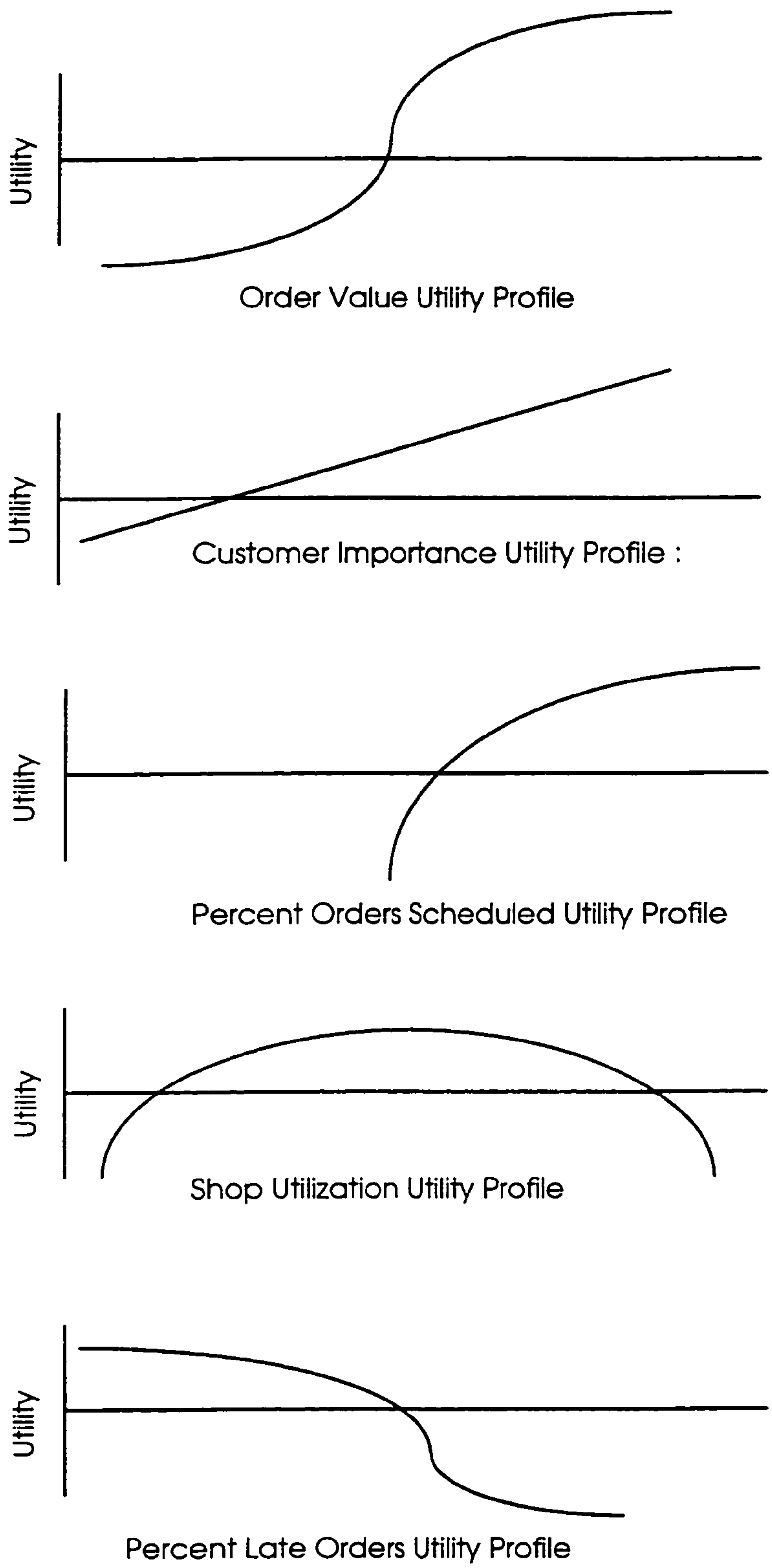


Figure 3-13 Utility Profiles for Sample Problem

Table 3-12 lists the range of values that were considered in estimating the sample utility function.

Table 3-12: Sample Values used for the Utility Function

Criterion Name	Lower Limit	Upper Limit	Scale	Utility Function
Order Value	5000	250000	Dollars	Exponential
Customer Importance	0	9	Ordinal	Straight Line
Percent Orders Completed	0	100	Percent	Exponential
Shop Utilization	0	100	Percent	Straight Line
Percent Orders Late	100	0	Percent	Exponential

3.8.5 The Objective Function

When a decision is made, information concerning outcomes and value systems is combined. To assure a consistent reviewable aggregation of this information into a relative score for each decision alternative, a mathematical expression or set of expressions, an objective function, is needed. To assure compatibility of the relative score of the value system used in making a decision, the objective function is based on the theoretically sound decision rule to maximize expected utility. Three different scheduling rules, Early Due Date, Order Priority, and Critical Ratio, were used for the sample problem. The results are as shown in Table 3-13.

Table 3-13: Performance of Different Scheduling Rules

Criterion	Scheduling Rule Chosen		
	Early Due date	**Order Priority**	**Critical ratio**
Order Value	175	140	280
Customer	150	135	265
% Orders Completed	60	60	100
Shop utilization	66	64	66
% Orders Late	All three methods give same, so can be omitted		
Total Utility	452	399	711

As can be seen from the above analysis, Critical ratio is the preferred scheduling rule followed by Due Date. This approach can be used to analyze different scheduling rules and their effectiveness. It may be worthwhile to store the historical performance values of the various scheduling rules that were employed along with shop loading parameters and other variables such as number of orders/operations considered for scheduling. This will permit the decision-makers to view historic schedule performance to gain further insight into the problem.

3.9 Computational Experience

Speed is of crucial importance in the successful implementation of any scheduling system in actual industrial settings. In a dynamic scheduling environment, frequent updating to keep the schedule realistic is inevitable. Response time is a key factor in the success of the scheduling system. The computational success of the scheduling system has been tested on a number of problems. The system runs on a PC with at least an Intel 486 chip with 16 MB of RAM (32 MB preferred) in an MS Windows environment. The problem of speed is becoming less restrictive with the availability of high-speed microprocessors over the 200 Mhz range.

3.10 Summary and Conclusions

In this chapter the scheduling heuristic that is part of the scheduling DSS has been described. A variety of features have been incorporated to handle complex real world scheduling problems. The system uses job-oriented heuristics, which have been shown to be computationally efficient for large-scale problems. Several features like operation splitting, stay-on-line, minimize moves, JIT, and minimizing setups have been incorporated into the DSS. These procedures will enable decision makers on the shop floor to manage problems involving due dates, rush jobs, customer inquiries and quotation of due dates. A utility theory based approach to evaluate schedule effectiveness has been presented. This approach will be beneficial in properly selecting the most appropriate scheduling rule. It is possible to store the historical performance of scheduling rules along with shop utilization so that refinements can be made in future selection of scheduling rules.

4. Design of the Decision Support System for Scheduling Construction Projects

4.1 Introduction

From the review of scheduling research, it is apparent that some dynamic, interactive methodology is required to find solutions for shop scheduling problems. A decision support system (DSS) seems to be the appropriate tool.

Decision-making is a common thread running through all managerial functions. It denotes an important category of managerial roles and is the essential mechanism that causes an organization's resources to be structured as they are at any given moment. Making decisions or, more properly, participating in the process by which they are made is an important and essential part of every manager's daily work. Decisions range from the profound to the trivial, from the complex to the very simple. There is general agreement in the management literature that a decision is a choice. Simon (1960), Costello and Zalkind (1963) indicate that it is a choice about a "*course of action.*" Fishburn (1964) has defined a decision as the choice of a "*strategy for action.*" Churchman (1968) indicates that a decision is a choice leading "*to a certain desired objective.*" These definitions suggest that we can think of decision making as an activity culminating in the selection of one from multiple alternative courses of action. It follows that a decision support system is a system that somehow assists in the selection.

Increasing competitive pressures have prompted manufacturing organizations to increase interest in decision support systems for industrial scheduling problems. This interest is particularly acute in dynamic situations, typical of a steel fabrication facility, where repeated scheduling of the production floor is required in response to events that change the capability of the production facilities, the demands placed on it, or the desired performance criteria. Significant performance improvements can accrue from making more informed scheduling decisions. A wide range of techniques from the disciplines of control theory, heuristic programming, operations research (OR), and artificial intelligence (AI) have been investigated as candidates for the construction of decision support tools. In this chapter, a DSS for the scheduling function is developed and described. From the application point of view, various scheduling problems with different degrees of importance, appear in almost every application area. However, although the application area is large and the need for support exists, the number of successfully applied OR results is not as large as it could be (McKay et al. 1988). Because the solution set is unbounded, and the set of criteria employed can shift, interaction with the decision-maker is required. Thus a DSS is a natural approach to the problem.

4.2 The Decision-Making Process

According to Simon (1960) the decision-making process involves three important phases, known as intelligence, design and choice. In any decision-making process, the decision-maker is concerned with solving problems in some sequence (Bonczek, Holsapple, and Whinston 1981). Any decision making process is governed by the decision-maker's strategy for reaching a choice (Janis and Mann 1977). A typical decision making process can be visualized as shown in Figure 4.1.

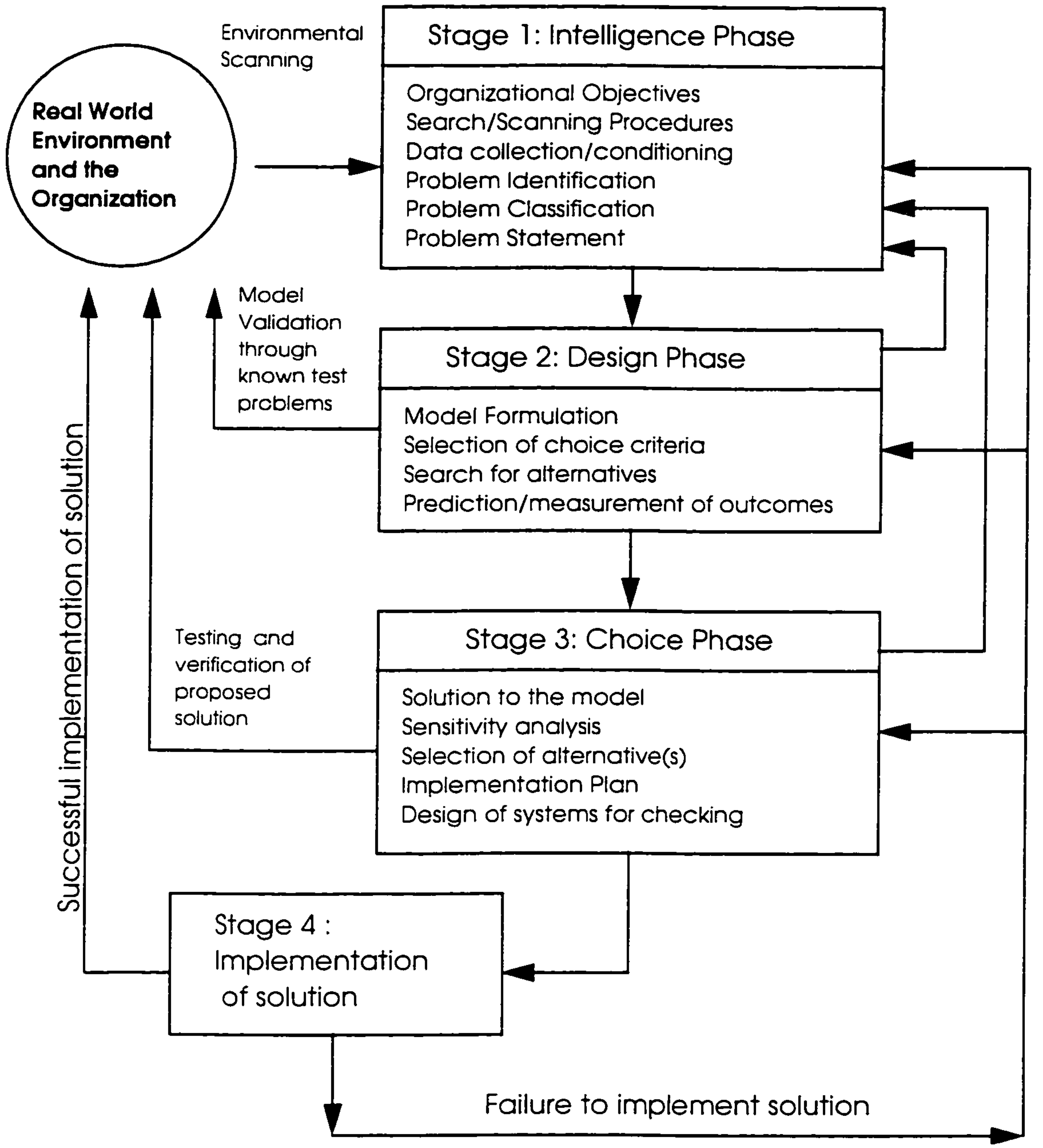


Figure 4-1: A Decision-Making Framework

The intelligence phase is a period wherein the decision-maker is alert for occasions to make decisions, pre-occupied with collecting knowledge and concerned with evaluating it in light of the organization's purpose. For example, a newly acquired piece of knowledge may suggest that an assembly line is not running as smoothly as it should, alerting the decision maker that a decision concerning corrective action will need to be made. Data gathering and conditioning for analysis is an integral part of the intelligence phase.

The design phase is a period wherein the decision-maker formulates alternative courses of action, analyzes those alternatives to arrive at expectations about the likely outcomes of choosing each, and evaluates those expectations with respect to the organization's purpose. During the design phase the decision-maker may find that additional knowledge is needed. This triggers the return to the intelligence phase to satisfy that need before continuing with the design activity. Continuing the example, the decision-maker formulates several alternative actions and thinks through the implications of each. The results of these analyses are assessed in light of the organization's ideal for assembly line performance. Evaluations of the alternatives are carried forward into the choice phase of the decision process, where they are compared and one is chosen. The design phase involves selecting the appropriate models needed for answering questions that the decision-maker is confronted with.

In the choice phase, the decision-maker exercises the authority to select an alternative. This is done in the face of internal and external pressures related to the nature of the decision-maker and the decision context. It is possible for none of the alternatives to be palatable, for several competing alternatives to yield very positive evaluations, or for the state of the world to have changed significantly since the alternatives were formulated and analyzed. Nevertheless, there comes a time when one alternative must be selected for implementation. If that point has not yet been reached, the decision-maker may return to one of the two earlier phases to collect more up-to-date knowledge, formulate new alternatives, re-analyze alternatives, re-evaluate them and so forth. Then, upon returning to the choice phase, some of the conflicts and pressures may have subsided, allowing for the selection to be more easily made.

Along with the above three phases, an implementation phase, as shown in Figure 4-1, is also considered as an integral part of the decision making process.

4.3 Overview of Production Planning

For most organizations, a production planning and control system usually involves a hierarchy of decision levels each concerned with a different time horizon (Gopalakrishnan, 1988). A typical system is shown in Figure 4-2 and is usually referred to as Manufacturing Resource Planning (MRP II) System. This system is comprised of aggregate, long term planning at the top, defining the Master Production Schedule (MPS), through medium term requirements analysis, probably using a variant of Material requirements Planning (MRP) to short term production activity control at the bottom. A typical steel fabrication shop will also have a similar system whereby management sets out broad guidelines at the MPS level (this could be the amount of steel fabricated and shipped on a monthly basis). The problem at the shop level in steel fabrication is more complex due to the fact that some orders may be 'one of a kind.' To support the decision making process, the modules in the hierarchy need access to a wide range of information, including: product design, bill of materials, process plans, machine and support resources availability and capability, and system demand in the form of forecasts and orders. Any decision support system designed for the purpose of scheduling should be capable of handling both the volume and diversity of information necessary for valid decisions to be made. Both relatively static information (products, components, process plans and resource capabilities) and dynamic information (current shop status, machine availability, current and future known orders) must be accessible and accurate. A large and appropriately structured database is needed to represent all this information to the level of detail required for finite capacity scheduling. The three essential components that this type of DSS for scheduling must possess, as identified by Pinedo (1995), are database management (for storing and accessing all the required information); model management - schedule generator, and user interface.

4.4 DSS for Scheduling

The construction of a model of a real life problem is always time consuming, but it is central to the entire decision making process and is therefore of the utmost importance. In particular, scheduling problems are hard to model since several types of variables are involved. The construction phase of a model for a production system comprises the identification of the characteristics of the system and the definition of all data relevant for the scheduling decisions.

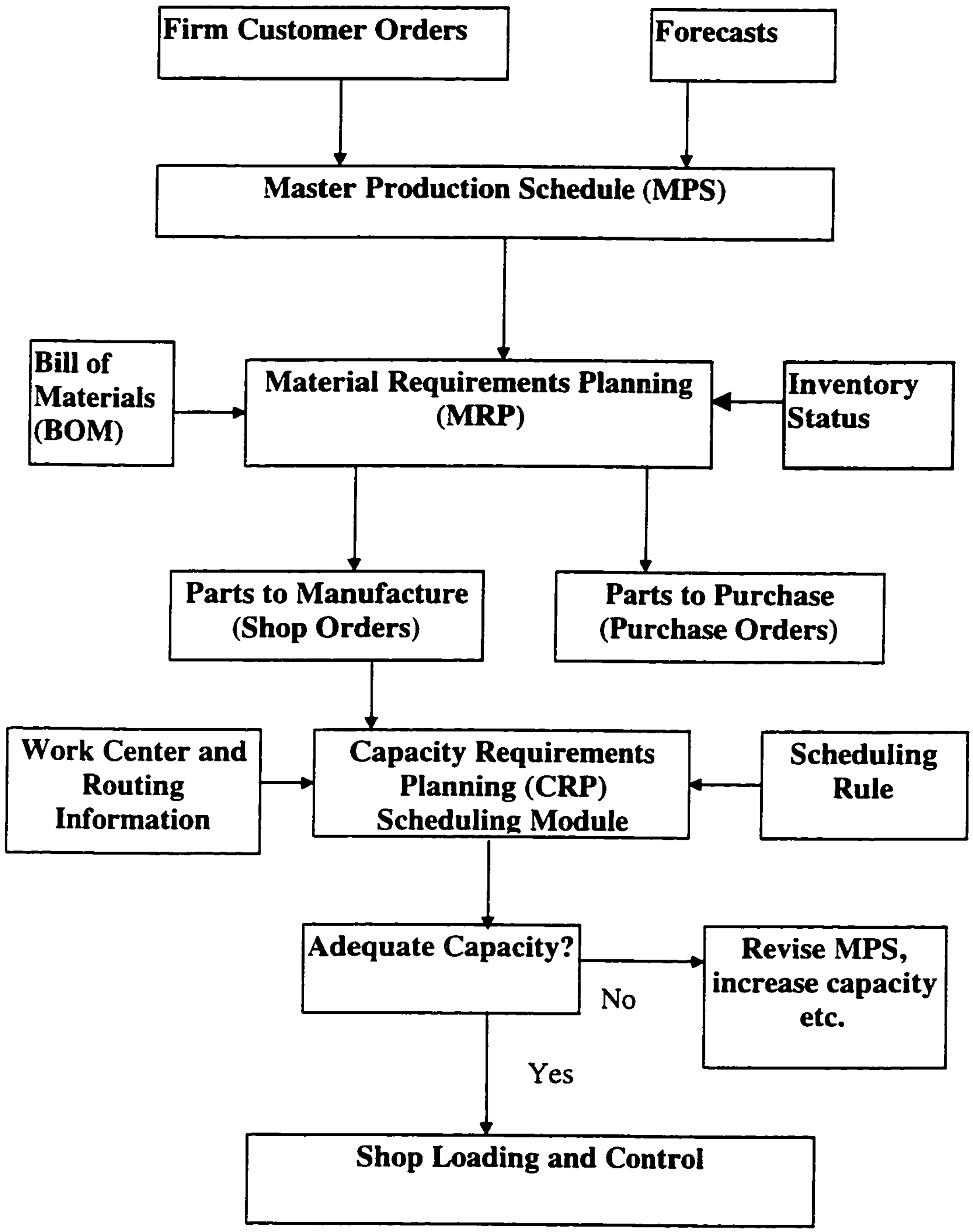


Figure 4-2: A Typical Manufacturing Resource Planning Methodology

4.4.1 The DSS Database

Data plays an important role in a DSS. Data is either accessed directly by the user or is an input to the models for processing. Care must be taken to ensure its availability and accuracy. There are several strategies for maintaining data. These include random access files, sequential files, indexed files, and so on. However, the availability of Relational database software like FoxPro, Access, and DBase has significantly reduced the amount of coding that is needed to store, retrieve and maintain information. The scheduling DSS utilizes such a relational database. A database is a collection of inter-related data organized in such a way that it corresponds to the needs and structure of an organization and can be used by more than one person for more than one application. Relational database management is a highly structured approach to knowledge representation. It is a technique for record keeping frequently used in developing transaction processing and MIS as well as DSS's. Generally, database management provides for the structured representation and flexible processing of data. Over the past decade, relational DBMS has emerged as the most widely used approach. The database is created, accessed and updated by a set of software programs called data base management system (DBMS). A DBMS performs three basic functions. It enables storage of data in the database, retrieval of information from the database, and control of the database.

The data for the DSS has been broken down into different classes and well-structured tables have been designed to collect and maintain the information. A table consists of records of a particular type. Thus, records about product could constitute one table, records about customer orders could constitute another table, and so forth. A collection of related tables is called a database. When using the relational approach, relationships among records in different tables are represented by field redundancy. Each table in a database consists of fields, which identify a category of data that can be contained in the table. A field can be numeric, character, date, memo or general type. Database management is concerned with the representation of knowledge in a database's tables and the processing of these tables for the ultimate decision support aim of extracting desired data from them.

The database for the DSS has been designed in such a way that all the relevant information required for the scheduling heuristic, management reports and shop floor control is easily available. The data can be easily entered manually into the tables for which specific forms have been designed or it can be directly imported from a variety of file formats (text, spreadsheet, dbfs, etc.).

The database contains a large number of tables. The most critical among them are briefly mentioned below.

- **Order Related Tables**

The shop must consider every issue to be an order for some 'item' to be processed. Hence, each shop issue should have an Order Number (see Figure 4-3), a part (or drawing) number to identify tasks to be done, due date, priority, class etc. This information will be maintained in the database as 'Order Related Tables', consisting of an 'Order Master Table', an 'Order Details Table', an 'Order Status Table', an 'Order Pegging Table', and an 'Order Resource Table.' The information from these tables is the starting point for scheduling. A detailed list of the fields of these tables is provided in Appendix-2.

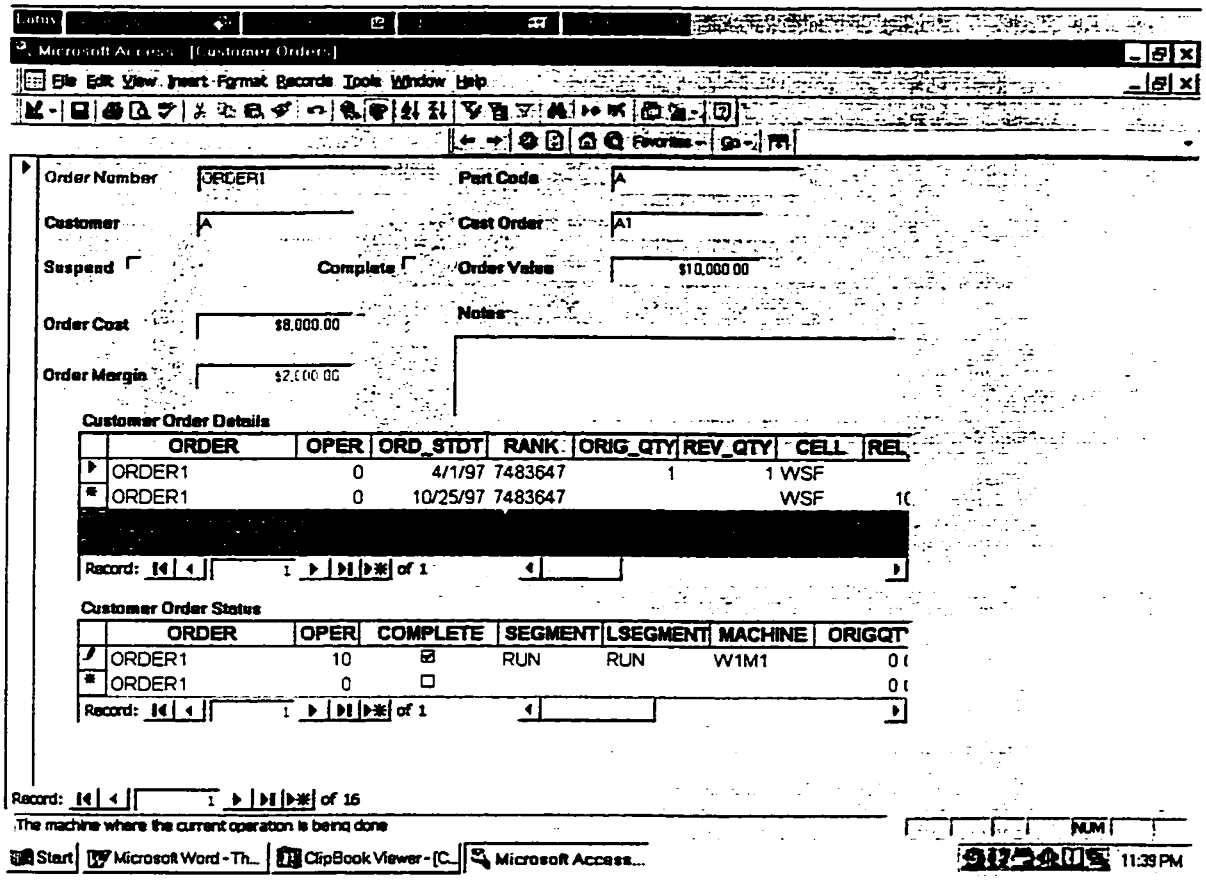


Figure 4-3: Order Related Information

- **Process Related Tables**

For every part (or drawing) number, there should be an unique process - a set of activities (operations like CUT, WELD, PAINT etc.) that must be completed in order to finish the part for shipment to the customer. Activities may be sequential or parallel. These operations can be numbered as 10, 20, 30 etc. to denote the sequence. Each of the operations in turn can be broken down into segments (like SETUP, RUN, CLEANUP etc.). It is also necessary to identify the machine (or shop facility) where each operation is to take place and the duration of each operation/segment. This information will be stored in the database as

'Process Related Tables' (Figure 4-4). Part of a sample of the process information for order 1700482 is shown in Table 4-1.

Table 4-1: Typical Process Information

Process	Operation	Machine	Segment	Duration
1700482	10	CUT1	SETUP	10
1700482	10	CUT1	RUN	30
1700482	20	GR1	RUN	15
1700482	30	FT1	RUN	45
1700482	40	WL1	SETUP	10
1700482	40	WL1	RUN	100
1700482	50	PT1	RUN	20

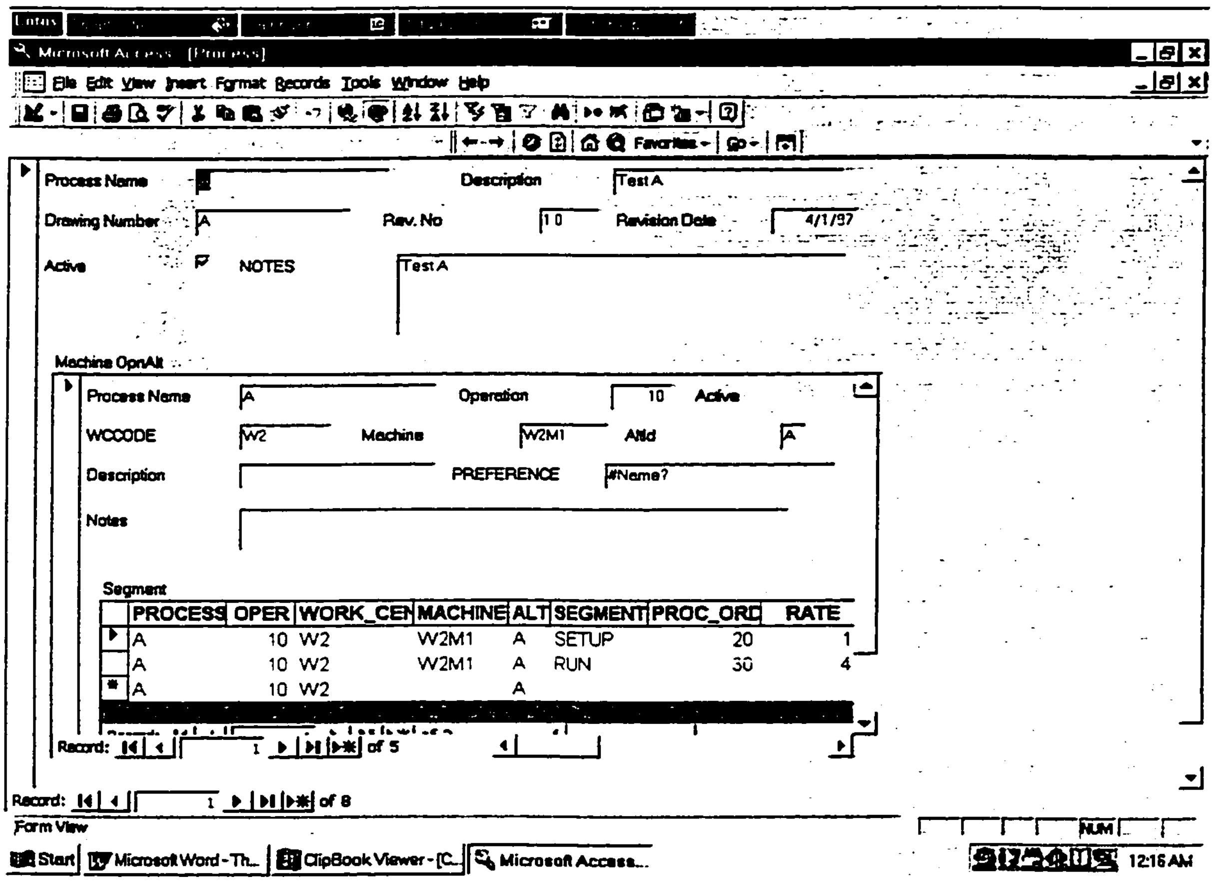


Figure 4-4: Sample Process Information

These are vital to the effectiveness of the DSS. They have been designed in such a way that almost any real-life shop floor can be modeled using the DSS. The parent-child relationship is as follows:

Part → Process → Operations → Machine →Segment →Resource.

This hierarchical design allows the user to specify any operation of a process to be performed at any number of alternate machines. Moreover, the segments and the resource specifications can be machine dependent. For example, Operation 10, if done on Machine A, could have SETUP, RUN, CLEANUP as three definite segments of certain duration. The same Operation 10, if done on Machine B, might have RUN as the only segment, so the duration for Machine B may well be different from that for Machine A.. If only the work center is specified, and the machine is not specified (default), the scheduler will schedule the job on a machine in the work center that will result in the operation being completed at the earliest time. This way, any process can be effectively modeled. A detailed list of the fields for these tables is provided in Appendix-2.

- **Work Center Related Tables**

The shop resources must be identified in terms of capacities available for use (usually hours of operation defined by a shift pattern) typically broken down into work centers and machines in each work center. There may be additional resources like Special Tools, Jigs and Fixtures, manpower, or materials that must be identified. Each resource can have a different availability pattern (some machines may be available only for 1 shift, others may be available for 2 shifts, or 3 shifts). All this information will be stored in the database as 'Work Center Related Tables.' For example, we can define Cutting, Grinding, Fitting, Welding and Painting as resources to be scheduled. Table 4-2 shows a sample definition of machines and Table 4-3 shows the shift pattern definition. It is possible to include outside resources (like subcontractor) in the resource definition tables. Outside resources will be assumed to have infinite capacity, as no knowledge of their shop loading may be available. In table 4-2, Painting is shown as an outside resource. The DSS only need to know the lead-time, say 1 day, to paint a given load of work orders.

Table 4-2: Sample Information of Machines

Work Center	Machine	Description	Internal	Shift Pattern Code	Other Information
CUT	CUT1	Cutter	Yes	PATTERN1	
GR	GR1	Grinder 1	Yes	PATTERN1	
FT	FT1	Fitter 1	Yes	PATTERN1	
FT	FT2	Fitter 2	Yes	PATTERN1	
WL	WL1	Welder 1	Yes	PATTERN1	
WL	WL2	Welder 2	Yes	PATTERN1	
PT	PT1	Painting	No	PATTERN1	

Table 4-3: Sample Definition of Shift Pattern

Shift Code	Day	Time From	Time To	QUANTITY
PATTERN1	MON	06:30	08:30	1
PATTERN1	MON	08:45	11:30	1
PATTERN1	MON	12:00	14:30	1
PATTERN1	TUE	06:30	08:30	1
PATTERN1	TUE	08:45	11:30	1

The shift code specifies the pattern of availability. For example, PATTERN1 denotes working times from 06:30 AM to 08:30 AM; 08:45 AM to 11:30 AM; 12:00 Noon to 02:30 PM on Monday, Tuesday etc. (assumed breaks during 08:30 – 08:45 and between 11:30 and 12:00). It is therefore possible to define a different work pattern for each day of the week, and for each resource.

A detailed list of the fields for these tables is provided in Appendix-2.

4.4.2 The DSS User Interface

Unlike the black box principle normally used to apply OR techniques, the DSS approach leans heavily on coherency between a system and its user. The dialog component of a DSS consists of software and hardware that provide the user interface for DSS. The term 'user interface' covers all aspects of the communications between a user and the DSS. It includes not only the hardware and software, but also the factors that deal with ease of use, accessibility and human-machine interaction. Some DSS experts feel that user interface is the most important component because much of the power, flexibility and ease of use characteristics of the DSS are derived from this component. An inconvenient user interface is one of the major reasons why managers have not used computers and quantitative analyses to the extent that these technologies have been available.

Appreciation for the importance of the dialog component is gained by recognizing that, from the user's perspective, the dialogue is the system. What the user has to know in order to use the system (the options for directing the system's actions and the alternative presentations of the system's responses) are what is important. The user typically has little interest in such considerations as the hardware and software used, how data are stored in memory, and the algorithms employed by the models. Such factors are often invisible to the user. When designing the DSS dialog, it is essential to identify the potential users. In some instances, there is a single user. More typically, the DSS has multiple users.

The Scheduling DSS has been developed to operate in the Windows environment. The dialog system is similar to the Windows menu-driven approach and also has tool bars that can be used with a mouse by clicking on them. This menu-oriented approach of using a mouse to access pull-down menus and tool bars offers both simplicity of use and flexibility; it is considered the dominant 'standard' for end-user computing. In the DSS, different menu choices can be made available depending on the particular user (The MENUS Table has been designed specifically for this purpose). Some users can only browse information. For such users, the options to Add, Edit, and Delete will not be made available.

4.4.3 Requirements considered in the design of the Scheduling DSS

The scheduling DSS has been designed to meet the requirements typically placed on expert systems and shells. These are:

1. A modular design must be used.
2. A hierarchy of schedules that corresponds to different levels of abstraction should be used.
3. A constraint monitor should be used.
4. Some deduction mechanism that can answer common scheduling queries, as well as that employed for deducing all other information, should be available to the user.
5. What-if scenarios should be allowed.
6. Provisions must be made for trade-offs between time versus completion of the schedule.
7. The ability to trace back from implementation item to requirements and vice versa should be provided.
8. A system should be provided to reduce the time needed to learn how to use the system.
9. A good model of what jobs cause what constraint violation should exist.
10. Comparison of several alternative schedules should be allowed.

11. Rescheduling must be easily handled.
12. Constraints must be modeled separately from job-resource interactions so that non-technical, 'political' constraints can be represented.
13. Similar jobs, resources, and constraints must each be group-able.
14. A user should be able to specify priorities.
15. Precedence relations between jobs must be handled both qualitatively and quantitatively.
16. Resources must be represented in a fashion that allows unique association with jobs, pooling of resources, or partial pooling of resources.
17. Resources should be allowed to be created, consumed, and refurbished as required by the various jobs in the schedule.
18. Constraints must be represented in a manner that allows them to be associated with particular jobs and resources.
19. Resource definition should be made possible at the order level, work-center level, machine level, or operational level.
20. The system must be able to represent heuristics that can intelligently guide the use of algorithmic scheduling techniques and short-circuit expensive searches.
21. The system must be able to combine the techniques of job scheduling, activity scheduling, and enumeration of alternatives so that the most appropriate technique can be used at any time in the schedule formation process.
22. The tool should support menu, mouse, and other-user friendly input facilities.
23. A user 'override' feature should be supported.
24. Some 'automatic' rescheduling capability should be provided.

The DSS has been designed in such a way that the user does not have to be an expert in scheduling rules and techniques. The system is open-ended in the sense that, specific codes (queries for example), are actually built from scratch every time the user specifies a parameter. This design approach permits the knowledge base to be expanded without limit, and the need for re-writing program codes to handle additional requests is minimized. The Structured Query Language (SQL) statement shown in Figure 4-5 is generated when the user selects the critical ratio as a schedule parameter.

```
SELECT [Customer Orders].ORDER, [Customer Order Details].REV_QTY,
[Customer Order Details].DUE_DATE, [Customer Order Details].PRIORITY,
[Customer Order Details].CLASS, Process.PROCESS, [Machine
OpnAlt].OPER, [Machine OpnAlt].WORK_CENTER, [Machine
OpnAlt].MACHINE, Segment.SEGMENT, Segment.PROC_ORDER,
Segment.RATE, IIf([SEGMENT]="SETUP",[RATE],[REV_QTY]*[RATE])
AS Gop1
        FROM ((Process INNER JOIN [Machine OpnAlt] ON
Process.PROCESS = [Machine OpnAlt].PROCESS) INNER JOIN Segment
ON ([Machine OpnAlt].ALTID = Segment.ALTID) AND ([Machine
OpnAlt].MACHINE = Segment.MACHINE) AND ([Machine
OpnAlt].WORK_CENTER = Segment.WORK_CENTER) AND ([Machine
OpnAlt].OPER = Segment.OPER) AND ([Machine OpnAlt].PROCESS =
Segment.PROCESS)) INNER JOIN (([Customer Orders] INNER JOIN Parts
ON [Customer Orders].PART = Parts.PART) INNER JOIN [Customer Order
Details] ON [Customer Orders].ORDER = [Customer Order Details].ORDER)
ON Process.PROCESS = Parts.PROCESS
        GROUP BY [Customer Orders].ORDER, [Customer Order
Details].REV_QTY, [Customer Order Details].DUE_DATE, [Customer Order
Details].PRIORITY, [Customer Order Details].CLASS, Process.PROCESS,
[Machine OpnAlt].OPER, [Machine OpnAlt].WORK_CENTER, [Machine
OpnAlt].MACHINE, Segment.SEGMENT, Segment.PROC_ORDER,
```

Figure 4-5: SQL Statement Generated from User Input

More complex SQL statements are generated in a similar manner when the user includes a number of scheduling parameters as shown in Figure 4-6.

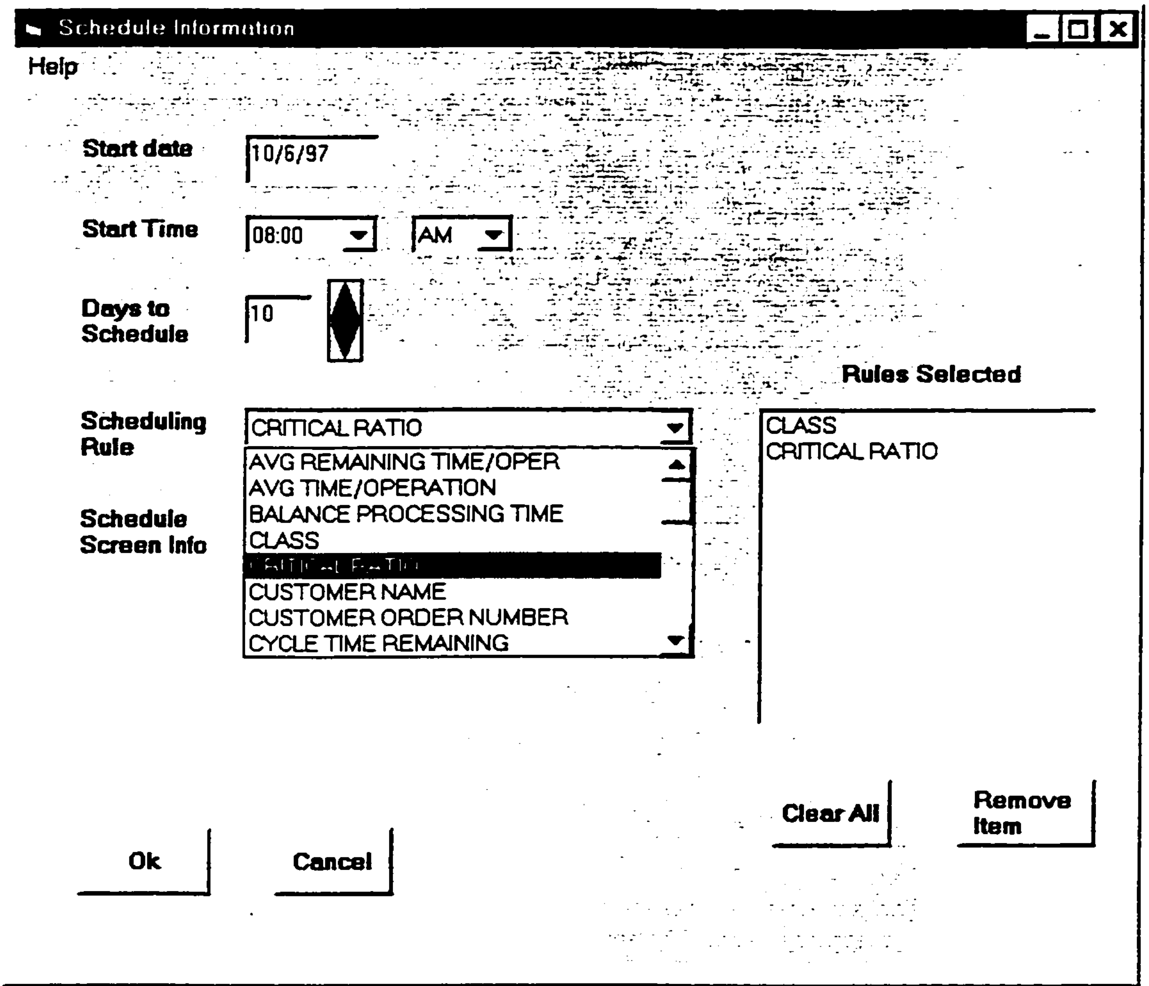


Figure 4-6: User Input for Scheduling

4.4.4 Scheduling Criteria Built in the Model Base of the DSS

The model base of the DSS contains a number of scheduling criteria to which others can easily be added. These are described below.

1. **AVERAGE REMAINING TIME/OPERATION**

   The remaining processing time for each work order divided by the number of remaining operations is a measure of the average remaining time per operation.

$$TAVG_i = \sum_{j=m}^{n} \frac{p_{ij}}{(n-m+1)}$$

Where $TAVG_i$ is average time remaining for order i, p_{ij} is the processing time for order i at stage j, m is the current operating stage, and n is the total number of operating stages.

If this function is included in the definition of the scheduling rule, orders will be scheduled in the order of the remaining processing time per operation. Orders with the smallest value will be scheduled before orders with a higher value if the parameter is tagged as being ascending, and vice versa.

2. AVERAGE TIME/OPERATION

The average time per operation is calculated by dividing total processing time for all operations by the total number of operations as defined in the process for an order. The resulting number is a measure of what is called the 'average time per operation.'

$$TAV_i = \sum_{j=1}^{n} \frac{p_{ij}}{n}$$

Where, TAV_i is average time for order i, p_{ij} is the processing time for order i at stage j, and n is the total number of operating stages.

If this function is included in the definition of the scheduling rule, orders will be scheduled in the order of average time per operation. Orders with the smallest value will be scheduled before orders with a higher value if the parameter is tagged as being ascending, and vice versa.

3. BALANCE PROCESSING TIME

Balance or remaining processing time is defined as the processing time remaining for operations that are in process or that have not yet been started. The processing time calculation is based on the active operations only and the time for first valid machine alternative.

$$BAV_i = \sum_{j=m}^{n} p_{ij}$$

Where BAV_i is the time remaining for order i, p_{ij} is the processing time for order i at stage j, m is the current operating stage, and n is the total number of operating stages.

If this parameter is used in the formulation of a scheduling rule as ascending, orders with the least balance processing time will be scheduled before orders with a higher value.

This is sometimes called the SPT (shortest processing time) rule. Use this rule to minimize WIP and maximize throughput.

4. CLASS

This parameter is assigned to each order attribute.

It is a three character code that is generally used to classify the various types of orders. For example, one can define FRM as a code for firm orders, PLN for planned orders, STK for in house stock orders.

If the firm orders are to take priority over the planned and stock orders, set the logic for this parameter to ascending.

5. **CRITICAL RATIO**

Critical Ratio is defined as the ratio of time available to process an order and the processing time for the order. The model base calculates the critical ratio for each order using the status of the order as of the base date and time.

The formula to critical ratio is as follows:

$$CR_i = \frac{(d_i - t)}{\sum_{j=m}^{n}(p_{ij} + q_{ij})}$$

Where, CR_i is the critical ratio for order i, d_i is the due date for order i, t is the current date, p_{ij} is the processing time for order i at stage j, q_{ij} is the waiting time or queue time for order i at stage j, m is the current operating stage, and n is the total number of operating stages.

The numerator is the actual time left between 'Current Date' and the due date of the order. The denominator is the processing time for the remaining active operations starting with the first incomplete operation.

6. **CUSTOMER NAME**

This is a free format parameter that can be used to define a scheduling rule that ensures that certain customers get priority over others.

7. **CUSTOMER ORDER NUMBER**

This is a free format function that can be used to ensure that all orders for a customer are processed in the order of the customer's order number. This is a useful function when there is more than one order for a customer.

8. **DAYS OVERDUE**

The number of days overdue is computed after a schedule is generated. The formula for days overdue is:

Days Overdue = Scheduled Finish Date - Due Date

This parameter is used to sort the scheduled orders according to the number of days overdue, thus enabling the user to focus on the really hot orders quickly. As the value of this function is only known after a schedule is generated, it should not be used in defining a scheduling rule.

9. **DUE DATE**

This parameter is a user defined value for each operation of the order.

If this function is used in defining a scheduling rule, the orders with earlier due dates will be given a higher priority when the parameter logic is ascending, and vice versa.

10. MOVED OPERATIONS

Moved operations are defined as operations that have been moved by the scheduler to a machine other than the one on which the heuristic initially scheduled the operation.

If this parameter is used in defining a scheduling rule, operations that have been moved will remain on the machine and the time they were moved to. Moved operations are scheduled first if this function is the first parameter of the scheduling rule.

11. NUMBER OF OPERATIONS REMAINING

The number of operations or steps in the manufacturing process to be completed is called 'Number of Operations Remaining.'

If this parameter is included in the definition of the scheduling rule, orders that have the least number of operations in the process (i.e. the total number of active operations minus number of operations that are complete) will be scheduled first.

12. NUMBER OF OPERATIONS

This is defined as the number of operations or steps in the manufacturing process.

If this parameter is included in the definition of a scheduling rule, orders that have the least number of operations in the process will be scheduled first.

This function is generally used when the objective of scheduling is to maximize turnover of orders or maximize throughput.

13. **ORDER NUMBER**

This parameter is useful in the definition of a scheduling rule when there is some logic in the order number that requires orders with lower order numbers to be scheduled first (if the function is defined as ascending).

14. **ORDER VALUE**

This parameter is another user defined function where the order value can be the sales value, the profit margin, cost, etc.

If this function is used in the definition of the scheduling rule, orders with the highest profit margin will be scheduled first (if the parameter logic is descending).

15. **PART NUMBER**

This parameter is useful in the definition of a scheduling rule when there is some logic in the part numbering scheme used. For example, if the part numbering scheme assigns smaller numbers to components and larger part numbers to assemblies.

16. **PRIORITY**

This parameter is similar to the class code, and is assigned to each order attrbute. It is a one character code that is generally used to assign priorities to the various types of orders.

For example, you could define 5 as a priority for all firm orders, and 9 for all planned orders. In this example, the firm orders take precedence over planned orders.

17. **RELEASE DATE – FIFO**

FIFO is an acronym for First In First Out.

If this parameter is included in the definition of a scheduling rule, orders with the earliest release date will be scheduled first when the logic is ascending. If, however, the logic is descending, the scheduling rule would become LIFO or last in first out.

18. **SLACK**

Slack is defined as the difference between the base date and the due date minus the balance cycle time for the order.

$$S_i = d_i - t - \sum_{j=m}^{n} p_{ij}$$

Where S_i is the slack for order i, d_i is the due date for order i, t is the current date, and p_{ij} is the processing time at stage j for order i, m is the current operaing stage, and n is the number of operating stages.

If this parameter is used in the definition of a scheduling rule, and the logic is ascending, orders with the least slack will be scheduled first.

19. **TOTAL PROCESSING TIME**

Processing time is defined as the time required to process all operations for an order. Processing time is based on all the active operations and the time for first valid machine alternative.

$$TP_i = \sum_{j=m}^{n} p_{ij}$$

Where TP_i is the total processing time for order i, m is the current stage of operation, p_{ij} is the processing time for order i, at stage j, and n is the number of operating stages.

If this parameter is used in the definition of a scheduling rule, and the logic is ascending, orders with the least processing time will be scheduled before orders with a higher value.

20. **UPDATED ORDER**

An updated order/operation is defined as one where production has been reported. An updated order is generally what a machine is currently producing. When this parameter is used as the first parameter in the definition of a scheduling rule, the schedule generated will ensure that any order being worked on automatically moves to the front of the queue.

21. **USER DEFINED FUNCTIONS**

The model base offers six user defined functions. These can have any values and can be combined in any manner to create customized scheduling rules.

The user can use any of the above parameters (or any combination of these) and the heuristic will schedule jobs based on them. The user, thereby increasing the scope of the scheduling knowledge base can easily add additional criteria to the list (as an entry to the rules table and the appropriate SQL statement can be generated)

Typical outputs of the scheduling run are the schedules generated and sorted by customer order, operations sorted by machines, and shop loading sorted by operations.

4.4.5 Features of the Scheduling DSS

The scheduling DSS has been designed to deal with a wide range of conditions and situations. It can:

- Permit definition of multiple-shift patterns.
- Eliminate 'setup' automatically if succeeding part is the same as the previous part.
- Respect user defined gaps between segments.
- Permit user 'over ride' feature enabling movement of operations from one machine (and time) to another.
- Handle internal as well as external machine capabilities.
- Handle sequence dependent changeovers.
- Handle 'time-sharing' machines like furnaces, and shot blasting machines.

- Permit a number of scheduling rules and criteria to be chosen.
- Modify and fine-tune existing schedules, and fit new orders.
- Support but not replace the decision-maker.
- Emphasize ease of use, user friendliness, user control, flexibility and adaptability.
- Contain heuristics for scheduling, ranking jobs and prioritization schemes

4.5 Conclusion

In this chapter the design of the Scheduling DSS was presented using the well-accepted conceptual DSS framework. The strength of this DSS for scheduling is its ability to model a large variety of production systems and use a variety of scheduling methods with ease of data handling and seamless interface to various modules.

5. Case Study: Implementation of Scheduling DSS in a Steel Fabrication Shop

5.1 Introduction

The scheduling DSS described in this thesis will be implemented in stages at a local steel fabricator, STEELR'US, Ltd. STEELR'US has one of the largest steel fabrication facilities in western Canada. In the past few years, their business has grown considerably; they have recognized the need to have a proper planning tool that will enable them to manage the busy shop in a proactive, rather than reactive manner. Based on interviews with the management, it was evident that they realized that the lack of a shop floor control system could lead to the many productivity problems that are associated with scheduling shop activities. These include not having the right item when it is needed, not having the right piece of equipment when it is needed, large WIP, using excess inventory to hide problems, and inflexibility and lack of responsiveness to design changes and changing customer requirements. Since project managers/estimators were constantly reacting to unplanned activities, they recognized the need for implementing sound management practices and decision support models. It is clear that the resolution of these difficulties could translate into considerable direct and indirect savings and enable STEELR'US to become more competitive in the global market place.

Perhaps the most difficult production problem in any complex manufacturing organization is scheduling. With the growth in business, STEELR'US found a need to implement a project management system. STEELR'US's activities include pre-fabrication, fabrication, and post-fabrication stages, with some fabricated material being shipped to vendors for application of protective coating. A preliminary study focused on preparing and investigating a scheduling application through the use of Primavera (P3) project management software. The main limitations found in the preliminary investigation results were:

1. Failure to allocate company resources while taking into account factors such as contractual constraints, fabrication practicalities, the detailing schedule, and company wide priorities. P3 is not able to handle multiple input.

2. Failure to map and interpret an event driven sequential process to a functional overlap at a higher abstraction level. P3 cannot interpret individual steel item progression through each of the company's departments. A process view must be modeled.

3. Failure to integrate P3 into other company information structures. Considering the company's existing system, this is a very difficult task. Hence, the impact of changes, revisions, and re-prioritizing of projects cannot be adequately reflected in the schedule.

STEELR'US has unique challenges that cannot be met by existing software. First, software capable of handling all of STEELR'US's requirements is not readily available as an integrated package. Second, even if different sub-modules are purchased from different vendors, there will be severe learning curve effects affecting productivity, and there will still be a definite need to customize; many modifications would be necessary for integrating the different pieces before it could become a worthwhile investment.

5.2 Special Requirements of Steel Fabrication Shops

Steel fabrication shops follow typical boom and bust cycles of the economy. Without a proper shop floor management system, it is very difficult to evaluate the impact of delay of one order on the performance of the organization. Moreover, fabricators must accommodate requests from customers for design changes, or shipment needed earlier than anticipated. Since fabrication jobs are typically of a longer duration (as opposed to manufacturing, where large quantities of the same item is often produced) and are non-repetitive in nature, a method for prioritizing jobs based on customer importance or re-assigned due dates can be very beneficial to the organization. In this environment, one of the often-raised questions is, "what is the optimal decision rule to use in scheduling?" That is, given a backlog of orders, which scheduling rule will give the best overall results? "Which measure is more important?" is, in itself, an interesting question, viz., "Is it better to satisfy the greatest number of customers or the ones with the largest orders?"

Since it answers these kinds of questions, the Scheduling DSS developed in this thesis was considered to be a prime candidate for implementation at STEELR'US. Initially, four main challenges were identified. They were:

1. Identification and collection of data on a timely basis and with the least possible effort. In the steel fabrication industry, the lack of shop floor data is a major issue that must be addressed. This presents considerable challenges and at the same time opportunities for incorporating state-of-the-art technologies like bar coding and electronic time-clocks for use in shop data collection, conditioning, and processing.
2. Representing data in different systems at different levels to suit the required view (e.g. for detailing purposes, data is tracked by pieces. For fabrication it is tracked by tonnage and type of piece, and for erection, by piece and physical property).

3. Resource allocation (optimization of resource allocation) especially on the shop floor to ensure achievement of required shipping milestones while at same time ensuring that the shop is continually loaded.

4 Quick and easy preparation of feasible shop schedules and their interface with drafting scheduling and construction scheduling.

5.3 Current Practices at STEELR'US

STEELR'US is primarily a large steel fabrication shop with a material storage yard. It undertakes both small (1 or 2 tons) and large (over 1000 tons) jobs. STEELR'US consists of many departments that work together to complete steel fabrication and erection projects. The processing characteristics are non-repetitive; mixed-batch type operations, as opposed to those found in flow shops or process industries. Typical operations include cutting (beam line/burn table), detailing, fitting, welding, inspection, surface conditioning and finishing, painting and shipping. The shop floor can be functionally divided into six main areas:

1. Cutting area – where beams and columns are cut to required lengths.
2. Detailing area – where the cut components are fitted together and tack welded.
3. Welding area – where the majority of the work is done.
4. Drilling, grinding, boring, and milling area.
5. Surface finishing area.
6. A painting/shipping area.

These six areas span 2 bays or shops.

The business cycle can be stated briefly as follows: STEELR'US prepares detailed estimates for the projects for which they bid. The project managers, who are also the estimators are initially involved in the estimation phase where customer specifications are analyzed and priced. Purchasing personnel is also involved at this phase by providing material pricing and availability information to the estimators. After the successful award of a tender, the project manager passes the project information to other departments.

Large projects are typically broken down in a hierarchical fashion from Project into Jobs, from Jobs into Divisions, from Divisions into Major Assemblies, from Major Assemblies into Pieces as shown in Figure 5-1.

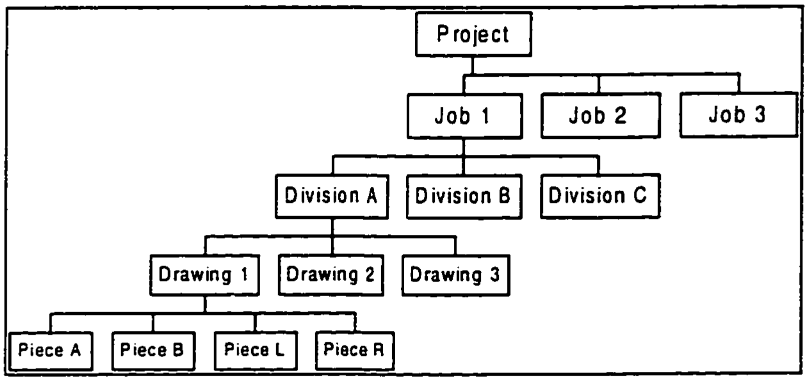


Figure 5-1: Typical Project breakdown

When the drafting department receives the client drawings, a preliminary material takeoff is done and is passed to purchasing so that steel can be reserved or ordered if required. Drafters will then proceed to produce detailed fabrication and erection drawings, if required. Depending on the requirements of the project, the engineering department may also get involved in the design of connection and lifting studies. As fabrication drawings are completed, they are sent for client approval. Once they are approved, drafting releases the drawings approved for construction to purchasing and the shop superintendent. The shop superintendent splits the list of drawings into 'shop load lists' based on his experience and delivery schedule information. A given "load list" consists of a collection of drawings that should be fabricated and shipped together. This decision depends on many factors including customer requirements, site logistics, shipping weight and physical restrictions. Purchasing issues material to the shop based on these "load lists". This sometime results in the same material being handled twice during the same day. A look ahead feature will be useful to avoid such double handling; the scheduling DSS will be able to provide this.

During production, the quality assurance (QA) department tracks heat number from fabricated material which can be referenced to the mill test reports so that turn-over packages can be easily prepared if requested by either the client or engineering. QA is also responsible for the inspection and testing of the various fabrication processes.

Currently, STEELR'US has no formal shop floor control system. The shop superintendent and lead shop hands have considerable experience in the production activities; they load the shop from previous historical performance bench marks, such as 100 Tons of shipment a day. This sometimes results in some work centers being over-loaded while others are under utilized. This is mainly due to the fact that there is no one-to-one relationship between weight of a piece and the amount of fabrication work involved. It is also not uncommon for the shop to stop the work currently in progress and start another job when a project manager intervenes to expedite his project. The effect of such actions on the overall system performance is not easily understood. This also causes load surges through the plant with some work centers being fully loaded while others wait for work.

When STEELR'US prepares bids, the project had to be estimated in terms of fitting, welding and painting man hours. However, this information is not maintained in a computer database so that it can be utilized at the shop floor level to plan production. This results in time consuming manual comparison of estimated versus actual hours for progress reporting purposes. Moreover, the shop status is updated purely from an accounting perspective for payroll purposes once a week. Production employees such as welders and fitters use time cards to record hours spent on the various jobs. The accounting department also tracks hours spent by drafters and engineers on a given job. Other costs that are assigned to jobs include invoicing from material suppliers and field bills. When production is complete, the shipping department coordinates the loading and shipping of the steel components, usually according to the initial load lists set up by the shop superintendent. Since the shop scheduling is presently done manually, this is not a problem. However, a computerized system cannot perform effectively unless daily updates are maintained. This presents the challenge of gathering all the relevant shop floor data in a timely manner. Bar coding of shop drawings and actual pieces will help in this regard and will be tested. The flow can be represented as shown in Figure 5-2.

Other types of inter-departmental communication also exist. Examples include the weekly production meetings held in order to discuss the status of the various projects and a project manager walking across the shop floor in order to gain an idea on the progress of his/her work.

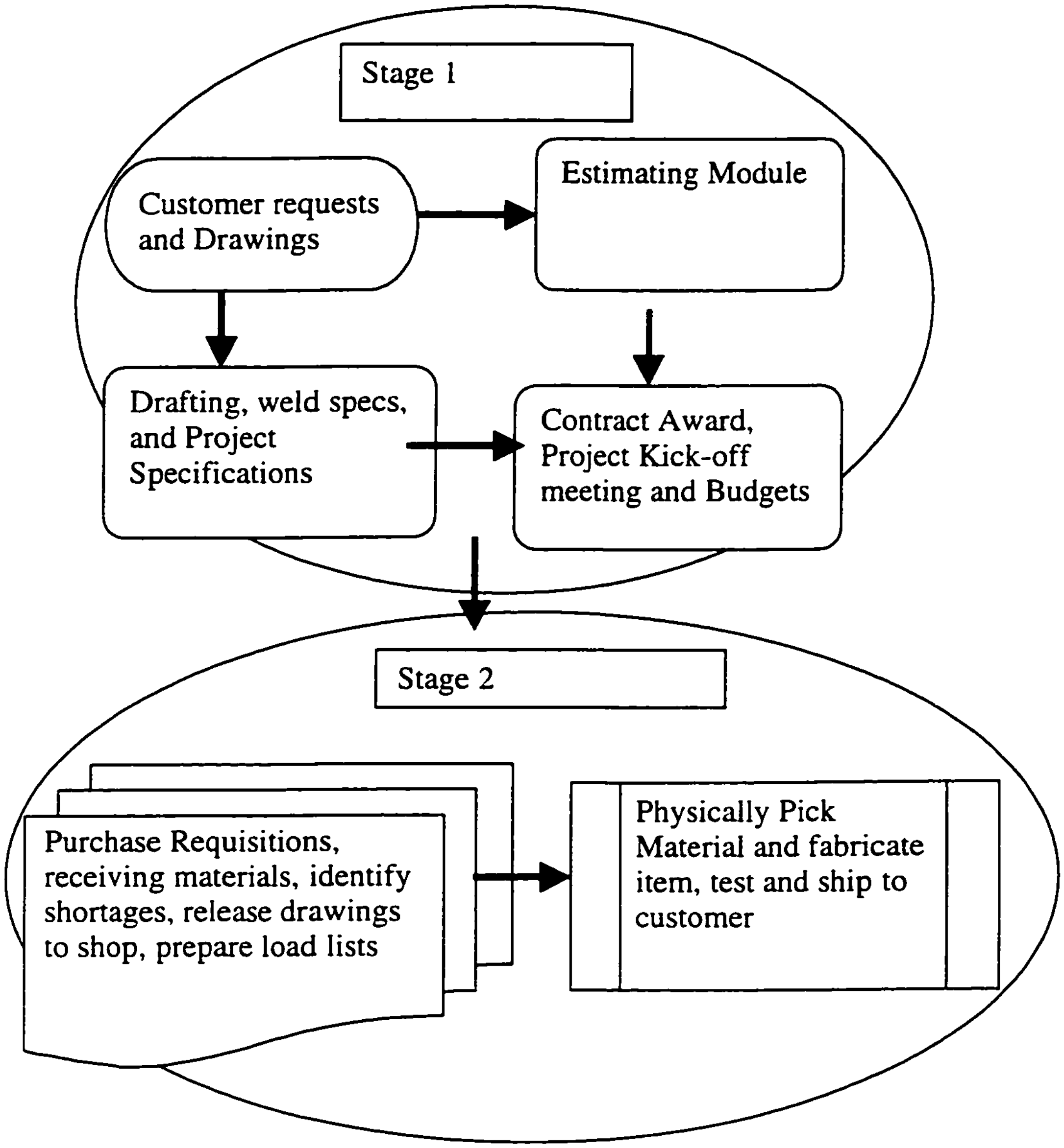


Figure 5-2 Overview of Current Process Flow

Planning at the macro level was to be undertaken through P3, and the scheduling DSS role was identified to be a shop floor control system. The purpose of the shop floor control subsystem is to release orders to the shop floor and manage the orders as they progress through the shop. The shop floor control system will help management correct all the day-to-day things that might go wrong, like machine breakdowns, loss of material etc. When these unplanned complications arise, decisions must be made regarding what to do next. Good decision making requires input-output control and information on job priorities from the shop floor control system. The scheduling DSS will be useful in addressing these issues and will help shop planners considerably by providing them with timely shop status and scheduling information.

Another limitation that is present in the construction industry is the limited availability of 'processes' and process information. Since jobs are of a non-repetitive type (and even though jobs typically progress from cutting, to fitting, to welding, then to painting), proper documentation of each and every process to be followed in the shop is not maintained in a file for use. However, a computer system needs this information in order to perform effectively. Generic templates have been designed for this purpose, they will contain the basic process flows, viz., cutting, detailing, welding, testing, painting and shipping. A box has been added to the generic list to identify special processes. These process templates are to be included in the drawings and will be filled in before release to the shop. Initially, the estimated time for operations will be 'rule of thumb' values, then as more shop floor information is gathered and maintained, these estimates will be continuously updated and revised for greater accuracy.

5.4 The Scheduling DSS: A Sample Session

The primary objective of the shop scheduling project is to assist STEELR'US to properly plan, execute and maintain/update a shop schedule based on a finite capacity planning methodology. The scheduling DSS is a PC-Network based management information and control system designed specifically for manufacturers who produce partially or totally to custom order. It is designed to meet the needs of the construction industry, which can be termed as custom or job shop manufacturing. This make-to-order manufacturing environment has special functional needs that traditionally have not been met by standard MRP II packages. These needs include manufacturing without utilizing part numbers, adding to or changing a job after it has been released to the shop floor, and using sub-contrators depending on the shop loading conditions. A complement of application modules comprises the base DSS to which users can add additional modules (decision rules) to fulfill their specific needs. Major functional areas include customer order entry, engineering definition (routing and bill of materials), production planning, scheduling, shop floor control, shipping, quality control, and job cost information gathering. Additional modules such as cost estimating and quotations, bar coding, and turn-over packages for customers can be easily added.

5.4.1 Capturing Management's Preferences

As previously mentioned, during the initial implementation planning of the scheduling DSS the question, "what is the optimal decision rule to use in scheduling?" was raised That is, given a backlog of orders, which scheduling rule will give the best overall results? Since previous published research usually employed over-simplified situations, the DSS design included models that could be used to establish the effectiveness of various decision rules in attaining the objectives of the organization. During interviews, managers were asked to list the major objectives that an ideal scheduling system should achieve. These objectives were then used as the performance measures against which the decision rules were compared. The objectives initially considered were: order value (with a view of minimizing work-in-process), customer importance, and shop utilization (in order to balance plant-wide work load – work tended to surge through the plant which caused peak utilization at some equipment and very low utilization at others. It is desirable to smooth out the workload to achieve on-time delivery performance). In order to rank the best decision rule, management was asked to prioritize these objectives. When they attempted to do this, they realized that the rules were dynamic in nature and were a function of economic conditions, new projects awarded, and current plant capacity levels. "Which measure is more important?" is, in itself, an interesting question, viz., "Is it better to satisfy the greatest number of customers or the ones with the largest orders?" To help STEELR'US cope with this problem, the model base in the DSS includes a utility theory based approach to rank different scheduling rules and suggest the most appropriate one for the current scheduling scenario. After much discussion with the managers, the criterion shown in Table 5-1 were chosen for trial runs.

Table 5-1: Management Criteria for Schedule Evaluation

Criterion Name	Criterion Scale	Weight Assigned
Order Value	Dollars	350
Customer Importance	0-9 Scale	150
Percent Orders Completed	Percent	150
Shop Utilization	Percent	100
Percent Orders Late	Percent	250
	Total Weight	1000

The criteria chosen can be expanded to include other management goals and the proper weights assigned. This approach of including all of the managements' objectives in the decision process was considered to be a very important feature of the scheduling DSS. While schedule generation cannot be guaranteed to be optimal, it provides management with the ability to evaluate different 'what-if' scenarios, and the model base is open-ended and can be augmented to include dynamic switching of rules by incorporating 'do this unless' type of rules. The overall scheduling environment can be visualized as shown in Figure 5-3.

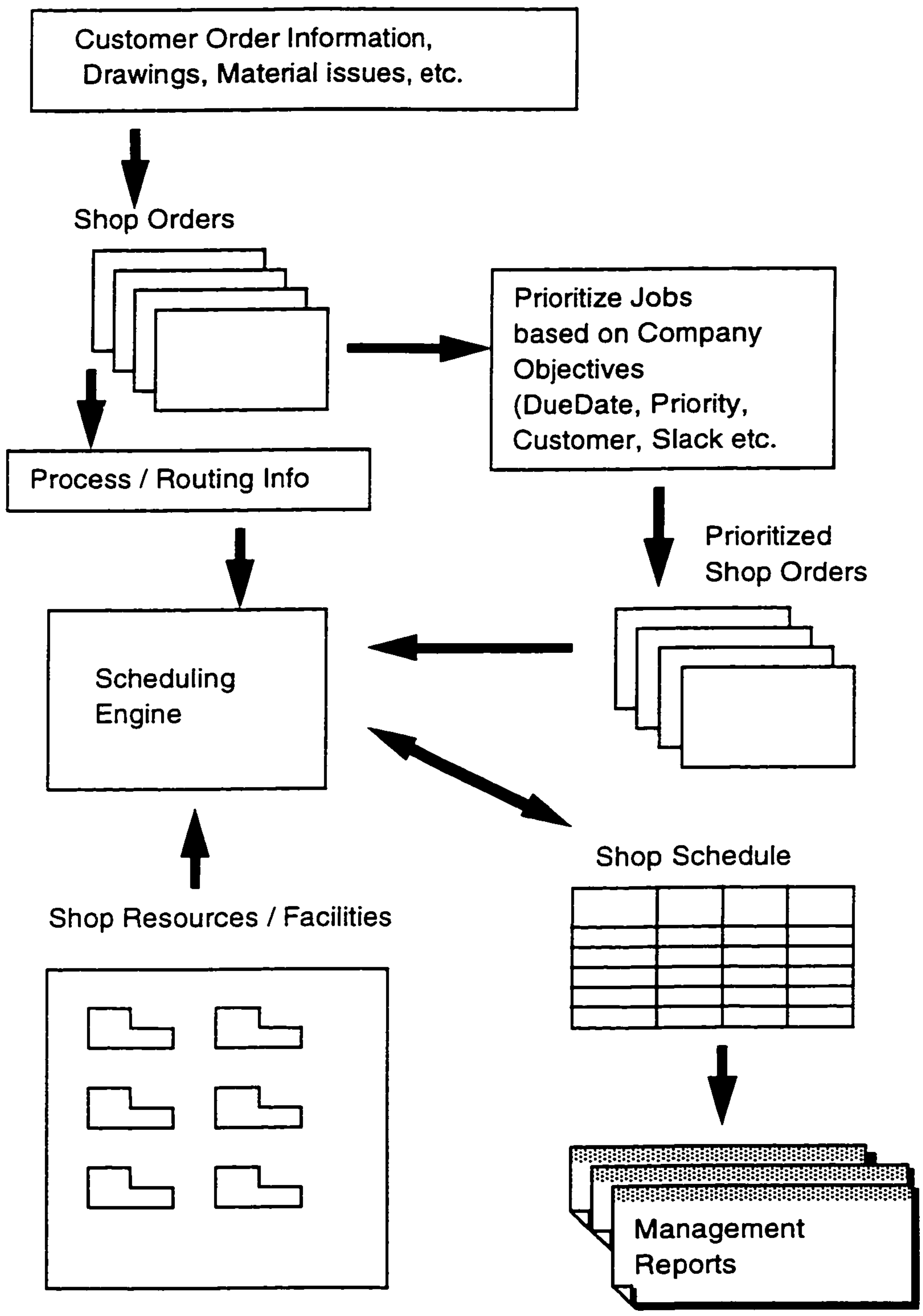


Figure 5-3: Framework for Shop Scheduling

5.4.2 Numerical Example

A problem presented in Jiang (1991) is chosen as a sample for use in the scheduling DSS, so that the results can be compared and the special features of the DSS illustrated. The data set comprises of 8 orders and 5 work centers containing 3, 2, 2, 3, and 1 machines, respectively. The details of the orders are shown in Tables 5-2 and 5-3 below.

Table 5-2: Order Information for the example

Order	Due Date	Number of alternate routes	Number of operations
A	40	1	4
B	55	2	55
C	56	1	5
D	46	1	4
E	42	2	3
F	60	2	5
G	43	2	4
H	60	2	5

The details of the processing information are provided in the form of {operation_no, work center, set-up time, processing time}.

Table 5-3: Process Routing Information for the example

Order	Process Details
A	{1 2 1 7}{2 1 0 10}{3 4 2 6}{4 3 3 10}
B	{1 3 0 9}{2 4 1 12}{3 1 1 13}{4 4 4 4}{5 2 0 10} {1 1 1 12}{2 2 3 14}{3 5 0 15}{4 1 2 7}{5 4 2 13}
C	{1 1 3 12}{2 3 0 9}{3 5 2 6}{4 1 0 7}{5 2 1 8}
D	{1 4 3 11}{2 2 4 5}{3 5 1 11}{4 3 2 9}
E	{1 2 2 11}{2 1 3 4}{3 5 4 5} {1 1 4 11}{2 2 4 7}{3 4 4 10}
F	{1 4 0 12}{2 2 1 9}{3 1 2 5}{4 5 0 13}{5 1 3 13} {1 3 1 12}{2 1 4 12}{3 3 0 9}{4 4 1 15}{5 2 2 15}
G	{1 2 1 11}{2 3 3 5}{3 5 1 5}{4 4 2 8} {1 5 3 11}{2 1 2 7}{3 2 1 14}{4 3 3 9}
H	{1 3 1 4}{2 2 2 13}{3 1 0 11}{4 4 3 10}{5 5 3 7} {1 4 1 6}{2 5 4 12}{3 4 0 13}{4 3 2 14}{5 2 3 10}

This information has been entered in the DSS, which offers greater flexibility in representing alternates. In Jiang's problem, the number of alternate routes applies to the complete order (Order B for example). In the DSS developed and presented here, the alternates can be specified at the individual operation level. The database relationships have been carefully designed to permit the handling of any complex process that may exist in a real-life scheduling situations. The relationships are as shown in Figure 5-4. The logic imbedded in the relationships in the DSS is as follows:

Each order is for a part that has a unique process. The process is made up of a number of steps called operations. Each of the operations can be done at a work center or its alternate, if one exist. At each of these work centers, the operation can be made up of sub-elements or segments, can take a finite duration, and may require additional resources.

Any number of operations can be defined for a process and each operation can have any number of alternate routings. At each of the alternate machines, any number of work segments can be defined and each segment can require any number of additional resources. It is also possible to define resource requirements (not considered in the sample) at the order level (which will apply to every operation of that order), or at the work center or machine level (which will apply to all work passing through that work center or machine).

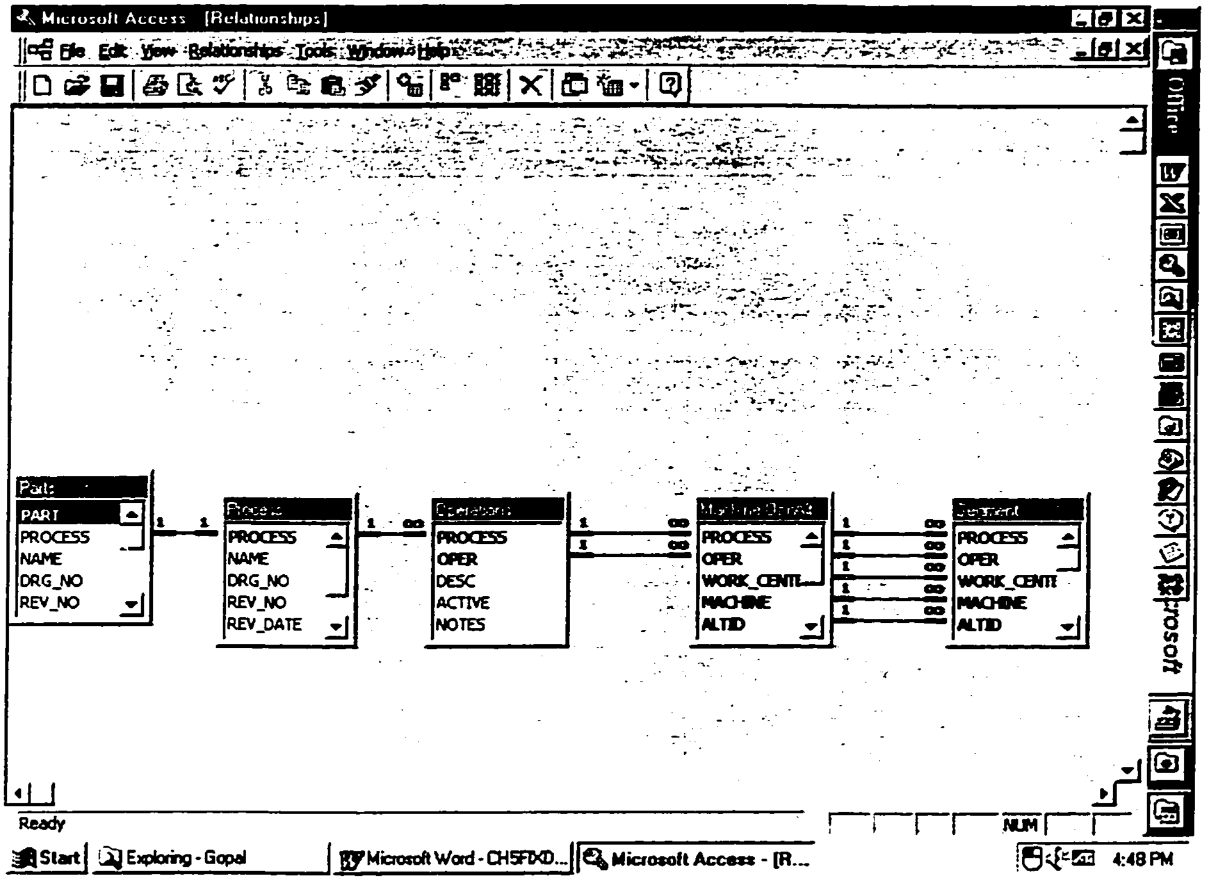


Figure 5-4: Database Relationships

5.4.2.1 Customer Order Entry and Order Management

Customer orders drive the entire system. The order entry module provides support to enter and track customer orders. It provides access to a long list of data options including suspending orders, order backlogging, start after a certain date, etc. Every issue to the shop is to be considered as an order for some product that is to be processed. Hence, each shop issue should have an Order Number, the part (or drawing) number – to identify what is to be done, a due date, priority, class etc. This information will be maintained in the database as Order Related Tables and has been designed to adequately handle the requirements of STEELR'US. Figure 5-5 shows the hierarchy of order information. There is an order master table that maintains customer information (Figure 5-6). The details of the order are maintained separately (Figure 5-7). The shop floor completions are gathered and the status updated using the order status updates (Figure 5-8). Pegging of orders can be accomplished through the order pegging tables (Figure 5-9) and definition of resources at the order level is done through the order resource tables (Figure 5-10).

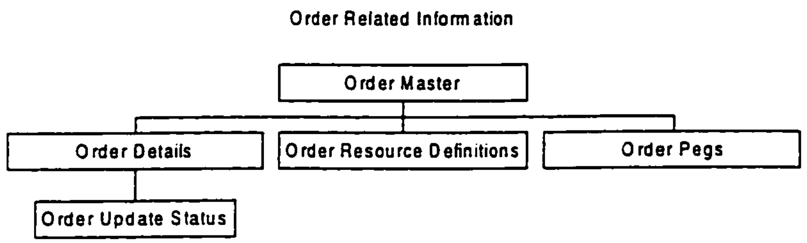


Figure 5-5: Order Information Hierarchy

From a scheduling point of view, one only needs an order number and the part that the order represents. If other information such as due date or priority are made available, then the scheduler can use that information to prioritize the orders before loading the shop.

Order Number	Part Code	Customer	Cust Order	Suspend	Complete	Order Value	Order Cost	Notes
ORDER1	A	A	A1	☐	☐	$10,000 00	$8,000 00	
ORDER2	B	B	B1	☐	☐	$220,000 00	$250,000 00	
ORDER3	C	B	B2	☐	☐	$48,000 00	$45,000 00	
ORDER4	D	D	D1	☐	☐	$999 00	$989 00	
ORDER5	E	E	E1	☐	☐	$4,000 00	$2,500 00	
ORDER6	F	E	E2	☐	☐	$19,875 00	$14,287 00	
ORDER7	G	E	E3	☐	☐	$4,400 00	$3,894 00	
ORDER8	H	E	E4	☐	☐	$100,000 00	$72,895 00	
				☐	☐			

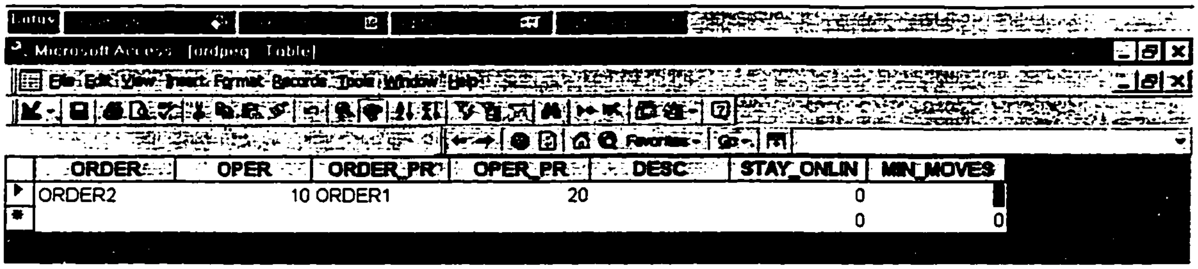
Figure 5-6: Order Information

Order	Order Qty	Rel Date	Due Date	Priority	Class	Start After	Order Rank	Efficiency	Lot Size	UDF1	UDF2	Ope
ORDER1	1	4/1/97	5/10/97	5	A	4/1/97	2147483647					
ORDER2	1	4/1/97	5/25/97	5	A	4/1/97	2147483647					
ORDER3	1	4/1/97	5/26/97	5	B	4/1/97	2147483647					
ORDER4	1	4/1/97	5/16/97	5	B	4/1/97	2147483647					
ORDER5	2	4/1/97	5/10/97	5	B	4/1/97	2147483647					
ORDER6	1	4/1/97	5/30/97	5	C	4/1/97	2147483647					
ORDER7	1	4/1/97	5/12/97	5	C	4/1/97	2147483647					
ORDER8	1	4/1/97	5/30/97	4	A	4/1/97	2147483647					
		10/25/97		5		10/25/97	2147483647					

Figure 5-7: Order Details

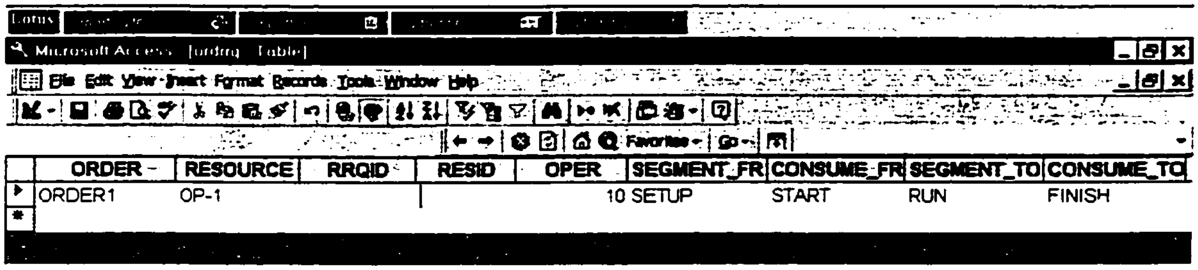
ORDER	OPER	Complete	SEGMENT	LASTVARSEG	MACHINE	CQUANTITY	CPERCENT	CTIME	OPER_ASTDT	OPER_ASTTI
ORDER1	10	☑	RUN	RUN	W1M1	1			7/15/97	10 2:
		☐								

Figure 5-8: Order Status Update



ORDER	OPER	ORDER_PR	OPER_PR	DESC	STAY_ONLIN	MIN_MOVES
ORDER2	10	ORDER1	20		0	
					0	0

Figure 5-9: Order Peg Definitions



ORDER	RESOURCE	RRQID	RESID	OPER	SEGMENT_FR	CONSUME_FR	SEGMENT_TO	CONSUME_TO
ORDER1	OP-1			10	SETUP	START	RUN	FINISH

Figure 5-10: Order Resource Definition

All the information maintained in the various orders related tables is presented to the user in a single form as shown in Figure 5-11.

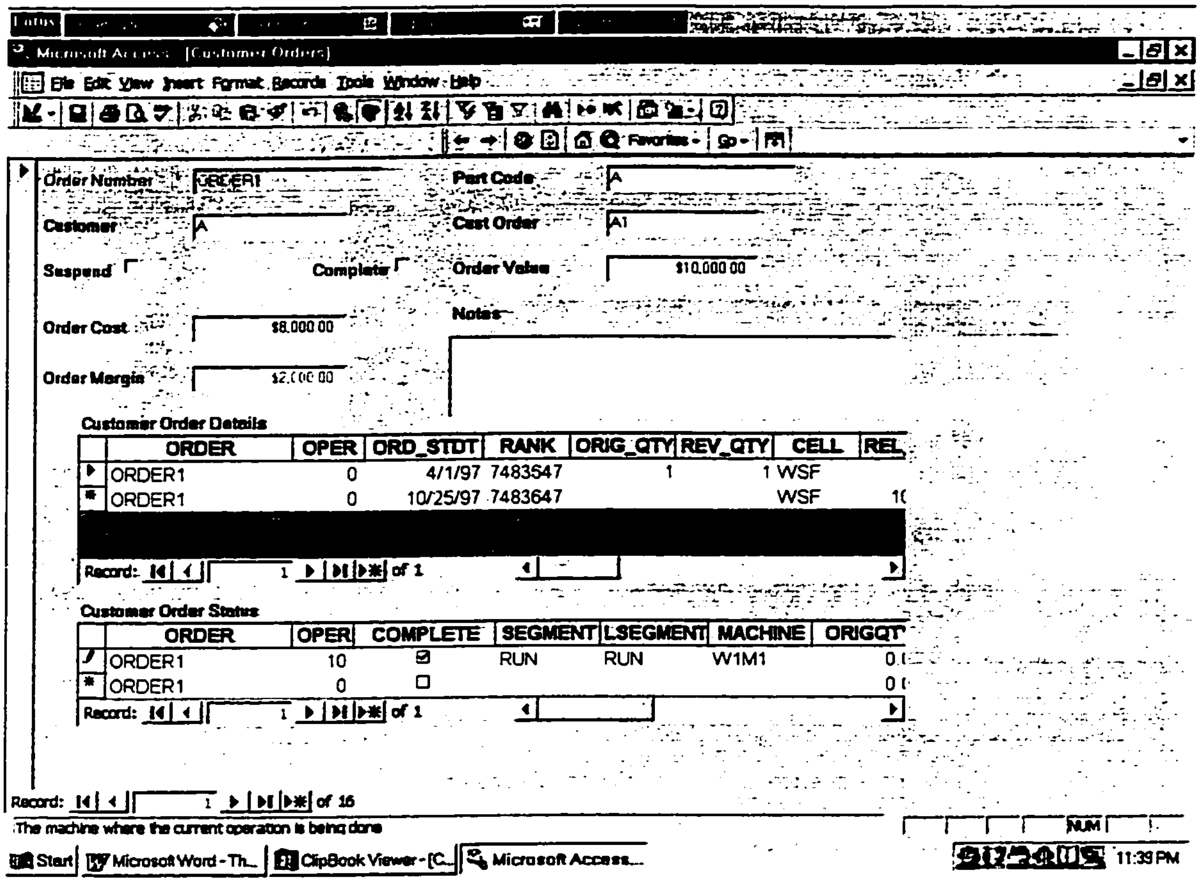
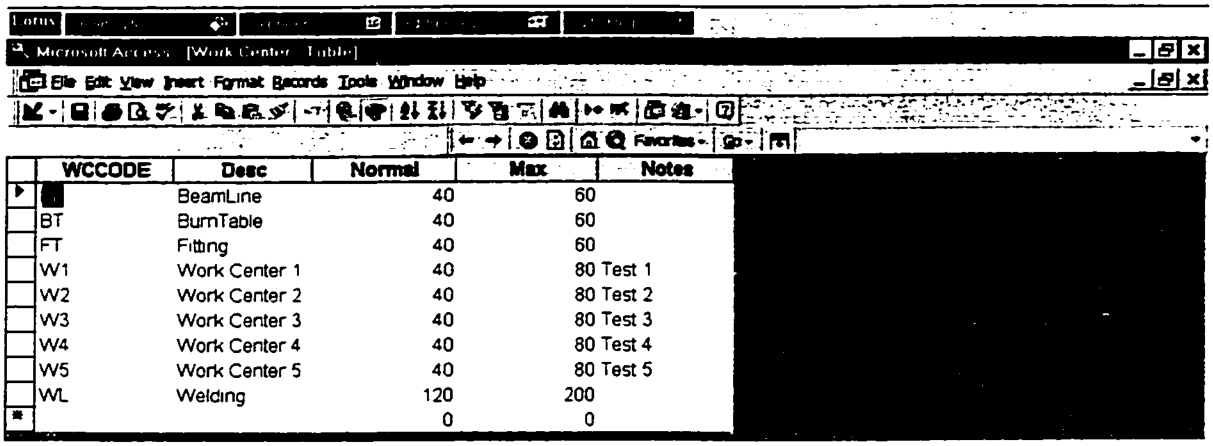


Figure 5-11: Customer Order Information Form

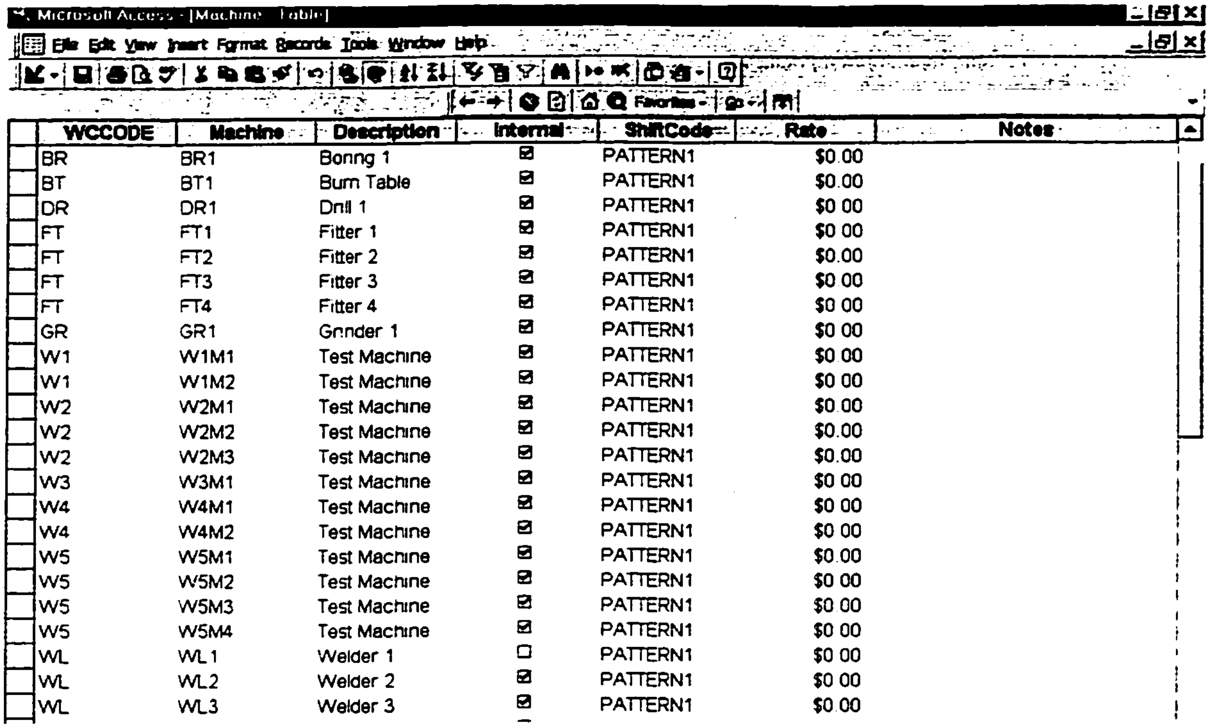
5.4.2.2 Shop Resource Information

The shop resources must be defined for use by the scheduling heuristic in terms of capacities available for use (usually hours of availability defined by a shift pattern). Typically, a shop will be broken down into work centers (Figure 5-12), and each work center may have one or more machines (Figure 5-13). There may be additional resources like Special Tools, Jigs and Fixtures, manpower, and materials that must be identified. Each of the resources can have a different availability pattern (Figure 5-14). Some machines may only be available for 1 shift, others may be available for 2 or 3 shifts. This can be easily represented by defining and using different shift patterns. It is possible to define external machines for use in scheduling, vendor facilities, for example, can be included in the machine table and marked as external. WL1 – Welder 1 shown in Figure 5-13, for example, is an external machine. The scheduling heuristic will offset the operations using external facilities by the operation times specified, assuming the external machines to have infinite capacity as we have no knowledge of the vendor's shop load information. All this information will be stored in the database as 'Resource Related Tables.'



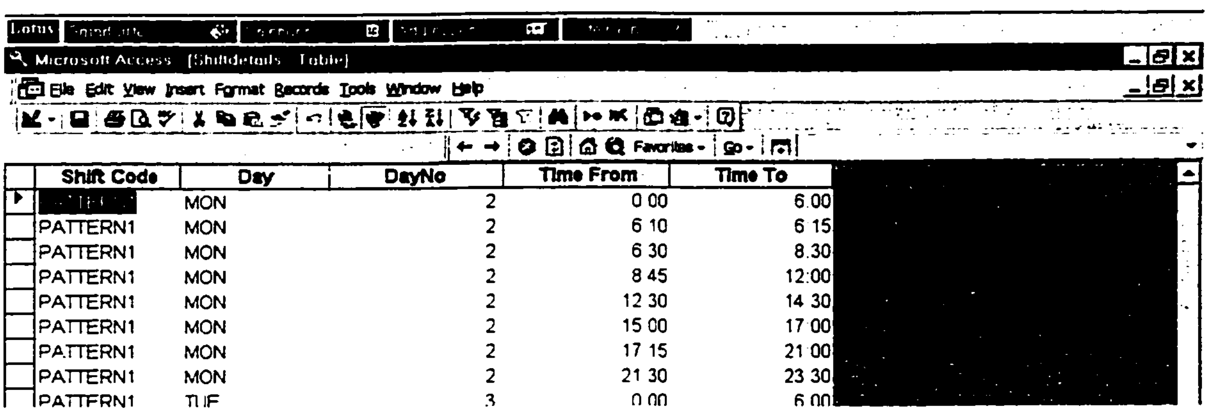
WCCODE	Desc	Normal	Max	Notes
[illegible]	BeamLine	40	60	
BT	BurnTable	40	60	
FT	Fitting	40	60	
W1	Work Center 1	40	80	Test 1
W2	Work Center 2	40	80	Test 2
W3	Work Center 3	40	80	Test 3
W4	Work Center 4	40	80	Test 4
W5	Work Center 5	40	80	Test 5
WL	Welding	120	200	
		0	0	

Figure 5-12: Work Center Information



WCCODE	Machine	Description	Internal	ShiftCode	Rate	Notes
BR	BR1	Boring 1	☑	PATTERN1	$0.00	
BT	BT1	Burn Table	☑	PATTERN1	$0.00	
DR	DR1	Drill 1	☑	PATTERN1	$0.00	
FT	FT1	Fitter 1	☑	PATTERN1	$0.00	
FT	FT2	Fitter 2	☑	PATTERN1	$0.00	
FT	FT3	Fitter 3	☑	PATTERN1	$0.00	
FT	FT4	Fitter 4	☑	PATTERN1	$0.00	
GR	GR1	Grinder 1	☑	PATTERN1	$0.00	
W1	W1M1	Test Machine	☑	PATTERN1	$0.00	
W1	W1M2	Test Machine	☑	PATTERN1	$0.00	
W2	W2M1	Test Machine	☑	PATTERN1	$0.00	
W2	W2M2	Test Machine	☑	PATTERN1	$0.00	
W2	W2M3	Test Machine	☑	PATTERN1	$0.00	
W3	W3M1	Test Machine	☑	PATTERN1	$0.00	
W4	W4M1	Test Machine	☑	PATTERN1	$0.00	
W4	W4M2	Test Machine	☑	PATTERN1	$0.00	
W5	W5M1	Test Machine	☑	PATTERN1	$0.00	
W5	W5M2	Test Machine	☑	PATTERN1	$0.00	
W5	W5M3	Test Machine	☑	PATTERN1	$0.00	
W5	W5M4	Test Machine	☑	PATTERN1	$0.00	
WL	WL1	Welder 1	☐	PATTERN1	$0.00	
WL	WL2	Welder 2	☑	PATTERN1	$0.00	
WL	WL3	Welder 3	☑	PATTERN1	$0.00	

Figure 5-13: Machine Information



Shift Code	Day	DayNo	Time From	Time To
[illegible]	MON	2	0 00	6.00
PATTERN1	MON	2	6 10	6 15
PATTERN1	MON	2	6 30	8.30
PATTERN1	MON	2	8 45	12:00
PATTERN1	MON	2	12 30	14 30
PATTERN1	MON	2	15 00	17 00
PATTERN1	MON	2	17 15	21 00
PATTERN1	MON	2	21 30	23 30
PATTERN1	TUE	3	0 00	6 00

Figure 5-14: Shift Definitions

Figure 5-15 presents the form that the user can use to define work centers and machines.

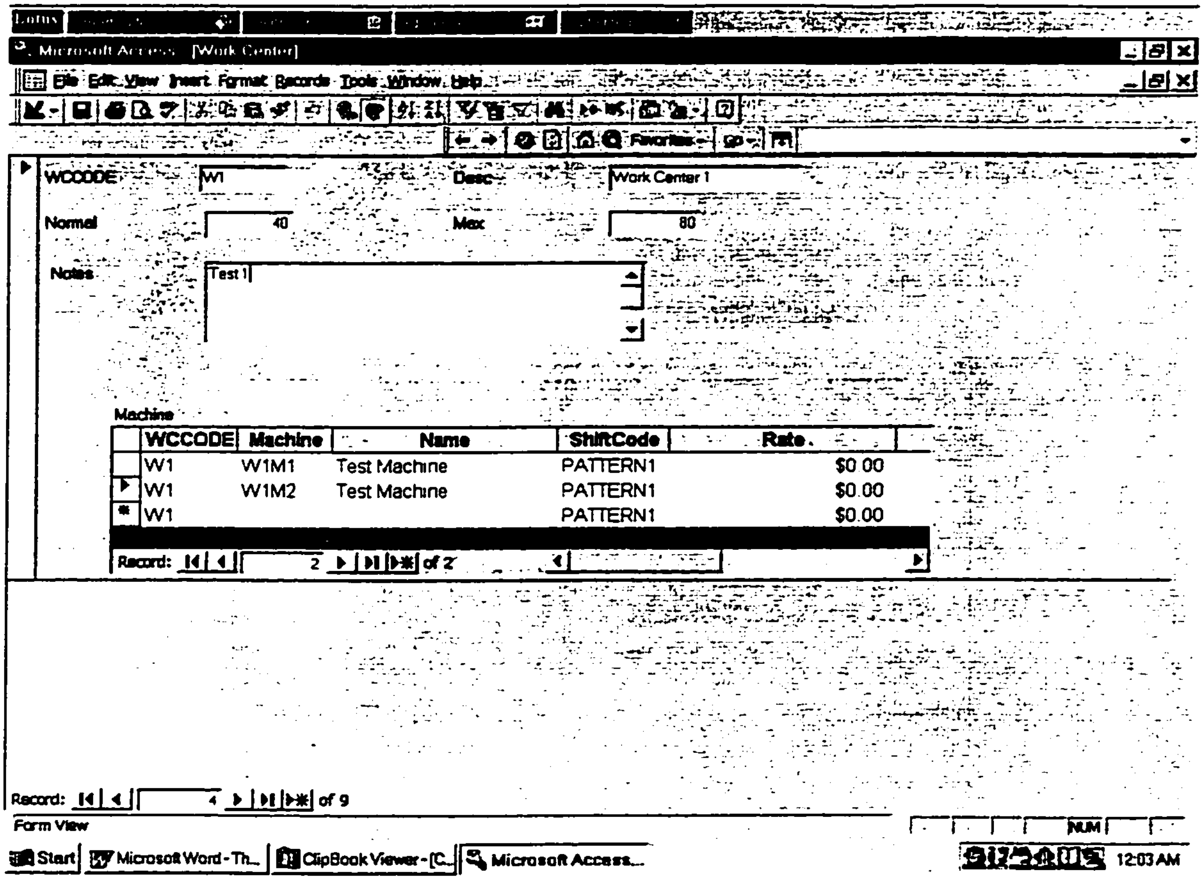


Figure 5-15: User Input form

5.4.2.3 Engineering Definition (Process Routings)

The DSS has been designed in such a way that complete configurations on all the manufacturing process can be easily maintained. Both standard routing and special routing can be maintained so that materials, tools, fixtures, or outside services can be designated as to where they are needed. As mentioned earlier, the hierarchy is defined in such as way that any real life shop can be easily modeled without additional programming effort.

Any number of operations can be defined for a process and each operation can have any number of alternate routings. At each of the alternate machines any number of work segments can be defined and at each segment any number of required resources can be specified. Sample process information is presented in Table 5-4 and the user form is shown in Figure 5-16.

Table 5-4: Sample Process Information

Process Name	Operation	Machine	Segment	Proc. Order	Rate	MaxMcn Delay	MaxSeg Delay	MaxMcn Gap	MaxSeg Gap
A	10.00	W2M1	SETUP	20.00	10	16	16	16	16
A	10.00	W2M1	RUN	30.00	40	16	16	16	16
A	10.00	W5M1	RUN	30.00	45	16	16	16	16
A	20.00	W1M1	RUN	30.00	10	16	16	16	16
A	30.00	W4M1	SETUP	20.00	20	16	16	16	16
A	30.00	W4M1	RUN	30.00	60	16	16	16	16
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
H	50.00	W2M1	SETUP	20.00	3	16	16	16	16
H	50.00	W2M1	RUN	30.00	10	16	16	16	16
H	50.00	W5M1	SETUP	20.00	3	16	16	16	16
H	50.00	W5M1	RUN	30.00	7	16	16	16	16

The scheduling DSS has special features built-in that are not usually found in standard packages. Four fields in the above table, viz., MaxMcnDelay, MaxSegDelay, MaxMcnGap, and MaxSegGap are designed to take care of one such special need.

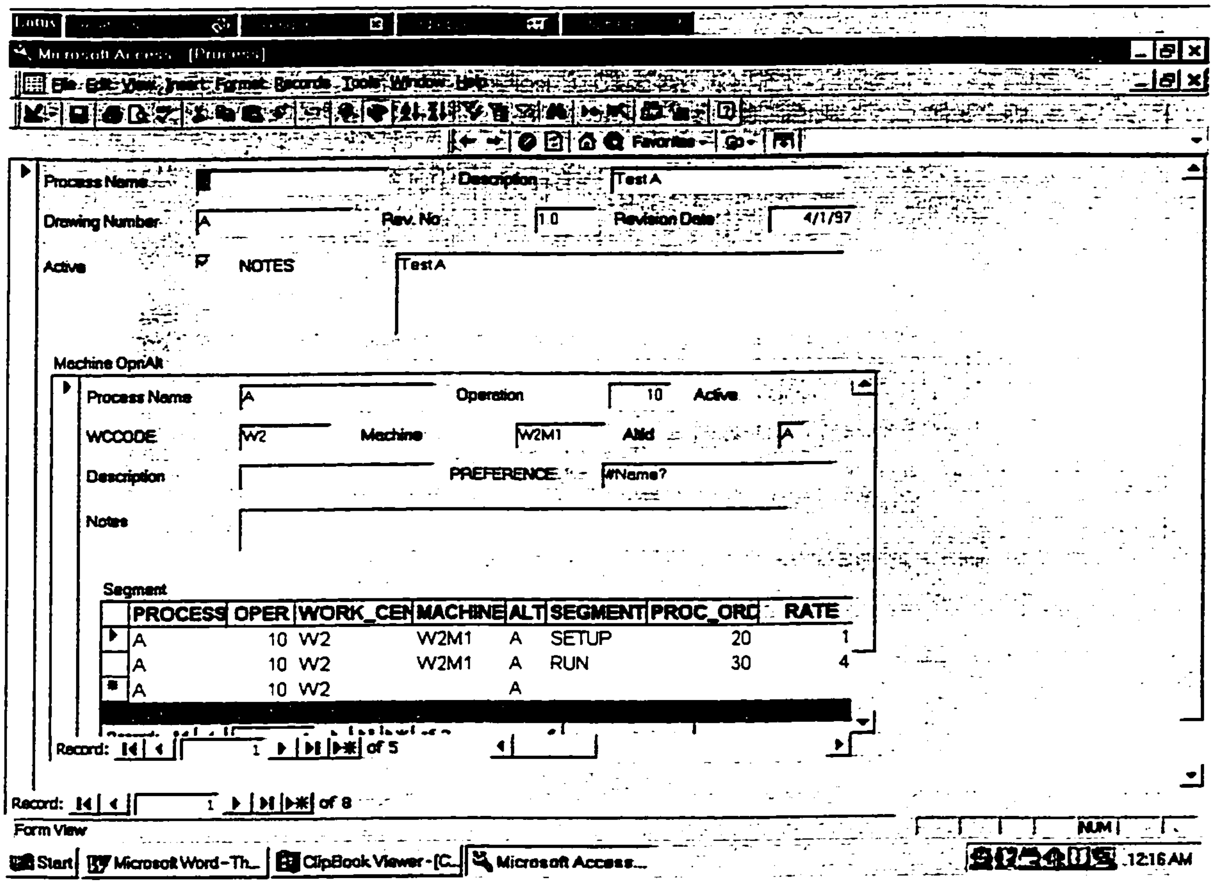


Figure 5-16: User Form for Process Information

For scheduling purposes, operations are broken down into segments. For example, a cutting operation may involve setup, run and cleanup as segments. Normally these segments would be scheduled together as shown in Figure 5-17-1. However, sometimes it may not be possible to schedule the segments together (For example, RUN may require some additional resource like an operator or tool that is only available later). MaxSegDelay specifies the amount of time delay permitted between segments as shown in Figure 5-17-2. Another possibility is a break in the segment (due to the end of a shift, for example). MaxSegGap specifies the amount of time gap permitted within a segment as shown in Figure 5-17-3. In some operations, gaps in segments may be prohibited (heat treatment, for example). In such cases the MaxSegGap can be specified to be '0' and the scheduling DSS will schedule the operation only if the segments can be completed without any breaks (Figure 5-17-1).

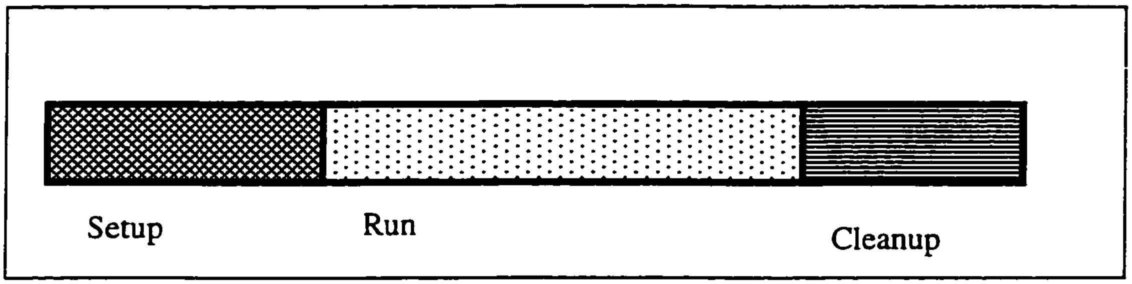


Figure 5-17-1: Segments scheduled without any delay

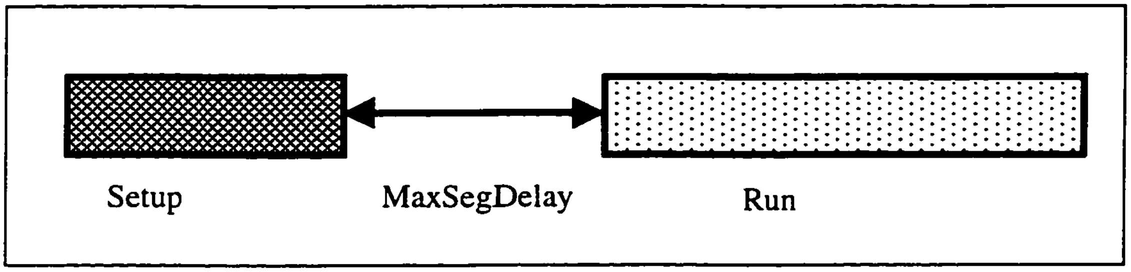


Figure 5-17-2: Segments scheduled with delay

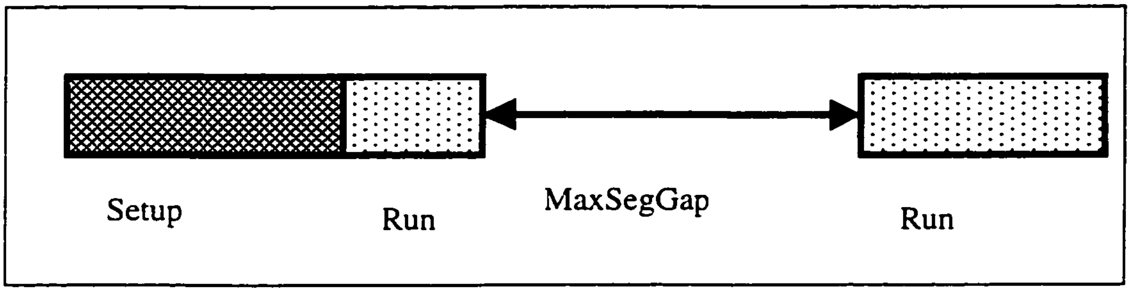


Figure 5-17-3: Segments scheduled with gaps

MaxMcnDelay and MaxMcnGap are similarly designed to take care of such situations between two operations on different machines.

5.4.2.4 Scheduling

In Jiang (1991), the schedule performance is measured using mean work-in-process (WIP), mean tardiness, and mean utilization. In the scheduling DSS, a number of scheduling rules have been built in so that the user can select either a single criterion or multiple criterion as shown below in Figure 5-18.

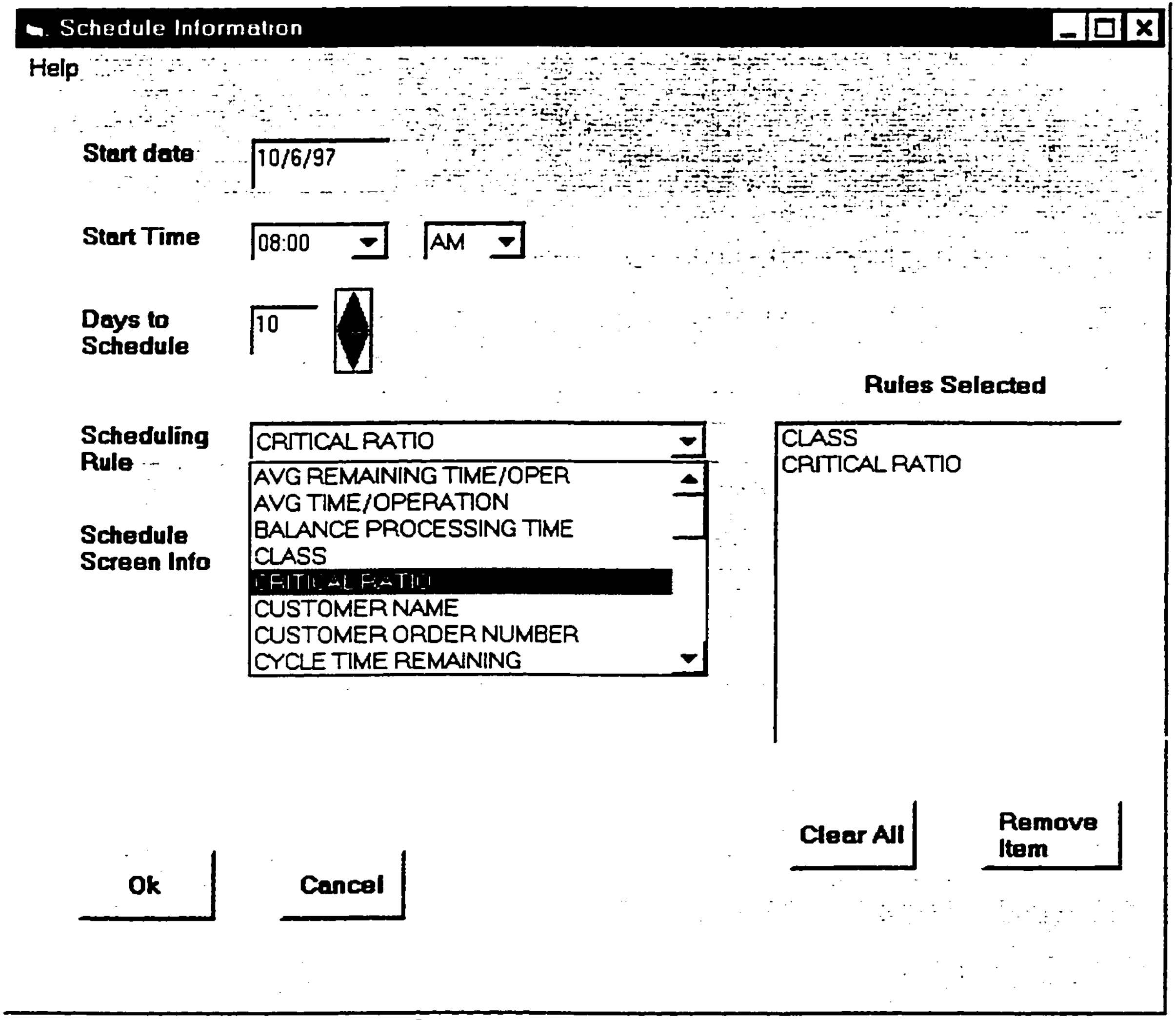


Figure 5-18: User Screen for Scheduling Parameters Selection

The scheduling DSS is very easy to use. Since the DSS schedules on a real calendar base, the user can specify a schedule start date, schedule start time, number of days to schedule, scheduling rules to be followed, and scheduling screen (for example, orders that are due within the next 5 days).

The schedule upon which the dispatch to the shop is based, is called the 'base' schedule. In addition, 'what-if' schedules can be generated to test the impact on order completion for different user specified configurations. The user can see if adding capacity (by additional shifts or overtime, or sub-contracting by specifying external machines) would resolve delivery problems, or if the bottleneck has now shifted elsewhere. Without tools like 'what-if', planning and analysis would be next to impossible.

Global scheduling takes all of the firm and released work orders, and schedules or re-schedules them. This process essentially generates a new schedule, taking into consideration what has been reported 'complete' on work order operations, what was planned to be completed but was not accomplished, and new work orders introduced to the mix since the last run of the scheduler. In addition, the DSS provides users with the option of freezing certain orders so that other orders are fitted around these orders without disturbing them.

From previous research, we know that the level of shop loading and the selected dispatching rule will affect the performance of the schedule. Since no one single dispatching rule can serve as a good performance indicator all the time, the user can select a number of dispatching rules of interest and a utility based analysis will indicate the best dispatching rule from those selected. A threshold value can also be specified to rate alternatives as significantly different from each other.

Since Jiang's output shown is not calendar based, in order to compare the results, the output of the DSS has been converted back to time units from its calendar base and is shown in Table 5-5.

Table 5-5: Summary of Jiang's Output

Order	Operation	Machine	Start	Finish
A	1	B1	1	8
	2	A3	9	18
	3	D2	17	24
	4	C1	28	40
B	1	C1	1	9
	2	B1	9	25
	3	A3	25	38
	4	D1	35	42
	5	B1	43	52
C	1	A1	1	15
	2	C1	19	27
	3	E1	28	35
	4	A1	36	42
	5	B2	42	50
D	1	D1	1	14
	2	B2	23	31
	3	E1	36	47
	4	C2	46	56
E	1	A2	1	15
	2	A1	16	22
	3	D1	19	32
F	1	D2	1	12
	2	B2	13	22
	3	A2	21	27
	4	D2	32	47
	5	A2	45	60
G	1	B2	1	12
	2	C2	10	17
	3	E1	18	23
	4	C2	21	32
H	1	C2	1	5
	2	E1	2	17
	3	D3	18	30
	4	D3	31	43
	5	E1	48	57

Schedule Statistics: The total make span is 60 time units and Utilization is 58%

Table 5-6: Output Summary from Scheduling DSS

Order	Operation	Machine	Start	Finish
A	1	B1	1	8
	2	A3	9	18
	3	D2	19	26
	4	C1	28	40
B	1	C1	1	9
	2	B1	10	26
	3	A3	27	39
	4	D1	40	47
	5	B1	48	57
C	1	A1	1	15
	2	C1	19	27
	3	E1	28	35
	4	A1	36	42
	5	B2	43	51
D	1	D1	1	14
	2	B2	23	31
	3	E1	36	47
	4	C2	48	58
E	1	A2	1	15
	2	A1	16	22
	3	D1	23	36
F	1	D2	1	12
	2	B2	13	22
	3	A2	23	29
	4	D2	32	47
	5	A2	48	63
G	1	B2	1	12
	2	C2	13	20
	3	E1	21	26
	4	C2	27	38
H	1	C2	1	5
	2	E1	6	21
	3	D3	22	34
	4	D3	35	47
	5	E1	48	57

Schedule Statistics: The total make span is 63 time units and Utilization is 58%

The utilization is the same in both cases as the number of orders to be scheduled is small (8) and the shop is not over loaded. The make span of the schedule developed by the DSS is larger by 3 time units mainly due to the method of scheduling followed. This is shown below in Figures 5-19-1 and 5-19-2.

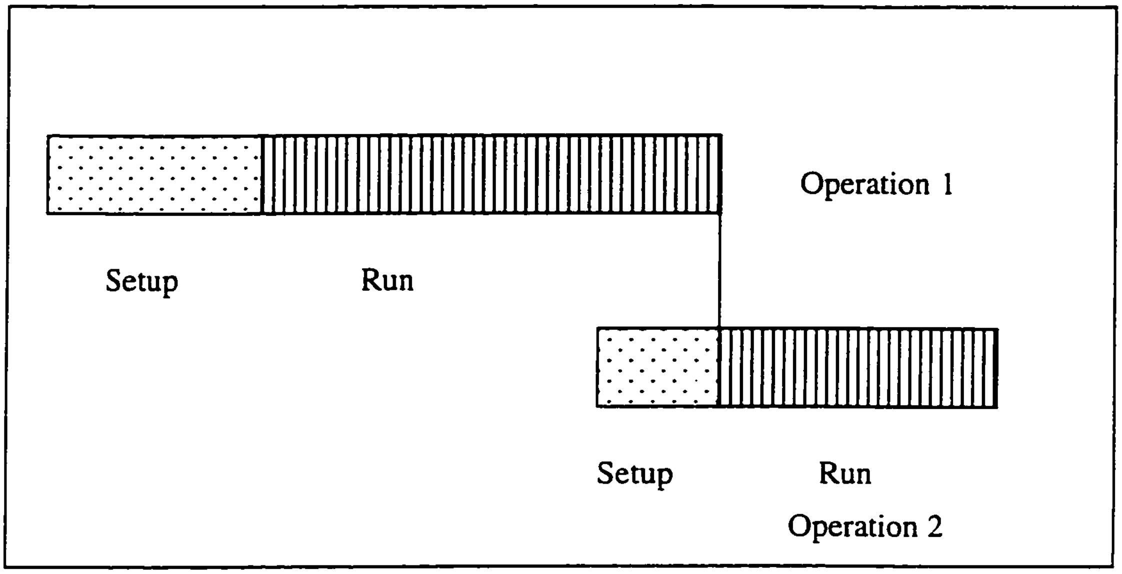


Figure 5-19-1: Jiang's scheduling method

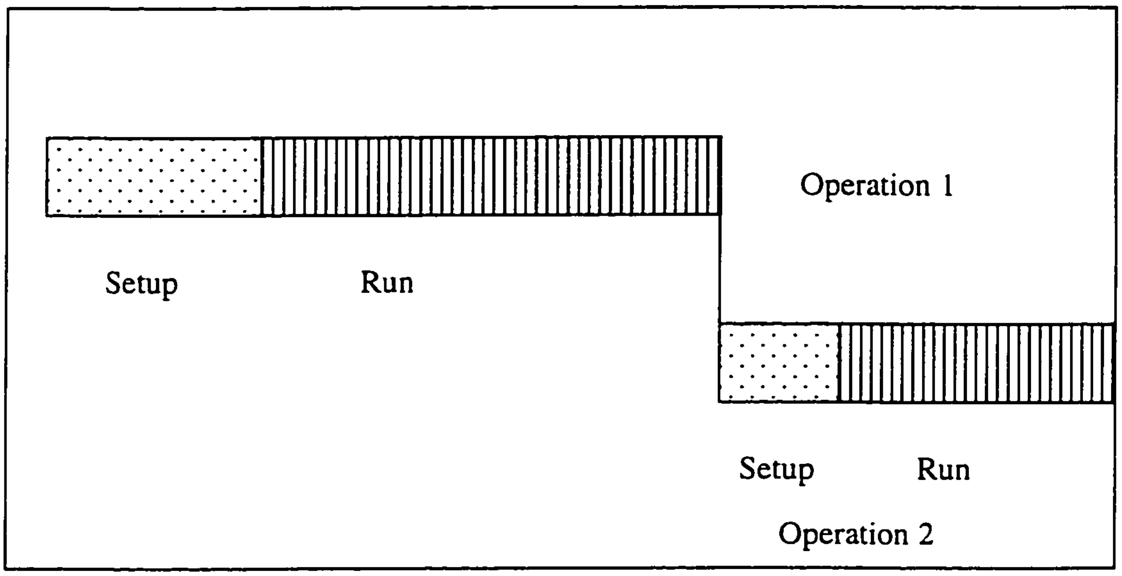
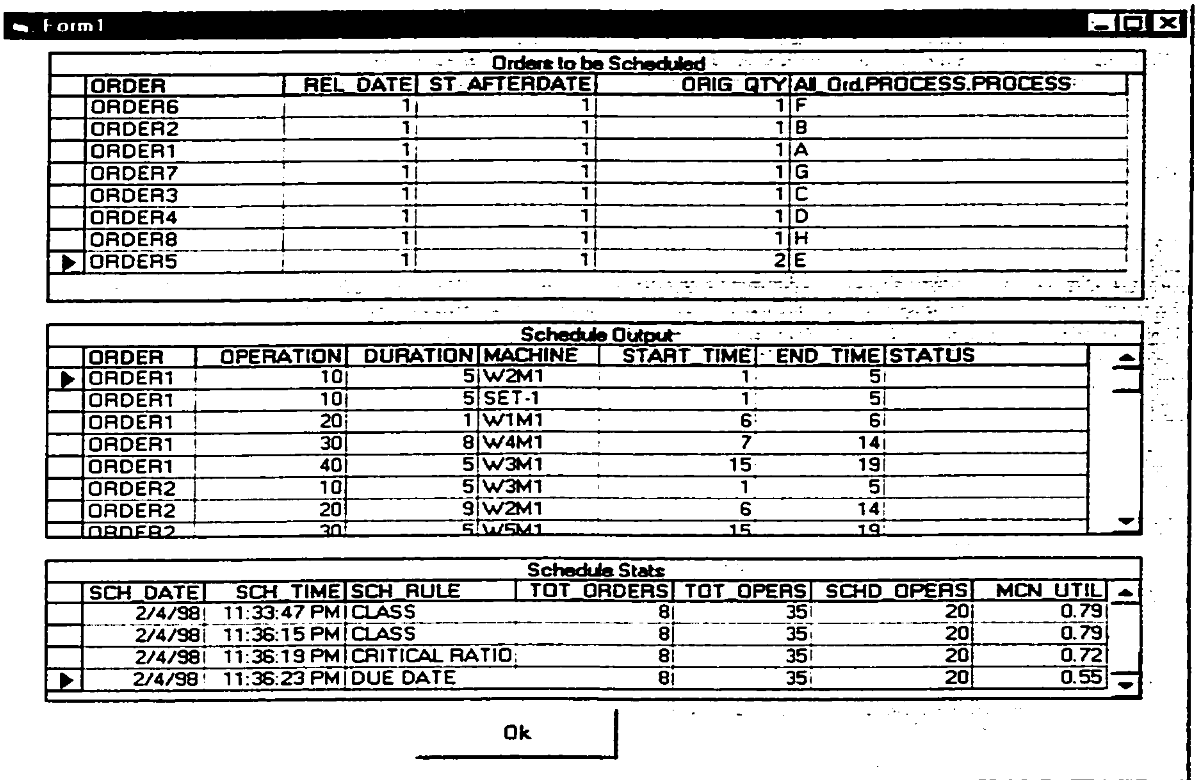


Figure 5-19-2: DSS scheduling method

If the DSS was modified to follow the same scheduling method, the results would be the same. The schedule generated by the DSS heuristic is therefore a valid schedule. The methodology followed (Figure 5-19-2) in the DSS has been designed so that segments can be automatically eliminated (if similar parts are processed one after another, setup or cleanup may not be required) without another pass of the scheduling heuristic. The standard output generated by the heuristic showing the schedule statistic along with the list of orders considered and the orders scheduled is shown in Figure 5-20.



Form1

Orders to be Scheduled

ORDER	REL_DATE	ST_AFTERDATE	ORIG_QTY	All_Ord.PROCESS.PROCESS
ORDER6	1	1	1	F
ORDER2	1	1	1	B
ORDER1	1	1	1	A
ORDER7	1	1	1	G
ORDER3	1	1	1	C
ORDER4	1	1	1	D
ORDER8	1	1	1	H
ORDER5	1	1	2	E

Schedule Output

ORDER	OPERATION	DURATION	MACHINE	START_TIME	END_TIME	STATUS
ORDER1	10	5	W2M1	1	5	
ORDER1	10	5	SET-1	1	5	
ORDER1	20	1	W1M1	6	6	
ORDER1	30	8	W4M1	7	14	
ORDER1	40	5	W3M1	15	19	
ORDER2	10	5	W3M1	1	5	
ORDER2	20	9	W2M1	6	14	
ORDER2	30	5	W5M1	15	19	

Schedule Stats

SCH_DATE	SCH_TIME	SCH_RULE	TOT_ORDERS	TOT_OPERS	SCHD_OPERS	MCN_UTIL
2/4/98	11:33:47 PM	CLASS	8	35	20	0.79
2/4/98	11:36:15 PM	CLASS	8	35	20	0.79
2/4/98	11:36:19 PM	CRITICAL RATIO	8	35	20	0.72
2/4/98	11:36:23 PM	DUE DATE	8	35	20	0.55

Ok

Figure 5-20: Output of Schedule Heuristic

5.5 Implementation Issues

A great deal of time and effort is required to successfully implement a scheduling DSS in an organization. Typical implementations take one to two years. The following strategies were identified for implementing the DSS at STEEL'US.

5.5.1 Implementation Planning

Implementation planning will help by advance planning and problem prevention efforts. Implementation planning was a pre-requisite for a smooth transition from the existing system to the DSS while avoiding confusion and misunderstanding. Implementation planning includes education of senior management; selection and training of proper personnel to run the system; formation of an implementation team with each functional representative it; clear statement of the objectives; and identification of costs, benefits, and time schedule for a phased implementation as there was a clear need for improvement of data accuracy. The management at STEELR'US came to appreciate the value of implementation planning as gaps in existing information were revealed (the need for maintaining original estimate information in a database, for example) at this phase and actions were initiated to rectify the situation.

5.5.2 Accuracy of data

In many organizations, record keeping is often lax and is managed by informal systems on the shop floor; STEELR'US is no exception. Informal systems are a major cause of failure. If accurate data is not available, the implementation should be put on hold and efforts to make data records accurate should be taken up. Once the records are accurate, they must be kept up to date. This should include both engineering changes and shop update status records. All other data, such as work center information, operations sequence information, and shop costs, should be screened to check for errors and then maintained in an error free manner. Keeping the data accurate for system integrity is the most important task in operating any computer system.

For the implementation of the scheduling DSS, four types of data were gathered and verified from STEELR'US. These were:

1. Order related information: That is information pertaining to the order viz., order number, customer identity and customer importance, order value, order priority, due date, and other user defined criteria. These were easy to gather from records. The order update status information, however, is currently lacking; data collection templates and methodologies have been designed for this purpose.
2. Shop resource related information: Information pertaining to the number of machines, their availability pattern and other supporting tools was modeled. The DSS's ability to consider outside vendor facilities was very helpful as some parts are sent out for surface finishing.
3. Process related information: For each part to be processed in the shop, a process detailing what is to be done (operations and their sequence), where it can be done (machine or alternate machines), its duration (how long it is expected to take), and any supporting resources (special jigs or setup personnel) must be specified. This area is the most lacking as the estimates are not yet readily available. The current shop loading philosophy of 'load lists' and rule-of-thumb value of 100 Tons of shipment a day is not sufficient for shop planning purposes. Generic templates are being added to the shop drawings to include standard processes of cutting, detailing, welding, inspection, painting and shipping. The initial runs would schedule parts according to the weight of the

pieces. A gradual time collection and standardization of processing times would enable a shift to schedule by duration in the future.

4. Scheduling related information: This consists of the length of the schedule, the rule(s) to be followed, the filters to be applied to screen orders, and files to write the output to. These are straight forward.

The DSS has been designed to maintain this information in the form of inter-related tables using a common database.

5.5.3 Automatic Data Collection using Bar Codes

In order for a newly generated schedule to be meaningful, the database must be kept up to date. Managers quickly discovered the need for automated collection and distribution of information. To speed the quality and timeliness of information, STEELR'US is in the process of installing a plant-wide network which features automatic up loading of shop floor processes from bar code data collection systems.

Transactions from the bar code systems can be processed in real time and managers will be able to see the status of work orders and review jobs from any desktop terminal or work station. The aim is to eliminate 'waste' – a term that includes more than idle, shelved inventory – moving, counting and inspecting inventory are all part of a wasteful process. This is where a bar code data collection system enters the picture. "With one swipe of a wand or a card, automated bar code systems can track part numbers, lot quantities, quality control, and activities by each work station." Currently, workers must manually count inventory at each work station, move material to the next work station and compute WIP.

The bar coding will be implemented in stages. The drafting area, shop issue area and shipping area will be the first to implement the bar coding technology. The drawings will be bar coded and the shop issues scanned. At a later date, in the fabrication area, an operator will scan a bar-coded drawing to indicate that a work order has been filled and forwarded to the next workstation. The bar code contains the work order number and the operator can key in the quantity and hit a pre-programmed function key, to indicate to which station the order has been moved. Pre-programmed keys can further streamline the gathering of information. The ability to quickly view the number of parts at any given work station will allow managers to shift workload to improve production. Work can be shifted from overloaded lines to those that are under utilized. Instant access to WIP status will enable managers to discover backlogs before production is slowed. At the end of each day, they will know exactly what they have produced and how close they are to operating on schedule. Bar code systems collect more information, more efficiently and make data readily available to decision-makers.

Installing bar code data collection systems throughout the plant should be a gradual process of installing – learning – correcting – relearning. Previously, operators filled out forms, which were then entered on a terminal by accounting personnel by the end of the week purely for payroll purposes. It took engineers days to analyze raw data. With bar code scanning, intelligent data collection terminals can assemble the data into transaction fields and upload complete transactions to the PC controller in real-time. The value of accurate, timely information rises exponentially, since managers will be tapping into that information daily to make critical competitive decisions.

5.5.4 Management support:

The importance of management support can hardly be over emphasized. Often top management is not fully committed to the system and its requirements. Lip service and passive support is not enough. Top managers must be involved in installing and operating the system. Managers must give their time and change the way they operate. The ultimate change required by personnel at all levels is to use the system and never override it by the informal system. The management at STEELR'US are fully committed to the DSS. They were involved in providing the management's preferences for the DSS to use for evaluating different scheduling rules.

5.5.5 Education and training

User knowledge is another important requisite for the DSS's success. A DSS requires an entirely new approach to manufacturing. All employees must grasp their new roles and responsibilities and understand how they will be affected by the DSS. When a DSS is first being installed, a few key personnel need to understand the system. As the system broadens in scope, the level of education within the organization must broaden as well. A team consisting of representatives from project management/estimating, drafting, purchasing, quality control and shop floor areas will initially be trained to use the system. After the trial runs are completed, these members will be able to train others in their departments to use the DSS.

5.5.6 Staged Implementation

Since STEELR'US is taking a leap from the current manual system to a computerized DSS, implementation will occur in stages. The order management modules will be initially implemented, as the data requirements of this module can be fully satisfied from current company records. In the second stage, generic templates will be added to the shop drawings and a shop representative will fill in the process information. The use of 'load lists' in the shop floor will continue concurrently through the 'Planning Board' that has been developed for such use. The planning board is merely a stop-gap tool consisting of a glorified electronic black board for planning shop activities (see Appendix 3). The third stage will involve the installation of bar coding and gathering of data using remote terminals. During this stage, the processing time estimates can be refined from actual shop performance data and the DSS can continuously update the information for future use. The last stage of the implementation will involve the design of management reports and integration of additional modules for estimating and accounting if needed.

5.6 Summary and Conclusions

STEELR'US is in the process of implementing the DSS for scheduling. Even though it is not operational at this time, the benefit to STEELR'US is expected to be considerable. They will be able to track orders as they progress through the shop, and analyze a variety of shop performance measures to identify variations between actual and estimated times. The ability to schedule orders based on organizational objectives and priorities, choose a proper scheduling rule, forecasting shop congestion information well ahead of its occurrence, predict the impact of delaying one order in preference to another, and other similar needs can easily be met once the scheduling DSS is operational.

The scheduling DSS can be used for operational level planning of steel fabrication shops. The purpose of designing the DSS is to close the gap between scheduling models in research and real scheduling problems faced by the construction industry. The DSS is able to take into account all the relevant shop information and solve complicated scheduling problems. To apply the DSS effectively, different organisations need different sequencing of dispatching rules in terms of different performance measures and other affecting factors. This problem is efficiently addressed by the utility theory based approach of the DSS in evaluating different scheduling rules.

6. Conclusion

6.1 Summary of Research

Developing a scheduling DSS for use in the steel fabrication industry has the potential for great gains, as there is a substantial need to allocate resources efficiently and effectively. A DSS for scheduling and control of a variety of shop configurations is developed and described in this thesis. The DSS provides a convenient and easy-to-use tool for planning shop floor activities. Implementing the research concept of generating a feasible finite capacity resource constrained schedule for use in the construction industry was made possible by the DSS. The DSS was designed to include a relational database that maintains a hierarchical structure for order information, process routing information, and shop resource information, along with a large number of scheduling rules that users can select.

There are a number of advantages to using the DSS for scheduling. The DSS contributes to decision making by providing a mechanism to evaluate different scenarios of interest. Traditional shop floor control tools such as order expedition, lack the perspective that the DSS can provide. For example, when due dates are missed, human schedulers can often act in a myopic fashion. One typical response is to give the single job that is behind schedule a higher priority so that it can be moved ahead to finish on time. However such actions can adversely affect on-time delivery of other jobs. A DSS can provide the human scheduler with the list of jobs that will be delayed if the late job is reprioritized.

The problem of scheduling a shop requires a tremendous amount of data. This research work attempts to overcome this handicap by permitting users to easily maintain and update an organizational database.

The scheduling heuristic employed in this research will always create a feasible schedule, although optimality cannot be guaranteed. The heuristic approach begins with identifying and ranking orders to be processed. The shop resource availability profile is then generated and orders are scheduled one by one in priority order until all orders are scheduled, or the end of the scheduling horizon is reached. The heuristic lends itself well to handling a variety of shop configurations and management policies such as batch flows, limited order schedules, limited scheduling horizon, order pegging, order overlapping, and inclusion of external vendor facilities.

A utility theory based approach has been proposed for the identification of a proper scheduling rule to be employed for scheduling based on the decision-makers' preferences.

6.1.1 Why a DSS

The present MRP approach to planning does not recognize the existence of capacity constraints or sequence-dependent setups in timing orders. Expediting, in the form of lot splitting or operation overlapping is not part of formal MRP logic, nor is the possibility of alternate routing for a particular job. The advantage of a DSS approach is that all these factors can be considered. The difficulty in implementing a sophisticated DSS for scheduling lies in the fact that manufacturing is a very dynamic environment.

Scheduling is both data and knowledge intensive. Human schedulers can be easily overburdened by the sheer magnitude of the problem due to the numerous permutations and combinations that are possible in any scheduling scenario. A computer-based system can easily cope with this problem, permitting the users to concentrate on better planning of shop activities.

6.1.2 Why a Heuristic

The scheduling problem that is presented in this thesis cannot be easily solved by mathematical programming approaches. This is mainly due to the infinite number of possible solutions to the problem. It is therefore not surprising that most shops use ad hoc scheduling procedures. The heuristic presented in this thesis provides a viable alternative for generating schedules in a short period of time, taking into account all the complexities of a real-life shop floor and management objectives. If needed, the shop personnel can then make modifications to the schedule to suit their mode of operations.

6.2 Applicability of DSS for Scheduling

The DSS for scheduling developed in this thesis is intended for use primarily in the industrial field, but can be a valuable tool in the academic field as well. The open structure of the database design permits modeling of a wide variety of shop configurations and easy additions to the model base of the heuristic as new dispatching rules are developed and tested for use. The heuristic can handle multiple resources or ignore them, if the user so chooses. As an academic tool, the DSS can be used to force the modeler to think of the many planning and control issues that must be dealt with in actual shop conditions.

6.3 Benefits of the Scheduling DSS

The scheduling DSS has been designed to provide the following benefits:

1. Ability to support the solution of complex problems. A variety of scheduling environments can be handled. Machines may be available for different shift patterns, the number of alternates available is not limited, and resources can be specified at the order, part, process, or machine segment levels.
2. Fast response to unexpected situations resulting in changed conditions. The DSS will enable a thorough quantitative analysis in a very short time. Even frequent changes to a scenario can be evaluated objectively in a timely manner.
3. Ability to try several different strategies under different configurations quickly and objectively, and to save schedules under different scenarios.
4. New insights and learning. The user can be exposed to new insights through the composition of the model and an extensive sensitivity 'what-if' analysis. The new insights can help in training inexperienced managers and other employees as well.
5. Facilitated communication. Data collection and model construction experimentation are executed with active users' participation, thus greatly facilitating communication between managers. The decision process can make employees more supportive of organizational decisions. The "what-if" analysis can be used to satisfy skeptics, thus improving teamwork.
6. Improved management control and performance. DSS can increase management control over expenditures and improve overall performance of the organization.
7. Cost savings. Routine applications of a DSS may result in considerable cost reduction, or in reduction (elimination) of the cost of wrong decisions.
8. Objective decisions. The decisions derived from DSS are more consistent and objective than decisions made intuitively.

9. Improvement of managerial effectiveness, allowing managers to perform a task in less time and/or with less effort. The DSS provides managers with more 'quality' time for analysis, planning and implementation.
10. DSS is quite useful in capacity management, giving the novice scheduler access to the advice of an expert.

6.4 Contribution to Knowledge

The research contributes to knowledge in the following ways:

- the MRP approach to planning does not recognize the existence of capacity constraints or sequence-dependent setups in timing orders. Expediting, in the form of lot splitting or operation overlapping is not part of formal MRP logic, nor is the possibility of alternate routing for a particular job. The advantage of a DSS approach is that all these factors can be considered.
- It provides a mechanism to integrate organizational data in a hierarchical database format thereby enabling organizational objectives to be evaluated through a DSS.
- It provides a heuristic that can generate feasible schedules employing many rule combinations specified by the user.
- It provides a new utility theory based approach for evaluating effectiveness of scheduling parameters chosen by users.
- It demonstrates the functionality and feasibility of scheduling large number of shop orders.
- Through the decision support tool, it provides a competitive edge to organizations to manage this complex capacity management issue.

6.5 Recommendations for Future Research

The DSS for scheduling demonstrates the feasibility of managing a large shop. With the design concepts described in this thesis; future work can be undertaken to study the effect of incorporating AI based approaches to this field.

There are a number of issues that need to be researched in greater detail.

- Schedule Stability: The stability of the schedules generated using different scheduling rules and shop congestion information is worth examining, as this will provide better insights into the proper selection of scheduling rules depending on shop conditions.
- Bottleneck Studies: The throughput of most shops is constrained by bottlenecks. The study of arrival patterns at bottlenecks, and the ability to bring in additional facilities once a certain loading level is reached at the bottleneck, can be incorporated in the model base to study the overall impact on the system.
- Capacity and Productivity Studies: In the area of 'one-of-a-kind' jobs, even though the generic processes are followed, a method to capture the complexities of the work based on a set of features can be taken up; a neural-network based approach can be utilized to arrive at expected durations for each of the activities that can be used for scheduling.
- User Interface: Data input and the graphical user interface can be further enhanced to provide a more 'user friendly' interface. The data input interface can be improved to include scanning of textual information, which can then be stored automatically in the appropriate fields in the database. The graphical user interface can be improved to include tree views for hierarchical information such as process routing.

- Dependent demand, dependent routing and sequence dependant setups should be studied in greater detail.
- Expert Systems: The development of expert systems for scheduling is in its infancy. The inclusion of neural networks for generating process information from drawings is another area with potential for further research.

Bibliography

1. Atabaksh, H., (1991), A survey of constrained based scheduling systems, AI Engineering.
2. Baker, K.R., (1974), Introduction to Sequencing and Scheduling, Wiley, New York.
3. Bensana, E., and Dubois, D., (1988), OPAL: A multi knowledge based system for industrial job shop scheduling, International Journal of Production Research.
4. Biegel, J.E., and Davern, J.J., (1990) Genetic Algorithms and job shop scheduling, Computers in Industrial Engineering, 19, 81-91.
5. Bitran, G.R., and TiruPathi, D., (1988), Planning and scheduling of epitaxial wafer production facilities, Operations Research.
6. Blackstone, J.H. Jr., Philips, D.T., and Hogg, G.L., (1982), A state-of-the-art survey of dispatching rules for manufacturing job shop operations, International Journal of Production Research.
7. Blazewick, J., Ecker, K.H., Schmidt, G., and Weglarz, J., (1994), Scheduling in Computer and Manufacturing Systems, Springer-Verlag, Berlin.
8. Blazewick, J., (1987), Selected topics in scheduling theory, in S. Martell et al., (eds.), Annals of discrete mathematics, Vol. 31, Elsevier, Amsterdam.
9. Bonczek, R.H., C.W. Holsapple, and A.B. Whinston, (1981), Foundations of Decision Support Systems, Academic Press, NY.
10. Bruno, L.J., Coffman, Jr., E.G., and Sethi, R., (1974), Scheduling independent tasks to reduce mean finishing time, ACM Communications, Vol. 17, pp 382-387.

11. Cheng, T.C.E., and Gupta, M.C., (1989), Survey of Scheduling Research involving Due-Date Determination Decisions, European Journal of Operational research, Vol. 38, No 2, pp 156-168.

12. Cheng, T.C.E., and Sin, C.C.S., (1990), A state-of-the-art review of parallel-machine scheduling research, European Journal of Operational Research, 47, 271-292.

13. Churchman, C.W., (1968), Challenge to Reason, McGraw-Hill, NY.

14. Coffman, Jr., E.G. (ed.) (1976), Computer and Job Shop Scheduling Theory, John Wiley.

15. Collinot, A.., Pape, C.L., and Pinoteau, G., (1988), SONIA: A knowledge based scheduling system, AI Engineering.

16. Conway, R.W., Maxell, W.L., and Miller, L.W., (1967), Theory of Scheduling, Addison-Wesley, Reading MA.

17. Costello, T.W. and S.S. Zalkind, (1983), Psychology in Administration: A Research Orientation, Prentice-Hall, Englewood-Cliffs, NJ.

18. Dempster, M.A.H., Lenstra, J.K., and Rinnooy Kan, A.H.G. (Eds.) (1982), Deterministic and Stochastic Scheduling, D. Reidel Publishing Company, Dordrecht.

19. Dudek, R.A., Panwalkar, S.S, and Smith, M.L., (1992), The lessons of flow shop scheduling research, Operations Research, Vol. 40, No. 1, pp 7-13.

20. Erman, L.d., et al., (1980), The Hearsay II speech understanding system: Integrating knowledge to resolve uncertainty, Computing Surveys.

21. Fishburn, P.C., (1964), Decision and Value Theory, John-Wiley, NY.

22. Fox, M.S., (1987), Constrained Directed Search: A Case Study in Job Shop Scheduling, Morgan Kaufman, California.

23. French, S., (1982), Sequencing and Scheduling: An introduction to the Mathematics of the Job Shop, Wiley.

24. Garey, M.R., and Johnson, D.S., (1979), Computers and Intractability: A Guide to the Theory of NP - Completeness, Freeman, San Francisco, CA.

25. Geoffrion, A.M., and T.J. Van Roy, (1979), Caution: Common Sense Planning Methods can be Hazardous to your Corporate Health, Sloan Management Review.

26. Gere, W.S., (1966), Heuristics in Job Shop Scheduling, Management Science, No. 13.

27. Gittins, J.C., (1979), Bandit processes and dynamic allocation indices, Journal of the Royal Statistical Society.

28. Gopalakrishnan, K.N., (1988), The Design of a Capacity Requirements Planning System and a Decision Support System for Starategic Planning, Unpublished M.Sc. thesis, University of Alberta.

29. Gorry, G.A., and M.S. Scott-Morton, (Fall 1971), A Framework for Management Information Systems, Sloan Management Review, Vol. 13, No. 1.

30. Gupta, J.N.D., (1971), The Generalized n-job m-machine Scheduling Problem, Opsearch, 8, 173-177.

31. Gupta, S.K., and Sen, T., (1983), Minimizing a quadratic function of job lateness on a single machine, Engineering Costs Production Economics.

32. Hastings, N.A.J., and Yeh, C.-H., (1990), Job Oriented Production Scheduling: Theory and Methodology, European Journal of Operational Research, Vol. 47, pp 35-48.

33. Holsapple, C.W., and Whinston, A.B., (1996), Decision Support Systems: A Knowledge-Based Approach, West Publishing Company.

34. Huang, W.V., and Jianxin Tang, (1990), A new scheduling method for manufacturing systems, Computers in Industrial Engineering, 19, 22-26.

35. Janis, I.L., and L. Mann, (1977), Decision Making: A Psychological Analysis of Conflict, Choice, and Commitment, The Free Press, NY.

36. Jiang, Jui-chin(Kenny), (1991), IS: An Intelligent Scheduler for Batch Manufacturing Systems, Computers and Industrial Engineering, Vol. 20, Nos 1-4, pp 319-323.

37. Karp, R.M., (1972), Reducibility among combinatorial problems in: R.E. Miller and J.W. Thatcher(eds.), Complexity of Computer Computations, Plenum Press, New York.

38. Karp, R.M., (1975), On the Computational Complexity of Combinatorial Problems, Networks, Vol. 5, pp 45-68.

39. King, J.R., (March, 1976), The Theory-practice gap in job-shop scheduling, Production Engineering.

40. Keen, P.G.W., (Fall 1980), Adaptive Design for Decision Support Systems, Database, Vol. 12, Nos. 1 and 2.

41. Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G., (1982), recent developments in deterministic sequencing and scheduling: A survey, in M.A.H. Dempster, J.K. Lenstra, and A.H.G. Rinnooy Kan (eds.), Deterministic and Stochastic Scheduling, D. Riedel Publishing Company, Dordrecht, pp 35-74.

42. Lee, Taehee, (1990), New Job Regeneration Methods for Computer Integrated Manufacturing Environment, Computers in Industrial Engineering, Vol. 19, No. 1-4, pp 32-36.

43. Lenstra, J.K., and Rinnooy Kan, A.H.G., (1979), Computational Complexity of Discrete Optimization, in: J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Van Emde Boas(eds.), Interfaces between Computer Science and Operations Research, Proceedings of a Symposium held at Mathematisch Centrum, Amsterdam, Mathematisch Centrum, pp 64-85.

44. Little, J.D.C., (1970), Models and Managers: The Concept of a Decision Calculus, Management Science, Vol. 16, No. 8.

45. Magee, J.F., and Boodman, D.M., (1967), Production Planning and Inventory Control, McGraw-Hill, New York.

46. Manne, A.S., (1960), On the job shop scheduling problem, Operations Research. No. 8.

47. McKay, K.N., Safayeni, F.R., and Buzacott, J.A., (1988), Job-Shop Scheduling: What is Relevant?, Interfaces, Vol. 18, No 4, pp 84-90.

48. Moodie, C.L., and Roberts, S.D., (1968), Experiments with priority dispatching rules in a parallel processing shop, International Journal of Production Research.

49. Muhelmann, A.P., et al., (1982), Job shop scheduling heuristic and frequency of scheduling, International Journal of Production Research.

50. Muth, J.F., et al., Industrial Scheduling, Prentice-Hall, Englewood Cliffs, NJ.

51. Newman, P.A., (1988), Scheduling in CIM systems, Artificial Intelligence: Implications for CIM, ed. Andrew Kusiak, Springner-Verlag.

52. O'Grady, P.J., (1985), A general search sequencing rule for job shop sequencing, International Journal of Production Research.

53. O'Grady, P.J., and Kwan, H.L., (1988), An intelligent cell control system for automated manufacturing, International Journal of Production Research.

54. Onur, L., and Fabrycky, J., (1987), An input-output control system for the dynamic job shop, IIE Transactions.

55. Orlicky, Joseph, (1976), Material Requirements Planning, McGraw-Hill Book Co., New York.

56. Ow, P.S., (1985), Focused scheduling in proportionate flow shops, Management Science.

57. Panwalkar, S.S., and Iskander, W., (1977), A Survey of Scheduling Rules, Operations Research, Vol. 25, No 1, pp 45-61.

58. Parunak, H.V.D., (1988), Distributed Artificial Intelligence, in A Kusiak (ed.), Artificial Intelligence: Implications for CIM, IFS, UK.

59. Parunak, H.V.D., (1991), Charactrizing the Manufacturing Scheduling Problem, Journal of Manufacturing Systems, No. 10, pp 241-259.

60. Pearl, J., (1984), Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, Reading:MA.

61. Pinedo, M., (1995), Scheduling: Theory, Algorithms, and Systems, Prentice Hall, Englewood Cliffs.

62. Rafaeli, D., Mahalel, D., and Prashker, J.N., (1993), Heuristic approach to task scheduling: weight and improve algorithms, International Journal of production Economics, 29, 175-186.

63. Rajasekara, J.R., et al., (1991), A due-date assignment model for a flow-shop with application in light guide cable shop, Journal of Manufacturing Systems, No. 10, pp 1-7.

64. Ramesh, R, and Carey, J.M., (1989), Multi-criteria job shop scheduling, Computers in Industrial Engineering.

65. Randhawa, S.U., and McDowell, E.D., (1990), An investigation of the applicability of expert systems to job shop scheduling, International Journal of Production research, No. 32, pp 203-213.

66. Reisman, A., Kumar, A., and Motwani, J., (1997), Flowshop Scheduling/Sequencing Research: A Statistical Review of the Literature, 1952-1994, IEEE Transactions on engineering Management, Vol. 44, No. 3, pp 316-329.

67. Rinnooy Kan, A.H.G., (1976), Machine Scheduling Problems: Classification, Complexity and Computations, Martinus Nijhoff, The Hague.

68. Rowe,A.J., (1988), The Meta Logic of Cognitively Based Heuristics, Special Report, University of Southern California.

69. Ruiz-Diaz, F. and French, S., (1983), A survey of multi-objective combinatorial scheduling, in S.French et al., (eds.), Multi-Objective Decision Making, Academic Press, NY.

70. Selen, W.J., and Heuts, R.M.J., (1990) Operational production planning in a chemical manufacturing environment, European Journal of Operational Research, 45, 38-46.

71. Shaffer Jr., F., and Lifson, M.W., Decision and Risk Analysis for Construction Management, John Wiley, 1982.

72. Shaw, M.J., (1987), A distributed scheduling method for computer integrated manufacturing: The use of local area network in cellular manufacturing, International Journal of Production Research.

73. Simon, H.A., (1960), The New Science of Management Decision, Harper and Row, New York.

74. Smith, R.G., (1980), The contract net protocol: High level communication and control in a distributed problem solver, IEEE Transactions in Computing.

75. Smith, S.F., Fox, M.S., and Peng, S.O., (1986), Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-based Factory Scheduling Systems, AI , Fall 1986, pp 45-61.

76. Smith, S.F., and Hynyen, J.E., (1987), Integrated decentralization of production management: An approch for factory scheduling. Symposium on Integrated and Intelligent Manufacturing, Boston, MA.

77. Speranza, M.G., and Woerlee, A.P., (1991) A decision support systems for operational production scheduling, European Journal of Operational Research, 55, 329-343.

78. Sprague, Jr., R.H., and Watson, H.J., (1996), Decision Support for Management, Prentice Hall, New Jersey.

79. Suresh, V., and Chaudhuri, D., (1993) Dynamic Scheduling - A survey of Research, International journal of Production Economics, 32, 53-63.

80. Svetka, J.A., and Jiang, Jui-chin (Kenny), (1990), Towards the development of realistic rescheduling methods in real batch shops, Computers and Industrial Engineering, Vol. 19, Nos 1-4, pp 6-10.

81. Talliard, E., (1990) Some efficient heuristic methods for the flow shop sequencing problem, European Journal of Operational Research, 47, 65-74.

82. Turban, E., (1993), Decision Support and Expert Systems: Management Support Systems, Macmillan Publishing Company.

83. Udo, G.J., (1993), An Investigation of Due-Date Assignment Using Workload Information of a Dynamic Shop, International. Journal of Production Economics, Vol. 29, pp 89-101.

84. White, C. (1986), Capacity Constrained Load Sequencing Techniques for Production Management, Unpublished PhD thesis, Monash University, Australia.

85. White, C., and Hastings, N.A.J., (1983), Scheduling techniques for medium scale industry, Australian society for operations research bulletin 3, 1-4.

86. Zanakis, S.H. and J.R. Evans, (1981), Heuristic Optimization, Why, When, and How to use it, Interfaces.

Appendix 1: State of the Art

In chapter 2, a brief introduction to scheduling theory and research was presented. In this appendix, a detailed review of the literature is presented.

Introduction

Sequencing (scheduling) decisions are common. They exist whenever there is a choice as to the order in which a number of tasks can be performed by a limited number of resources. A problem could involve: jobs in a manufacturing plant, aircraft waiting for landing clearances, bank customers at a row of tellers' windows, programs to be run at a computer center, or just Saturday afternoon chores at home. Sequencing problems obviously gets solved, since most of the tasks are performed: aircraft land, the bank customers transact their business, and at least some of the Saturday chores are completed. However, often these problems are solved quite casually and automatically without the explicit recognition that a problem even existed, much less that a solution was obtained. Sometimes an ordering is determined essentially by chance; more often tasks are performed in the order in which they arise. An inherent sense of fair play has elevated the "first-come, first-served" solution of sequencing problems to an eminence out of all proportion to its basic virtue. It may be appropriate for patrons in a box-office line, but it need not necessarily be applied to inanimate jobs on a factory floor. There are problems in industry, transportation, and governmental and institutional activities in which the results of sequence are non-trivial and systematic consideration is worthwhile. Yet all too often, these problems are solved by default rather than design. The factory machine operator who decides which of the several waiting jobs to process next often uses criteria, which have little to do with the company's objectives. Programs may be run on a computer in the order submitted, a procedure that is undeniably fair, but far from optimal from anyone's' point of view.

The Job Shop Process

The general job shop problem is a fascinating challenge. This scheduling problem is in essence very simple to state: "*a set of 'n' jobs with different priorities have to pass through 'm' processes in such a way that some objective function(s) is (are) optimized.*" Although it is easy to state, and to visualize what is required, it is extremely difficult to make any progress whatsoever towards an optimal solution.

Almost all the theory that has been developed to date has been concerned with highly restrictive conditions that can be said to define a simple job shop process. Additional restrictions are placed on the definition of the job set and the machines, as well as on the manner in which a schedule may be constructed. Typically these are:

- each machine is available all the time,
- each operation can be performed by only one machine in the shop,
- setup time is sequence independent,
- overlapping of operations is not considered,
- lot - splitting is not considered by default, and
- each machine can handle only one job at a time.

These restrictions are primarily for simplicity of structure, but at the same time they increase the generality of the model. We should make one further very important restriction explicit by noting that the simple job shop process has only one limiting resource. We have assumed that an operation requires only one machine for its processing. In real life it may be that several machines, or a machine, an operator, and a tool, were simultaneously required to perform the process. Attributes would then have to be added to each operation to identify the particular type of operator and the type of tool required, and the assignment of an operation to a point in time could not be made until the appropriate machine, operator and tool were all simultaneously available.

A Classification of Scheduling Problems

It is usual to specify a scheduling problem by four types of information:

1. The jobs and operations to be processed.
2. The number and types of machines that comprise the shop.
3. Disciplines that restrict the manner in which assignments can be made.
4. The criteria by which a schedule will be evaluated.

Problems differ in the number of jobs that are to be processed, the manner in which the jobs arrive at the shop, and in the order in which the different machines appear in the operation of the individual jobs. The nature of the job arrival provides the distinction between static and dynamic problems. In the static problem a certain number of jobs arrive simultaneously in a shop that is idle and immediately available for work. No further jobs will arrive, so the attention can be focused on scheduling this completely known and available set of jobs. In a dynamic problem the shop is a continuous process. Jobs arrive intermittently, at times that are predictable only in a statistical sense, and arrivals will continue indefinitely into the future. The order in which the machine appears in the operation of individual jobs determines whether a shop is a flow-shop. A flow-shop is one in which all the jobs follow essentially the same path from one machine to another. At the opposite extreme is the randomly routed job shop, in which there is no common pattern of movement from machine to machine. Either of these extremes is undoubtedly rare in practice, with most real shops falling somewhere between these two limits, but almost all research in scheduling has assumed one or the other of these two extreme cases. Thus a four-parameter notation written as **A/B/C/D** is usually followed.

A - describes the job arrival process. For dynamic problems, A will identify the probability distribution of the times between arrival. For static problems, it will specify the number of jobs - assumed to arrive simultaneously unless otherwise stated. When '**n**' is given as the first term, it denotes an arbitrary, but finite, number of jobs in a static problem.

B - describes the number of machines in the shop. A second term of '**m**' denotes an arbitrary number of machines.

C - a system configuration/machine layout parameter that describes the flow pattern in the shop. The principal symbols are **F** for the flow shop limiting case, **R** for the randomly routed shop limiting case, **G** for a completely general or arbitrary flow pattern.

D - describes the performance criterion by which the schedule is to be evaluated. The performance criterion is an index used to select the better schedule for implementing a system when more than one schedule exists (completion-time based, due-date based, or flow-time based performance measures).

As an example, a $\mathbf{n/2/F/F_{max}}$ is a sequence of an arbitrary number of jobs in a two machine flow shop so as to minimize the maximum flow time (Johnson's problem) and a $\mathbf{n/m/G/F_{max}}$ is the general job shop problem - still unsolved - schedule '**n**' jobs in an arbitrary shop of '**m**' machines so that the last is finished as soon as possible.

Measures of Schedule Evaluation and Shop Performance

The variety of different criteria that have been employed in theoretical studies of scheduling partly reflects the variety of different circumstances in which interesting scheduling problems arise and the different costs and values that are relevant in each case. However, the choice of criteria has undeniably also been influenced by the prospect of obtaining a solution. In some models it has been possible to find optimal procedures only by departing from what would be considered the most natural and realistic criteria. Variables used in scheduling problems include waiting time of job, completion time of the job, flow time of the job, lateness/earliness and tardiness of the job. In almost all of the theoretical work on scheduling, very simple measures of performance have been employed. Broadly speaking, two different classes of performance measures can be identified, viz., time-based measures and economic-based measures. Time-based measures are the most common approach used and these have been the average or maximum of the values of completion time, flow time, lateness or tardiness.

It is sometimes convenient to evaluate a schedule with respect to measures that relate to the shop, rather than the individual jobs. The most obvious and important of these are facility utilization and work-in-process inventory. Utilization is a fraction of available machine capacity that is employed in the required processing, i.e. the ratio of processing time to the available time. It is assumed throughout that the total amount of processing time is predetermined and not affected by the scheduling decision, it can only be the denominator of the fraction, (i.e. the available time) that is of interest. In the continuous process model the machines are always available so that the average utilization is simply a given parameter of the problem and not affected by the schedule. It is true that the schedule determines just when, and in what pattern the idle time occurs in each machine, and a different pattern might have different utility - longer, connected intervals of idle time might be preferred to frequent short periods - but no work has been done on the problem with this preference as a criterion. Equipment utilization, WIP inventory and job flow-time are all interesting and more or less important, but the ability to fulfill delivery promises on time undoubtedly dominates these other consideration. The importance of this issue has led to many research studies in the past two decades (Udo, 1993).

Costs Associated with Scheduling Decisions

Although in practice the questions of when and in what order tasks are performed would have some effect on many of the costs of performance, and indeed, on whether the task is performed at all, the costs that can be directly associated with the limited question of pure sequence are restricted (Conway et al., 1967). The assumption that the set of tasks is determined before hand and that it is unaffected by scheduling decisions means that the total revenue of the enterprise is fixed, or at least unaffected by, and irrelevant to, scheduling. The assumption that the method and equipment to be used and the efficiency with which they will be employed are also unaffected by scheduling decisions mean that all of the costs that are normally classified as direct costs are irrelevant for our purposes. In fact, the costs that may be attributed to decisions of pure sequence are entirely what would be classified as facility costs rather than product costs. Even cost items such as overtime wage, premiums or penalties exacted for late deliveries that could be identified with a particular job are really consequences of decisions made on many other jobs and it would be unfair to assign them to the unfortunate jobs with which they are identified.

There are three principal types of costs that can be affected by the decisions of pure sequence. These are the costs of inventory, utilization and lateness. Much has been written on the costs of inventory. These costs are real, non-trivial (estimates of 2 % to 3 % of value per month are not unusual), in some way proportional to the physical level of inventory, and very difficult to quantify. In different circumstances, costs might be related to the number of jobs, work content, or work completed, but in general we can conclude that there are economic reasons for our interest in reducing average inventories. Facility utilization is a very important economic consequence of sequencing decisions. The ability to compact the busy intervals and produce a short schedule-time or a low mean flow-time simply implies a procedure that will permit a given facility to do more work. Conversely, an efficient scheduling procedure will permit a given work load to be accomplished with a smaller aggregate demand on facilities. In the long run, this will be reflected in either the amount of plant required or the amount of business that can be accommodated. In the short run, it is reflected in the costs of overtime, additional shift operation, and overload sub-contracts. In some situations, especially steel fabrication projects with high penalty costs, the costs of lateness are obvious, explicit, and unequivocal. For example, a penalty of X dollars per day may be deducted from payment for each day that completion is delayed beyond a specified due-date. In manufacturing situations the penalty for lateness is seldom this obvious and immediate but presumably no less real, since the customer's displeasure will have implications for future business. Conway et al. (1967) suggest the relative importance of this consideration by stating that one scheduler stated to them that he might be admonished for high inventories, but that he would be fired for excessive lateness.

Whether one considers the limited questions of pure sequence or a broader and more realistic view of scheduling, it is still true that the costs and values are difficult to identify and harder still to measure. A modern costing system includes procedures specifically designed to call attention to inefficient use of labor or material or to indirect items that exceed budget allowances, but there are no accounts or procedures that clearly signal inefficiency in scheduling. The costs are nonetheless real, and judging from the demonstrated differences in performance between alternative scheduling procedures there must be some situations in which this choice is vital.

Types of Schedule Generation Methods

It is clear that there are many schedules for any job shop problem, since idle times can be inserted into any given schedule in infinitely many ways. Even for relatively small problems, the number of possible schedules is still very large so that solution by exhaustive enumeration and comparative evaluation is not feasible. Many of the heuristic, sampling and computational approaches to the job shop problem require the generation of a number of schedules for a particular problem. Schedule-generation procedures have a basic similarity in that each operates on the set of (G) operations, selecting operations one at a time and assigning a starting time to each. The order in which the operations are selected and the manner in which the starting time is determined characterize a schedule-generation procedure. The most important distinction is between single-pass and adjusting procedure. Under single-pass procedure, once a starting time is assigned to a particular operation, this time is permanent and may not be changed to accommodate a later assignment. However involved the decision rule may be that selects the operation to be assigned and its starting time, there is just one pass through the set of operations and precisely G starting time decisions are made. Under an adjusting procedure, each starting time assignment is tentative and subject to repeated modification until the entire schedule has been completed. One might initially suppose that the restriction placed on single-pass procedures would make some schedules inaccessible to this type of generation, so that adjusting procedures would be theoretically preferable. Strictly speaking, this is not the case, since for any schedule at all there is a corresponding single-pass procedure capable of producing it. A special type of single-pass procedure that has been particularly important in research and in actual industrial practice is the class of dispatching procedures. These are single-pass procedures in which the starting times for any given machine are determined in such an order that they form a strictly non-decreasing sequence of numbers. This means that the decisions are made in the same order as they will be implemented, and that the scheduling process can be spread out in time, making each decision immediately before it is to be implemented. Another type of schedule generation followed in practice is the job-at-a-time or list schedule generation procedure. It schedules all the operations of the job consecutively without any intervening decisions on any operations of any other job.

In most cases, jobs are not all equally important. The concept of job or operation priority is inherent in many schedule generation procedures. A priority is simply a numerical attribute of a job or operation on which selection is based. For example, in job-at-a-time schedule generation there must be some mechanism for determining the order in which the jobs are selected and it is convenient to view this in terms of a priority system; jobs are always selected in increasing value of a numerical attribute assigned to each job. Then there is really only one job at a time schedule generation procedure, but there are n! different ways of assigning priority to the jobs. A priority system must always have sufficient precision to lead to a unique selection so that, two competing jobs should never have precisely the same value of priority. This may require that in support of the primary-priority attribute there may have to be secondary attributes assumed in order to resolve ties.

A class structure can be superimposed on the simple sequencing rules by requiring that a certain subset of the jobs be placed first, in some order; the jobs of another subset next, in some order; etc. A discipline of this type has three components: (1) a rule for partitioning the 'n' jobs into distinct, mutually exhaustive subsets called classes, (2) a rule for specifying sequence among the classes, and (3) a rule or rules for specifying the sequence of jobs within each class. For example, a manufacturer might have some jobs for specific customer orders and some jobs for stock replenishment. He might decide to do the customer jobs first say in early due date (EDD) order, and then the stock jobs in shortest processing time (SPT) order.

Theory of NP - Completeness

The theory of NP completeness provides a criterion to classify problems into classes of P or NP (P is the abbreviation of a deterministic polynomial time algorithm and NP denotes a non-deterministic polynomial time algorithm). Completeness means that the algorithm that solves a particular problem will also solve other problems in the same class; in this sense, completeness means the possibility of sharing the same algorithm to solve all problems of a given class.

The milestone of the development of scheduling theory is the emergence of the concept of NP completeness. Through this concept, an intractable scheduling problem can be shown to be too hard to solve. It means that a polynomial time algorithm for this scheduling problem is very unlikely to exist. This problem belongs to the class of NP complete problems. The NP complete problems are regarded as hard problems. P problems are relatively easy problems. The development of this theory provides a guideline on what type of results can be expected on what type of scheduling problems. For the study of NP complete problems, the direction of the study should be focused on developing exponential algorithms and heuristic rules. The trend of scheduling research has been either to find a polynomial algorithm for a tractable problem, or to show that the problem is intractable. In the latter case attention will be directed toward the design of an approximation algorithm to search for the sub optimal schedule or the design of an exponential time algorithm to search for the optimal schedule.

Many scheduling problems have been shown to be NP complete (Bruno et al 1974; Garey and Johnson 1979; Karp 1972,1975; Lenstra and Rinnooy Kan 1979,1984). Job shop scheduling is among the hardest combinatorial optimization problems. Not only is it NP-Complete, but also a member of the latter class it belongs to the worst in practice. It is therefore natural to look for approximation methods that produce an acceptable schedule in a reasonable time. Thus, heuristic approaches are far more preferable than simulation approaches (Jiang, 1991).

Survey of Scheduling Approaches

From the point of view of scheduling research, the nature of the problem studied can be categorized in a number of ways, viz. (1) number of machines - single machine case, two-machine case, multiple machine case, (2) finite capacity or infinite capacity - and resource constraints, (3) nature of the shop - flow shop (all process follow same route), again open or closed, generalized job shop, (4) nature of jobs and job related information processing -deterministic, static, stochastic, or dynamic., and (5) performance criterion - minimize tardiness, minimize mean flow time, or some other criteria.

Many studies have shown that the more complex the shop structure, the poorer its performance. The complexity of shop structure is usually measured by the number of machines, or number of machines and number of operators. That is to say a two-machine shop is more complex in structure than a single-machine shop. Cheng and Gupta (1989) surveyed scheduling research studies involving due-date assignments and found that very little research has been done on multiple machine scheduling problems. According to them, the main reason many researchers prefer to study single machine shops is that the single machine scheduling problems are less complex and more convenient than multi-machine problems. A survey of 40 practicing schedulers conducted by McKay, Safeyeni and Buzacott (1988) reveals that one of the reasons scheduling research has no impact on real-world situations is the fact that most scheduling studies simulate unrealistic environments and make impractical assumptions.

There is an increased call for multi-machine shop scheduling studies in recent times for the following reasons: (i) most real world shops do not operate with a single machine, (ii) results of single machine shops are not guaranteed in multiple machine scenario, and (iii) work stations in modern shops are integrated and not isolated as portrayed in single machine studies. Multiple machines scheduling theory is the study of constructing schedules of machine processing for a set of jobs in order to ensure the execution of all the jobs in the set in a reasonable amount of time. The major concern of multiple machine scheduling theory is how to provide a perfect match or near perfect match of machines and jobs and subsequently determine the processing sequence of the jobs on each machine in order to achieve some prescribed goal. Until recently only a few multiple machines scheduling problems have been solved or shown to be tractable, such as the two-machine flow shop problem. Yet the only way to find the optimal schedules of other unsolved scheduling problems is by implicit enumeration.

The class of parallel machine scheduling problems has been a subject of extensive study by computer scientists for a long time because scheduling incoming jobs on parallel processors presents a major operational problem for running a timesharing computer system (Coffman, 1976). The same problem is also encountered in a machine shop where job orders are to be scheduled on groups of identical production facilities. A variety of sequencing and scheduling research and an extensive treatment of the problems of machine scheduling can be found in the works of Conway et al. (1967), Baker (1974), Coffman et al. (1976), Rinnooy Kan (1976), Lenstra (1977), Panwalkar and Iskander (1977), Lawler et al. (1982), and French (1982). Job shop and Flow shop problems are among the most attractive to researchers. One significant result of this research has been to clarify the complexity of scheduling problems. Lenstra (1976), Rinnooy Kan (1976), and Coffman (1976) discuss the complexity results of typical scheduling problems. Beyond two machines the job shop problem is inherently intractable in the sense that the existence of a polynomially bounded optimization algorithm is very unlikely. The difficulties of finding an optimal solution are entirely computational. Existing optimum seeking approaches can only solve problems of the order of 10 jobs on 10 machines (Lawler et al. 1982).

The character and content of scheduling research has tried to keep pace with the theoretical developments in operations research and computer science areas. Scheduling research adopted mathematical programming techniques such as linear programming, integer programming (Manne, 1960), and later multiple objective programming (Ruiz and French 1983). The need for practical implementations took research towards implementing branch and bound techniques (Conway 1967) (Baker 1974), and heuristic rules with simulation (Gere, 1966) (Conway 1967). The adoption of simulation led to the use of queuing networks (Rajasekara et al., 1991).

Apart from the developmental aspects of computer hardware and software, the primary impact of computer technology on scheduling was the development of the complexity theory - the theory of NP completeness (Garey and Johnson 1979). This theory led to a cleaner classification of scheduling problems on the basis of solution feasibility (Blazevicz, 1987) (Cheng and Sin, 1990). Developments in artificial intelligence (AI) and expert systems technologies paved the way for a host of new approaches to tackling the scheduling problem (Fox 1987) (Randhawa and McDowell, 1990). AI led to distributed AI, where solutions to scheduling problems are explored using intelligent problem solving agents distributed across the factory. As discussed in chapter 2, we can group scheduling approaches into three broad categories, viz., the conventional, knowledge based and distributed approaches as shown below in Figure A1-1.

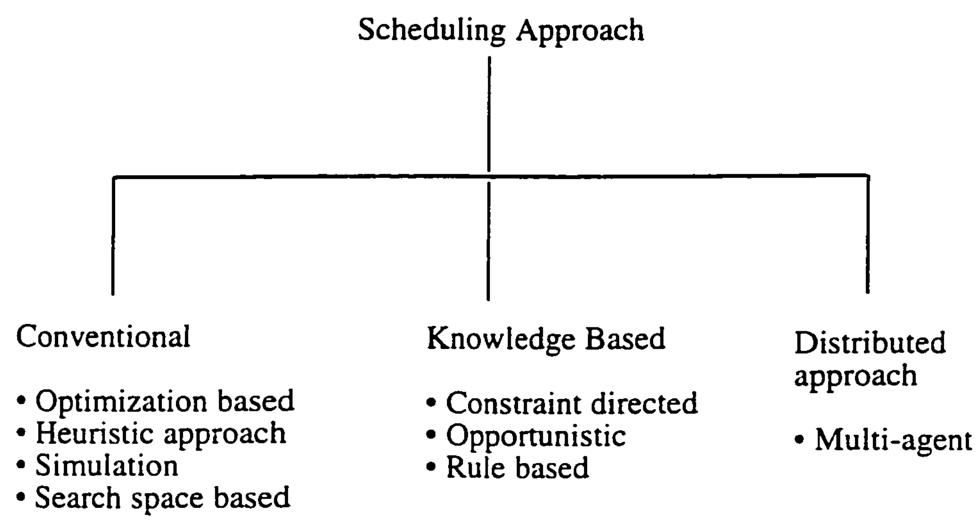


Figure A1-1: Approaches for Scheduling

Conventional Approach

The scope for mathematical programming is limited because of the complexity of the scheduling environment. Although many of the constraints can be represented mathematically, soon, with only a few of the situations considered, the problem becomes intractable. Hence much of the research using these methods involves highly simplified versions of the actual problem (King, 1976). All deterministic scheduling problems are combinatorial optimization problems. There are four major classes of solution methods for combinatorial optimization problems, viz., complete enumeration, exponential time algorithms, polynomial time algorithms and approximation algorithms. Each class of solution method yields a different level of efficiency and accuracy.

One common way of solving scheduling problems is to determine a lower bound on the cost and then use a branch and bound method to determine the optimal solution. Gupta and Sen (1983) present improvements to branch and bound algorithms by using the concept of entrapment for single machine problems. Unfortunately, the branch and bound method has exponential computation time in the worst case. Additionally the problem must be solved whenever changes occur in the system - machine breakdown, jobs requiring more time than anticipated or some other change occuring during the course of the planning horizon.

One useful technique for scheduling problems has been heuristics. Heuristics have the advantage over branch and bound in terms of efficiency, the advantage over polynomial time approximation in terms of widespread applicability. In general, however, heuristics offer no guarantee that the solution is within an acceptable margin of error when compared to the optimal solution. A heuristic approach usually involves a priority dispatch rule, such as the shortest processing time, which determines the next job to be processed on a machine, or a complex rule combining several dispatching rules. There are many heuristic-dispatching rules, with over 100 listed by Panwalkar (1977). However, based on the results of past scheduling research, no dispatching rule dominates all other rules with regard to all criteria. Normally, a simulation model of the shop floor supports the heuristic approach. The main drawback of the heuristic approach is that the decisions tend to be myopic.

Heuristics for Industrial Scheduling

Two classes of heuristic procedures can be distinguished in scheduling research: operations oriented and job oriented. In operation oriented heuristics, scheduling decisions are made at each machine, by selecting the next operation from a queue of operations that are awaiting processing when the machine becomes idle or available. Based on a dispatching rule, priorities are assigned to the operations in the queue and one is selected from the set of schedulable operations based on the priority. Quite a few dispatching rules appear in the research literature. Panwalkar and Iskander (1977), Blackstone et al. (1982) give an extensive investigation of dispatching rules on the basis of different classification schemes. The evaluation of dispatching rules constitutes the substantial body of job shop scheduling research. The most common approach is to conduct experiments with a hypothetical shop via simulation. One consistent result of the simulation studies is that the shortest processing time (SPT) rule is the best one for minimizing throughput time for a set of jobs. However, in actual industrial settings, the meeting of due dates is more important than minimizing throughput time of a set of jobs (Panwalkar et al. 1977). No single priority rule dominates performance comparison in due date scheduling (Baker 1984).

Instead of working with individual operations by dispatching rules, Job oriented heuristics (JOH) schedule one job at a time, so that all the operations of the job are scheduled before the next job is considered. Very little research has been reported on this class of heuristics (Hastings and Yeh, 1990). The book by Magee and Boodman (1967) proposes a JOH procedure for loading jobs onto machines. Conway et al. (1967) reports another JOH procedure from a research report by T.B. Crabill at Cornell University. This procedure attempts to fit operations of a job into the best position in an existing schedule so as to minimize flow-time. To achieve this, the schedule times of previously scheduled operations may be reassigned after subsequent operations are inserted into the schedule. A single pass JOH procedure designated forward scheduling is proposed in the paper by

Hastings et al. (1982). Jobs are scheduled from the start date of the schedule one by one. The operations of a job are fitted in succession onto the earliest available time slots on the corresponding machines. The objective of forward scheduling is to complete the jobs as early as possible. White and Hastings (1983) extend this procedure to backward scheduling, which schedules jobs one by one backwards from their due dates. The operations of a job are fitted in succession onto the latest available time slots on the corresponding machines, starting from the last operation. Jobs with no due dates are scheduled forwards. If, when a job is scheduled backwards, its start time would be in the past in order for it to meet its due date, then the job is rescheduled using forward scheduling. In this case, the job will finish after its due date, but in every case the schedule will be feasible. Backward scheduling aims to complete the jobs on or close to their due dates.

White (1986) has conducted an extensive experiment designed to examine the overall performance comparison between JOH and operation oriented heuristics. His finding show that JOH is generally superior to the six commonly used dispatching rules in terms of overall quality of the schedule. Another finding reported in his study is that JOH consumes far less computational resources than the equivalent dispatching rules. No evidence indicates that JOH should be eliminated from consideration in either scheduling research or scheduling practice. Indeed from the research cited the following conclusions regarding JOH can be stated:

- They are computationally efficient.
- They produce good quality of schedules, as confirmed by experimental testing conducted by white (1986).
- They are intelligible to production personnel, in that they facilitate the control of individual jobs.
- They are well suited to situations involving job precedence constraints arising from products in which components and assemblies have to be made.

- They can easily accommodate changes in job status, due to changing priorities, rework or shortage of materials.

It is fair to state that the advantageous features of JOH suggest that it still remains the most promising approach to industrial scheduling even at the present time.

Search based systems attempt to reduce the search space for possible solutions, as well as the time to search the space (O'Grady 1985).

Due-date based Studies

The ability to meet pre-assigned due dates is of greatest interest to companies faced with real life scheduling problems. Shop managers strive to set accurate due dates (predictability) that are attainable (controllability). The importance of this issue has led to many research studies in the past two decades. Research in the area of due date assignment tends to fall into two classes. One group assumes that ready times for jobs are given, so that the research effort is directed towards determining the flow allowance in order to set the due dates. The other group of studies assumes that due dates are given, so that the research task becomes that of determining the lead time in order to set ready times for the job. For either group of studies, the method of setting due dates can be dynamic or static. The dynamic method employs job characteristics (content) and job shop congestion information in determining due dates. The static method on the other hand, considers only the job content information such as arrival times, routing, and processing-times. Many studies have consistently concluded that assigning due dates based on job content and shop congestion information can lead to better shop performance than assigning due dates based only on job content

Flow-Shop Studies

The study of flow shop scheduling has attracted considerable interest among researchers over many years. Taillard (1990) reports that even though many researchers have been working on the flow shop-sequencing problem for many years, there has been no published result about the distribution of the objective function and the distribution of the optima of this function. The flow shop-scheduling problem consists of arriving at a schedule so that some performance measure is optimized. Some of these measures include the mean flow time, mean tardiness, maximum tardiness, number of tardy jobs, and most commonly the makespan. In general, an optimal schedule need not have the same sequence of jobs on each machine. However the computational effort involved in evaluating such schedules is prohibitive. In general $[n!]^{m-2}$ schedules have to be evaluated to obtain an optimal solution. Lenstra et al. (1979) have shown that this is not computationally feasible for most moderate sized problems. Over the years a number of heuristics have been formulated for sequencing jobs in a flow shop. For a setup where the number of jobs and machines are not very small there is usually a tradeoff between the quality of the sequence and the computational effort required in arriving at that sequence. The criterion of minimizing total flow-time has been found to reduce the scheduling costs significantly. Moreover, it has been found to be an important real-life objective in industries since it results in the even utilization of resources, even turn-over of finished jobs and reduced in-process inventory (Baker 1974) (French 1982).

In many flow shops, there exists a constraint that once the processing of a job begins, subsequent processing must be carried out with no delay in the passage of the job from machine to machine. Such a flow shop can be termed a 'constrained flow shop' or 'no-wait flow shop'. Some typical situations are encountered in chemical processing, metal processing and hot-rolling industries.

Simulation is the most widely used technique to investigate dynamic scheduling (French 1982). In one of the early papers dealing with dynamic scheduling, Muth, investigated the effect of uncertainty in job times on optimal schedules. The paper discusses a simulation study of a flow shop with job processing times having bi-variate lognormal probability distribution. The study concludes that the completion time of the schedule is not very sensitive to moderately large errors in estimated job times. Conway et al. (1967) also discusses job shop scheduling under stochastic processing times. They derived equations for expected value of flow times and also gave an expression for the expected value of the contribution of a job to the expected value of the mean flow time. Moodie and Roberts (1968) consider a parallel processor shop, where input of jobs is distributed over time. A simulation study is made to evaluate the performance of various priority dispatching rules such as SPT, Slack rule, or EDD. A new weighted objective rule is proposed, which combines several dispatching rules to derive a priority index for a job on a particular machine. The results reported indicate the consistency of the rule under changing shop loads. Gittins (1979) developed a dynamic allocation index for each job and scheduled the job in decreasing order of this index. As jobs are processed these indices are updated, thus allowing the schedule to adapt to the pattern of arrivals and actual processing times. Muhelmann et al (1982) studied the performance of a number of heuristics under different scheduling frequencies (weekly, biweekly, or monthly). Other aspects considered include uncertainty of process times and machine breakdowns. The results show that it is difficult to come up with a "best" heuristic approach. Ow (1985) describes a focused scheduling method based on an idle time rule for a proportionate floor shop where bottleneck conditions were dynamic. A priority is worked out for all jobs at a given time and the job with the highest priority is scheduled on the machine. The priority function involves a weight being attached to a job that is late, its processing time, due date and start time. Two parameters are used - a resource parameter which reflects the opportunity cost of a time unit on an alternate machine, and a look ahead parameter which attempts to increase the span of consideration for the priority computation beyond the immediate jobs. Onur and Fabrycky (1987) proposed a control system with feed forward and feedback loops for controlling a dynamic job shop. A composite cost function is minimized to determine periodically the set of jobs to be released. The problem is formulated as a 0 - 1 linear mixed integer programming problem and an interactive heuristic approach is proposed. Bitran and Tirupathi (1988) model a semiconductor wafer production facility as a single stage parallel machine and consider a case where jobs are released as and when orders are received, and the process time includes a stochastic test component. A two-phase algorithm is suggested for tardiness criteria and they also propose a different approach using the make span criteria. Ramesh and Carey (1989) suggest a multi criteria approach to job shop scheduling. They have proposed three different algorithms, which combine process time based rules, due date-based rules and dynamic shop floor conditions to evolve dynamic scheduling strategies.

Knowledge based Approach

The fundamental characteristic of a knowledge-based approach is the manipulation of knowledge encoded about a particular domain of problem solving, say, scheduling. There are three major approaches to expert system based solutions: rule-based, pattern directed and a combination of the two (Newman 1988). Newman discusses various schedule generation techniques such as hierarchical, non-hierarchical, and opportunistic. A detailed survey of knowledge based systems can be found in Atabaksh (1991).

ISIS, a system developed at Carnegie Mellon is one of the first AI based scheduling systems. A wide variety of constraints can be represented using this system. ISIS performs a constraint directed search to derive a schedule. It also has a constraint relaxation component, which is used when conflict arises. The dynamic situations are handled by the rescheduling component, which reschedules the affected orders by selectively relaxing some of the constraints. Priority grouping of orders is also used for preempting reservations of orders of lower priority when situations arise. OPIS, a successor of ISIS (Smith and Hynyen, 1987) is based on the idea that multiple scheduling perspectives broaden the range of conflicts that can be addressed effectively. The organization of OPIS is a variation of the HEARSAY II blackboard style architecture (Erman et al. , 1980). A dynamic control policy incorporating resource based scheduling or an order based scheduling strategy is used depending upon the situation. Reactive management of the schedule in response to unanticipated status in factory status is a special case of OPIS scheduling methodology. SONIA is a job scheduling system, which consists of predictive and reactive scheduling components (Collinot 1988). The reactive component is used to solve inconsistencies arising from deviations in the schedule. Delays, capacity conflicts and breakdowns are considered. OPAL is another system designed for job scheduling (Bensana and Dubois , 1988). OPAL uses production rules and heuristics to determine precedence relations between operations. Although OPAL is designed to construct only predictive schedules, if the schedules are delayed, the system may be rerun with the current status as an input to generate fresh schedules. ICCS, an Intelligent Cell Control System (O'Grady and Kwan ,1988), has been implemented using a multi-blackboard, actor based framework. There are four blackboard subsystems: a scheduling blackboard which schedules resources within the cell so as to achieve the goals, an operation dispatching blackboard which generates detailed operations sequences, a monitoring blackboard which filters and classifies the feedback and an error handling blackboard which recognizes and analyzes the errors and problems and provides possible corrective solutions.

Jiang (1991) describes an Intelligent Scheduler (IS) which takes into account multiple machines, multiple fixtures and a variety of dispatching rules that the user may select for scheduling. He develops the knowledge base through an experimental design consisting of 1120 sample runs. The significant factors are dispatching rules and the interaction of the dispatching rules with the number of jobs while considering a single dependent factor. When investigating multiple dependent factors, dispatching rules interactions with number of jobs is a significant factor. This shows that the level of shop loading and using the proper dispatching rule will affect the performance of the schedule. However, different sequences of dispatching rules will result from different shop configurations. So, no one dispatching rule can serve as a good performance measure in practice.

As the size of the problem grows, expert systems take large amounts of computer time to solve the problem. This is because the number of possible combinations available for the given problem increases tremendously. The main obstacle in using the expert systems solution is the number of choices available for the given problem.

Distributed Problem Solving Approach

Distributed problem solving is the cooperative solution of problems by a set of decentralized and loosely connected intelligent problem solving agents. These agents are processors with local procedures, and/or rules, which can be applied to the problem. They need to cooperate, as none of the agents possess the information necessary to solve the global problem. Speed, reliability, extensibility and fault-tolerance are some of the fundamental characteristics of distributed problem solvers (Smith, 1980). Application of these techniques to scheduling is of recent origin.

Shaw (1987), describes a distributed planning method for cellular flexible manufacturing systems (CFMS). The system consists of loosely coupled flexible cells, the autonomous cell host (computer) for planning and control uses a local area network for communication. An augmented Petri-net model is used to describe the task negotiation protocol by bidding. The award of the bid to a cell is made using the negotiation protocol (Smith, 1980). YAMS (Parunak 1988) is yet another distributed scheduling system. Here scheduling is done hierarchically by negotiations by the nodes at that level for distributing the tasks. A contrast-net model is the basis for negotiation. Heuristic techniques have been built to meet real time constraints and decide whether the task can be guaranteed.

Observations

It is clear from the review of literature that attempting to solve the real life scheduling problem is a worthwhile endeavor, especially in the construction industry where virtually every thing is made to order and where no 'economies of scale' type of approaches can be applied. Scheduling in this environment is a very knowledge intensive activity requiring a vast array of data. A DSS for scheduling is an appropriate approach for solving the problem. The general scheduling problem cannot be easily handled by optimization methods, but heuristic approaches can form the first steps in our understanding of the complexities of the shop floor.

Appendix –2: List of Fields in DSS database

The tables in the database are shown in Figure A2-1.

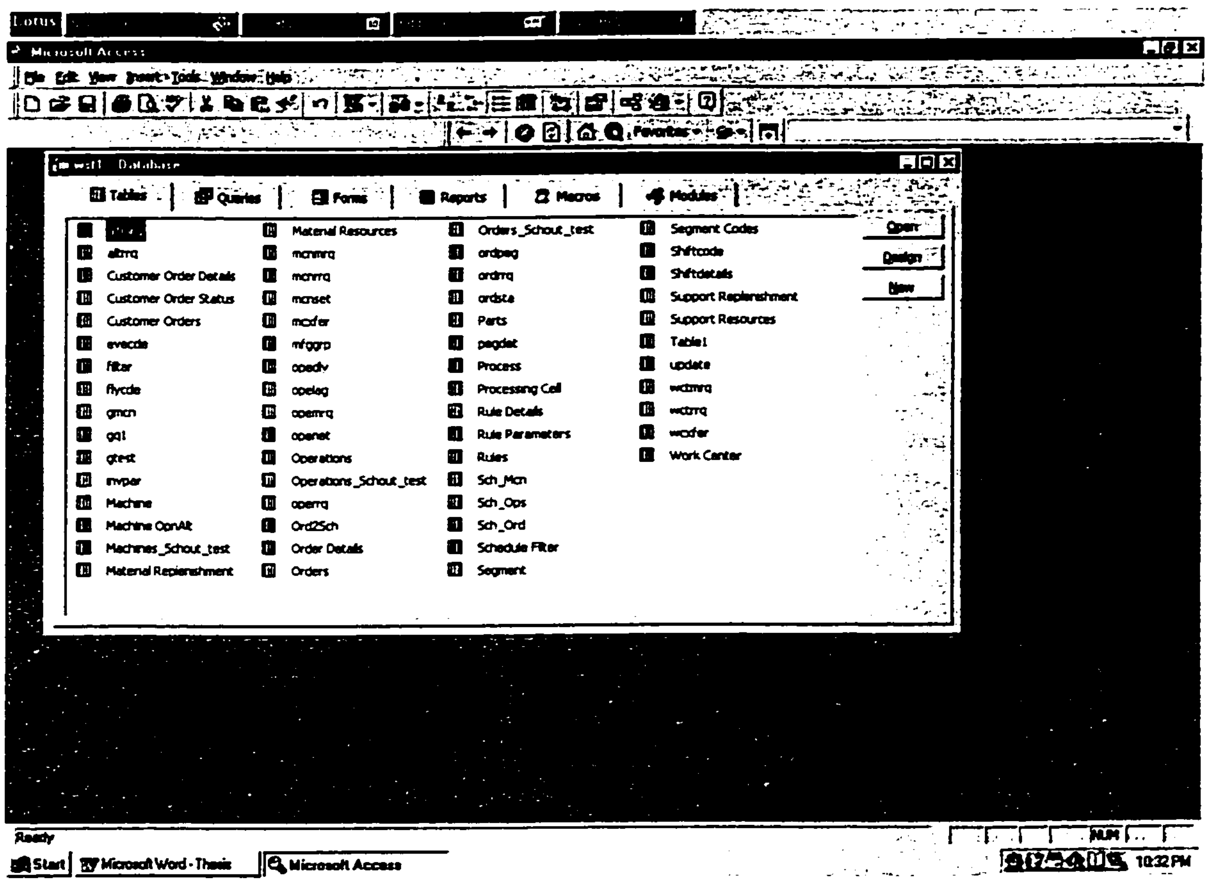


Figure A2-1: Database Tables Information

The following pages give the details of the fields in the various tables.

C:\My Documents\testdata.mdb

Table: Customer Order Details

Columns

Name	Type	Size
ORDER	Text	25
OPERATION	Number (Integer)	2
ORD_STDT	Date/Time	8
RANK	Number (Long)	4
ORIGINAL_QUANTITY	Number (Double)	8
REVISED_QUANTITY	Number (Double)	8
RELEASE_DATE	Date/Time	8
DUE_DATE	Date/Time	8
PRIORITY	Text	1
CLASS	Text	3
START_AFTERDATE	Date/Time	8
START_AFTERTIME	Date/Time	8
REQUIRED_ORDQTY	Number (Double)	8
EFFECTIVE_PIECES	Number (Double)	8
TRANSFER_LOT	Number (Double)	8
UDF1	Text	10
UDF2	Text	10
UDF3	Text	10
UDF4	Text	10
UDF5	Text	10
UDF6	Text	10
Notes	Memo	-

C:\My Documents\testdata.mdb

Table: Customer Order Status

Columns

Name	Type	Size
ORDER	Text	25
OPERATION	Number (Integer)	2
COMPLETE	Yes/No	1
SEGMENT	Text	6
LASTSEGMENT	Text	6
MACHINE	Text	10
ORIGINAL_QUANTITY	Number (Integer)	2
COMPLETED_QUANTITTY	Number (Integer)	2
ACTUAL_START_DATE	Date/Time	8
ACTUAL_START_TIME	Date/Time	8
ACTUAL_FNISHI_DATE	Date/Time	8
ACTUAL_FINISH_TIME	Date/Time	8

C:\My Documents\testdata.mdb

Table: Customer Orders

Columns

Name	Type	Size
ORDER	Text	25
PART	Text	25
CUSTOMER	Text	20

CUSTOMER_ORDERNO	Text	20
SUSPEND	Yes/No	1
COMPLETE	Yes/No	1
ORDER_VALUE	Number (Double)	8
ORDER_COST	Number (Double)	8
Notes	Memo	-

C:\My Documents\testdata.mdb
Table: event code

Columns

Name	Type	Size
CODE	Number (Double)	8
NAME	Text	30

C:\My Documents\testdata.mdb
Table: filter setup

Columns

Name	Type	Size
SCHEDULE_SCREEN	Text	25
EXPRESSION	Text	200

C:\My Documents\testdata.mdb
Table: Machine

Columns

Name	Type	Size
Work _Center	Text	6
Machine	Text	6
Name	Text	30
Internal	Yes/No	1
Shift_Code	Text	4
Cost	Currency	8
Notes	Memo	-

C:\My Documents\testdata.mdb
Table: Machine Alterantives

Columns

Name	Type	Size
PROCESS	Text	25
OPERATION	Number (Integer)	2
WORK_CENTER	Text	10
MACHINE	Text	10
DESC	Text	25
PRIMARY	Yes/No	1
ACTIVE	Yes/No	1
Notes	Memo	-

C:\My Documents\testdata.mdb
Table: Material Replenishment

Columns

Name	Type	Size
RESOURCE	Text	10
QUANTITY	Number (Double)	8
LOGIC	Text	1
DATE_FROM	Date/Time	8
TIME_FROM	Date/Time	8
DATE_TO	Date/Time	8
TIME_TO	Date/Time	8

C:\My Documents\testdata.mdb
Table: Material Resources

Columns

Name	Type	Size
RESOURCE	Text	10
DESC	Text	30
SHIFT_CODE	Text	6
CRITICAL	Yes/No	1
AUTO_REPLENISH	Yes/No	1
RESERVE	Yes/No	1
TOLERANCE	Number (Double)	8
DISCRETE	Yes/No	1
CONSIDER	Yes/No	1
QUANTITY_ONHAND	Number (Double)	8
DATE_ONHAND	Date/Time	8
TIME_ONHAND	Date/Time	8
UNITS	Text	5
UNIT_COST	Number (Double)	8
MIN_LEVEL	Number (Double)	8
MAX_LEVEL	Number (Double)	8
LOCATION	Text	30
SUPPLIER	Text	30

C:\My Documents\testdata.mdb
Table: Operations

Columns

Name	Type	Size
PROCESS	Text	25
OPERATION	Number (Integer)	2
DESC	Text	10
ACTIVE	Yes/No	1
Notes	Memo	-

C:\My Documents\testdata.mdb
Table: Ord2Sch

Columns

Name	Type	Size
ORDER	Text	25
OPERATION	Number (Integer)	2
PART	Text	25
PROCESS	Text	25
ORDER_STARTDATE	Date/Time	8
CUSTOMER	Text	20
CUSTOMER_ORDER	Text	20
SUSPEND	Yes/No	1
COMPLETE	Yes/No	1
ORDER_VALUE	Number (Double)	8
RANK	Number (Long)	4
OPCOUNT	Number (Double)	8
CRATIO	Number (Double)	8
CELL	Text	10
REL_DATE	Date/Time	8
DUE_DATE	Date/Time	8
PRIORITY	Text	1
CLASS	Text	3
START_AFTERDATE	Date/Time	8
START_AFTERTIME	Date/Time	8
REQUIRED_QUANTITY	Number (Double)	8
EFFECTIVE_PIECES	Number (Double)	8
TRANSFER_LOT	Number (Double)	8
UDF1	Text	10
UDF2	Text	10
UDF3	Text	10
UDF4	Text	10
UDF5	Text	10
UDF6	Text	10

C:\My Documents\testdata.mdb
Table: order status

Columns

Name	Type	Size
ORDER	Text	18
OPERATION	Number (Double)	8
COMPLETE	Yes/No	1
SEGMENT	Text	5
LASTSEG	Text	5
MACHINE	Text	9
REVISED_QUANTITY	Number (Double)	8
COMPQUANTITY	Number (Double)	8
COMPPERCENT	Number (Double)	8
COMPTIME	Number (Double)	8
OPERATION_ACTSTDT	Date/Time	8
OPERATION_ACTSTTI	Number (Double)	8
OPERATION_ACTFIDT	Date/Time	8
OPERATION_ACTFITI	Number (Double)	8

C:\My Documents\testdata.mdb
Table: Parts

Columns

Name	Type	Size
PART	Text	25
PROCESS	Text	25
NAME	Text	30
DRAWING_NO	Text	20
REVISION_NO	Text	6
REVISION_DATE	Date/Time	8
UNITS	Text	10
LABOR_COST	Number (Double)	8
MATERIAL_COST	Number (Double)	8
BUDGET_CENTER	Text	5
PURCHASED	Yes/No	1
NOTES	Memo	-

C:\My Documents\testdata.mdb
Table: Process

Columns

Name	Type	Size
PROCESS	Text	25
NAME	Text	30
DRAWING_NO	Text	20
REVISION_NO	Text	6
REVISION_DATE	Date/Time	8
ACTIVE	Yes/No	1
NOTES	Memo	-

C:\My Documents\testdata.mdb
Table: Rule Parameters

Columns

Name	Type	Size
PARAMETER	Text	25
PARMETER_NO	Number (Byte)	1
NOTES	Memo	-

C:\My Documents\testdata.mdb
Table: Sch_Mcn

Columns

Name	Type	Size
MACHINE	Text	9
DATE	Date/Time	8
TIME	Number (Double)	8
DURATION	Number (Double)	8
ORDER	Text	18
OPERATION	Number (Double)	8

SEGMENT	Text	5
REASON	Text	2
EVENT_CODE	Number (Double)	8

C:\My Documents\testdata.mdb
Table: Sch_Ops

Columns

Name	Type	Size
ORDER	Text	18
OPERATION	Number (Double)	8
OPERACT_FINDT	Date/Time	8
COMPLETE	Yes/No	1
LOT_WORKCENTER	Text	9
LOT_MACHINE	Text	9
INTERNAL	Yes/No	1
LOT_STARTDATE	Date/Time	8
LOT_STARTTIME	Number (Double)	8
LOT_FINISHDATE	Date/Time	8
LOT_FINISHTIME	Number (Double)	8
LOT_DURATION	Number (Double)	8
ACT_LOTS	Number (Double)	8
LOT_QTY	Number (Double)	8
ORDER_PREV	Text	18
OPERATION_PREV	Number (Double)	8
MACHINE_PREV	Text	9
EARLY_DAYS	Number (Double)	8
LATE_DAYS	Number (Double)	8
LOT_SCHED	Yes/No	1
INPROCESS	Yes/No	1
LOCKED	Yes/No	1
CLASS	Text	3
PRIORITY	Text	1
DUE_DATE	Date/Time	8
START_AFTERDATE	Date/Time	8
START_AFTERTIME	Number (Double)	8

C:\My Documents\testdata.mdb
Table: Sch_Ord

Columns

Name	Type	Size
ORDER	Text	18
RANK	Number (Double)	8
ORDER_STARTDATE	Date/Time	8
ORDER_STARTTIME	Number (Double)	8
ORDER_FINISHDATE	Date/Time	8
ORDER_FINISHTIME	Number (Double)	8
SUSPENDED	Yes/No	1
COMPLETED	Yes/No	1

C:\My Documents\testdata.mdb
Table: Schedule Filter

Columns

Name	Type	Size
FILTER_CODE	Text	10
Text	Text	200

C:\My Documents\testdata.mdb
Table: Segment

Columns

Name	Type	Size
PROCESS	Text	25
OPERATION	Number (Integer)	2
WORK_CENTER	Text	10
MACHINE	Text	10
SEGMENT	Text	6
PROC_ORDER	Number (Byte)	1
RATE	Number (Double)	8
OVERLOAD_PERCENT	Number (Double)	8
OVERLOAD_TIME	Number (Double)	8
MAXMCNDLY	Number (Double)	8
MAXSEGDLY	Number (Double)	8
MAXMCNGAP	Number (Double)	8
MAXSEGGAP	Number (Double)	8
Notes	Memo	-

C:\My Documents\testdata.mdb
Table: Segment Codes

Columns

Name	Type	Size
PROCESSING_ORDER	Number (Byte)	1
SEGMENT	Text	6
TIME_UNITS	Text	1
FACTOR	Text	8
DESC	Text	30
Notes	Memo	-

C:\My Documents\testdata.mdb
Table: Shift code

Columns

Name	Type	Size
SHIFT_CODE	Text	6
NAME	Text	30

C:\My Documents\testdata.mdb
Table: Shift details

Columns

Name	Type	Size
SHIFT_CODE	Text	6
WEEKDAY	Text	3
DAYNO	Number (Integer)	2
TIME_FROM	Date/Time	8
TIME_TO	Date/Time	8

C:\My Documents\testdata.mdb
Table: Support Replenishment

Columns

Name	Type	Size
RESOURCE	Text	10
QUANTITY	Number (Double)	8
LOGIC	Text	1
DATE_FROM	Date/Time	8
TIME_FROM	Date/Time	8
DATE_TO	Date/Time	8
TIME_TO	Date/Time	8

C:\My Documents\testdata.mdb
Table: Support Resources

Columns

Name	Type	Size
RESOURCE	Text	10
DESC	Text	30
SHIFT_CODE	Text	6
CRITICAL	Yes/No	1
AUTO_REPLENISH	Yes/No	1
RESERVE	Yes/No	1
TOLERANCE	Number (Double)	8
DISCRETE	Yes/No	1
CONSIDER	Yes/No	1
QUANTITY_ONHAND	Number (Double)	8
DATE_ONHAND	Date/Time	8
TIME_ONHAND	Date/Time	8
UNITS	Text	5
UNIT_COST	Number (Double)	8
MIN_LEVEL	Number (Double)	8
MAX_LEVEL	Number (Double)	8
LOCATION	Text	30
SUPPLIER	Text	30

C:\My Documents\testdata.mdb
Table: Work Center

Columns

Name	Type	Size
WORK_CENTER_CODE	Text	10
WORK_CENTER_NAME	Text	25
WORK_CENTER_NORMAL	Number (Long)	4
WORK_CENTER_MAX	Number (Long)	4
Notes	Memo	-

Appendix –3: The Planning Board Approach

STEELR'US currently is in the process of implementing bar coding technology of its drawings and pieces on the shop floor. A process template has been added to the drawing so that an experienced person can fill in routing information and duration can be tied in from original bid estimates by approximation. This will certainly improve data gathering, maintaining and updating the databases so allowing the scheduling to be properly done. They are also studying the feasibility of using 'neural networks' to generate process information given the type, say a beam or a column of a known length. While this process is underway, a computerized planning board has been developed for shop use. The shop superintendent can preview the drawings released to him from drafting and prepare 'load lists' on the computer. Two planning boards have been designed. In one, he can then pick and place these various loads on different machines / different dates at will. Resource levels can be set for each machine in terms of weight (tonnage) so that a warning will be issued when resources are overloaded. The idea behind this approach is to gather 'expert' opinion. When he picks and places the shop loads, he is informally deciding the processes to be followed and this information can be captured and over time enough knowledge can be accumulated from the expert's behavior and used by the system to do some on its own. The other planning board is useful if some sort of time estimate is available. The DSS will utilize the times to load to shop to capacity. This will provide with list of jobs to be undertaken in different work centers even though the ordering of the jobs is left to the shop personnel. An estimate of current shop loads and loads not currently scheduled is also computed so that the shop superintendent can decide on running an extra shift or over time. The planning boards are shown in Figures A3-1 and A3-2.

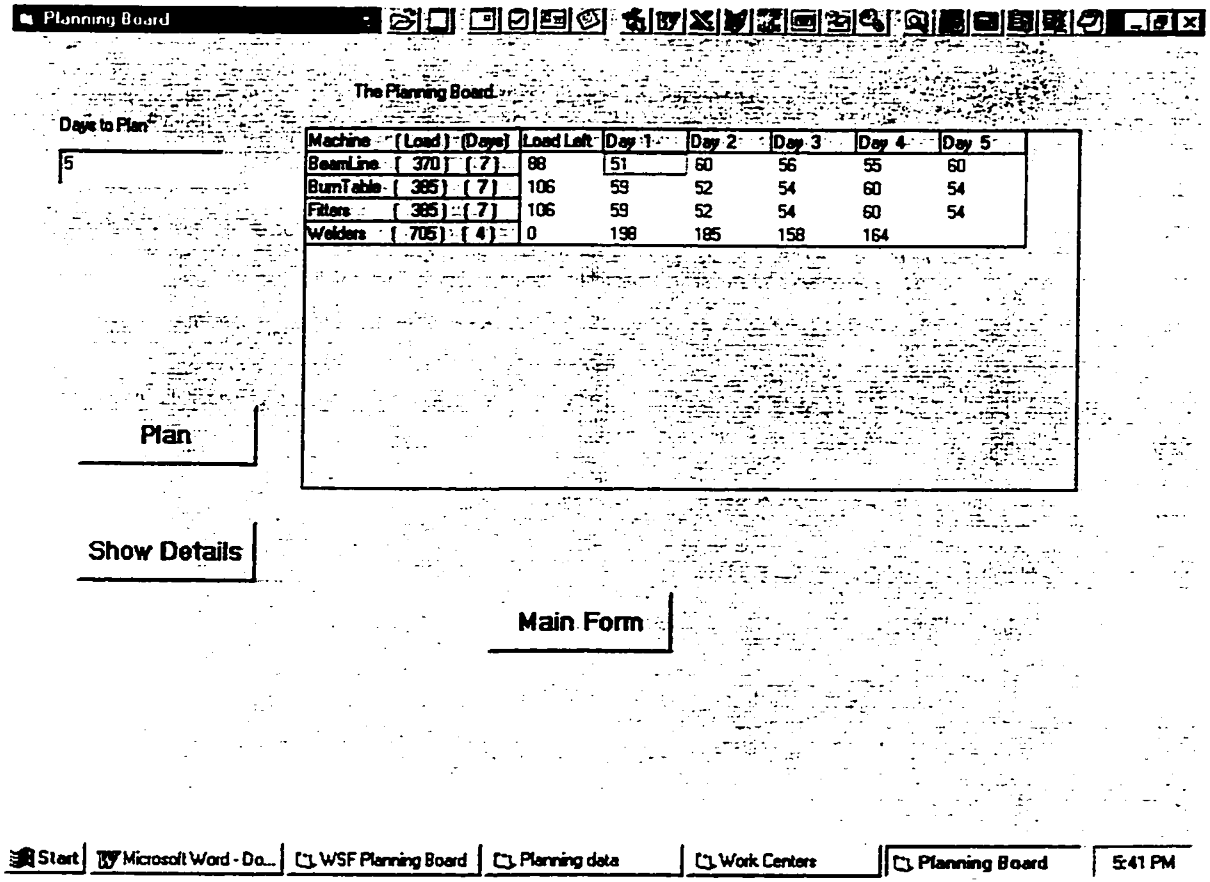


Figure A3-1: Planning Board 1

This approach assumes that times estimates are known

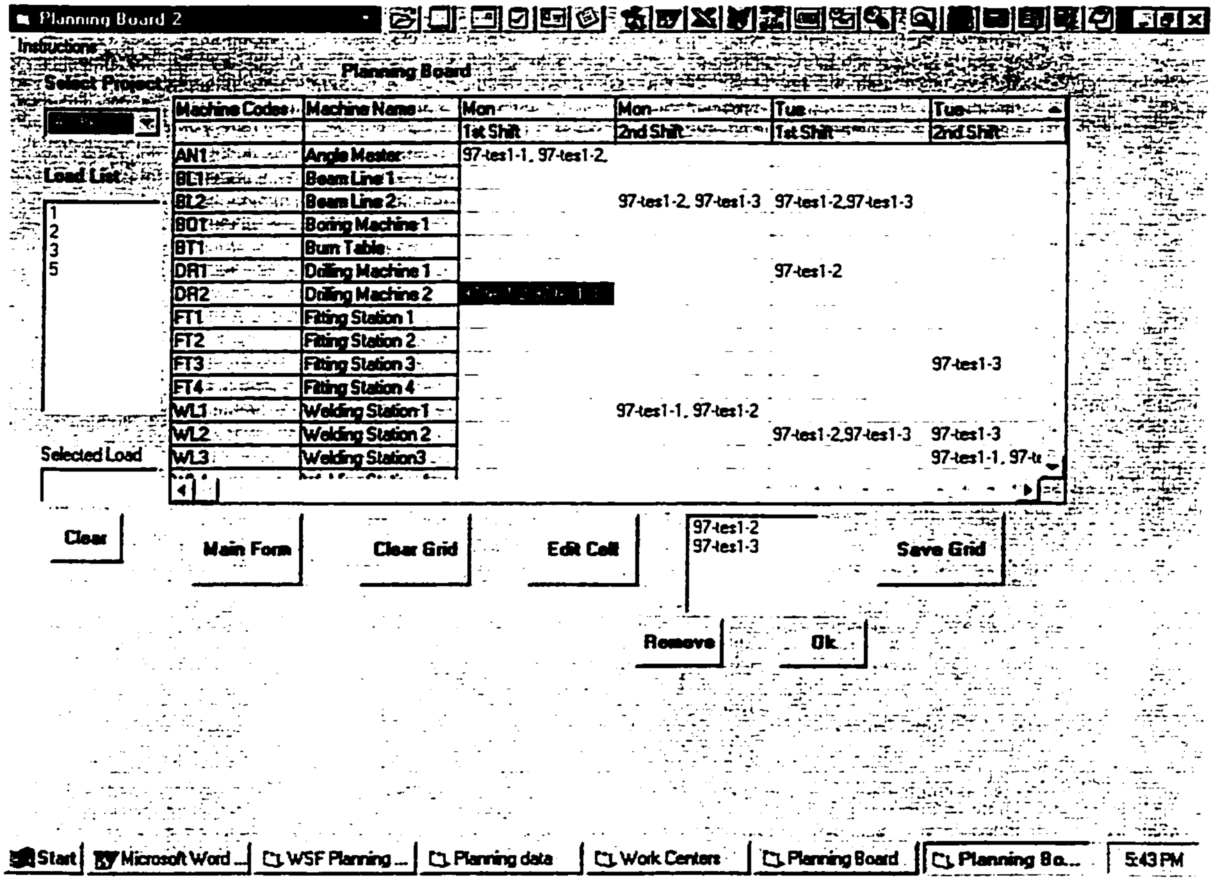
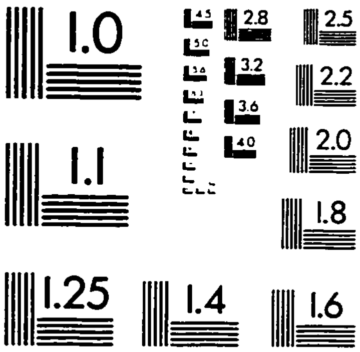


Figure A3-2: Planning Board 2

This is useful when time estimates are not available and the shop personnel would plan the "loading" of the shop.

IMAGE EVALUATION TEST TARGET (QA-3)



150mm

6"

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989