Algorithms for Pipeline Transfer Scheduling

John A. Young

VICTORIA UNIVERSITY OF TECHNOLOGY

FOOTSCRAY CAMPUS

THESIS SUBMISSION

FOR

MASTER OF APPLIED SCIENCE

(BY RESEARCH)

ALGORITHMS FOR PIPELINE TRANSFER SCHEDULING

Thesis by: **John Andrew YOUNG**
Department of Computer and Mathematical Sciences
Faculty of Science
Victoria University of Technology: Footscray Campus
Student number: 8401417

Supervisor: **Professor Robert E. JOHNSTON**
Dean
Faculty of Engineering
Victoria University of Technology: Footscray Campus

December 1992

CONTENTS

CONTENTS

ACKNOWLEDGEMENT

I would like to thank two people, without whose support the preparation of this thesis would not have been possible: Professor R E Johnston for his effective guidance and coaching throughout the duration of this study, and my fiancee Ruth, for her encouragement and support.

ABSTRACT

A common logistics problem in the petroleum industry is the scheduling of the pipeline transfer of petroleum fuels from oil refineries to distribution terminals. With a fixed set of pipelines, each with its own restrictions and constraints, the scheduler must organize the products and timing of transfers through each pipeline so as to deliver the required quantity of each product to the appropriate terminals to meet the demands of the market.

Of the many possible approaches to the scheduling problem which are reviewed in this thesis one basic approach was chosen for future research. Specifically this thesis results from an investigation of the feasibility of using a tree-search algorithm based upon a similar approach successfully used in paper machine trim scheduling.

The computer based system presented in this paper includes a data structure for handling the daily scheduling data, and a tree search algorithm to generate the schedule. The algorithm uses combinations of products on pipelines as the major decision variables. At each stage in the tree-search a limited set of "good" combinations is generated and ranked. The algorithm proceeds by choosing the highest ranked combination, adding that to the schedule and moving, chronologically, to the next stage. Backtracking is used to improve the solution quality by removing a selected combination and replacing it by the next ranked combination with a view to improving the objective function. The key to the algorithm is in the heuristic for the generation of the "good" combinations.

SUMMARY

A common logistics problem in the petroleum industry is the scheduling of the pipeline transfer of petroleum fuels from oil refineries to distribution terminals. Despite the complexity of this scheduling task most schedules appear to be prepared manually, often with the aid of a Gantt chart. This method is time consuming and relies heavily on the experience of the scheduler to produce acceptable solutions.

This work investigates a number of techniques developed to improve the Gantt based method used at Mobil Oil Australia's Altona Refinery. The work was directed at improving the decision making of the scheduler, principally by reducing the amount of time required for schedule preparation, thereby allowing more time for schedule evaluation.

The computer based system developed uses a heuristic approach imbedded in a tree-search structure. The algorithm uses combinations of products on pipelines as the major decision variables. At each stage in the tree-search a limited set of "good" combinations is generated and ranked. The algorithm proceeds by choosing the highest ranked combination, adding that to the schedule and moving, time-wise, to the next stage, until a solution is reached.

Backtracking is used to improve the solution quality by removing a selected combination and replacing it by the next ranked combination with a view to improve the objective function. The key to the algorithm is in the heuristic for the generation of the "good" combinations.

The algorithm, coded in FORTRAN 77, operates on the Company's mainframe computer. Solution times are short, typically of the order of 1 or 2 seconds. The algorithm could easily be adapted to run on a personal computer if required.

The algorithm's performance was tested using instance evaluation and simulation experiments. Performance was evaluated with respect to total weighted tardiness and work completed. Selection rules, used to rank jobs and job combinations, were analyzed as a function of machine load to determine relative performance over typical operating conditions.

The Shortest Processing Time (SPT) rule, supplemented with the pattern utilization heuristic, provided the best performance over the range of machine loads tested. The performance of this rule without pattern utilization enhancement was poor.

The Cost Over Time (COVERT) rule provided the best performance over the range of machine loads tested when the pattern utilization heuristic was not employed. At very high machines loads the Apparent Tardiness Cost rule provided the best performance.

The performance of the algorithm, both in terms of solution quality and ease of use, is acceptable for practical use. Further development will enhance performance, principally through improved backtracking and selection rule tuning.

Chapter 1 PROBLEM CHARACTERIZATION

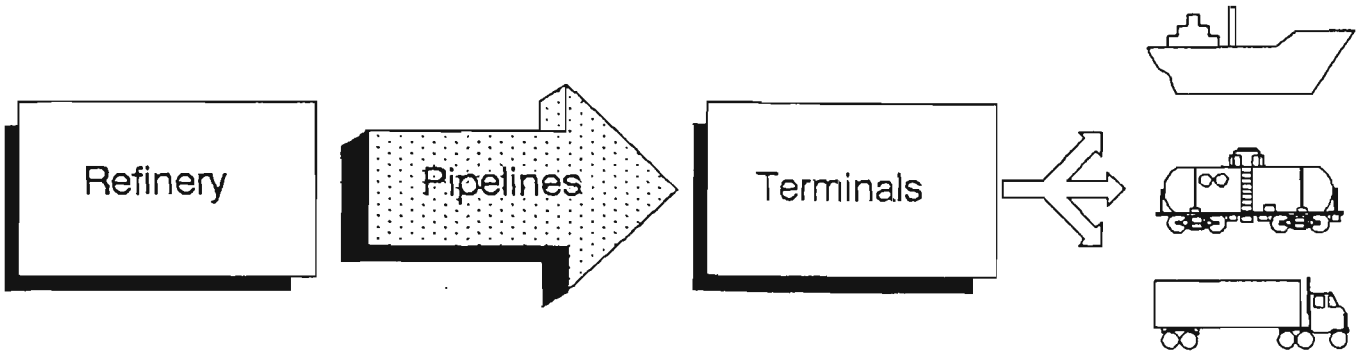


Figure 1.1 Typical Oil Refinery Product Distribution

1.1 Introduction

Pipeline networks are commonly used to transport petroleum fuels from oil refineries to distribution terminals. A schematic representation of this process highlighting the main entities in the distribution chain is shown above. In a typical world-scale refinery of 100 thousand barrels per day capacity this process involves moving approximately 16 million litres per day of product (about 13 thousand tonnes per day). A typical product range may include 10 to 20 distinct types, such as gasolines (petrols), jet fuels, diesel fuels and fuel oils. The pipeline system may consist of a single pipeline, through which all products are transferred, or may include multiple pipelines with dedicated services. The distribution terminals may be located far from the refinery and,

importantly from a scheduling viewpoint, may not be available continuously. Within these constraints the system must deliver the required quantity of each product to the appropriate terminals to meet the demands of the market, whilst maintaining segregation at each point in the distribution chain.

Despite the complexity of the task most schedules appear to be prepared manually, usually with the aid of a Gantt chart. This is due to a number of reasons. First, each refinery's pipeline network is unique and the lack of commonality hinders the application of generalized scheduling software. Secondly, competitive pressures tend to prevent developed systems reaching the public domain, if such systems do indeed exist. Finally, and perhaps most importantly, manually prepared Gantt based methods adequately handle the scheduling demands from an operational standpoint, although they may fall well short of simplifying the scheduler's task.

The principal shortfalls of any manual method are the time taken to generate the schedule and the reliance upon the scheduler's experience. The first factor is particularly important when schedules need to be generated frequently, because the time spent in generation reduces the time available to assess the implications of the proposal. In the system studied this generation can take many hours, and changing circumstances require daily revision of the schedule. The heavy reliance on the scheduler's experience for both the schedule quality and adherence to technological constraints imposes severe strains during initial training and risks penalties due to poor system performance.

This work investigates a number of techniques developed to improve the Gantt based method used at Mobil Oil Australia's Altona Refinery, located in Melbourne, Australia. The work was directed at improving the decision making of the scheduler, principally by reducing the amount of time required for schedule preparation, thereby allowing more time for schedule evaluation. This report describes the approach selected and methods developed to achieve these aims. Whilst the methods developed are not expected to be applicable to all refinery distribution networks they should provide sufficient flexibility to handle similarly structured problems.

1.2 Classification

Scheduling problems are typically characterized by four types of information (Conway [20], French [30]):

1. The jobs and operations to be performed.
2. The number and type of machines that comprise the shop.
3. The job/machine assignments which can be made.
4. The criteria by which the schedule will be evaluated.

In the above context, a job is the physical entity that is the object of work, and also refers to the work that is performed. In the pipeline scheduling problem a job is a specified quantity of a given product to be transported from the refinery to a distribution terminal.

A job may require one or more operations, in which each operation refers to a particular task performed on a machine. In this situation each job

only requires one type of operation, pumping. However, as explained later, each job may be split into a number of parcels which are processed as separate operations, either on different pipelines or at different times during the schedule.

A machine is a processor which performs one or more of the specified operations or tasks which constitute the job. In this case the machines are the pipelines which link the refinery to the distribution terminals. One or more machines are referred to collectively as a shop.

A resource or tool is entity required by a machine in order to perform an operation. This resource may or may not be consumed in the operation, and more than one resource may be required for the particular operation. The resources in this system are explained in Section 1.5.

An assignment is the allocation of a job to a machine for processing. This may be governed by a number of restrictions: for example in this problem certain product types can only be pumped on certain pipelines due to physical and logistical constraints.

In addition to the above taxonomy the following characteristics significantly affect the scheduling process:

The Schedule Horizon, which is the time-frame over which the schedule holds. In this case the horizon is typically one week (of seven days) from the date the schedule is prepared. Certain features outside this period may be known in advance, but will only be implicitly factored into the current schedule by priorities on certain tasks, such as

shifting more of one product than another, or more product to one offtaker than another.

The Generation Frequency, which is the time between generation of old and new schedules. Despite the horizon of about one week the schedule in this study is revised at least daily, and often three or four times each day to account for changing circumstances. These externalities typically include equipment malfunctions, delays in product releases or other distribution disruptions.

The distribution network studied is shown schematically in Figure 1.2 (page 25) and the main features of this system are described below.

1.3 Job Definition

In refinery jargon jobs are called "transfers", which are defined by the following major attributes:

1. Offtaker, which is the customer who will receive the transfer.

2. Product, which is the type of material the offtaker will receive, such as gasoline (petrol), jet fuel, diesel, etc.

3. Ready time, which is the time the product becomes available for processing, otherwise known as the "release time".

4. Due time, which is the time by which the customer expects to have received all of the transfer.

5. Quantity, which is the amount of product the offtaker expects to receive.

6. Priority, which is the relative importance of the transfer amongst its peers.

These are discussed in the following sections.

1.3.1 Offtaker

The offtakers in this study are the major oil companies competing in the Australian market place. Whilst it may appear illogical for a marketer/refiner to supply its competitors, the transportation costs resulting from Australia's geographical size encourage this form of supply optimization. This arrangement allows each marketer to compete in regions in which it does not have a refinery without transporting products long distances from its manufacturing centre.

For example, if this arrangement did not exist, a company would need to transport gasoline from its refinery on the west coast to sell in eastern Australia, while another company would transport an equivalent volume of the same product from its refinery in the east to sell on the west coast. These bilateral and multilateral exchanges bring significant benefits by minimizing needless transportation costs, and also reduce the risk of accidents when bulk quantities of fuel are moved. They also hinder the development of a monopoly in the region

housing the company's refinery that would otherwise arise due to a transport logistics advantage (BTCE [15], p35).

Following rationalization of the industry there are now approximately 5 offtakers, operating 7 distribution terminals. However, when this study commenced there were 7 offtakers operating 8 terminals, and prior to this there were at least 9 offtakers operating 10 terminals.

1.3.2 Product

The products in this study are fuels, such as aviation gasolines (avgas), motor gasolines (mogas or petrol), kerosene, jet fuel, light heating oil, distillates, and residuals. Each of these categories contains a number of different grades. This totals about 15 different materials, each requiring segregation during storage and transfer.

1.3.3 Ready Time and Arrival Pattern

Transfers are sourced from product batches. Whilst oil refineries operate continuously, products are blended in batches, with each batch corresponding to the contents of a single tank. In this way the required product quality can be guaranteed before the material is released into the market. Most product grades can be sourced from more than one tank. This allows certified product meeting sales quality specifications to be transferred while unfinished product is blended for the next batch.

Tank sizes are typically several times larger than transfer sizes, although occasionally large transfers may be composed of two or more

tank batches. In each case this results in a static job arrival pattern. The release time of each tank batch can be predicted in advance, and the scheduler determines the transfers which will be sourced from this batch. Hence the release times of the jobs sourced from a single tank are clustered around the time the batch is released. Judging which jobs should be sourced from each batch is a major problem in itself, but is beyond the scope of this study. (The loading and unloading of refinery tanks is examined by Christofides et al [17,18].)

1.3.4 Due Time

The due time for each transfer is set by negotiation between the refinery (the supplier) and the offtaker (the receiver). Obviously the priorities of the two parties are often different, and the offtaker must compete with its peers for product and pipeline availability.

Late transfers incur penalties due to demurrage charges, which in the case of coastal and international shipping are particularly severe. Quite apart from these financial considerations, delays result in a deterioration in customer relationships. These ultimately reduce the customer's willingness to cooperate and affect the flexibility of the system to meet demands.

Early transfers are not penalized, nor do they earn a credit to offset possible future delays of other transfers.

1.3.5 Quantity and Processing Time

The quantity of each transfer is set by negotiation between the refinery and the offtaker. A monthly plan, decided in advance by the respective supply departments of the organizations involved in the exchange program (not the refinery itself), allocates the monthly refinery production. It is the scheduler's task to determine how and when this allocation is provided. This must ensure an equitable distribution, so that individual offtakers are not disadvantaged if potential production problems disrupt supply and monthly targets are not achieved.

The specified transfer quantity is invariably fixed for each transfer without appreciable tolerance for over or under supply. This is principally due to the sourcing of transfers from tank batches of given volume: allowing a tolerance on one transfer would imply a corresponding tolerance on another transfer to balance the total batch quantity.

The processing time of a job is a function of the quantity transferred and the processing rate of the pipeline used. Whilst the quantity of the transfer is known the portion transferred by each operation comprising the transfer is not known until the schedule is developed. However, the pumping rate is known and fixed for each pipeline/offtaker combination and hence the problem is deterministic in nature.

1.3.6 Priority

Transfers acquire priorities due to a number of factors. These include the demand for particular product types (often a function of the offtaker(s) involved), ullage constraints at the refinery, and

downstream distribution mode: for example, seaborne liftings involve much higher demurrage penalties than road or rail. These priorities are predominantly unquantifiable, but qualitative assessments are used to meet the demands of the distribution system.

1.3.3 Number

The number of transfers processed each week (i.e. within the scheduling horizon) varies significantly, due to a number of factors including market demand, product availability and, quite possibly, the style of the scheduler involved. Data collected over a 32 week period are given in Appendix A and summarized in Table 1.1. These show that a typical schedule involves on average approximately 40 jobs, although this may range from about 25 to 50. This distribution is shown in Appendix A. Pipelines 1 and 4 are grouped together because of their common service (gasoline).

Table 1.1 Number of Jobs per Week

	Pipeline				
	1	4	2	3	Total
Average	12	11	11	4	38
Minimum	8	7	5	0	26
Maximum	17	16	17	8	49
Variance	6	5	9	4	35
Std Deviation	2	2	3	2	6

1.4 Machine Definition

1.4.1 "Main" Pipelines

Four main pipelines link the refinery to the offtaker distribution terminals. Each main pipeline is restricted to the type of products which may be transferred through it, due to the suction pipeline arrangement (described later) and by physical design. In addition, not all pipelines connect the refinery to each of the distribution terminals.

In the system studied two of the pipelines join prior to reaching a number of the terminals. This restricts these pipelines to transferring the same product type when transferring simultaneously to those terminals located after the confluence. This is equivalent to these pipelines sharing a common offtaker pipeline at each of these distribution terminals, and is shown on the diagram as such.

Each pipeline has its own pump, although some pipelines share a spare. Pipeline capacity (i.e. pumping rate) is a function of the main pipeline and the offtaker to which the product is being pumped (principally due friction losses over the distance pumped). Pipeline capacity is effectively independent of the product type being transferred.

1.4.2 Pipeline Availability

The pipelines are usually available continuously, however, planned shutdowns for maintenance and unplanned shutdowns due to equipment malfunctions do occur. Planned shutdowns may last for days, and the

schedules must take this into account. Unplanned shutdowns are rare, due to the sparing of critical equipment such as pumps, but may last for long enough to require revision of schedules to accommodate the lost capacity, or lost flexibility.

1.5 Resource Definition

1.5.1 "Suction" Pipelines

The suction pipelines connect the refinery tanks to the main transfer pipelines of the distribution network. (The pipeline network pumps draw product through these pipelines and hence the term "suction"). No tank is connected to all main pipelines, and some main pipelines share the same suction pipelines. The main pipelines sharing the same suction pipelines must therefore transfer the same tank contents, and hence same product type, if they operate simultaneously.

1.5.2 "Offtaker" Pipelines

The offtaker pipelines connect the main pipelines to the offtaker distribution terminals. Not all offtakers are connected to all main pipelines and some offtakers use the same pipeline to connect their terminal to more than one main pipeline. To maintain product integrity each main pipeline must transfer the same product type if the offtaker accepts transfers through a single offtaker pipeline simultaneously. Due to piping and valving arrangements some of these offtaker pipelines which connect multiple main pipelines cannot accept simultaneous transfers on the main pipelines.

1.5.3 Offtaker Distribution Terminals

The offtaker distribution terminals receive the product transferred from the refinery. The products are distributed from these terminals to the market by road, rail and sea. Each terminal contains limited tankage for each of the selected product types.

1.5.4 Offtaker Availability

One of the major factors affecting scheduling is that some offtakers are only willing to receive product during specified periods, or windows. These windows are typically a function of the day of the week. They can be extended by the offtaker working overtime if mutually convenient. Whilst this is obviously in the scheduler's interest, it may not be so for the offtaker.

1.6 Assignment Definition

Three main types of job/machine assignment or flow pattern discipline are recognized within the "machine shop" (French [30], p14):

F the flow shop, in which the machine order for all jobs is the same,

P the permutation job shop, in which the machine order for all jobs is the same and the job order for each machine is the same, and

G the general job shop, in which there are no restrictions on the form of technological constraints.

Our study involves a special case of the general category: although jobs may be processed on one or more machines (i.e. pipelines), the machines are not identical and are not independent of each other. This interdependency is due to common suction and offtaker pipelines and prevents the problem being decomposed into sub-sets dealing with each main pipeline in isolation. Hence at each point in the schedule the technological constraints of the suction and offtaker pipelines must be satisfied before the task of assigning jobs to machines can proceed.

1.6.1 Assignment Options

The offtaker and product define the job/machine assignments available within the system, i.e. the particular pipelines on which the transfer can be pumped. This in turn sets the requirement for resources, such as the suction and offtaker pipelines needed to link the main pipelines with the refinery and offtaker terminals.

1.6.2 Setup Times

Product integrity requires transfers of different product types to be segregated appropriately to avoid contamination. In some systems this is assisted by the use of "pigs" (solid devices which can pass through the pipeline, placed between dissimilar products), although this technique is not used in the network studied. Whether pigs are used or not the interface between such transfers must be identified and the pipeline purged until contaminants are reduced below an acceptable level. This results in sequence dependent setup times which must be recognized during preparation of the schedule.

The setup, or "lineclear", times are a function of the main pipeline and the dissimilar product types involved. In the system studied this pairwise interaction is most strongly dependent on the product about to scheduled and for practical purposes the dependence on the initial product is ignored. The setup times are not a function of the offtaker receiving the transfer because all lineclears are transferred to one terminal (that of the company operating the pipeline network).

The setup time between transfers of similar product, arising from valving and tankage changes, are typically small in comparison to the product transfer time and can be ignored.

1.6.3 Preemptability

Transfers, once commenced, may be halted to allow higher priority transfers to proceed, or satisfy some constraint, such as an offtaker terminal closing for the night, or to switch to another pipeline to take advantage of a higher processing rate. There is no limit to the number of times a transfer may be preempted, although each preemption results in some loss in processing capacity, due to the setup time involved.

1.6.4 Precedence Constraints

Precedence constraints are restrictions on the possible ordering of jobs, due to either technological constraints (i.e. some physical impossibility) or external policy. The are no general precedence constraints in this system, although restrictions may intermittently occur in which a defined ordering or transfers is required. This is most often due to a logistical requirement within an offtaker terminal.

1.7 Performance Evaluation

1.7.1 Objectives

The objective of the schedule is not easily quantifiable in terms of any economic function. The "cost" of a particular schedule is difficult to define, and similarly does not necessarily measure what the scheduler is trying to achieve. Instead, the scheduler's assessment of what constitutes a "good" schedule, or whether one schedule is "better" than another, is largely qualitative. Hence the main characteristics of "good" schedules and the incentives for improving schedule quality need to be examined to enable improvements to the existing method to be measured. Indeed, whether an "optimal" schedule can be defined, and whether optimality is needed or justified, should be reviewed.

In the system studied "good" schedules exhibit the following qualities. They:

1. balance refinery production within ullage constraints,
2. satisfy offtaker product demands, and
3. maintain offtaker cooperation.

Each of these characteristics shall be examined briefly:

Balance Production

Refineries operate continuously and process large quantities of material with limited storage capability. Plant start-ups and shut-downs are time consuming, expensive and adversely affect product quality. Costs due to lost production would typically approach hundreds of thousands of

dollars per day. Therefore, the main priority of the schedule is to minimize any disruption to the refinery operation resulting from failure to distribute product as quickly as it is manufactured.

Satisfy Demand

This involves supplying the required amounts of each product to each offtaker when required to meet the demands of the market. However, because the offtaker requirements may not necessarily match the refinery's priorities in terms of quantity, type, timing or disposition the differing priorities must be resolved in some mutually convenient manner. This leads to the third goal:

Maintain Cooperation

Cooperation of all parties involved in the distribution system is essential to smooth operation. Unsatisfactory performance on the part of the scheduler, due to late deliveries or perceived bias during supply shortages could potentially result in claims for damages. However, the main manifestation of customer complaints would be in terms of future inflexibility in receiving product transfers, significantly increasing the already difficult task of producing feasible schedules.

1.7.2 Performance Measures

Each of the above qualities is difficult to quantify explicitly, or even model implicitly through proxy measures. However, some parameters are required to assist assessment of schedule quality.

The nature of the production balancing goal suggests a need to quantify machine utilization. Maximizing utilization would imply that the work

the facility could perform was maximized, providing of course that the work was performed efficiently, and would reduce any likelihood of the machines limiting refinery production.

The second and third goals imply a need to monitor customer satisfaction and hence a measure of delivery performance is logical (other measures may also be appropriate). Other considerations, such as product quality, are seldom affected by delivery decisions in this system and are beyond the scope of the scheduling problem.

1.7.3 Optimality Requirement

The discussion of the previous sections suggests that optimality is likely to be elusive, not only due to the complexity of the scheduling task, but also because the performance measures are not rigidly definable. However, common sense suggests in any case that finding the optimal solution is not required. Instead, what is sought is a "good" solution, or solutions, which satisfy the requirements of the refinery and offtakers in a reasonable manner. Thus what is required is the ability to tell if a particular schedule meets the minimum requirements, and, if having met these requirements, one schedule is better than another. In addition, one needs to tell when to stop searching for better solutions: from a practical standpoint there is no need to press on with further improvement if the costs of doing so outweigh the benefits of the result.

1.8 Current Approach

A brief description of the current approach is necessary to understand the problems which initiated the need for this research.

The current method is a manual method which relies on the experience of the scheduler to prepare workable solutions. By way of background the scheduler is typically an engineer with one to two years experience with the company following recruitment, usually immediately after graduation. The person is generally assigned to the position for six to twelve months before rotation into another area.

1.8.1 Tasks

The primary tasks of the scheduler are:

1. Plan blending of finished products from intermediate blendstocks.
2. Organize laboratory testing of finished blends for quality certification.
3. Manage intermediate and finished product inventories.
4. Set job due dates in conjunction with offtakers, recognizing refinery and offtaker priorities and constraints.
5. Allocate transfers to batches.
6. Schedule pipeline transfers.
7. Discuss schedules with field personnel prior to implementation.
8. Monitor and report system performance.

Thus the scheduler is required to interact with a wide range of refinery personal and customers in order to obtain the required information and services to complete the scheduling task.

1.8.2 Preparation

The schedule is prepared with the aid of a Gantt chart. As discussed earlier, the schedule is revised on a daily basis, and often more frequently, for a seven day rolling horizon. The operational nature of the job, combined with the 24 hour operation of the refinery, results in occasional "out-of-hours" requirements for schedule revision. Hence the appeal of a manual Gantt based method due to its portability. A typical Gantt chart is shown in Figure 1.3 (page 26), in which the product transfers are shown as blocks with an internal arrow. This chart demonstrates the high utilization of three of the four major pipelines. Combined with the limited offtaker reception windows and physical assignment constraints, this results in a difficult scheduling task.

Schedule preparation typically takes about three hours per day, including offtaker liaison to set delivery requirements, for an experienced scheduler. Schedulers are trained during a "handover" period lasting about one to two weeks by the previous incumbent. This involves explanation of "rules of thumb" found useful for efficient operation, although scheduling techniques are developed by each scheduler through practice.

1.8.3 Implementation

Following preparation the schedule information is manually translated from the Gantt chart into a format suitable for implementation in the field by operators. A typical field schedule is shown in Figure 1.4 (page 27).

1.8.4 Problems

The major problem with the current method is the length of time required to generate suitable schedules. This leaves the scheduler little time to evaluate the implications of each proposal, due to the frequency with which revisions are required. Time constraints are such that often all that is sought is a feasible operation. Very little time can be devoted to improving what can sometimes be a poor solution to the operational problems at hand.

The most serious ramification of a poor schedule is lost production. This may not be manifested immediately, but may surface later due to loss of operational flexibility. In addition, increased operating costs due to labour overtime penalty payments may be incurred if transfers are not confined to the offtakers' standard availability windows.

Obviously infeasible solutions cannot be accepted. The refinery's production must be transferred to the offtaker terminals: only in extreme circumstances would production be reduced to satisfy the pipeline constraints. The scheduler simply must persist until a feasible schedule is obtained. It is the time taken to achieve this goal in difficult circumstances that presents the major problem.

Schedule quality is a strong function of the experience of the scheduler. Particularly during the initial learning phase, the novice must devote an extreme amount of time simply to obtain a feasible solution. This reduces the time available for other required tasks, such as gasoline blending. The refinery is particularly vulnerable during these periods to operational disruptions, whether resulting directly from poor or inflexible schedules, or indirectly through non-optimal performance to correct product quality deficiencies.

A further inefficiency of the current method is the manual transcribing of data from the Gantt chart the scheduler uses, to the operating instructions for the operations personal. Not only is the time consuming, it increases the chances for errors, which may result operational problems, such as poor system performance or product contamination.

1.9 Previous Investigation

A study of this pipeline scheduling problem was conducted (Young [75]), in which a mixed integer programming (MIP) approach was used. Although many of the technological constraints were able to be adequately modelled, the computational complexity of the MIP method prevented practical application. Solution times on even the smallest problems were considered too long for practical use, in a position in which efficient time management is crucial to meeting operational deadlines. On larger problems solutions were sometimes unattainable due to rounding errors arising during computation.

In short, the MIP method was considered impractical for the size of the problem and the solution frequency required. Even if the MIP method had been successful from this point, a further difficulty was that the method (at that time and probably still) required mainframe computing facilities, and access to these would have been awkward, though probably not impossible, when "out of hours" revisions were required.

1.10 Study Objectives

The objective of this study is to address the major problems currently experienced. This can be defined as follows:

1. Reduce the schedule preparation time to allow more time for evaluation of the solution.

2. Provide solutions at least as good as those generated manually by experienced schedulers.

3. Reduce the dependence of the schedule quality on the experience of the scheduler.

4. Achieve the above with a simple to use and simple to understand system, providing practical and reliable performance, in a cost effective manner.

The first and third objectives have been well discussed in preceding sections. The second category, whilst initially obvious, raises a difficult question: how can the schedule quality be evaluated? General

performance measures have already been briefly discusses, and schedule quality is reviewed within the context of the reported literature in Chapter 2.

The final objective recognizes the user requirements of any installed system. Unless the system makes make the scheduler's current job easier it is unlikely to be used to its full potential. However elegant the underlying mathematical model and good the final solution, if the user doesn't feel comfortable with its predictions, or finds it unwieldy to use, it will not fulfil its purpose. It is important to recognize that any new system is not intended to replace, or reduce, the scheduler's involvement in the scheduling process. Instead what is sought is an improved tool to assist the user in the successful solution of the scheduling task.

Figure 1.2 Distribution Network

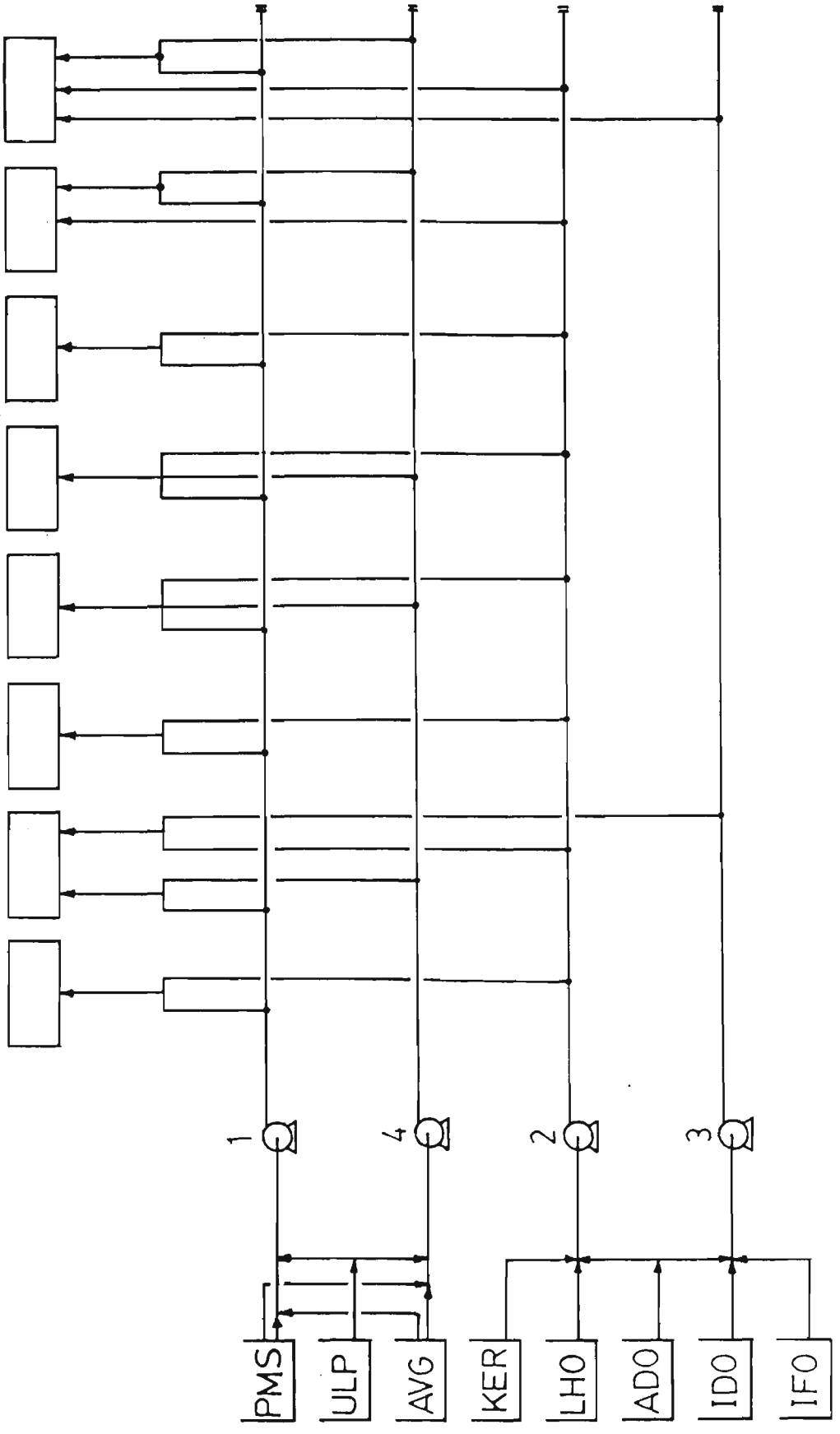


Figure 1.3 Typical Gantt Chart

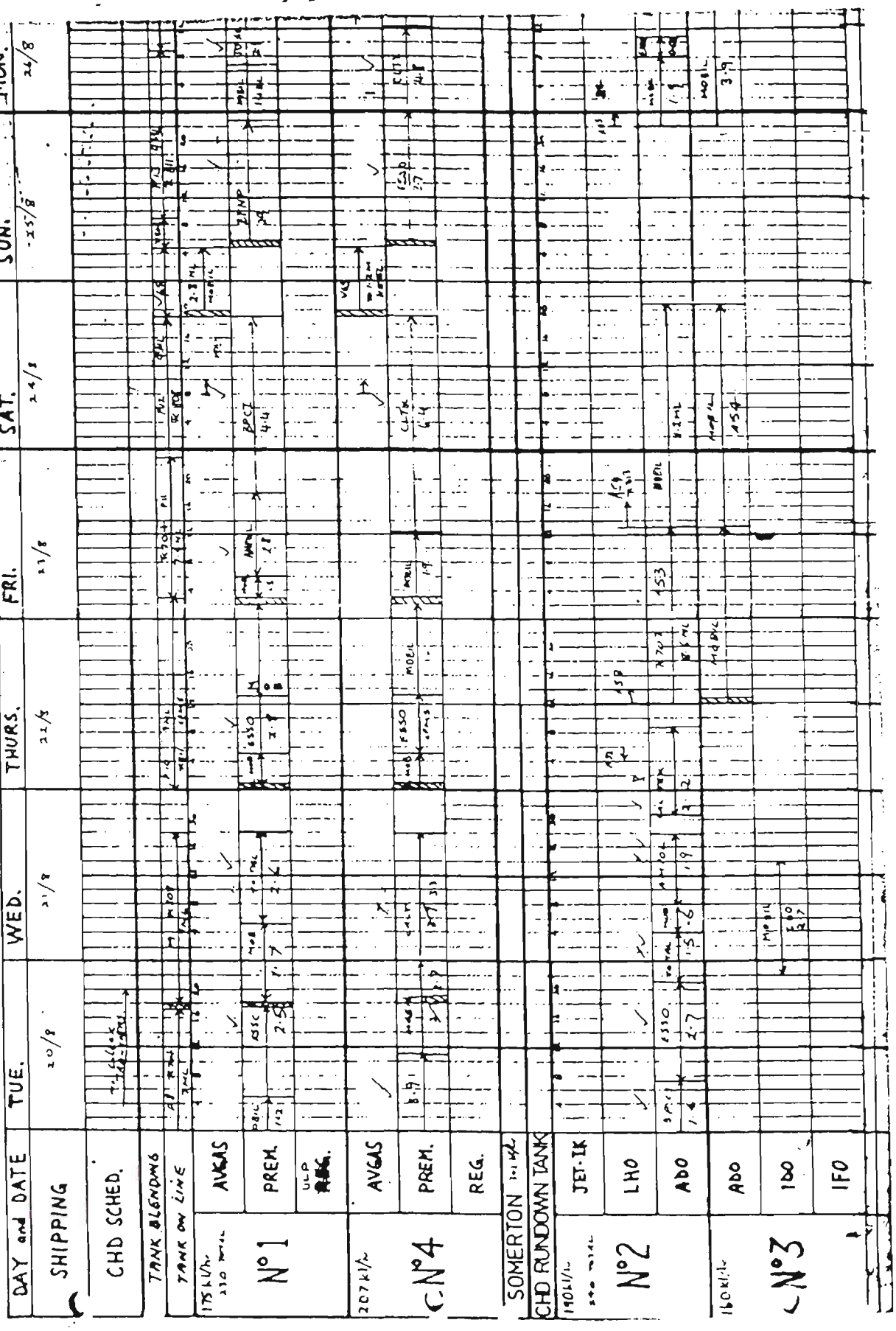


Figure 1.4 Typical Field Schedule

NOV 16 '88 09:23 P.R.A.ALTONA 391 0790 P.4/4

PETROLEUM REFINERIES (AUST). ALTONA REFINERY

PIPELINE SCHEDULE - version two (rip up version 1)
DATE : FRIDAY 11/11 ---> MONDAY 14/11

	DAY	PROD	TANK	COMPANY	VOLUMES & COMMENTS
NO. 1 LINE	static on PMS				
	FRIDAY	V69	508	MOBIL	L/C see note 1
	FRIDAY	V69	508	MOBIL	1000 KL
	static on AVGAS				
	SATURDAY	P9	704	MOBIL	L/C see note 2
	SATURDAY	P9	704	MOBIL	2500 KL
	SATURDAY	P10	811	MOBIL	L/C
	SATURDAY	P10	811	MOBIL	2800 KL
	static on PMS				
NO. 4 LINE	FRIDAY	P7	808	CALTEX	1500 KL
	static on PMS				
	FRIDAY	V69	508	MOBIL	L/C see note 1
	FRIDAY	V69	508	MOBIL	1150 KL
	static on AVGAS				
	SATURDAY	P9	704	MOBIL	L/C see note 2
	SATURDAY	P9	704	MOBIL	3000 KL
	SATURDAY	P10	811	MOBIL	L/C
	SATURDAY	P10	811	MOBIL	1200 KL
	SUNDAY	P10	811	CALTEX	2500 KL
NO. 2 LINE	FRIDAY	A21	812	BP ALT	640 KL
	FRIDAY	A21	812	CALTEX	920 KL
	SATURDAY	A22	700	MOBIL	3400 KL see note 3
	SUNDAY	A21	812	MOBIL	L/C
	SUNDAY	A21	812	ESSO	2000 KL
	SUNDAY	A23	812	MOBIL	3300 KL
	MONDAY	J30	602	MOBIL	L/C
	MONDAY	J30	602	MOBIL	TO CAPACITY
NO. 3 LINE	FRIDAY	A21	812	AMPOL	850 KL
	SATURDAY	A22	700	MOBIL	6100 KL see note 3
	SUNDAY	A21	812	MOBIL	L/C
	SUNDAY	A21	812	MOBIL	6000 KL

SOMERTON LINE

COMMENCE	VOL	PROD	TANK	PERCENT OWNERSHIP MOBIL : ESSO : BP : CTX
No transfers planned				

NOTES

Note 1. Start AVGAS transfer as soon as lab has released.

Note 2. Special cargo of Pacific Island Premium. Let Mobil know that this is coming down and must be segregated.

Note 3. Pacific Island ADO. Let Mobil know when this tank starts and ends.

Chapter 2 LITERATURE REVIEW

2.1 Introduction

The complexity of the general job shop scheduling problem is well recognized (Conway et al [20], p 102) and viable optimizing algorithms for all but the simplest versions have yet to be found. Graves [35] noted that actual scheduling environments are so variable that classification is very difficult, and the inherent uniqueness generally requires problem-specific solutions. He observed that a distinct lack of correspondence exists between theory and practice, and concluded that for complex settings the theory is often not sufficiently developed to be immediately applicable.

Even prior to conducting a literature search the writer had reason to doubt previous successful solution of similar pipeline scheduling operations. First, the organization for which this study was carried out is a major oil company operating globally. Despite involvement in the oil industry for many years, a large research and development organization, and refinery operations located in many countries, no "better" practical methods beyond the existing manual method were in use or known at the time. Secondly, the company's research and development group involved in scheduling applications were not aware of any competitor's developments in this area. And finally, no vendors had offered products suitable for this scheduling task.

Therefore, one must ask why are better methods not generally available? A number of reasons have been mentioned in Chapter 1. Graves' [35] experience confirms a number of these; namely that individual problems

usually require situation-specific solutions and competitive confidentiality prevents the disclosure of many methods. In addition, personal computer facilities with sufficient processing capability have only recently become available. Schedulers have been unwilling to grasp computer based methods while access and ease of use have been unsatisfactory. It will simply take time before suitable applications are developed, and become available from the user's desk top.

By far the most predominant solutions are purely manually based scheduling systems (Graves [35]). These systems rely primarily on the expertise of experienced schedulers, who typically use nothing more than graphical aids such as a Gantt chart. Schedule evaluation appears to be qualitative and the dominant schedule evaluation criteria are schedule feasibility and flexibility. Human understanding of the problem, adaptability to new constraints and qualitative assessment of conflicting priorities invariably produces good solutions to practical problem instances, albeit after extensive human input.

The scheduling literature tends to concentrate on simple systems with somewhat idealized constraints. This is understandable given the inherent difficulty of many practical scheduling problems. However, the assumptions introduced to make the problems tractable often render the solution algorithm unsuitable for real world problems. Obviously extensions to model certain practical aspects have been developed. However, the writer has been unable to locate any reported computer based method, which adequately handles all of the constraints and relationships necessary to model the pipeline transfer task at hand.

In this chapter some of the methods which have been used to tackle certain features of the problem at hand are examined. The aim is to find a solution strategy (or strategies) worthy of further consideration and which can be developed into a practical scheduling tool. This review is not intended to be an exhaustive evaluation of alternative methods to justify the "best" approach from a theoretical standpoint, if in fact such an approach can be demonstrated. Instead, it provides a brief investigation of general methodologies to assess their potential suitability, and then examines reported approaches to specific characteristics of the pipeline scheduling problem.

2.2 Manual Methods

One striking feature in the current literature is the lack of commentary about manual scheduling methods. This is not perhaps surprising given that it is primarily dissatisfaction with manual systems that leads people to investigate alternative (computer based) methods. Nevertheless, given the extent of use of manual methods it is important to determine whether refinements to this method exist which are worth considering.

As mentioned above, Gantt chart based methods appear to be the primary manual tool. Their use is noted and briefly described in most basic scheduling texts (Conway et al [20], French [30]). Clark [19] provides a thorough explanation of the preparation and use of these charts for scheduling and evaluation. Whilst his work discusses the range of applications of Gantt methods to layout, load and progress charts, it does not highlight any techniques which would likely lead to significant

productivity improvement over those currently employed. His work, however, emphasises that Gantt charts provide one of the best ways of recording the developed schedule so that intelligent decisions can be drawn from them.

One principal enhancement of manual Gantt charts has been to use them as the the basis for computer graphics representations. In this way the computer can be used to check adherence to technological constraints as the jobs are moved about on the screen by the scheduler. One major Australian airline currently uses this approach for flight scheduling.

Jones [48] discusses the extension of Gantt charts to 3 dimensions, a feature made practicable by the introduction of computer graphics software. Jones does not present better methods for determining the schedule sequence, but concentrates on the improvement in problem representation achievable with current technology. He noted that the three dimensional chart was probably not useful by itself in helping decision makers produce better schedules, but the ability to animate sensitivity analysis would be extremely useful.

Akers and Freidman's graphical procedure is reported in most basic scheduling texts (Conway [20, French [30]). Whilst the 2 job job-shop problem is easily visualized (on 2-dimensional rectangular Cartesian coordinates), the extension of this method to more jobs, and hence higher dimensions, is extremely unwieldy, and unsuitable for practical use.

The prime advantage of manual methods is that the scheduler usually understands all the critical aspects of the problem. It is well

recognized that the scheduler may have knowledge of scheduling considerations which are impossible to capture within any given model of the scheduling system (Baker [8], p44). As a result the scheduler is usually able to produce good schedules. However, when the combinatorial nature of the problem becomes too complex the human scheduler tends to rely on simplifying rules of thumb. The real role of computer based algorithms is to guide the user away from these rules when these rules become non-optimal (Baker [8], p44).

2.3 Generic Numerical Methods

The lack of success with constructive algorithms (see Section 2.5) has encouraged the development of enumerative and heuristic methods (French [30]). Obviously, one approach to obtaining optimal solutions would be to evaluate (that is, enumerate) all candidates and select the best solution based upon some performance measure. The difficulty with this approach is simply the enormous number of solutions possible for problems of non-trivial size. French ([30], p 19,77) notes that complete enumeration of the general job shop produces (n!)m candidate solutions, where n is the number of jobs and m is the number of machines available to process the jobs. Thus even reasonably small problems are currently unsolvable using this method within reasonable time limits on even the largest computers. Computational intractability therefore forces us to adopt a more sophisticated approach. We shall briefly review some enumerative methods which implicitly exclude non-optimal solutions by intelligently selecting and evaluating candidate solutions.

2.3.1 Mathematical Programming Methods

2.3.1.1 Integer Programming

Several attempts have been recorded of recasting scheduling problems as mathematical programs and in particular mixed integer programs (MIP). The initial attraction of this approach is that once recast in this form the problem may be solved by readily available standard algorithms, and hence optimal solutions can be obtained. However, practical experience shows that this approach is not promising.

Computational Experience

The standard mathematical programming algorithms currently appear to be practical for only small problems (Garfinkel and Nemhauser [32] p387), although this is changing with increases in computer processing power and improvements in solution algorithms. However, the size of the resultant MIP problem can often lead to time-consuming and erratic behaviour of algorithm codes. As noted in Chapter 1, solution failure due to numerical accuracy errors was observed in earlier pipeline scheduling work (Young [75]), and solution times, when optimization was achieved, were unsatisfactory from a practical standpoint. Darby-Dowman and Mitra [22] note that the use of set covering, set partitioning and set packing models has not been an unqualified success.

The main problem of the MIP approach is that no generalized method yet exists to optimize large problems in a reasonable time period. Stepping from standard linear programming (LP), in which all of the variables are continuous, to MIP in which some of the variables take integer values, increases the computational complexity of the solution enormously.

Zionts ([76], p477) notes that at least with the present level of technology, IP and MIP problems are many orders of magnitude more difficult than LP problems. Added to this, he states that "just because the words "Optimal Solution" appear on the printout does not mean that the solution is optimal", nor does confirmation of this solution as the optimum by any other code. Most importantly, he notes that many problems are formulated as IP problems because they can be, and not because they should be.

Algorithms

Williams ([74], p 155) discusses general methods of solving IP and MIP problems, and Zionts ([76], p 480) illustrates several techniques for accelerating convergence. Garfinkel and Nemhauser ([32] p388) note that algorithms based exclusively on cutting planes or implicit enumeration without surrogate constraints have not, in general, been effective for medium and large problems.

Land and Powell [51] and Tomlin [67] note that all commercially available MIP codes use a form of "branch and bound" solution strategy to work systematically through the integer variables. Invariably these codes require the user to select the parameters which control the order in which the integer variables are evaluated (IBM [41]). Haverly Systems Inc. [39] (p 10.7) state that "no single strategy is best for all models. In fact, little is known about the best strategy to use for various types of models, or even how to classify models". Customization is required to tune the computational strategy to the problem at hand. Results can be unpredictable despite this tuning and the optimal solution can not always be found.

Formulation

Much work has been done on improving the performance of MIP as an optimization technique, both in the area of problem formulation and algorithm development. Johnson et al [43] present methods to assist in the solution of MIP planning models, some of which could be applied to scheduling applications. Their approach takes advantage of the hierarchical structure of the integer variables and strengthens the model formulation, enabling some previously insoluble problems to be optimized. Bruvold and Evans [14] and Jeroslow and Lowe [42] present approaches which reduce the number of required binary variables, which eases the task of the MIP optimizer and increases the size of the problems which can be successfully solved. Reported solution times for both of these latter approaches were of the order of minutes for problems with only a small number of integer variables.

Some of these techniques could be applied to the initial MIP model constructed to solve this pipeline problem (discussed in Chapter 1), and possibly could enable the problem to be successfully optimized. However, the relatively long solution times required would still preclude using MIP as a practical scheduling tool for this situation.

2.3.1.2 Dynamic Programming

Dynamic programming (DP) has been investigated as a scheduling technique, originating with the work of Held and Karp, and extended by others (Conway et al [20]). Its principle advantage over complete enumeration is that it eliminates many possible schedules as it constructs the optimal processing order. Despite this advantage its use

appears limited to problems of up to about 25 jobs (French [30], p97), too small for the pipeline problem.

Dimensionality

The restriction in job numbers is due to the rapid increase in computational and storage requirements as the problem size grows. The number of calculations is approximately proportional to $n*2^{(n-1)}$ (French [30], p97). Additionally, at each stage of the optimization all of the partially constructed schedules are stored until all of the problems at the next stage have been solved. It is this storage requirement that is probably the practical limit on problem size, rather than the amount of calculation (Conway et al [20], p65).

Formulation

The introduction of precedence constraints may reduce the calculations involved and hence enable problems with more jobs to be solved. Similarly, the use of dominance conditions (elimination criteria) may significantly reduce the solution difficulty. On the other hand this tends to make substantial demands on storage requirements and hence effectively limits DP to problems with less than 25 jobs (French [30], p102).

2.3.1.3 Branch and Bound

Branch and bound (B+B) is a general purpose strategy that is widely used for many combinatorial problems, such as in most commercially available MIP optimization codes and as the basis for many heuristic methods, and not just scheduling applications. Branching partitions the original problem into two or more sub-problems, and bounding places a value on

the optimal solution to curtail enumeration of non-optimal sub-problems. The crux of successful methods is the order in which the sub-problems are selected for evaluation and the efficiency of pruning sub-optimal schedules.

The basic concept is shown in Figure 2.1. The problem is partitioned into a number of sub-problems, termed nodes. The problem may be visualized as a tree, with the branches representing the connection between the parent node and successor (or child or offspring) nodes. The first node is termed the root node and a node which has no successors is called a terminal node.

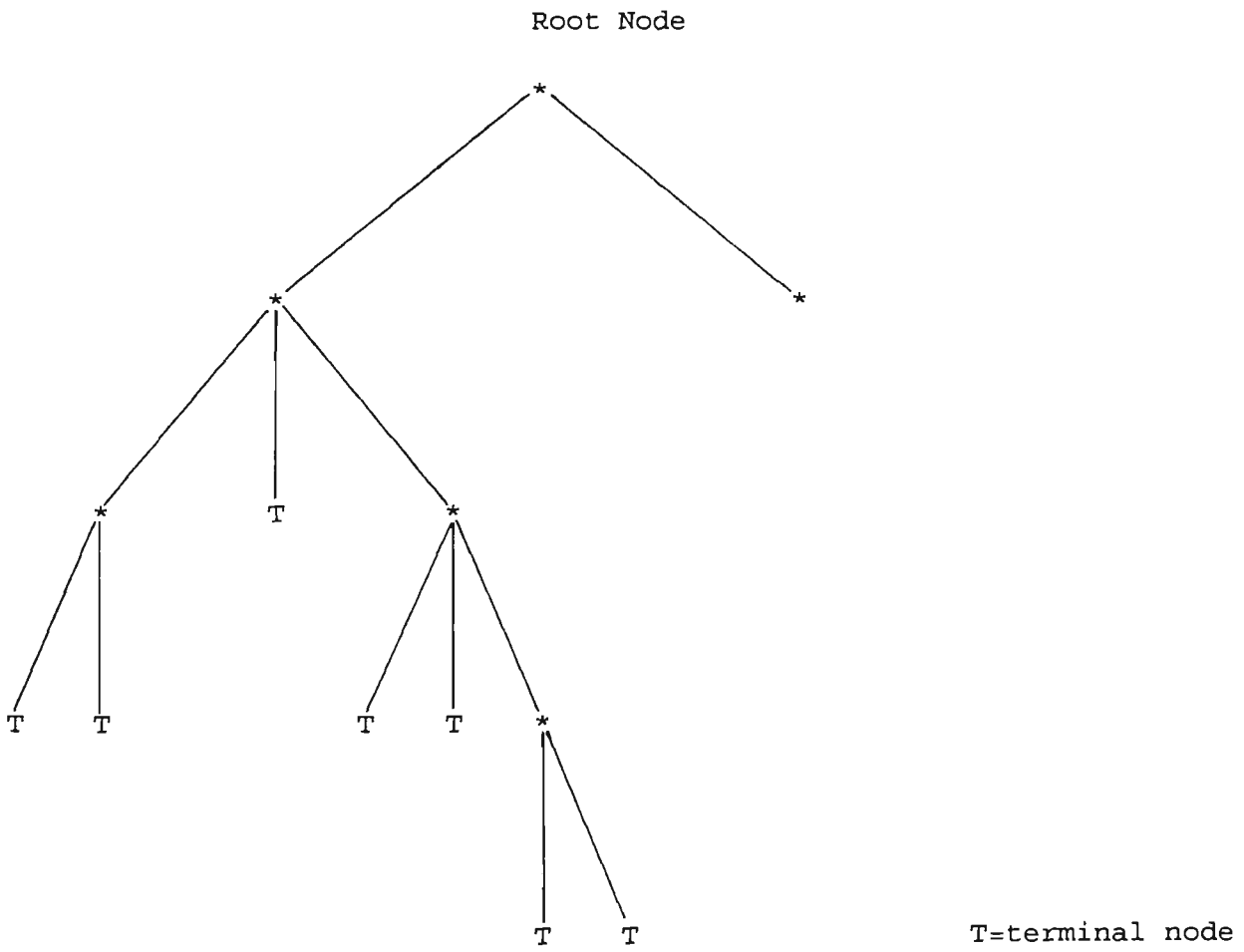


Figure 2.1 Branch and Bound Tree

Explanations of the basic methodology of the branch and bound method are widely available (French [30], Conway et al [20], Baker [5]). The following section concentrates instead on the practical application of this technique to the pipeline scheduling problem and discusses reported limitations.

Graves (1981) [35] notes that all optimization approaches to the job shop category appear to be B+B procedures. These procedures differ primarily with respect to the branching rules employed, to the bounding mechanism and bound generation. Despite the research conducted on the method the largest problems reported to have been solved have involved less than 10 tasks scheduled on less than 10 processors.

Branching

Barker and McMahon [11] concur that the most successful algorithms for scheduling the general job-shop are of the B+B type. They note that two classes of enumeration scheme may be distinguished. In the first an operation is chosen to be scheduled adjacent to part of the schedule already fixed. In the second branching takes place on the basis of some conflict which seems important, where-ever it appears in the schedule. In their approach (based upon the second class), each node contains a complete schedule. The conflict resolution which takes place at each node is not usually between two operations, but rather between one operation and a block of several others.

Stinson et al [63] note that schedules rapidly become very large. Therefore it is necessary whenever possible to prune away portions of the tree, hopefully in its earlier stages of development. Potts and van Wassenhove [58] concluded that in order to solve difficult combinatorial

problems all available fast pruning devices should be used. Pruning operations are done in two ways: by dominance pruning and by lower bound pruning.

Dominance Pruning

Stinson et al [63] note that the most efficient implementation of dominance pruning is by "left-shifting". Thus if any operation assigned a start time in some partial schedule can be left shifted to an earlier start time without violating a precedence or resource constraint the partial schedule is dominated. This is easily visualized on a Gantt chart.

Lower Bound Pruning

In lower bound pruning selected constraints are relaxed in order to obtain a lower bound estimate of the objective value of the complete schedule. The inclusion of the constraints can only serve to reduce the quality of the objective value. Thus if the estimate to the relaxed problem is greater than that of some other known complete schedule the existing partial schedule can be pruned away.

Obviously the B+B tree is more effectively pruned with a stronger lower bound. Stinson et al [63] note three types of lower bounds for multiple resource constrained schedules:

1. precedence based, in which the lower bound is calculated by ignoring any resource constraints, and hence the resulting schedule is due only to precedence relationships,

2. resource based, in which precedence constraints are ignored and the schedule duration is determined by the available resources required by the remaining tasks, and

3. critical sequence, in which precedence and resource constraints are simultaneously taken into account.

In their approach all three lower bounds are calculated and the largest of the three is taken to be the lower bound of the schedule.

Node Selection

The manner in which the tree develops is a function of the node selection heuristic. Stinson et al [63] note that to some degree pruning is enhanced through the rules which govern the manner in which the tree is allowed to grow. They claim that the use of a single node selection rule for very large problems may be of limited effectiveness. This arises because in large trees many nodes may have the same lower bound. They supported the use of a series of tie breaking rules, called a decision vector, for selecting the next node from which to branch. Their decision vector contained a combination of "look-ahead" and "look-back" parameters, such a lower bounds and machine utilization respectively.

Computational Performance

Stinson et al [63] presented results for a number of project, flow shop and job shop scheduling problems. Their results are probably not directly applicable to the pipeline scheduling problem because it does not contain the precedence relationships which figured prominently in their project scheduling and flow shop problems. They noted the

performance of their procedure was significantly influenced by the improved critical sequence lower bound. The performance of their algorithm deteriorated with increasing problem size and most job shop problems with 10 jobs on 6 machines (the largest problem size reported) were not able to be solved. Importantly they noted that the disadvantage with their B+B approach (and others in general) is the essentially unpredictable variance in computation time from problem to problem.

Potts and Van Wassenhove [57] noted that none of the algorithms in the literature, including both branch and bound and dynamic programming, have been successfully applied to problems with more than 20 jobs. The branch and bound methods are limited by computational times and the dynamic programming algorithms are limited by core storage requirements.

Their algorithm for the single machine weighted tardiness problem incorporated a new Lagrangian relaxation method and checked dynamic programming dominance during the node search. All testing appeared to be conducted on synthetically constructed problems rather than practical situations. They concluded that their method was suitable for single machine problems with up to 40 jobs.

2.3.2 Heuristic Methods

Much has been written on heuristic methods and their application to sequencing and scheduling. Eglese [25], Silver et al [60], Fisher and Rinooy Kan [27], Ball and Magazine [10] and Foulds [29] discuss the general design, analysis and implementation of heuristics in operations

research. Pearl [55] provides a formal treatment of search strategies, performance analysis and properties of heuristic methods. French [30] examines some applications of heuristics to schedule generation techniques. Given the breadth and depth of this subject this discussion reviews the basic types of methods before examining some applications to specific scheduling problems.

In the following discussion the definition of a heuristic provided in Eglese [25] and Silver et al [60], and attributed to Nicholson, is used:

> A heuristic is a procedure for solving problems by an intuitive approach in which the structure of the problem can be interpreted and exploited intelligently to obtain a reasonable solution.

Why use a heuristic?

Silver et al [60] and Fisher and Rinnoy Kan [27] note the historical interest in heuristic methods. Essentially, early research efforts were directed at finding optimal, or exact, solutions to management science problems. However, the failure to find solutions to some problems and the discoveries about computational complexity (NP-completeness, etc) indicated that for many problems effective optimization algorithms probably cannot be developed. This has turned attention to heuristic methods, and their ability to efficiently find good approximate solutions.

Whilst computational intractability is the chief reason for the use of heuristics to solve mathematical problems other important reasons exist. Silver et al [60] and Eglese [25] list a number of these. These include ease of understanding, development time and costs, inexact or limited

data, and the inclusion of secondary objectives. Silver et al note that because heuristic methods are often simpler to understand, this markedly increases the likelihood that the approach will be implemented in practice. Both articles include Woolsey and Swanson's observation that "people would rather live with a problem they cannot solve than accept a solution they cannot understand." Eglese states that this ability to understand the solution gives users a better idea of whether the model produces acceptable results. Haessler [36] argues instead that industrial grade heuristics will be very complex, and a rational user should be concerned with the quality of the answer and the effort required to get it, not with how easy it is to describe the procedure. The author's own industrial experience, however, is that senior management are unwilling to accept the solutions provided by models unless the predictions can be rationalized in terms of simple known relationships. Hence a dilemma arises in explaining in simple terms the output from what may be a complex computational procedure in order to gain the necessary support of the organization.

2.3.2.1 Types of Heuristic

Eglese [25] notes that many classifications have been proposed for heuristic methods. His classification, which closely matches that of Ball and Magazine [10], is based upon the method by which the solution is obtained. On the otherhand, Pearl [55] recognizes three categorizes of heuristics, based upon the type of solution sought. Both approaches are useful in developing an appropriate solution methodology. However, the latter, "ends" as opposed to "means" approach, is noted because it emphasises the solution goal. These categories are:

1. Optimizing, in which the aim is to establish the optimum solution.

2. Satisficing, in which the aim is to discover any (feasible) solution with as little effort as possible.

3. Semi-Optimizing, in which the aim is to find a "good" solution with a reasonable amount of search effort.

Semi-optimizing problems fall into one of two categories:

1. Near-Optimization, in which the boundaries of the acceptance neighbourhood (i.e. the proximity of the "good" solution with respect to the optimum) are sharply defined, such as within a specified factor of the optimum

2. Approximate-Optimization, in which the "good" solution is near the optimum with a sufficiently high probability (i.e. "most of the time").

As discussed in Chapter 1 a good rather than optimal solution is sought, and hence the "approximate-optimization" heuristics are of primary interest.

2.3.2.2 Basic Search Procedures

Heuristic search procedures may be broadly grouped into systematic and unsystematic approaches. Whilst each approach has specific advantages Pearl [55] emphasises the importance of using a systematic search, so that "no stone is left unturned" and "no stone is turned more than

once". The following section summarizes the basic approaches which are relevant to the problem at hand and discusses the selection of specific heuristics used to guide the search for a solution. The procedures essentially use a branch and bound framework, but incorporate a heuristic to determine the order in which the nodes are evaluated and pruned to trim the problem tree to manageable size.

Unsystematic Search: Hill Climbing

The hill climbing strategy is based on local optimizations, choosing the direction of steepest ascent from the current position. This strategy is termed "greedy", because each successive step is taken so as to maximize the immediate gain. It is the simplest form of search strategy because it involves little computational effort, and maintains no memory of past attempts or the path taken to get to the current position.

This strategy amounts to repeatedly expanding a node (generating all successors of a parent node), inspecting all its newly generated successors, and choosing and expanding the best of these. No further reference is retained of the parent or sibling nodes. The computational simplicity of this method is not without its shortcomings, standing a high chance of missing the global "optimum" due to be lured into an region in which a local "optimum" resides, or one in which no feasible solution exists. In its basic form this strategy is irrevocable because it does not permit backtracking to previously suspended alternatives.

Uninformed Systematic Search

A number of strategies exist for systematically exploring the solution space and avoiding the pitfalls of unsystematic search. In uninformed methods the order of the search does not depend on the nature of the

solution sought. Thus the location of the goal node does not alter the order in which the nodes are expanded. These strategies tend to be inefficient and are usually impractical on large problems. However, they can be readily altered to incorporate heuristic information to order the node expansion.

1. Depth-First

In depth-first search priority is given to nodes at deeper levels in the search tree. After node expansion, one of the newly generated children is selected for expansion. As a consequence the process is a last-in-first-out (LIFO) policy. The forward exploration is pursued until, for some reason, progress is blocked. If blocking occurs, the process resumes from the deepest of all nodes left behind. This policy works well when solutions are plentiful and equally desirable, or when there are reliable early warning signals to indicate an incorrect candidate direction.

If unchecked, however, this strategy can continue to probe deeper and deeper along some fruitless path. For this reason, depth-first algorithms are usually equipped with a "depth bound": a stopping rule that, when triggered, returns the algorithms attention to the deepest alternative not exceeding this bound. Thus the program backtracks under two conditions: the depth bound is exceeded, or a dead end is reached. The latter event occurs when a node fails to pass a test for some property that must hold true for any node on a path to a solution.

2. Backtracking

This is a version of the depth-first search that applies the LIFO policy to node generation instead of node expansion. Thus when a node is

selected for exploration, only one of its successors is generated. This new node, unless it is found to be terminal or a dead end, is again submitted for exploration. If the generated node meets a stopping criterion, the program backtracks to the closed unexpanded ancestor.

The main advantages of backtracking over depth-first are its storage economy and the fact that is minimizes the computation required to expand each node. This advantage is marred by being unable to use heuristic information for evaluating the candidate successors, as in the informed version of depth-first (discussed later).

More sophisticated backtracking strategies use "backmarking", with which, after meeting a dead end condition, they back up several levels at once. This is done by analyzing the dead end condition to see if the cause for that condition can be placed on one of the earlier ancestors along the traversal path.

3. Breadth-First

These strategies assign a higher priority to nodes at the shallower levels of the search space, progressively exploring sections of the tree in layers of equal depth. This is implemented by a first-in-first-out (FIFO) policy, by expanding the nodes which have been generated, but unexplored, for the longest time.

Unlike depth-first strategies, breadth-first search of locally finite trees is guaranteed to terminate with a solution if a solution exists. Moreover, it is guaranteed to find the shallowest possible solution. However, the breadth-first method must retain in storage the entire

portion of the tree it explores, to enable it to come back to expand nodes suspended earlier.

Informed Best-First Search

Informed strategies make use of heuristic information to aid the search for a solution. The most natural stage for using heuristic information is in deciding which node to expand next. A similar decision is also taken in hill climbing strategies when selecting the most promising direction to proceed. However, what sets best-first apart is that it selects the best from among all the nodes encountered so far, no matter where it is in the partially developed tree. This is analogous to the frontier search method described in French [30].

The promise of a node can be estimated in a number of ways: by assessing the difficulty of solving the sub-problem represented by the node; by estimating the quality of the successors of the node; and by considering the amount of information gained by expanding the node. In all these the promise of the node is estimated numerically by a heuristic evaluation function.

The implementation of best-first strategies requires a trade-off between increased computation to provide a history of the nodes which have been explored, and possible duplicate description without this memory.

Hybrid Strategies

Characteristics of the three main search strategies described above, namely hill-climbing, backtracking and best-first, can be combined to achieve better computational performance within the memory requirements of the computer. In this manner the user can modify the number of

alternatives considered in each decision and the degree to which the search strategy allows recovery from disappointing search avenues to reaccess previously suspended alternatives.

2.3.3 Cutting Stock Algorithms

Cutting stock problems arise when a requirement exists for a number of lengths of material to be cut from larger pieces, such as in the paper and steel industries. The objective is typically to minimize the cost of meeting the requirements, usually by minimizing the resultant waste. A distinction is drawn between integer problems, in which the demand for a required length must be met by whole pieces, and fractional problems, in which the required length can be met by summing a number of pieces not necessarily whole. This distinction has practical significance due to the difficulty of developing solution techniques for the integer problem.

Typical solutions to one-dimensional cutting stock problems can be pictured diagrammatically, as in Figure 2.2. A particular form of the one-dimensional cutting stock problem is when each pattern must contain a set number of distinct items, and it is this form which provides the analogy with pipeline scheduling. For example, a 4 pipeline problem and a cutting stock problem in which 4 items must constitute each pattern have a conceptual equivalence whereby the mix of products on the pipelines can be related to a cutting pattern. In the same way that longer run lengths of patterns result in less knife setups, increased pumping time of product mixes requires less setups. In both systems setup times adversely affect system performance. Hence encouraging

"patterns" with large usage levels minimizes the time lost due to changeovers.

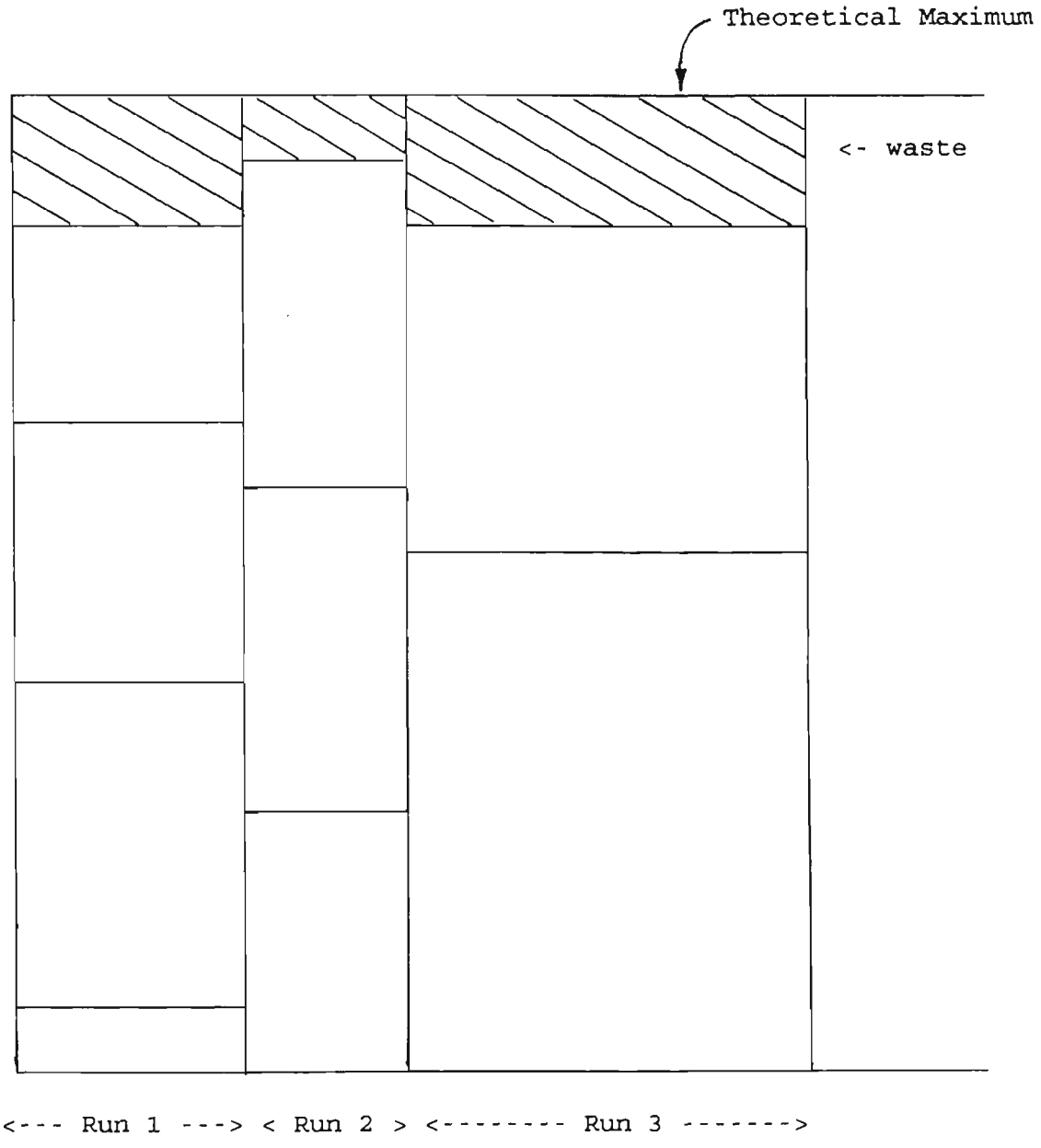


Figure 2.2 Cutting Stock Representation

Much of the work on cutting stock problems has been directed towards maximizing run lengths or minimizing the number of distinct patterns. Because obtaining solutions with as few changeovers (requiring setups) as possible is one of the major objectives in pipeline scheduling solution methods for cutting stock problems is examined below. This was a major motivation for the methods used in Chapter 4 and 5.

Gilmore and Gomory developed a linear programming approach to the solution of cutting stock problems (Gilmore [34]). However, a conventional LP formulation of a cutting stock problem results in a matrix with many columns, and an LP approach to an integer programming problems results in a matrix with many rows. Thus a traditional LP approach to the integer cutting stock problem results in a very large matrix, and a corresponding difficulty in finding exact solutions (Gilmore [34]). Lasdon [52] notes that problems involving matrices with many columns often have a large number of near-optimal solutions prior to optimality, imposing serious problems for their solution.

Column Generation Techniques

A proposal for overcoming the difficulty of the large number of columns was to generate a library of only useful columns, by some measure of "useful", and to use only these columns in the matrix. However, it is difficult to find measures of "usefulness" that sufficiently restrict the size of the matrix without distorting the solution (Gilmore [34]). This problem was eliminated by recognizing that the pricing out step of the primal simplex LP algorithm could be replaced by the solution of a knapsack problem. (The knapsack problem is so named due to its analogy to filling a knapsack with objects of the greatest total worth subject to the weight limitation that a hiker can carry.) The method depends

heavily on being able to rapidly solved knapsack problems, and fortunately several effective algorithms are available. Effectively this approach enables one to implicitly consider all the possible columns while holding only a few at any one time.

Heuristic Modification

Johnston [44] and Haessler [37] note that the primary difficulties of the LP model of the cutting stock problem are related to the lack of control over small batch sizes and the excessive number of batch sizes, and hence setups, which result. Inherent also in the approach is the difficulty of rounding fractional solutions to feasible integer solutions. Haessler notes that when the width of the average order is small relative to the width of the master roll the LP approach is often unsuitable, due to the excessive number of patterns generated. Instead, a sequential pattern generation technique, which heuristically selects patterns with high usages, is better. However, when the width of the master roll is a small multiple of the width of the average order, these sequential methods may perform poorly and approaches using an LP solution as a starting point are better.

Johnston [44] uses heuristically based penalty methods, within the LP framework, to encourage the generation of low cardinality solutions (that is, those with a small number of batches with high usage levels). This is achieved by maximizing the total potential maximum batch size of patterns, while keeping the run length constant and the waste within set limits. (The potential maximum batch size is the maximum number of times a particular pattern may be used.) In itself this maximization is not sufficient because several patterns with equally large potential maximum batch sizes could remain in the solution at usage levels well

below their maximum. Thus, several heuristics are used to differentiate between patterns with similar potential maximum batch sizes.

One successful approach has been to embody these heuristics in a tree-search procedure (Johnston [45,46]). The structure of the tree, branching technique and backtracking procedure used are shown in Figure 2.3. This method has been adopted in a similar way in the algorithm described in Chapter 4.

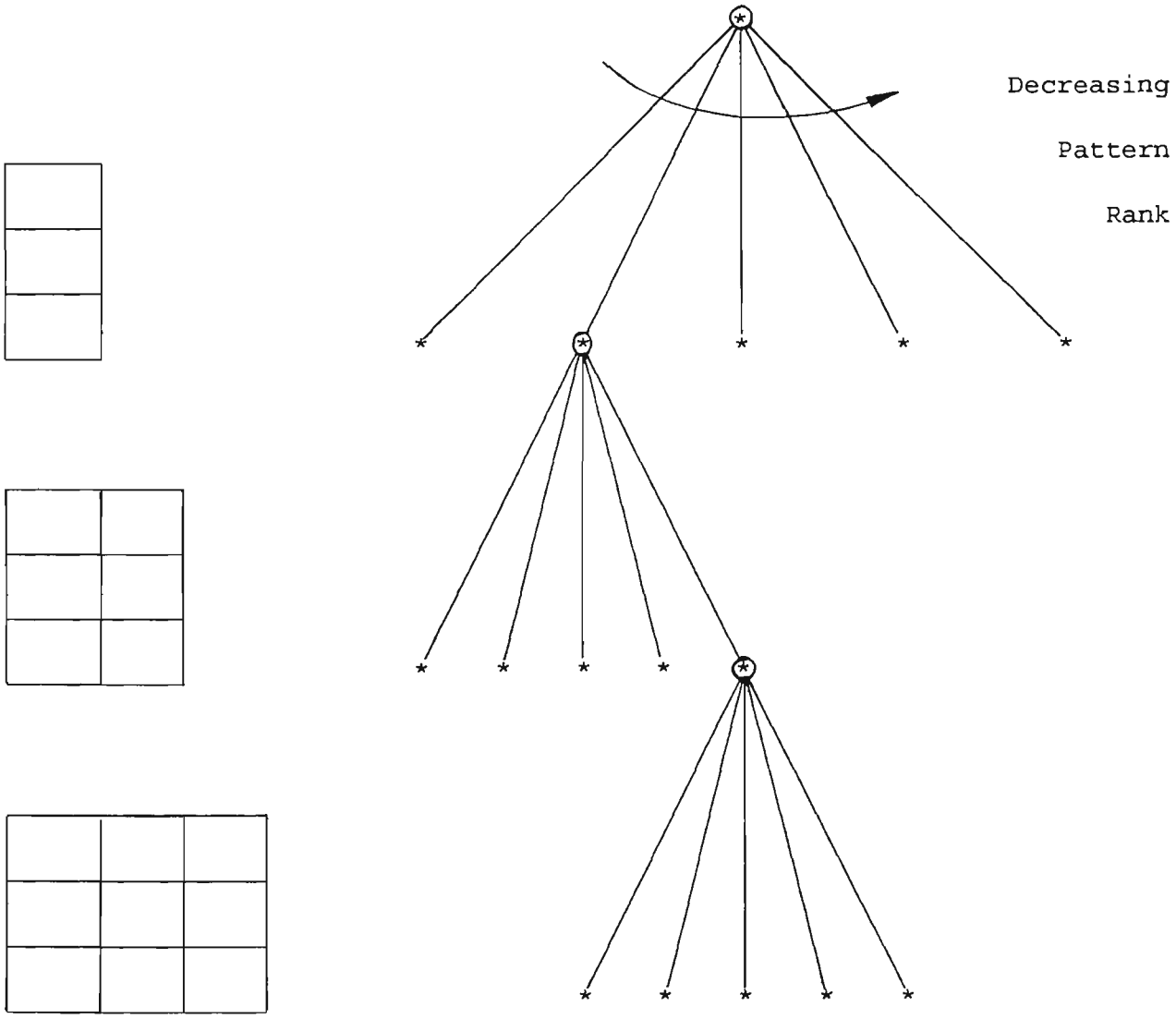


Figure 2.3 Tree-Search With Backtracking

In this approach a set of patterns, representing the different cutter arrangements, are created. The "best" pattern is selected according to some performance measure, such as minimum waste, and is added to the partial schedule. This is represented by a branch of the tree. A new set of patterns is then created at the next node of the tree and the selection and scheduling process is repeated.

In this way the schedule is built up by progressively adding patterns to the partial schedule. The unused patterns may be stored for use in later backtracking. During backtracking the algorithm selects a point in the schedule from which to reschedule and selects the best untried pattern at this point. This pattern is then added to the partial schedule and the pattern generation, selection and scheduling process recommences. A mechanism is employed to decide when backtracking should cease, such as when no further improvement in the solution quality appears to be easily obtainable.

2.4 Scheduling Rules Approach

Scheduling rules, otherwise known as priority rules, are disciplines for choosing the order in which jobs are selected for scheduling. Panwalker and Iskander [54] list over 100 reported rules. They classify these into three main categories:

1. Simple Priority Rules, which are based on information relating to each specific job, such as its due date, processing time, etc. Random dispatching is also included in this category, as are combinations of these rules.

2. Heuristic Scheduling Rules, which involve more complex considerations, such as anticipated machine loading, effect of alternate routing, etc. These rules are usually used in conjunction with simple priority rules.

3. Other Rules, which involve rules designed for a specific shop, or combination of priority indexes based on mathematical functions of job parameters.

They note the consensus among researchers appears to be that a combination of simple priority rules, or a combination of heuristics with a simple priority rule, works better than individual priority rules alone. They report that simple priority rules are the most commonly used (70 %), followed by "other" (20 %) and "heuristic" (10 %).

Priority Rules

Vepsalainen and Morton [70] summarized the principal features of the major priority rules as follows:

1. The First Come / First Served (FCFS) rule is easy to implement, but such a backward looking approach is detrimental to most performance criteria.

2. The weighted Shortest Processing Time (WSPT) and SPT rules tend to reduce tardiness in congested shops by giving priority to shorter jobs, and frequently achieve a remarkably low total number of tardy jobs without using explicit due date information.

3. The Earliest Due Date (EDD) rule emphasizes job urgency. Arkin and Roundy [3] note that EDD sequences do not do well for general weighted tardiness problems. However, they work well when the weights of the jobs are roughly proportional to the processing times, and when very little tardiness occurs.

4. The Minimum Slack (MSLACK) rule applies additional processing time information to the EDD rule, but in a way that counteracts the SPT rule: of several jobs with the same due date, priority is given to the longest job. Hence some schedulers have experimented with ratio rules, such as Slack per Remaining Processing Time (S/RPT), which tend to compensate for the anti-SPT tendencies of the MSLACK rule.

5. The Modified Due Date (MDD) rule, which adjusts the due date to the earliest possible completion time once the job becomes critical

(once the slack becomes negative), is very efficient for non-weighted tardiness scheduling.

They note that the common priority rules are known to be deficient at certain shop load conditions. For example, Earliest Due Date, Minimum Slack and Slack per Remaining Processing Time rules perform reasonably well with light load levels but deteriorate in congested shops. On the other hand, the most common priority scheduling rule, the Shortest Processing Time rule, fails with light load levels and generous due data allowances.

Heuristics

Gere [33] conducted a thorough simulation of a job shop to test the performance of selected priority rules and heuristics. He concluded that "alternate operation" and "look ahead" heuristics significantly improved schedules when used to augment a reasonable priority rule. In fact he suggested that the selection of the priority rule itself was relatively unimportant. Moreover, a "poor" rule when augmented with a suitable heuristic is better than a "good" rule without one, and he observed that heuristics tend to reduce the difference in performance between priority rules. The two heuristics noted above were more effective when used together than either one used alone.

Because there is little difference in effectiveness of the priority rules after they are combined with two or more heuristics, Gere recommends that a simple rule should be used in preference to a complex rule. Of the simple rules tested, non-random rules were significantly more effective than random rules (first-in-first-out (FIFO) rules were also considered random). There appeared to be little difference in

effectiveness between rules based in some reasonable way upon job slack, with the simplest rule (job slack) at least as good as more complex rules (job slack per operation, job slack ratio, modified job slack ratio).

Gere contended that dynamic rules (those which are regularly updated) are generally more effective than static rules (those which determine priorities before scheduling begins), because dynamic rules take into account additional information not available to the static rules. In addition, he believed it reasonable to expect that effective priority rules will take into account due dates or processing times or both. Furthermore, because each operation is another opportunity for delay as the job waits in a queue at a machine, the number of operations appears to be relevant. Also, machine loading should be a significant factor, because expected delays are greater on heavily loaded machines.

The significance of the heuristic augmentation on schedule performance warrants a brief description of the heuristics used. The "Alternate Operation" procedure revoked the selection of a simple priority rule when that rule caused another job to become "critical" (that is, tardy, or close to becoming tardy). "Look Ahead" checked whether a critical job was due to reach the machine before the selected operation was completed, and if so the selected operation was dropped and the machine idled. "Insert" searched for the longest operation which could then be scheduled in the available idle time, without delaying the critical job.

More complex heuristics were proposed ("Time Transcending Schedule", "Subset of Critical Jobs", etc), but Gere contended that they were

probably no more effective than the simple heuristics above, whilst being much harder to program and expected to take longer to solve.

The application of scheduling rules as they relate to the specific characteristics of the pipeline scheduling problem are discussed in the following sections.

2.4.1 Job Characteristics

Ready Times

Posner [56] considered single machine sequencing of clustered jobs, in which jobs within each cluster had similar, though not necessarily the same, release dates, which were distinct from those in other clusters. Obviously clustering is exhibited in the pipeline problem by jobs which are sourced from the same product batch. However, Posner assumed precedence relationships existed between clusters, such that no job from a later cluster could be scheduled until all the jobs from the earlier cluster had been completed. In addition, his method was directed at ordering the jobs within each cluster, and hinged on the release dates of these jobs being different.

Baker et al [6] investigated preemptive scheduling of a single machine subject to general release dates. Their objective was to minimize the total cost which was a monotone nondecreasing function of the job completion times. Their approach was to decompose the schedule into blocks. A block was defined as a minimal set of jobs processed without idle time, such that each job not in the block is completed not later than the first job in the block is released, or not released before the last job in the block has been completed. Blocking thus enables the

schedule to be broken into sub-problems, and jobs within each block scheduled independently of jobs in other blocks. Preemption is essential for their algorithm to determine in advance the block structure upon which the optimal schedule is based. Their results can't easily easily be extended to accommodate the resource constraints, sequence dependent setup times and multiple machines of the pipeline scheduling problem at hand.

Due Dates

Historically, job delivery with regard to due date commitments has concentrated on tardiness; that is, performance only becomes an issue once the job is late. However, the increased significance now placed in Just-In-Time (JIT) production methods often results in earliness as well as tardiness being discouraged. Baker and Scudder [7] observe that the vast majority of articles on earliness / tardiness (E/T) problems deal with single machine models, although some single-machine results have been extended to parallel machines (these are frequently identical machines, although uniform machines have occasionally been considered).

They suggest that problems with distinct due dates appear to be intrinsically different from solutions to problems with a common due date, due to differing properties of the optimal schedule for each problem class. They observe that there has been less progress with distinct due dates, and in general, it appears that only branch and bound techniques have been effective at solving problems in this class. Moreover, E/T criteria are likely to require comparatively sophisticated scheduling procedures. Dispatching procedures (that is, those which allow a scheduling decision to be made in real time when a machine

becomes idle), even sophisticated ones, may be inappropriate as effective solutions may need to make use of inserted idle time.

Job Priorities

Vepsalainen and Morton [70] note that mainstream research in priority dispatching has considered jobs with equal delay penalties, thereby ruling out strategic differentiation of customer orders. They analyzed dispatching rules based upon greedy heuristics in which priority is given to the job with the highest expected delay cost per imminent machine processing time. For a job with some slack, the expected tardiness cost is reduced according to a look-ahead feature. They found that the resulting "Apparent Tardiness Cost" (ATC) rule dominated the competing rules (first come/first served, earliest due date, shortest processing time, etc) tested in all shop load conditions studied. They found that the results, achieved with predetermined parameter values for look-ahead and lead time estimation, could often be improved by adjusting the parameters on the basis of shop load.

The major differences between their assumptions and the conditions of the pipeline scheduling problem included: no preemption, no sequence dependent setup times and continuous availability of machines.

1. Apparent Tardiness Cost

The ATC priority index for job i in a single stage shop at time t is as follows:

$$ATC(t) = V_i / P_i * \exp(-S_i / (k*P))$$

where:

$$S_i = \max((D_i - P_i - t), 0)$$

and:

ATC(t) = apparent tardiness cost index at time t

V_i = tardiness cost per unit time of job i

P_i = processing time of job i at time t

D_i = due date of job i

S_i = positive slack of job i at time t

k = constant

$\bar{P}$ = average processing time of the waiting jobs

The constant k is a look-ahead parameter which scales the slack according to the expected number of competing jobs. Vepsalainen and Morton recommended a value of k=2 for static flow shops and k=3 as a reasonable average for dynamic flow shops. They suggested the value can be adjusted within the range of $1.5 < k < 4.5$ to reduce the weighted tardiness costs in extremely slack or congested shops.

2. Cost Over Time Rule

In addition to the ATC rule, Vepsalainen and Morton discussed the performance of the Cost Over Time (COVERT) rule. This rule performed well in minimizing weighted tardiness, although not quite as well as the ATC rule in minimizing the number of tardy jobs.

The COVERT priority index for job i in a single stage shop at time t is as follows:

$$COVERT(t) = V_i / P_i * (1 - (S_i / (k*b*P))^+$$

The constant b is a multiplier which relates the processing time to the expected waiting time. They used a value of b=2 and k=2 for their experiment. The "+" sign indicates that only positive values are taken.

Earliness/Tardiness Penalties

Baker and Scudder [7] note that different earliness and tardiness penalties (for the same job) are often observed in practice. This is because the earliness penalty tends to be endogenous (that is, it is set within the "shop", usually with the aim of minimizing inventory holding costs), while the tardiness penalty tends to be exogenous, due to the interaction with the customer. The primary role of both penalties is to guide solutions towards the task of meeting all due dates exactly.

2.4.2 Machine Characteristics

Parallel Machines

Three primary categories can be envisaged for parallel machines.

1. Identical machines, in which each machine is the same in terms of the tasks which can be processed and the rate at which these tasks can be processed.

2. Uniform machines, in which the machines differ only in their processing speed. Hence the time taken to process a task on one machine is related to the time taken on another machine by a "speed" factor.

3. Unrelated machines, in which the machines differ in the tasks which they can perform, and may also differ in their processing rate for similar tasks.

The second two categories are collectively referred to as non-identical machines.

The decreasing lack of symmetry as one moves from category 1 to category 3 corresponds to a decreasing level of research reported in the literature. Graves [35] notes that few of the simple results and algorithms from the single machine case can be carried over to the parallel machine problem. Most of the results obtained for the parallel extension assume that the machines are identical, and less frequently, uniform. The most common criteria studied are weighted flow time, maximum flow time (makespan) and weighted tardiness.

Horowitz and Sahni [40] provide exact and approximation algorithms to minimize finish time (makespan) and weighted mean flow time for selected two machine problems, for both uniform and non-identical machines. They state that these methods can readily be extended to problems with more machines. Whilst their exact algorithms are exponential in the worst case, and hence are limited in the number of tasks which can be handled in reasonable time, their approximation algorithms are claimed to be linear or quadratic at worst. Of interest is that these approximation algorithms can guarantee an answer within a user specified error of the optimal. As expected, the solution time increases as this error is reduced.

The simplifying assumptions on which the above solution is based preclude the direct use of these algorithms to the pipeline application. Specifically, these methods assume non-preemptive tasks, zero ready times, no sequence dependent setup times, no resource constraints and continuous availability of all processors.

Van de Vel and Shijie [69] applied bin-packing techniques to uniform processor scheduling. No computational results were provided, although they note that their algorithm is non-polynomial. Their search algorithm, which checks the feasibility of schedule configurations, is not discussed in detail. Adams et al [2] focused attention on the bottleneck machine, progressively shifting through all the machines in the shop. Dror et al [24] investigated the case in which the processing rate of individual machines was inversely proportional to the number of machines simultaneously at work.

Machine Downtime

Most research assumes the continuous availability of machines, and French ([30], p 201) notes that only a limited amount of work has been done for problems involving planned downtime of single machine processing.

2.4.3 Resource Characteristics

Three resource categories are generally recognized (Talbot [66]):

1. Renewable, in which the resources are available in limited quantities each time period. For example, labour.

2. Non-renewable, in which the total consumption of the resource over the life, or part of the life, of the project is constrained. For example, project capital.

3. Doubly constrained, in which the per-period and total availability are limited.

In the problem at hand the resources belong to the first category, because the use of suction and offtaker pipelines is restricted in each time interval. Moreover, these restrictions are imbedded in time window constraints. Outside these windows processing is not possible for jobs which require resources which are unavailable; inside the windows jobs contend for the resources required before processing can commence.

Most literature dealing with time window constraints appears to be directed towards vehicle routing, while machine scheduling usually assumes continuous machine availability. Solomon [62] considers the design and analysis of vehicle routing algorithms with time windows, noting that heuristic approaches are favoured due to the computational hardness of the problem class. Although the vehicle routing results are not directly applicable to the pipeline problem they demonstrate the application of insertion heuristics, and their usefulness in problems involving tight time windows.

French ([30] p 198) notes that in general resource constrained scheduling problems are very difficult and usually NP-hard. Talbot and Patterson [65] present an integer programming algorithm with network cuts, aimed primarily at project scheduling problems involving precedence constraints. They suggest this method provides a reliable

optimization technique for problems with 30 to 50 jobs and three different resource types. In their method the network cuts are integer time periods which are used in elimination rules to remove partial schedules that cannot possibly lead to improved solutions.

2.4.4 Assignment Characteristics

Setup Times

Wilbrecht and Prescott [73] used simulation to assess the effect of different scheduling rules involving setup times on job shop performance. They concluded that for shops working at full capacity, giving priority to jobs which did not require setup produced the best performance based upon their criteria. (They actually selected jobs with similar setups, but defined the machine setup as the difference between the setup of the job most recently processed and the job about the be processed. This resulted in zero machine setup for jobs with equal relative setup.) Defining job processing time as the sum of run time plus setup time, they also noted that both "longest processing" and "shortest processing" rules were better than "longest run" and "shortest run" rules. For shops with excess capacity they surmised that setup times probably have less influence on shop performance.

So [61] presents three heuristics for scheduling jobs on parallel identical machines with setups. A primary assumption is that the jobs with the shortest processing time are assumed to have the highest rewards. Thus the optimal schedule can be determined by selecting jobs with the shortest processing times. He concluded that the "greedy" heuristic provided the best performance/time trade-off. In addition to performing well on all test cases, and producing the best solution in

many of these, it ran much faster than the sequential and decomposition approaches, both of which used a dynamic programming procedure.

Preemption

Preemption is said to occur when the processing of a job that has been started is stopped before completion (Conway et al [20]). Different kinds of preemption are possible depending on the treatment of the interrupted job when it returns to the machine for further work.

In preempt-resume, processing continues where it left off without any extra work or time being incurred due to the interruption. Thus the processing time of the job is constant, independent of the number of interruptions.

In preempt-repeat, the benefit of any processing that has been done is forfeited with the interruption. The stated processing time is therefore a minimum, achieved only when the job is processed without interruption.

A commonly occurring discipline between these two extremes involves preempt-resume operation for the processing time with preempt-repeat for the setup time. This is the discipline observed in the pipeline system studied.

One primary difference between preempt-repeat and preempt-resume is that in the former advance knowledge of job arrivals has a definite effect on the schedule. If there is no job that can be completed before the next (higher priority) arrival and withstand its preemption, then the machine should probably remain idle until that arrival.

Conway et al [20] noted that no general optimal procedures have been offered for preempt-repeat circumstances. Baker et al [6] examined preempt-resume scheduling of a single machine with release dates and precedence constraints. No computational results were provided. Federgruen and Groenevelt [26] used similar-path augmenting techniques to maximize spare capacity on preemptive machines.

Precedence Constraints

Although precedence constraints rarely occur in the system studied, situations can be envisaged in which the schedule must recognize this form of relationship amongst jobs. Baker et al [6] approached the precedence constraint by modifying the release date of jobs that must wait until other jobs had finished, such that the modified release date was greater than the completion time of the other jobs. In this manner explicit recognition of the precedence relationship during schedule preparation was not required.

Splitting Between Machines

The literature search conducted uncovered no results which dealt with situations in which individual jobs could be processed by more than one machine simultaneously. Instead, the common assumption is that each job constitutes an indivisible entity which can only be assigned to a single machine at any particular time, although operations on more than one machine, each as a separate stage, are often required.

Inserted Idle Time

Idle time becomes relevant when preemption exists. If a preempt-resume operating mode exits with no minimum processing requirement, then no advantage is gained if a machine is left idle, providing that resource

constraints do not detract from another machine's performance. However, if preempt-repeat operation exists, or any of the above qualifications fail, then the machine should probably remain idle until a higher priority job becomes available (Conway et al [20]).

Schedule Generation Techniques

French [30] notes three classes of schedules:

1. Semi-Active, in which each operation is started as soon as it can, within the technological constraints and the defined processing sequence.

2. Active, in which the processing sequence is such that no operation can be started any earlier without delaying some other operation or violating a technological constraint. These form a sub-class of the semi-active.

3. Non-Delay, in which no machine is kept idle when it could start processing some operation. These form a sub-class of the active, and hence also of the semi-active.

Obviously each sub-class is smaller than its parent class, reducing the number of possible candidates. French [30] notes that, although the optimum schedule is not necessarily non-delay, strong empirical evidence suggests that algorithms should concentrate on only this type. However, commonsense suggests that this strategy is not necessarily relevant for problems with preemption and setup times.

2.4.5 Matchup Scheduling

Revision or modification of existing schedules probably occurs more frequently than generating new schedules from scratch, particularly in situations with a rolling horizon. Despite this, almost all reported work deals with the later task, unencumbered with the constraints imposed by previous decisions. However, these prior decisions constitute an agreement between provider and client regarding delivery expectations and cannot simply be ignored. Clearly, clients are unlikely to willingly restructure their plans to accommodate a change sought by the scheduler to fulfil a new or revised demand of another client. Thus the scheduler must usually work within the existing constraints, albeit with some possible alteration of "soft" restrictions, when revising the schedule to account for changing circumstances.

Bean et al [12] note that "failure to recognize the ongoing nature of the problem constitutes as significant over-simplification" of the task. Their work on matchup scheduling presents a method for adapting a preplanned schedule (known as a "preschedule") to a changing scheduling environment. Their strategy is to follow the preschedule until the disruption occurs. They then reschedule part of the preschedule to accommodate the disruption such that the revised schedule matches up with the preschedule at some specified future time. The revised portion of the schedule seeks to compensate for the disruption to minimize the total tardiness.

This method assumes that the disruptions are spread far enough apart to allow matchup to occur between them. In a system with ample and well

distributed slack resources matchup can be quickly achieved and this assumption is usually valid. However, if the disruptions occur too close together then a preschedule will not generally be useful and complete rescheduling is probably more appropriate.

In practice, determining the matchup time and rescheduling from the disruption to the matchup time is accomplished heuristically. Bean et al assume a future time and seek initially to matchup within a specified cost threshold (such as a given increase in tardiness), rescheduling only the machine upon which the disruption occurs. If the cost exceeds the threshold they increment the future time and retry, until the threshold is achieved or some maximum time is exceeded. If the maximum time is exceeded they reallocate the jobs amongst the available machines, reset the future time to the initially assumed value and reschedule. During the matchup phase the algorithm checks for job interchanges using six different ordering rules and selects the lowest cost solution.

Four distinct matchup strategies are used:

1. Pushback: in which machine assignments and job sequences stay the same, and only the start and finish times shift to accommodate the disruption.

2. Dynamic: in which priority rules, such as earliest due date and modified due date, used to reorder the jobs.

3. Total Reschedule: which attempts more optimization than Dynamic by involving full rescheduling of all jobs and ignoring preschedule assignments.

4. Matchup: which seeks to maintain preschedule assignments as much as possible, switching job/machine assignments as needed to correct for the disruption.

Bean et al concluded that assuming the original schedule were optimal, then if the disruptions are sufficiently spaced over time the optimal rescheduling strategy is to matchup with the preschedule. They found that a MIP approach for multi-machine lot reassignment was better than a priority rule approach when the machine utilizations were high, but took much longer.

2.5 Other Methods

2.5.1 Constructive Algorithms

Many successful methods have been demonstrated for obtaining optimal solutions for single machine problems (Conway et al [20], French [30]). These constructive algorithms build an optimal schedule by following a set of rules which exactly determine the processing order. However, their extension to multiple machine shops has not been as successful. In both flow shop and job shops only a few specialized cases with two or more machines are amenable to such analysis (French [30], p66).

2.5.2 Simulated Annealing

Abramson [1] investigated the application of simulated annealing to constructing school timetables. This is a Monte-Carlo based technique which simulates the cooling of a collection of hot vibrating atoms. When applied to scheduling the jobs replace the atoms and the system energy is replaced by the schedule cost. The "temperature" as the system "cools" is used to control the probability of an increase in cost. One of the advantages of simulated annealing over greedy algorithms (for example, hill climbing) is that it is less likely to get caught in local optima, because the cost can increase as well as decrease.

Abramson noted some extremely long solution times (up to 14 hours for one problem, although most took a number of minutes). This is not surprising given that the schedule candidates are selected at random. In addition, the solution obtained was a function of the random number starting seeds, and thus usually more than one run was required to find the best solution. However, the algorithm was able to incorporate complex technological constraints explicitly.

2.5.3 Expert Systems

Although the writer was not aware of any applications of expert systems to pipeline scheduling when the literature search commenced in 1985 several applications were reported in the late 1980's and early 1990's (Brazile and Swigger [13], True [68]). By this time work was well progressed on a heuristic approach to the problem and hence the expert systems approach was not evaluated in more detail. However, some

reported applications which demonstrate promise in providing practical solutions are briefly noted.

True [68] discusses two applications of petroleum products pipeline scheduling operations. The expert systems incorporate the "rules of thumb" that experienced schedulers use to manually prepare rolling schedules for long distance pipeline systems with multiple offtakes. In one example, prior to the introduction of the expert system, manual preparation of 5 day schedules typically took 4 to 5 hours. The current prototype scheduling system, applied to one of four major pipeline segments, devises 35 day schedules in about 8 minutes. When complete the expert-system is expected to have about 1000 rules incorporated into the knowledge base.

Since 1991 an expert system solution has been used to schedule shore-to-ship transfers of lubricants at Mobil Oil Australia's Adelaide Refinery. This mini computer based system determines the sequencing of transfers on a single pipeline to minimize setups and leave the pipeline filled with a specified product. All transfers have the same release date and due date.

2.5.4 Critical Path Methods

Taha [64] provides a basic introduction to project scheduling using the Critical Path Method (CPM) (also known as the Project Evaluation and Review Technique (PERT)). This approach is particularly suited to projects which consist of interrelated tasks with many precedence relationships. In these type of problems the precedence relationships are typically more problematic than the resource constraints. This is

the reverse situation of most production scheduling problems in which the machine constraints figure more prominently than job precedence considerations.

Stinson et al [63] applied a multiple resource constrained branch and bound procedure to a number of project, flow shop and job shop scheduling problems. Their results highlighted the differences in the nature of these problem classes, which the success in solving project scheduling problems not evident in large job shop problems. They noted that the performance of their procedure was significantly influenced by the critical task sequence which was dependent on upon precedence constraints.

2.6 Performance Evaluation

2.6.1 Solution Quality

The computational complexity of most practical scheduling problems precludes guaranteeing that the optimum solution can or has been found. Practitioners are therefore forced to accept methods which find feasible solutions. With this recognition that approximation methods may provide sub-optimal solutions comes the need to determine the quality of the solutions generated.

Instance Evaluation

A traditional method for evaluating approximation algorithms had been to run them on selected problem instances; that is, on particular combinations of tasks and constraints. Garey et al [31] note that for

some algorithms this is still the only practical approach. However, the major drawbacks of this method are the difficulty of selecting realistic sample problem instances and the difficulty of obtaining an absolute performance measure of the algorithm. They note that this approach is better adapted to comparing alternative heuristics than to determining how "near-optimal" each algorithm is. In most cases researchers have had to append to their results a disclaimer on their ability to transfer the results to other shops with different conditions.

Simulation

Simulation is essentially the same as instance evaluation; the principal distinction being that simulation is typically conducted using synthetically derived problems: jobs arrivals, processing times and due dates are drawn from random or specified distributions. Conway et al [20] note that simulation has been approached from two directions. First, by investigators attempting to extend theoretical results and, secondly, by practitioners seeking to pre-test procedures before applying them to actual situations.

In typical test cases attempts are made to measure the equilibrium or steady state performance of algorithms and run-in and run-out periods are excluded from sampling results. Conway et al note that there appears to be no evidence to suggest that the use of actual shop data and dimensions significantly alters the comparative performance of key procedures. On the otherhand Crowder et al [21] note a number of advantage and disadvantages of hand selected and randomly generated test problems with respect to experimental results. Principally this relates to the uncertainty of generalizations based on hand selected problems due to the frequently unknown problem population. However, individual

problems are usually representative of real-world behaviour, whereas randomly generated problems typically are not.

Probabilistic Analysis

An alternative approach is to use probabilistic techniques to derive the expected values of the approximation algorithm and of the optimum solution. However, it is often difficult to determine a probability distribution that can be dealt with mathematically and which simulates the problems instances that arise in practice (Garey et al [31]). Moreover, both this approach and that of instance evaluation above only provide an indication of the average performance of the approximation algorithm.

Performance Guarantees

The performance guarantee approach overcomes this disadvantage by providing a bound on the worst case behaviour of a particular algorithm. For a minimization problem this typically takes the following form (Garey et al [31]):

$$A(I) \leq r*OPT(I) + d \qquad \text{for all instances } I$$

where:

A(I)	is the value of the schedule found by the approximation algorithm for instance I
OPT(I)	is the optimal value of all feasible schedules
I	is an instance of the scheduling problem
r and d	are specified constants

In general the additive constant "d" is asymptotically negligible and the dominant factor is the ratio "r". The performance guarantee approach seeks to find the smallest value of "r" for which the above theorem can be proved, called the "worst case performance ratio" (or simply the performance ratio).

The performance ratio then provides information which supplements that obtained by other approaches, and provides a rigorously defined quantity with which different approximation algorithms can be compared. The drawback with this approach is that in practice an algorithm may perform much better than its worst case performance suggest, and that problem instances causing worst case behaviour may be unrealistic (Garey et al [31]). Haessler [36] states that a simple heuristic procedure guaranteed to do no worse than X % of the optimum most likely will not give results good enough to use in practice. In addition, he considers that an industrial grade procedure that provides usable results will be far too complex to have any theoretically provable statement made about its performance.

Instance Evaluation Revisited

Haessler [36] concludes that the true benchmark of any algorithm is the quality of the solutions generated by whatever procedures are currently in use, such as the "ad hoc" procedures used by people to solve such problems. This of course implies a qualitative assessment of the algorithm's performance. Given the complexity of the scheduling process for most practical situations and the variety and "fuzziness" of the objectives, it is unlikely that a single parameter, or a number of parameters, could adequately measure schedule quality. Thus qualitative assessment seems an entirely reasonable approach. Haessler [37] defines

a good solution as one that cannot easily be improved by someone who is knowledgeable about the problem.

Haessler [36] advocates testing of algorithms on a set of representative sample problems. Care must be exercised to ensure that the problems selected are representative. What is sought is a procedure that is robust: one that will work well on a variety of situations. In situations in which the algorithm yields inferior solutions to the existing method, the reasons should be analyzed and a decision made on whether to modify the algorithm to deal with the causes. In addition, various control parameter settings, decision rules and multi-pass approaches can be tested to improve the effectiveness of the procedure.

2.6.2 Computational Performance

The performance of different algorithms on particular scheduling problems is difficult to quantify, due to factors such as problem size, type of constraints, nature of the objective function, and the wide variety of computers on which the algorithms are implemented. Rarely are all the experimental parameters recorded to enable the results to be reproduced. Crowder et al [21] noted this difficulty, and discuss the requirements for designing and reporting computational experiments.

Test Problems

Barker and McMahon [11] and Carlier and Pinson [16] note that the problems proposed by Muth and Thompson have become the defacto standard for testing of general job-shop algorithms. In practice, however, algorithms are usually developed to solve specific scheduling problems.

Thus these standard test problems may or may not be of use, depending on the characteristics of the specific situation being investigated.

Hardness

Potts and van Wassenhove [57] report that researchers have observed that problem "hardness" depends on two parameters: the relative range of due dates (R) and the average tardiness factor (T). They concluded that problems with relative processing times in the range (1,10) were substantially easier that those in the range (1,100). Many more dominance relations can be established in the former case because the processing times have a much larger probability of being equal. They highlighted that this observation was particularly important when comparing computational results, especially when dynamic programming is involved, because DP relies heavily on precedence relationships.

Optimality

Haessler and Talbot [38] observed that proving optimality consumed from three to ten times as much CPU time as did simply finding the optimal solution to a corrugator trim problem. They proposed using a stopping rule to terminate program execution when an improved solution is not found within a time period equal to three times the cumulative CPU time need to obtain the current best solution. They found that this rule almost always permitted their 0-1 integer programming algorithm to find the optimal solution without consuming an inordinate amount of time verifying optimality.

From a practical standpoint, schedulers in work place situations, faced with predetermined deadlines, are interested in how long schedule preparation will take. This is the total time, from gathering and

inputting the data, to solving and reporting the results. If a proposed method cannot offer the convenience and speed of the existing technique, then the new method is unlikely to be adopted, regardless of any touted improvement in schedule quality.

2.6.3 Human Factors

The "user-friendliness" of personal computer applications, however hackneyed the term may be, has raised the expectations of users of computer based systems. Users are probably more concerned with the effort required to obtain a solution than they are with the quality of that solution, particularly if other proven methods are available.

Obviously the presentation of a vendor supported product is likely to be more refined than most "in-house" applications. However, the primary concern is not the "bells and whistles" offerings of everyday worksheet and word processing packages, but in the ease of use of an algorithm's implementation. Baker [8] discusses the advantages of an interactive system that arose from a consideration of human factors in the process scheduling environment. He emphasises the benefits of a system that is neither "too conversational or overtly didactic". Instead, scheduling information was presented on the visual display unit (VDU) in spatial rather than numerical form, such as through Gantt chart format. This enabled the users to make full use of their innate pattern recognition capabilities. A "help" facility providing on-line descriptions of system functions was seldom used after the initial training phase.

Part of the basic design philosophy of Baker's [8] interactive scheduling system was the need to quickly return a solution to the user.

Even if the solution search had not finished the algorithm was interrupted after a predetermined number of subproblems had been evaluated and returned the best solution found to that point. The user was then given the choice of continuing the search, or of changing the solution or basic data. If the changes made by the user invalidated the branching tree, the model was regenerated and the branching algorithm started from scratch.

Apart from the above aspects relating to the implementation of scheduling algorithms, Baker [9] notes that, in general, algorithms must be designed to do what humans don't do well. Compared with computers, humans are good at abstraction and problem recognition. However, most human minds overload rapidly when working on complex combinatorial problems. People who deal routinely with such problems tend to formulate simplifying relationships or rules of thumb in order to deal with this complexity. Habit leads them to follow these rules of thumb in situations where they don't apply.

Baker [9] observed that any model of an operational scheduling problem, no matter how complex, is always incomplete. He concluded that these models can be much simpler if the users are provided with the capability to impose their knowledge on the scheduling process.

2.6.4 Economic Evaluation

Economic evaluation provides one means of assessing the overall performance of a scheduling solution, by reducing to a single dimension (that is, dollars) the components which constitute the objective function. Conway et al [20] note three principal types of costs that

can be affected by the decisions of jobs sequencing: those of inventory, utilization and lateness. Jones [47] recognizes a fourth category due to long promises, observing that businesses providing very long due dates lose sales to more responsive competition. He notes that there are essentially no costs associated with long promises until the promises get close to the length of the competitors' promises, at which point they increase rapidly. Jones assumed a cost function based on the leading edge of a sine wave in his analysis.

Inventory costs are recognized to be real, non-trivial and difficult to quantify (Conway et al [20]). Jones [47] assumes these costs to be linear with respect to the number of jobs in the shop. However, it is difficult in the problem at hand to envisage inventory levels driving scheduling decisions, except from the point of ullage levels. That is, the physical ability to store product is of major concern and outweighs the cost of holding a particular inventory level.

Facility utilization is probably the prime economic consequence of scheduling decisions in the pipeline system studied. The ability to avoid or reduce pipeline idle time and produce a short schedule time implies a procedure that will permit the facility to do more work. This minimizes the probability of the distribution system constraining the upstream operations of the refinery, which would result in lost production capacity.

Conway et al [20] note that in manufacturing systems the costs associated with lateness are seldom obvious and immediate. Instead, they are usually felt by the effect of the customers' displeasure on future business. They stressed that penalties arising from late

delivery of a particular job are really consequences of decision made on many other jobs, and it would be unreasonable to assign them to the particular jobs with which they are identified.

It is important to recognize the trade-off that exists between facility utilization and job tardiness, and the implication this has on overall system costs. Jones [47] observed that dispatching rules which performed well with regard to facility utilization, such as the "shortest processing time" rule, resulted in very large tardiness for some jobs. Conversely, rules based upon delivery performance (and hence using due date information in some way), such as the "slack per operation" rule, consistently experienced lower tardiness penalties but got less work out of the shop. Obviously the relative magnitude of the coefficients of utilization and tardiness costs (and the long promise and inventory costs) affects the perceived performance of any particular dispatching rule.

2.6.5 Job Related Performance Measures

Numerical measures of weighted tardiness depend on problem size (number of jobs and machines) and the distribution of processing times and delay penalties. Vepsalainen and Morton [70] proposed the following normalization to allow intuitive interpretation and comparison between studies:

$$NWT = WT / nmPV$$

where:

NWT = normalized weighted tardiness

WT = weighted tardiness

n = number of jobs

m = average number of operations per job

P = average operation processing time

V = average delay penalty per unit time

2.6.6 Machine Related Performance Measures

Utilization is a primary measure of machine performance. Jones [47] notes that although one might expect changing load conditions would be best served by changing the rules by which the jobs are assigned to the machines, little work appears to have done in this regard. Part of this is no doubt due to the narrow load range where differing performance can be observed: when there is too little work many machines may be idle regardless of the scheduling rule; too much work permits any rule to fully utilize the machines. Thus rules designed to improve utilization have their greatest relative advantage in a middle range. Jones considered that most shops in practice tended to operate with work-in-progress inventories which caused at most two to three percent idle time due to lack of work. In this region there is little relative advantage of any particular scheduling rule with regard to machine utilization.

2.6.7 Shop Related Performance Measures

Maintaining low in-process inventory is a traditional yardstick for scheduling rules (Vepsalainen and Morton [70]). Two different measures are commonly defined: Work-In-Process (WIP), which is the cost of holding jobs from the start of the first operation to the completion of

the last operation; and Work-In-System (WIS), which charges inventory until the due date, assuming that the customer would not accept early shipment due to JIT materials management. Thus:

$$WIP = \sum_{i=1}^{n} s_i*(C_i - a_i) \; / \; nmPS$$

and:

$$WIS = \sum_{i=1}^{n} s_i*(\max(C_i,d_i) - a_i) \; / \; nmPS$$

where:

s_i = size, job i

C_i = completion time, job i

a_i = starting time, job i

d_i = due time, job i

n = number of jobs

m = average number of operations per job

P = average operation processing time

S = average holding cost

However, a better assessment of an algorithm's performance can be obtained from plotting average job tardiness (or a related measure) as a function of capacity utilization. Vepsalainen and Morton demonstrate this approach for a number of scheduling rules under a varying date tightness. This enables an intuitive assessment to be formed based on

the anticipated range of shop conditions, rather than relying on the single measures.

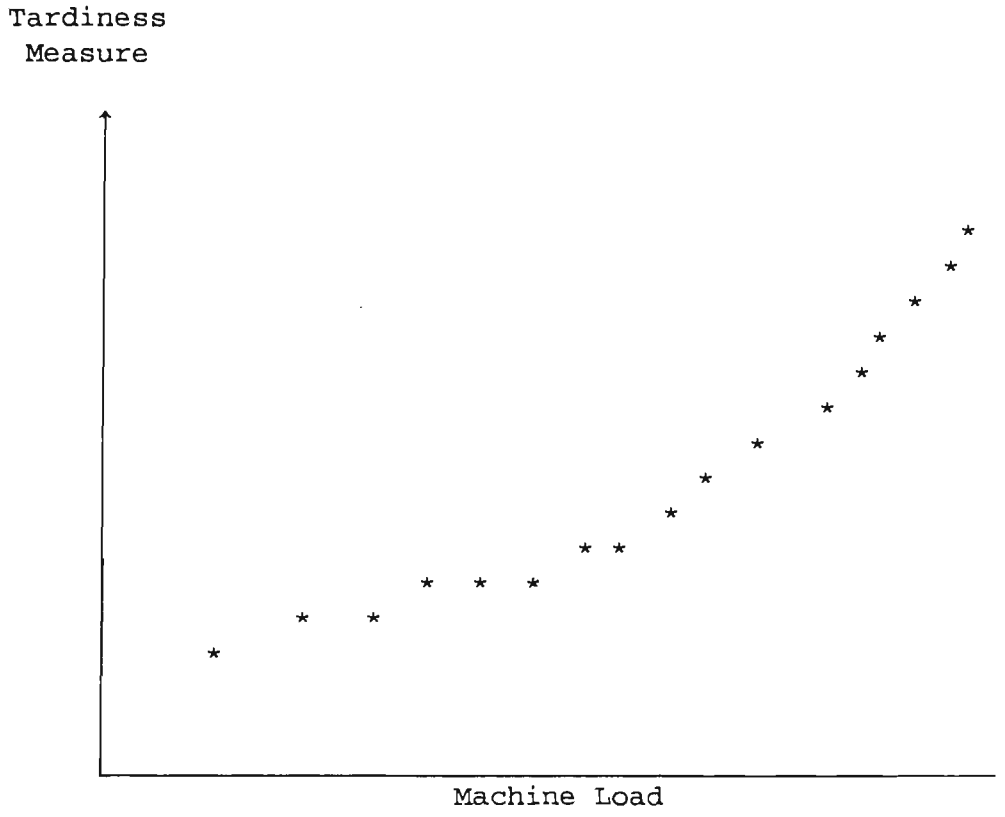


Figure 2.4 Algorithm Performance

Chapter 3 MODEL FORMULATION

3.1 Introduction

Williams ([74], pp 3-5) defines as model as a structure purposely built to exhibit features and characteristics of some object or system. The actual features and characteristics represented in the model depend on the use to which the model is put. In the case of a mathematical model this structure involves a set of mathematical equations which correspond to the real world relationships, such as technological constraints, physical laws, production capacities, etc, and an objective function, which measures the value of the problem solution and provides a means of comparing one solution with another.

This chapter discusses the characteristics of the pipeline problem that the model seeks to represent, summarizes the primary assumptions implicit in the model, and defines the relationships which link the characteristics and the objective function.

3.2 Intended Scope

As highlighted in Chapter 1, the scheduling system is composed of a number of major activities. The problems with the existing method centre around the manual schedule preparation task, primarily due to the time involved and difficulty of satisfying all constraints. This model attempts to address these weaknesses, but does not seek to solve other tasks which can be handled much more effectively by the scheduler, such as batch allocation and due date setting.

3.3 Assumptions

The major assumptions upon which the model is based are classified as follows:

Job Related Assumptions

1. Each job may be split into any number of operations for processing.

2. The ready time, due time and priority of every job are known in advance.

3. Although the processing time of each job is unknown (due to the processing options available), the quantity of product to be transferred is known in advance. The processing time can be derived from this once the processing options are defined.

Machine Related Assumptions

1. The machines are unrelated (that is, they differ in the jobs they can process and the rate at which they can process these jobs).

2. Each machine can only process one operation at any one time.

3. Each machine is only available during time windows, which are known in advance.

4. Machines may be idle.

Resource Related Assumptions

1. Three types of resources exist: suction pipelines, offtaker pipelines and offtaker terminals.

2. Some of the suction and offtaker pipelines can serve more than one machine at any one time, depending upon the jobs being processed on those machines at that time.

3. Each offtaker terminal is only available during time windows, which are known in advance.

4. The cost of using each offtaker terminal is a function of the position within the time window.

Assignment Related Assumptions

1. Sequence dependent setup times exist, depending on the job/machine assignment and the previously assigned job.

2. Preemption of operations is allowed under some circumstances. Preempt-repeat discipline holds for operation setup times, preempt-resume discipline holds for operation processing times.

3. Precedence constraints rarely occur and are ignored for simplicity. (If required these can be handled as discussed in Section 2.4.4).

4. Some jobs may be processed on more than one machine simultaneously.

5. No queue length or waiting time constraints exist. That is, there is no restriction on how long a job may remain in a queue waiting to be processed, and no restriction on the number of jobs waiting in the queue.

Other Assumptions

1. No randomness in job arrival, due times, processing rate and job quantities.

3.4 Nomenclature

The following nomenclature is used in the relationships explained below. Where possible these subscripts and variables are consistent with the nomenclature commonly reported in the literature. Additional subscripts and variables have been introduced to handle specific characteristics not commonly reported.

Subscripts

i	job	i = 1,n
j	machine	j = 1,m
k	operation (of job i or machine j)	
r	resource	
c	clique	
t	time period	

Job Related Variables

A_i	tardiness weighting
B_i	priority
C_i	completion time
D_i	due time
E_i	earliness
F_i	slack time
K_i	clique
P_i	processing time
R_i	ready time
S_i	start time
T_i	tardiness
V_i	volume

Machine Related Variables

Q_j processing rate

N_j resource requirement

Resource Related Variables

Y_r resource capacity

Assignment Related Variables

$L_{i,j}$ setup time

Other Variables

$W_{j,t}$ machine window flag

$W_{r,t}$ resource window flag

$W_{i,j,t}$ simultaneous machine/resource window flag

3.5 Objectives

As discussed in Chapter 1, the scheduling objectives are difficult to quantify due to their qualitative nature. Minimizing the total weighted tardiness is considered to be the best proxy for these objectives, because of its customer orientated focus of meeting deadlines. The objective function is therefore:

$$\text{minimize} \quad Z = \sum_{i=1}^{n} A_i * T_i \qquad (1)$$

subject to the following constraints.

3.6 Constraints

Job Related Constraints

Ready Time

Product from each batch cannot be transferred until after it has been blended and tested. Hence no job can start processing until ready:

$$S_i \geq R_i \qquad (2)$$

Due Time

Each job completed after its due time incurs a penalty resulting from demurrage and customer dissatisfaction. No credit (or penalty) is incurred for jobs completed before their due time:

$$T_i = \max (C_i - D_i, 0) \qquad (3)$$

Quantity

The quantity transferred in each operation is a function of the processing time and the processing rate of the machine used. Thus the total quantity transferred is the sum of the quantity transferred in all operations, and is not more than the total quantity available:

$$\sum_{k=1}^{p} P_k * Q_k \leq V_i \qquad (4)$$

$$Q_k = f(i,j) \qquad (4a)$$

where k relates to operations for each job

Machine Related Constraint

Processing Limit

Each product can only pump one product at any one time. Hence a new operation cannot start until after the previous operation has finished:

$$S_{k+1} \geq C_k \quad \text{for all k, except k=p} \qquad (5)$$

Windows

Pipelines may be unavailable due to maintenance requirements, etc. Thus jobs must be confined to periods when the machine is available. Therefore, for all windows :

$$S_k \geq t(\text{window open}) \quad \text{for all k} \qquad (6)$$

$$C_k \leq t(\text{window close}) \quad \text{for all } k \qquad (7)$$

Resource Related Constraints

In the following section k relates to the operations for each resource.

Suction Pipelines Some suction pipelines can accommodate the simultaneous processing of multiple jobs, providing that the jobs are of the same product type. Thus for each resource:

Compatibility $S_{k+1} \geq C_k$ if $K_{k+1} <> K_k$ (8)

Capacity $S_{k+1} \geq C_k$ if $N_j > Y_r$ (9)

and $j = f(k)$, $r=f(k)$ (9a)

Windows not modelled (10)

Offtaker Pipelines as for suction pipelines

Compatibility $S_{k+1} \geq C_k$ if $K_{k+1} <> K_k$ (8)

Capacity $S_{k+1} \geq C_k$ if $N_j > Y_r$ (11)

and $j = f(k)$, $r=f(k)$ (11a)

Windows not modelled (12)

Offtaker Terminals

Windows

Terminals may be unavailable due to maintenance requirements or work practices, or may only be available at cost due to overtime penalties. Thus jobs must be confined to periods when the resource is available. Therefore, for all windows :

$$S_k \geq t(\text{window open}) \tag{13}$$

$$C_k \leq t(\text{window close}) \tag{14}$$

Assignment Related Constraints

In the following section k relates to the operations for each machine.

Setup Time

Pipelines must be purged clean between transfers of dissimilar product types to avoid contamination, resulting in sequence dependent setup times:

$$S_{k+1} \geq C_k + L_k \tag{15}$$

Preemption

Preemption of jobs in process is allowed if:

1. the "old" job is not close to finishing

$$C_k \geq t + x1 \tag{16}$$

2. the "old" operation has not just started

$$S_k \leq t + x2 \tag{17}$$

3. the "old" priority is less than the "new"

$$B_k < B_{k+1} \tag{18}$$

where x1 and x2 are user defined constants.

Chapter 4 ALGORITHM DESCRIPTION

4.1 Introduction

The preceding chapter discussed the mathematical relationships which describe the main characteristics of the pipeline scheduling system. However, these relationships do not provide an indication of how the problem should be solved. They simply describe the system constraints and how the solutions can be measured and compared against each other (through the objective function). The purpose of this chapter is to describe the algorithm employed to solve this problem and how this algorithm is implemented in an overall computer application.

The computing resources available at the start of this study in the mid 1980's had a significant bearing on the solution strategy adopted. Principally, the facilities available at the time and the level of ongoing support expected indicated that a simple approach was required. In addition, a significant portion of the scheduling task is performed out-of-hours and off-site, suggesting the need for a method which could operate effectively on a personal computer (PC) during these periods.

Analysis of the scheduling problems encountered indicated that the main benefit resulting from improving the scheduling method would be a reduction in the time taken for schedule preparation. Economic benefits were difficult to quantify and perceived to be relatively small within the overall refinery context. Thus the amount of capital available to finance the project was considered to be limited, constraining the hardware and software options available.

Considering these factors and the reported experiences discussed in Chapter 2, a heuristic approach, built into a tree-search, was judged to be particularly suitable for this pipeline scheduling task. Such an approach could be relatively easily implemented using existing computing facilities, would meet the operating time and availability demands of the user, and would meet the budgetary requirements of the company. Hence this approach was selected for development and experimental investigation to develop a better understanding of the method, capability and performance.

The algorithm was coded in FORTRAN 77 because the language is well supported within the organization and is available for mainframe and PC applications. The attractions offered by more advanced codes were considered insufficient to offset the potential problems with ongoing maintenance and development, based upon historical experience within the company.

4.2 Organization

The scheduling application is arranged into seven main files, as shown in Figure 4.1. The first file is the program code. The input data supplied by the user is allocated amongst four files, based upon the type of information supplied. The remaining two files are the report files which contain the program output.

The file unit numbers to which these files are allocated are defined by data statements in the program code and can be changed to suit the computer system on which the program operates (Metcalf [53]). The file

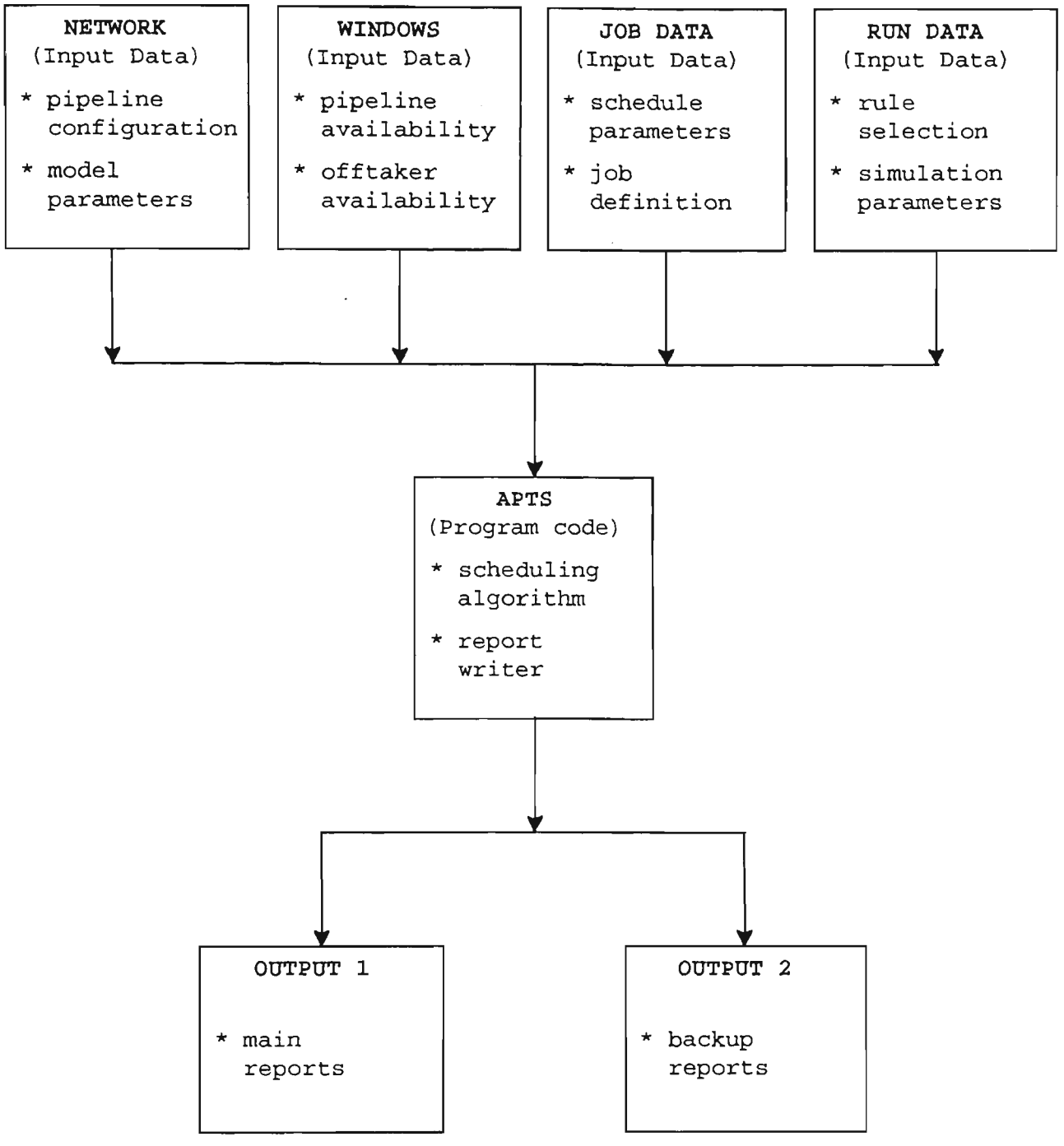


Figure 4.1 Data Organization

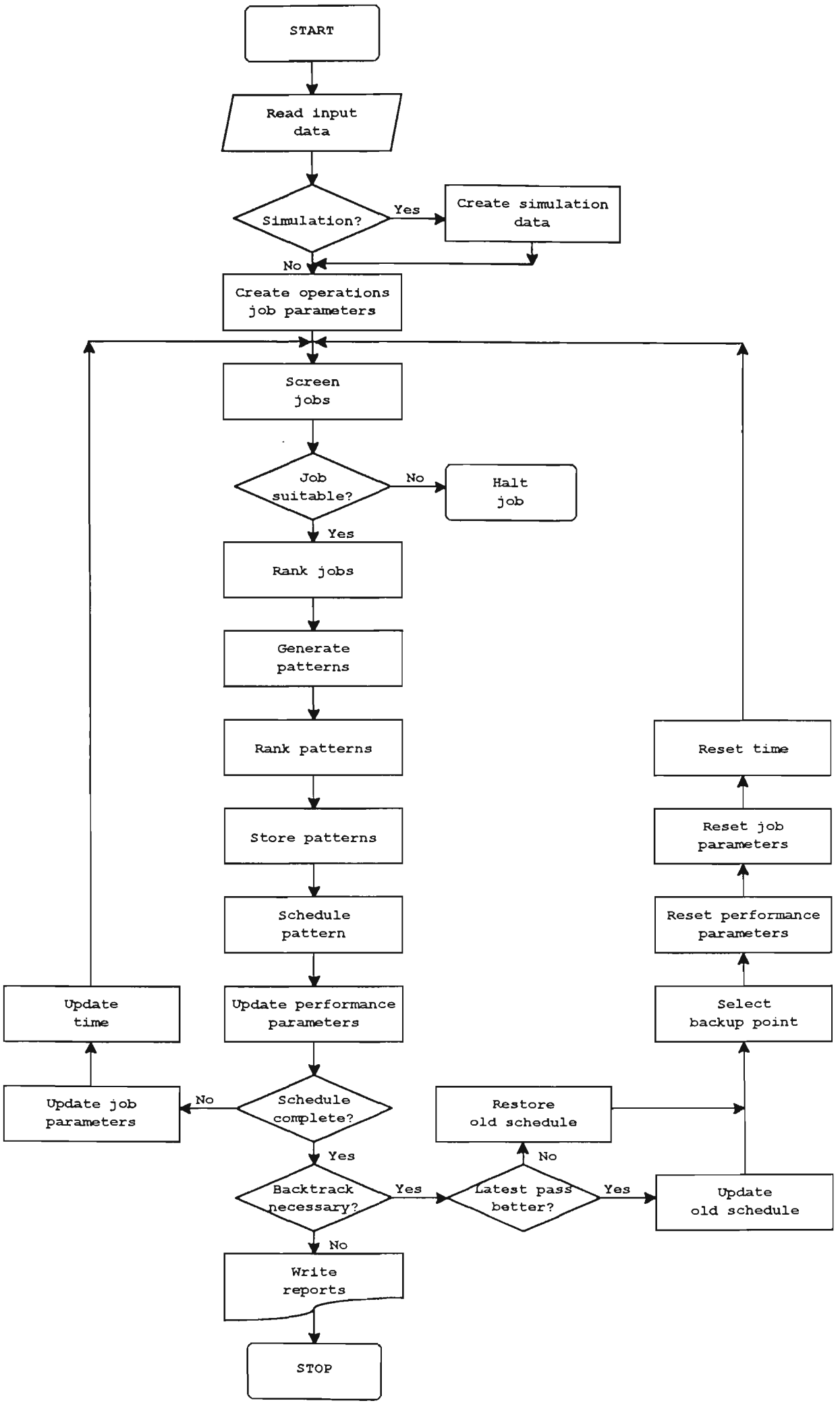


Figure 4.2 Program Schematic

names assigned to these unit numbers are defined by the user. This is explained in Appendix F.

4.2.1 Program

The program is organized into a main program and a number of supporting subroutines. Extensive use of "named common" is used to minimize data storage requirements and to efficiently transfer data between the main program and the subroutines (Metcalf [53], Fox [28]).

BLOCK1

This is the block data program which assigns default values to the named common. Arrays and variables are grouped into named common based upon the type of data to which they relate. For example, all of the job data are grouped into JOBDAT and all of the pipeline data into PLNDAT.

MAIN

This is the main program which contains the scheduling algorithms. This program reads the input data and processes the information, calling the subroutines when required.

CONVRT

This subroutine converts the ready time and due time for each job supplied in the user input file from the user input format to the internal format used by the program. It also performs the reverse conversion from the internal program format to the user format prior to the program output reports being written.

SORT

This subroutine performs a tree sort of keys and tags into ascending key order. This is called at a number of stages within the scheduling algorithm; for example, to sort queues of jobs waiting to be selected and to sort patterns after generation prior to scheduling.

REPORT

This subroutine contains all of the reports. Each report is identified by a unique code which is passed from the main program when the subroutine is called.

Other

The other main subprograms include subroutines to provide graphical output and their supporting functions, and functions used during simulation to generate pseudo-random numbers and associated frequency distributions. These are discussed in Chapter 5.

4.2.2 Input Data

The input data is organized into four files. An example of each of these files is located in Appendix F.

NETWORK

This file defines the pipeline network configuration, such as refinery product tankage, suction pipeline, main pipeline and offtaker pipeline linkages; pipeline pumping rates and lineclear (setup) times. This file also defines the major model parameters such as number of pipelines, number of offtakers and number of products. Once defined, the

information contained in this file only changes when the pipeline configuration or capacities alter.

WINDOWS

This file defines the machine and resource time windows (the main pipeline and offtaker availability windows). These windows are defined for one calendar week, and, for offtaker availability, distinguish between normal and overtime work periods. This information only changes when the time windows alter.

JOBDATA

This file defines the schedule parameters, such as schedule horizon and time base, and provides all the job input data, such as ready and due times, product type and volume, and priority. This information changes with each schedule.

RUNDATA

This file defines the parameters which control the algorithm's performance. This includes the specific job priority rule or rules and the relative weighting of job parameters to be used during scheduling. This enables the user to tune the algorithm to improve the overall scheduling performance.

This file also allows the user to define the parameters used during simulation experiments (as discussed in Chapter 5). During simulation experiments all of the job input data is created from information contained in the RUNDATA file. Hence in these cases the JOBDATA file is ignored.

4.2.3 Output Data

Two output files are defined:

OUTPUT1

The primary output file contains the summary reports of interest to all users. The reports may be selected by the user, as explained in Appendix F.

OUTPUT2

The secondary output file contains the supplementary reports used to track the internal computations of the program. These are only occasionally required and, due to the amount of information produced, are segregated from the primary output reports. As above, individual reports may be selected by the user. These are explained in Appendix F.

4.3 Algorithm Description

This section describes in detail the methodology and tasks of the scheduling algorithm, which are shown in Figure 4.2. The functions and frequency distributions used to generate input data for the simulation experiments are discussed in Chapter 5.

4.3.1 Read and Prepare Input Data

The input data contained in the four input files described above is read, commencing with the NETWORK file which sets many of the system parameters. Extensive data checking is performed to detect out of range

values and warning messages are printed. Depending on the severity of the error program interruption may occur if default values cannot be assumed.

Numbers are assigned to offtaker, product and pipeline names for internal program storage and manipulation. Ready and due time data are converted to integer format for internal calculations. A number of arrays are created to minimize repetitive calculation during execution, most notably arrays to identify simultaneous offtaker and pipeline availability.

For each job defined by the user, or during simulation, a set of operations is created for each processing option available (viz, for each pipeline on which the job can be processed). Each of these operations can be preempted during processing if necessary to complete the job. The reason for creating these operations is that the processing characteristics of every job are a function of the pipeline on which the operation is processed. For example, suction and offtaker pipeline requirements, setup times and processing rate all depend primarily upon the main pipeline on which the job is processed. It is easier to manipulate these operations than constantly refer to the original jobs during program execution.

Four special jobs and their associated operations are defined to represent non-standard uses of the pipelines:

1. Setup: the pipeline is unavailable for use while the system is made ready for a new operation.

2. Contention: the pipeline is inactive because although one or more operations could potentially use the pipeline they are precluded due to contention with higher priority operations on another pipeline (i.e. because of incompatible offtaker and/or suction pipeline requirements).

3. No Operation Available: the pipeline is inactive because no operations are available to be processed at that time.

4. Idle: the pipeline is deliberately held inactive even though operations exist which could be processed at that time.

4.3.2 Screen Jobs

At each stage of the schedule generation phase five classes of jobs (and operations) exist:

1. Operations which are currently being processed and which must continue to be processed (i.e. cannot be preempted).

2. Operations which are currently being processed and which may be preempted if desired.

3. Operations which are eligible to be processed but which are not currently being processed.

4. Operations which have not been completed but which are currently ineligible to be processed.

5. Jobs which have been completed.

The purpose of the screening step is to exclude those operations belonging to classes 4 and 5 above from further examination during the next schedule generation stage. Each operation is tested against the following criteria and excluded if any of the conditions are satisfied:

Job Based Criteria

1. The job has already been completed.

2. The job is not ready to be processed.

3. The operation is a sibling of another operation which is currently being processed and is almost finished.

4. The operation is the same product type to the same offtaker as another operation already being processed.

   There is nothing to be gained in considering such a switch due to capacity losses associated with setup times.

Machine (Pipeline) Based Criteria

1. The pipeline is currently being used to process a non-preemptable operation.

   If an operation currently being processed is so close to finishing that is would be inefficient to replace it with another then the algorithm excludes all other operations that require the same pipeline. The algorithm also excludes all other operations which

require the same suction or offtaker pipelines if they are of different product type or if the suction or offtaker pipelines cannot handle multiple simultaneous operations.

2. The pipeline is currently unavailable.

3. The pipeline is currently available but won't be available for long enough to justify scheduling the operation.

The algorithm assumes that capacity losses due to setup times impose a minimum time limit for efficient operation processing. The value of this limit may be controlled by the user. This period, plus any setup time, is compared with the operation processing time remaining and the time available within the scheduling horizon, and the minimum of these three is selected. The algorithm looks ahead to ensure that the pipeline is available for all of the minimum period identified. If not the operation is excluded.

Resource (Offtaker) Based Criteria

1. The offtaker is not available, unless the operation was previously being processed (i.e. on-line) and the offtaker is willing to continue.

The algorithm assumes that if the operation was being processed in the previous time stage then the offtaker will wish to continue receiving the operation if it can be finished within a reasonable period. This reasonable period is assumed to correspond to an acceptable overtime limit, defined by the user. If the total job cannot be finished within this period the algorithm assumes that

the offtaker would stop the current operation and recommence during normal business hours when labour costs are cheaper and labour is readily available.

2. The offtaker is currently available but won't be available for long enough to justify scheduling the operation.

The algorithm looks ahead to ensure that the offtaker is available for all of the minimum period identified above. If so the operation is included. If not the algorithm determines whether the job can be finished if the offtaker availability window is extended by a reasonable overtime period (providing that the window does not include any time that the pipeline is unavailable). If so the algorithm assumes that the offtaker will work the overtime and include the operation. If not the operation is excluded.

Resource (Suction Pipeline) Based Criterion

1. The suction pipeline is currently being used by another operation and, either the suction pipeline cannot process more than one operation simultaneously, or the operation in question is of a different product type than the operation already using the suction pipeline.

Resource (Offtaker Pipeline) Based Criterion

1. The offtaker pipeline is currently being used by another operation and, either the offtaker pipeline cannot process more than one operation simultaneously, or the operation in question is of a different product type than the operation already using the offtaker pipeline.

The anticipated completion time of the job, the finish time of the current operation, and the processing time of the current operation are calculated at this stage for later use.

4.3.3 Rank and Queue Jobs

Each job passing through the screening phase is ranked to control the order in which jobs are selected for scheduling. Ranked jobs are queued in a common queue and then sorted in descending order of priority. The different ranking parameters investigated in this study to determine optimal system performance, both in isolation and in combination with each other, are as follows.

1. Work Based

Shortest Job Processing Time (SPT)

The processing time remaining is the job volume remaining divided by the processing rate of the machine upon which the job will be processed. If the job is already being processed it is the volume of the job remaining divided by the total processing rate of the machine or machines on which the job is currently being processed. The highest priority is given to the job with the shortest processing time.

Longest Job Processing Time (LPT)

The highest priority is given to the job with the longest processing time (i.e. the job with the most work remaining to be completed).

Longest Operation Processing Time (LOP)

The highest priority is given to the job with the longest next operation processing time. The duration of the next operation is limited by either offtaker or pipeline availability or the amount of work remaining. In this later case the operation processing time is equal to the job processing time.

2. Due Time Based

Earliest Due Date (EDD)

The highest priority is given to the job with the earliest due date.

3. Tardiness Based

Slack Time (SLACK)

In this scheme each operation is ranked using a measure based upon its slack time. This slack time is the difference between the available processing time (between the current time and the due time) and the remaining job processing time. Small positive values indicate an urgency to commence processing the job. Negative values indicate that the job cannot be completed before the due time, i.e. the job will be tardy.

Float Time (FLOAT)

This parameter is analogous the the slack time, but in this application is measured against the dead line, as opposed to the due time.

Weighted Tardiness (WTAR)

The highest priority is given to the operation with the highest weighted tardiness. The tardiness is based upon an estimate of the earliest possible completion time of the job, recognizing any relevant offtaker or machine downtime.

Weighted Tardiness Delta (WDEL)

In case the operation fails to start during the current time period the next possible starting time, completion time and tardiness are estimated. The highest priority is given to the operation with the highest weighted tardiness increment that will result if the operation does not start immediately.

Apparent Tardiness Cost (ATC)

Operations are ranked in order of the Apparent Tardiness Cost index (Vepsalainen and Morton [70]), discussed in Chapter 2. The index may be tuned by the user by adjusting the value of a look-ahead parameter which scales the slack time according to the expected number of competing jobs.

Cost Over Time (COT)

Operations are ranked in order of the cost over time index (Vepsalainen and Morton [70]), discussed in Chapter 2. The index may be tuned by the user by adjusting the value of a parameter which relates the processing time to the expected waiting time.

4. Other

First In First Out (FIFO)

Priorities are assigned based upon the order in which the job becomes available (i.e. the job ready time).

Random (RANDOM)

Priorities are assigned randomly. This option is used principally as a base or control to enable comparison of the other ranking methods during experimental investigation.

Load Levelling (LOAD)

The machine load is defined as the number of operations competing for the machine at any particular time. Hence this provides a measure of the probability that one particular operation will be successful in seizing the machine. It makes sense to hold an operation which may be processed when the machine is lightly loaded, i.e. during periods when other operations are unable to be processed, until that time arises. This avoids the problem of processing such operations during heavily loaded periods and then leaving the machine idle due to lack of jobs later on.

For each machine the number of operations that could possibly require processing is estimated for each time period. Operations are excluded if their parent job has already finished, of if their offtaker or machine is unavailable.

This information is used to estimate the average machine load during the period required to process each job to completion, for each possible

starting time between the current time and the end time. The minimum of these average machine loads is then found.

The highest priority is given to the operation with the highest minimum average machine load. The greater the value of this minimum the less flexibility the operation has to be processed during lightly loaded periods.

This value may also be thought of as measuring the "price" that a particular operation is willing to "pay" in order to use the machine. An operation which has a lower minimum average load will be prepared to wait for a more lightly loaded period, when it can use the machine for a lower fee.

Scaling

Each rule is activated by setting a selection flag to 1 and deactivated by setting the flag to 0. Multiple rules may be selected and their relative importance specified by the magnitude of the selection flag. Hence, setting the Shortest Processing Time flag to 10 and the Earliest Due Date flag to 1, would result in jobs being ranked based on both processing time and due date, with processing time being 10 times more important than due date.

In ranking jobs the value of each of the indices corresponding to the active selection rules are computed. Each index is then scaled from 0 to 1 before the weighting factors are applied and the values aggregated. For example, all of the shortest processing time indices are scaled from 0 (longest, lowest priority) to 1 (shortest, highest priority), and

likewise for the earliest due date indices, from 0 (most distant, lowest priority) to 1 (closest, highest priority). These scaled values are then multiplied by their relative weights and added together to define the parameter used to rank the jobs during the sorting phase.

4.3.4 Generate Job Patterns

In this phase a number of operation "patterns" (typically limited to ten) are generated. Each pattern is a unique allocation of operations to machines, and may include any number of idle machines. Hence, if we have three operations A, B and C, and three machines 1, 2 and 3, possible patterns would include:

Table 4.1 Pattern Generation Example

Pattern	Machine		
	1	2	3
1	A	B	C
2	A	C	B
3	A	B	-
4	A	C	-
5	A	-	B
6	A	-	C

and so on.

Patterns are generated by selecting each operation in turn from the selection queue (in which the operations have been sorted based upon one or more of the priority rules discussed above). Each operation is tested against the following criteria and excluded if any of the following conditions occurs:

1. The operation uses the same main pipeline as a higher priority operation already allocated in this particular pattern.

2. The operation uses the same suction pipeline as a higher priority operation and, either the operation product is different than the higher priority operation product, or the suction pipeline cannot handle multiple simultaneous operations.

3. The operation uses the same offtaker pipeline as a higher priority operation and, either the operation product is different than the higher priority operation product, or the offtaker pipeline cannot handle multiple simultaneous operations.

4. The operation is a sibling of a higher priority operation which is nearly completed and the operation was not on-line in the previous scheduling stage. Hence there is little point in allocating the new operation because the parent job will be completed shortly (at the end of the operation of the higher priority sibling).

After generation each pattern is checked to ensure it is unique for the current scheduling stage. Non-unique patterns are rejected.

The selection queue is reordered to ensure a different starting point for the generation of each pattern. This is achieved by selecting a new starting operation from a controlled point within the queue, moving all operations closer to the front of the queue backwards one place, and placing the selected starting operation at the front of the queue. For example, the first four patterns would be drawn from the following ordered queues:

Table 4.2 Queue Reordering Example

Pattern	Queue Order							
	1	2	3	4	.	.	.	n
1	A	B	C	D	.	.	.	.
2	B	A	C	D	.	.	.	.
3	C	B	A	D	.	.	.	.
4	D	C	B	A	.	.	.	.

and so on.

This process is repeated until a specified number (usually 5 or 10) of unique patterns have been generated or the queue exhausted. Although this scheme does not generate all possible patterns a sufficient number are created to provide viable options for the algorithm to choose from during backtracking.

4.3.5 Sort Job Patterns

Usually a number of patterns are created from the previous phase and tuning factors are used to weight the relative importance of the main pattern characteristics to enable a choice to be made. Job patterns are sorted based upon a performance measure calculated from a weighted average of volume processed, average processing rate, pattern duration and pattern creation order.

Volume

The total volume of material processable by the pattern is a measure of the amount of work the pattern can achieve. It is not a measure of the rate at which the work is achieved and hence is not necessarily a measure of utilization (a long running pattern using, say, 3 machines,

may be able to process more work than a short running pattern using 4 machines, but obviously does less work in the same amount of time).

Utilization

Ranking patterns based upon the average processing rate achieved by the pattern encourages maximum machine utilization. The average processing rate is the volume transferred by the pattern divided by the pattern duration. This includes any debits associated will loss of capacity during machine setups and unused machines. Hence patterns which have all machines utilized are selected in preference to patterns in which one or more of the machines are idle.

Length

Patterns with long sustainable duration imply less setups and hence less capacity loss in the longer term. However, these patterns implicitly lower the priority of patterns in which multiple offspring operations of one job (or more) are processed simultaneously. This is because the duration of these patterns is shorter because the job is completed earlier than when only one sibling is scheduled.

Order

The order in which the patterns are created reflects the order in which the operations are ranked, with the first pattern containing the most important operations. This occurs because prior to pattern generation all candidate operations are ranked based upon the outcome of the priority rule or rules invoked.

Value

Not all the operations ranked by the previous step can necessarily be scheduled, due to machine and resource constraints. Hence lower order patterns sometimes result in more operations being scheduled and more efficient machine utilization than higher order patterns. The value in this case is the priority assigned by the job selection rule which ranks the jobs prior to pattern generation. Priority is given to patterns with higher average values.

4.3.6 Store Data For Backtracking

All of the patterns obtained above are stored to allow retrieval during any later backtracking. In addition, all of the information which describes the status of the overall system is stored so that the current stage can be re-established when required. This includes:

Job Related Data

Total volume remaining to be processed

Minimum volume remaining to be processed

Setup time remaining

Current job status

Machine Related Data

Current pipeline status

Allocation Related Data

Pattern allocation

Pattern selection order

Current pattern selection

Pattern duration and total volume transferred

Some job related data for operations currently being processed are stored with the machine related data. This minimizes the amount of storage required, because the number of machines is much less than the number of jobs (for example, 5 machines versus 300 operations).

In the backtracking phase the algorithm searches for times in which machine utilization is low due to potentially active jobs being precluded as a result of contention with higher priority jobs. Often the patterns generated at each stage are ranked based upon machine utilization. Hence, if the pattern first selected has a low machine utilization, it is likely that all remaining patterns at that stage will also have a low utilization. Therefore, the algorithm "backs-up" to the preceding stage (the one before this problem stage) to select a new pattern. In this manner the algorithm attempts to find a starting point which will improve later machine utilization.

Thus, rather than store data for every scheduling stage, only those stages immediately preceding a stage in which an operation is precluded due to contention or when a machine is deliberately idled are stored. This is achieved by storing the data for every stage as it arises, and overwriting this with the next stage if the next stage does not have a precluded operation or idled machine.

4.3.7 Schedule Selected Pattern

If any machine is idle, either due to no suitable operations being available, or potential operations being precluded due to contention, or

deliberately idled, then the selected pattern is scheduled for only one time period. On the other hand, if all machines are utilized then the pattern is scheduled for the minimum operation processing time calculated in the pattern generation phase. This minimum time is subject to an upper limit, definable by the user, to allow the algorithm to re-evaluate the appropriate pattern at suitable intervals. Usually, however, this minimum time limit is set at the resolution limit (one hour).

During this scheduling phase the Gantt chart is updated to reflect setup and processing intervals. For each scheduled job the total volume remaining to be processed and the minimum volume remaining to be processed for the current operation are calculated. The status of the operations currently being processed is reviewed and non-preemptable operations and completed jobs identified. Performance parameters, such as the cumulative volume transferred, cumulative number of jobs precluded due to contention, and cumulative weighted tardiness are calculated.

4.3.8 Check if Finished

The cumulative performance of the current schedule is checked against the performance of the previous best completed schedule. If performance is satisfactory and the current schedule is unfinished (i.e. uncompleted jobs remain to be processed) the algorithm updates the clock and returns to the job screening step. If the current phase is completed, the algorithm proceeds to the backtracking step.

If the current solution is worse than the previous best solution the portion of the solution updated in the current solution phase is reset to the previous best solution and the algorithm proceeds to the backtracking step.

Performance Measures (Objective Function)

In the experiments discussed in Chapter 5 the objective function was based on total weighted tardiness. However, the algorithm allows the relative performance between solutions is assessed based upon the sum of a number of weighted parameters if required, such as:

1. Total weighted tardiness
2. Total setup count
3. Total offtaker overtime cost
4. Makespan
5. Total volume

4.3.9 Backtrack if Necessary

The algorithm tests to see if backtracking can or should continue. Backtracking is terminated if:

1. All backtracking options have been exhausted,

2. There has been no performance improvement in a specified number of the past consecutive passes,

3. The number of backtracking attempts has exceeded a specified maximum.

If backtracking is to continue the solution from the most recent backtracking pass is compared to the previous best solution and this best solution is updated if necessary.

The backtracking strategy is to backup to the most recent backup point identified in the storage step and select the next untried pattern available at that point. If all patterns have been tried at this point and no improvement in the solution has been observed then the algorithm backs up to the previous untried backup point. Backup continues in this manner until either a suitable backup point is found or the start of the schedule is reached, at which point the program is terminated. This backtracking process is demonstrated in Figure 4.3.

4.4 Tuning Factors

Tuning factors are used to modify the performance of the algorithm by controlling selection and prioritization of jobs. These factors are described in detail in Appendix F.

```
HYPOTHETICAL REFINERY                JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

CASE:   1   PASS:   1   VALUE:   1520      *** B A C K T R A C K    S U M M A R Y ***


CURRENT BACKTRACK PASS                                  1
OBJECTIVE VALUE, CURRENT PASS                        1520
OBJECTIVE VALUE, PREVIOUS BEST                     999999

LATEST PASS BETTER THAN PREVIOUS BEST: UPDATING SCHEDULE

OLD BACKTRACK POINT                                     1
OLD BACKTRACK TIME                                      0
OLD BACKTRACK PATTERN                                   0

NEW BACKTRACK POINTS IDENTIFIED                         6
TOTAL BACKTRACK POINTS AVAILABLE                        7

BACKTRACK POINT        1     2     3     4     5     6     7
BACKTRACK TIME         0    31    40    43    51    57    69
TARDINESS              0     0   120   240   640  1120  1360
PATTERN COUNT          0     8     3     6     4     5     2
LAST PATTERN USED      0     1     1     1     1     1     1

NEW BACKTRACK POINT                                     7
NEW BACKTRACK TIME                                     69
NEW BACKTRACK PATTERN                                   2
 PIPELINE            PL1      PL2      PL3      PL4
 JOB NUMBER           14       58       58       57

CASE:   1   PASS:   2   TIME:   69 H    JOBS CHOSEN:   14   58   58   57    0
```

Figure 4.3 Backtracking Strategy

Chapter 5 EXPERIMENTAL INVESTIGATION

5.1 Introduction

Instance evaluation and simulation appear to be the most useful techniques for assessing the performance of practical scheduling algorithms, based upon the literature reviewed in Chapter 2.

Instance evaluation provides a qualitative comparison between schedules produced by different methods; for example, between the existing manual method and the proposed algorithm. Complementing this, simulation allows the statistical comparison of schedules produced under varying heuristic and decision rule conditions. Thus simulation enables conclusions to be drawn about the mean and variance of schedule performance parameters, which can assist in tuning the algorithm to give the "best" overall performance.

5.2 Instance Evaluation Experiments

The objectives of this series of experiments were to observe how schedules generated by the proposed scheduling algorithm compared with manually prepared schedules and assess the relative performance of different scheduling rules. The explanation of the observed performance differences is given in the next chapter.

Five manually prepared schedules were selected at random and information obtained from these schedules was used to provide input data to the scheduling algorithm. The differences observed are discussed in the

following section. The manually prepared Gantt charts are shown in Appendix 3.

5.2.1 Experimental Conditions

Accurate re-creation of all the conditions which applied at the time the actual schedule was developed is difficult because all the relevant information was not recorded. Typically only the Gantt chart representation was available, and information such as job priorities, ready and due times and offtaker availability had to be estimated or assumed. The following assumptions were made to provide sufficient information to enable a practical assessment of the algorithm's performance.

Schedule Horizon

This was assumed to be seven days, corresponding to the period covered by each Gantt chart. All charts started on a Tuesday and ran to the following Monday.

Jobs

Jobs were determined by analysis of the operations shown on the Gantt chart. Most jobs are composed of single operations. However, with some operational experience it is relatively easy to assess which combination of operations constitute a single job.

Product Type

The job product was obtained directly from the specific Gantt chart row. Non-standard products are recorded directly on the Gantt chart; for

example on Appendix C, Figure C.1, Pacific Islands' quality Automotive Diesel Oil is recorded as "Pac" on Saturday, 12th November.

Offtaker

This was obtained directly from the Gantt chart.

Job Volume

This was obtained directly from the chart, either from the volume recorded, or by multiplying the job time shown by the pipeline average pumping rate.

Job Priority

This information was not able to be obtained. All jobs were assumed to have the same relative priority.

Ready Time

The ready time of each job was assumed equal to the starting time of the first job judged to have been scheduled from the same product batch. For example, operational experience suggests that, in Appendix C, Figure C.1, the first two operations on each of Pipelines 1 and 4 starting on Wednesday, 9th November, came from the same batch. This is because they are of the same product type and the sum of their volumes (8.5 ML) is approximately equal to the standard product batch size for that product (premium leaded gasoline, 9.0 ML). (In addition, the two operations to Mobil, 1.7 ML Pipeline 1 and 2.0 ML Pipeline 4, are judged to be part of the same job.)

Due Time

The due time of each job was assumed equal to the completion time of the last operation judged to belong to that job. The due time of any job which was still being processed at the end of the scheduling horizon was estimated by adding the processing time to the starting time of the last operation. The processing time was estimated by dividing the stated job volume by the pipeline average pumping rate.

Deadline

This information was not able to be obtained from the Gantt chart. Hence no jobs were assumed to have a deadline.

Offtaker Availability

Analysis of the Gantt charts indicated many instances of operations being scheduled during periods in which the receiving offtaker would not normally be available. In these instances the standard availability of the offtaker was extended to cover the period the operation was being processed.

Pipeline Availability

In two of the examples one of the operations is unavailable for all or part of the schedule horizon, most probably due to equipment maintenance. The standard pipeline availability was adjusted to reflect the reduced operating window.

Batch Number and Tank Number

These are simply used for reporting purposes and do not affect the generation of the schedule. This data was provided if known.

5.2.2 Experimental Data

The data obtained from the Gantt charts is shown in Appendix C. The data defining the pipeline system and algorithm run parameters is common to each example. Separate job data are shown for each example.

5.3 Instance Evaluation Experimental Results

To enable comparison of results between different cases the results have been expressed as a ratio of the best performance. The results shown are the weighted average of the performance for each of the five cases. For example, for any parameter sought to be minimized:

Table 5.1 Performance Indicator Calculation

	Case 1	Case 2	Case 3	Case 4	Case 5	Average
Raw Data:						
k=2.0	300	400	500	600	600	480
k=3.0	200	250	300	350	300	280
k=4.0	100	120	140	180	160	140
k=5.0	200	300	400	200	300	280
Ratio of Best:						
k=2.0						3.4
k=3.0						2.0
k=4.0						1.0
k=5.0						2.0

In all of the instance evaluation experiments the following job preemption and backtracking parameters were used.

Job Preemption Tuning:

Sibling work remaining	4 hours
Job work remaining	6 hours
Minimum work remaining	4 hours

Backtracking Parameters:

Maximum passes	50
Maximum consecutive passes	10
Maximum time increment	1 hour

Performance Criteria

The primary performance criteria upon which the scheduling rules were evaluated are:

1. Total weighted tardiness: how well the rule performs overall in minimizing the tardiness of all jobs.

2. Maximum tardiness: how consistent the rule is in minimizing the maximum tardiness incurred by any job.

3. Volume delivered: how much work the rule is able to achieve in the time available.

Experimental Objectives

Four experiments were conducted:

1. Apparent Tardiness Cost Tuning
2. Cost Over Time Tuning
3. Scheduling Rule Evaluation

4. Pattern Selection Rule Evaluation

A fifth experiment to analyze the specific differences between manual and computer generated schedules was not reported due to the sensitivity of the results to the individual problem instance and scheduling rule employed.

5.3.1 Experiment 1: Apparent Tardiness Cost Tuning

Objective

As discussed in Chapters 2 and 4, the performance of the ATC rule can be tuned by adjusting the value of the look-ahead parameter, k. The objective of this experiment was to observe the relative performance of the ATC rule as a function of k.

Experimental Conditions

Objective Function Weighting: weighted tardiness

Job Priority Weighting: all jobs medium priority

Pattern Priority Weighting: pattern generation order

ATC Rule Parameter k:

Varied from 2.0 to 5.0 in increments of 0.5

Results

The results are shown in Appendix D and summarized in Figures 5.1.1 and 5.1.2 below.

1. Weighted Tardiness

The best performance is observed with k in the region of 4.0 to 4.5. Performance deteriorated as k was changed in either direction; the deterioration as k increased above 4.5 was particularly rapid. This behaviour was observed in Cases 1 to 3. In Case 4, the total tardiness at k=4.0 and k=4.5 was actually marginally worse than at each of the other values. In this case the best performance was observed at k=3.0 and k=3.5. In Case 5 there was no change in performance as k ranged from 2.0 to 5.0.

2. Maximum Tardiness

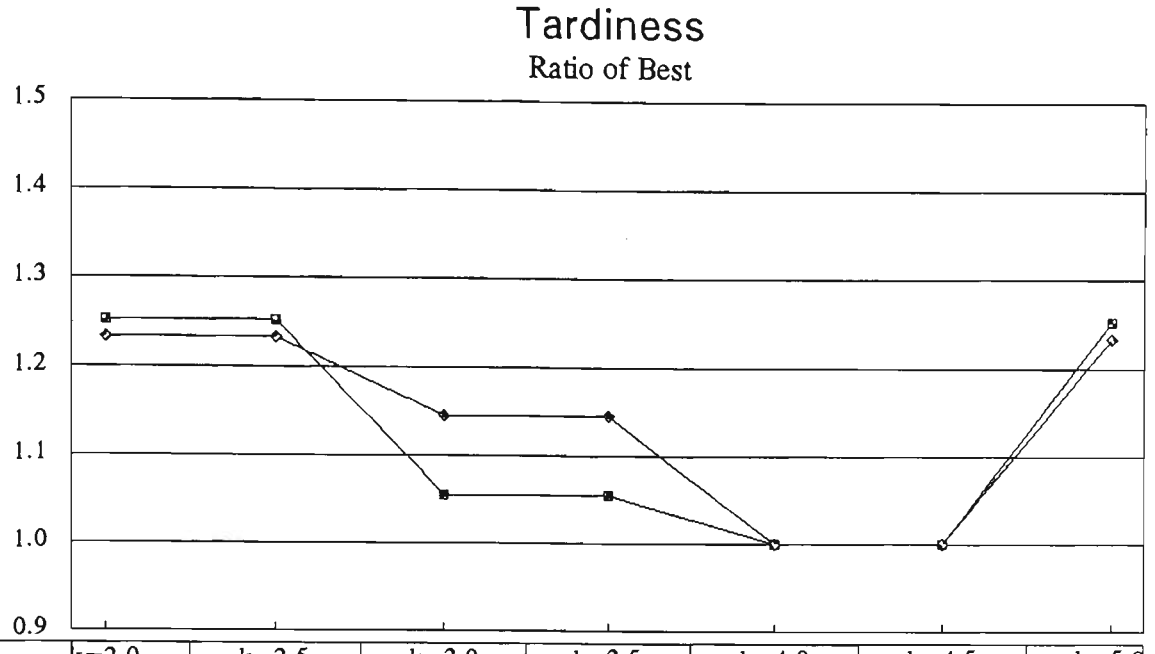
The best performance was again observed with k in the region of 4.0 to 4.5. Performance deteriorated as k was changed in either direction. The improvement in performance as k moved to between 4.0 and 4.5 was more marked than in the weighted tardiness parameter. This behaviour was observed in Cases 1 and 2. In Cases 3, 4 and 5 there was no change in maximum tardiness with changes in k.

3. Volume Delivered

The difference in volume delivered as a function of k was quite small (less than 0.5 %). The best performance was observed with k in the range of 3.0 to 4.5. Performance deteriorated as k was changed in either direction outside this range. This behaviour was observed in Case 2. In all the other cases the volume delivered did not change with changes in k.

Conclusion

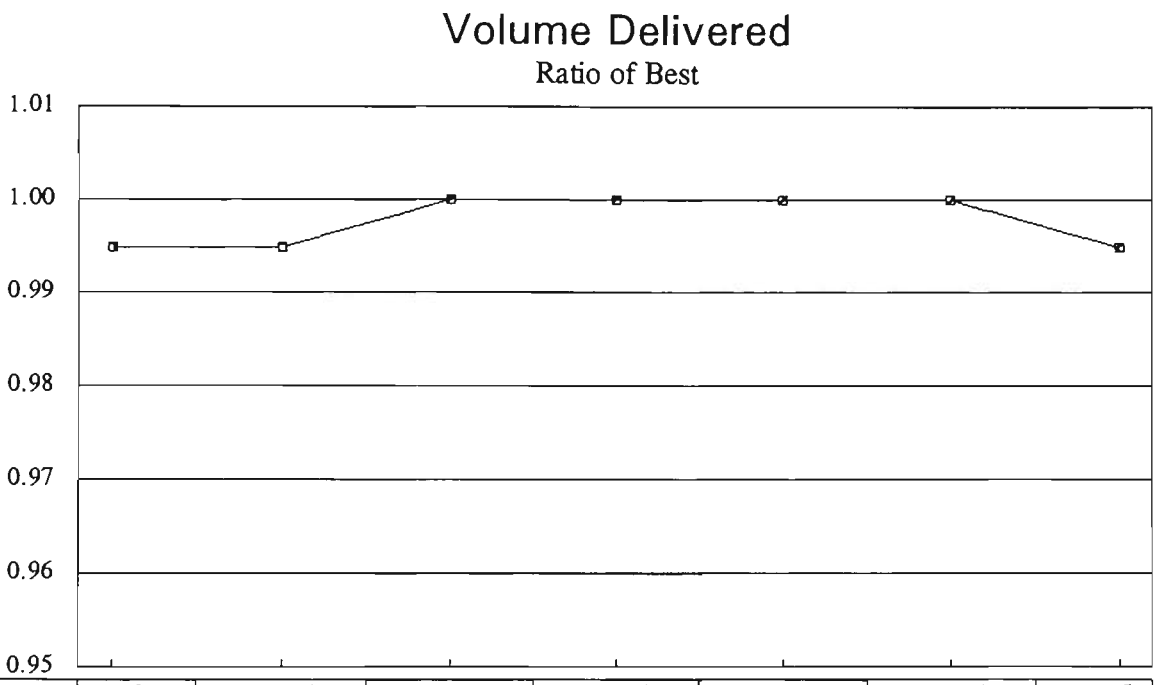
The best overall performance is achieved with k=4.0.



	k=2.0	k=2.5	k=3.0	k=3.5	k=4.0	k=4.5	k=5.0
Total	1.25	1.25	1.05	1.05	1.00	1.00	1.25
Maximum	1.23	1.23	1.14	1.14	1.00	1.00	1.23

1.0 = Best

Figure 5.1.1 Tardiness response to "look-ahead" parameter k



	k=2.0	k=2.5	k=3.0	k=3.5	k=4.0	k=4.5	k=5.0
Delivered	0.99	0.99	1.00	1.00	1.00	1.00	0.99

1.0 = Best

Figure 5.1.2 Volume Delivered response to "look-ahead" parameter k

5.3.2 Experiment 2: Cost Over Time Tuning

Objective

As discussed in Chapters 2 and 4, the performance of the COT rule can be tuned by adjusting the values of k and b. The objective of this experiment was to observe the relative performance of the COT rule as a function of k*b.

Experimental Conditions

Objective Function Weighting: weighted tardiness

Job Priority Weighting: all jobs medium priority

Pattern Priority Weighting: pattern generation order

COT Rule Parameter k*b:

Varied from 1.0 to 6.0 in increments of 1.0. Selected instances examined at increments of 0.5 where additional definition required.

Results

The results are shown in Appendix D and summarized in Figures 5.2.1 and 5.2.2 below.

1. Weighted Tardiness

The best performance is observed with k*b of 5.0. Performance deteriorated as k*b was changed in either direction, although performance improved again at k*b=1.0. At this point performance was almost as good as at 5.0. This behaviour was not consistent from case to case. In Cases 2 and 4 the performance at k*b=1.0 was worse than at any other value of k*b tested. In Cases 1 and 4 the total tardiness at k*b=1.0 was equal to or better than performance at each of the other

values. In Case 3 the performance at k*b=1.0 was about midway between the best and worst performance.

Performance at k*b=5.0 was more consistent and, although never ranking first (except when tied with other values), was usually close to the best performance. Performance at k*b=3.0 was inconsistent.

2. Maximum Tardiness

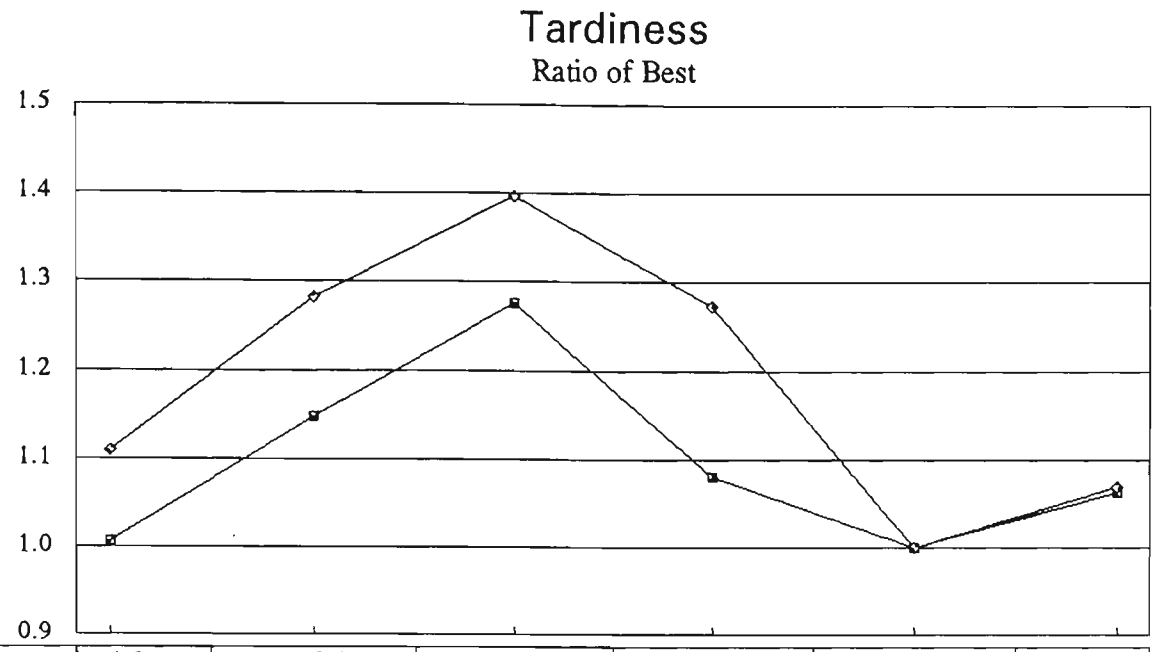
The best performance was again observed with k*b of 5.0. Performance deteriorated as k was changed in either direction, but improved again at k*b=1.0. In Cases 3 and 4 performance was relatively insensitive to changes in k*b over the range examined. Performance at k*b=5.0 was relatively consistent, although not always the best.

3. Volume Delivered

The best performance was observed with k*b in the range of 5.0 to 6.0 and also at k*b=1.0. Performance deteriorated as k approached 3.0. In Cases 1, 3 and 4 the volume delivered was relatively insensitive to changes in k*b. Performance in Cases 2 and 5 was inconsistent: in the former k*b=1.0 gave the worst performance and in the later it gave the best. As with the ATC rule, the average performance of the worst case was relatively small, in this experiment about 1 % below that of the best.

Conclusion

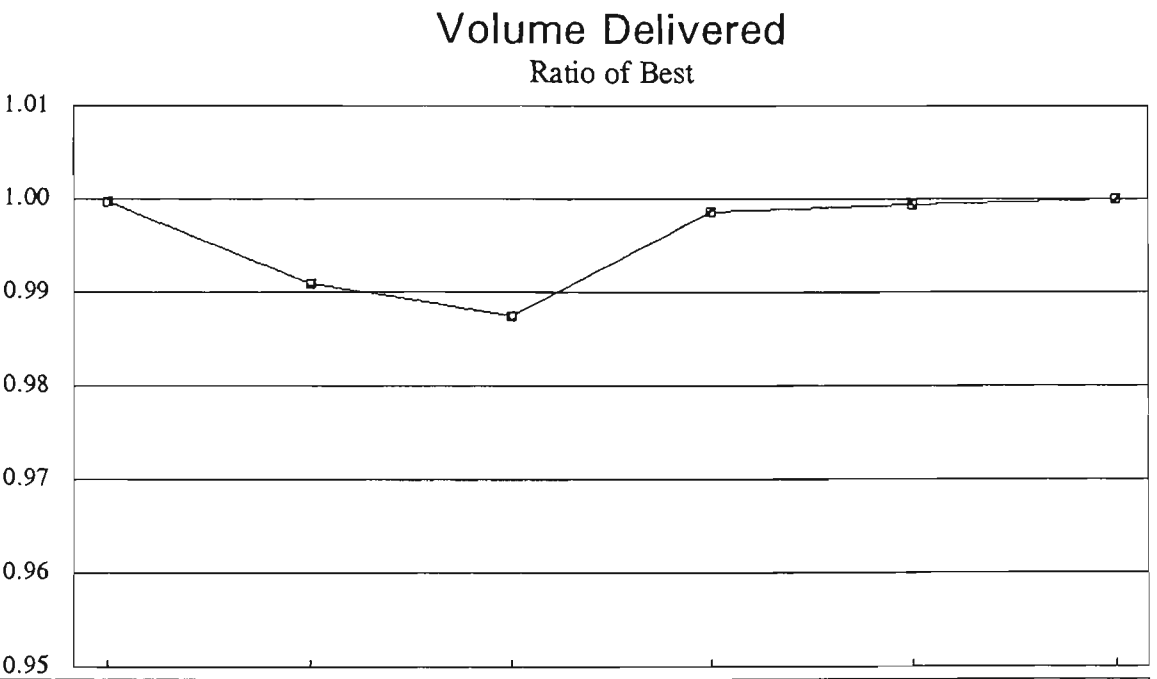
The best overall performance is achieved with k*b=5.0.



	k=1.0	k=2.0	k=3.0	k=4.0	k=5.0	k=6.0
Total	1.01	1.15	1.28	1.08	1.00	1.06
Maximum	1.11	1.28	1.40	1.27	1.00	1.07

1.0 = Best

Figure 5.2.1 Tardiness response to "waiting time" parameter k*b



	k=1.0	k=2.0	k=3.0	k=4.0	k=5.0	k=6.0
Delivered	1.00	0.99	0.99	1.00	1.00	1.00

1.0 = Best

Figure 5.2.2 Volume Delivered response to "waiting time" parameter k*b

5.3.3 Experiment 3: Job Scheduling Rule Performance

Objective

The objective of this experiment was to observe the relative performance of selected scheduling rules. The ATC and COT rules observed with k=4.0 and k*b=5.0 respectively in all cases.

Experimental Conditions

Objective Function Weighting:	weighted tardiness
Job Priority Weighting:	all jobs medium priority
Pattern Priority Weighting:	pattern generation order

Results

The results are shown in Appendix D and summarized in Figures 5.3.1 and 5.3.2 below.

1. Weighted Tardiness

The best average performance was achieved with the COT rule. Consistent performance was also achieved with this rule, giving the best result in all but Case 4, in which it came a distant second, but well ahead of the other rules. In Case 4 the EDD rule performed particularly well. The EDD rule also gave the second best average performance, followed by the SPT rule. The LPT and LOP rules consistently gave the poorest performance and, notably, were the only rules which were unable to exceed the results achieved by random selection.

2. Maximum Tardiness

The COT rule again performed best on average, although its performance was not consistently better than the other rules. On average there was

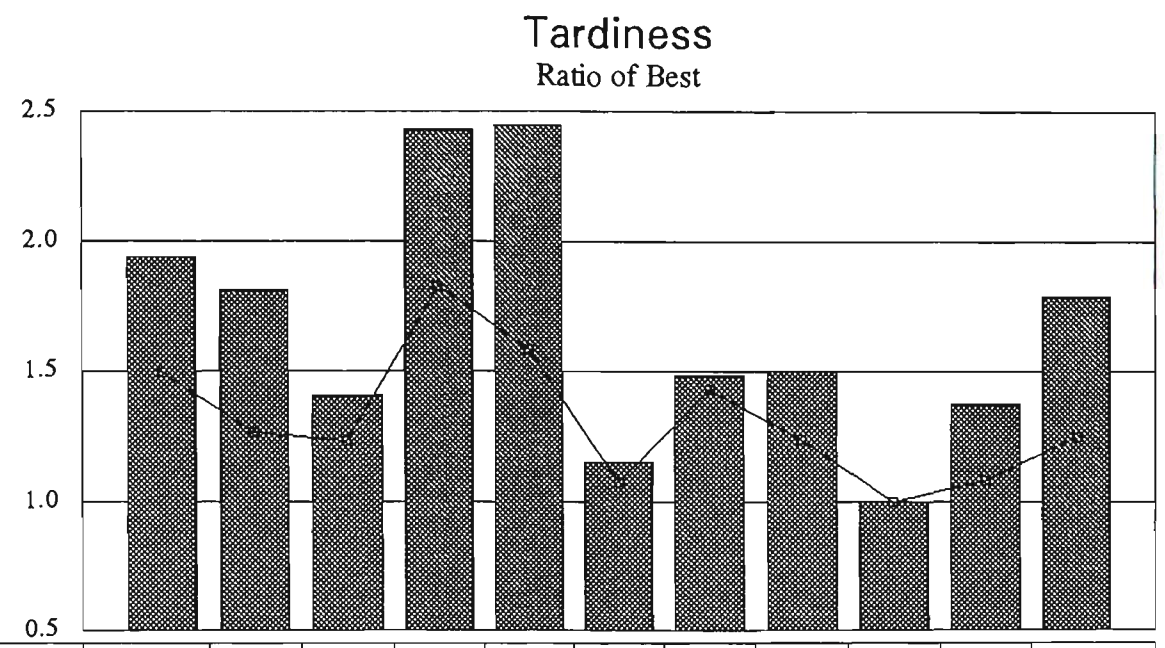
less variation in performance for all rules than was observed with total weighted tardiness. The performance of the EDD and WTAR rules was only slightly poorer than that achieved by COT. The performance of the LPT and LOP rules was again very poor and worse than that achieved by the RANDOM rule.

3. Volume Delivered

The variation observed in the volume delivered, i.e. the amount of work performed, was quite small. The best performance was achieved by the SPT, ATC and WTAR rules. This was closely followed by the SLACK, COT and LOAD rules. The performance of the RANDOM rule was equivalent to these second tier performers. The poorest performance was again achieved by the LPT and LOP rules.

Conclusion

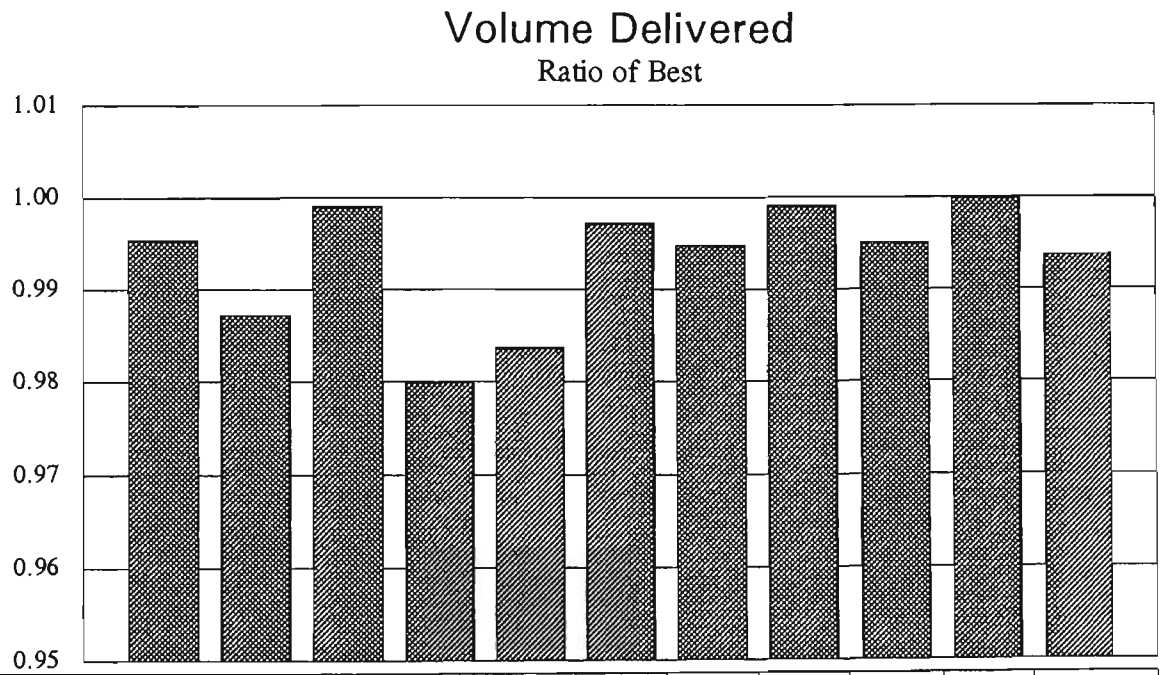
The COT rule provides the best average performance with respect to total and maximum tardiness. Its performance on volume delivered is only slightly poorer than the best achieved by the other rules.



	RAN	FCFS	SPT	LPT	LOP	EDD	SLACK	ATC	COT	WTAR	LOAD
Total	1.9	1.8	1.4	2.4	2.4	1.2	1.5	1.5	1.0	1.4	1.8
Maximum	1.5	1.3	1.2	1.8	1.6	1.1	1.4	1.2	1.0	1.1	1.3

1.0 = Best

Figure 5.3.1 Tardiness response to job scheduling rule



	RAN	FCFS	SPT	LPT	LOP	EDD	SLACK	ATC	COT	WTAR	LOAD
Delivered	1.00	0.99	1.00	0.98	0.98	1.00	0.99	1.00	1.00	1.00	0.99

1.0 = Best

Figure 5.3.2 Volume Delivered response to job scheduling rule

5.3.4 Experiment 4: Pattern Selection Rule Performance

Objective

The objective of this experiment was to observe the effect of pattern utilization on scheduling rule performance. The experiment was limited to the rules which performed best in Experiment 3.

Experimental Conditions

Objective Function Weighting: weighted tardiness

Job Priority Weighting: all jobs medium priority

Pattern Priority Weighting: pattern utilization

(pattern order was used to break ties in utilization)

The test was limited to the following rules: RANDOM, SPT, EDD, COT, WTAR, LOAD. The RANDOM rule was included as a control. The LOAD rule was included to investigate the effect of pattern utilization on the only rule based directly on pipeline utilization.

Results

The results are shown in Appendix D and summarized in Figures 5.4.1 and 5.4.2 below.

1. Weighted Tardiness

The performance of most rules was unchanged. The performance of the COT rule actually deteriorated by about 10 % on average and in all cases except Case 2, where it was unchanged. The WTAR rule was the only rule to show significant improvement on average. This improvement was exhibited on Case 1 and particularly Case 5; Case 2 and 3 were almost

unchanged, and Case 4 was actually slightly poorer. The performance of the LOAD and SPT rules was almost unchanged.

2. Maximum Tardiness

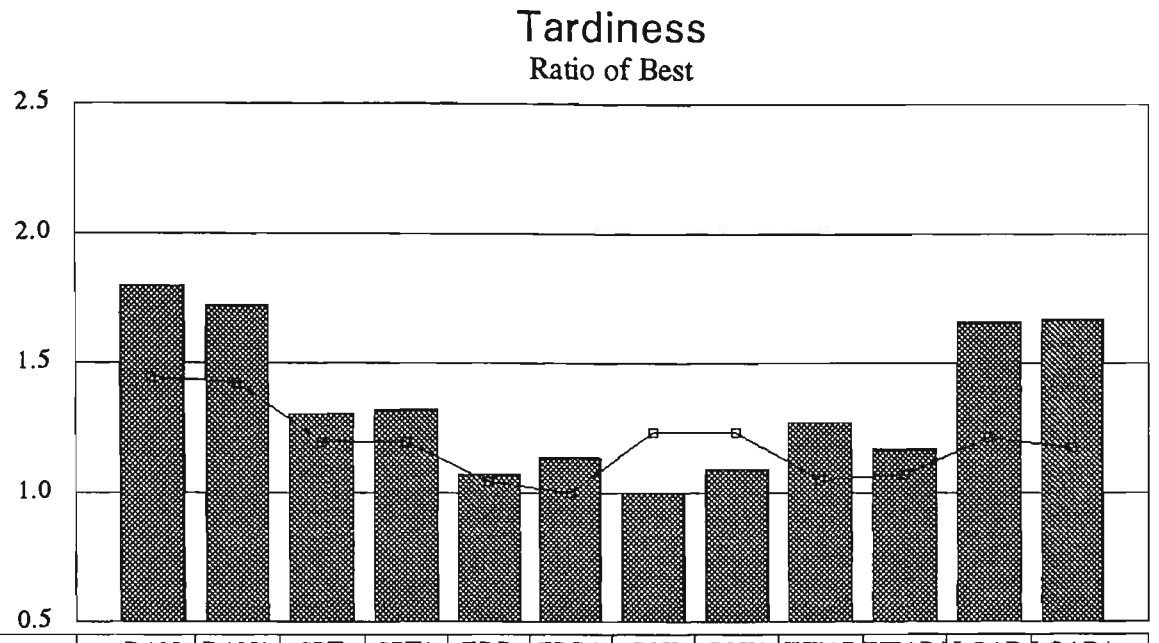
The performance of the EDD and LOAD rules improved marginally on average, and was either better or unchanged in all cases. The performance of the SPT and COT rules was unchanged and that of the WTAR rule was slightly poorer.

3. Volume Delivered

The performance of the SPT, EDD, WTAR and LOAD rules improved on average, and was either better or unchanged in all cases. Most notably, the WTAR rule in Case 5 finished all jobs, the only rule which did so, and well ahead of the next contender. The COT rule was the only rule whose performance deteriorated on average, due to Case 5. In Cases 1, 2 and 3 its performance was unchanged and in Case 4 it improved.

Conclusion

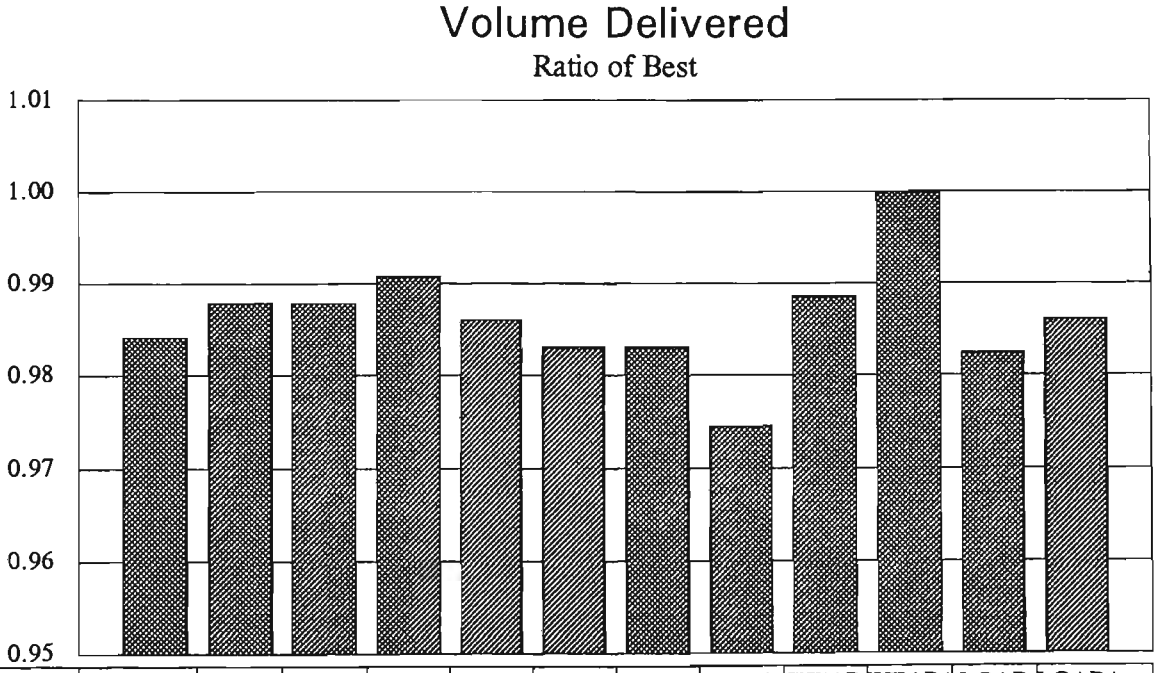
The most cases pattern utilization provides a small improvement in volume delivered. Total and maximum tardiness performance is largely insensitive to pattern utilization. The effect of pattern utilization on COT rule performance requires further evaluation.



	RAN	RAN*	SPT	SPT*	EDD	EDD*	COT	COT*	WTAR	WTAR*	LOAD	LOAD*
Total	1.80	1.73	1.30	1.32	1.07	1.14	1.00	1.09	1.27	1.17	1.66	1.67
Maximum	1.45	1.42	1.19	1.19	1.04	1.00	1.23	1.23	1.04	1.07	1.21	1.17

1.0 = Best

Figure 5.4.1 Tardiness response to pattern selection rule



	RAN	RAN*	SPT	SPT*	EDD	EDD*	COT	COT*	WTAR	WTAR*	LOAD	LOAD*
Delivered	0.98	0.99	0.99	0.99	0.99	0.98	0.98	0.97	0.99	1.00	0.98	0.99

1.0 = Best

Figure 5.4.2 Volume Delivered response to pattern selection rule

5.4 Simulation Experiment

The objective of this series of experiments was to measure the relative performance of the scheduling algorithm under varying heuristic and decision rule conditions, and under varying operating conditions, such as machine utilization and due time tightness.

The simulation experiments were conducted by sampling from distributions using a pseudo-random number generator. Forty simulation runs were conducted for each case evaluated. Mean and variance statistics were calculated for a number of performance parameters, which are discussed later.

Pseudo-Random Number Generator

Pseudo-random numbers were generated using the multiplicative congruential method, described in detail in Taha [64]. The seed and modulo function parameters described in Taha were used.

The same number of random number calls were made by the algorithm regardless of the algorithm parameters chosen for each run (except as described in Chapter 6). Hence each run was based on the same job input data for a given starting seed.

Scheduling Horizon

The scheduling horizon was set equal to the maximum of 7 days in all the runs conducted.

Starting Day Distribution

The schedule starting day of each run in each case was generally set to Tuesday, which is the principal starting day for manually prepared schedules. However, the starting day could be selected at random if desired. In this situation, the starting day was drawn from a uniform distribution of 7 days.

Pipelines

In all simulation cases four pipelines were assumed. The airport and dock pipelines which are also used for transporting product are independent of the main product pipeline network and were excluded from the analysis.

Job Number Distribution

This option was not used. However, if selected, the number of jobs used in each run is drawn from a discrete normal distribution. The job number obtained from the distribution was adjusted for the schedule horizon and number of pipelines if required (i.e. if the schedule horizon was not equal to seven days and/or the number of pipelines was not equal to four.

Machine Load

As discussed, the number of jobs was not specified explicitly. Instead jobs were selected until the total job volume reached a specified machine load level. This load level was based on the analysis of "average" and "nominal capacity" refinery production levels (described in Appendix B).

The "average" refinery production was set equal to the 1985 full year actual refinery production. The production volumes of each product were assigned to the appropriate pipelines. Hence, all of the gasoline production, such as premium leaded, unleaded and avgas, were assigned to Pipelines 1 and 4. All of the distillate, most of the jet and some of the fuel oil were assigned to Pipelines 2 and 3. Specifically, 65 % of the total jet production was assumed to be pumped on the airport pipeline, which was excluded from this analysis due to its independence of the main pipeline network. Similarly, 90 % of fuel oil production was excluded because it was pumped on a different pipeline network.

This allocation of products resulted in the pipeline utilizations matching almost precisely the pipeline utilization observed in practice over the 30 week trial period. These calculations are presented in Appendix B.

The "nominal capacity" refinery production was based on the maximum rated refinery throughput. The product mix was assumed equal to the "average" refinery product mix. Similarly 65 % of the jet and 90 % of the fuel oil were excluded from the utilization calculation.

Job Volume Distribution

The volume of each job was drawn from the empirical distribution shown in Appendix B. This distribution was based on data drawn from 5 weeks of actual schedules, shown in Appendix B also. Beta and gamma distributions calculated from the observed data are shown for comparison with the observed data and the modified empirical distribution used. As can be seen, the modified empirical distribution provides a better fit than the beta and gamma predictions.

The upper and lower limits of the distribution were truncated at 10.0 ML and 0.5 ML respectively, corresponding to the minimum and maximum parcel sizes generally observed. The continuous distribution was approximated by a discrete distribution of 20 bins of equal width. The mode of 2.0 ML was maintained, although its frequency was reduced slightly to ensure and area of 1 under the curve. The average and mean of the observed data and modified distribution are shown in Appendix B.

Product Distribution

The product type associated with each job was drawn from a distribution based upon the adjusted 1985 full year production mix described above. This distribution is shown in Appendix B.

Offtaker Distribution

The distribution of offtaker destinations was based upon an assessment of the amount of product transferred to each offtaker. This was estimated from the offtaker's overall market share and an assessment of how much of each offtaker's demand was sourced from the refinery. The distribution is shown in Appendix B.

A uniform distribution was used to allocate product between terminals for offtakers with multiple terminals.

Priority Distribution

No historical information was available to estimate this distribution. Instead, a discrete distribution was assumed in which jobs were assigned "high", "medium" or "low" priority. The probability of each of these categories was assumed to be 10, 80 and 10 per cent respectively.

Ready Time Distribution

In practice, a number of jobs are sourced from each product batch, and hence the job ready times are clustered at the batch release times. This characteristic was modelled in the simulation experiments by ensuring jobs of the same product type had the same ready time until the volume of the batch was exhausted. A new batch time was then established and the process repeated, as shown in Appendix B.

The batch release times were assumed to be uniformly distributed between 24 and 36 hours after the preceding batch release time for the product concerned. A constant batch size of 9.0 ML was assumed for each product.

Due Time Distribution

Two options were used for assigning job due times. In the first, the due time was offset from the job ready time by a multiple of the nominal job processing time. The nominal processing time equalled the job volume divided by the average processing rate for the particular offtaker / product combination. The multiplier used was specified by the user, thus allowing specific due date tightness scenarios to be simulated. In the second, the due times were offset randomly from the ready time.

Deadline Distribution

Job deadlines were not used in any of the simulation experiments. However, the job deadline could be specified in a similar manner to the due time if required. In the second option, however, the deadline was offset from the due time rather than the ready time. This ensured that the deadline never preceded the due time.

Offtaker Availability

The standard offtaker availability for normal and overtime working periods was assumed. This is shown in Appendix C.

Pipeline Availability

All pipelines were assumed to be available for the whole schedule horizon.

Batch Number and Tank Number

These are simply used for reporting purposes and do not affect the generation of the schedule. This data was ignored.

5.4.1 Performance Measures

The following statistics were collected to enable a meaningful comparison to be made between the experimental conditions studied. These statistics were collected for each simulation run of each of the cases studied, and the mean, variance, minimum and maximum were calculated for analysis.

Total Jobs

This is the total number of jobs required to be scheduled. As discussed earlier this number was not specified directly; instead jobs were selected until the estimated average pipeline load reached a given level.

Unfinished Jobs

This is the number of jobs whose processing has not been finished, i.e. not all of the volume associated with the job has been pumped to the offtaker.

Passes

This is the number of backtracking passes required by the algorithm to determine its best solution. This also includes the passes evaluated after the best solution has been found to establish that no better solution can reasonably be found.

Makespan

This is the time taken to complete all of the work in the schedule. If some work remained at the end of a schedule the makespan was set to the end time of the schedule.

First Value

This is the value of the objective function corresponding to the first solution found, i.e. at the end of the first pass.

Best Value

This is the value of the objective function corresponding to the best solution found.

Total Tardiness

This is the sum of the tardiness of each of the tardy jobs in each case.

Maximum Tardiness

This is the maximum tardiness observed for any single job in each case.

Normalized Weighted Tardiness

As defined by Vepsalainen and Morton [70], this is the total weighted tardiness divided by the product of the number of jobs, average number of operations per job, average processing time and average delay penalty per unit time.

Volume Required

This is the sum of the job volumes. The individual job volumes were drawn from a frequency distribution as discussed above.

Volume Remaining

This is the sum of the job volumes which had not been delivered at the end of the schedule.

Setup Count

This is the count of the number of setups required in the best schedule of each case.

Pipeline Utilization

This is the pipeline utilization observed, i.e. based upon the volume delivered and not the volume required. This is less than the load level specified by the user if some of the jobs remain unfinished at the end of the schedule horizon.

5.5 Simulation Results

Experimental Objectives

Four experiments were conducted:

6. Apparent Tardiness Cost Tuning

7. Cost Over Time Tuning

8. Pattern Selection Rule Evaluation

9. Scheduling Rule Evaluation

Data Smoothing

The raw data at each load level were smoothed using least squares linear regression to obtain a performance versus machine load trend. As discussed in Chapter 6 this technique simplified the data analysis, which would otherwise have been difficult due to the scatter in the experimental results.

5.5.1 Experiment 6: Apparent Tardiness Cost Tuning

Objective

The objective of this experiment was to observe the relative performance of the ATC rule as a function of k, based on simulation conditions.

Experimental Conditions

Objective Function Weighting: weighted tardiness

Job Priority Weighting: priority distribution

Pattern Priority Weighting: pattern generation order

ATC Rule Parameter k:

Varied from 2.0 to 6.0 in increments of 1.0

Results

The results are shown in Appendix E and summarized in Figures 5.6.1 and 5.6.2 below.

1. Weighted Tardiness

The best performance was observed with k=3.0, although similar performance was observed at high utilizations with k=4.0. Performance deteriorated as k was changed in either direction. Notably, the raw performance at k=1.0, 2.0, 5.0 and 6.0 was identical for all pipeline loads considered.

2. Volume Delivered

Again the best performance was observed with k=3.0 for all pipeline loads (smoothed data). The performance at k=4.0 tracked that at k=3.0, but with a constant offset of approximately 1.7 ML. Performance

deteriorated as k was changed in either direction, and this deterioration became more marked as the pipeline load increased. Notably, the raw performance at k=1.0, 2.0, 5.0 and 6.0 was identical for all pipeline loads considered.

Conclusion

The best overall performance is achieved with k=3.0.

An experiment was conducted in which the value of k was varied from 1.0 to 6.0 in increments of 0.5 at a fixed pipeline load of 65 %. In all cases the performance at fractional values of k was identical to the performance at the corresponding integer value (i.e. the performance at k=3.5 was the same as at k=3.0, etc).

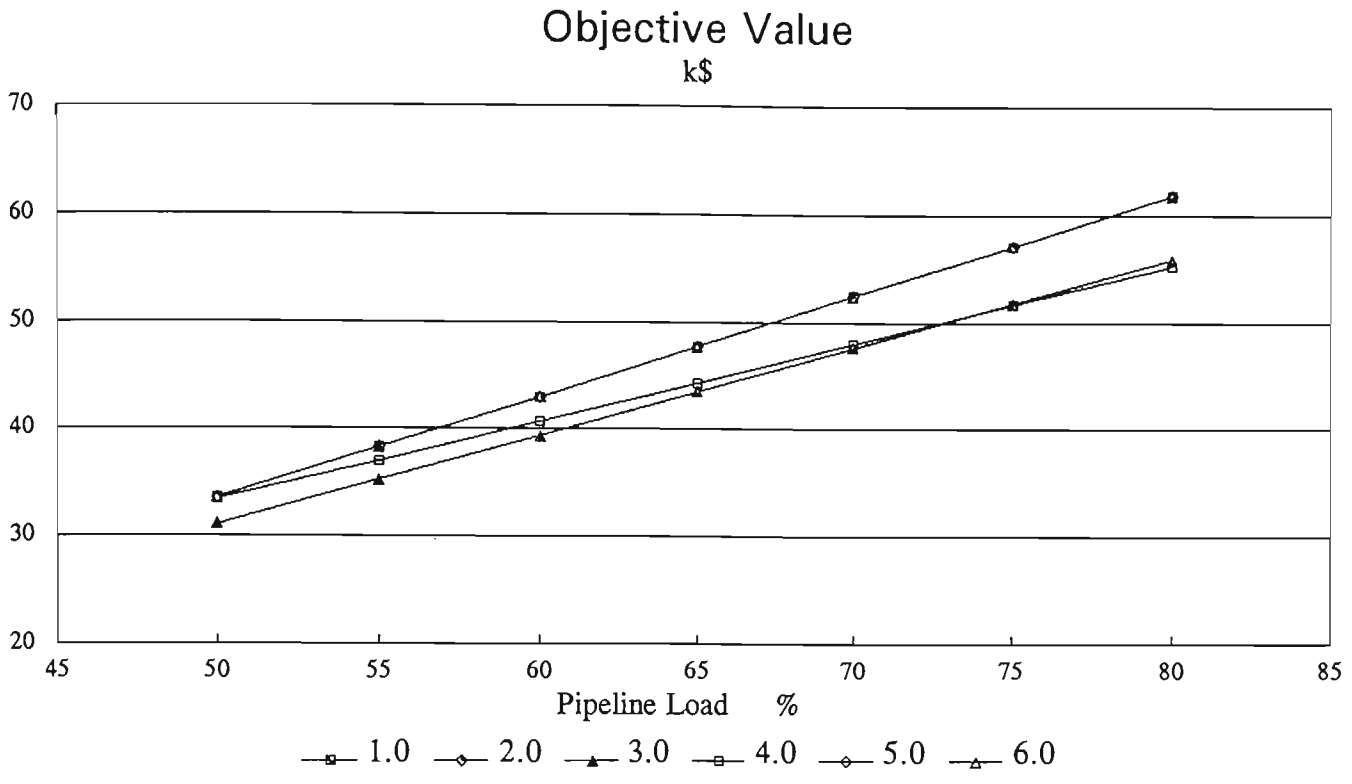


Figure 5.6.1 Tardiness response to "look-ahead" parameter k

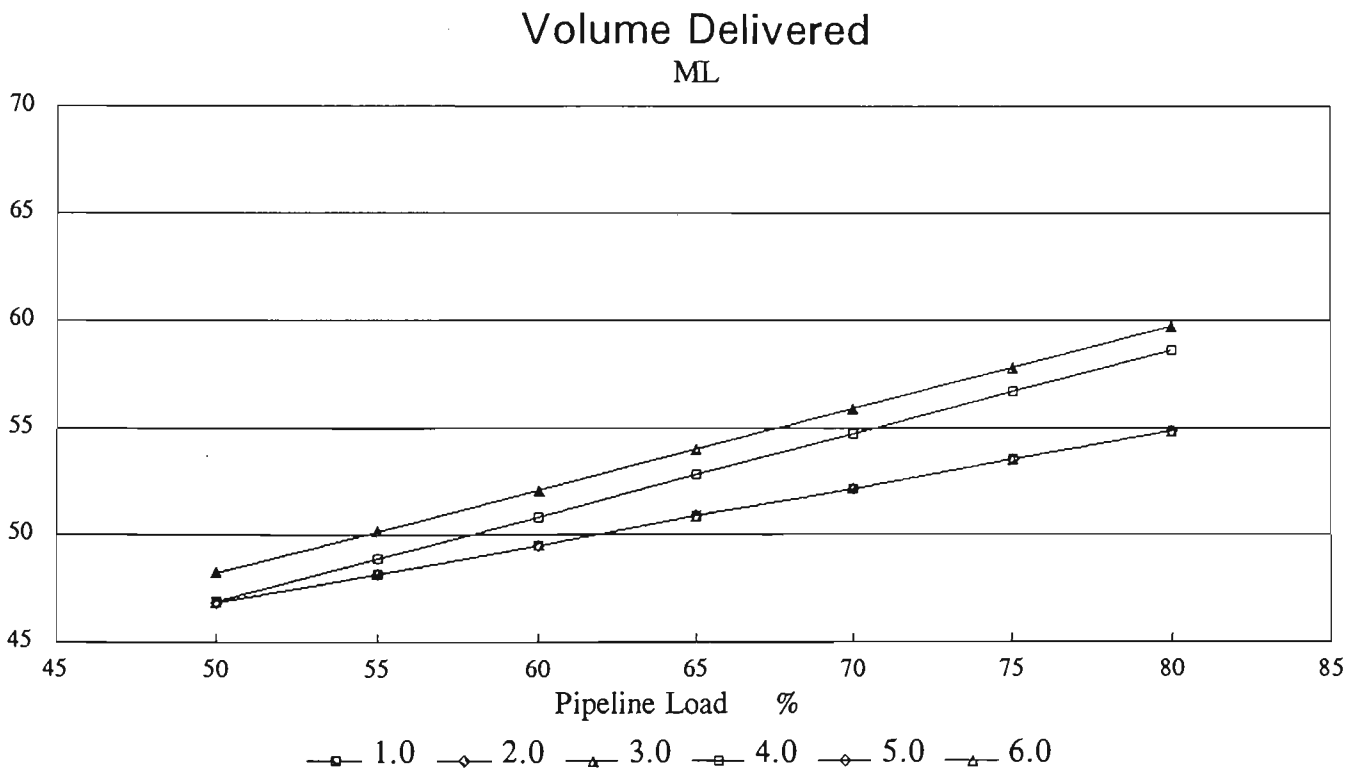


Figure 5.6.2 Volume Delivered response to "look-ahead" parameter k

5.5.2 Experiment 7: Cost Over Time Tuning

Objective

The objective of this experiment was to observe the relative performance of the COT rule as a function of k*b, based on simulation conditions.

Experimental Conditions

Objective Function Weighting: weighted tardiness

Job Priority Weighting: priority distribution

Pattern Priority Weighting: pattern generation order

COT Rule Parameter k*b:

Varied from 1.0 to 6.0 in increments of 1.0.

Results

The results are shown in Appendix E and summarized in Figures 5.7.1 and 5.7.2 below.

1. Weighted Tardiness

The best performance was not as clear as in the ATC experiment. The best at low pipeline loads was observed with k*b=3.0, however, at higher loads, both k*b=4.0 and 5.0 were better. Looking at the raw data, the performance at k*b=4.0 was more consistent than at k*b=5.0 as the pipeline load varied, however, once smoothed these two values gave similar results. The performance deteriorated as k*b changed in either direction.

2. Volume Delivered

Again the best performance shifted depending on the pipeline load. At low loads, k*b=3.0 gave significantly better performance than the other values. However, beyond 70 % load, both k*b=4.0 and 5.0 gave better performance. The raw data shows the relatively wide variation in performance of this rule at almost all values as the pipeline load varied. A common observation was the oscillation in performance as the load changed, particularly above 60 % load.

Conclusion

The best overall performance was achieved with k*b=3.0. However, the relative performance is sensitive to pipeline load and the value of k*b should be chosen to match the expected load range.

An experiment was conducted in which the value of k*b was varied from 1.0 to 6.0 in increments of 0.5 at a fixed pipeline load of 65 %. In all cases the performance at fractional values of k*b was identical to the performance at the corresponding integer value (i.e. the performance at k*b=3.5 was the same as at k*b=3.0, etc).

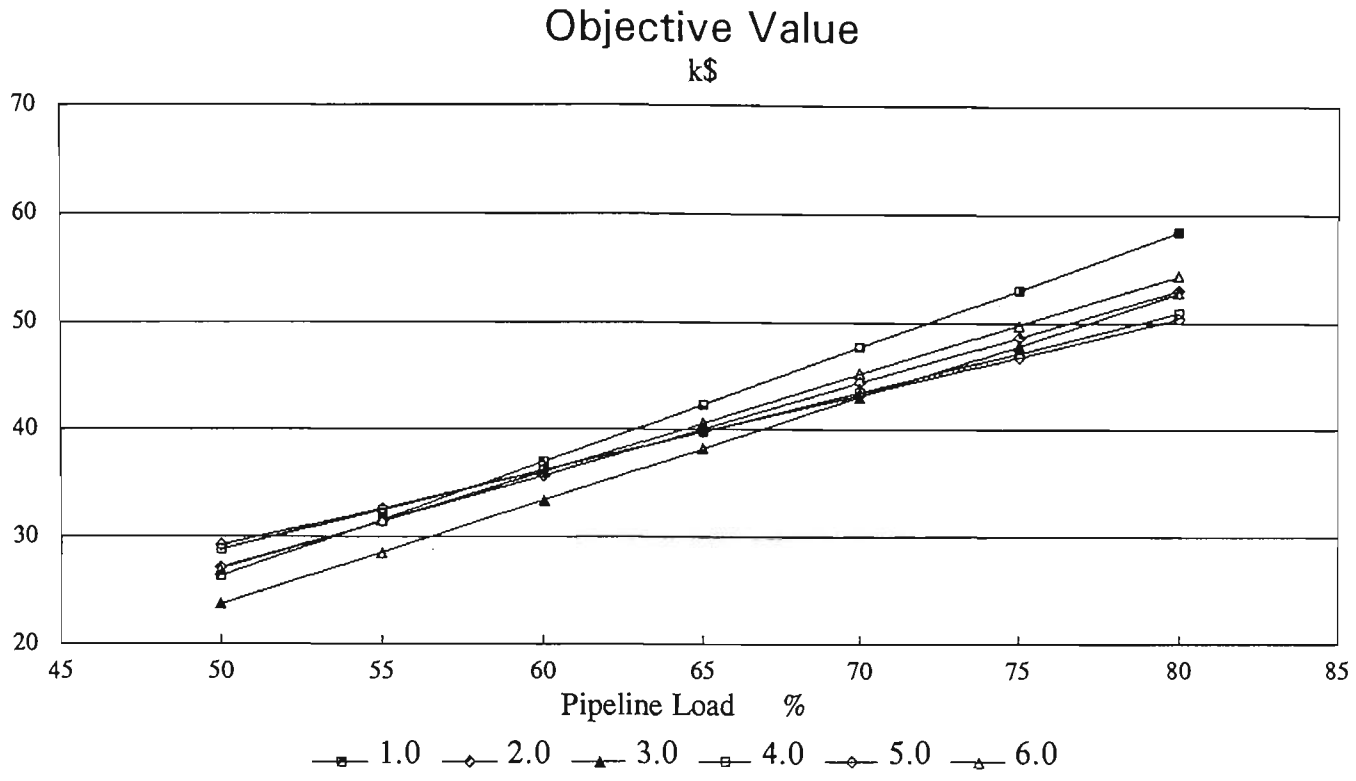


Figure 5.7.1 Tardiness response to "waiting time" parameter k*b

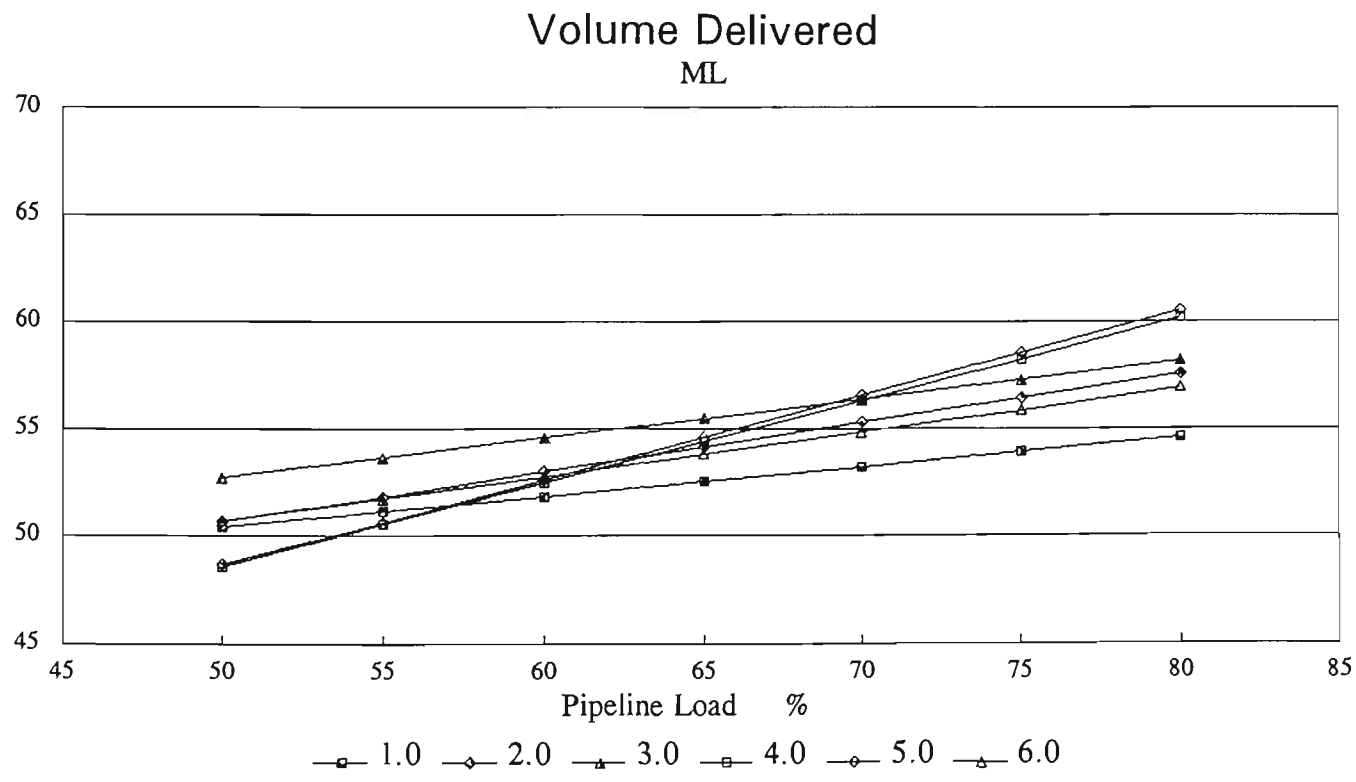


Figure 5.7.2 Volume Delivered response to "waiting time" parameter k*b

5.5.3 Experiment 8: Pattern Selection Rule Performance

Objective

The objective of this experiment was to observe the effect of pattern utilization on scheduling rule performance. The ATC and COT rules observed with k=3.0 and k*b=3.0 respectively at all load levels.

Experimental Conditions

Objective Function Weighting: weighted tardiness

Job Priority Weighting: priority distribution

Pattern Priority Weighting: pattern generation order/utilization

(pattern order was used to break ties in utilization)

Results

The results are shown in Appendix E and summarized in Figures 5.8.1.1 to 5.8.9.2.

1. Weighted Tardiness

The effect of pattern utilization was inconsistent between scheduling rules. The performance of the SPT and LOAD rules was favourably affected at all pipeline loads. The performance of the EDD, SLACK and WTAR rules was slightly unfavourable at low loads, but was marginally favourable at high loads. The ATC and COT rules were unfavourably affected at all loads, whilst the RANDOM and FCFS rules were unchanged.

The effect of pattern utilization was sometimes inconsistent for constant scheduling rules under varying load conditions. This behaviour was observed with all rules except SPT (the result at 50 % load was effectively unchanged).

2. Volume Delivered

Again the effect of pattern utilization was inconsistent between scheduling rules. A general improvement was observed for the RANDOM, FCFS and SPT rules. In the latter rule an improvement was observed at all load levels; in the RANDOM rule at all load levels except 60 %, and in FCFS at all except 50 and 70 %. In the EDD and SLACK rules the effect was inconsistent, both in the raw and smoothed data. The performance improvement in the smoothed data for the LOAD rule was apparent although the raw data was oscillated. In the WTAR rule the effect was favourable at high load levels and inconsistent at low loads. The effect was generally unfavourable for the ATC and COT rules, particularly in the former.

Conclusion

The effect of pattern utilization on overall performance is dependent upon the scheduling rule used. Some rules, most notably the SPT rule are favourably affected at all load levels, while the performance of other rules is unchanged or deteriorates.

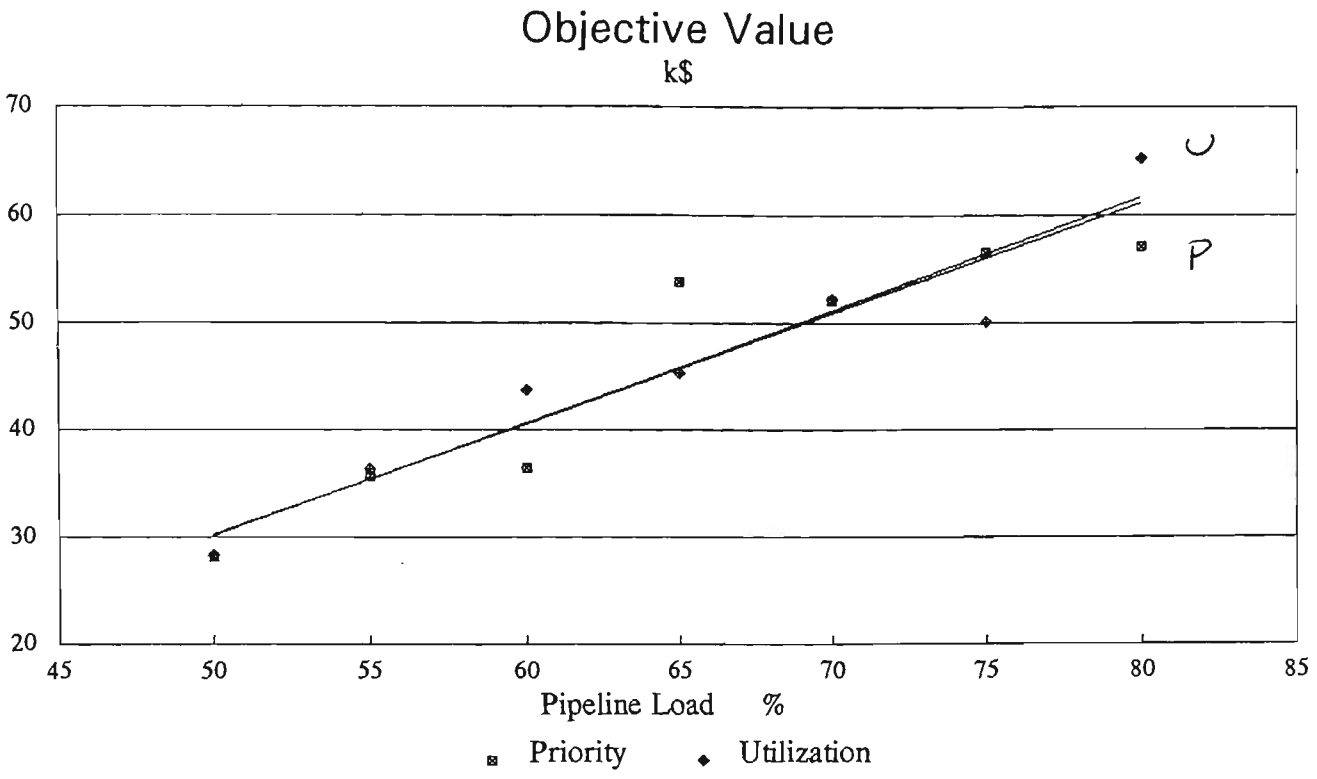


Figure 5.8.1.1 Weighted Tardiness response to pattern selection: Random

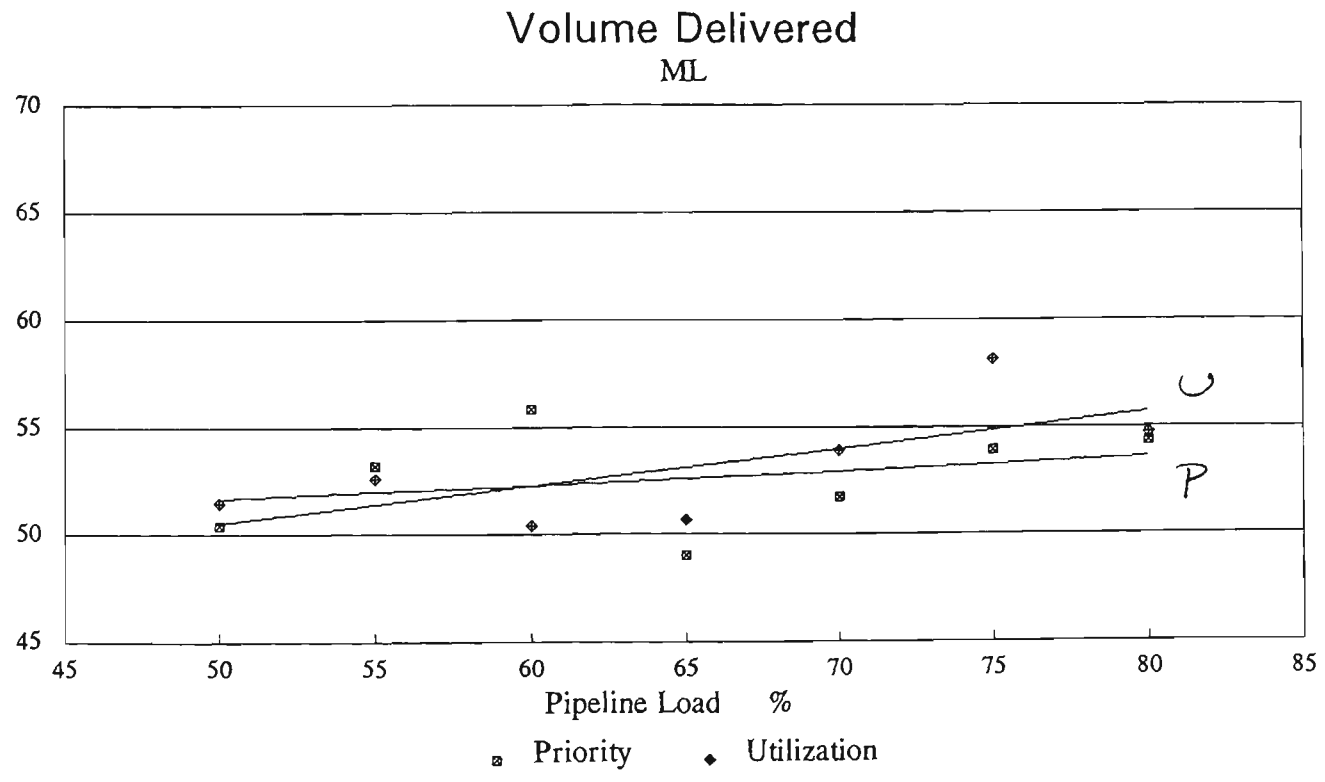


Figure 5.8.1.2 Volume Delivered response to pattern selection: Random

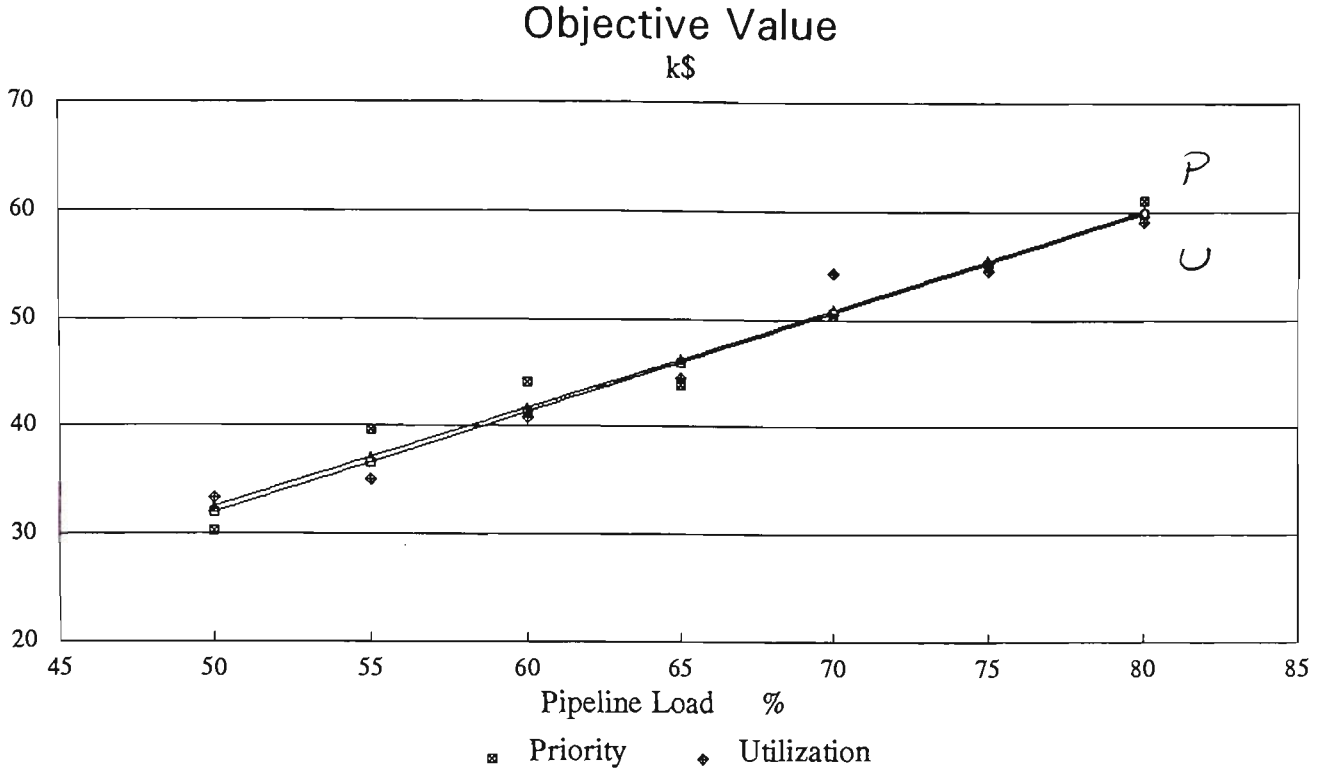


Figure 5.8.2.1 Weighted Tardiness response to pattern selection: FCFS

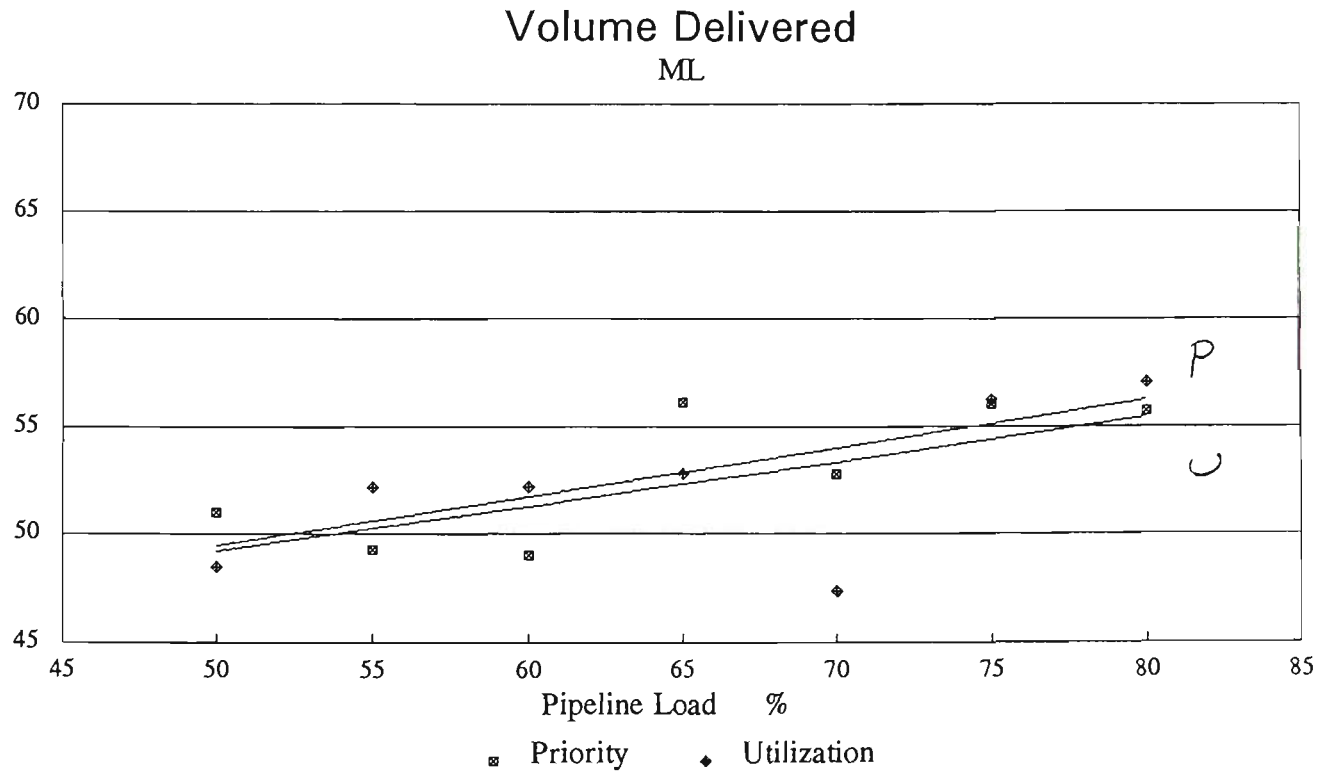


Figure 5.8.2.2 Volume Delivered response to pattern selection: FCFS

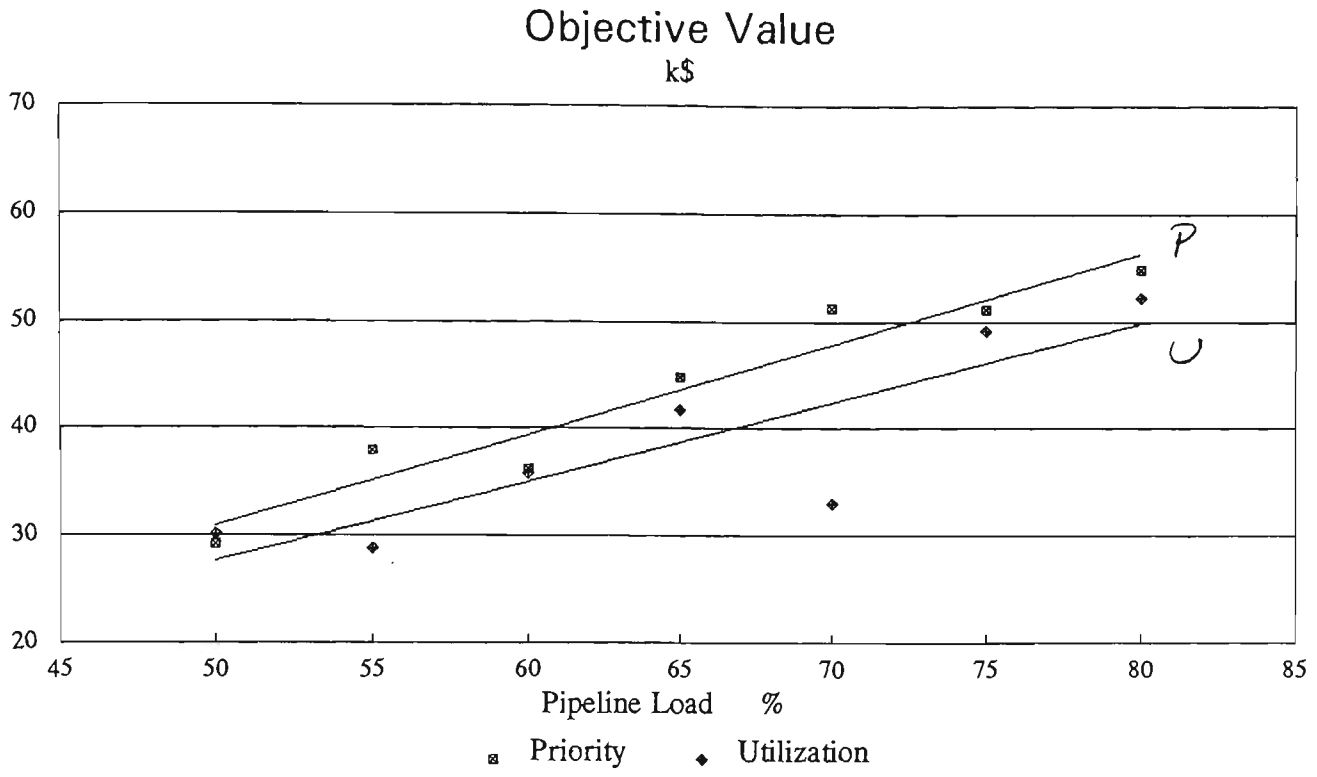


Figure 5.8.3.1 Weighted Tardiness response to pattern selection: SPT

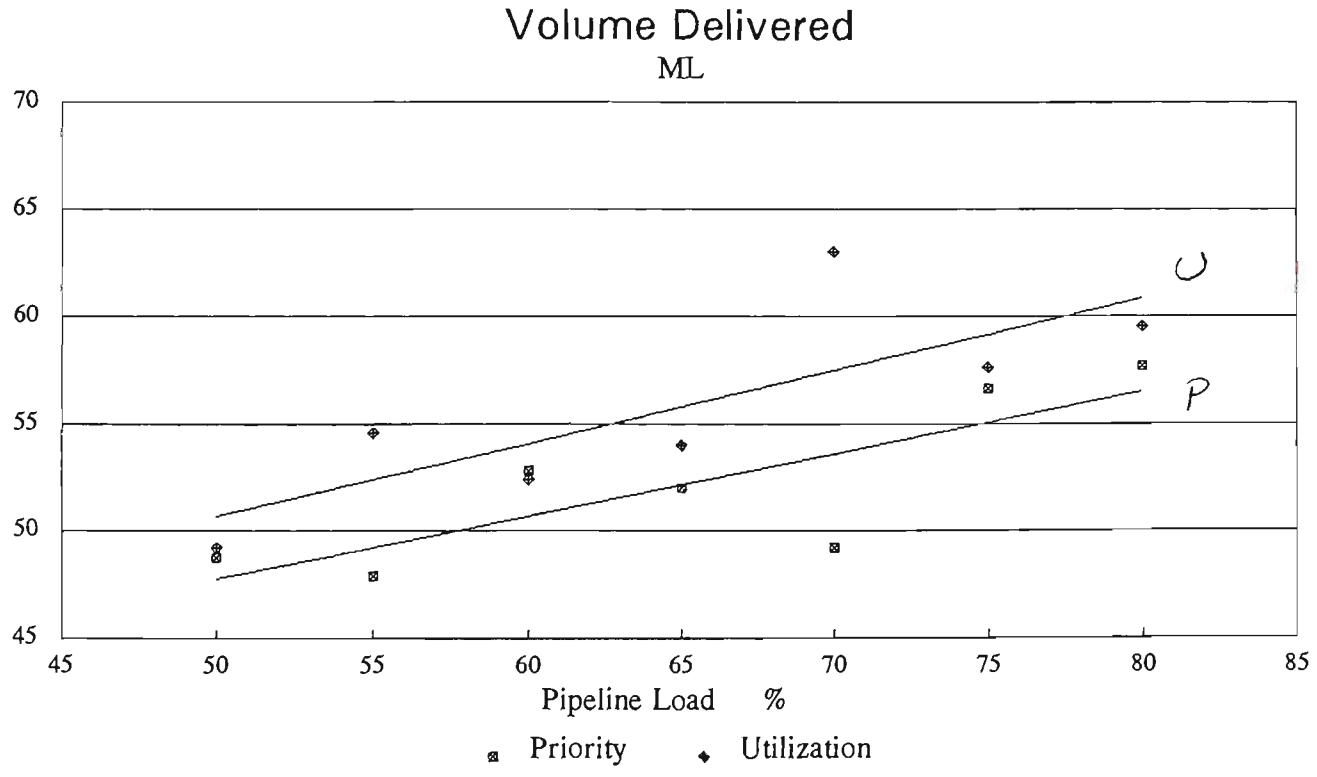


Figure 5.8.3.2 Volume Delivered response to pattern selection: SPT

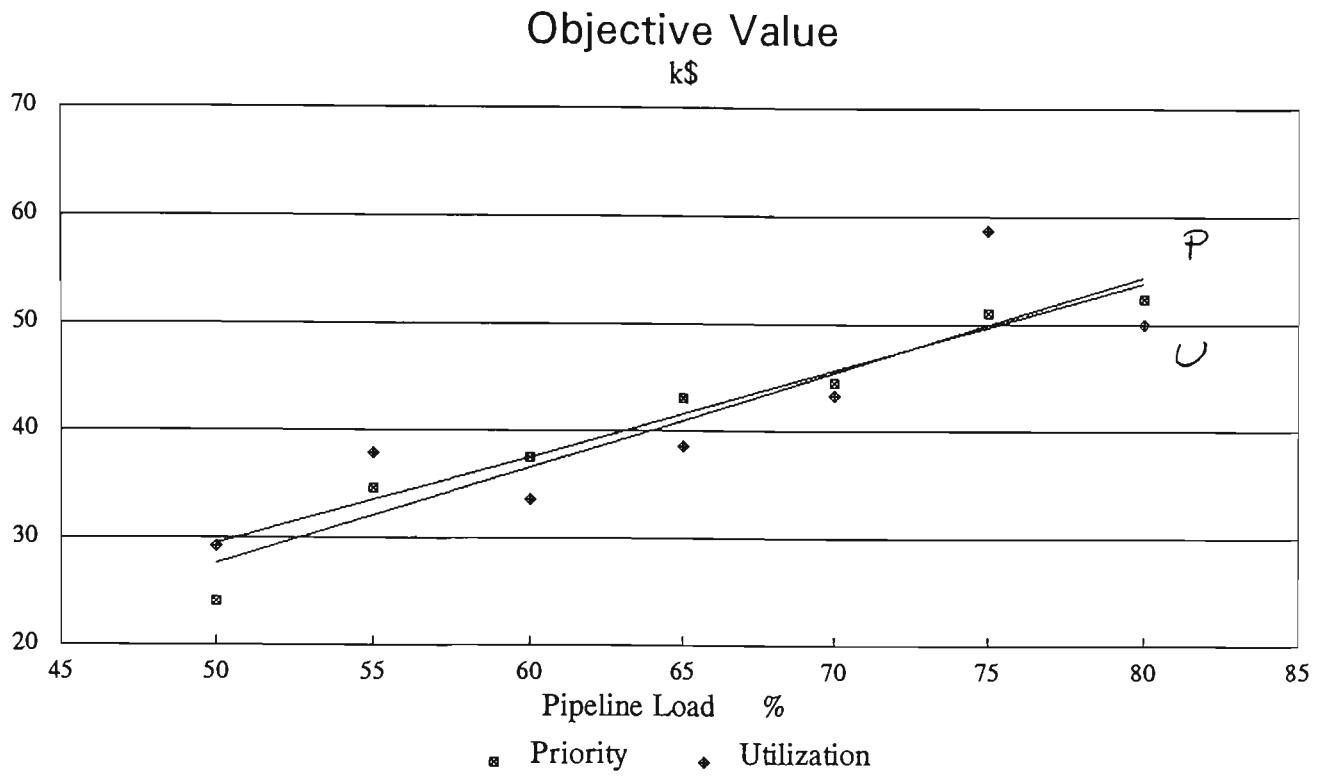


Figure 5.8.4.1 Weighted Tardiness response to pattern selection: EDD

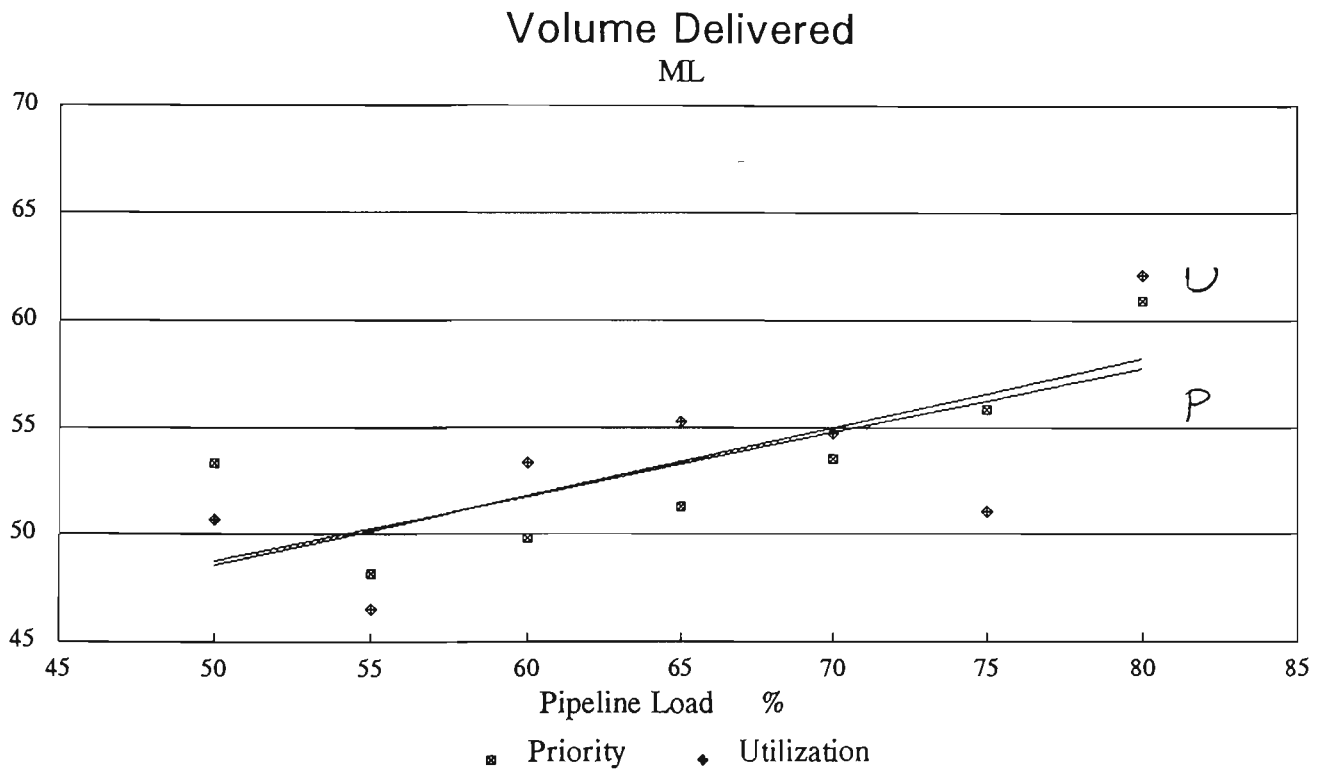


Figure 5.8.4.2 Volume Delivered response to pattern selection: EDD

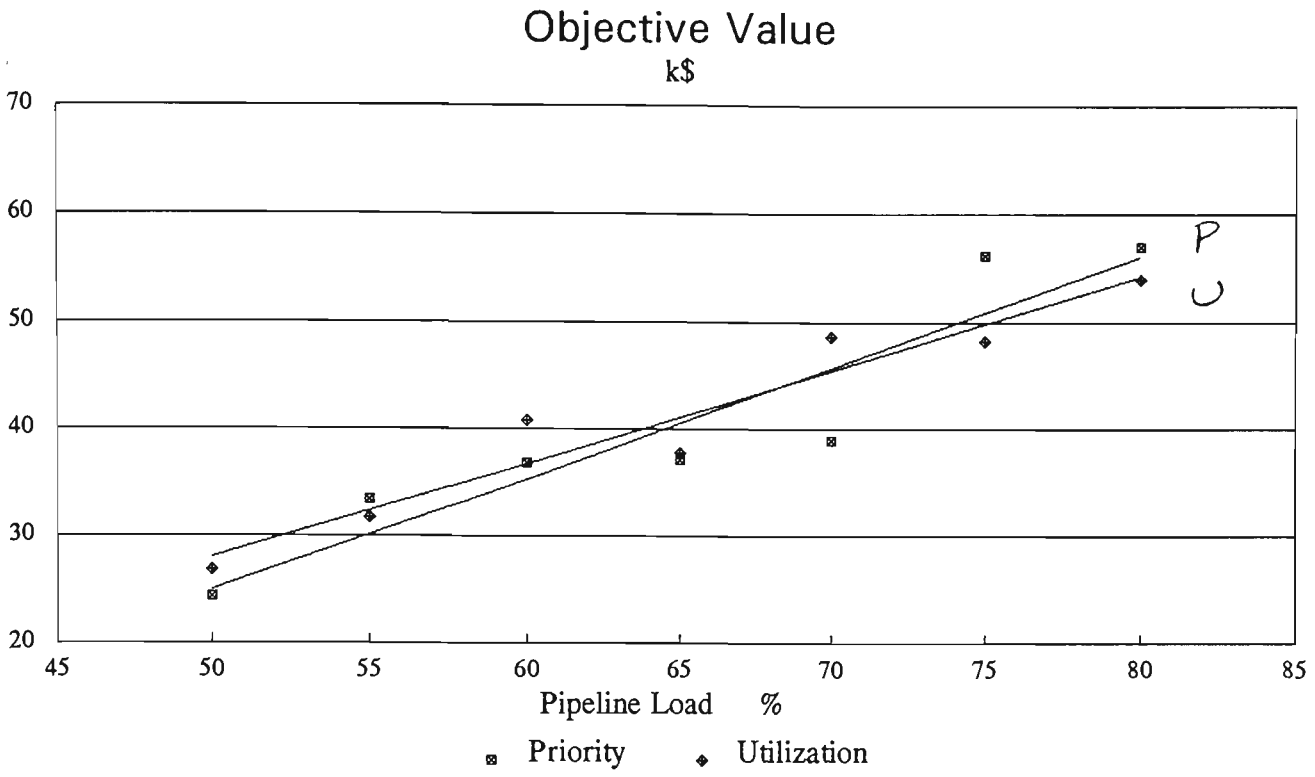


Figure 5.8.5.1 Weighted Tardiness response to pattern selection: SLACK

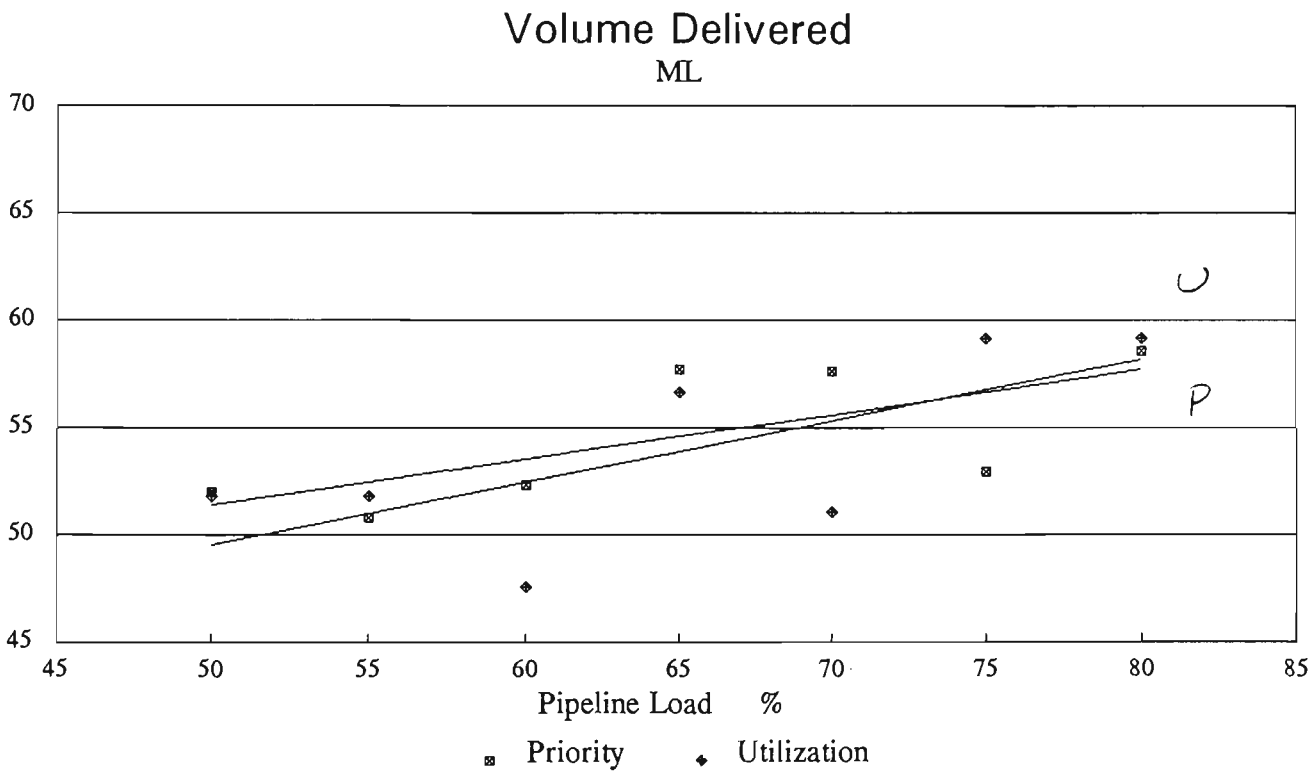


Figure 5.8.5.2 Volume Delivered response to pattern selection: SLACK

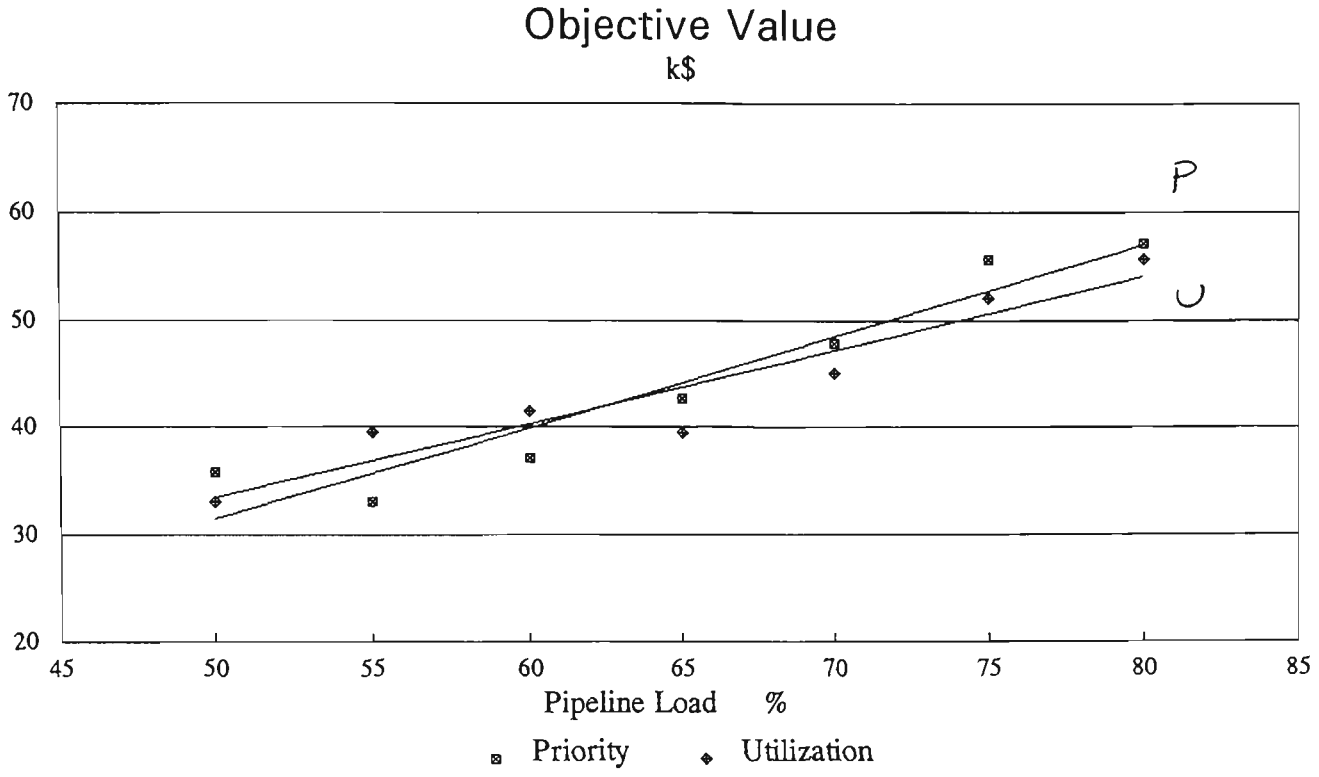


Figure 5.8.6.1 Weighted Tardiness response to pattern selection: WTAR

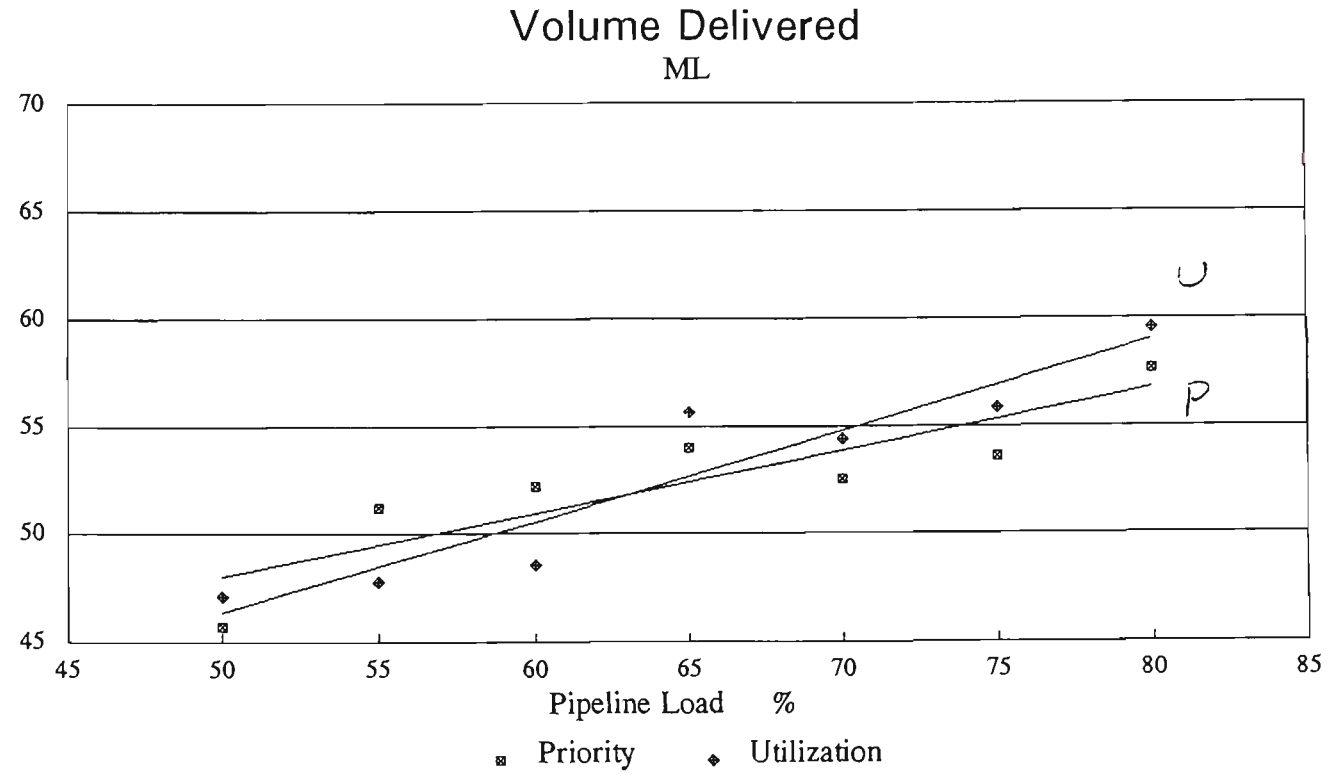


Figure 5.8.6.2 Volume Delivered response to pattern selection: WTAR

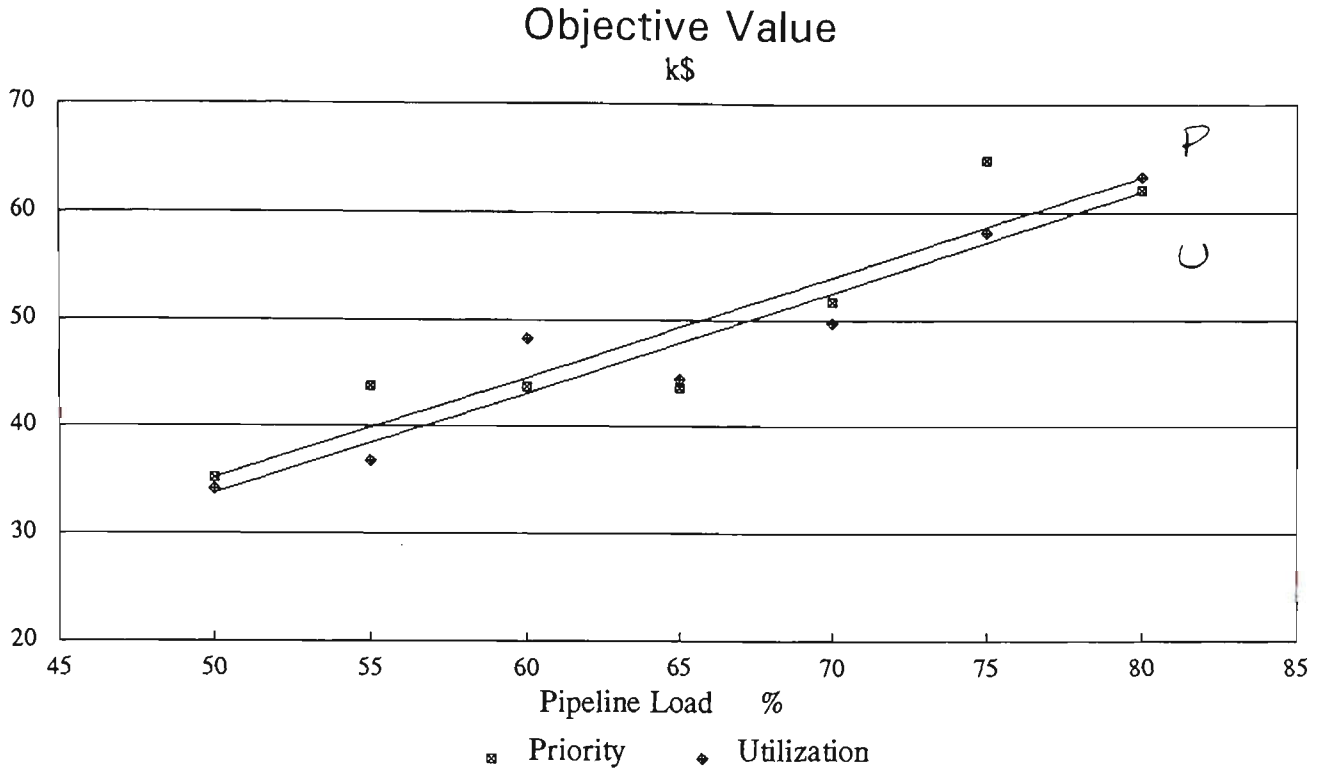


Figure 5.8.7.1 Weighted Tardiness response to pattern selection: LOAD

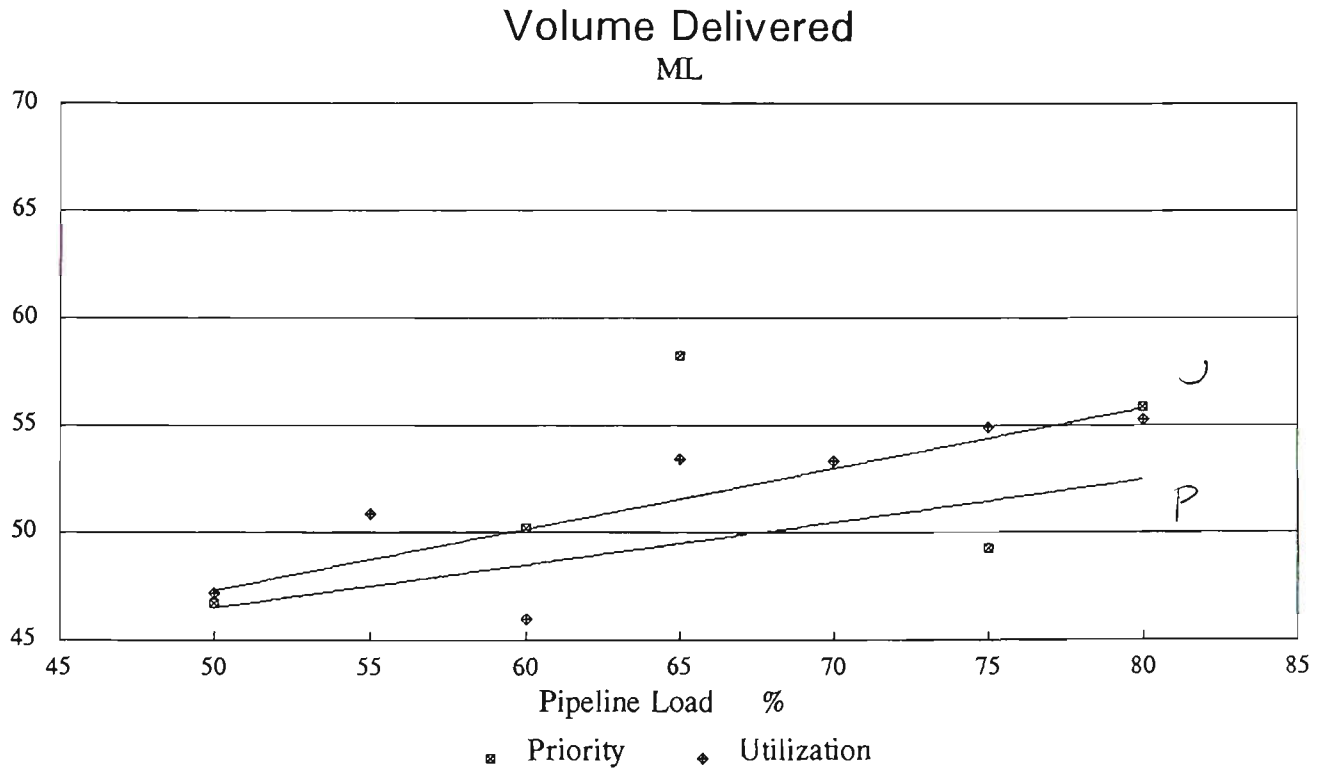


Figure 5.8.7.2 Volume Delivered response to pattern selection: LOAD

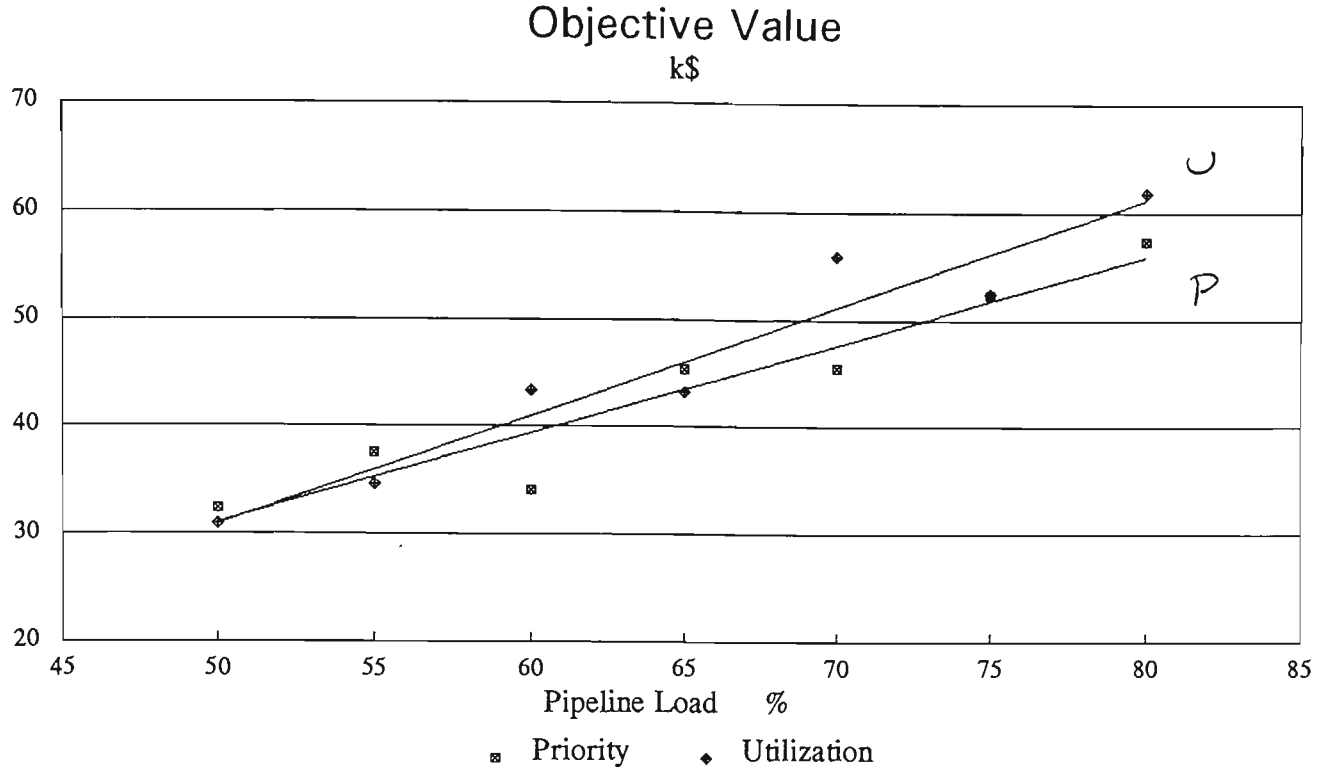


Figure 5.8.8.1 Weighted Tardiness response to pattern selection: ATC

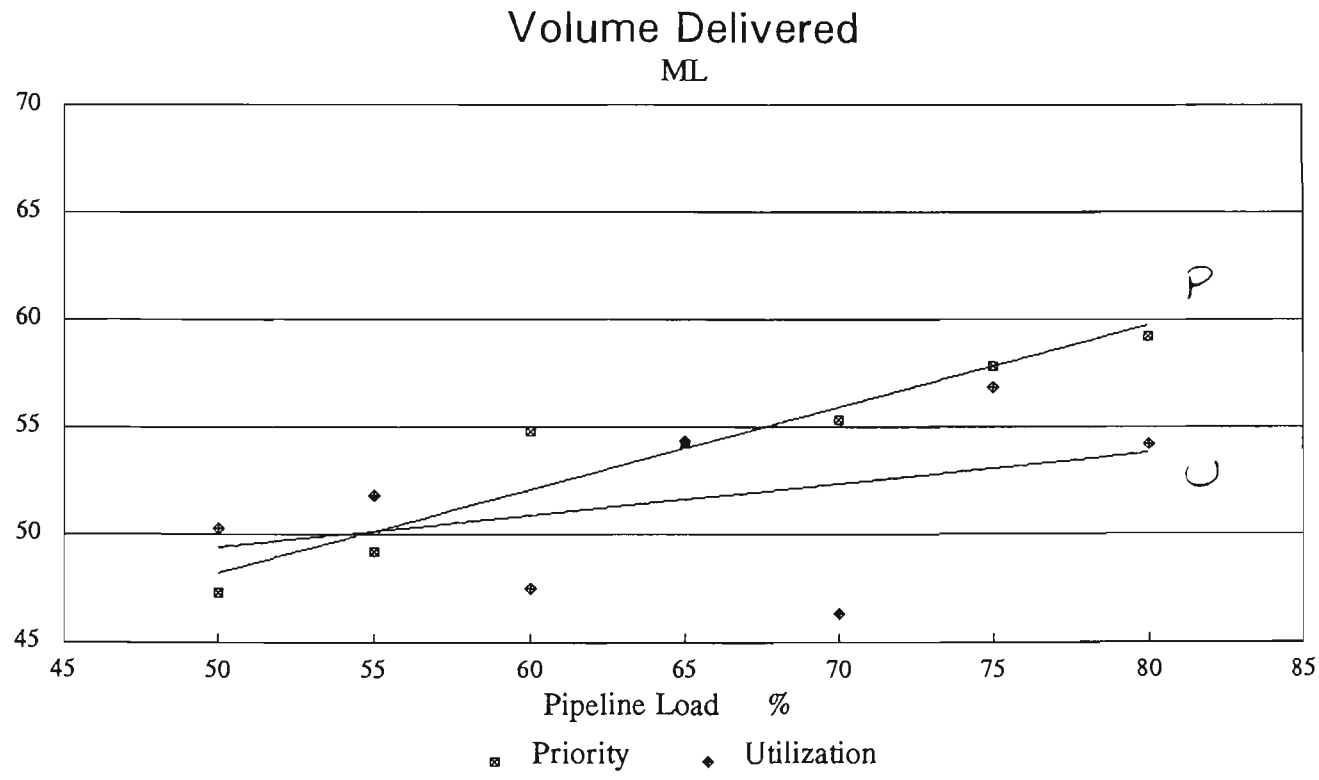


Figure 5.8.8.2 Volume Delivered response to pattern selection: ATC

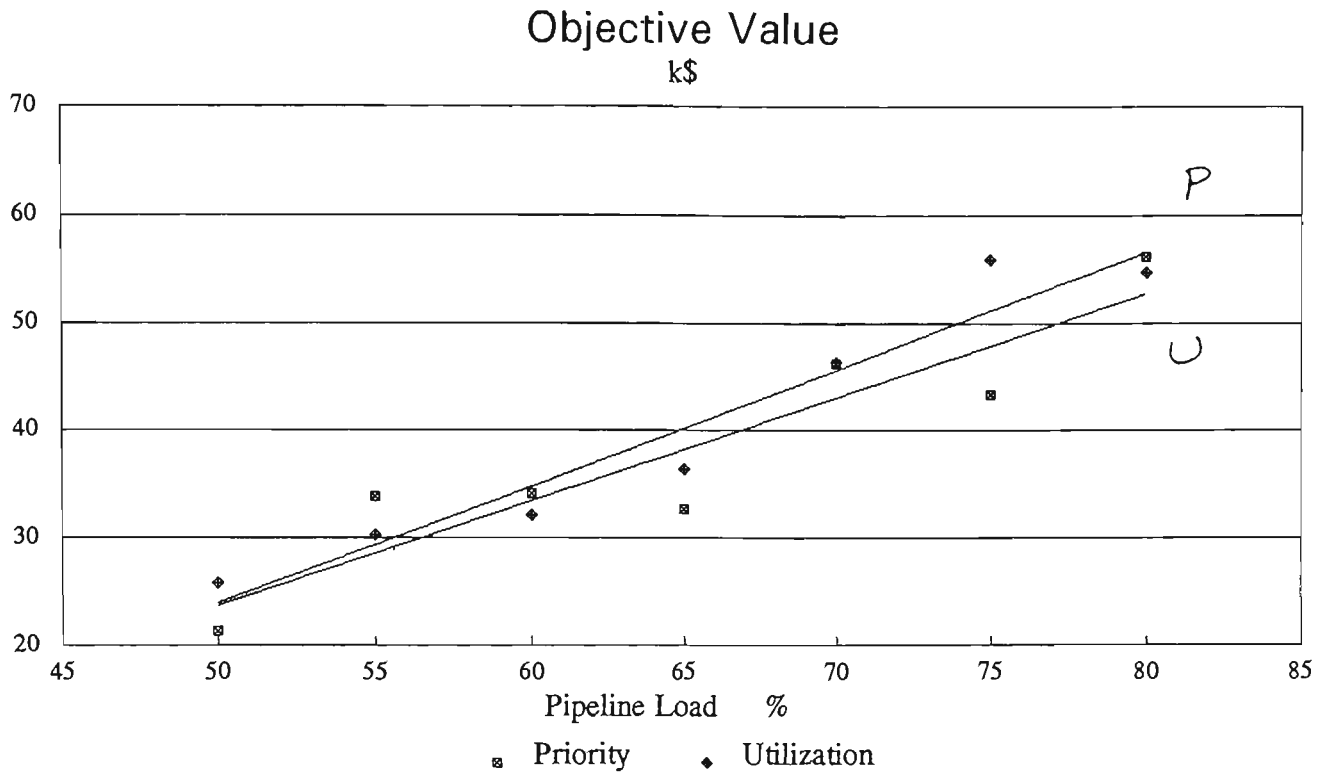


Figure 5.8.9.1 Weighted Tardiness response to pattern selection: COT

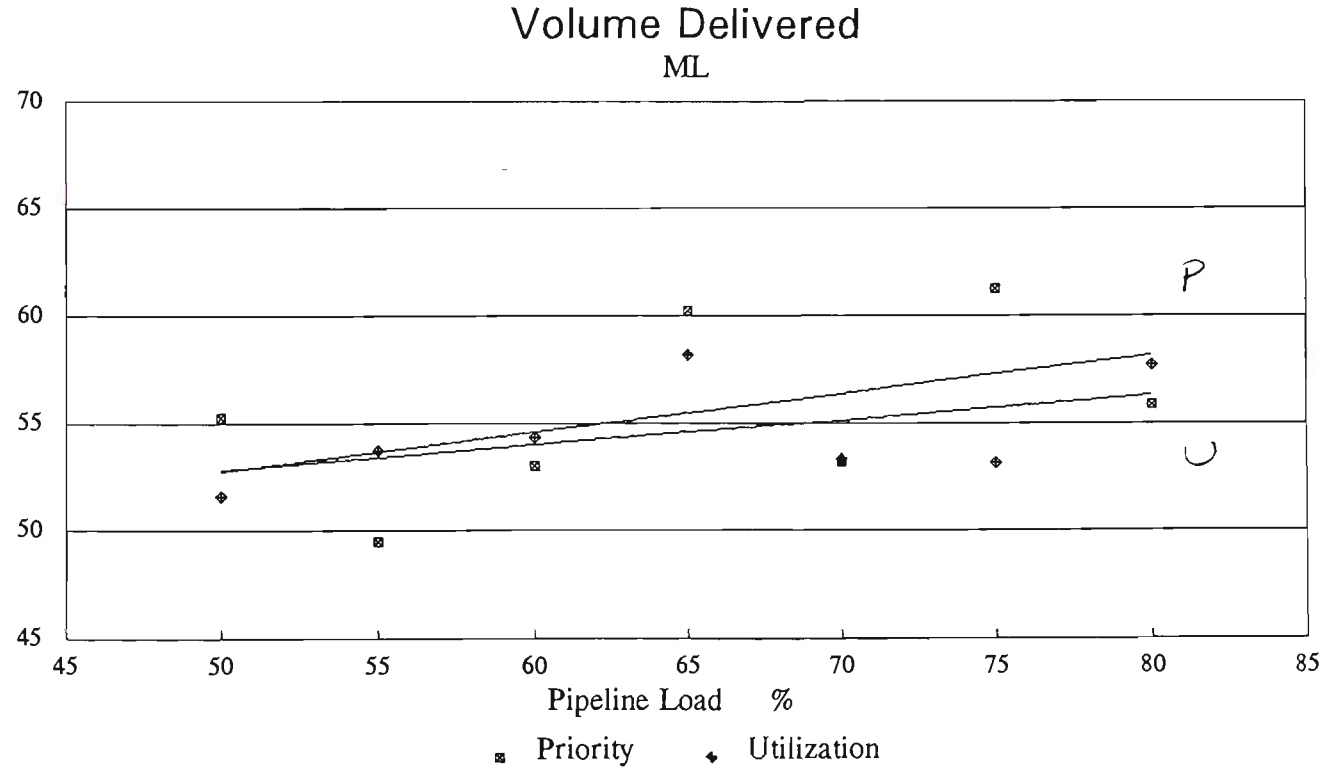


Figure 5.8.9.2 Volume Delivered response to pattern selection: COT

5.5.4 Experiment 9: Job Scheduling Rule Performance

Objective

The objective of this experiment was to observe the relative performance of selected job scheduling rules. The ATC and COT rules were observed with k=3.0 and k*b=3.0 respectively at all load levels.

Experimental Conditions

Objective Function Weighting: weighted tardiness

Job Priority Weighting: priority distribution

Pattern Priority Weighting: pattern generation order/utilization

(based on performance in Experiment 8)

(pattern order was used to break ties in utilization)

Results

The results are shown in Appendix E and summarized in Figures 5.9.1 to 5.9.12 below.

1. Weighted Tardiness

The best performance at low loads was achieved with the COT rule. At loads greater than 65 % the SPT rule was superior. The performance of the SLACK rule was good at low load levels, but deteriorated at a greater rate with increasing load level. The FCFS rule was similar to RANDOM, whilst the LOAD rule was poorer. The WTAR rule, although poor at low load levels, was better than most rules at high pipeline loads, approaching the performance of the COT rule. The performance of the EDD and ATC rules was about average, improving relative to most of the other rules as the load level increased.

3. Volume Delivered

Again the best performance at low load levels was achieved with the COT rule. At loads greater than 65 % the SPT rule was superior. It was difficult to distinguish the relative performance of these rules based on the raw data: depending on the load level one rule would perform significantly better than the other, however, no trend was apparent.

The performance of the SLACK rule was good at low load levels, but did not keep pace with the improvement of the SPT rule as the load level increased. The FCFS rule was slightly worse than the RANDOM rule. The EDD rule performed poorly at low load levels (worse than RANDOM), but increased at a similar rate to the SPT rule (increasing relative to most of the other rules) with increasing load level, and hence was one of the better performing rules at high loads. The WTAR, LOAD and ATC rules all performed more poorly than the RANDOM rule at low load levels. However, at high loads the WTAR and ATC rules were better. The performance of the LOAD rule improved relative to the RANDOM rule with increasing pipeline loads, but lagged the other rules.

Conclusion

The SPT and COT rules performed the best. However, it is difficult to distinguish between the relative performance of these two rules. This is discussed further in Chapter 6.

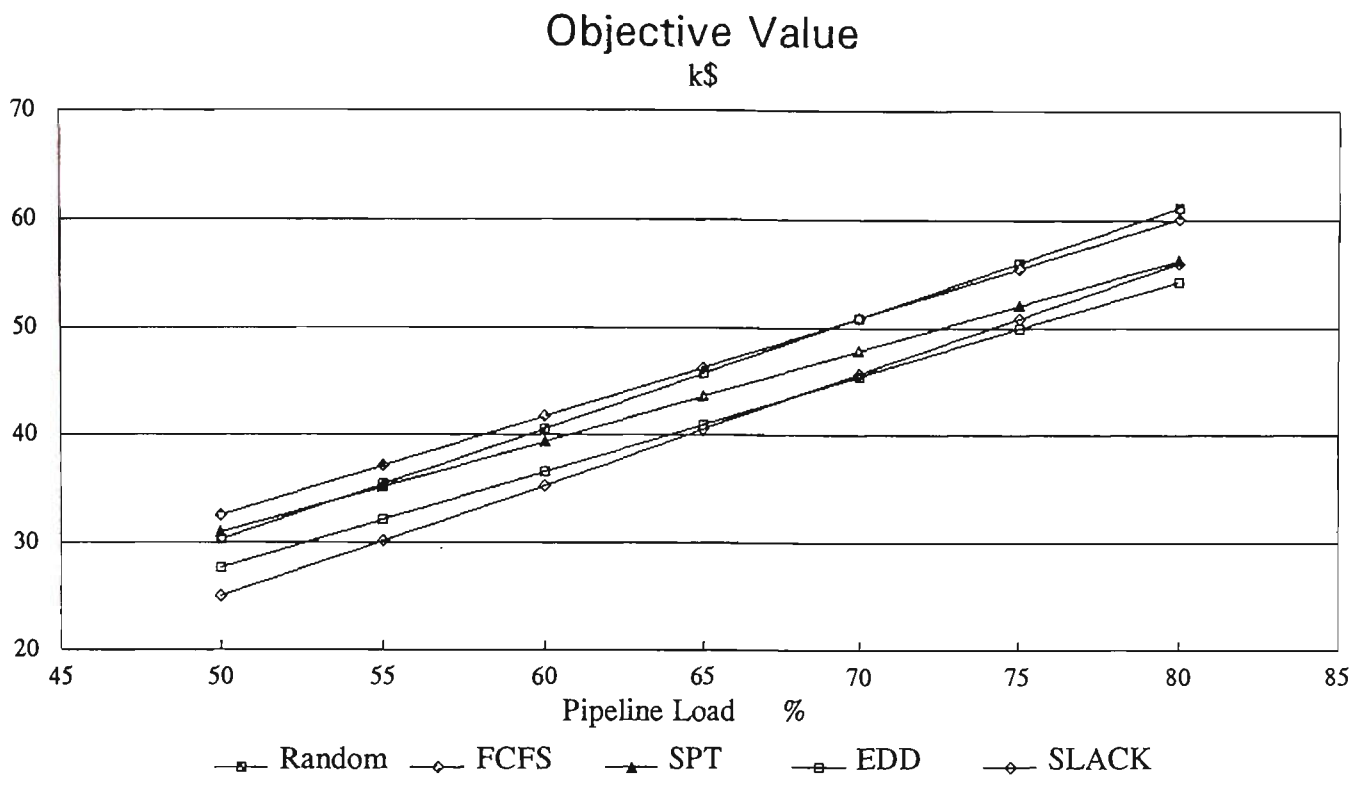


Figure 5.9.1 Performance based on pattern priority

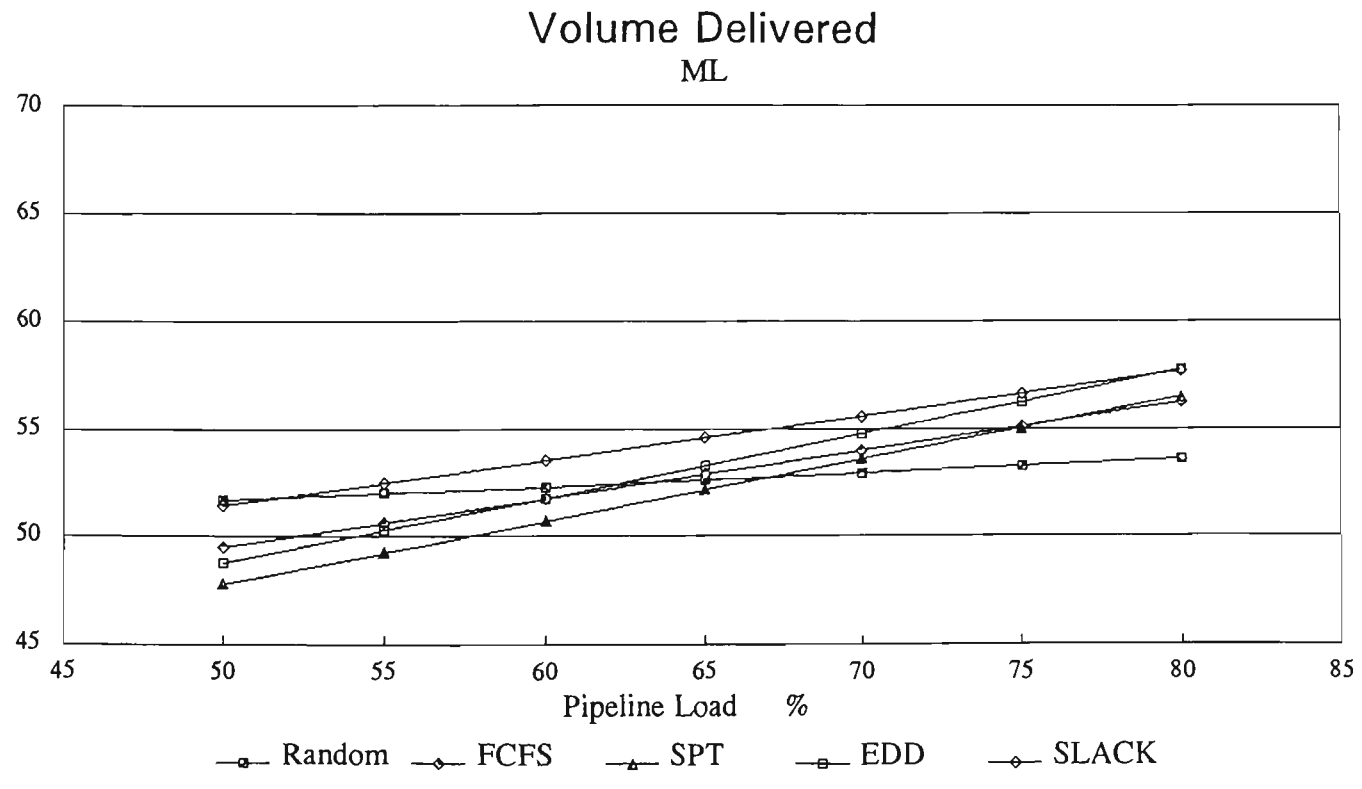


Figure 5.9.2 Performance based on pattern priority

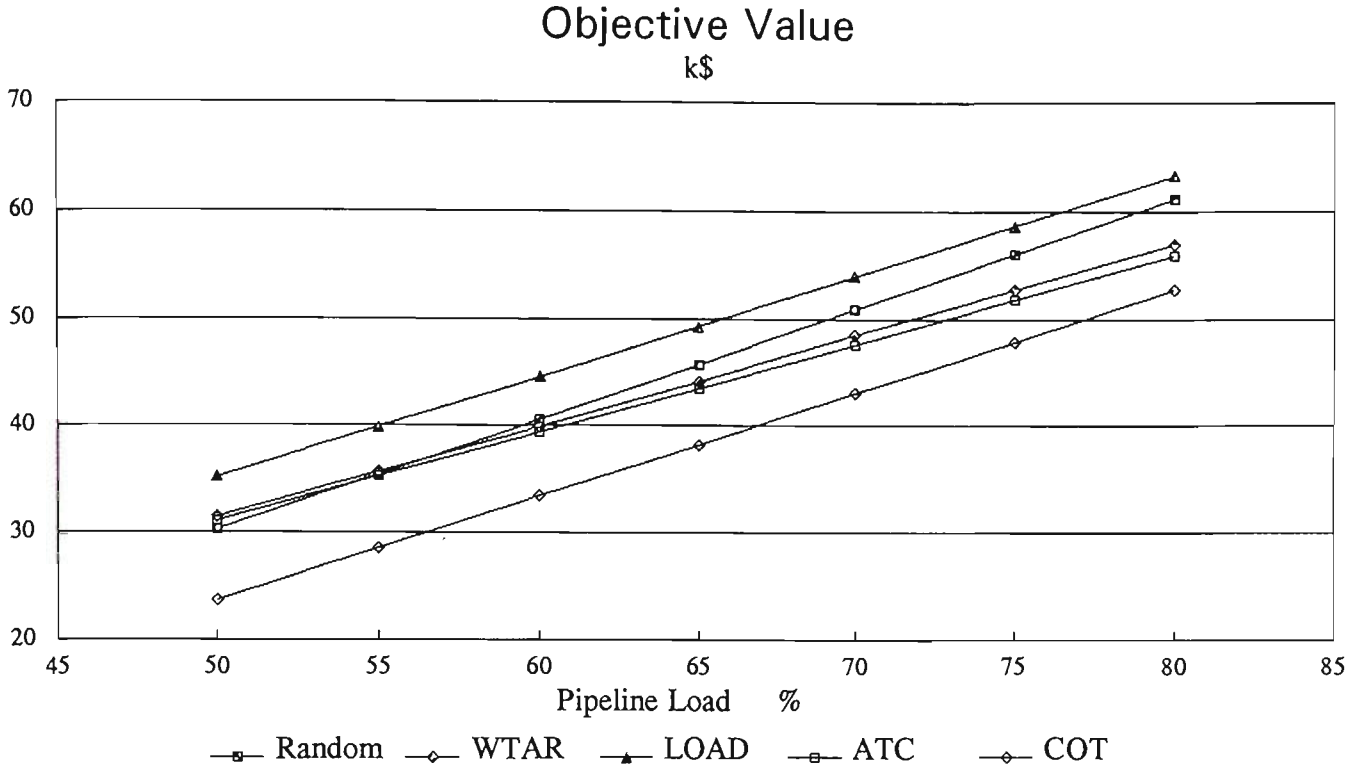


Figure 5.9.3 Performance based on pattern priority

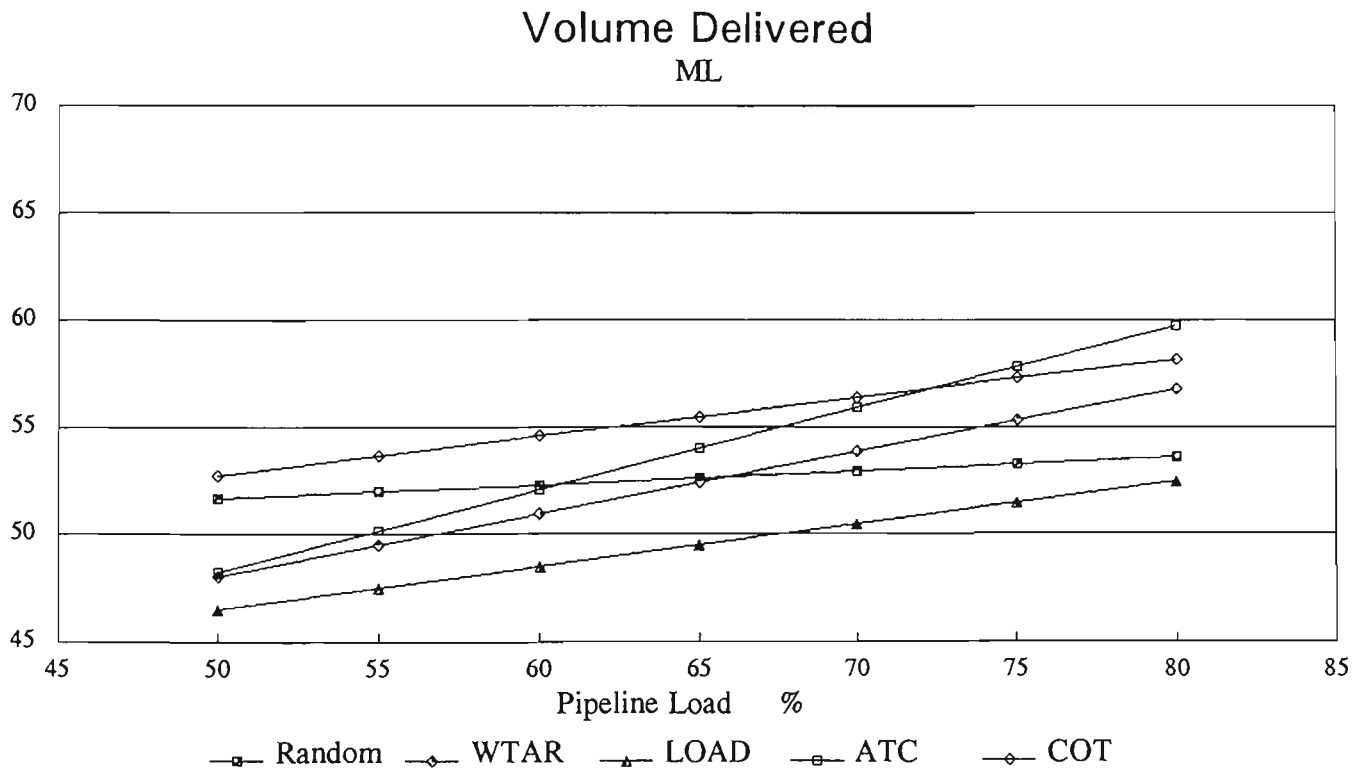


Figure 5.9.4 Performance based on pattern priority

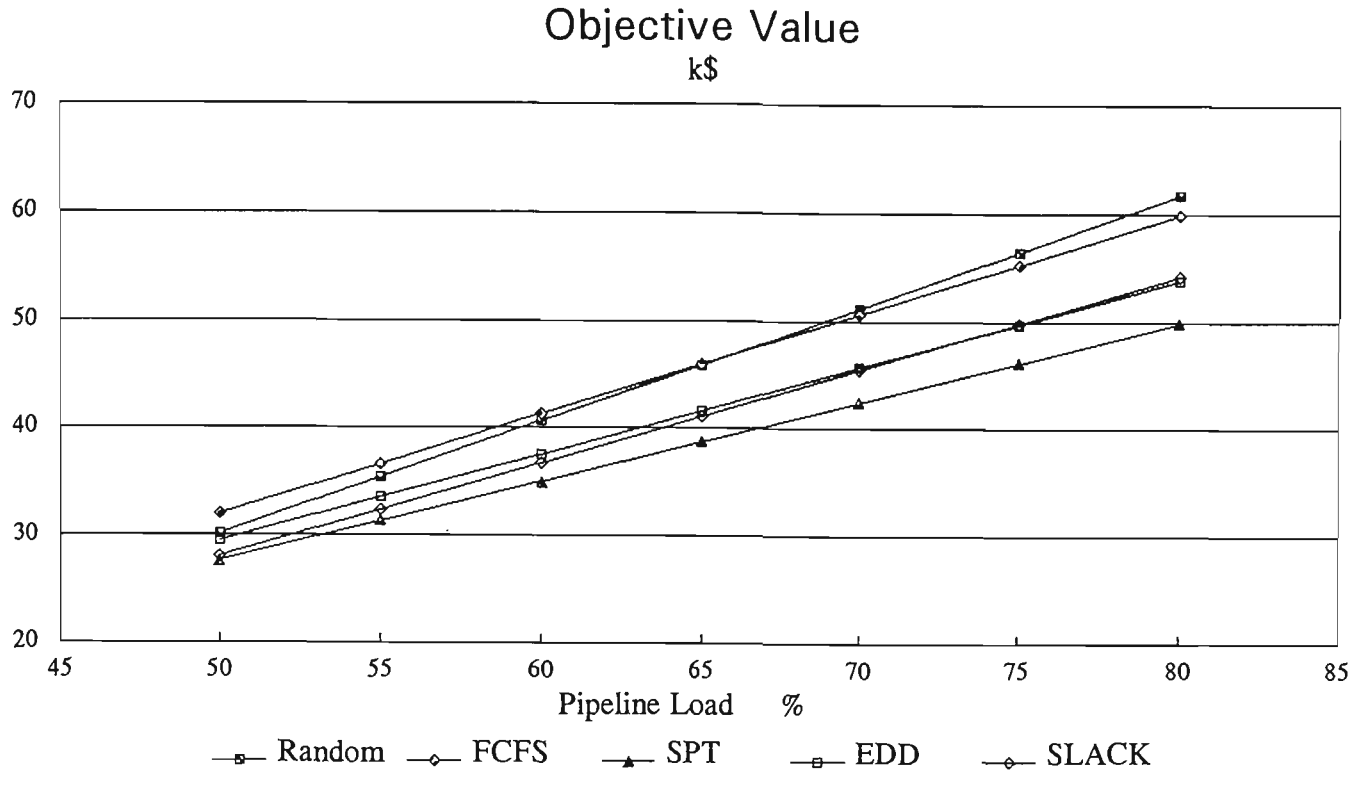


Figure 5.9.5 Performance based on pattern utilization

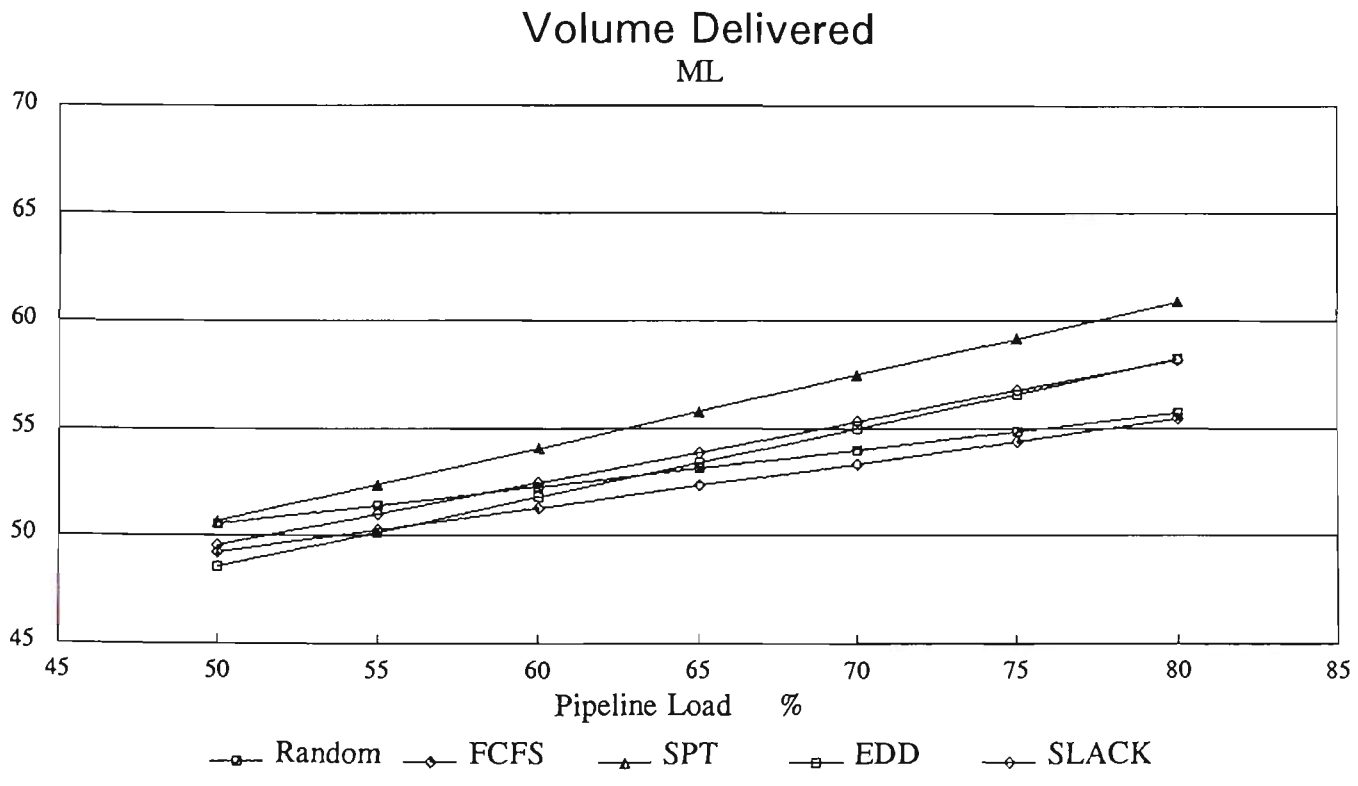


Figure 5.9.6 Performance based on pattern utilization

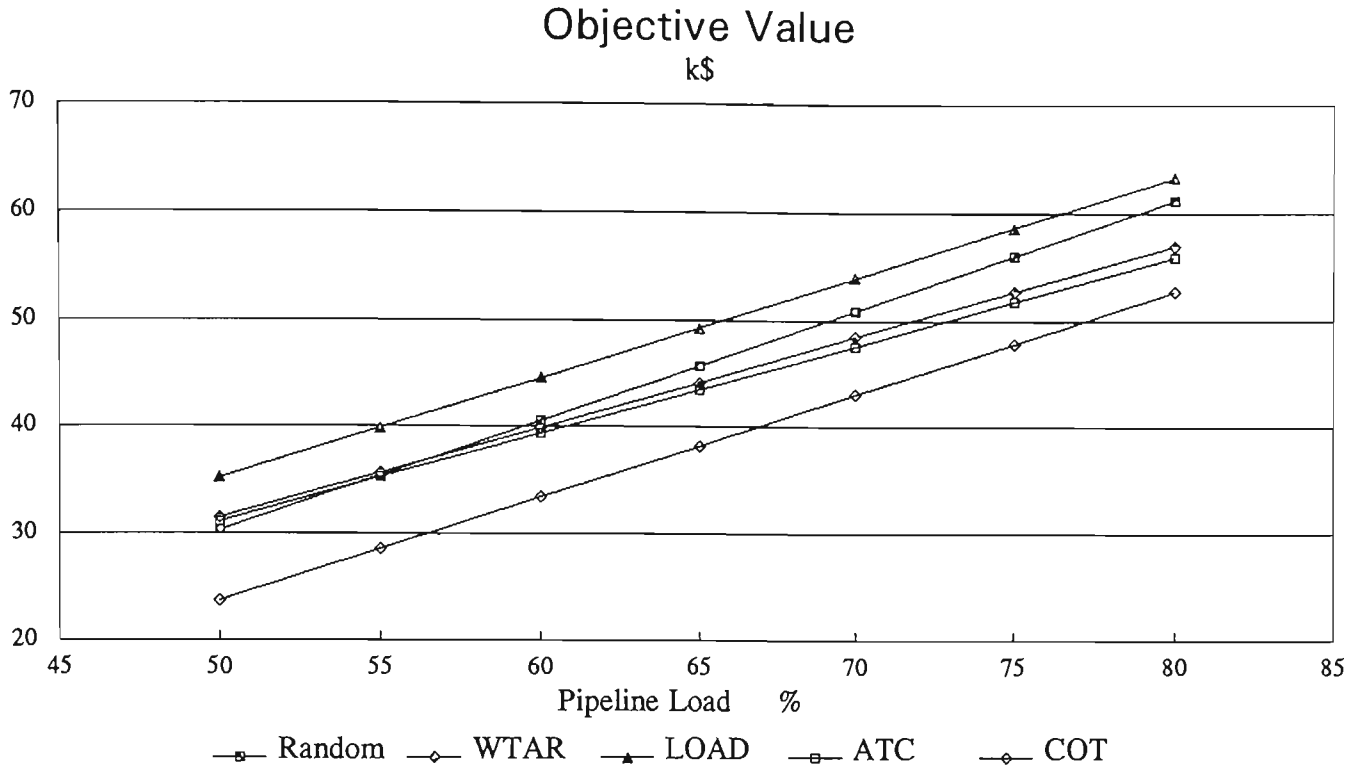


Figure 5.9.7 Performance based on pattern utilization

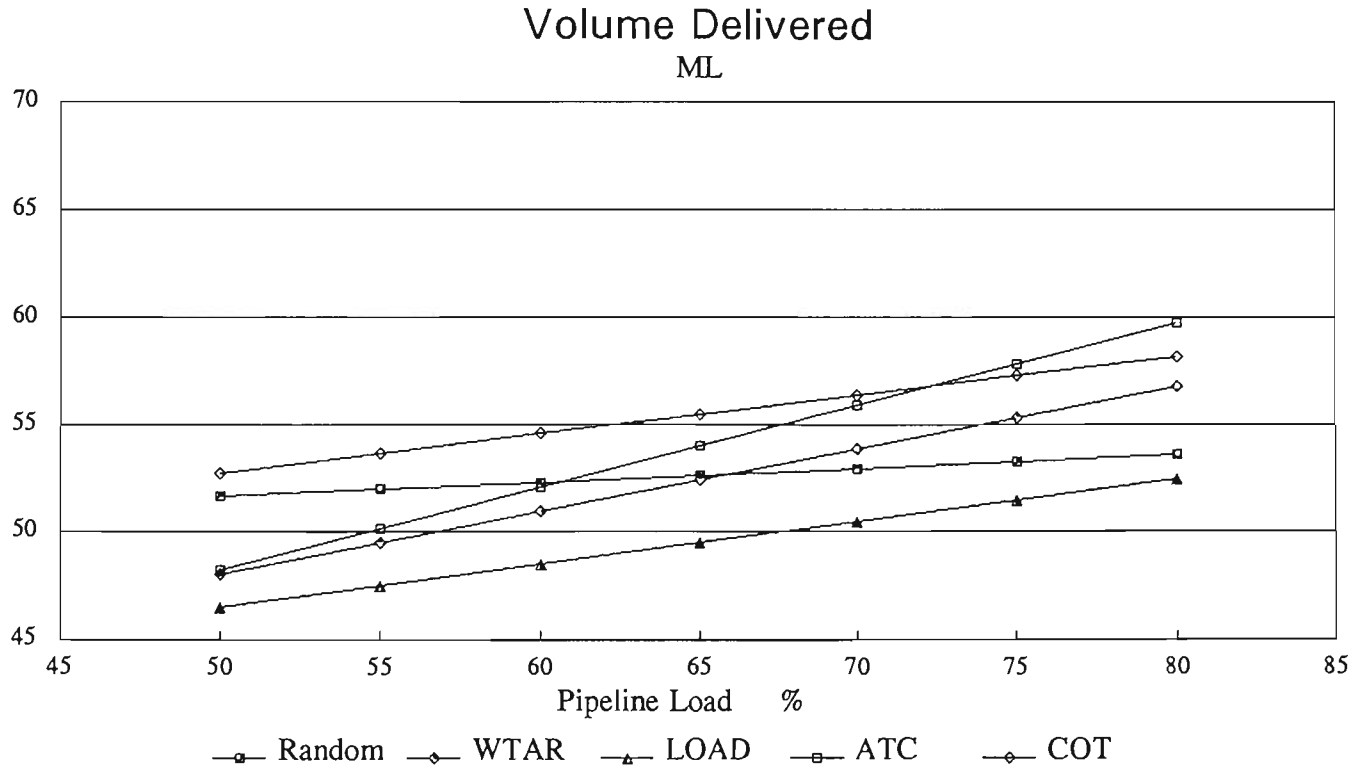


Figure 5.9.8 Performance based on pattern utilization

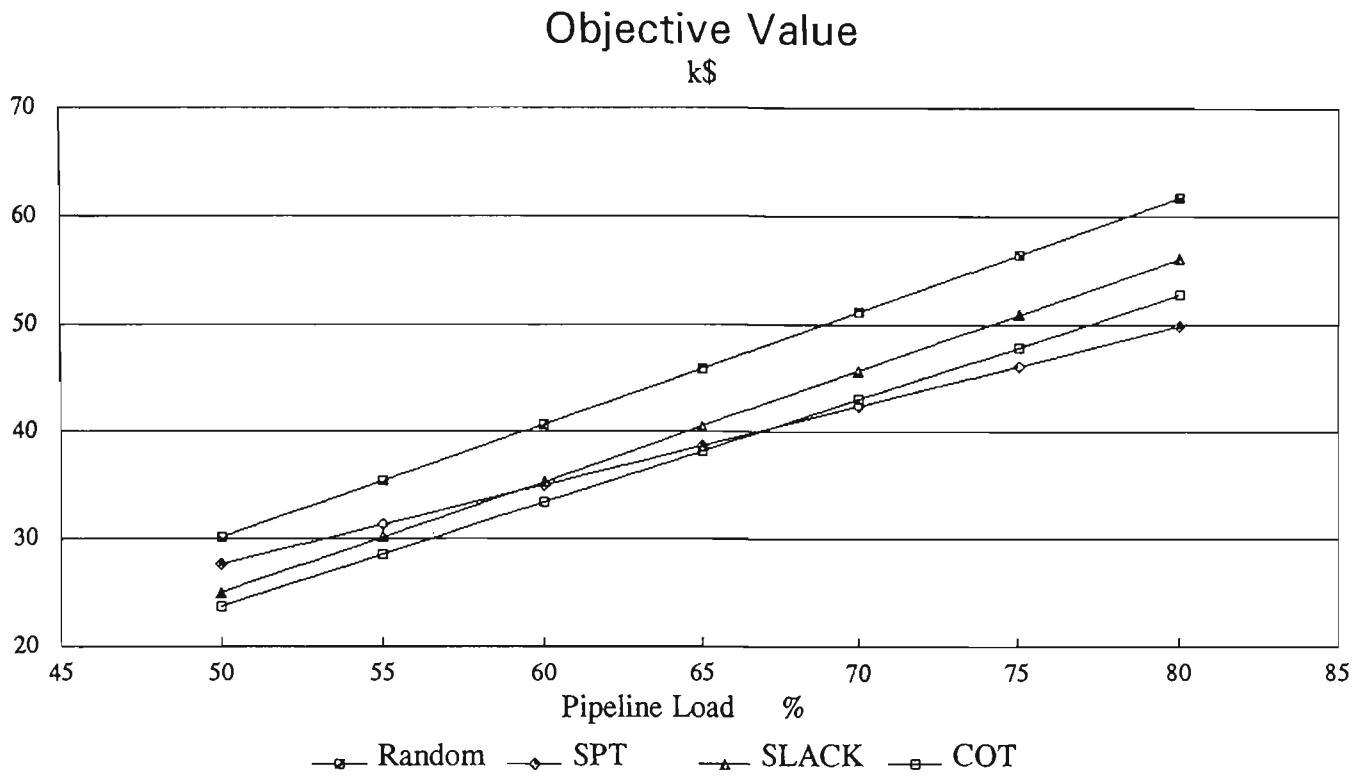


Figure 5.9.9 Random and SPT based on utilization, SLACK and COT on priority

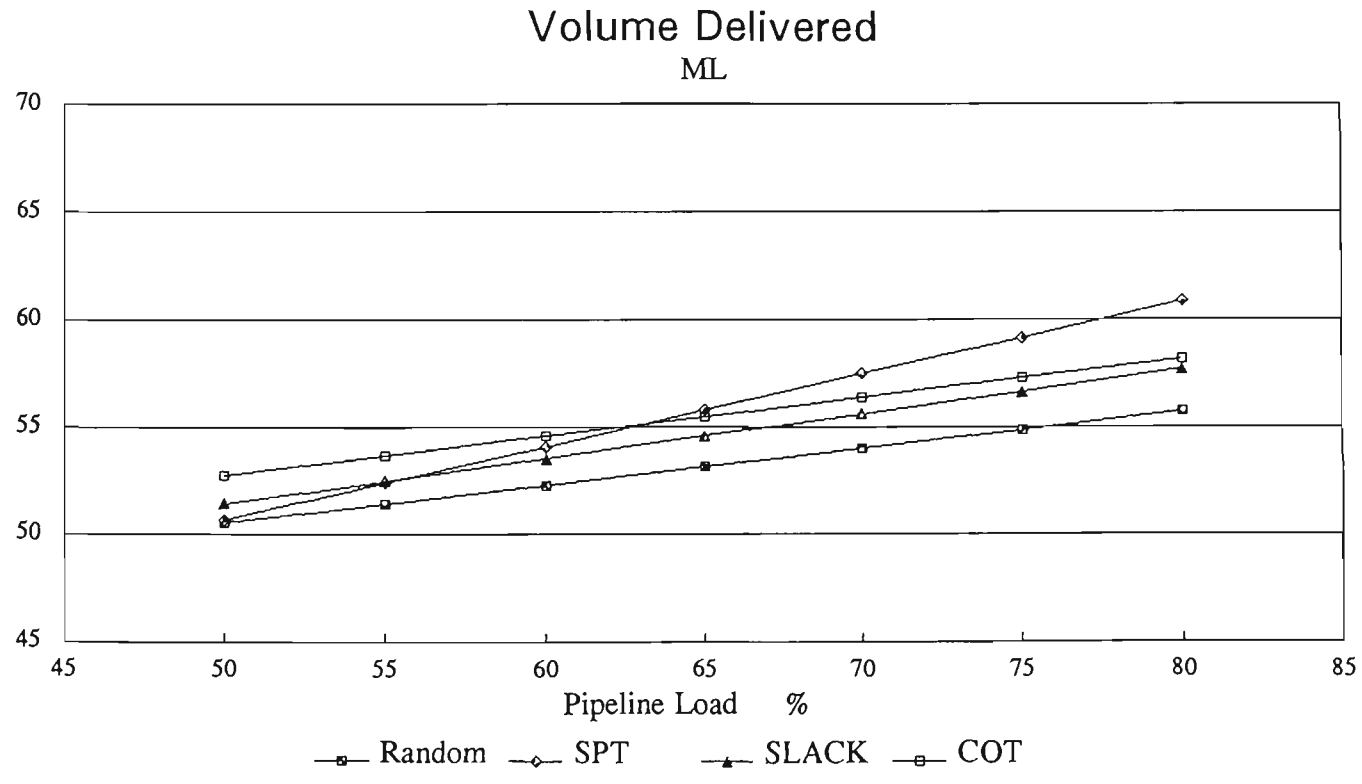


Figure 5.9.10 Random and SPT based on utilization, SLACK and COT on priority

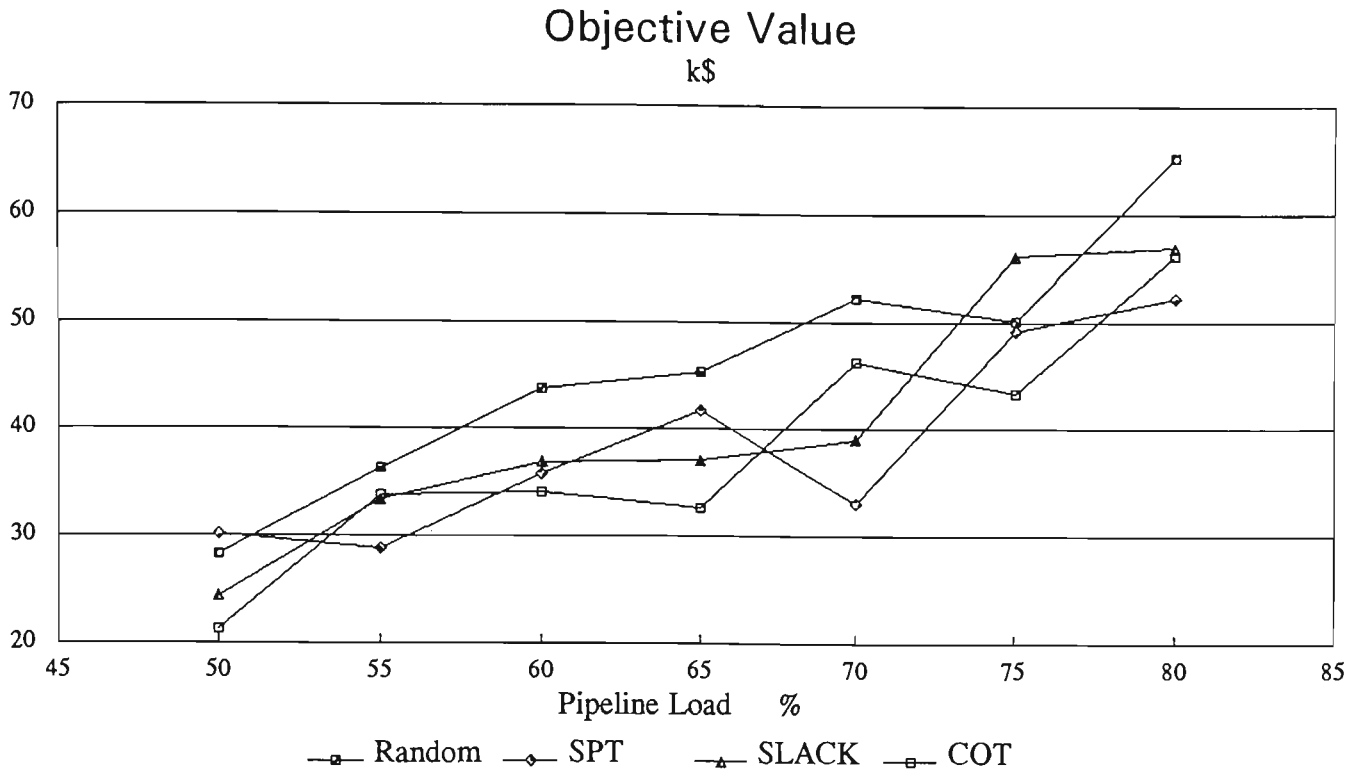


Figure 5.9.11 Random and SPT based on utilization, SLACK and COT on priority

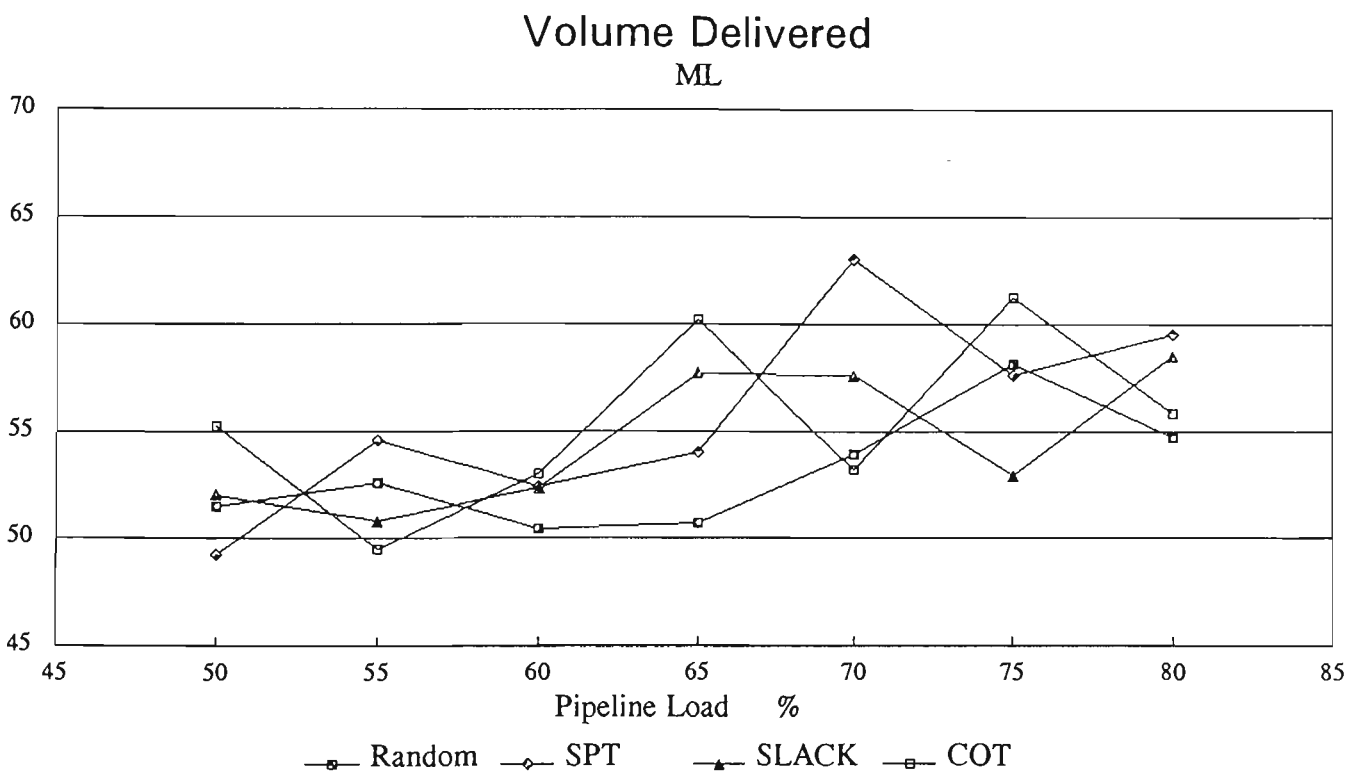


Figure 5.9.12 Random and SPT based on utilization, SLACK and COT on priority

Chapter 6 DISCUSSION

6.1 Evaluation Methodology

Chapter 5 presented the results of instance evaluation and simulation studies, which were considered to be the most appropriate means of evaluating algorithmic performance. With the benefit of these experimental results it is now appropriate to evaluate the suitability of these assessment techniques. Their suitability, however, is linked to the evaluation criteria upon which the performance assessment is based, and this will be examined first.

6.1.1 Evaluation Criteria

These are the key performance indices used to measure solution quality and ease of use. Two parameters are considered paramount:

1. Work completed: the amount or quantity of work achieved.
2. Weighted tardiness: the quality of the work achieved.

Of these, only the second parameter was explicitly considered in the objective function of the experiments conducted. The facility to measure the work completed (i.e. volume delivered) was provided, but not used, in this study. However, the work completed was accounted for implicitly, because schedules with low work completed have high tardiness. Tardiness is reduced by completing more work in the time provided (and also by rescheduling the order of this work).

From the refinery's perspective, the primary criterion (ignoring customer service implications) is the amount of work achieved. The customers on the other hand (i.e. the offtaker) would consider the tardiness to be of paramount importance. This applies regardless of whether they receive or do not receive their product; product which does not arrive would simply be considered to be very tardy. Hence the focus of this series of experiments was the quality of the solutions, and the objective function was formulated accordingly.

6.1.2 Instance Evaluation

Relevance of Test Cases

The five test cases were selected at random. Two of these could be considered atypical: one of the pipelines was unavailable for all and part of the scheduling horizon respectively. These two cases do, however, represent conditions that are encountered periodically as a result of system maintenance, and the algorithm must have the capability to handle this situation.

The results of the instance evaluation and simulation experiments demonstrate the wide variability in performance with changing conditions (such as machine load). This raises an obvious limitation with the instance evaluation method: the nature of the method limits the number of cases which can reasonably be considered, and hence the conclusions drawn must recognize the narrow observational base. The validity of these conclusions cannot necessarily be extended beyond the instances evaluated.

Evaluation Method

Because each of the instances considered contained a different number and type of jobs direct comparison of results between instances was not possible. Instead, the results were expressed as a ratio of the best performance for that particular instance. A similar approach was used by Kozan [50], except the results were reported as percentage difference with respect to the best performance. The ratio parameter was adopted in this study because of the magnitude of the relative performances.

The average performance was weighted for each of the five instances. That is, the raw data were averaged, and the performance ratios calculated, as opposed to directly averaging the performance ratio of each instance.

6.1.3 Simulation

Experimental Conditions

Considerable effort was directed towards providing realistic simulation conditions. In many areas this was achieved, such as offtaker requirements and product mix. However, due to complexity of actual situations some areas were only able to be approximated. These are discussed as follows.

1. Job Definition

The clustering of jobs drawn from common product batches was adequately modelled. However, in actual situations volume allocation from product batches recognizes offtaker availability. Hence if a particular offtaker will not be available within a reasonable time of a batch being released (i.e. ready to send) no product will be allocated to that

offtaker. Product is usually only allocated to an offtaker when there is a reasonable expectation of the offtaker being able to accept the product before the due time. This was not able to be modelled in the simulation. Hence product was often released, and due time limits set, when offtakers were unavailable. Tardiness in these situations was frequently unavoidable, contributing to the high tardiness levels observed.

2. Offtaker Availability

As noted during the instance evaluation experiments, jobs were frequently scheduled during periods in which the offtakers would normally be considered to be unavailable. The author believes, through his own experience and anecdotal evidence, that offtaker availability windows are commonly extended by mutual agreement in order to meet schedule requirements. This flexibility was not modelled in the simulation experiments and could well make comparisons with manual solutions difficult. As noted, only normal availability windows with standard overtime extensions were used. This contributed to the high tardiness levels and number of uncompleted jobs observed.

Sample Size

The sample size in all experiments was limited to 40 cases. This limit was based on theoretical and practical considerations. First, for sample sizes greater than 30 the sample variance usually provides a good estimate of the population variance (Walpole and Myers [71], p174). From a practical perspective, simulation run times increased in direct proportion to sample size. The significant number of runs conducted (in excess of 200) meant that the duration of the experimental program would have been unacceptably extended if samples sizes were too large. A

balance was required between results and effort. From a purely pragmatic position, a sample size of 40 met the sampling requirements and was able to be reported on a single page, minimizing printing and data handling.

Machine Load

Experimental runs were conducted at various machine loads to assess the variation in performance of specific selection rules under differing load conditions. The variability of performance with load is an important algorithmic consideration, as has been reported by Vepsalainen and Morton [70]. The range of machine loads was selected to span conditions reasonably expected to be encountered in practice. "Average" and "Nominal Capacity" conditions shown in Appendix B indicated machine load levels of 66 % and 73 % respectively. Conditions during periods of reduced refinery throughput or short term maximum processing could be expected to result in loads within the 50 % to 80 % range selected.

Data Smoothing

The scatter in performance as a function of load level was expected to be less than that observed in the experiments, given the sample size evaluated. Least squares linear regression was employed to smooth the data and aid analysis. The averages of the weighted tardiness and volume delivered for each run (average of sample size of 40) were processed in this manner. The regression line for each parameter from each run was plotted with the observed data to assess goodness of fit and overall performance.

Without this data smoothing evaluation of the relative and absolute performance of the selection rules would have been extremely difficult.

It would have been unreasonable to base conclusions upon the performance at one machine load when the variation in performance with load was subject to considerable scatter and a wide range of machine loads are encountered in practice.

Pseudo-Random Number Generation

As discussed in Chapter 5, the simulation algorithm was structured to provide the same sequence of random numbers for each experimental run. Hence the same number and type of jobs were created for each case. This control was achieved for all job selection rules except RANDOM, because the RANDOM rule also made calls to the generator subroutine. This resulted in a different random number sequence being used and hence the number and type of jobs was slightly different. When averaged over the 40 cases in each experiment the effect was small, as observed by parameters such as number of jobs and total job volume. Hence this was not considered to have significantly biased the results and should not affect the validity of the conclusions. These differences are summarized below.

Table 6.1 Sample Variation: RANDOM rule versus others

Machine Load	Total Jobs			Total Volume		
	Random	Others	Delta	Random	Others	Delta
%	-	-	%	ML	ML	%
50	26.9	27.0	(0.4)	66.9	67.6	(0.1)
55	29.8	30.0	(0.7)	73.2	73.7	(0.7)
60	31.5	31.2	1.0	79.2	78.4	1.0
65	34.9	33.8	3.3	85.7	85.3	0.5
70	36.8	36.4	1.1	92.4	91.9	0.5
75	39.6	40.4	(2.0)	97.7	97.9	(0.2)
80	42.0	42.0	0.0	104.4	104.6	(0.2)
Total	241.5	240.8	0.3	599.5	599.4	0.0

Note: Random results shown for pattern priority selection rule.

6.2 Performance Evaluation

A brief summary of the results from each experiment is included in Chapter 5. The following section discusses the relevance of these results and analyzes the observed trends.

The variability encountered in the results indicates the risk associated with drawing conclusions based only upon the analysis of a handful of instances. For this reason the main focus of this discussion will be directed towards assessing the results of the simulation experiments, while recognizing the limitations of the method discussed above. Section 6.2.1 discusses the relative performance with pattern selection based solely on pattern generation order (i.e. pattern priority). Section 6.2.2 discusses the effect of pattern utilization on the performance of each of the decision rules.

6.2.1 Job Ranking

Cost Over Time (COT)

All conditions being equal, the Cost Over Time (COVERT, or COT for short) rule provided the best overall performance. As discussed in Vepsalainen and Morton [70] the basic COVERT rule is a dynamic rule for average weighted tardiness scheduling which incorporates job weights into a slack-based approach. The COVERT priority index represents the expected tardiness cost per unit of imminent processing time (hence Cost OVER Time). In effect, if the slack exceeds some generous "worst case" estimate of waiting time, the expected tardiness cost is set to zero. They note that previous simulation studies have found it convenient to estimate the waiting time of an operation as a multiple of the processing time. They note that this multiple, the lead time estimation parameter, should reflect the anticipated machine utilization.

The simulation results of the previous chapter tend to support this last observation. Analysis of the results indicates that the best overall performance, expressed as total weighted tardiness or volume delivered (i.e. work done) (Figures 5.7.1 and 5.7.2) is achieved with a waiting time parameter value of 3 at low machine loads, and with a value of 4 or 5 at high machine loads. Vepsalainen and Morton [70] used a value of 4 as the basis for their evaluation, which was conducted at higher loads than those evaluated in Chapter 5.

Apparent Tardiness Cost (ATC)

Vepsalainen and Morton [70] concluded that the ATC rule was superior to competing rules for minimizing weighted tardiness penalties in all load conditions studies. This was not borne out directly in the results of

Chapter 5, which are summarized in Figures 5.8.3 and 5.8.4. The explanation appears to lie in the range of load conditions in the separate studies. Apart from the many other conditions which differ, Vepsalainen and Morton evaluated performance at loads from 80 to 97 %. The simulation experiments of Chapter 5 used load levels from 50 to 80 %. Figure 5.8.4 indicates that it is only at the upper end of the load range that the ATC rule dominates the other rules. Although the weighted tardiness at these loads is higher than the best rule (COVERT) the relative rate of increase of tardiness of the ATC rule is lower. If the trend were to continue the ATC rule would result in lower tardiness than all other rules at higher loads.

As discussed in Chapter 2, the ATC rule adjusts the average processing time of the waiting jobs by a "look-ahead" parameter related to the number of competing critical and near-critical jobs. This is used to determine a standard reference against which each job is compared. They used a scaling factor of 2 in static flow shops and 3 as a reasonable average in dynamic job shops. In the scheduling system studied in Chapter 5, a value of 3 gave the best performance with regard to total weighted tardiness and volume delivered at all machine load levels (Figures 5.6.1 and 5.6.2). Performance at 4 was second best, and well ahead of all other values, particularly at high load levels. The performance achieved at all load levels for k=1, 2, 5 and 6 was the same, even for the raw data (Figures E.6.1 and E.6.2 in Appendix E).

Random (RANDOM)

With respect to volume delivered the RANDOM rule performed well at low load levels but very poorly at high loads. As shown in Figure 5.8.1.2 the volume delivered was relatively insensitive to machine load.

Weighted tardiness, however, was relatively poor at all load levels. At low loads this was because no effort was made to schedule jobs to minimize tardiness. This effect was exacerbated at high loads because of the amount of unfinished work, due to the minimal increase in volume delivered. The only rule to perform consistently worse than RANDOM was the LOAD rule, in both weighted tardiness and volume delivered.

First Come First Served (FCFS)

The FCFS rule performed only marginally better than the RANDOM rule overall. At low machine loads the objective value was actually poorer than RANDOM (although not as poor as LOAD), but improved relative to RANDOM as machine load increased. The volume delivered was much poorer than RANDOM at low machine loads but better at high loads. The raw data based on pattern priority exhibited less scatter than most other rules as a function of machine load (Figures 5.8.2.1 and 5.8.2.2).

Shortest Processing Time (SPT)

At low machine loads the volume delivered by the SPT rule was very poor, and only the LOAD rule performed worse. At high loads the performance was better, and similar to the FCFS rule. The corresponding trend was observed with weighted tardiness. These observations are consistent with Vepsalainen and Morton [70] who noted that the SPT rule "fails with light load levels and generous due date allowances".

The data trend with respect to machine load was reasonably consistent, except for very poor volume delivered at a machine load of 70 % (Figure 5.8.3.2). Surprisingly the objective value at this level, although poor, did not deviate from the trend as much as the volume delivered.

Earliest Due Date (EDD)

The overall performance of the EDD rule was quite good. At low machine loads the volume delivered was poor, but good performance was observed as the loading increased. The objective value was lower than most other rules regardless of machine load level. Moderate scatter was observed in the raw data at most load levels. However, the volume delivered at each end of the load range was much better than that achieved in the interval (Figure 5.8.4.2).

Minimum Slack (SLACK)

The SLACK rule performed well at all load levels, although some scatter was observed in the raw data (Figures 5.8.5.1 and 5.8.5.2). Only the COT rule performed consistently better. The SLACK rule was expected to performed reasonably well because it incorporated tardiness and processing time information in the ranking index, and the results validated this expectation.

Weighted Tardiness (WTAR)

The overall performance, both in terms of objective value and due date, was moderate. This rule was initially expected to perform well because it incorporates the objective function parameter directly in the job selection index. However, this rule has no "look ahead" capability; the tardiness is not predicted, but registered after it has occurred. The information thus arrives too late for the algorithm to act effectively. Instead it must wait until a job becomes tardy before it has a basis for ranking it with respect to the competing jobs.

Load Levelling (LOAD)

The performance of this rule was disappointing. Initial expectations were that this rule would be very effective because it took machine loading directly into account. The aim was to allocate jobs to those time periods where minimal competition existed for selection, thereby avoiding instances of machines being unutilized by previous imprudent allocation. This rule was intended to directly recognize the time windows imposed by offtaker availability and make the best use of available resources. Hence the work completed was expected to be high. Weighted tardiness, on the otherhand, was not expected necessarily to be good, due to the lack of inclusion of due time information in the job ranking. It was, however, expected to be affected favourably by the (expected) high volume delivered.

The simulation results, however, were to the contrary. Both volume delivered and total weighted tardiness were poor. Results were inconsistent: at 65 % machine load, the performance of the load levelling rule was good. Results at higher machine loads were very poor (these were improved by the introduction of utilization into the pattern selection criteria). Further work is required to understand this behaviour and to improve the effectiveness of the method which has fundamental potential.

6.2.2 Pattern Ranking

As summarized in Chapter 5 the effect of pattern utilization on smoothed performance was inconsistent. The ATC and COT rules were unfavourably affected at most load levels, the FCFS was largely unchanged, the performance of the RANDOM, EDD, SLACK and WTAR rules was slightly

unfavourable at low loads, but was marginally favourable at high loads, and the SPT and LOAD rules were favourably affected at all pipeline loads.

Within a given scheduling rule, the effect of pattern utilization under varying load conditions was generally inconsistent. This behaviour was observed with all rules except the SPT (the result at 50 % load was effectively unchanged). For example, the performance improvement in the smoothed data for the LOAD rule was apparent although the raw data oscillated, while in the WTAR rule the effect was favourable at high load levels and inconsistent at low loads.

The major improvement in performance was achieved with the SPT rule. This effect elevated the SPT rule from marginal performance, based solely on pattern priority, to the best performance at high load levels.

6.2.3 Backtracking

Performance Improvement

Overall performance improvement as a result of backtracking was relatively modest, averaging 3 % over all rules. Individual results are shown in Table 6.2. The improvement in performance is defined as the difference between the objective value of the first solution pass (First Value) and the best solution pass (Best Value), expressed as a percentage of the First Value. The values shown in Table 6.2 are the sum of the individual objective values at each machine load level. No data were collected to measure the improvement in the volume delivered as a result of backtracking.

The small increase in performance may be due to two main reasons. First, the performance of the algorithm may be such that the quality of the first solution obtained may be good, leaving little room for improvement. Second, the performance of the backtracking routine may be poor, such that improvement is unlikely. A combination of these two factors may also exist.

Table 6.2 Performance Improvement Due to Backtracking

Job Rule	Pattern Priority			Pattern Utilization		
	First Value k$	Best Value k$	Delta %	First Value k$	Best Value k$	Delta %
RANDOM	331.0	319.9	3.4	332.8	321.3	3.5
FCFS	334.9	324.2	3.2	331.1	321.6	2.9
SPT	310.5	305.4	1.6	277.7	271.2	2.3
EDD	293.4	287.1	2.1	297.1	291.4	1.9
SLACK	291.9	283.8	2.8	295.6	287.9	2.6
WTAR	316.9	309.4	2.4	314.6	306.2	2.7
ATC (k=3)	313.4	304.5	2.8	330.2	322.3	2.4
COT (k*b=3)	275.3	267.6	2.8	286.3	281.5	1.7
LOAD	358.2	344.8	3.7	346.9	334.7	3.5
Average	313.9	305.2	2.8	312.5	304.2	2.6

Backtracking Passes

The backtracking method is discussed in Chapter 4. In the experiments of Chapter 5 the maximum number of backtracking passes was limited to 50. This limit was rarely reached, and on average 17 passes were evaluated before termination. The number of passes was relatively insensitive to the scheduling rule or pattern ranking rule used. The data in Appendix E exhibits a slight decrease in the number of patterns evaluated with increasing machine loading. The average number of passes for each scheduling rule, for all load levels, is summarized in Table 6.3.

Table 6.3 Average Number of Backtracking Passes

Job Rule	Pattern Priority	Pattern Utilization
RANDOM	18.3	17.5
FCFS	18.8	16.8
SPT	14.2	15.9
EDD	15.5	14.5
SLACK	18.0	18.2
WTAR	15.4	16.5
ATC (k=3)	17.0	15.3
COT (k*b=3)	16.2	14.6
LOAD	19.7	18.9
Average	17.0	16.5

Termination

Termination of the algorithm was forced when either no backup points remained to be evaluated or a specified maximum number of passes had occurred without improving the objective function. This specified maximum value was set at 10 passes for the experiments reported in Chapter 5.

The reason for termination was not recorded for each simulation case. If due to lack of improvement in the objective function this would imply that, on average, solution improvement ceased after about 7 passes (17-10). This would suggest that the backtracking routine should be examined to determine why more opportunities are not available for improvement. This could be due a number of reasons, such as:

1. Lack of patterns at each backup point, due to lack of jobs or due to the pattern generation process itself. The number of patterns generated at any point was subject to an upper limit of 10. In addition, for simplicity the pattern generation process only a

generated a limited sequence of pattern combinations at each point in the schedule.

2. Lack of backup points, due to the backup point identification process. This was restricted to only those points at which a pattern was scheduled in which a job, eligible to be scheduled, was precluded from being scheduled, due to contention with a higher priority job.

Potential modifications to the algorithm to address these issues are discussed in Section 6.4.

6.2.4 Comparison With Manual Method

As discussed in Chapter 5, comparison of the performance of the algorithm with the solution prepared by the manual method is difficult for past instances because of lack of data. The author has not had the opportunity to compare results based on current manual methods, in which all the data could be obtained, including precise information on availability windows, due to remoteness from the worksite. However, a number of comparisons are evident.

Solution Speed

First, the computer based algorithm can generate a schedule much more rapidly. Given that the same amount of data is required by both computer and manual methods, the computer method can generate a schedule from this information in a few seconds. Generation by the manual method takes tens of minutes, if not hours in difficult circumstances.

Although the same amount of information is required by the two methods, the way this is generated and used is quite different. The computer method requires all the information to be available at the start of the run. Obviously the program can be rerun if some of the data changes at a later stage. In the manual method, the data is typically only generated as required. That is, the scheduler makes an assessment of the jobs to be sourced from a single product batch, those jobs are scheduled, and then the next batch is allocated.

The scheduler recycles to the earlier batch if an inconsistency or poor performance becomes apparent later. This approach occurs because it is difficult to accurately specify the size and timing of jobs far in advance, and such specification belies the variability possible in meeting customer demands in the short term. Hence the computer based method is forced to work with "harder" data than the "soft" data manipulated by the user. The user of the computer method is able to compensate for this, however, due to the rapid generation of the schedule provided by the computer method. A case can be defined, run, evaluated, and redefined quickly until a suitable solution is found. This has not been trialled in practice, but is will obviously be faster than the manual method.

Solution Quality

This is difficult to measure. However, based on the range of instances evaluated earlier the computer based method is somewhat poorer than the manual method. Part of this is due to the incompleteness of the information used to construct the computer runs. Part is perhaps due to the performance variability of the algorithm as observed in the simulation experiments. Solution quality could probably be improved by

continued fine tuning for each specific instance. This enhancement was not used; the results reported in Chapter 5 are those arising from single untuned runs.

The important issue is not whether the computer method produces solutions better or equal to the manual method, but whether it produces acceptable solutions, recognizing the time / quality trade-off. This evaluation has not been attempted yet because it would require use of the computer method in the practical scheduling task for some period, which is not currently possible.

The advantage of the computer method is that its speed allows it to evaluate a wider range of solutions than is possible otherwise. As the sophistication of the algorithm improves with continued development the solution quality should increase to meet or exceed the expectations of the users.

Solution Flexibility

Clearly, the manual method is superior, limited only by the imagination and experience of the scheduler concerned. The scheduler can balance qualitative concerns in addition to the quantitative measures to which the algorithm is constrained, such as balancing a customer's reluctance to work overtime with that customers need for product. Rutherford [59] noted that the "human being is incomparably superior to the computer at acquiring and using experience, adapting to new and unusual events, recognizing patterns, and applying common sense". The function of the computer method, however, is not to attempt to replace the role of the scheduler, but to provide an aid to assist the ability of the scheduler

to make better decisions. In this role its current performance is considered acceptable.

6.2.5 Ease of Use

Clearly a rigorous assessment of this area awaits the use of the program by users other than the author. However, favourable opinion is expected due to the simplicity of the program operation and data input requirements, and the reporting options available.

Operation

Mainframe operation is described in detail in Appendix F. This is very simple due to the availability of a customized file management and program execution system on the company's computer. Jobs are run on-line and execute in seconds. PC operation has not yet been explored, due to the ready access to the mainframe. Performance is expected to be good due to the computational power of the current range of personal computers in use on site.

Reports

The reports currently available for solution analysis are discussed in Appendix 5. These have been designed to meet the needs of the scheduler. In particular, the main solution report has been designed to present the schedule in Gantt chart format, similar to the current manual method. This graphical layout allows the user to quickly evaluate the schedule, in a form consistent with that refined over many years experience.

6.2.5 Changing Environment

A number of changes which directly affect the scheduling system have occurred in the business environment since this project commenced. First, continuing rationalization in the Australian oil industry has lead to a reduction in the number of offtakers. This has lead to a reduction in the number of disparate priorities requiring satisfaction. At the same time the average offtaker availability has increased because the remaining offtakers work longer hours to maximize productivity from existing facilities. Second, pipeline capacity has been increased, with additional spare capacity to cater for future grows. Third, the product mix has altered due to increasing penetration of unleaded gasoline into the market, reducing the dominance of leaded gasoline as the major product.

These changes have, if anything, reduced the overall complexity of the task required of this scheduling algorithm. Hence the performance in practice would be expected to be at least as good as that observed with the experiments evaluated in Chapter 5. Most importantly, however, these changes highlight the significant shifts in requirements that can occur, and demonstrate the need for a flexible and robust algorithm.

6.3 Algorithm Structure

The performance of the algorithm as a function of the job selection and pattern ranking rules has already been discussed. This focus of this section is directed towards analyzing the overall structure of the algorithm itself, to understand its fundamental strengths and

weaknesses. From this understanding, development efforts can be directed towards improving the algorithm's performance.

6.3.1 Solution Strategy

The fundamental principle of the solution strategy was to ensure that no technological constraints were ever violated, and hence the partial schedule at any point was always feasible. This was a considered approach to the difficulties created by the interdependence of the machines. It was felt that it would be extremely difficult to remove any infeasibilities created by solving a relaxed solution once imbedded in the schedule. It was very important from the user's point of view that any schedule created would work if implemented: lack of credibility would lead to disuse due to loss of confidence.

This requirement for feasibility was a major reason for constructing the schedule from the starting time, rather than at some other point, such as the end, or when some high priority job became available. The primary reason for starting at the front, however, was because the definition of the early jobs is much better than the later jobs due to the uncertainties associated with projections of product, machine and offtaker availability. As a result of this uncertainty the schedule will almost certainly be revised before later jobs can be scheduled. The schedule created is part of a larger, rolling schedule, which is never completed, but simply revised to take account of changing needs.

6.3.2 Job Selection and Ranking

The job selection process is a necessary, but not sufficient condition, to ensure that only feasible partial schedules are created. It rejects jobs which do not meet any feasibility criteria at each stage, and also halts those jobs judged unsuitable for further consideration. This later task is a quantitative assessment of what are largely qualitative criteria and hence the algorithm makes used of parameterized rules to provide user control.

The job ranking process has been well discussed in previous sections. The primary objective is to order the jobs in such a way that the overall objective function is minimized (or maximized, depending on its formulation). Although a second (pattern) ranking process occurs prior to construction of the next stage of the partial schedule, the job order is fundamental to the order in which the jobs are allocated to patterns, and hence ultimately to the schedule.

6.3.3 Pattern Generation and Ranking

Pattern generation is used to ensure only feasible partial schedules are created. The need for this process arises due to the interdependence of the main pipelines, created by the suction and offtaker pipeline interconnections. These interconnections create either hydraulic constraints, in which two (or more) flows physically cannot occur simultaneously, or quality constraints, in which two (or more) flows cannot be mixed. Any pattern passing through the pattern generation phase can be added to the partial schedule without violating any technological constraint.

The pattern ranking process sorts the feasible patterns and hence controls the jobs added to the partial schedule at the current stage. This also controls the order in which any later schedule evaluation is undertaken from that stage if backtracking occurs.

Pattern ranking is a feature of the cutting stock approach which was used as a motivation for this work. However, a difference exists between the pipeline scheduling and cutting stock situations. In the latter the aim is usually to select the pattern which results in the longest run length. This minimizes cutter setups and improves machine utilization. In the former, due date constraints and job priorities impact on performance. Giving preference to patterns with long run lengths discourages the simultaneous processing of jobs on more than one machine, reducing the flexibility inherent in the system. Hence, patterns are ranked by some other index, such as average machine utilization, or generation order.

Generating a number of patterns at this stage has advantages which offset the additional computation required. First, the patterns can be ranked and the "best" pattern selected at that stage of the partial schedule. Second the patterns can be ordered to aid the backtracking process if returned to that stage. The pattern ranking process provides a mechanism to include heuristic information in the solution process that cannot be achieved by only generating the patterns when required.

6.3.4 Backtracking

As discussed in Chapter 4 a depth-first strategy is adopted to quickly establish a solution (in which the end time is reached, or all jobs are scheduled). This bound is then used in the subsequent backtracking phase to halt evaluation if the new partial schedule is worse than the previous best solution (lower bound pruning), and is updated when a better solution is found.

The depth-first strategy was adopted because it rapidly provides a feasible solution (feasible in the sense that all technological criteria are met, even in some jobs remain incomplete. It also reduced the data storage requirements needed to successfully implement a backtracking strategy for solution improvement (a breadth-first implementation was considered too unwieldy for this application). Heuristic information is included in the node evaluation step to determine the order in which the successor nodes should be expanded.

The tree search implementation ensures that solution branches are not precluded from evaluation. Hence the algorithm backs up to the most recently backup point. This is the most recently suspended node judged worthy of further evaluation, due to some identified contention between jobs. It then proceeds to attempt to improve the schedule, identifying additional backup points in the re-evaluated portion of the schedule if appropriate. Once the end of the schedule is identified, or some lower bound exceeded, the algorithm returns to the most recently identified backup point and continues.

This overall strategy is considered suitable for the task required. The main reason for the small increase in solution quality resulting from the backtracking phases is considered to be due to the choice of backup points identified, as discussed in Section 6.2.

6.4 Further Work

Whilst the algorithm in its current form provides a practical scheduling tool, development in a number of areas could be expected to improve solution quality and useability further.

Performance Enhancement

The series of experiments conducted in Chapter 5 investigated the performance of various decision rules thoroughly, but over a limited set of conditions. Further work remains to evaluate the effect of varying the tuning parameters under the user's control. These include:

1. Job selection parameters, which control the jobs considered for scheduling at each stage, such as minimum processing time requirement, minimum offtaker availability, job work remaining, etc.

2. Objective function weighting, to assess the effect of volume delivered, setup count and offtaker overtime, on overall performance.

3. Pattern priority weighting, to assess the effect of pattern duration and average job value on performance.

4. Dynamic adjustment by the algorithm of the ATC or COVERT scheduling rule parameters to recognize the load level of the problem being solved.

Scheduling Rule Modification

Modification of certain rules, such as the inclusion of a look ahead feature to detect job criticality in SLACK and WTAR, could reasonably be expected to improve performance. The introduction of a mechanism to ensure that deadlines are never violated warrants attention.

The ability to deliberately idle a machine, in readiness for the anticipated arrival of a high priority job, requires investigation. The corresponding reduction in tardiness achieved on the higher priority job needs to be balanced with the lower machine utilization, and hence loss of capacity to do work.

A further enhancement could include the solution of the problem using a number of scheduling rules in succession, in which the algorithm would automatically select the best overall solution. This approach is currently gaining favour amongst practitioners [23,72]. The current fast solution times suggest that the increased computation associated with this approach should not be of concern.

The performance of the LOAD rule could possibly be improved by adopting the approach of Keng et al [49]. They first ranked the operations according to their "criticality": i.e. the ratio of processing time remaining to demand window availability (the demand window being the difference in time between the due time and the greater of the current time or job ready time). The most critical operation was then scheduled

according to the "cruciality" of the machine: the period in which there was least contention with the other jobs requiring this machine (i.e. when the potential load on the machine was lowest). This enables tardiness information to be used effectively in ranking the jobs, and at the same time allocates machine resources appropriately. The difficulty remaining to be overcome in the problem at hand is how to ensure that the feasibility of the pattern is maintained at each decision stage.

Backtracking Modification

A number of issues regarding the backtracking mechanism were raised in Sections 6.2 and 6.3. Evaluation of the improvement in performance as a function of the existing control parameters is required. The greatest benefit is expected to come from careful analysis of the appropriate points in the schedule at which the backtracking should occur. This could be extended from the current contention based approach to include points at which deliberate idling occurred, or where a significant change in the objective function was detected.

Restart Capability

Frequently when a schedule is revised the main objective is to minimize the disruption to the remainder of the schedule not requiring revision. This minimizes the disruption to the expectations, plans and labour requirements of the offtakers. In its current form, the algorithm cannot distinguish between old and new jobs (the new jobs being the ones that must be rescheduled). Hence, some mechanism needs to be introduced to identify those jobs which should not be rescheduled unless necessary.

One proposal is to arrange for the program to retain the previous solution to act as a starting point for the revision. The user can then

identify those jobs which require revision and, if desired, identify those jobs which may or may not be rescheduled to achieve a new solution.

PC Capability

This extension is relatively simple and can be conducted if the need exists. When this study was initiated it was considered that given that a significant part of the scheduling task is conducted out of normal business hours PC capability would provide the ability for the scheduler to operate remotely. Technological and business changes since that time have resulted in the connection of many remote PCs to the company mainframe by modem. Hence mainframe access can be made available at the scheduler's home and the need for operation on a PC is reduced. However, to provide maximum flexibility for the scheduler installation on a portable PC is considered desirable in the longer term. With this is mind the application has been written in standard FORTRAN 77 and with minimal computational requirements to enable suitable operation on a personal computer when needed.

Chapter 7 CONCLUSIONS

The conclusions drawn from the experimental work can be classified into three sections: first, the suitability of the heuristic approach as a solution method; second, the relevance of instance evaluation and simulation as techniques for performance evaluation; and third, the choice of job scheduling rule.

Suitability of Heuristic Methodology

* Of the solution methodologies reviewed in Chapter 2, the heuristic approach appeared to offer the best practical approach to the solution of the problem at hand.

* The algorithm successfully applied this method to the constraints and objectives of the pipeline scheduling task.

* The experimental work demonstrated the solution times and solution quality required for this approach to be of practical use.

* Backtracking for performance enhancement was successfully implemented within a tree-search structure.

Relevance of Performance Evaluation Techniques

* Despite the sample size tested, scatter was observed in performance as a function of machine load. Smoothing of raw data was required to simplify analysis.

* This scatter demonstrates the limitations of instance evaluation as a suitable evaluation technique. The results achieved from

instance evaluation are expected to be highly sensitive to the instance selected.

* Simulation provides the ability to rapidly examine system performance over a range of controlled experimental conditions. Most system characteristics can be adequately modelled by appropriate formulation of simulation routines.

* Expressing performance as a function of machine load provides a useful mechanism for analyzing overall performance.

Choice of Job Scheduling Rule

* The Shortest Processing Time (SPT) rule, supplemented with the pattern utilization heuristic, provided the best performance over the range of machine loads tested. The performance of this rule without pattern utilization enhancement was poor.

* The Cost Over Time (COVERT) rule provided the best performance over the range of machine loads tested when the pattern utilization heuristic was not employed.

* At low machine loads the best performance of the COVERT rule was achieved with the waiting time parameter equal to 3. At high loads the best performance was achieved with this equal to 4.

* The relative performance of the Apparent Tardiness Cost rule improves with increasing machine load. At loads above those tested it may provide better performance than the SPT and COVERT rules.

REFERENCES

1. Abramson D, "Constructing School Timetables Using Simulated Annealing: Sequential and Parallel Algorithms", Management Science, Vol. 37, No. 1, January (1991), pp 98-113.

2. Adams J, Balas E and Zawack D, "The Shifting Bottleneck Procedure for Job Shop Scheduling", Management Science, Vol. 34, No. 3, March (1988), pp 391-401.

3. Arkin E M and Roundy R O, "Weighted-Tardiness Scheduling on Parallel Machines with Proportional Weights", Operations Research, Vol. 39, No. 1, January - February (1991), pp 64-80.

4. Assad A and Golden B, "A Categorized Bibliography of Survey Articles in Management Science and Operations Research", Management Science, Vol. 28, No. 4, April (1982), pp 425-438.

5. Baker K R, "Introduction to Sequencing and Scheduling", John Wiley and Sons, (1974).

6. Baker K R, Lawler E L, Lenstra J K and Rinnooy Kan A H G, "Preemptive Scheduling of a Single Machine to Minimize Maximum Cost Subject to Release Dates and Precedence Constraints", Operations Research, Vol. 31, No. 2 (1983), pp 381-386.

7. Baker K R and Scudder G D, "Sequencing with Earliness and Tardiness Penalties: A Review", Operations Research, Vol. 38, No. 1, January - February (1990), pp 22-36.

8. Baker T E, "A Branch and Bound Network Algorithm for Interactive Process Scheduling", Mathematical Programming Study, 15, (1981), pp 43-57.

9. Baker T E, "Algorithms for Interactive Scheduling", ASOR Bulletin, Vol. 2, No. 3, November (1982), pp 9-14.

10. Ball M and Magazine M, "The Design and Analysis of Heuristics", Networks, 11, (1981), pp 215-219.

11. Barker J R and McMahon G B, "Scheduling the General Job-Shop", Management Science, Vol. 31, No. 5, May (1985), pp 594-598.

12. Bean J C, Birge J R, Mittenthal J, Noon C E, "Matchup Scheduling with Multiple Resources, Release Dates and Disruptions", Operations Research, Vol. 39, No. 3, May-June (1991), pp 470-483.

13. Brazile R P and Swigger K M, "GATES, An Airline Gate Assignment and Tracking Expert System", IEEE Expert, Summer (1988), pp 33-39.

14. Bruvold N T and Evans J R, "Flexible Mixed-Integer Programming Formulations for Production Scheduling Problems", IIE Transactions, Vol. 17, No. 3 (1985), pp 2-7.

15. Bureau of Transport and Communications Economics (BTCE), "Transport of Hydrocarbons in the Oil and Gas Industries", Information Paper 31, Australian Government Publishing Service (1989).

16. Carlier J and Pinson E, "An Algorithm for Solving the Job-Shop Problem", Management Science, Vol. 35, No. 2, February (1989), pp 164-176.

17. Christofides N, Mingozzi A and Toth P, "Dynamic Loading and Unloading of Liquids into Tanks", Summer School in Combinatorial Optimization, SOGESTA, Urbino (Italy), May 30/June 11 (1977).

18. Christofides N, Carpaneto G, Mingozzi A and Toth P, "The Loading of Liquids into Tanks", Summer School in Combinatorial Optimization, SOGESTA, Urbino (Italy), May 30/June 11 (1977).

19. Clark W, "The Gantt Chart", Sir Isaac Pitman and Sons Ltd, London, 3rd Ed, (1955).

20. Conway R W, Maxwell W L, and Miller L W, "Theory of Scheduling", Addison-Wesley Inc. (1967).

21. Crowder H P, Dembo R S and Mulvey J M, "Reporting Computational Experiments in Mathematical Programming", Mathematical Programming, 15, (1978), pp 316-329.

22. Darby-Dowman K and Mitra G, "An Extension of Set Partitioning with Application to Scheduling Problems", European Journal of Operational Research, 21, (1985), pp 200-205.

23. Davis E W, "Networks: Resource Allocation", Journal of Industrial Engineering, Vol. 6, (1974), pp 22-32.

24. Dror M, Stern H L and Lenstra J K, "Parallel Machine Scheduling: Processing Rates Dependent on Number of Jobs in Operation", Management Science, Vol. 33, No. 8, August (1987), pp 1001-1009.

25. Eglese R W, "Heuristics in Operational Research", book edited by Belton V and O'Keefe R M, "Recent Developments in Operational Research", Pergamon, Oxford, pp 49-67.

26. Federgruen A and Groenevelt H, "Preemptive Scheduling of Uniform Machines by Ordinary Network Flow Techniques", Management Science, Vol. 32, No. 3, March (1986), pp 341-349.

27. Fisher M L and Rinooy Kan A H G, "The Design, Analysis and Implementation of Heuristics", Management Science, Vol. 34, No. 3, March (1988), pp 263-265.

28. Fox B L, "Data Structures and Computer Science Techniques In Operations Research", Operations Research, Vol. 26, No. 5, September - October (1978), pp 686-717.

29. Foulds L R, "The Heuristic Problem-Solving Approach", J. Opl. Res. Soc., Vol. 34, No. 10, (1983), pp 927-934.

30. French S, "Sequencing and Scheduling", Ellis Horwood Ltd, (1982).

31. Garey M R, Graham R L and Johnson D S, "Performance Guarantees for Scheduling Algorithms", Operations Research, Vol. 26, No. 1, January - February (1978), pp 3-21.

32. Garfinkel R and Nemhauser G L, "Integer Programming", John Wiley and Sons (1972).

33. Gere W S, "Heuristics in Job Shop Scheduling", Management Science, Vol. 13, No. 3, November (1966), pp 167-190.

34. Gilmore P C, "Cutting Stock, Linear Programming, Knapsacking, Dynamic Programming and Integer Programming,some Interconnections", Annals of Discrete Mathematics, 4, (1979), pp 217-235.

35. Graves S C, "A Review of Production Scheduling", Operations Research, Vol. 29, No 4, July - August (1981), pp 646-675.

36. Haessler R W, "Developing an Industrial-Grade Heuristic Problems-Solving Procedure", Interfaces, 13:3, June (1983), pp 62-71.

37. Haessler R W, "Selection and Design of Heuristic Procedures for Solving Roll Trim Problems", Management Science, Vol. 34, No. 12, December (1988), pp 1460-1471.

38. Haessler R W and Talbot F B, "A 0-1 Model for Solving the Corrugator Trim Problem", Management Science, Vol. 29, No. 2, February (1983), pp 200-209.

39. Haverly Systems, "OMNI Modelling and Data Management System Reference Manual", Haverly Systems Inc., Denville, NJ, USA, Version 4.3, February (1985).

40. Horowitz E and Sahni S, "Exact and Approximate Algorithms for Scheduling Nonidentical Processors", Journal of the Association for Computing Machinery, Vol. 23, No. 2, April (1976), pp 317-327.

41. IBM, "IBM Mathematical Programming System Extended/370, General Information Manual", Program No. 5740-XM3 (OS/VS), GH 19-1090-4, File No. S370-82, IBM Corporation, NY, USA, 5th Edition, (1979).

42. Jeroslow R G and Lowe J K, "Experimental Results on the New Techniques for Integer Programming Formulations", Journal of the Operational Research Society, Vol. 36, No. 5 (1985), pp 393-403.

43. Johnson E L, Kostreva M M and Suhl U H, "Solving 0-1 Integer Programming Problems Arising from Large Scale Planning Models", Operations Research, Vol. 33, No. 4 July - August (1985), pp 803-819.

44. Johnston R E, "Reducing Cardinality in Linear Programmed Cutting Stock Solutions", Asia-Pacific Journal of Operational Research, Vol. 7, No. 1, May (1990), pp 1-8.

45. Johnston R E, "Developing Factory-Wide Control Strategies", Japan Paper Journal, Vol. 28, No. 1, (1985).

46. Johnston R E, "A Direct Combinatorial Algorithm of Cutting Stock Problems", in "The Applications of Mathematics to Industry", Anderson R S and de Hoog F R editors, Martinus Nijhoff, The Hague, (1982), pp 137-152.

47. Jones C H, "An Economic Evaluation of Job Shop Dispatching Rules", Management Science, Vol. 20, No. 3, November (1973), pp 293-307.

48. Jones C V, "The Three-Dimensional Gantt Chart", Operations Research, Vol. 36, No. 6, November - December (1988), pp 891-903.

49. Keng N P, Yun D Y Y and Rossi M, "Interaction-Sensitive Planning System for Job-Shop Scheduling", Expert Systems and Intelligent Manufacturing, (1988), pp 57-69.

50. Kozan E, "Comparison of Heuristic Rules for Job Shop Scheduling", ASOR Bulletin, Vol. 10, No. 3, Sep (1991), pp 2-10.

51. Land A and Powell S, "Computer Codes for Problems of Integer Programming", Annals of Discrete Mathematics, 5 (1979), pp 221-269.

52. Lasdon L S, "Optimization Theory for Large Systems", Collier MacMillan, London, (1970).

53. Metcalf, "Fortran Optimization", Revised Ed, Academic Press, (1985).

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

55. Pearl J, "Heuristics: Intelligent Search Strategies for Computer Problem Solving", Addison-Wesley Inc., (1984).

56. Posner M E, "A Sequencing Problem with Release Dates and Clustered Jobs", Management Science, Vol. 32, No. 6, June (1986), pp 731-738.

57. Potts C N and Van Wassenhove L N, "A Branch and Bound Algorithm for the Total Weighted Tardiness Problem", Operations Research, Vol. 33, No. 2, March - April (1985), pp 363-377.

58. Potts C N and Van Wassenhove L N, "Algorithms for Scheduling A Single Machine to Minimize the Weighted Number of Late Jobs", Management Science, Vol. 34, No. 7, July (1988), pp 843-858.

59. Rutherford N, "Optimizing the Mill Bottleneck: A Computer-Based Production Coordination Tool", PIMA, March (1984), pp 44-49.

60. Silver E A, Vidal R V and de Werra D, "A Tutorial on Heuristic Methods", European Journal of Operational Research, 5, (1980), pp 153-162.

61. So K C, "Some Heuristics for Scheduling Jobs on Parallel Machines with Setups", Management Science, Vol. 36, No. 4, April (1990), pp 467-475.

62. Solomon M M, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", Operations Research, Vol. 35, No. 2, March-April (1987), pp 254-265.

63. Stinson J P, Davis E W and Khumawala B M, "Multiple Resource - Constrained Scheduling Using Branch and Bound", AIIE Transactions, Vol. 10, No. 3, September (1978), pp 252-259.

64. Taha H A, "Operations Research", 2nd Ed, Collier MacMillan Co. Inc., (1976).

65. Talbot F B and Patterson J H, "An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems", Management Science, Vol. 24, No. 11, July (1978), pp 1163-1174.

66. Talbot F B "Resource-Constrained Project Scheduling with Time - Resource Trade-offs: The Non-preemptive Case", Management Science, Vol. 28, No. 10, October (1982), pp 1197-1210.

67. Tomlin J A, "Large Scale Mathematical Programming Systems", Computers and Chemical Engineering, Vol. 7, No. 5, (1983), pp 575-582.

68. True W R, "Systems Yield Data to Expedite Products-Pipeline Scheduling", Oil and Gas Journal, January 9 (1989), pp 38-42.

69. Van de Vel H and Shijie S, "An Application of the Bin-packing Technique to Job Scheduling on Uniform Processors", J. Opl Res. Soc., Vol 42, No. 2, (1991), pp 169-172

70. Vepsalainen A P J and Morton T E, "Priority Rules for Job Shops with Weighted Tardiness Costs", Management Science, Vol. 33, No. 8, August (1987), pp 1035-1047.

71. Walpole R E and Myers R H, "Probability and Statistics for Engineers and Scientists, 2nd Ed., Collier Macmillan, (1978).

72. White D J, "A Complimentary Greedy Heuristic for the Knapsack Problem", European Journal of Operational Research, Vol. 62, (1992), pp 85-95.

73. Wilbrecht J K and Prescott W B, "The Influence of Setup Time on Job Shop Performance", Management Science, Vol. 16, No. 4, December (1969), pp B274-B280.

74. Williams H P, "Model Building in Mathematical Programming", 2nd Ed, John Wiley and Sons (1985).

75. Young J A, "Scheduling of Pipeline Transfers from PRA's Altona Refinery using a Mixed Integer Programming Approach", Industry Project Report, Footscray Institute of Technology, (1985).

76. Zionts S, "Linear and Integer Programming", Prentice-Hall Inc. (1974).

Appendix A HISTORICAL DATA

A.1 Job and Lineclear Numbers

Table A.1 Observed Job and Lineclear Numbers

Week commencing	Jobs PL1	PL4	PL2	PL3	Total	Lineclears PL1	PL4	PL2	PL3	Total
12/03/86	11	10	10	1	32	2	2	6	1	11
19/03/86	8	9	5	4	26	4	4	3	2	13
26/03/86	11	10	10	5	36	0	0	3	3	6
02/04/86	9	9	7	5	30	0	0	4	4	8
09/04/86	12	12	11	4	39	5	5	0	0	10
16/04/86	11	12	17	4	44	4	4	4	2	14
23/04/86	14	13	8	5	40	3	3	3	2	11
30/04/86	14	14	13	2	43	5	2	3	1	11
07/05/86	12	7	14	6	39	3	2	2	2	9
14/05/86	15	12	10	3	40	2	3	3	2	10
21/05/86	16	11	16	5	48	2	2	2	0	6
28/05/86	12	14	11	3	40	2	1	1	3	7
04/06/86	11	12	11	5	39	4	2	2	4	12
11/06/86	16	12	15	4	47	2	2	2	3	9
18/06/86	10	12	9	4	35	2	4	2	1	9
25/06/86										
02/07/86										
09/07/86										
16/07/86										
23/07/86										
30/07/86	11	12	16	3	42	3	3	5	3	14
06/08/86	13	13	10	4	40	3	3	2	2	10
13/08/86	14	16	13	6	49	2	2	4	2	10
20/08/86	14	11	9	3	37	4	4	0	1	9
27/08/86	12	8	13	7	40	2	2	4	2	10
03/09/86	12	12	10	5	39	2	2	1	0	5
10/09/86	14	8	8	5	35	2	2	1	2	7
17/09/86	11	10	10	8	39	0	0	3	0	3
24/09/86	17	15	6	7	45	4	4	6	0	14
01/10/86	8	10	12	6	36	2	2	6	2	12
08/10/86	10	9	11	7	37	0	0	2	2	4
15/10/86	14	11	10	7	42	2	2	2	3	9
22/10/86	12	15	11	7	45	5	5	4	3	17
29/10/86	8	10	8	1	27	3	3	2	0	8
05/11/86	9	12	7	3	31	2	4	2	1	9
12/11/86	9	9	8	0	26	2	2	0	0	4
19/11/86	11	11	14	0	36	2	2	2	0	6
Total	381	361	343	139	1224	80	78	86	53	297
Average	11.9	11.3	10.7	4.3	38.3	2.5	2.4	2.7	1.7	9.3
Minimum	8.0	7.0	5.0	0.0	26.0	0.0	0.0	0.0	0.0	3.0
Maximum	17.0	16.0	17.0	8.0	49.0	5.0	5.0	6.0	4.0	17.0
Variance	5.6	4.5	8.6	4.2	34.6	1.9	1.8	2.6	1.5	10.2
Std Dev	2.4	2.1	2.9	2.1	5.9	1.4	1.3	1.6	1.2	3.2
Count	32	32	32	32	32	32	32	32	32	32

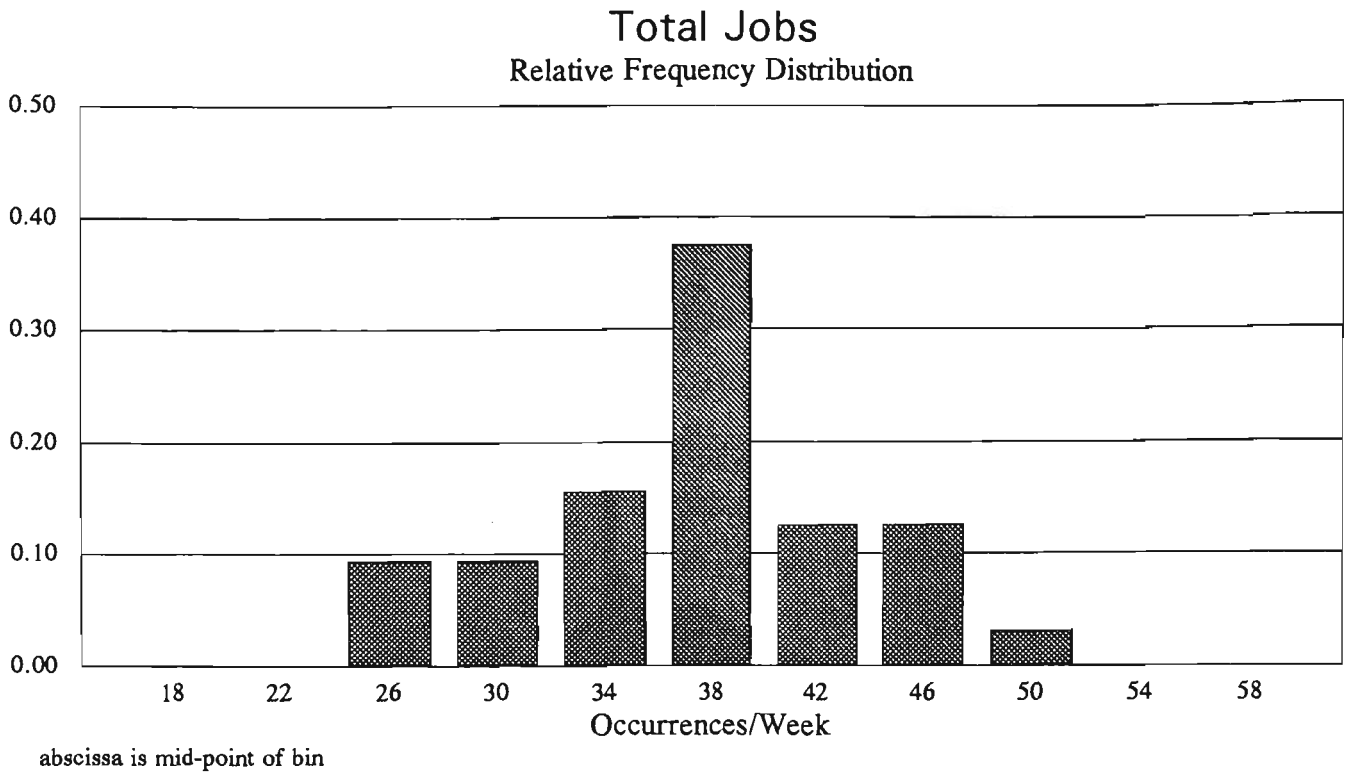


Figure A.1 Total Jobs relative frequency distribution

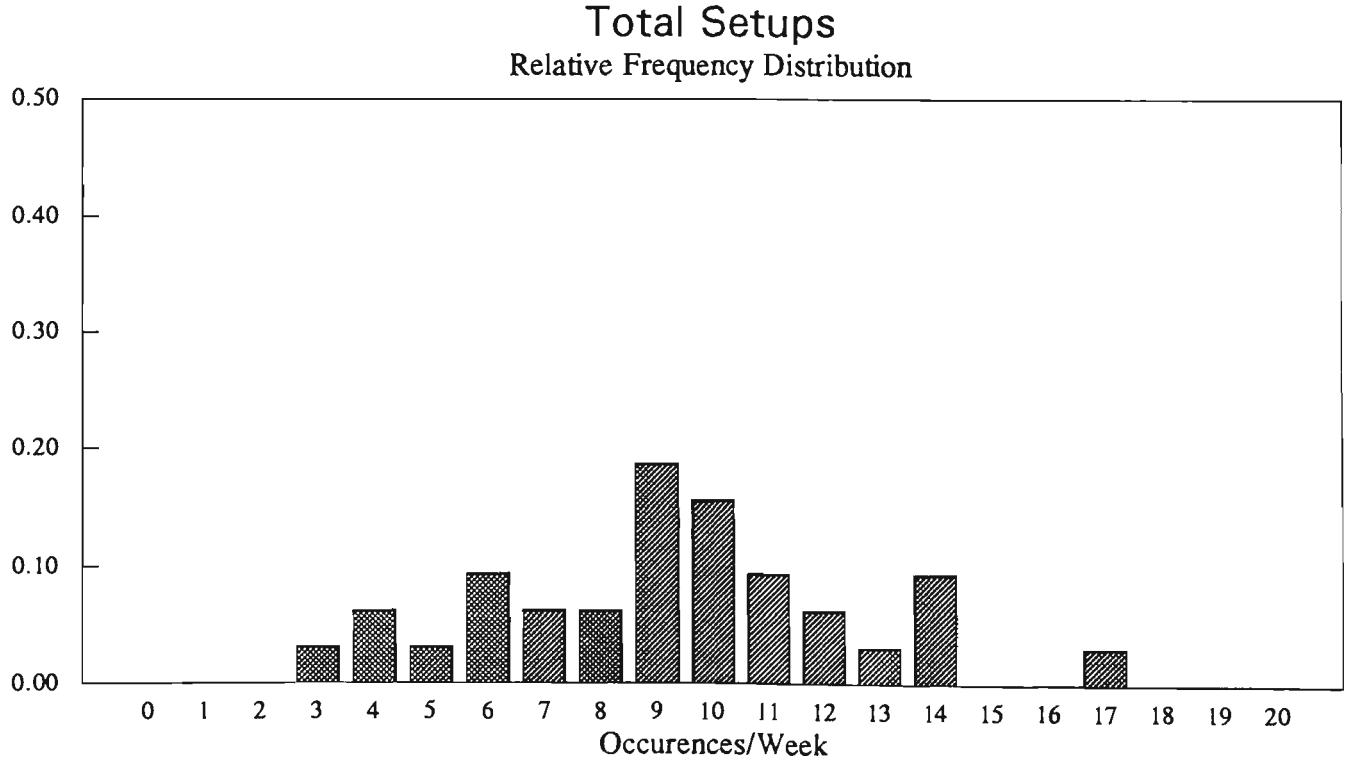


Figure A.2 Total Setups relative frequency distribution

Appendix B FREQUENCY DISTRIBUTIONS

B.1 Ready Time Distribution

In practice, a number of jobs are sourced from each product batch, and hence the job ready times are clustered at the batch release times. This characteristic was modelled in the simulation experiments by ensuring jobs of the same product type had the same ready time until the volume of the batch was exhausted. A new batch time was then established and the process repeated. For example:

Table B.1 Job Ready Time Example

Batch Number	Job Number	Product Type	Job Volume ML	Batch Volume Remain. ML	Batch Release Time DD/HH	Job Ready Time DD/HH
J01		JET		9.0	1/06	
A01		ADO		9.0	1/12	
	1	JET	2.0	7.0		1/06
	2	ADO	3.5	6.5		1/12
	3	ADO	4.0	2.5		1/12
	4	JET	3.0	4.0		1/06
	5	ADO	2.5	0.0		1/12
A02		ADO		9.0	2/20	
	6	ADO	5.0	4.0		2/20
	7	JET	2.0	2.0		1/06

The batch release times were assumed to be uniformly distributed between 24 and 36 hours after the preceding batch release time for the product concerned. A constant batch size of 9.0 ML was assumed for each product.

B.2 Due Time Distribution

As discussed in Chapter 5, the due time was offset from the job ready time by a multiple of the nominal job processing time. The nominal processing time equalled the job volume divided by the average processing rate for the particular offtaker / product combination.

Although not used in the simulation experiment, the due time could also be specified in terms of the average batch processing time, to reflect tank scheduling limitations. The multiplier used in this option was estimated as follows, using motor gasoline (mogas) as typical of all products.

1. Average production rate:

Average mogas batch size	=	9.0 ML
Nominal mogas production rate	=	55 ML/wk
Therefor, average production rate/batch	=	27 h

2. Average pumping rate:

Total mogas pumping rate (PL1+PL4)	=	385 kL/h
Therefor, average batch pumping time	=	23 h

Allowing for capacity losses due to setups and delays the above estimates indicate that the average pumping rate and average production rate are approximately balanced. Three tanks are available for mogas service; typically two are being pumped intermittently (one has just started being pumped and one has almost finished being pumped), and one is being blended, as follows.

Tank	Time ->
A	<------Pump------><-Blend-><------Pump------><-Blend->
B	<-Blend-><------Pump------><-Blend-><------Pump------>
C	<..Pump-><-Blend-><------Pump------><-Blend-><-Pump..>

Almost always each tank is emptied before being re-blended. Hence the due time for all the jobs allocated from a particular tank batch must be less than the batch ready time plus twice the total batch pumping time.

$$D_k \leq R_k + 2*P \qquad \text{for all } k$$

where P = average batch pumping time, and
k = jobs sourced from the given product batch

B.3 Machine Utilization

The "average" refinery production was set equal to the 1985 full year actual refinery production. The production volumes of each product were assigned to the appropriate pipelines. Hence, all of the gasoline production, such as premium leaded, unleaded and avgas, was assigned to Pipelines 1 and 4. All of the distillate, 65 % of the jet and 10 % of the fuel oil were assigned to Pipelines 2 and 3 (the remainder of the jet and fuel oil being pumped on other pipeline networks).

The "nominal capacity" refinery production was based on the maximum rated refinery throughput. The product mix was assumed equal to the "average" refinery production mix.

Table B.2 Pipeline Utilization

		PL1 G	PL4 G	PL2 D	PL3 D	Total
Average:						
Volume	ML/wk	24	28	21	10	83
Average Rate	kL/h	175	207	190	160	-
Pumping Time	h/wk	136	136	109	60	441
Utilization	%	81	81	65	36	66
Nominal Capacity:						
Volume	ML/wk	26	31	23	11	91
Average Rate	kL/h	175	207	190	160	-
Pumping Time	h/wk	149	149	121	69	488
Utilization	%	89	89	72	41	73
Note:						
Capacity	kL/h	175	207	190	160	732
	%	24	28	26	22	100

"G" = gasolines (premium unleaded, unleaded, leaded, avgas, etc)
"D" = distillates (jet, kerosene, heating oil, diesel, etc)

Gasoline Capacity = PL1 + PL4 = 175+207 = 382 kL/h = 52 %
Distillate Capacity = PL2 + PL3 = 190+160 = 350 kL/h = 48 %

The following table summarizes the average allocation of pipeline time observed during 1985. The closure reported below is the ratio of the total time accounted for relative to the total time available, and hence measures the quality of the allocation.

Table B.3 Pipeline Time Allocation

		PL1 G	PL4 G	PL2 D	PL3 D	Total
Pumping Time:						
Volume	ML/wk	24	28	21	10	83
Average Rate	kL/h	175	207	190	160	-
Pumping Time	h/wk	136	136	109	60	441
Setup Time:						
Operations		11.9	11.3	10.7	4.3	38.2
Lineclears		2.5	2.6	2.7	1.7	9.5
Time/op.	h	1	1	1	1	-
Time/lc.	h	2	2	2	2	-
Setup: op.	h	12	12	12	4	40
Setup: lc.	h	5	5	6	5	21
Total setup	h	17	17	18	9	61
Idle Time:	h	9	7	40	86	226
Total time	h	162	160	167	155	644
Available	h	168	168	168	168	672
closure	%	96	95	99	92	96

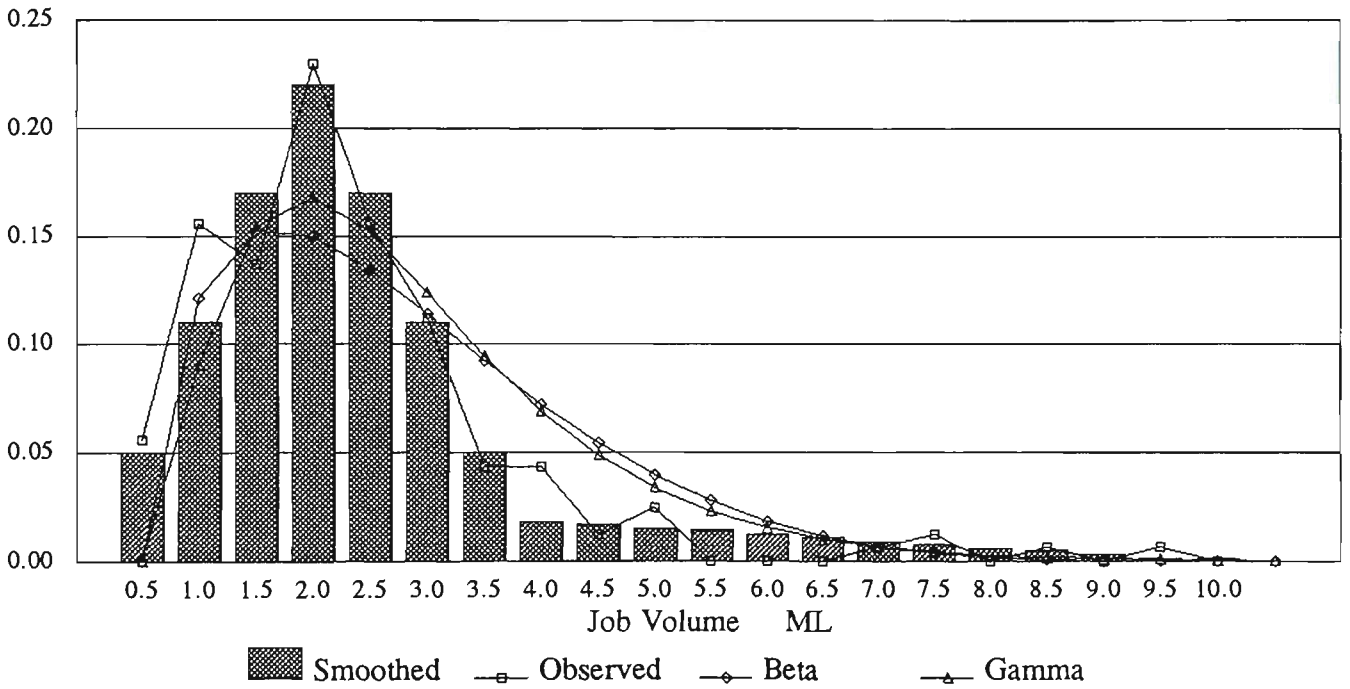
"G" = gasolines (premium unleaded, unleaded, leaded, avgas, etc)
"D" = distillates (jet, kerosene, heating oil, diesel, etc)

B.4 Job Volume Distribution

The volume of each job was drawn from the empirical distribution shown below. This distribution was based on data drawn from 5 weeks of actual schedules, shown in Appendix A. Beta and gamma distributions calculated from the observed data are shown for comparison with the observed data and the modified empirical distribution used. As can be seen, the modified empirical distribution provides a better fit than the beta and gamma predictions.

The upper and lower limits of the distribution were truncated at 10.0 ML and 0.5 ML respectively, corresponding to the minimum and maximum parcel sizes generally observed. The continuous distribution was approximated by a discrete distribution of 20 bins of equal width. The mode of 2.0 ML was maintained, although its frequency was reduced slightly to ensure and area of 1 under the curve.

Figure B.1 Job Volume relative frequency distribution



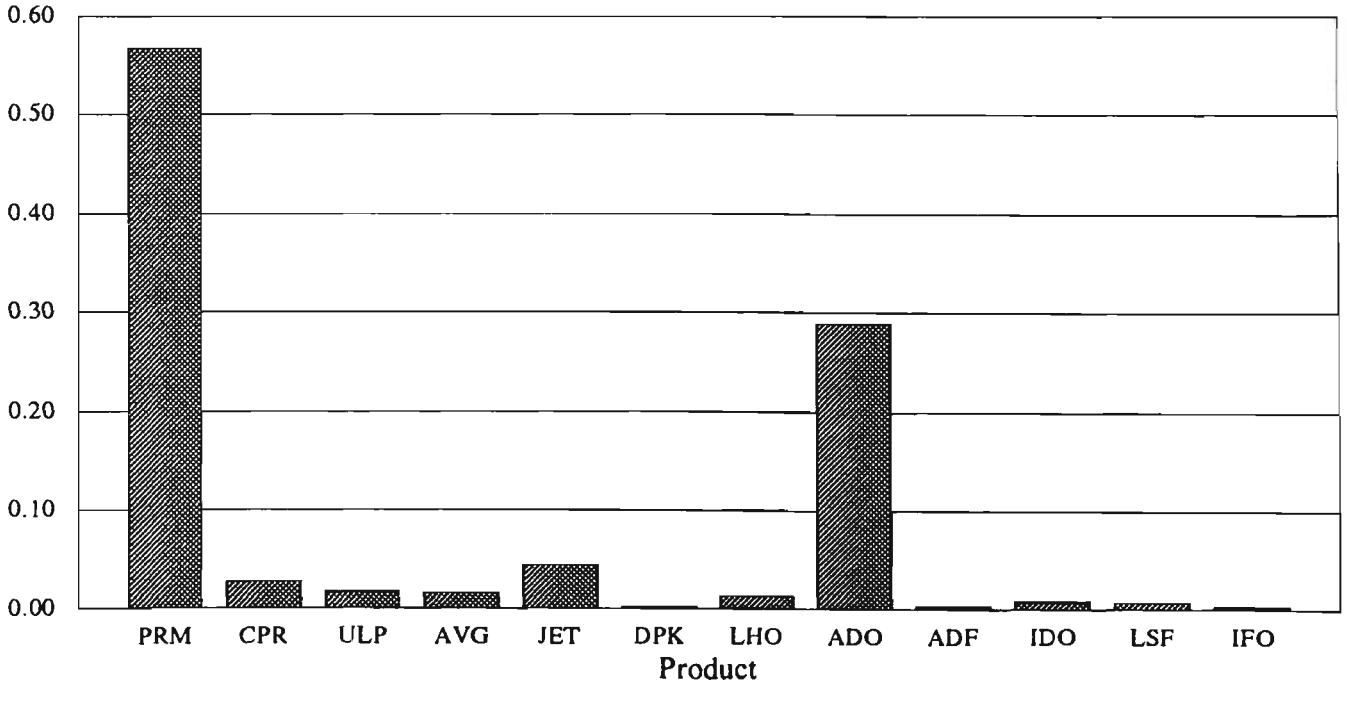
B.5 Product Distribution

The product type distribution was based upon the adjusted 1985 full year production mix described. Of the total 1985 refinery production only 35 % of the total jet production was assumed to be pumped on the pipeline network, the remaining 65 % was pumped on an independent pipeline. 90 % of fuel oil production was also excluded because it was pumped on a different pipeline network.

Table B.4 Product Frequency Distribution

Product Code	Name	Refinery Production vol %	Moved over Pipeline %	Raw Freq. Distbn %	Normal. Freq. Distbn %	Cumul. Freq. Distbn %
PRM	Premium Gasoline	49.4	100	49.4	56.7	56.7
CPR	Country Premium	2.4	100	2.4	2.8	59.5
ULP	Unleaded Gasoline	1.6	100	1.6	1.8	61.3
AVG	Avgas	1.4	100	1.4	1.6	62.9
JET	Jet Fuel 1A	10.8	35	3.8	4.4	67.3
DPK	Dual Purpose Kero	0.2	100	0.2	0.2	67.5
LHO	Light Heating Oil	1.1	100	1.1	1.3	68.8
ADO	Auto Diesel Oil	25.1	100	25.1	28.9	97.7
ADF	Auto Diesel Fuel	0.3	100	0.3	0.3	98.0
IDO	Industrial Diesel	0.8	100	0.8	0.9	98.9
LSF	Low Sulphur Fuel Oil	6.6	10	0.7	0.8	99.7
IFO	Industrial Fuel Oil	0.3	100	0.3	0.3	100.0
Total		100.0	-	87.1	100.0	-

Figure B.2 Product Type relative frequency distribution



B.6 Offtaker Distribution

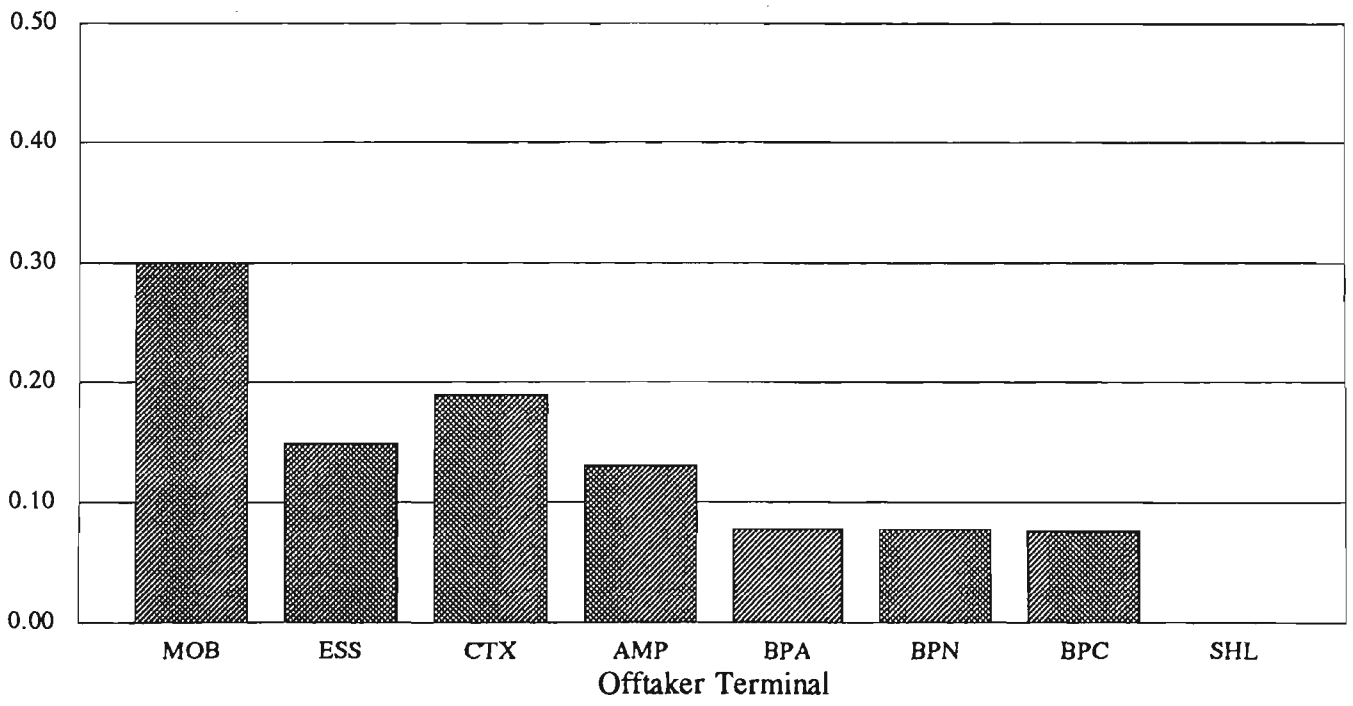
The relative frequency distribution of offtaker destinations was based upon an assessment of the amount of product transferred to each offtaker. This was estimated based upon the offtaker's overall market share and an assessment of how much of each offtaker's demand was sourced from the refinery.

A uniform distribution was used to allocate product between terminals for offtakers with multiple terminals.

Table B.5 Offtaker Frequency Distribution

Offtaker Code	Name	Total Market Share %	Altona Supplied Amount %	Raw Freq. Distbn %	Normal. Freq. Distbn %	Cumul. Freq. Distbn %
MOB	Mobil	14	100	14	30	30
ESS	Esso	7	100	7	15	45
CTX	Caltex	17	50	9	19	64
AMP	Ampol	12	50	6	13	77
BPA	BP Altona	7	50	3	7	84
BPN	BP Newport	8	50	4	9	93
BPC	BP Coode Island	7	50	4	7	100
SHL	Shell	28	0	0	0	100
Total		100	-	47	100	-

Figure B.3 Offtaker Terminal relative frequency distribution



Appendix C EXPERIMENTAL INPUT

Figure C.1 Example 1: Schedule

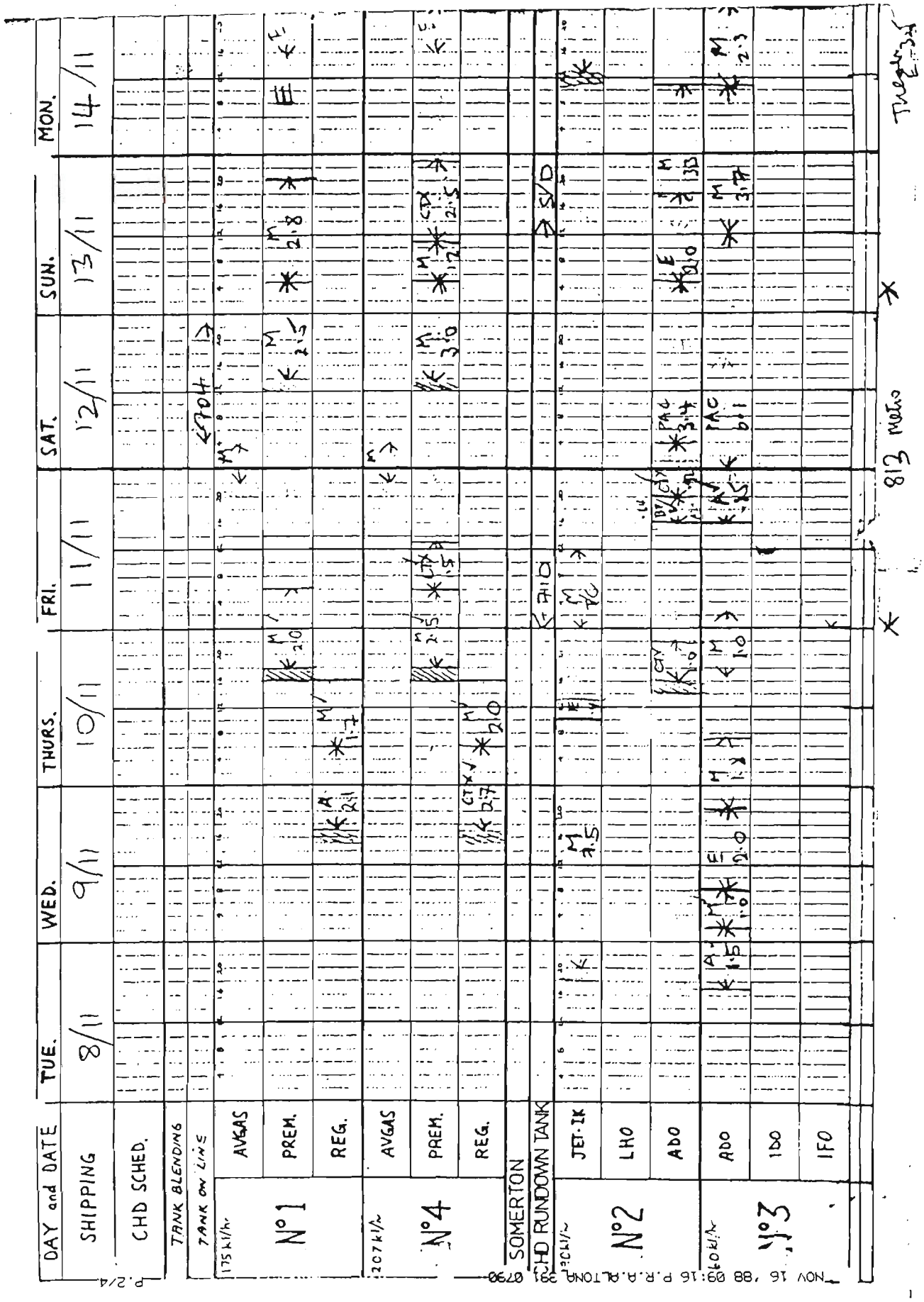


Figure C.2 Example 1: Job Definition File

```
DSPRT   921213-162736-SUN DSNAME M995092.TECH.DATA(APTSJOB1)

REMARK: JAYOUNG, 26 OCT 92, EXAMPLE 1 (08 NOV 88): RAN (PTY UTZ)
*       DAY        DATE     MNTH     MAXD     NDAY
DATES : TUE          8.       11.      30.       7.
*
DETAILS:OFT   PRD   BCH   TNK   PTY      VOL      RDY      DUE      DEAD
*
              PRM                PL1       .        .        .         .
              PRM                PL4       .        .        .         .
              JET                PL2       .        .        .         .
              ADO                PL3       .        .        .         .
        AMP   ULP                          2.1     9.16    10.06       .
        MOB   ULP                          3.7     9.16    10.16       .
        CTX   ULP                          2.7     9.16    10.06       .
        MOB   PRM                          4.5    10.12    11.06       .
        CTX   PRM                          1.5    10.12    11.13       .
        MOB   AVG   V69   508              2.5    11.20    12.04       .
        MOB   CPR   P09   704              5.5    12.12    13.05       .
        MOB   PRM   P10   811              4.0    13.00    13.20       .
        CTX   PRM   P10   811              2.5    13.00    13.23       .
        MOB   JET                          9.5     8.17    11.12       .
        ESS   JET                          0.4     8.17    10.13       .
        AMP   ADO                          1.5     8.17     9.02       .
        MOB   ADO                          2.8     8.17    11.03       .
        ESS   ADO                          2.0     8.17     9.21       .
        MOB   ADO                          1.0     8.17    11.03       .
        CTX   ADO                          1.0     8.17    10.22       .
        BPA   ADO   A21   812              0.6    11.16    11.20       .
        CTX   ADO   A21   812              0.9    11.16    12.01       .
        AMP   ADO   A21   812              0.9    11.16    12.00       .
        MOB   NZA   A22   700              9.5    11.16    13.12       .
        MOB   ADO   A21   812              7.0    11.16    14.11       .
        ESS   ADO   A21   812              2.0    11.16    13.18       .
```

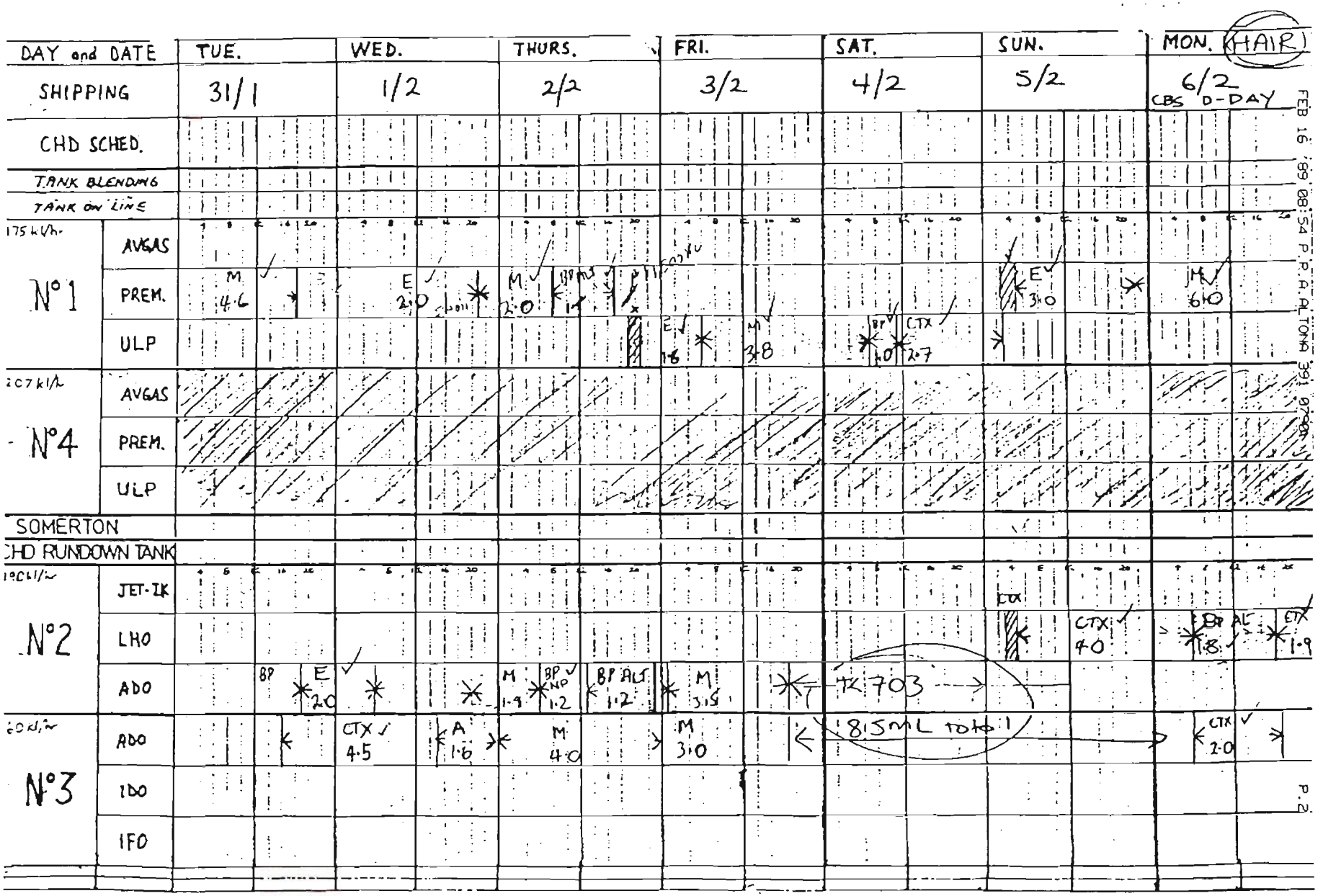


Figure C.3 Example 2: Schedule

Figure C.4 Example 2: Job Definition File

```
DSPRT   921213-162738-SUN DSNAME M995092.TECH.DATA(APTSJOB2)

REMARK: JAYOUNG, 26 OCT 92, EXAMPLE 2 (31 JAN 89), RAN (PTY UTZ)
*       DAY        DATE     MNTH     MAXD     NDAY
DATES : TUE         31.       1.      31.       7.
*
DETAILS:OFT    PRD    BCH    TNK    PTY      VOL      RDY      DUE      DEAD
*
        MOB    PRM                  PL1      3.2    31.00    31.18       .
               PRM                  PL4       .       .        .        .
               ADO                  PL3       .       .        .        .
        ESS    PRM                           2.0     1.05     1.20       .
        MOB    PRM                           2.0     1.05     2.08       .
        BPA    PRM                           1.5     1.05     2.17       .
        ESS    ULP                           1.8     2.19     3.04       .
        MOB    ULP                           3.8     2.19     4.07       .
        BPN    ULP                           1.0     2.19     4.11       .
        CTX    ULP                           2.7     2.19     5.03       .
        ESS    PRM                           3.0     5.03     5.22       .
        MOB    PRM                           6.0     5.03     7.09       .
        BPN    ADO                  PL2      1.8    31.00    31.19       .
        ESS    ADO                           2.0    31.16     1.06       .
        MOB    ADO                           1.9    31.16     2.06       .
        BPN    ADO                           1.2    31.16     2.13       .
        BPA    ADO                           1.2    31.16     2.23       .
        MOB    ADO                           6.5    31.16     3.19       .
        CTX    LHO                           5.9     5.03     7.04       .
        BPA    LHO                           1.8     5.03     6.18       .
        CTX    ADO                           4.5    31.16     1.15       .
        AMP    ADO                           1.6    31.16     2.00       .
        MOB    ADO                           4.0    31.16     3.00       .
        MOB    ADO                           8.5     3.12     6.06       .
        CTX    ADO                           2.0     3.12     6.19       .
```

Figure C.5 **Example 3: Schedule**

Figure C.6 Example 3: Job Definition File

```
DSPRT   921213-162740-SUN DSNAME M995092.TECH.DATA(APTSJOB3)

REMARK: JAYOUNG, 26 OCT 92, EXAMPLE 3 (07 FEB 89): RAN (PTY UTZ)
*       DAY        DATE     MNTH     MAXD     NDAY
DATES : TUE          7.       2.      28.       7.
*
DETAILS:OFT   PRD   BCH   TNK   PTY      VOL      RDY      DUE      DEAD
*
        MOB   PRM                PL1      1.4     7.00     7.08      .
              PRM                PL4       .        .        .        .
              ADO                PL3       .        .        .        .
        ESS   PRM                         1.5     7.00     7.15      .
        MOB   PRM                         2.1     7.00     8.05      .
        ESS   PRM                         2.2     7.00     8.18      .
        MOB   ULP                         2.4     8.18     9.15      .
        BPA   ULP                         0.5     8.18     9.18      .
        ESS   ULP                         2.1     8.18    10.06      .
        CTX   ULP                         2.7     8.18    10.22      .
        MOB   ULP                         1.1     8.18    11.04      .
        ESS   PRM                         2.5    11.04    11.20      .
        CTX   PRM                         3.5    11.04    12.16      .
        MOB   PRM                         5.0    11.04    13.14      .
        ESS   PRM                         2.4    11.04    14.04      .
        CTX   PRM                         2.5    11.04    14.02      .
        CTX   LHO                PL2      1.3     7.00     7.07      .
        MOB   LHO                         1.5     7.00     7.18      .
        CTX   LHO                         2.8     7.00     8.10      .
        MOB   JET                         4.2     8.12     9.19      .
        CTX   JET                         0.5     8.12    10.00      .
        MOB   LHO                         3.8    10.00    10.20      .
        MOB   JET                         2.8    10.20    11.12      .
        MOB   ADO                         7.5    11.12    13.10      .
        BPA   ADO                         3.0    11.12    14.02      .
        ESS   ADO                         3.0     7.16     8.12      .
        MOB   ADO                         5.0     8.12     9.16      .
        ESS   ADO                         3.0     8.12    10.13      .
        MOB   ADO                         1.8     8.12    11.00      .
        CTX   ADO                         3.0    11.00    11.12      .
        MOB   ADO                         3.8    11.00    12.20      .
        MOB   IFO                         2.6    11.00    13.14      .
        ESS   ADO                         2.0    11.00    14.07      .
```

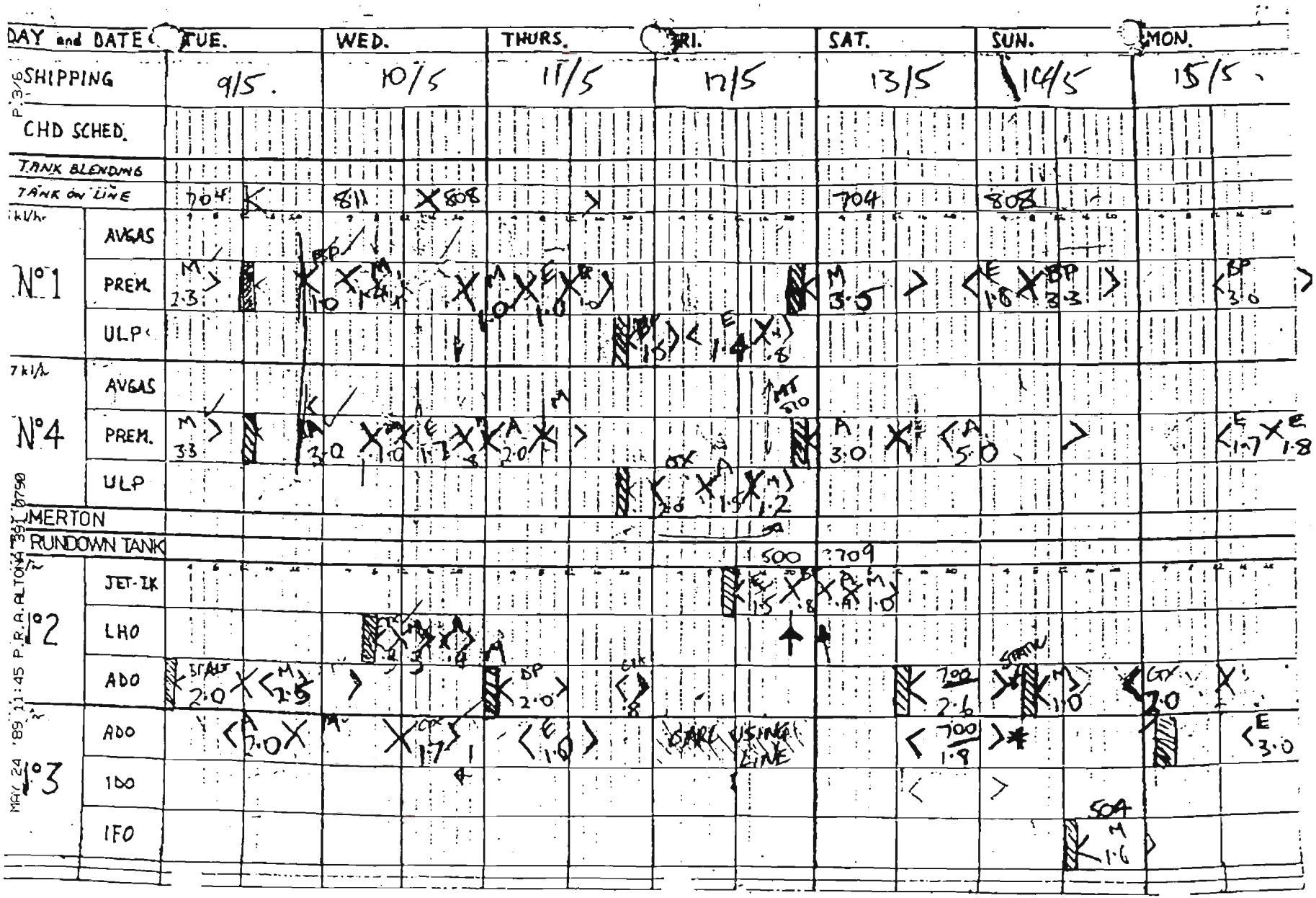


Figure C.7 Example 4: Schedule

Figure C.9 Example 4: Job Definition File

```
DSPRT   921213-162746-SUN DSNAME M995092.TECH.DATA(APTSJOB4)

REMARK: JAYOUNG, 26 OCT 92, EXAMPLE 4 (09 MAY 89): RAN (PTY UTZ)
*       DAY        DATE     MNTH     MAXD     NDAY
DATES : TUE          9.       5.      31.       7.
*
DETAILS:OFT    PRD    BCH    TNK    PTY      VOL      RDY      DUE      DEAD
*
        MOB    PRM                  PL1      2.1     9.00     9.12       .
        MOB    PRM                  PL4      2.4     9.00     9.12       .
               JET                  PL2       .        .        .        .
               ADO                  PL3       .        .        .        .
        BPN    CPR           811             1.0     9.12    10.04       .
        MOB    CPR           811             2.4     9.12    10.20       .
        MOB    CPR           808             1.0    10.15    11.06       .
        ESS    CPR           808             1.0    10.15    11.12       .
        BPN    CPR           808             1.0    10.15    11.18       .
        BPN    ULP           704             1.5    11.18    12.04       .
        ESS    ULP           704             1.4    11.18    12.16       .
        MOB    ULP           704             2.0    11.18    12.20       .
        MOB    PRM           808             3.5    12.20    13.16       .
        ESS    PRM           808             1.8    12.20    14.08       .
        BPN    PRM           808             3.3    12.20    14.20       .
        BPN    PRM                           3.0    15.12    16.05       .
        AMP    CPR           811             3.0     9.12    10.08       .
        ESS    CPR           811             1.7     9.12    10.22       .
        MOB    CPR           808             0.8    10.15    11.00       .
        AMP    CPR           808             2.0    10.15    11.08       .
        MOB    CPR           808             1.0    10.15    11.14       .
        CTX    ULP                           2.0    11.18    12.06       .
        AMP    ULP                           1.5    11.18    12.14       .
        AMP    PRM           704             8.0    12.20    14.16       .
        ESS    PRM           808             1.7    15.12    15.20       .
        BPA    ADO                           2.0     9.00     9.12       .
        MOB    ADO                           2.5     9.00    10.06       .
        CTX    LHO                           1.0    10.06    10.24       .
        BPN    ADO                           2.0    11.00    11.12       .
        CTX    ADO                           0.8    11.00    12.00       .
        ESS    JET                           1.5    12.10    12.20       .
        BPN    JET                           0.8    12.10    13.01       .
        AMP    JET                           0.9    12.10    13.05       .
        MOB    JET                           1.0    12.10    13.12       .
        MOB    ADO           700             4.4    13.12    14.06       .
        MOB    ADO                           1.0    14.06    14.20       .
        CTX    ADO                           2.0    14.06    15.13       .
        AMP    ADO                           2.0     9.00     9.20       .
        MOB    ADO                           2.5     9.00    10.12       .
        CTX    ADO                           1.7     9.00    10.21       .
        ESS    ADO                           1.5     9.00    11.16       .
        MOB    IFO           504             1.6    14.12    15.02       .
        ESS    ADO                           3.0    15.02    16.12       .
```

Figure C.10 Example 5: Schedule

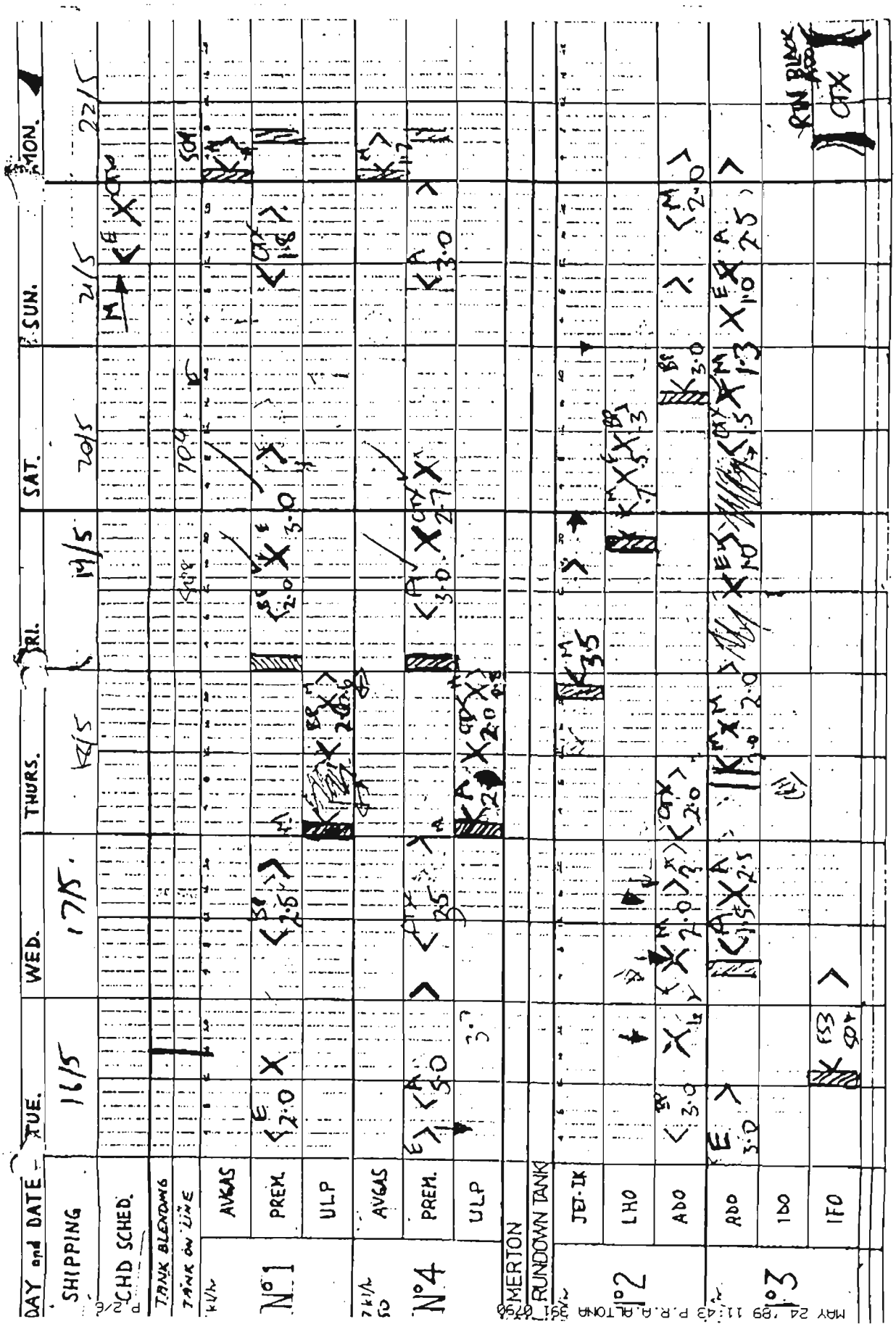


Figure C.11 Example 5: Job Definition File

```
DSPRT   921213-162748-SUN DSNAME M995092.TECH.DATA(APTSJOB5)

REMARK: JAYOUNG, 26 OCT 92, EXAMPLE 5 (16 MAY 89): EDD (PTY UTZ)
*       DAY        DATE     MNTH     MAXD     NDAY
DATES : TUE         16.       5.      31.       7.
*
DETAILS:OFT   PRD   BCH   TNK   PTY     VOL     RDY     DUE     DEAD
*
        ESS   PRM                 PL4     1.0   16.00   16.12      .
        ESS   PRM                 PL3     1.9   15.17   16.12      .
        ESS   PRM                         2.0   16.00   16.14      .
        BPN   PRM                         2.5   17.08   17.20      .
        MOB   ULP                         0.9   18.00   18.12      .
        BPN   ULP                         2.0   18.00   18.20      .
        MOB   ULP                         0.6   18.00   19.00      .
        BPN   PRM                         2.0   19.00   19.17      .
        ESS   PRM                         3.0   19.00   20.10      .
        CTX   PRM                         1.8   21.08   21.20      .
        MOB   AVG                         1.7   22.00   22.07      .
        AMP   PRM                         5.0   16.06   17.04      .
        CTX   PRM                         3.5   17.08   18.00      .
        AMP   ULP                         2.0   18.00   18.12      .
        CTX   ULP                         2.0   18.00   18.21      .
        MOB   ULP                         0.5   18.00   19.00      .
        AMP   PRM                         3.0   19.00   19.20      .
        CTX   PRM                         2.7   19.00   20.06      .
        AMP   PRM                         3.0   21.08   22.00      .
        BPN   ADO                         3.0   16.00   16.18      .
        MOB   ADO                         2.0   17.06   17.18      .
        CTX   ADO                         2.0   18.00   18.10      .
        MOB   JET                         3.5   18.20   19.15      .
        MOB   LHO                         0.7   19.18   20.05      .
        ESS   LHO                         0.5   19.18   20.10      .
        BPN   LHO                         0.3   19.18   20.16      .
        BPN   ADO                         3.0   20.16   21.10      .
        MOB   ADO                         2.0   20.16   22.06      .
        MOB   IFO   F53   504           2.4   16.12   17.05      .
        AMP   ADO                         4.0   17.05   17.16      .
        MOB   ADO                         2.0   18.10   19.02      .
        ESS   ADO                         1.0   19.12   19.21      .
        CTX   ADO                         1.5   20.08   20.18      .
        MOB   ADO                         1.3   20.08   21.04      .
        ESS   ADO                         1.0   20.08   21.10      .
        AMP   ADO                         2.5   20.08   22.04      .
        CTX   IFO                         2.7   22.07   23.00      .
```

Figure C.12 Common Data: Network Definition File

```
DSPRT   921213-162748-SUN DSNAME M995092.TECH.DATA(APTSLINE)

TITLE  : HYPOTHETICAL REFINERY
PIPELINES            4.
PRODUCTS            15.
OFFTAKERS            9.
*
SUCTION              PL1      PL2      PL3      PL4      SOM
*
         NO.          -1.       2.       3.      -1.       4.
*
LINECLEARS           PL1      PL2      PL3      PL4      SOM
*
         PRM           1.                         1.
         CPR           1.                         1.
         ULP           2.                         2.
         SUP           2.                         2.
         AVG           2.                         2.
         ABL           2.                         2.
         JET                     1.                          1.
         JP4                     1.
         LHO                     1.
         ADO                     1.       1.
         NZA                     1.       1.
         TSA                     1.       1.
         IDO                              1.
         IFO                              1.
         LSF                              1.
*
OFFTAKERS            PL1      PL2      PL3      PL4      SOM
*
         MOB          -1.       2.       3.      -1.
         ESS          -1.       2.       3.      -1.
         CTX           1.       1.       1.       1.
         AMP           1.       2.       2.       1.
         BPA           1.       1.
         BPN           1.       1.       1.
         BPC           1.       1.
         SHL           1.       1.
         SOM                                                  1.
*
RATES                PL1      PL2      PL3      PL4      SOM
*
         MOB         175.     190.     160.     205.
         ESS         175.     190.     160.     205.
         CTX         175.     190.     160.     205.
         AMP         175.     190.     160.     205.
         BPA         175.     190.
         BPN         175.     190.     160.
         BPC         175.     190.
         SHL         175.     190.
         SOM                                                200.
```

Figure C.13 Common Data: Run Definition File

```
DSPRT   921213-162752-SUN DSNAME M995092.TECH.DATA(APTSUSER)

*
REPORTS                        (1=Y,0=N)
*
PIPELINE SYSTEM DATA              0.
JOB INPUT DATA                    0.
PIPELINE STATUS                   0.
JOB SCREENING                     0.
JOB QUEUES                        0.      (1=SORTED,2=UNSORTED+SORTED)
PATTERN GENERATION                0.
PATTERN QUEUES                    0.      (1=SORTED,2=UNSORTED+SORTED)
PIPELINE SCHEDULE                 0.      (1=BEST,2=BETTER,3=ALL)
JOB SUMMARY                       0.      (1=BEST,2=BETTER,3=ALL)
AVAILABILITY SUMMARY              0.      (1=BEST,2=BETTER,3=ALL)
PIPELINE UTILIZATION              0.
BACKTRACK SUMMARY                 0.
TARDINESS PROFILE                 0.
PERFORMANCE GRAPH                 0.
TARDINESS VS UTILIZ.              0.
VOL REMAIN VS UTILIZ.             0.
JOBS/VOLUME VS SAMPLE             0.
SIMULATION STATISTICS             1.
*
OBJ FUNCTION WEIGHTING
*
WEIGHTED TARDINESS                1.
SETUP COUNT                       0.
MAKESPAN                          0.
OFFTAKER OVERTIME                 0.
*
JOB PRIORITY WEIGHTING         ($/H)
*
HIGH                           1500.
MEDIUM                           40.
LOW                               1.
*
PATTERN PRIORITY WEIGHTING
*
VOLUME TRANSFERRED                0.              0.
PATTERN DURATION                  0.              0.
UTILIZATION                     100.            100.
GENERATION ORDER                  1.              1.
AVERAGE JOB VALUE                 0.              0.
*
JOB PREEMPTION TUNING         (HOURS)
*
SIBLING WORK REMAINING            4.              4.
JOB WORK REMAINING                6.              6.
MINIMUM WORK REMAINING            4.              4.
CRITICAL: DUE TIME               12.             12.
CRITICAL: DEADLINE               24.             24.
*
JOB PRIORITY ADJUSTMENT       (HOURS)
*
NEW JOB                           1.              1.
OLD JOB (ALREADY ONLINE)          4.              4.
NON-PREEMPTABLE JOB             500.            500.
CRITICAL JOB                    250.            250.
```

```
SIBLING JOB SWITCH          -250.          -250.
JOB AVERAGE CAPACITY           0.       (1=Y,0=N)
*
JOB SCHEDULING RULE SELECTION  (1=Y,0=N)
*
RANDOM                         0.
FIRST-COME-FIRST-SERVED        0.
SHORTEST PROC TIME             1.
LONGEST PROC TIME              0.
LONGEST OPERATION              0.
EARLIEST DUE DATE              0.
MODIFIED DUE DATE              0.
MINIMUM SLACK (DUE DATE)       0.
MINIMUM SLACK (DEADLINE)       0.
MODIFIED SLACK                 0.
APPARENT TARDINESS COST        0.
COST OVER TIME                 0.
WEIGHTED TARDINESS             0.
TARDINESS DELTA                0.
LOAD LEVELLING                 0.
*
JOB SCHEDULING RULE PARAMETERS
*
APPARENT TARD COST K           3.0            3.0
COST OVER TIME K               3.0            2.0
COST OVER TIME B               1.0            2.0
*
HEURISTIC SELECTION
*
PIPELINE IDLING                0.
LOOK AHEAD                     0.
*
BACKTRACKING PARAMETERS
*
MAX BACKTRACKING PASSES       50.
MAX CONSECUTIVE PASSES        10.
MAXIMUM TIME INCREMENT         1.
*
SIMULATION
*
SIMULATION RUN                 1.     (1=Y,0=N)
SAMPLE SIZE                   40.     (100 MAX)
SCHEDULE HORIZON (DAYS)        7.     (1<=HORIZON<=7)
SCHEDULE START DAY             1.     (0=RANDOM,1=TUE,2=WED,ETC)
PIPELINE LOAD                 80.     (0=RANDOM,>1=% LOAD)
READY TIMES                    1.     (0=NO,1=RANDOM)
DUE DATE TIGHTNESS           300.     (0=NO,1=RANDOM,>1=% TIGHTNESS)
DEADLINE TIGHTNESS             0.     (0=NO,1=RANDOM,>1=% TIGHTNESS)
RANDOM NUMBER SEED             5.     (65539.0)
REMARK: 22 NOV 92    SPT                DUE.T=300, LOAD=70, PAT UTZ
```

Appendix D INSTANCE EVALUATION RESULTS

Experiment	Description
1	Apparent Tardiness Cost Tuning
2	Cost Over Time Tuning
3	Scheduling Rule Evaluation
4	Pattern Selection Evaluation

Experiment 1 Average Apparent Tardiness Cost tuning

Average: 22/11/92 16:43	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
ATC k=2.0	31	1	6	31	9232	231	67	73.3	2.6	33	13	6	415	65
ATC k=2.5	31	1	6	31	9232	231	67	73.3	2.6	33	13	6	415	65
ATC k=3.0	31	1	7	31	7776	194	62	73.7	2.2	33	13	6	417	66
ATC k=3.5	31	1	7	31	7776	194	62	73.7	2.2	33	13	6	417	66
ATC k=4.0	31	1	4	31	7376	184	54	73.7	2.2	33	13	6	418	66
ATC k=4.5	31	1	4	31	7376	184	54	73.7	2.2	33	13	6	418	66
ATC k=5.0	31	1	6	31	9232	231	67	73.3	2.6	33	13	6	415	65
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	max
Best	1	4	31	7376	184	54	73.7	2.2	33	13	6	415	66
Worst	1	7	31	9232	231	67	73.3	2.6	33	13	6	418	65
Average	1	6	31	8286	207	62	73.5	2.3	33	13	6	416	65
Ratio of best													
ATC k=2.0	1.20	1.38	1.00	1.25	1.25	1.23	0.99	1.17	1.01	1.00	1.04	1.00	0.99
ATC k=2.5	1.20	1.38	1.00	1.25	1.25	1.23	0.99	1.17	1.01	1.00	1.04	1.00	0.99
ATC k=3.0	1.00	1.62	1.00	1.05	1.05	1.14	1.00	1.00	1.00	1.00	1.04	1.01	1.00
ATC k=3.5	1.00	1.62	1.00	1.05	1.05	1.14	1.00	1.00	1.00	1.00	1.04	1.01	1.00
ATC k=4.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.02	1.05	1.00	1.01	1.00
ATC k=4.5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.02	1.05	1.00	1.01	1.00
ATC k=5.0	1.20	1.38	1.00	1.25	1.25	1.23	0.99	1.17	1.01	1.00	1.04	1.00	0.99

Example 1: 22/11/92 16:43	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
ATC k=2.0	22	0	20	155	4200	105	59	68.1	0.0	23	15		382	57
ATC k=2.5	22	0	20	155	4200	105	59	68.1	0.0	23	15		382	57
ATC k=3.0	22	0	20	155	4200	105	59	68.1	0.0	23	15		382	57
ATC k=3.5	22	0	20	155	4200	105	59	68.1	0.0	23	15		382	57
ATC k=4.0	22	0	7	155	2720	68	20	68.1	0.0	25	18		384	57
ATC k=4.5	22	0	7	155	2720	68	20	68.1	0.0	25	18		384	57
ATC k=5.0	22	0	20	155	4200	105	59	68.1	0.0	23	15		382	57
Criteria	min	min	min	min	min	min	max	min	min	min	min		min	max
Best	0	7	155	2720	68	20	68.1	0.0	23	15			382	57
Worst	0	20	155	4200	105	59	68.1	0.0	25	18			384	57
Average	0	16	155	3777	94	48	68.1	0.0	24	16			383	57
Ratio of best														
ATC k=2.0	1.00	2.86	1.00	1.54	1.54	2.95	1.00	1.00	1.00	1.00			1.00	1.00
ATC k=2.5	1.00	2.86	1.00	1.54	1.54	2.95	1.00	1.00	1.00	1.00			1.00	1.00
ATC k=3.0	1.00	2.86	1.00	1.54	1.54	2.95	1.00	1.00	1.00	1.00			1.00	1.00
ATC k=3.5	1.00	2.86	1.00	1.54	1.54	2.95	1.00	1.00	1.00	1.00			1.00	1.00
ATC k=4.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.09	1.20			1.01	1.00
ATC k=4.5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.09	1.20			1.01	1.00
ATC k=5.0	1.00	2.86	1.00	1.54	1.54	2.95	1.00	1.00	1.00	1.00			1.00	1.00

Example 2: 22/11/92 16:43	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
ATC k=2.0	23	2	1		9920	248	61	66.9	3.0	26	3	6	385	77
ATC k=2.5	23	2	1		9920	248	61	66.9	3.0	26	3	6	385	77
ATC k=3.0	23	1	6		2800	70	37	68.8	1.1	25	3	6	396	79
ATC k=3.5	23	1	6		2800	70	37	68.8	1.1	25	3	6	396	79
ATC k=4.0	23	1	6		2800	70	37	68.8	1.1	25	3	6	396	79
ATC k=4.5	23	1	6		2800	70	37	68.8	1.1	25	3	6	396	79
ATC k=5.0	23	2	1		9920	248	61	66.9	3.0	26	3	6	385	77
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	max	
Best	1	1		2800	70	37	68.8	1.1	25	3	6	385	79	
Worst	2	6		9920	248	61	66.9	3.0	26	3	6	396	77	
Average	1	4		5851	146	47	68.0	1.9	25	3	6	391	78	
Ratio of best														
ATC k=2.0	2.00	1.00		3.54	3.54	1.65	0.97	2.73	1.04	1.00	1.00	1.00	0.97	
ATC k=2.5	2.00	1.00		3.54	3.54	1.65	0.97	2.73	1.04	1.00	1.00	1.00	0.97	
ATC k=3.0	1.00	6.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.03	1.00	
ATC k=3.5	1.00	6.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.03	1.00	
ATC k=4.0	1.00	6.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.03	1.00	
ATC k=4.5	1.00	6.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.03	1.00	
ATC k=5.0	2.00	1.00		3.54	3.54	1.65	0.97	2.73	1.04	1.00	1.00	1.00	0.97	

Example 3: 22/11/92 16:43	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
ATC k=2.0	31	2	3		10440	261	52	79.2	4.3	33	13	7	454	65
ATC k=2.5	31	2	3		10440	261	52	79.2	4.3	33	13	7	454	65
ATC k=3.0	31	2	3		10440	261	52	79.2	4.3	33	13	7	454	65
ATC k=3.5	31	2	3		10440	261	52	79.2	4.3	33	13	7	454	65
ATC k=4.0	31	2	3		9680	242	52	79.2	4.3	33	13	7	454	65
ATC k=4.5	31	2	3		9680	242	52	79.2	4.3	33	13	7	454	65
ATC k=5.0	31	2	3		10440	261	52	79.2	4.3	33	13	7	454	65
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	max	
Best	2	3		9680	242	52	79.2	4.3	33	13	7	454	65	
Worst	2	3		10440	261	52	79.2	4.3	33	13	7	454	65	
Average	2	3		10223	256	52	79.2	4.3	33	13	7	454	65	
Ratio of best														
ATC k=2.0	1.00	1.00		1.08	1.08	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
ATC k=2.5	1.00	1.00		1.08	1.08	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
ATC k=3.0	1.00	1.00		1.08	1.08	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
ATC k=3.5	1.00	1.00		1.08	1.08	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
ATC k=4.0	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
ATC k=4.5	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
ATC k=5.0	1.00	1.00		1.08	1.08	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	

Experiment 1 Example 4 Apparent Tardiness Cost tuning

Example 4: 22/11/92 16:43	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
ATC k=2.0	41	1	1		4640	116	38	78.3	3.0	41	15	4	438	65
ATC k=2.5	41	1	1		4640	116	38	78.3	3.0	41	15	4	438	65
ATC k=3.0	41	1	1		4480	112	38	78.3	3.0	41	15	4	438	65
ATC k=3.5	41	1	1		4480	112	38	78.3	3.0	41	15	4	438	65
ATC k=4.0	41	1	1		4720	118	38	78.3	3.0	42	15	3	439	65
ATC k=4.5	41	1	1		4720	118	38	78.3	3.0	42	15	3	439	65
ATC k=5.0	41	1	1		4640	116	38	78.3	3.0	41	15	4	438	65
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	1		4480	112	38	78.3	3.0	41	15	3	438	65
Worst		1	1		4720	118	38	78.3	3.0	42	15	4	439	65
Average		1	1		4617	115	38	78.3	3.0	41	15	4	438	65
Ratio of best														
ATC k=2.0		1.00	1.00		1.04	1.04	1.00	1.00	1.00	1.00	1.00	1.33	1.00	1.00
ATC k=2.5		1.00	1.00		1.04	1.04	1.00	1.00	1.00	1.00	1.00	1.33	1.00	1.00
ATC k=3.0		1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.33	1.00	1.00
ATC k=3.5		1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.33	1.00	1.00
ATC k=4.0		1.00	1.00		1.05	1.05	1.00	1.00	1.00	1.02	1.00	1.00	1.00	1.00
ATC k=4.5		1.00	1.00		1.05	1.05	1.00	1.00	1.00	1.02	1.00	1.00	1.00	1.00
ATC k=5.0		1.00	1.00		1.04	1.04	1.00	1.00	1.00	1.00	1.00	1.33	1.00	1.00

Example 5: 22/11/92 16:43	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
ATC k=2.0	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
ATC k=2.5	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
ATC k=3.0	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
ATC k=3.5	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
ATC k=4.0	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
ATC k=4.5	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
ATC k=5.0	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	max
Best	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
Worst	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
Average	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
Ratio of best														
ATC k=2.0	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ATC k=2.5	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ATC k=3.0	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ATC k=3.5	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ATC k=4.0	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ATC k=4.5	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ATC k=5.0	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Experiment 2 Average Cost Over Time tuning

Average: 22/11/92 17:05	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
COT k*b=1.0	31	1	1	31	4952	124	49	73.4	2.5	33	12	5	416	65
COT k*b=2.0	31	1	4	31	5648	141	56	72.7	3.1	32	12	5	413	65
COT k*b=3.0	31	1	10	31	6280	157	61	72.5	3.4	33	13	5	411	65
COT k*b=4.0	31	1	12	31	5312	133	56	73.3	2.5	33	13	5	415	65
COT k*b=5.0	31	1	12	31	4920	123	44	73.4	2.5	33	12	6	416	66
COT k*b=6.0	31	1	12	31	5232	131	47	73.4	2.4	33	13	6	416	65
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	max
Best	1	1	31	4920	123	44	73.4	2.4	32	12	5	411	66
Worst	1	12	31	6280	157	61	72.5	3.4	33	13	6	416	65
Average	1	8	31	5391	135	52	73.1	2.7	33	13	5	415	65
Ratio of best														
COT k*b=1.0	1.20	1.00	1.00	1.01	1.01	1.11	1.00	1.01	1.02	1.05	1.00	1.01	0.99
COT k*b=2.0	1.00	3.17	1.00	1.15	1.15	1.28	0.99	1.27	1.00	1.05	1.00	1.00	0.99
COT k*b=3.0	1.20	8.17	1.01	1.28	1.28	1.40	0.99	1.38	1.02	1.12	1.00	1.00	0.98
COT k*b=4.0	1.00	10.00	1.00	1.08	1.08	1.27	1.00	1.04	1.03	1.07	1.08	1.01	0.99
COT k*b=5.0	1.20	9.83	1.00	1.00	1.00	1.00	1.00	1.02	1.03	1.00	1.20	1.01	1.00
COT k*b=6.0	1.20	10.17	1.00	1.06	1.06	1.07	1.00	1.00	1.04	1.07	1.28	1.01	1.00

Example 1: 22/11/92 17:05	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
COT k*b=1.0	22	0	1	155	1920	48	18	68.1	0.0	23	17		380	56
COT k*b=2.0	22	0	1	155	1920	48	18	68.1	0.0	23	17		380	56
COT k*b=3.0	22	0	8	157	4840	121	43	68.1	0.0	25	18		382	56
COT k*b=4.0	22	0	12	155	1920	48	16	68.1	0.0	24	16		381	56
COT k*b=5.0	22	0	17	155	2080	52	16	68.1	0.0	25	15		382	57
COT k*b=6.0	22	0	17	155	2080	52	16	68.1	0.0	25	15		382	57
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	min	max
Best	0	1	155	1920	48	16	68.1	0.0	23	15		380	57
Worst	0	17	157	4840	121	43	68.1	0.0	25	18		382	56
Average	0	9	155	2460	62	21	68.1	0.0	24	16		381	56
Ratio of best														
COT k*b=1.0	1.00	1.00	1.00	1.00	1.00	1.13	1.00	1.00	1.00	1.13		1.00	0.98
COT k*b=2.0	1.00	1.00	1.00	1.00	1.00	1.13	1.00	1.00	1.00	1.13		1.00	0.98
COT k*b=3.0	1.00	8.00	1.01	2.52	2.52	2.69	1.00	1.00	1.09	1.20		1.01	0.98
COT k*b=4.0	1.00	12.00	1.00	1.00	1.00	1.00	1.00	1.00	1.04	1.07		1.00	0.98
COT k*b=5.0	1.00	17.00	1.00	1.08	1.08	1.00	1.00	1.00	1.09	1.00		1.01	1.00
COT k*b=6.0	1.00	17.00	1.00	1.08	1.08	1.00	1.00	1.00	1.09	1.00		1.01	1.00

Example 2: 22/11/92 17:05	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
COT k*b=1.0	23	2	2		5480	137	55	66.8	3.1	26	5	4	389	77
COT k*b=2.0	23	1	2		3400	85	55	67.2	2.7	25	5	10	391	78
COT k*b=3.0	23	1	2		3400	85	55	67.2	2.7	25	5	10	391	78
COT k*b=4.0	23	1	2		3480	87	55	67.2	2.7	26	5	10	390	78
COT k*b=5.0	23	1	1		2240	56	19	68.8	1.1	26	3	8	398	80
COT k*b=6.0	23	1	7		2200	55	38	68.8	1.1	26	3	8	397	79
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	max	
Best	1	1		2200	55	19	68.8	1.1	25	3	4	389	80	
Worst	2	7		5480	137	55	66.8	3.1	26	5	10	398	77	
Average	1	3		3367	84	46	67.7	2.2	26	4	8	393	78	
Ratio of best														
COT k*b=1.0	2.00	2.00		2.49	2.49	2.89	0.97	2.82	1.04	1.67	1.00	1.00	0.96	
COT k*b=2.0	1.00	2.00		1.55	1.55	2.89	0.98	2.45	1.00	1.67	2.50	1.01	0.98	
COT k*b=3.0	1.00	2.00		1.55	1.55	2.89	0.98	2.45	1.00	1.67	2.50	1.01	0.98	
COT k*b=4.0	1.00	2.00		1.58	1.58	2.89	0.98	2.45	1.04	1.67	2.50	1.00	0.98	
COT k*b=5.0	1.00	1.00		1.02	1.02	1.00	1.00	1.00	1.04	1.00	2.00	1.02	1.00	
COT k*b=6.0	1.00	7.00		1.00	1.00	2.00	1.00	1.00	1.04	1.00	2.00	1.02	0.99	

Example 3: 22/11/92 17:05	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
COT k*b=1.0	31	2	1		4360	109	47	78.9	4.5	31	10	10	453	65
COT k*b=2.0	31	2	1		4320	108	47	78.9	4.5	31	10	10	453	65
COT k*b=3.0	31	2	9		4360	109	47	78.9	4.5	31	12	10	452	65
COT k*b=4.0	31	2	10		4120	103	47	78.9	4.5	32	10	9	454	65
COT k*b=5.0	31	2	5		4440	111	47	79.1	4.4	33	9	10	453	65
COT k*b=6.0	31	2	1		6040	151	43	79.3	4.2	35	13	12	455	65
Criteria	min	min	min		min	min	min	max	min	min	min	min	min	max
Best	2	1			4120	103	43	79.3	4.2	31	9	9	452	65
Worst	2	10			6040	151	47	78.9	4.5	35	13	12	455	65
Average	2	5			4607	115	46	79.0	4.4	32	11	10	453	65
Ratio of best														
COT k*b=1.0	1.00	1.00			1.06	1.06	1.09	0.99	1.07	1.00	1.11	1.11	1.00	1.00
COT k*b=2.0	1.00	1.00			1.05	1.05	1.09	0.99	1.07	1.00	1.11	1.11	1.00	1.00
COT k*b=3.0	1.00	9.00			1.06	1.06	1.09	0.99	1.07	1.00	1.33	1.11	1.00	1.00
COT k*b=4.0	1.00	10.00			1.00	1.00	1.09	0.99	1.07	1.03	1.11	1.00	1.00	1.00
COT k*b=5.0	1.00	5.00			1.08	1.08	1.09	1.00	1.05	1.06	1.00	1.11	1.00	1.00
COT k*b=6.0	1.00	1.00			1.47	1.47	1.00	1.00	1.00	1.13	1.44	1.33	1.01	1.00

Experiment 2 Example 4 Cost Over Time tuning

Example 4: 22/11/92 17:05	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
COT k*b=1.0	41	1	1		3080	77	37	78.3	3.0	43	15		438	65
COT k*b=2.0	41	1	8		2280	57	37	78.3	3.0	41	15	2	440	66
COT k*b=3.0	41	1	8		2280	57	37	78.3	3.0	41	15	2	440	66
COT k*b=4.0	41	1	8		2280	57	37	78.3	3.0	41	15	2	440	66
COT k*b=5.0	41	1	8		2280	57	37	78.3	3.0	41	15	2	440	66
COT k*b=6.0	41	1	8		2280	57	37	78.3	3.0	41	15	2	440	66
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	1		2280	57	37	78.3	3.0	41	15	2	438	66
Worst		1	8		3080	77	37	78.3	3.0	43	15	2	440	65
Average		1	7		2413	60	37	78.3	3.0	41	15	2	440	66
Ratio of best														
COT k*b=1.0		1.00	1.00		1.35	1.35	1.00	1.00	1.00	1.05	1.00		1.00	0.98
COT k*b=2.0		1.00	8.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
COT k*b=3.0		1.00	8.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
COT k*b=4.0		1.00	8.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
COT k*b=5.0		1.00	8.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
COT k*b=6.0		1.00	8.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Example 5: 22/11/92 17:05	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
COT k*b=1.0	37	1	1		9920	248	86	74.8	1.7	40	15	11	420	62
COT k*b=2.0	37	1	7		16320	408	124	71.2	5.3	40	15	3	400	59
COT k*b=3.0	37	2	22		16520	413	124	69.9	6.6	41	16	3	391	58
COT k*b=4.0	37	1	28		14760	369	124	74.0	2.5	41	17	6	412	61
COT k*b=5.0	37	2	28		13560	339	100	72.5	3.9	40	17	10	407	60
COT k*b=6.0	37	2	28		13560	339	100	72.5	3.9	40	17	10	407	60
Criteria	min	min	min	min	min	min	max	min	min	min	min	min	max	
Best	1	1		9920	248	86	74.8	1.7	40	15	3	391	62	
Worst	2	28		16520	413	124	69.9	6.6	41	17	11	420	58	
Average	2	19		14107	353	110	72.5	4.0	40	16	7	406	60	
Ratio of best														
COT k*b=1.0	1.00	1.00		1.00	1.00	1.00	1.00	1.00	1.00	1.00	3.67	1.07	1.00	
COT k*b=2.0	1.00	7.00		1.65	1.65	1.44	0.95	3.12	1.00	1.00	1.00	1.02	0.95	
COT k*b=3.0	2.00	22.00		1.67	1.67	1.44	0.93	3.88	1.03	1.07	1.00	1.00	0.94	
COT k*b=4.0	1.00	28.00		1.49	1.49	1.44	0.99	1.47	1.03	1.13	2.00	1.05	0.98	
COT k*b=5.0	2.00	28.00		1.37	1.37	1.16	0.97	2.29	1.00	1.13	3.33	1.04	0.97	
COT k*b=6.0	2.00	28.00		1.37	1.37	1.16	0.97	2.29	1.00	1.13	3.33	1.04	0.97	

Experiment 3 Average Scheduling Rule Evaluation

Average: 22/11/92 17:19	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	31	2	12	31	9552	239	65	73.4	2.5	36	13	6	413	65
FCFS	31	1	17	31	8920	223	55	72.8	3.1	32	12	6	411	64
SPT	31	1	8	31	6928	173	54	73.7	2.2	33	13	6	417	66
LPT	31	2	13	31	11928	298	80	72.3	3.6	33	12	3	409	64
LOP	31	2	20	32	12008	300	69	72.5	3.3	34	12	1	411	64
EDD	31	2	9	31	5672	142	47	73.5	2.3	34	11	5	416	65
SLACK	31	1	6	31	7296	182	63	73.4	2.5	34	14	6	416	65
ATC k=4.0	31	1	4	31	7376	184	54	73.7	2.2	33	13	6	418	66
COT k*b=5.0	31	1	12	31	4920	123	44	73.4	2.5	33	12	6	416	65
WTAR	31	1	11	31	6768	169	47	73.7	2.1	35	14	6	416	65
LOAD	31	2	12	31	8824	221	55	73.3	2.6	35	13	4	415	65
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	4	31	4920	123	44	73.7	2.1	32	11	1	409	66
Worst		2	20	32	12008	300	80	72.3	3.6	36	14	6	418	64
Average		1	11	31	8199	205	58	73.2	2.6	34	13	5	414	65
Ratio of best														
RANDOM		1.6	2.8	1.0	1.9	1.9	1.5	1.00	1.2	1.1	1.2	5.3	1.01	0.98
FCFS		1.2	4.1	1.0	1.8	1.8	1.3	0.99	1.4	1.0	1.0	5.2	1.01	0.98
SPT		1.0	2.0	1.0	1.4	1.4	1.2	1.00	1.0	1.0	1.2	4.8	1.02	1.00
LPT		1.8	3.2	1.0	2.4	2.4	1.8	0.98	1.7	1.0	1.1	2.2	1.00	0.97
LOP		1.6	4.8	1.0	2.4	2.4	1.6	0.98	1.6	1.0	1.0	1.0	1.00	0.98
EDD		1.6	2.1	1.0	1.2	1.2	1.1	1.00	1.1	1.0	1.0	4.2	1.02	0.99
SLACK		1.4	1.5	1.0	1.5	1.5	1.4	0.99	1.2	1.1	1.2	5.3	1.02	0.99
ATC k=4.0		1.0	1.0	1.0	1.5	1.5	1.2	1.00	1.0	1.0	1.2	4.7	1.02	1.00
COT k*b=5.0		1.0	2.8	1.0	1.0	1.0	1.0	1.00	1.2	1.0	1.0	5.0	1.02	1.00
WTAR		1.0	2.6	1.0	1.4	1.4	1.1	1.00	1.0	1.1	1.2	4.7	1.02	1.00
LOAD		1.6	2.9	1.0	1.8	1.8	1.3	0.99	1.2	1.1	1.1	3.7	1.02	0.99

Example 1: 22/11/92 17:19	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	22	0	21	155	6480	162	59	68.1	0.0	24	17	2	381	56
FCFS	22	0	40	155	3960	99	23	68.1	0.0	24	17	4	385	56
SPT	22	0	7	155	2720	68	20	68.1	0.0	23	18		384	57
LPT	22	0	22	157	9960	249	64	68.1	0.0	23	18		383	56
LOP	22	0	25	159	10760	269	63	68.1	0.0	24	16		385	56
EDD	22	0	5	155	3560	89	43	68.1	0.0	24	16	1	380	56
SLACK	22	0	11	155	5600	140	59	68.1	0.0	23	19	2	381	56
ATC k=4.0	22	0	7	155	2720	68	20	68.1	0.0	25	18		384	57
COT k*b=5.0	22	0	17	155	2080	52	16	68.1	0.0	25	15		381	56
WTAR	22	0	36	157	6040	151	61	68.1	0.0	23	20		381	56
LOAD	22	0	23	155	2800	70	30	68.1	0.0	24	17		379	56
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		0	5	155	2080	52	16	68.1	0.0	23	15		379	57
Worst		0	40	159	10760	269	64	68.1	0.0	25	20	4	385	56
Average		0	19	156	5153	129	42	68.1	0.0	24	17	1	382	56
Ratio of best														
RANDOM		1.0	4.2	1.0	3.1	3.1	3.7	1.00	1.0	1.0	1.1		1.01	0.98
FCFS		1.0	8.0	1.0	1.9	1.9	1.4	1.00	1.0	1.0	1.1		1.02	0.98
SPT		1.0	1.4	1.0	1.3	1.3	1.3	1.00	1.0	1.0	1.2		1.01	1.00
LPT		1.0	4.4	1.0	4.8	4.8	4.0	1.00	1.0	1.0	1.2		1.01	0.98
LOP		1.0	5.0	1.0	5.2	5.2	3.9	1.00	1.0	1.0	1.1		1.02	0.98
EDD		1.0	1.0	1.0	1.7	1.7	2.7	1.00	1.0	1.0	1.1		1.00	0.98
SLACK		1.0	2.2	1.0	2.7	2.7	3.7	1.00	1.0	1.0	1.3		1.01	0.98
ATC k=4.0		1.0	1.4	1.0	1.3	1.3	1.3	1.00	1.0	1.1	1.2		1.01	1.00
COT k*b=5.0		1.0	3.4	1.0	1.0	1.0	1.0	1.00	1.0	1.1	1.0		1.01	0.98
WTAR		1.0	7.2	1.0	2.9	2.9	3.8	1.00	1.0	1.0	1.3		1.01	0.98
LOAD		1.0	4.6	1.0	1.3	1.3	1.9	1.00	1.0	1.0	1.1		1.00	0.98

Example 2: 22/11/92 17:19	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	23	2	1		10600	265	61	67.1	2.8	26	3	5	388	77
FCFS	23	2	1		10280	257	61	65.9	4.0	23	3	8	379	76
SPT	23	1	7		2240	56	37	68.8	1.1	26	3	7	395	79
LPT	23	1	1		11520	288	75	66.9	3.0	24	3	2	386	77
LOP	23	1	1		11520	288	75	66.9	3.0	24	3	2	386	77
EDD	23	2	2		3480	87	55	66.8	3.1	26	5	6	388	77
SLACK	23	2	2		3240	81	55	66.8	3.1	26	5	5	387	77
ATC k=4.0	23	1	6		2800	70	37	68.8	1.1	25	3	6	396	79
COT k*b=5.0	23	1	1		2240	56	19	68.8	1.1	26	3	8	398	80
WTAR	23	1	1		3240	81	21	68.8	1.1	26	3	8	396	79
LOAD	23	1	14		5760	144	43	68.8	1.1	29	2	9	397	79
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	1		2240	56	19	68.8	1.1	23	2	2	379	80
Worst		2	14		11520	288	75	65.9	4.0	29	5	9	398	76
Average		1	3		6084	152	49	67.7	2.2	26	3	6	391	78
Ratio of best														
RANDOM		2.0	1.0		4.7	4.7	3.2	0.98	2.5	1.1	1.5	2.5	1.02	0.96
FCFS		2.0	1.0		4.6	4.6	3.2	0.96	3.6	1.0	1.5	4.0	1.00	0.95
SPT		1.0	7.0		1.0	1.0	1.9	1.00	1.0	1.1	1.5	3.5	1.04	0.99
LPT		1.0	1.0		5.1	5.1	3.9	0.97	2.7	1.0	1.5	1.0	1.02	0.96
LOP		1.0	1.0		5.1	5.1	3.9	0.97	2.7	1.0	1.5	1.0	1.02	0.96
EDD		2.0	2.0		1.6	1.6	2.9	0.97	2.8	1.1	2.5	3.0	1.02	0.96
SLACK		2.0	2.0		1.4	1.4	2.9	0.97	2.8	1.1	2.5	2.5	1.02	0.96
ATC k=4.0		1.0	6.0		1.3	1.3	1.9	1.00	1.0	1.1	1.5	3.0	1.04	0.99
COT k*b=5.0		1.0	1.0		1.0	1.0	1.0	1.00	1.0	1.1	1.5	4.0	1.05	1.00
WTAR		1.0	1.0		1.4	1.4	1.1	1.00	1.0	1.1	1.5	4.0	1.04	0.99
LOAD		1.0	14.0		2.6	2.6	2.3	1.00	1.0	1.3	1.0	4.5	1.05	0.99

Example 3: 22/11/92 17:19	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	31	3	14		13120	328	90	79.6	3.9	41	15	9	451	65
FCFS	31	2	1		9400	235	76	77.2	6.3	32	10	9	439	63
SPT	31	2	3		9080	227	52	79.2	4.3	33	13	7	454	65
LPT	31	4	24		14440	361	75	76.5	7.0	37	14	2	435	62
LOP	31	4	39		14080	352	75	76.9	6.6	37	15	2	437	63
EDD	31	4	1		4360	109	47	80.0	3.5	35	10	9	458	66
SLACK	31	3	7		7960	199	82	80.1	3.4	38	13	9	459	66
ATC k=4.0	31	2	3		9680	242	52	79.2	4.3	33	13	7	454	65
COT k*b=5.0	31	2	5		4440	111	47	79.1	4.4	33	9	10	453	65
WTAR	31	2	1		5600	140	39	80.0	3.5	37	12	8	457	66
LOAD	31	3	8		11080	277	60	79.6	3.9	37	13	3	454	65
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		2	1		4360	109	39	80.1	3.4	32	9	2	435	66
Worst		4	39		14440	361	90	76.5	7.0	41	15	10	459	62
Average		3	10		9385	235	63	78.9	4.6	36	12	7	450	65
Ratio of best														
RANDOM		1.5	14.0		3.0	3.0	2.3	0.99	1.1	1.3	1.7	4.5	1.04	0.98
FCFS		1.0	1.0		2.2	2.2	1.9	0.96	1.9	1.0	1.1	4.5	1.01	0.95
SPT		1.0	3.0		2.1	2.1	1.3	0.99	1.3	1.0	1.4	3.5	1.04	0.98
LPT		2.0	24.0		3.3	3.3	1.9	0.96	2.1	1.2	1.6	1.0	1.00	0.94
LOP		2.0	39.0		3.2	3.2	1.9	0.96	1.9	1.2	1.7	1.0	1.00	0.95
EDD		2.0	1.0		1.0	1.0	1.2	1.00	1.0	1.1	1.1	4.5	1.05	1.00
SLACK		1.5	7.0		1.8	1.8	2.1	1.00	1.0	1.2	1.4	4.5	1.06	1.00
ATC k=4.0		1.0	3.0		2.2	2.2	1.3	0.99	1.3	1.0	1.4	3.5	1.04	0.98
COT k*b=5.0		1.0	5.0		1.0	1.0	1.2	0.99	1.3	1.0	1.0	5.0	1.04	0.98
WTAR		1.0	1.0		1.3	1.3	1.0	1.00	1.0	1.2	1.3	4.0	1.05	1.00
LOAD		1.5	8.0		2.5	2.5	1.5	0.99	1.1	1.2	1.4	1.5	1.04	0.98

Example 4: 22/11/92 17:19	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	41	1	10		3600	90	38	79.2	2.1	45	15	8	439	65
FCFS	41	1	22		5720	143	38	78.3	3.0	42	14	2	435	64
SPT	41	1	21		3640	91	37	78.3	3.0	42	15	3	437	65
LPT	41	2	8		7080	177	87	76.6	4.7	42	13	5	428	63
LOP	41	1	16		3600	90	34	78.3	3.0	43	10	2	436	65
EDD	41	1	23		720	18	11	78.3	3.0	42	11	2	436	65
SLACK	41	1	10		4320	108	41	78.3	3.0	43	15	8	439	65
ATC k=4.0	41	1	1		4720	118	38	78.3	3.0	42	15	3	439	65
COT k*b=5.0	41	1	8		2280	57	37	78.3	3.0	41	15	2	440	66
WTAR	41	1	10		4360	109	44	78.3	3.0	44	16	10	436	65
LOAD	41	1	13		5880	147	41	78.3	3.0	44	17	10	440	65
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	1		720	18	11	79.2	2.1	41	10	2	428	66
Worst		2	23		7080	177	87	76.6	4.7	45	17	10	440	63
Average		1	13		4175	104	41	78.2	3.1	43	14	5	437	65
Ratio of best														
RANDOM		1.0	10.0		5.0	5.0	3.5	1.00	1.0	1.1	1.5	4.0	1.03	0.98
FCFS		1.0	22.0		7.9	7.9	3.5	0.99	1.4	1.0	1.4	1.0	1.02	0.97
SPT		1.0	21.0		5.1	5.1	3.4	0.99	1.4	1.0	1.5	1.5	1.02	0.98
LPT		2.0	8.0		9.8	9.8	7.9	0.97	2.2	1.0	1.3	2.5	1.00	0.95
LOP		1.0	16.0		5.0	5.0	3.1	0.99	1.4	1.0	1.0	1.0	1.02	0.98
EDD		1.0	23.0		1.0	1.0	1.0	0.99	1.4	1.0	1.1	1.0	1.02	0.98
SLACK		1.0	10.0		6.0	6.0	3.7	0.99	1.4	1.0	1.5	4.0	1.03	0.98
ATC k=4.0		1.0	1.0		6.6	6.6	3.5	0.99	1.4	1.0	1.5	1.5	1.03	0.98
COT k*b=5.0		1.0	8.0		3.2	3.2	3.4	0.99	1.4	1.0	1.5	1.0	1.03	1.00
WTAR		1.0	10.0		6.1	6.1	4.0	0.99	1.4	1.1	1.6	5.0	1.02	0.98
LOAD		1.0	13.0		8.2	8.2	3.7	0.99	1.4	1.1	1.7	5.0	1.03	0.98

Example 5: 22/11/92 17:19	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	37	2	12		13960	349	79	73.0	3.5	44	16	8	405	60
FCFS	37	1	23		15240	381	79	74.5	2.0	40	15	8	419	62
SPT	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
LPT	37	2	12		16640	416	100	73.2	3.3	41	13	4	411	60
LOP	37	2	20		20080	502	100	72.5	4.0	41	15		409	59
EDD	37	1	13		16240	406	79	74.5	2.0	42	15	7	418	61
SLACK	37	1	2		15360	384	76	73.5	3.0	40	17	8	414	60
ATC k=4.0	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
COT k*b=5.0	37	1	28		13560	339	100	72.6	3.9	40	17	10	407	60
WTAR	37	1	6		14600	365	71	73.5	3.0	43	17	2	412	61
LOAD	37	3	3		18600	465	100	71.6	4.9	40	15		406	59
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	2		13560	339	71	74.5	2.0	40	13		405	62
Worst		3	28		20080	502	124	71.6	4.9	44	17	12	419	59
Average		1	12		16200	405	94	73.4	3.1	41	16	6	412	61
Ratio of best														
RANDOM		2.0	6.0		1.0	1.0	1.1	0.98	1.8	1.1	1.2		1.00	0.97
FCFS		1.0	11.5		1.1	1.1	1.1	1.00	1.0	1.0	1.2		1.03	1.00
SPT		1.0	2.0		1.3	1.3	1.7	0.99	1.3	1.0	1.3		1.02	1.00
LPT		2.0	6.0		1.2	1.2	1.4	0.98	1.7	1.0	1.0		1.01	0.97
LOP		2.0	10.0		1.5	1.5	1.4	0.97	2.0	1.0	1.2		1.01	0.95
EDD		1.0	6.5		1.2	1.2	1.1	1.00	1.0	1.1	1.2		1.03	0.98
SLACK		1.0	1.0		1.1	1.1	1.1	0.99	1.5	1.0	1.3		1.02	0.97
ATC k=4.0		1.0	2.0		1.3	1.3	1.7	0.99	1.3	1.0	1.3		1.02	1.00
COT k*b=5.0		1.0	14.0		1.0	1.0	1.4	0.97	2.0	1.0	1.3		1.00	0.97
WTAR		1.0	3.0		1.1	1.1	1.0	0.99	1.5	1.1	1.3		1.02	0.98
LOAD		3.0	1.5		1.4	1.4	1.4	0.96	2.5	1.0	1.2		1.00	0.95

Experiment 4 Average Pattern Selection Evaluation

Average: 22/11/92 17:33	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	31	2	12	31	9552	239	65	73.4	2.5	36	13	6	413	65
RANDOM*	31	1	16	31	9176	229	64	73.7	2.2	37	14	6	415	56
SPT	31	1	8	31	6928	173	54	73.7	2.2	33	13	6	417	66
SPT*	31	1	6	31	7032	176	54	73.9	2.0	34	12	6	417	66
EDD	31	2	9	31	5672	142	47	73.5	2.3	34	11	5	416	65
EDD*	31	2	13	31	6040	151	45	73.3	2.5	34	12	4	415	65
COT k*b=4.0	31	1	12	31	5312	133	56	73.3	2.5	33	13	5	415	65
COT k*b=4.0*	31	2	4	31	5800	145	56	72.7	3.2	34	13	5	412	65
WTAR	31	1	11	31	6768	169	47	73.7	2.1	35	14	6	416	65
WTAR*	31	1	9	65	6208	155	48	74.6	1.3	37	13	6	421	66
LOAD	31	2	12	31	8824	221	55	73.3	2.6	35	13	4	415	65
LOAD*	31	2	10	31	8872	222	53	73.5	2.3	36	12	4	416	65
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	4	31	5312	133	45	74.6	1.3	33	11	4	412	66
Worst		2	16	65	9552	239	65	72.7	3.2	37	14	6	421	56
Average		1	10	34	7182	180	54	73.6	2.3	35	13	5	416	64
Ratio of best														
RANDOM		1.60	2.90	1.00	1.80	1.80	1.45	0.98	1.92	1.10	1.16	1.78	1.00	0.98
RANDOM*		1.40	4.10	1.00	1.73	1.73	1.42	0.99	1.70	1.12	1.19	1.67	1.01	0.85
SPT		1.00	2.10	1.00	1.30	1.30	1.19	0.99	1.70	1.01	1.16	1.61	1.01	1.00
SPT*		1.00	1.50	1.00	1.32	1.32	1.19	0.99	1.53	1.04	1.09	1.78	1.01	1.00
EDD		1.60	2.20	1.00	1.07	1.07	1.04	0.99	1.81	1.03	1.00	1.39	1.01	0.99
EDD*		1.60	3.25	1.00	1.14	1.14	1.00	0.98	1.98	1.05	1.05	1.00	1.01	0.98
COT k*b=4.0		1.00	3.00	1.00	1.00	1.00	1.23	0.98	1.98	1.00	1.11	1.50	1.01	0.99
COT k*b=4.0*		1.60	1.00	1.00	1.09	1.09	1.23	0.97	2.48	1.05	1.11	1.50	1.00	0.98
WTAR		1.00	2.70	1.01	1.27	1.27	1.04	0.99	1.66	1.05	1.19	1.56	1.01	0.99
WTAR*		1.00	2.25	2.09	1.17	1.17	1.07	1.00	1.00	1.13	1.12	1.78	1.02	1.00
LOAD		1.60	3.05	1.00	1.66	1.66	1.21	0.98	2.02	1.06	1.12	1.22	1.01	0.98
LOAD*		1.60	2.60	1.00	1.67	1.67	1.17	0.99	1.81	1.11	1.07	1.11	1.01	0.99

Example 1: 22/11/92 17:33	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	22	0	21	155	6480	162	59	68.1		24	17	2	381	56
RANDOM*	22	0	6	155	7200	180	59	68.1		31	22		379	56
SPT	22	0	7	155	2720	68	20	68.1		23	18		384	57
SPT*	22	0	5	155	2640	66	18	68.1		25	14		382	57
EDD	22	0	5	155	3560	89	43	68.1		24	16	1	380	56
EDD*	22	0	5	155	3560	89	43	68.1		24	16	1	380	56
COT k*b=4.0	22	0	12	155	1920	48	16	68.1		24	16		381	56
COT k*b=4.0*	22	0	1	155	2360	59	20	68.1		24	17		384	57
WTAR	22	0	36	157	6040	151	61	68.1		23	20		381	56
WTAR*	22	0	24	156	5760	144	60	68.1		24	20		379	56
LOAD	22	0	23	155	2800	70	30	68.1		24	17		379	56
LOAD*	22	0	14	155	4600	115	22	68.1		26	16		381	56
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		0	1	155	1920	48	16	68.1	ERR	23	14		379	57
Worst		0	36	157	7200	180	61	68.1	ERR	31	22	2	384	56
Average		0	13	155	4137	103	38	68.1	ERR	25	17	0	381	56
Ratio of best									ERR					
RANDOM		1.00	21.00	1.00	3.38	3.38	3.69	1.00	ERR	1.04	1.21		1.01	0.98
RANDOM*		1.00	6.00	1.00	3.75	3.75	3.69	1.00	ERR	1.35	1.57		1.00	0.98
SPT		1.00	7.00	1.00	1.42	1.42	1.25	1.00	ERR	1.00	1.29		1.01	1.00
SPT*		1.00	5.00	1.00	1.38	1.38	1.13	1.00	ERR	1.09	1.00		1.01	1.00
EDD		1.00	5.00	1.00	1.85	1.85	2.69	1.00	ERR	1.04	1.14		1.00	0.98
EDD*		1.00	5.00	1.00	1.85	1.85	2.69	1.00	ERR	1.04	1.14		1.00	0.98
COT k*b=4.0		1.00	12.00	1.00	1.00	1.00	1.00	1.00	ERR	1.04	1.14		1.01	0.98
COT k*b=4.0*		1.00	1.00	1.00	1.23	1.23	1.25	1.00	ERR	1.04	1.21		1.01	1.00
WTAR		1.00	36.00	1.01	3.15	3.15	3.81	1.00	ERR	1.00	1.43		1.01	0.98
WTAR*		1.00	24.00	1.01	3.00	3.00	3.75	1.00	ERR	1.04	1.43		1.00	0.98
LOAD		1.00	23.00	1.00	1.46	1.46	1.88	1.00	ERR	1.04	1.21		1.00	0.98
LOAD*		1.00	14.00	1.00	2.40	2.40	1.38	1.00	ERR	1.13	1.14		1.01	0.98

Example 2: 22/11/92 17:33	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	23	2	1		10600	265	61	67.1	2.8	26	3	5	388	77
RANDOM*	23	2	1		10600	265	61	67.1	2.8	26	3	5	388	77
SPT	23	1	7		2240	56	37	68.8	1.1	26	3	7	395	79
SPT*	23	1	7		2240	56	37	68.8	1.1	26	3	7	395	79
EDD	23	2	2		3480	87	55	66.8	3.1	26	5	6	388	77
EDD*	23	2	2		3480	87	55	66.8	3.1	26	5	6	388	77
COT k*b=4.0	23	1	2		3480	87	55	67.2	2.7	26	5	10	390	78
COT k*b=4.0*	23	1	2		3480	87	55	67.2	2.7	26	5	10	390	78
WTAR	23	1	1		3240	81	21	68.8	1.1	26	3	8	396	79
WTAR*	23	1	1		3240	81	21	68.8	1.1	29	3	8	396	79
LOAD	23	1	14		5760	144	43	68.8	1.1	29	2	9	397	79
LOAD*	23	1	14		5760	144	43	68.8	1.1	28	2	9	397	79
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	1		2240	56	21	68.8	1.1	26	2	5	388	79
Worst		2	14		10600	265	61	66.8	3.1	29	5	10	397	77
Average		1	5		4800	120	45	67.9	2.0	27	4	8	392	78
Ratio of best														
RANDOM		2.00	1.00		4.73	4.73	2.90	0.98	2.55	1.00	1.50	1.00	1.00	0.97
RANDOM*		2.00	1.00		4.73	4.73	2.90	0.98	2.55	1.00	1.50	1.00	1.00	0.97
SPT		1.00	7.00		1.00	1.00	1.76	1.00	1.00	1.00	1.50	1.40	1.02	1.00
SPT*		1.00	7.00		1.00	1.00	1.76	1.00	1.00	1.00	1.50	1.40	1.02	1.00
EDD		2.00	2.00		1.55	1.55	2.62	0.97	2.82	1.00	2.50	1.20	1.00	0.97
EDD*		2.00	2.00		1.55	1.55	2.62	0.97	2.82	1.00	2.50	1.20	1.00	0.97
COT k*b=4.0		1.00	2.00		1.55	1.55	2.62	0.98	2.45	1.00	2.50	2.00	1.01	0.99
COT k*b=4.0*		1.00	2.00		1.55	1.55	2.62	0.98	2.45	1.00	2.50	2.00	1.01	0.99
WTAR		1.00	1.00		1.45	1.45	1.00	1.00	1.00	1.00	1.50	1.60	1.02	1.00
WTAR*		1.00	1.00		1.45	1.45	1.00	1.00	1.00	1.12	1.50	1.60	1.02	1.00
LOAD		1.00	14.00		2.57	2.57	2.05	1.00	1.00	1.12	1.00	1.80	1.02	1.00
LOAD*		1.00	14.00		2.57	2.57	2.05	1.00	1.00	1.08	1.00	1.80	1.02	1.00

Example 3: 22/11/92 17:33	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	31	3	14		13120	328	90	79.6	3.9	41	15	9	451	65
RANDOM*	31	3	27		13120	328	90	79.6	3.9	41	15	9	451	65
SPT	31	2	3		9080	227	52	79.2	4.3	33	13	7	454	65
SPT*	31	2	1		9080	227	54	79.4	4.1	35	13	7	454	65
EDD	31	4	1		4360	109	47	80.0	3.5	35	10	9	458	66
EDD*	31	4	20		4520	113	43	79.9	3.6	35	10	9	458	66
COT k*b=4.0	31	2	10		4120	103	47	79.0	4.5	32	10	9	454	65
COT k*b=4.0*	31	3	1		4280	107	43	79.0	4.5	35	10	10	454	65
WTAR	31	2	1		5600	140	39	80.0	3.5	37	12	8	457	66
WTAR*	31	3	1		5560	139	40	81.2	2.3	39	12	8	464	66
LOAD	31	3	8		11080	277	60	79.6	3.9	37	13	3	454	65
LOAD*	31	3	6		11120	278	60	80.1	3.4	38	12	3	455	65
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		2	1		4120	103	39	81.2	2.3	32	10	3	451	66
Worst		4	27		13120	328	90	79.0	4.5	41	15	10	464	65
Average		3	8		7920	198	55	79.7	3.8	37	12	8	455	65
Ratio of best														
RANDOM		1.50	14.00		3.18	3.18	2.31	0.98	1.70	1.28	1.50	3.00	1.00	0.98
RANDOM*		1.50	27.00		3.18	3.18	2.31	0.98	1.70	1.28	1.50	3.00	1.00	0.98
SPT		1.00	3.00		2.20	2.20	1.33	0.98	1.87	1.03	1.30	2.33	1.01	0.98
SPT*		1.00	1.00		2.20	2.20	1.38	0.98	1.78	1.09	1.30	2.33	1.01	0.98
EDD		2.00	1.00		1.06	1.06	1.21	0.99	1.52	1.09	1.00	3.00	1.02	1.00
EDD*		2.00	20.00		1.10	1.10	1.10	0.98	1.57	1.09	1.00	3.00	1.02	1.00
COT k*b=4.0		1.00	10.00		1.00	1.00	1.21	0.97	1.96	1.00	1.00	3.00	1.01	0.98
COT k*b=4.0*		1.50	1.00		1.04	1.04	1.10	0.97	1.96	1.09	1.00	3.33	1.01	0.98
WTAR		1.00	1.00		1.36	1.36	1.00	0.99	1.52	1.16	1.20	2.67	1.01	1.00
WTAR*		1.50	1.00		1.35	1.35	1.03	1.00	1.00	1.22	1.20	2.67	1.03	1.00
LOAD		1.50	8.00		2.69	2.69	1.54	0.98	1.70	1.16	1.30	1.00	1.01	0.98
LOAD*		1.50	6.00		2.70	2.70	1.54	0.99	1.48	1.19	1.20	1.00	1.01	0.98

Example 4: 22/11/92 17:33	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	41	1	10		3600	90	38	79.2	2.1	45	15	8	439	65
RANDOM*	41	1	1		5240	131	38	79.0	2.3	44	15	4	440	66
SPT	41	1	21		3640	91	37	78.3	3.0	42	15	3	437	65
SPT*	41	1	8		4240	106	37	79.2	2.1	44	15	8	441	66
EDD	41	1	23		720	18	11	78.3	3.0	42	11	2	436	65
EDD*	41	1	22		720	18	11	78.3	3.0	43	11	2	436	65
COT k*b=4.0	41	1	8		2280	57	37	78.3	3.0	41	15	2	440	66
COT k*b=4.0*	41	1	8		2440	61	37	79.2	2.1	45	15	4	442	66
WTAR	41	1	10		4360	109	44	78.3	3.0	44	16	10	436	65
WTAR*	41	1	5		4560	114	44	78.3	3.0	46	16	8	436	65
LOAD	41	1	13		5880	147	41	78.3	3.0	44	17	10	440	65
LOAD*	41	1	15		4920	123	40	78.3	3.0	47	16	8	439	65
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best		1	1		720	18	11	79.2	2.1	41	11	2	436	66
Worst		1	23		5880	147	44	78.3	3.0	47	17	10	442	65
Average		1	12		3550	89	35	78.6	2.7	44	15	6	439	65
Ratio of best														
RANDOM		1.00	10.00		5.00	5.00	3.45	1.00	1.00	1.10	1.36	4.00	1.01	0.98
RANDOM*		1.00	1.00		7.28	7.28	3.45	1.00	1.10	1.07	1.36	2.00	1.01	1.00
SPT		1.00	21.00		5.06	5.06	3.36	0.99	1.43	1.02	1.36	1.50	1.00	0.98
SPT*		1.00	8.00		5.89	5.89	3.36	1.00	1.00	1.07	1.36	4.00	1.01	1.00
EDD		1.00	23.00		1.00	1.00	1.00	0.99	1.43	1.02	1.00	1.00	1.00	0.98
EDD*		1.00	22.00		1.00	1.00	1.00	0.99	1.43	1.05	1.00	1.00	1.00	0.98
COT k*b=4.0		1.00	8.00		3.17	3.17	3.36	0.99	1.43	1.00	1.36	1.00	1.01	1.00
COT k*b=4.0*		1.00	8.00		3.39	3.39	3.36	1.00	1.00	1.10	1.36	2.00	1.01	1.00
WTAR		1.00	10.00		6.06	6.06	4.00	0.99	1.43	1.07	1.45	5.00	1.00	0.98
WTAR*		1.00	5.00		6.33	6.33	4.00	0.99	1.43	1.12	1.45	4.00	1.00	0.98
LOAD		1.00	13.00		8.17	8.17	3.73	0.99	1.43	1.07	1.55	5.00	1.01	0.98
LOAD*		1.00	15.00		6.83	6.83	3.64	0.99	1.43	1.15	1.45	4.00	1.01	0.98

Example 5: 22/11/92 17:33	Total Jobs	Unfin Jobs	Passes	Mkspn h	BVal $	Tard h	Max Tard	VolD ML	VolR ML	Parcels	Setups	O/T h	Proc T h	Utilzn %
RANDOM	37	2	12		13960	349	79	73.0	3.5	44	16	8	405	60
RANDOM*	37	1	47		9720	243	74	74.6	1.9	41	13	12	417	15
SPT	37	1	4		16960	424	124	74.0	2.5	41	17	12	415	62
SPT*	37	1	9		16960	423	124	74.0	2.5	40	17	10	414	61
EDD	37	1	13		16240	406	79	74.5	2.0	42	15	7	418	61
EDD*	37	1	16		17920	448	74	73.5	3.0	44	18		414	60
COT k*b=4.0	37	1	28		14760	369	124	74.0	2.5	41	17	6	412	61
COT k*b=4.0*	37	3	8		16440	411	124	69.9	6.6	42	16	3	388	58
WTAR	37	1	6		14600	365	71	73.5	3.0	43	17	2	412	61
WTAR*	37		14	168	11920	298	76	76.5		47	13	8	428	63
LOAD	37	3	3		18600	465	100	71.6	4.9	40	15		406	59
LOAD*	37	3	3		17960	449	100	72.4	4.1	43	15		410	60
Criteria		min	min	min	min	min	min	max	min	min	min	min	min	max
Best			3		9720	243	71	76.5		40	13		388	63
Worst		3	47	168	18600	465	124	69.9	6.6	47	18	12	428	15
Average		2	14	56	15503	388	96	73.5	3.0	42	16	6	412	57
Ratio of best														
RANDOM			4.00		1.44	1.44	1.11	0.95		1.10	1.23		1.04	0.95
RANDOM*			15.67		1.00	1.00	1.04	0.98		1.03	1.00		1.07	0.24
SPT			1.33		1.74	1.74	1.75	0.97		1.03	1.31		1.07	0.98
SPT*			3.00		1.74	1.74	1.75	0.97		1.00	1.31		1.07	0.97
EDD			4.33		1.67	1.67	1.11	0.97		1.05	1.15		1.08	0.97
EDD*			5.33		1.84	1.84	1.04	0.96		1.10	1.38		1.07	0.95
COT k*b=4.0			9.33		1.52	1.52	1.75	0.97		1.03	1.31		1.06	0.97
COT k*b=4.0*			2.67		1.69	1.69	1.75	0.91		1.05	1.23		1.00	0.92
WTAR			2.00		1.50	1.50	1.00	0.96		1.08	1.31		1.06	0.97
WTAR*			4.67		1.23	1.23	1.07	1.00		1.18	1.00		1.10	1.00
LOAD			1.00		1.91	1.91	1.41	0.94		1.00	1.15		1.05	0.94
LOAD*			1.00		1.85	1.85	1.41	0.95		1.08	1.15		1.06	0.95

Appendix E SIMULATION RESULTS

Experiment	Description
6	Apparent Tardiness Cost Tuning
7	Cost Over Time Tuning
8	Pattern Selection Evaluation
9	Scheduling Rule Evaluation

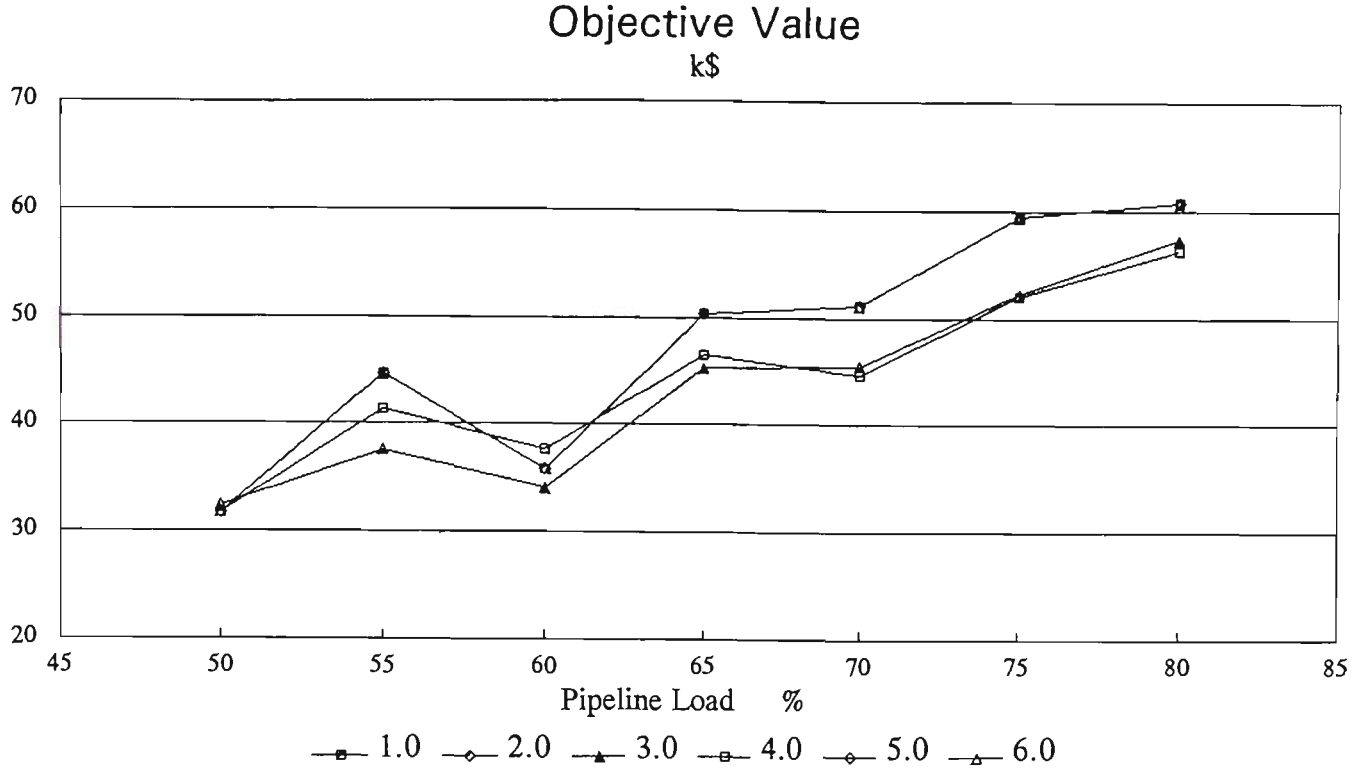


Figure E.6.1 Weighted Tardiness response to "look ahead" parameter k

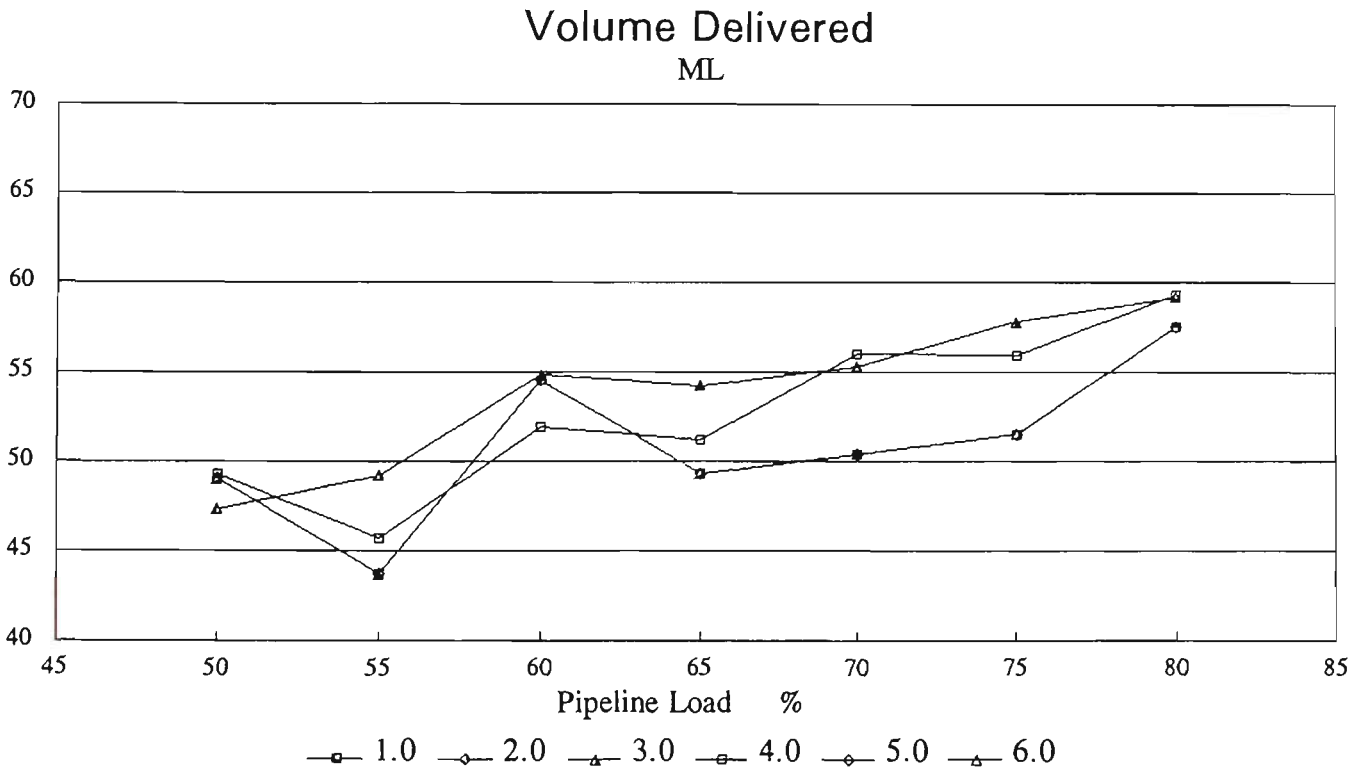


Figure E.6.2 Volume Delivered response to "look ahead" parameter k

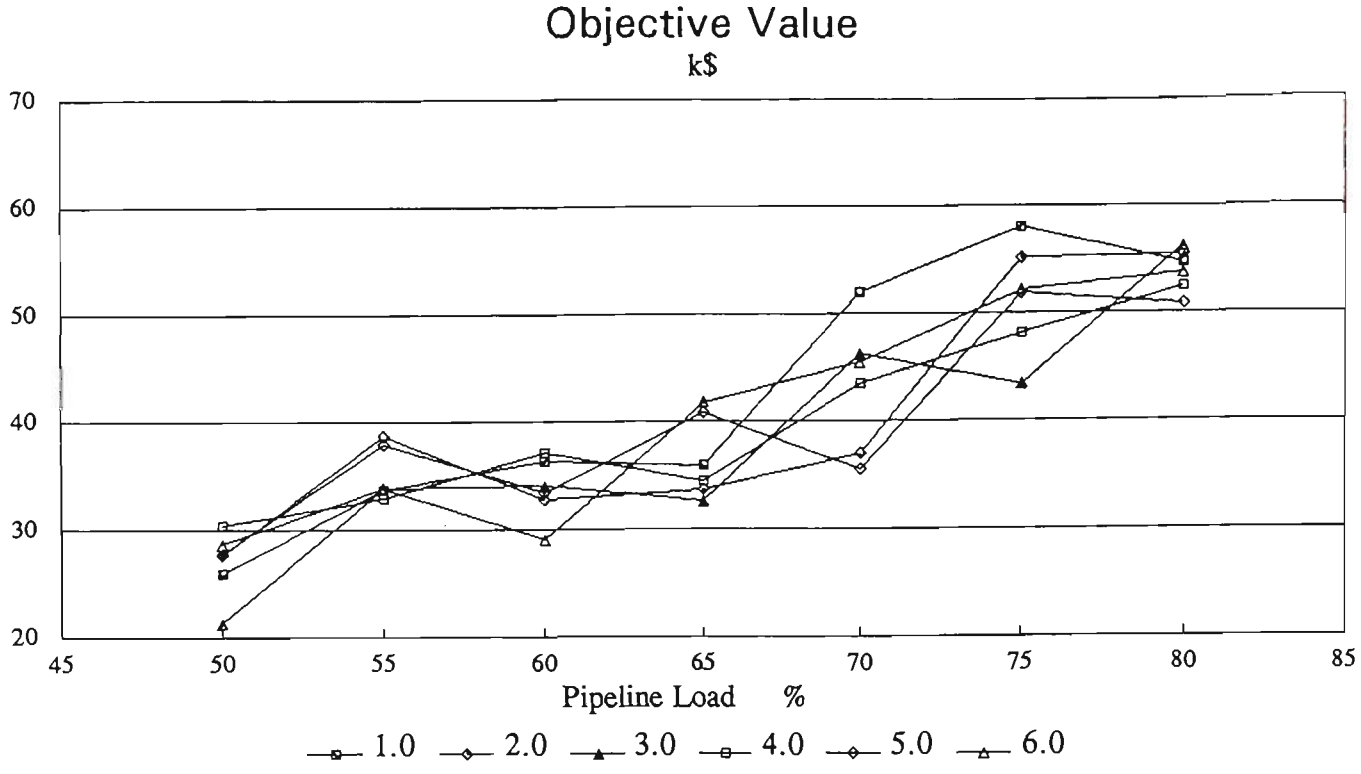


Figure E.7.1 Weighted Tardiness response to "waiting time" parameter k*b

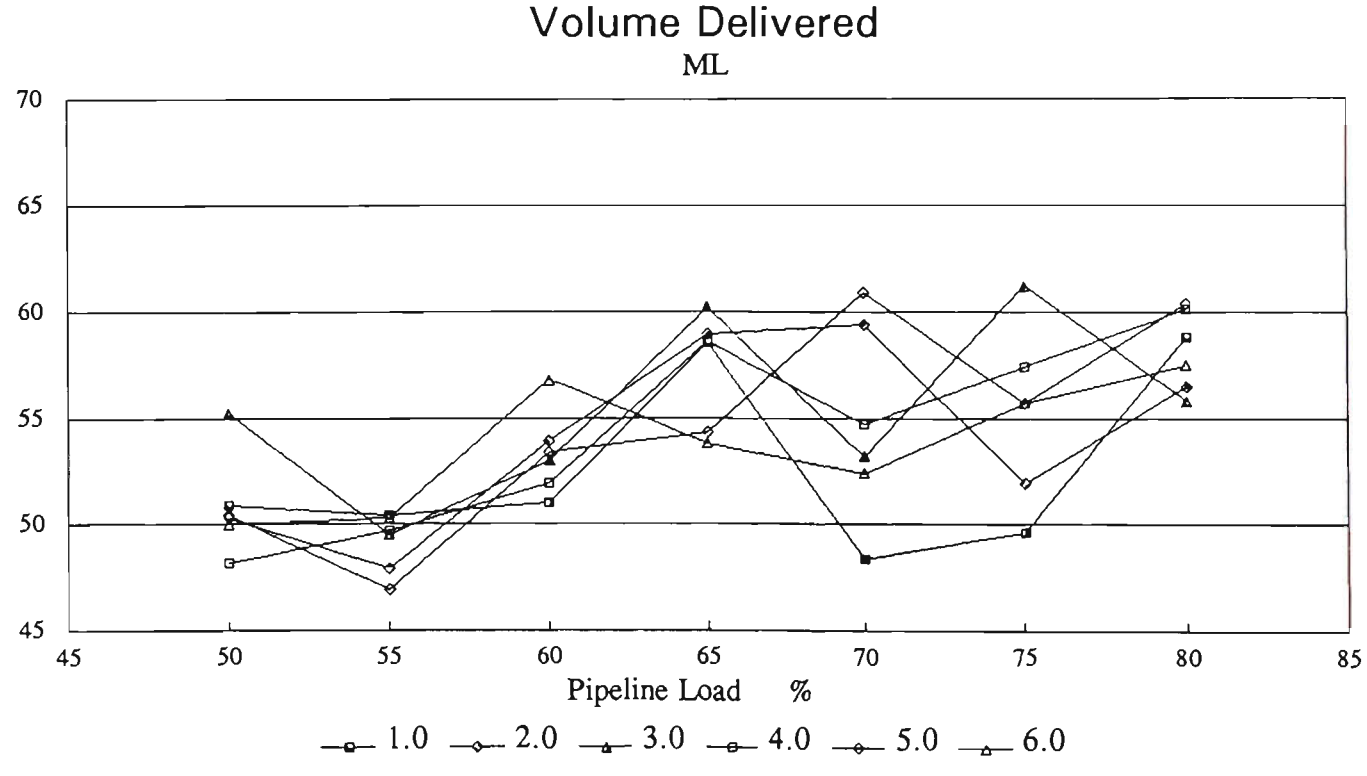


Figure E.7.2 Volume Delivered response to "waiting time" parameter k*b

Experiment 6 Apparent Tardiness Cost tuning (k = 1.0) (simulation)

Experiment 6.1: 10/11/92 20:43	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
ATC (k=1.0) (Utz=50)	27.0	7.4	24.1	33.2	31.8	796.2	133.8	28.9	49.0	18.6	8.4	41.6	33.6	46.9
ATC (k=1.0) (Utz=55)	30.0	11.9	16.0	45.8	44.6	114.5	136.8	36.8	43.7	30.0	8.1	36.8	38.3	48.2
ATC (k=1.0) (Utz=60)	31.2	9.2	25.6	37.6	35.9	898.6	132.6	26.2	54.5	23.9	9.1	45.8	43.0	49.5
ATC (k=1.0) (Utz=65)	33.8	13.9	14.4	51.4	50.4	1260.6	137.5	32.2	49.3	36.0	9.3	41.4	47.7	50.8
ATC (k=1.0) (Utz=70)	36.4	15.8	14.1	52.8	51.1	1277.0	136.2	30.7	50.4	41.5	10.1	42.9	52.4	52.2
ATC (k=1.0) (Utz=75)	40.4	18.5	15.6	60.9	59.4	1485.4	141.3	29.1	51.5	46.4	9.8	43.5	57.1	53.5
ATC (k=1.0) (Utz=80)	42.0	19.0	15.1	61.9	60.7	1516.6	143.2	26.0	57.5	47.1	10.5	48.6	61.8	54.8
												a=	-13.4	33.7
												b=	0.939	0.264
												r^2=	0.850	0.425
Pattern Utilization														
ATC (k=1.0) (Utz=50)														
ATC (k=1.0) (Utz=55)														
ATC (k=1.0) (Utz=60)														
ATC (k=1.0) (Utz=65)														
ATC (k=1.0) (Utz=70)														
ATC (k=1.0) (Utz=75)														
ATC (k=1.0) (Utz=80)														
												a=		
												b=		
												r^2=		

Experiment 6 Apparent Tardiness Cost tuning (k = 2.0) (simulation)

Experiment 6.2: 22/11/92 17:39	Total Jobs	Unfin Jobs	Passes	FVal $	BVal $	Tardi h	Max Tardi	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
ATC (k=2.0) (Utz=50)	27.0	7.4	24.1	33.2	31.8	796.2	133.8	28.9	49.0	18.6	8.4	41.6	33.6	46.9
ATC (k=2.0) (Utz=55)	30.0	11.9	16.0	45.8	44.6	1114.5	136.8	36.8	43.7	30.0	8.1	36.8	38.3	48.2
ATC (k=2.0) (Utz=60)	31.2	9.2	25.6	37.6	35.9	989.6	132.6	26.2	54.5	23.9	9.1	45.8	43.0	49.5
ATC (k=2.0) (Utz=65)	33.8	13.9	14.4	51.4	50.4	1260.6	137.5	32.2	49.3	36.0	9.3	41.4	47.7	50.8
ATC (k=2.0) (Utz=70)	36.4	15.8	14.1	52.8	51.1	1277.0	136.2	30.7	50.4	41.5	10.1	42.9	52.4	52.2
ATC (k=2.0) (Utz=75)	40.4	18.5	15.6	60.9	59.4	1485.4	141.3	29.1	51.5	46.4	9.8	43.5	57.1	53.5
ATC (k=2.0) (Utz=80)	42.0	19.0	15.1	61.9	60.7	1516.6	143.2	26.0	57.5	47.1	10.5	48.6	61.8	54.8
												a=	-13.4	33.7
												b=	0.939	0.264
												r^2=	0.850	0.425
Pattern Utilization														
ATC (k=2.0) (Utz=50)														
ATC (k=2.0) (Utz=55)														
ATC (k=2.0) (Utz=60)														
ATC (k=2.0) (Utz=65)														
ATC (k=2.0) (Utz=70)														
ATC (k=2.0) (Utz=75)														
ATC (k=2.0) (Utz=80)														
												a=		
												b=		
												r^2=		

Experiment 6.3: 11/11/92 19:26	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
ATC (k=3.0) (Utz=50)	27.0	8.1	17.4	33.8	32.4	810.7	135.1	30.8	47.3	20.3	8.4	40.4	31.1	48.3
ATC (k=3.0) (Utz=55)	30.0	9.7	18.8	39.0	37.6	939.1	132.3	31.5	49.2	24.5	9.4	41.0	35.3	50.2
ATC (k=3.0) (Utz=60)	31.2	8.8	18.5	35.6	34.1	853.7	132.2	24.5	54.8	23.6	8.5	46.0	39.4	52.1
ATC (k=3.0) (Utz=65)	33.8	12.3	14.3	46.0	45.3	1133.2	139.6	29.1	54.2	31.1	9.5	45.6	43.5	54.0
ATC (k=3.0) (Utz=70)	36.4	13.9	16.3	46.5	45.5	1136.6	132.9	25.0	55.3	36.6	10.3	46.4	47.6	55.9
ATC (k=3.0) (Utz=75)	40.4	16.0	16.9	53.8	52.3	1307.7	139.0	24.5	57.8	40.1	10.8	48.8	51.8	57.8
ATC (k=3.0) (Utz=80)	42.0	17.8	16.8	58.7	57.3	1433.3	140.9	24.8	59.2	45.4	11.8	50.0	55.9	59.7
												a=	-10.1	29.2
												b=	0.825	0.381
												r^2=	0.914	0.911
Pattern Utilization														
ATC (k=3.0) (Utz=50)	27.0	7.3	16.8	32.5	30.9	773.7	129.3	30.9	50.3	17.3	9.5	43.1	30.9	49.4
ATC (k=3.0) (Utz=55)	30.0	8.8	18.5	35.6	34.6	864.2	132.4	28.9	51.8	21.9	10.2	43.7	36.0	50.1
ATC (k=3.0) (Utz=60)	31.2	12.1	14.5	44.6	43.4	1084.4	136.4	33.3	47.5	30.9	9.3	40.1	41.0	50.9
ATC (k=3.0) (Utz=65)	33.8	12.1	15.8	44.3	43.2	1080.7	134.6	28.1	54.3	31.0	10.3	45.5	46.0	51.6
ATC (k=3.0) (Utz=70)	36.4	17.5	10.0	56.8	55.9	1397.6	141.1	33.0	46.3	45.6	9.3	39.1	51.1	52.3
ATC (k=3.0) (Utz=75)	40.4	16.3	15.8	53.5	52.5	1312.8	139.8	25.5	56.8	41.1	10.3	48.2	56.1	53.1
ATC (k=3.0) (Utz=80)	42.0	19.9	15.6	62.9	61.8	1545.5	138.3	25.9	54.2	50.4	10.5	45.7	61.2	53.8
												a=	-19.4	42.1
												b=	1.007	0.146
												r^2=	0.931	0.171

Experiment 6 Apparent Tardiness Cost tuning (k=4.0) (simulation)

Experiment 6.4: 10/11/92 20:43	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
ATC (k=4.0) (Utz=50)	27.0	7.3	16.8	33.1	31.7	791.4	130.8	30.9	49.3	18.3	9.1	42.4	33.4	46.9
ATC (k=4.0) (Utz=55)	30.0	11.0	15.6	42.8	41.4	1034.0	136.2	36.1	45.7	28.0	9.3	38.6	37.0	48.9
ATC (k=4.0) (Utz=60)	31.2	10.0	18.9	38.8	37.7	941.5	136.1	27.6	51.9	26.5	8.4	43.6	40.7	50.8
ATC (k=4.0) (Utz=65)	33.8	13.3	14.5	47.4	46.5	1163.1	137.9	29.1	51.2	34.1	9.3	42.9	44.3	52.8
ATC (k=4.0) (Utz=70)	36.4	13.5	13.9	45.6	44.6	1116.1	136.6	25.4	56.0	35.9	10.5	47.3	48.0	54.7
ATC (k=4.0) (Utz=75)	40.4	16.4	16.1	53.6	52.1	1302.3	140.6	25.5	55.9	42.0	10.0	47.3	51.6	56.7
ATC (k=4.0) (Utz=80)	42.0	17.9	16.6	57.7	56.3	1408.2	139.2	24.1	59.3	45.3	11.0	50.0	55.3	58.6
												a=	-3.1	27.5
												b=	0.729	0.389
												r^2=	0.885	0.828
Pattern Utilization														
ATC (k=4.0) (Utz=50)														
ATC (k=4.0) (Utz=55)														
ATC (k=4.0) (Utz=60)														
ATC (k=4.0) (Utz=65)														
ATC (k=4.0) (Utz=70)														
ATC (k=4.0) (Utz=75)														
ATC (k=4.0) (Utz=80)														
												a=		
												b=		
												r^2=		

Experiment 6 Apparent Tardiness Cost tuning (k=5.0) (simulation)

Experiment 6.5: 10/11/92 20:43	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
ATC (k=5.0) (Utz=50)	27.0	7.4	24.1	33.2	31.8	796.2	133.8	28.9	49.0	18.6	8.4	41.6	33.6	46.9
ATC (k=5.0) (Utz=55)	30.0	11.9	16.0	45.8	44.6	1114.5	136.8	36.8	43.7	30.0	8.1	36.8	38.3	48.2
ATC (k=5.0) (Utz=60)	31.2	9.2	25.6	37.6	35.9	898.6	132.6	26.2	54.5	23.9	9.1	45.8	43.0	49.5
ATC (k=5.0) (Utz=65)	33.8	13.9	14.4	51.4	50.4	1260.6	137.5	32.2	49.3	36.0	9.3	41.4	47.7	50.8
ATC (k=5.0) (Utz=70)	36.4	15.8	14.1	52.8	51.1	1277.0	136.2	30.7	50.4	41.5	10.1	42.9	52.4	52.2
ATC (k=5.0) (Utz=75)	40.4	18.5	15.6	60.9	59.4	1485.4	141.3	29.1	51.5	46.4	9.8	43.5	57.1	53.5
ATC (k=5.0) (Utz=80)	42.0	19.0	15.1	61.9	60.7	1516.6	143.2	26.0	57.5	47.1	10.5	48.6	61.8	54.8
												a=	-13.4	33.7
												b=	0.939	0.264
												r^2=	0.850	0.425
Pattern Utilization														
ATC (k=5.0) (Utz=50)														
ATC (k=5.0) (Utz=55)														
ATC (k=5.0) (Utz=60)														
ATC (k=5.0) (Utz=65)														
ATC (k=5.0) (Utz=70)														
ATC (k=5.0) (Utz=75)														
ATC (k=5.0) (Utz=80)														
												a=		
												b=		
												r^2=		

Experiment 6 Apparent Tardiness Cost tuning (k = 6.0) (simulation)

Experiment 6.6: 10/11/92 20:43	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
ATC (k=6.0) (Utz=50)	27.0	7.4	24.1	33.2	31.8	796.2	133.8	28.9	49.0	18.6	8.4	41.6	33.6	46.9
ATC (k=6.0) (Utz=55)	30.0	11.9	16.0	45.8	44.6	1114.5	136.8	36.8	43.7	30.0	8.1	36.8	38.3	48.2
ATC (k=6.0) (Utz=60)	31.2	9.2	25.6	37.6	35.9	898.6	132.6	26.2	54.5	23.9	9.1	45.8	43.0	49.5
ATC (k=6.0) (Utz=65)	33.8	13.9	14.4	51.4	50.4	1260.6	137.5	32.2	49.3	36.0	9.3	41.4	47.7	50.8
ATC (k=6.0) (Utz=70)	36.4	15.8	14.1	52.8	51.1	1277.0	136.2	30.7	50.4	41.5	10.1	42.9	52.4	52.2
ATC (k=6.0) (Utz=75)	40.4	18.5	15.6	60.9	59.4	1485.4	141.3	29.1	51.5	46.4	9.8	43.5	57.1	53.5
ATC (k=6.0) (Utz=80)	42.0	19.0	15.1	61.9	60.7	1516.6	143.2	26.0	57.5	47.1	10.5	48.6	61.8	54.8
												a=	-13.4	33.7
												b=	0.939	0.264
												r^2=	0.850	0.425
Pattern Utilization														
ATC (k=6.0) (Utz=50)														
ATC (k=6.0) (Utz=55)														
ATC (k=6.0) (Utz=60)														
ATC (k=6.0) (Utz=65)														
ATC (k=6.0) (Utz=70)														
ATC (k=6.0) (Utz=75)														
ATC (k=6.0) (Utz=80)														
												a=		
												b=		
												r^2=		

Experiment 7 Cost Over Time tuning k*b = 1.0 (simulation)

Experiment 7.1: 09/11/92 19:34	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
COT (1.0) (Utz=50)	27.0	6.6	17.1	26.8	26.0	649.5	128.2	24.3	50.9	16.7	9.9	44.9	26.3	50.4
COT (1.0) (Utz=55)	30.0	9.2	15.9	34.3	33.7	843.4	130.7	28.1	50.4	23.3	10.5	42.6	31.7	51.1
COT (1.0) (Utz=60)	31.2	10.5	14.1	37.2	36.3	908.1	135.6	27.5	51.0	27.4	9.8	42.8	37.0	51.8
COT (1.0) (Utz=65)	33.8	10.6	14.6	36.5	36.0	899.7	133.8	23.4	58.5	26.8	11.0	49.4	42.3	52.5
COT (1.0) (Utz=70)	36.4	16.9	7.4	52.1	51.9	1298.4	139.2	29.6	48.4	43.5	10.7	40.9	47.7	53.2
COT (1.0) (Utz=75)	40.4	19.2	9.7	58.3	58.0	1449.5	143.7	27.7	49.6	48.3	10.9	42.0	53.0	53.9
COT (1.0) (Utz=80)	42.0	18.1	13.4	55.1	54.5	1361.5	140.6	22.9	58.8	45.8	11.5	49.6	58.4	54.6
												a=	-27.2	43.5
												b=	1.069	0.139
												r^2=	0.887	0.123

Pattern Utilization

COT (1.0) (Utz=50)
COT (1.0) (Utz=55)
COT (1.0) (Utz=60)
COT (1.0) (Utz=65)
COT (1.0) (Utz=70)
COT (1.0) (Utz=75)
COT (1.0) (Utz=80)

Experiment 7 Cost Over Time tuning k*b = 2.0 (simulation)

Experiment 7.2: 09/11/92 19:34	Total Jobs	Unfin Jobs	Passes	FVal $	BVal $	Tardi h	Max Tardi	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Utilization														
COT (2.0) (Utz=50)	27.0	7.1	14.4	28.1	27.6	689.6	132.9	26.9	50.3	17.3	9.3	43.4	27.1	50.7
COT (2.0) (Utz=55)	30.0	10.6	12.6	39.3	38.7	967.8	132.3	33.3	47.9	25.8	9.5	40.8	31.4	51.8
COT (2.0) (Utz=60)	31.2	9.3	16.6	33.8	32.8	819.1	134.3	26.7	53.9	24.5	10.2	45.6	35.7	53.0
COT (2.0) (Utz=65)	33.8	9.9	14.9	34.4	33.7	843.0	135.0	21.1	58.9	26.4	10.3	49.6	40.0	54.1
COT (2.0) (Utz=70)	36.4	11.8	15.6	37.7	37.0	924.0	130.1	21.6	59.4	32.5	11.7	50.0	44.3	55.3
COT (2.0) (Utz=75)	40.4	18.2	10.2	55.6	55.1	1378.0	143.3	26.8	51.9	46.0	10.3	44.1	48.6	56.4
COT (2.0) (Utz=80)	42.0	18.6	10.4	55.8	55.4	1384.6	139.7	24.0	56.5	48.1	11.6	47.8	52.9	57.6
												a=	-15.9	39.2
												b=	0.860	0.229
												r^2=	0.718	0.321

Pattern Utilization

COT (2.0) (Utz=50)
COT (2.0) (Utz=55)
COT (2.0) (Utz=60)
COT (2.0) (Utz=65)
COT (2.0) (Utz=70)
COT (2.0) (Utz=75)
COT (2.0) (Utz=80)

Experiment 7 Cost Over Time tuning (k*b=3.0) (simulation)

Experiment 7.3: 11/11/92 20:14	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
COT (3.0) (Utz=50)	27.0	4.8	21.7	23.2	21.3	531.4	121.9	21.2	55.2	12.4	10.8	49.5	23.7	52.7
COT (3.0) (Utz=55)	30.0	9.4	15.3	34.8	33.9	848.6	132.1	28.8	49.5	24.2	9.9	41.8	28.6	53.6
COT (3.0) (Utz=60)	31.2	9.8	18.4	35.2	34.1	851.7	132.6	26.1	53.0	25.4	9.1	44.4	33.4	54.5
COT (3.0) (Utz=65)	33.8	9.4	16.8	33.7	32.6	815.8	131.2	20.8	60.2	25.1	10.5	50.8	38.2	55.4
COT (3.0) (Utz=70)	36.4	14.8	14.0	47.0	46.2	1156.0	138.2	27.0	53.2	38.7	10.0	45.1	43.1	56.4
COT (3.0) (Utz=75)	40.4	14.1	15.8	44.6	43.4	1085.3	134.0	20.6	61.2	36.7	11.3	51.8	47.9	57.3
COT (3.0) (Utz=80)	42.0	19.3	11.7	56.8	56.1	1401.9	136.8	23.8	55.8	48.8	11.0	47.1	52.7	58.2
												a=	-24.7	43.6
												b=	0.968	0.181
												r^2=	0.859	0.225
Pattern Utilization														
COT (3.0) (Utz=50)	27.0	6.4	15.7	26.6	25.8	645.1	125.9	26.0	51.6	16.0	10.0	44.4	23.9	52.8
COT (3.0) (Utz=55)	30.0	8.0	17.0	30.9	30.3	756.7	133.1	25.6	53.7	20.0	10.0	45.5	29.3	53.4
COT (3.0) (Utz=60)	31.2	9.1	14.8	32.8	32.1	803.5	132.2	25.8	54.3	24.1	10.0	45.7	34.8	54.0
COT (3.0) (Utz=65)	33.8	10.5	16.8	37.1	36.3	907.6	135.2	23.5	58.1	27.2	10.4	48.8	40.2	54.5
COT (3.0) (Utz=70)	36.4	14.8	13.1	47.0	46.4	1160.3	137.0	27.1	53.3	38.6	10.3	45.0	45.7	55.1
COT (3.0) (Utz=75)	40.4	17.9	13.7	56.7	55.9	1398.3	141.1	27.0	53.1	44.8	11.0	44.8	51.1	55.7
COT (3.0) (Utz=80)	42.0	18.3	11.3	55.2	54.7	1367.3	136.9	23.7	57.7	46.9	11.5	48.7	56.5	56.3
												a=	-30.5	47.1
												b=	1.087	0.115
												r^2=	0.939	0.259

Experiment 7 Cost Over Time tuning k*b=4.0 (simulation)

Experiment 7.4: 09/11/92 19:34	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
COT (4.0) (Utz=50)	27.0	7.7	11.9	31.0	30.4	759.7	133.7	32.6	48.2	19.4	9.6	42.1	28.8	48.6
COT (4.0) (Utz=55)	30.0	9.3	17.6	34.3	32.9	823.1	132.0	28.1	49.7	24.0	10.0	41.7	32.5	50.5
COT (4.0) (Utz=60)	31.2	10.2	16.9	38.0	37.1	928.1	132.8	29.6	51.9	26.5	9.3	43.8	36.2	52.4
COT (4.0) (Utz=65)	33.8	10.0	14.8	35.4	34.5	862.5	136.4	21.8	58.6	26.7	9.8	49.2	39.8	54.4
COT (4.0) (Utz=70)	36.4	14.1	11.3	44.7	43.5	1087.2	131.4	25.8	54.7	37.2	10.6	46.4	43.5	56.3
COT (4.0) (Utz=75)	40.4	15.9	14.1	48.9	48.1	1202.7	139.0	23.1	57.4	40.5	9.9	48.7	47.2	58.2
COT (4.0) (Utz=80)	42.0	17.4	13.1	52.9	52.3	1307.8	138.1	23.4	60.1	44.5	12.0	50.8	50.8	60.1
												a=	-7.8	29.3
												b=	0.732	0.385
												r^2=	0.915	0.824

Pattern Utilization

COT (4.0) (Utz=50)
COT (4.0) (Utz=55)
COT (4.0) (Utz=60)
COT (4.0) (Utz=65)
COT (4.0) (Utz=70)
COT (4.0) (Utz=75)
COT (4.0) (Utz=80)

Experiment 7.5 09/11/92 19:34	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
COT (5.0) (Utz=50)	27.0	6.9	14.8	29.1	27.8	694.9	128.3	29.5	50.4	17.2	9.6	44.0	29.2	48.7
COT (5.0) (Utz=55)	30.0	10.5	15.3	39.0	38.0	949.4	136.6	32.8	46.9	26.8	9.7	39.4	32.7	50.6
COT (5.0) (Utz=60)	31.2	9.5	18.8	34.6	33.5	837.6	133.7	24.8	53.4	25.0	8.9	45.2	36.2	52.6
COT (5.0) (Utz=65)	33.8	12.0	16.1	41.6	40.8	1020.7	138.3	26.4	54.3	31.0	10.8	45.7	39.7	54.6
COT (5.0) (Utz=70)	36.4	11.4	19.0	36.4	35.5	887.0	132.2	20.4	60.9	31.0	11.1	51.1	43.3	56.5
COT (5.0) (Utz=75)	40.4	16.9	12.4	52.6	51.8	1295.3	141.9	26.1	55.7	42.2	10.4	47.3	46.8	58.5
COT (5.0) (Utz=80)	42.0	17.0	14.6	51.3	50.8	1270.8	135.3	22.1	60.4	44.2	12.4	50.8	50.3	60.5
												a=	-6.0	29.0
												b=	0.704	0.394
												r^2=	0.737	0.709

Pattern Utilization

COT (5.0) (Utz=50)
COT (5.0) (Utz=55)
COT (5.0) (Utz=60)
COT (5.0) (Utz=65)
COT (5.0) (Utz=70)
COT (5.0) (Utz=75)
COT (5.0) (Utz=80)

Experiment 7 Cost Over Time tuning k*b = 6.0 (simulation)

Experiment 7.6: 09/11/92 19:34	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
COT (6.0) (Utz=50)	27.0	7.3	15.6	29.2	28.7	717.0	135.5	28.2	50.0	17.6	10.2	43.8	27.0	50.7
COT (6.0) (Utz=55)	30.0	9.3	16.8	34.8	33.8	844.9	135.4	28.3	50.3	23.4	9.8	42.2	31.5	51.7
COT (6.0) (Utz=60)	31.2	8.1	19.5	30.1	29.1	727.2	125.4	21.3	56.8	21.6	9.1	47.7	36.1	52.8
COT (6.0) (Utz=65)	33.8	12.1	15.1	42.5	41.8	1044.1	138.9	27.4	53.8	31.5	10.5	45.2	40.6	53.8
COT (6.0) (Utz=70)	36.4	14.7	11.9	46.3	45.4	1135.1	136.3	26.7	52.4	39.5	9.9	43.9	45.2	54.8
COT (6.0) (Utz=75)	40.4	16.9	13.6	53.3	52.1	1301.5	141.2	26.0	55.7	42.2	9.8	47.2	49.8	55.9
COT (6.0) (Utz=80)	42.0	18.1	10.6	54.1	53.6	1340.9	138.6	22.8	57.5	47.1	11.3	48.5	54.3	56.9
												a=	-18.6	40.4
												b=	0.911	0.206
												r^2=	0.900	0.543

Pattern Utilization

COT (6.0) (Utz=50)
COT (6.0) (Utz=55)
COT (6.0) (Utz=60)
COT (6.0) (Utz=65)
COT (6.0) (Utz=70)
COT (6.0) (Utz=75)
COT (6.0) (Utz=80)

Experiment 8.1: 22/11/92 17:42	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
RANDOM (Load=50)	26.9	6.5	19.1	29.8	28.2	704.2	131.6	26.9	50.4	16.5	9.6	44.3	30.3	51.7
RANDOM (Load=55)	29.8	8.5	22.0	37.7	35.7	893.3	136.1	28.9	53.2	20.0	10.5	45.1	35.4	52.0
RANDOM (Load=60)	31.5	9.3	24.3	38.8	36.5	911.7	130.8	26.3	55.8	23.4	10.1	47.1	40.6	52.3
RANDOM (Load=65)	34.9	15.3	13.9	55.0	53.8	1345.0	138.2	33.4	49.0	36.7	9.8	41.2	45.7	52.6
RANDOM (Load=70)	36.8	15.8	17.5	53.4	52.1	1302.3	141.2	28.4	51.7	40.7	10.0	43.5	50.8	52.9
RANDOM (Load=75)	39.6	17.4	18.5	57.7	56.5	1412.0	140.0	26.6	53.9	43.8	10.0	45.7	56.0	53.3
RANDOM (Load=80)	42.0	19.3	13.1	58.6	57.1	1426.5	143.4	24.5	54.3	50.1	10.1	45.9	61.1	53.6
												a=	-21.1	48.4
												b=	1.028	0.064
												r^2=	0.876	0.086
Pattern Utilization														
RANDOM (Load=50)	27.5	6.1	25.4	30.5	28.3	706.9	133.5	27.9	51.5	14.8	9.3	43.6	30.2	50.5
RANDOM (Load=55)	29.7	9.3	20.9	38.4	36.3	908.6	133.2	29.3	52.6	22.0	10.6	44.4	35.4	51.4
RANDOM (Load=60)	32.2	11.8	17.0	45.2	43.8	1094.5	141.3	30.8	50.4	29.5	9.4	42.1	40.7	52.3
RANDOM (Load=65)	34.3	13.2	14.4	46.5	45.3	1131.7	136.5	29.7	50.7	34.4	8.8	42.9	45.9	53.1
RANDOM (Load=70)	37.9	15.5	16.1	53.6	52.2	1305.1	137.7	27.5	53.9	37.7	10.2	45.5	51.2	54.0
RANDOM (Load=75)	39.7	15.7	16.6	51.7	50.1	1252.0	139.9	24.4	58.1	40.2	12.1	49.2	56.4	54.9
RANDOM (Load=80)	42.6	20.7	12.2	66.9	65.3	1631.3	142.9	26.7	54.7	49.9	11.3	46.1	61.7	55.7
												a=	-22.4	41.9
												b=	1.050	0.172
												r^2=	0.918	0.471

Experiment 8 Performance Evaluation: FCFS (simulation)

Experiment 8.2: 22/11/92 17:45	Total Jobs	Unfin Jobs	Passes	FVal $	BVal $	Tardi h	Max Tardi	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
FCFS (Load=50)	27.0	6.6	24.6	32.4	30.3	758.7	132.3	27.4	51.0	16.6	9.3	44.1	32.5	49.5
FCFS (Load=55)	30.0	10.0	21.9	41.8	39.7	992.5	136.0	32.1	49.3	24.4	9.3	41.8	37.1	50.6
FCFS (Load=60)	31.2	11.7	15.8	44.9	44.1	1101.8	140.6	33.8	49.0	29.4	9.6	41.4	41.7	51.7
FCFS (Load=65)	33.8	11.6	20.6	45.3	43.9	1098.3	136.0	27.1	56.1	29.2	9.3	47.2	46.3	52.8
FCFS (Load=70)	36.4	15.1	16.9	51.6	50.3	1257.3	138.6	28.8	52.8	39.1	10.6	44.8	50.9	54.0
FCFS (Load=75)	40.4	16.9	15.1	56.6	54.9	1372.7	141.5	26.2	56.0	41.9	10.6	47.5	55.5	55.1
FCFS (Load=80)	42.0	19.5	16.6	62.3	61.0	1525.1	141.6	25.7	55.7	48.9	11.3	47.2	60.1	56.2
												a=	-13.4	38.3
												b=	0.919	0.224
												r^2=	0.960	0.588
Pattern Utilization														
FCFS (Load=50)	27.0	7.9	18.4	34.9	33.3	831.9	134.4	33.5	48.5	19.1	9.9	41.7	32.0	49.3
FCFS (Load=55)	30.0	8.5	20.9	37.0	35.0	874.5	134.6	29.6	52.2	21.5	10.4	44.3	36.7	50.3
FCFS (Load=60)	31.2	10.3	22.0	42.6	40.9	1022.0	136.5	31.3	52.2	26.2	9.7	44.0	41.3	51.3
FCFS (Load=65)	33.8	12.6	14.9	45.7	44.5	1112.9	138.3	29.0	52.8	32.5	9.9	44.7	45.9	52.3
FCFS (Load=70)	36.4	17.1	9.2	54.9	54.3	1357.0	139.5	32.3	47.4	44.5	9.9	40.4	50.6	53.4
FCFS (Load=75)	40.4	16.8	17.8	55.8	54.6	1363.9	141.7	25.7	56.2	41.7	10.5	47.5	55.2	54.4
FCFS (Load=80)	42.0	18.9	14.4	60.2	59.0	1474.1	141.3	25.6	57.0	47.6	11.3	48.3	59.8	55.4
												a=	-14.3	39.0
												b=	0.926	0.205
												r^2=	0.965	0.387

Experiment 8.3: 22/11/92 17:45	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
SPT (Load=50)	27.0	7.5	17.1	30.0	29.2	730.7	132.1	29.5	48.8	18.8	8.0	41.9	31.0	47.8
SPT (Load=55)	30.0	10.1	12.7	38.5	38.0	949.9	133.9	32.0	47.9	25.8	8.9	40.3	35.2	49.3
SPT (Load=60)	31.2	9.8	17.1	37.1	36.2	903.8	132.2	28.6	52.8	25.6	9.2	44.3	39.4	50.7
SPT (Load=65)	33.8	13.1	14.5	45.6	44.8	1119.1	139.5	29.1	52.0	33.3	10.0	43.6	43.6	52.1
SPT (Load=70)	36.4	16.5	8.1	51.7	51.3	1282.6	138.7	30.8	49.2	42.7	10.0	41.6	47.8	53.6
SPT (Load=75)	40.4	16.3	14.0	51.7	51.1	1277.4	138.9	24.5	56.6	41.3	10.2	47.5	52.1	55.0
SPT (Load=80)	42.0	18.3	15.6	55.9	54.8	1369.1	135.6	23.3	57.7	46.9	11.4	48.7	56.3	56.5
												a=	-11.2	33.3
												b=	0.844	0.289
												r^2=	0.930	0.658
Pattern Utilization														
SPT (Load=50)	27.0	7.4	18.4	31.5	30.2	754.8	131.7	30.9	49.2	18.4	9.6	41.7	27.6	50.7
SPT (Load=55)	30.0	7.5	18.6	29.7	28.9	722.7	127.6	23.5	54.6	19.1	10.0	46.3	31.3	52.4
SPT (Load=60)	31.2	10.0	15.1	37.1	35.9	896.7	137.1	27.3	52.4	26.0	9.3	43.9	35.0	54.1
SPT (Load=65)	33.8	12.0	12.1	42.5	41.8	1046.2	135.1	27.7	54.0	31.3	10.8	45.5	38.7	55.8
SPT (Load=70)	36.4	10.7	16.1	33.8	33.1	827.3	127.7	18.7	63.0	28.9	10.8	53.0	42.4	57.5
SPT (Load=75)	40.4	15.8	14.8	50.1	49.1	1227.5	138.3	24.4	57.6	40.3	11.1	48.6	46.1	59.2
SPT (Load=80)	42.0	17.4	16.0	53.0	52.2	1304.6	138.4	22.8	59.5	45.1	11.8	50.2	49.8	60.8
												a=	-9.4	33.7
												b=	0.740	0.339
												r^2=	0.755	0.627

Experiment 8 Performance Evaluation: EDD (simulation)

Experiment 8.4: 22/11/92 17:45	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
EDD (Load=50)	27.0	5.8	19.2	25.2	24.1	603.3	126.7	23.8	53.3	14.3	10.5	46.6	27.7	48.8
EDD (Load=55)	30.0	9.8	18.4	35.7	34.6	866.0	132.8	27.4	48.2	25.5	9.4	40.5	32.1	50.3
EDD (Load=60)	31.2	10.9	16.6	38.8	37.5	936.6	136.7	29.1	49.8	28.6	9.4	42.2	36.6	51.8
EDD (Load=65)	33.8	13.0	10.9	43.8	43.1	1077.5	138.5	28.9	51.3	34.0	8.9	43.6	41.0	53.3
EDD (Load=70)	36.4	14.6	12.0	45.4	44.5	1113.4	136.1	25.9	53.5	38.4	10.6	45.5	45.5	54.7
EDD (Load=75)	40.4	16.8	15.1	51.5	51.0	1274.4	136.8	24.8	55.8	42.1	10.3	47.4	49.9	56.2
EDD (Load=80)	42.0	17.5	16.2	53.0	52.3	1307.2	136.4	21.8	60.9	43.7	11.9	51.4	54.3	57.7
												a=	-16.7	33.9
												b=	0.889	0.298
												r^2=	0.948	0.584
Pattern Utilization														
EDD (Load=50)	27.0	7.1	15.3	30.2	29.3	733.1	130.2	30.6	50.7	16.9	10.4	43.1	29.5	48.6
EDD (Load=55)	30.0	10.5	10.2	38.3	37.9	946.4	134.2	32.8	46.5	27.2	9.7	39.7	33.5	50.2
EDD (Load=60)	31.2	9.4	17.8	35.2	33.6	840.1	134.2	26.3	53.3	25.1	10.0	45.0	37.6	51.8
EDD (Load=65)	33.8	11.4	15.4	39.3	38.6	964.8	131.2	25.1	55.2	30.1	11.3	46.7	41.6	53.4
EDD (Load=70)	36.4	13.9	15.0	43.9	43.4	1084.1	131.8	25.8	54.7	37.2	11.0	46.3	45.7	55.0
EDD (Load=75)	40.4	18.9	11.3	59.0	58.6	1465.1	142.0	29.0	51.1	46.8	10.5	43.3	49.7	56.6
EDD (Load=80)	42.0	16.4	16.3	51.2	50.0	1250.2	137.2	22.0	62.1	42.5	12.5	52.4	53.8	58.2
												a=	-11.0	32.6
												b=	0.809	0.320
												r^2=	0.763	0.509

Experiment 8.5: 22/11/92 17:45	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
SLACK (Load=50)	27.0	6.2	17.1	25.7	24.4	610.0	125.1	22.8	52.0	15.6	10.2	46.2	25.0	51.4
SLACK (Load=55)	30.0	9.3	19.1	34.6	33.4	834.5	132.6	26.5	50.8	22.9	9.8	43.5	30.2	52.5
SLACK (Load=60)	31.2	10.3	23.8	38.6	36.9	923.5	134.0	27.9	52.3	26.1	9.5	44.5	35.4	53.5
SLACK (Load=65)	33.8	10.9	19.0	38.1	37.1	927.7	135.0	23.3	57.7	27.6	10.9	49.1	40.5	54.5
SLACK (Load=70)	36.4	12.8	17.1	40.4	39.0	976.1	127.8	22.8	57.6	34.3	11.6	49.1	45.7	55.6
SLACK (Load=75)	40.4	18.2	15.1	56.9	56.1	1401.3	140.6	27.7	52.9	45.0	10.4	45.3	50.9	56.6
SLACK (Load=80)	42.0	18.6	15.0	57.6	56.9	1421.6	138.2	24.4	58.5	46.1	12.2	49.8	56.1	57.7
												a=	-26.8	41.1
												b=	1.036	0.207
												r^2=	0.885	0.475
Pattern Utilization														
SLACK (Load=50)	27.0	6.5	18.6	28.0	26.8	669.3	130.8	26.5	51.8	15.8	10.7	45.6	28.1	49.6
SLACK (Load=55)	30.0	8.7	20.8	32.9	31.7	792.8	128.4	26.5	51.8	21.9	10.6	44.2	32.4	51.0
SLACK (Load=60)	31.2	11.8	17.0	41.7	40.9	1021.6	135.4	31.7	47.6	30.8	9.8	40.4	36.8	52.4
SLACK (Load=65)	33.8	11.0	17.1	38.8	37.8	943.8	134.1	24.5	56.6	28.7	10.6	48.1	41.1	53.9
SLACK (Load=70)	36.4	15.7	14.8	49.6	48.6	1214.7	137.0	29.3	51.1	40.8	10.3	43.4	45.5	55.3
SLACK (Load=75)	40.4	15.8	17.8	49.1	48.2	1204.7	137.3	23.5	59.1	38.8	10.8	50.2	49.8	56.7
SLACK (Load=80)	42.0	18.1	21.1	55.5	53.9	1346.7	138.1	22.9	59.1	45.5	12.0	50.2	54.2	58.2
												a=	-15.5	35.3
												b=	0.871	0.286
												r^2=	0.926	0.485

Experiment 8 Performance Evaluation: WTAR (simulation)

Experiment 8.6: 22/11/92 17:45	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
WTAR (Load=50)	27.0	8.9	12.3	36.8	35.9	897.7	130.6	35.9	45.7	21.9	10.3	39.1	31.4	48.0
WTAR (Load=55)	30.0	8.6	18.9	34.2	33.1	828.6	125.9	27.0	51.2	22.5	10.1	43.4	35.7	49.5
WTAR (Load=60)	31.2	10.1	21.0	38.9	37.2	931.0	135.0	26.0	52.2	26.2	9.9	44.1	39.9	50.9
WTAR (Load=65)	33.8	12.3	13.1	43.8	42.7	1067.3	136.0	26.8	54.0	31.3	10.3	45.5	44.2	52.4
WTAR (Load=70)	36.4	15.0	12.7	48.6	47.8	1194.7	135.9	28.3	52.5	39.4	11.5	44.6	48.5	53.9
WTAR (Load=75)	40.4	17.8	16.1	56.6	55.6	1390.7	138.8	26.0	53.6	44.3	9.9	45.5	52.7	55.3
WTAR (Load=80)	42.0	18.7	14.0	58.0	57.1	1426.8	138.5	24.4	57.6	47.0	12.5	48.9	57.0	56.8
												a=	-11.1	33.5
												b=	0.851	0.291
												r^2=	0.918	0.768
Pattern Utilization														
WTAR (Load=50)	27.0	8.3	15.8	34.3	33.0	824.1	132.7	34.2	47.1	20.5	9.2	40.2	33.4	46.4
WTAR (Load=55)	30.0	10.3	15.3	40.2	39.5	987.8	133.5	33.4	47.8	25.9	10.5	40.5	36.9	48.5
WTAR (Load=60)	31.2	11.7	16.9	42.7	41.5	1038.1	135.2	31.8	48.6	29.8	10.0	40.9	40.3	50.6
WTAR (Load=65)	33.8	11.3	20.7	40.6	39.5	988.3	132.6	25.0	55.6	29.7	10.8	46.9	43.7	52.7
WTAR (Load=70)	36.4	14.5	15.8	46.7	45.0	1124.8	132.4	26.2	54.4	37.5	10.9	46.1	47.2	54.8
WTAR (Load=75)	40.4	16.7	14.3	52.9	52.1	1302.9	140.6	24.0	55.8	42.1	10.8	47.3	50.6	56.9
WTAR (Load=80)	42.0	17.8	16.9	57.2	55.6	1391.2	137.1	24.6	59.5	45.1	12.9	50.3	54.1	59.0
												a=	-1.1	25.3
												b=	0.689	0.421
												r^2=	0.903	0.892

Experiment 8.7: 22/11/92 17:45	Total Jobs	Unfin Jobs	Passes	FVal k$	BVal k$	Tard h	Max Tard	NWT $	VolD ML	VolR ML	Setups	Utilzn %	BVal smthd	VolD smthd
Pattern Priority														
LOAD (Load=50)	27.0	8.6	21.0	37.3	35.2	881.2	137.4	31.8	46.7	20.9	8.8	40.3	35.2	46.5
LOAD (Load=55)	30.0	11.7	20.3	45.7	43.7	1091.6	137.7	36.0	44.7	29.0	9.0	37.8	39.9	47.5
LOAD (Load=60)	31.2	11.4	24.3	46.1	43.7	1092.5	137.2	32.2	50.2	28.2	9.3	42.5	44.6	48.5
LOAD (Load=65)	33.8	11.3	26.0	46.0	43.6	1090.6	137.6	26.7	58.2	27.1	10.0	49.0	49.3	49.5
LOAD (Load=70)	36.4	16.3	15.9	53.3	51.7	1293.0	28.0	50.5	41.4	10.3	42.8	44.6	53.9	50.5
LOAD (Load=75)	40.4	20.1	13.9	66.1	64.8	1620.3	143.7	30.3	49.3	48.6	9.5	41.8	58.6	51.5
LOAD (Load=80)	42.0	19.8	16.3	63.7	62.1	1553.3	140.7	26.3	55.8	48.8	11.0	47.2	63.3	52.4
												a=	-11.5	36.6
												b=	0.935	0.198
												r^2=	0.869	0.129
Pattern Utilization														
LOAD (Load=50)	27.0	8.3	19.1	35.6	34.1	852.4	135.4	32.3	47.2	20.4	9.4	40.7	33.7	47.3
LOAD (Load=55)	30.0	9.3	18.6	38.3	36.8	920.7	134.2	31.3	50.9	22.8	10.0	43.2	38.4	48.8
LOAD (Load=60)	31.2	13.1	18.0	50.2	48.3	1206.7	139.6	36.7	46.0	32.4	9.1	38.8	43.1	50.2
LOAD (Load=65)	33.8	12.5	21.9	47.1	44.4	1109.7	137.1	29.0	53.4	31.9	9.9	45.1	47.8	51.6
LOAD (Load=70)	36.4	15.4	14.2	50.9	49.7	1242.8	136.8	29.1	53.3	38.6	11.2	45.2	52.5	53.0
LOAD (Load=75)	40.4	17.9	20.8	60.0	58.1	1453.3	139.4	27.2	54.9	43.0	10.6	46.5	57.2	54.4
LOAD (Load=80)	42.0	20.1	19.9	64.8	63.3	1581.4	141.0	26.4	55.2	49.4	10.8	46.6	61.9	55.8
												a=	-13.3	33.3
												b=	0.940	0.281
												r^2=	0.923	0.680

Appendix F USER'S GUIDE

F.1 Introduction

This appendix contains the program user's guide written to assist schedulers operate the "APTS" computer program. It is written as a self-contained document because it will be used in practice without the supporting information of this thesis. Hence the page numbering and section numbering are independent of that used elsewhere in this thesis.

VICTORIA UNIVERSITY OF TECHNOLOGY

FOOTSCRAY CAMPUS

ALGORITHMS FOR PIPELINE TRANSFER SCHEDULING

PROGRAM USER'S GUIDE

Thesis by: **John Andrew YOUNG**
Department of Mathematics and Operations Research
Victoria University of Technology: Footscray Campus
Student number: 8401417

Supervisor: **Professor Robert E. JOHNSTON**
Professor, Australian Pulp and Paper Institute
Department of Chemical Engineering
Monash University

December 1992

CONTENTS

SUMMARY

This guide is intended to assist users operate the APTS pipeline scheduling program. APTS is an acronym for "Algorithms for Pipeline Transfer Scheduling". This is a collection of algorithms developed to assist in scheduling pipeline transfers of refined petroleum products from Mobil Oil Australia's Altona Refinery to nearby distribution terminals.

The guide explains the requirements to input data, execute the program and retrieve and analyse the program output. Example input files and corresponding output reports are provided.

A complete description of the underlying methodology and analysis of the algorithm's performance is described in [2].

Section 1 INTRODUCTION

1.1 Scope

This guide is intended to assist users operate the APTS pipeline scheduling program. The guide explains the requirements to input data, execute the program and retrieve and analyse the program output.

A complete explanation of the underlying algorithm and methodology employed is contained in Young [1]. Explanation of the MOA MVS Technical System "front end" is provided in Doran [1].

1.2 Program Description

APTS is an acronym for "Algorithms for Pipeline Transfer Scheduling". This is a collection of algorithms developed to assist in scheduling pipeline transfers of refined petroleum products from Mobil Oil Australia's Altona Refinery to nearby distribution terminals.

The program uses a heuristic approach to select a good feasible schedule from the many candidate schedules available. By "heuristic" we mean that the program makes use of certain "rules of thumb" which experience and commonsense have shown to be useful. In this manner the program can, relatively simply, discard many alternative solutions which are unlikely to be suitable, without having to fully develop them. This approach is necessary to enable a good solution to be reached quickly with a reasonable amount of computation.

Because of the constantly changing nature of the scheduling environment no program can hope to capture and exploit all of the system constraints. However, the usefulness of this program is its ability to quickly generate a good feasible schedule which can serve as the basis for review by the scheduler before implementation.

1.3 Program Model

The underlying model employed is shown schematically in Figure 1.1 below. Briefly this is composed of the following segments:

1. Refinery Finished Product Tankage, in which the product is stored prior to transfer.

2. Suction Pipelines, which connect the refinery tankage to the main pipelines. Note that some suction pipelines feed more than one main pipeline.

3. Main Pipelines, which connect the refinery to the offtaker distribution terminals. Each main pipeline is dedicated to transferring only a sub-set of the total range of refinery products.

4. Offtaker Pipelines, which connect the main pipelines to the offtaker finished product tankage in the offtaker distribution terminals. Some offtaker pipelines connect more than one main pipeline to the offtaker tankage.

5. Offtaker Distribution Terminals. From these terminals petroleum products are distributed to the market by road, rail and sea.

1.4 System Requirements

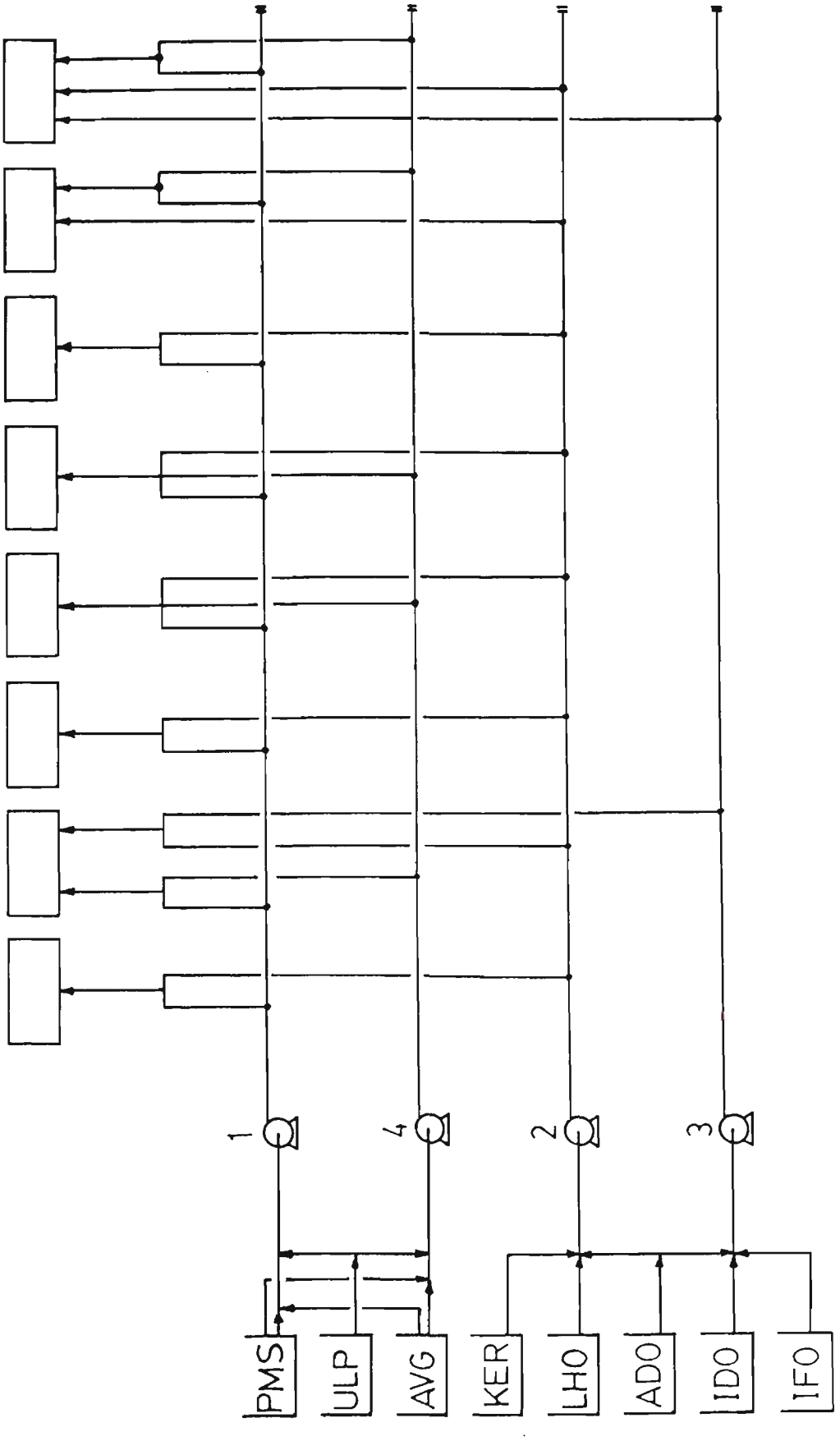
APTS is coded in Fortran 77 and operates in the MVS environment under the Technical System front end on MOA's mainframe. The program can easily be converted to run in a PC environment if required.

1.5 Program Limitations

The program has been designed to accommodate the following maximum problem size. These can easily be expanded if required.

Number of jobs	50
Number of main pipelines	5
Number of suction pipelines	5
Number of offtaker pipelines	5
Number of product types	20
Number of offtakers	10
Scheduling horizon	7 days
Scheduling resolution	1 hour

Figure 1.1 Program Model



Section 2 INPUT REQUIREMENTS

2.1 Application Organization

The scheduling application is arranged into seven main files. The first file is the program code. The input data supplied by the user is allocated amongst four files, based upon the type of information supplied. The remaining two files are the report files which contain the program output.

The file unit numbers to which these files are allocated are defined by data statements in the program code and can be changed to suit the computer system on which the program operates. The file names assigned to these unit numbers are defined by the user.

2.1.1 Program Organization

The program is organized into a main program and a number of supporting subroutines. Extensive use of "named common" is used to minimize data storage requirements and to efficiently transfer data between the main program and the subroutines.

BLOCK1
This is the block data program which assigns default values to the named common. Arrays and variables are grouped into named common based upon the type of data to which they relate. For example, all of the job data is grouped into JOBDAT and all of the pipeline data into PLNDAT.

MAIN
This is the main program which contains the scheduling algorithms. This program reads the input data and processes the information, calling the subroutines when required.

CONVRT
This subroutine converts the ready time and due time for each job supplied in the user input file from the user input format to the internal format used by the program. It also performs the reverse conversion from the internal program format to the user format prior to the program output reports being written.

SORT
This subroutine performs a tree sort of keys and tags into ascending key order. This is called at a number of stages within the scheduling algorithm, for example, to sort queues of jobs waiting to be selected and to sort patterns after generation prior to scheduling.

REPORT
This subroutine contains all of the reports. Each report is identified by a unique number which is passed from the main program when the subroutine is called.

2.1.2 Input Data Organization

The input data is organized into four files:

NETWORK
This file defines the pipeline network configuration, such as refinery product tankage, suction pipeline, main pipeline and offtaker pipeline linkages; pipeline pumping rates and lineclear (setup) times. This file also defines the major model parameters such as number of pipelines, number of offtakers and number of products. Once defined, the information contained in this file only changes when the pipeline configuration or capacities alter.

WINDOWS
This file defines the machine and resource time windows (the main pipeline and offtaker availability windows). These windows are defined for one calendar week, and, for offtaker availability, distinguish between normal and overtime work periods. This information only changes when the time windows alter.

JOBDATA
This file defines the schedule parameters, such as schedule horizon and time base, and provides all the job input data, such as ready and due times, product type and volume, and priority. This information changes for each schedule.

RUNDATA
This file defines the parameters which control the algorithm's performance. This includes the particular heuristics and job priority rules selected, and the relative weighting of job parameters to be used during scheduling. This allows the user to tune the algorithm to give the best overall scheduling performance. This information is usually common for each schedule. This file also allows the user to define the parameters used during simulation experiments.

2.1.3 Output Data Organization

Two output files are defined:

OUTPUT1
The main output file contains the summary reports of interest to all users. The reports may be selected by the user, as explained in Appendix F (Program User's Guide).

OUTPUT2
The secondary output file contains the supplementary reports used to track the internal computations of the program. These are only occasionally required and due to the amount of information produced are segregated from the primary output reports. As above, the reports may be selected by the user.

2.1.4 Input Data Format

In general, numerical input data is read in "real", as opposed to "integer" format. It is internally converted by the program into the data types required for program execution. This technique avoids potential problems that could occur with integer data being read incorrectly if not precisely placed on the data records of the input file. By reading the data in real format the data values only need to be placed within the field provided and the decimal point effectively defines the magnitude of the values.

2.2 Initial Customization

Prior to executing the first run the user must define the system parameters and data labels. The following comments refer to the example depicted in Figure 2.1 and sample input data shown in Figures 2.2, 2.3 and 2.4.

2.2.1 System Configuration

Data defining the system configuration is contained in the NETWORK file.

System Parameters

1. Title: primary title for all reports. Character format, 30 characters or less. Typically this is the name of the facility; in this case "HYPOTHETICAL REFINERY".

2. Pipelines: total number of main pipelines. This is the total number of main pipelines which exist in the system, not just the number that may be active in a particular program run. Real format (maximum of 5). In this example, PIPELINES = 5.

3. Products: total number of products in the system. Real format (maximum of 20). In this case, PRODUCTS = 10.

4. Offtakers: total number of offtakers in the system. Real format (maximum of 10). In this case, OFFTAKERS = 9.

Suction Pipeline Configuration Table

This defines the linkage between each main pipeline and the suction pipeline which connects it to the refinery product tankage. Each main pipeline may have only one suction pipeline, but each suction pipeline may feed more than one main pipeline.

The columns of the table are the main pipelines. This table has a single row. The suction pipeline number is entered under each main pipeline label. Suction pipelines which may feed more than one main pipeline simultaneously are identified by a negative sign.

Main Pipeline Setup Times

This table serves three main purposes. First, it defines the product types which may be transferred on each main pipeline. Secondly, it defines the setup (or lineclear) times required for sequential transfers of dissimilar product types. Thirdly, it defines the labels used to identify product types throughout the program.

The columns of the table are the main pipelines. The rows of the table are the product types. The table intersection values are the setup times (in hours) for allowable allocations. The product label is defined by the row name. A separate row is used for each product type.

Offtaker Pipeline Configuration Table

This table performs two tasks. First, it defines the linkage between each main pipeline and the offtaker distribution terminals which it feeds. Each offtaker may be connected to more than one main pipeline, and each main pipeline may feed more than one offtaker. Secondly, it defines the labels used to identify offtaker names throughout the program.

The columns of the table are the main pipelines. The rows of the table are the offtakers. The table intersection values are the offtaker pipeline number of each offtaker. Offtaker pipelines which may be fed by more than one main pipeline simultaneously are identified by a negative sign. The offtaker label is defined by the row name. A separate row is used for each offtaker.

Main Pipeline Pumping Rate Table
This table defines the pumping rate of each pipeline to each offtaker distribution terminal. The columns of the table are the main pipelines. The rows of the table are the offtakers. The table intersection values are the pumping rates (in kL/h). Pumping rates are required for all offtaker/main pipeline allocations defined in the offtaker configuration table above. The offtaker labels in the pumping rate table must match the offtaker labels in the configuration table.

2.2.2 System Availability

Data defining the system availability is contained in the WINDOWS file. A separate table is used to define the availability windows for each pipeline and offtaker. All tables have a common format. The columns of the table are the hours of the day. The rows of the table are the days of the week. The table intersection values define availability:

1	denotes the pipeline or offtaker is available
2	denotes the pipeline or offtaker is available on overtime
blank	denotes the pipeline or offtaker is unavailable

Each table is identified by a label in the first row following the column heading. The pipeline and offtaker labels must match the pipeline and offtaker labels defined in the NETWORK file. A separate table must be provided for each pipeline and offtaker defined in the NETWORK file.

2.3 Standard Operation

Data defining the basic case information and job requirements is contained in the JOBDATA file.

2.3.1 Case Description

Case Title
This enables the user to provide a comment to describe the case. This comment is concatenated with the system title provided in the NETWORK file and printed as the header on each report. Character format, 72 characters or less. In this case "EXAMPLE RUN".

Dates
The user must define the day and date of the schedule, the number of days in the month (to account for leap years) and the number of days of the scheduling horizon. The day of the week is entered as a 3 character label; the month, days in the month and scheduling horizon are entered in real format.

2.3.2 Job Details

This table contains all of the information to describe each job (i.e. each transfer). The columns of the table are as follows:

OFT Offtaker, the offtaker who will receive the job. This is the appropriate 3-character code of the offtakers defined in the NETWORK file.

PRD Product, the product type (such as gasoline, jet fuel, etc). This is the appropriate 3-character code of the products defined in the NETWORK file.

BCH Batch, the batch number of the batch from which the product is sourced. This is the 3-character batch code, used for reporting purposes only. Optional.

TNK Tank, the tank number of the tank from which the product is sourced. This is the 3-character tank code, used for reporting purposes only. Optional.

PTY Priority, the priority of the job amongst its peers. This is the 1-character priority code. If the job is currently being processed on a pipeline this code is the main pipeline code. Otherwise this is blank or one of the following codes. Blank defines normal priority and assigns MED priority.

H = high priority, M = medium priority, L = low priority.

VOL Volume, the volume of product to be transferred to the offtaker. This is the volume in megalitres (ML), entered in real format to one decimal place.

RDY Ready Time, the time the job becomes available to be transferred. This is entered in real format, with the 2-figure integer portion defining the date and the 2-figure decimal portion defining the time in hours (24 hour clock). For example, for jobs available at

7 am and 3 pm on the 23 rd of the month the ready times would be 23.07 and 23.15 respectively.

DUE Due Time, the time the job is to be finished; that is, all of the volume received by the offtaker. This is entered in the same manner as the ready time.

DLN Deadline Time, the time the job must be finished; that is, all of the volume must be received by the offtaker. This is entered in the same manner as the ready time. Optional.

A separate record is required for each job. The program automatically assigns a job number to the job based on the order in which the record appears in the table. Jobs may be deactivated by the user by entering an asterisk (*) in column 1 of the job record, in which case the program ignores the record. The table length is unrestricted. However, the number of active jobs is limited by the problem size defined in Section 1. If more jobs are defined the program ignores the excess jobs and prints a warning message.

2.4 Run Customization

Data defining the run details, such as report selection and tuning factors is contained in the RUNDATA file. Typically these parameters remain unchanged once initial customization is complete.

2.4.1 Reports

This section allows the user to select the type and frequency of reports generated by the program.

Primary Reports

Complete examples of all primary reports from the main output file are shown in Figures 4.1 to 4.5. All of the primary reports are written to the main output file.

Pipeline Schedule
This report in Gantt chart format enables the scheduler to quickly assess the overall performance and suitability of the generated schedule. It is designed to closely match the existing manually prepared Gantt chart. Normally only the chart corresponding to the best solution is printed. The user may, however, request the report be written for all solution passes, or written only when a better solution is identified, thus allowing the progress of the algorithm to be observed.

Transfer Summary
This report summarizes the primary input and output data for each job in tabular format. Normally only the summary corresponding to the best solution is printed. The user may, however, request the report be written for all solution passes, or written only when a better solution is identified, thus allowing the progress of the algorithm to be observed.

Availability Summary
This report indicates offtaker and pipeline availability windows in Gantt chart format. This enables the user to quickly assess any overtime requirements. Normally only the summary corresponding to the best solution is printed. The user may, however, request the report be written for all solution passes, or written only when a better solution is identified, thus allowing the progress of the algorithm to be observed.

Pipeline Utilization
This provides overall performance statistics for each pipeline. Not normally used. If selected only the statistics associated with the best solution pass are reported.

Simulation Statistics
This report summarizes the main performance statistics during simulation runs. Not normally used.

Secondary Reports

Extracts of each secondary report are shown in Figures 4.6 to 4.13. These are not normally used during standard operation, but are used to assess or check algorithm performance during development and tuning. If selected most of these reports are printed to the secondary output file, due to the quantity of information produced. Those reports which are written to the main output file are identified separately.

Pipeline System Data
Summarizes basic system structure, such as pipeline connections, pumping rates and setup data. Used to check input data. If selected this report is written to the main output file.

Job Input Data
Part 1 summarizes job input data including assigned product and offtaker numbers. Part 2 provides a listing of all possible operations, including suction and offtaker pipeline requirements. If selected this report is written to the main output file.

Pipeline Status
Summarizes main pipeline status at each time increment. Information includes current and previous product type (and hence setup requirement), job status, and required suction and offtaker pipelines.

Job Screening
Summarizes operation status at each time increment, listing the reason for elimination from the selection process if applicable. Includes setup and processing time requirements for each operation.

Job Queues
Summarizes the unsorted and sorted operation queues at each time increment prior to pattern generation. The summary can be limited to reporting only the sorted queues if required.

Pattern Generation
Summarizes the pattern generation phase, showing operation selection and pattern feasibility testing at each time increment.

Pattern Queues
Summarizes the unsorted and sorted pattern queues at each time increment after pattern generation but prior to pattern selection. The summary can be limited to reporting only the sorted queues if required.

Backtrack Summary
Summarizes backtracking information at each backtracking pass, showing backtracking point selection criteria.

Graphs

Performance Graph
The graph plots the objective value as a function of the number of solution passes.

2.4.2 Tuning Factors

The following tuning factors allow the user to customize the performance of the algorithm to best meet their particular needs. The program first

checks to determine if the each value given by the user is within the allowable range for the specific tuning factor. If not, the default value is assumed and a warning message printed.

Objective Function Weighting
This section allows the user to select and weight the parameters used in defining the objective function. Normally this is limited to weighted tardiness. However, the user can choose to explicitly include setup count and offtaker overtime into the objective function if required. Another option, yet to be explored, is to seek to minimize the makespan, i.e. the overall time taken to complete all of the jobs.

Job Priority Weighting
This section allows the user to specify the tardiness weights to be assigned to specific job classes for use in the objective function. Three job classes are defined (High, Medium and Low). The tardiness weighting factors are based on the demurrage costs incurred by each class, as shown in Attachment 1.

Pattern Priority Weighting
A pattern is a particular assignment of operations to machines at any given time. Usually a number of patterns are possible at any time and tuning factors are used to weight the relative importance of the main pattern characteristics to enable a choice to be made.

1. Pattern volume
   This value weights the total volume of material processable by the pattern. This is a measure of the amount of work the pattern can achieve, based on the duration of the pattern and the effective utilization of the pattern (the time spent transferring material as opposed to the time spent setting up or not being used).

   Minimum = 0, maximum = 9999, default = 0

2. Pattern duration
   This value weights the length of time the pattern is able to be processed. The pattern obviously ceases once one of the jobs included in the pattern is no longer able to be processed, either because that job has been completed, or an offtaker or pipeline is no longer available.

   Minimum = 0, maximum = 9999, default = 0

3. Pattern utilization
   This value weights the average processing rate of a pattern. The average processing rate is the volume transferred by the pattern divided by the pattern duration. This is a measure of average machine utilization which gives preference to patterns with all machines utilized and minimal required setup times.

   Minimum = 0, maximum = 9999, default = 1000

4. Pattern creation order
   This value weights the order in which the pattern was created amongst its peers.

   Minimum = 0, maximum = 9999, default = 100

5. Average Job Value
This value weights the average value of the operations which make up the pattern. The value in this case is the priority assigned by the job selection rule which ranks the jobs prior to pattern generation. Because of contention between jobs during pattern generation and the queue reordering method employed, patterns created latter in the generation phase may contain higher priority jobs than those created earlier.

Minimum = 0, maximum = 9999, default = 1

Job Preemption Factors
This section allows the user to alter the job tuning factor values to modify the solution obtained by controlling the treatment of jobs during scheduling. The user can alter the following factors:

1. Sibling work remaining
This value defines the time remaining to finish a sibling operation (which is already being processed), below which the operation being processed will not be stopped from being finished by the new operation. (A sibling operation is an operation with the same parent job as the operation currently being considered.)

Minimum = 0 h, maximum = 16 h, default = 4 h

2. Job work remaining
This value defines the processing time remaining to finish a job below which the job cannot be stopped from being finished by another job which needs the same machine. This value prevents inefficient preemption of machines that would result if a job that was nearly finished was stopped and then restarted at a later time.

Minimum = 0 h, maximum = 16 h, default = 12 h

3. Minimum work remaining
This value defines the minimum processing time that must be able to be performed without interruption to justify scheduling the operation. This prevents the inefficient utilization of machines that would result if operations were scheduled which could only be processed for a short time before some condition required their interruption.

Minimum = 0 h, maximum = 16 h, default = 4 h

Job Adjustment Factors
This section allows the user to adjust the relative priority of the operations to control the selection of jobs during scheduling. The user can alter the following factors:

1. Critical: Due time
This value defines the point, prior to the due time, beyond which the priority of the operation is increased to recognize the approaching milestone. This provides some "look ahead" capability, increasing the urgency of completing processing prior to tardiness occurring. The increased priority increases linearly within the critical interval.

Minimum = 0 h, maximum = 999 h, default = 12 h

2. Critical: Deadline
This value defines the point, prior to the deadline, beyond which the priority of the operation is increased to recognize the approaching milestone. This provides some "look ahead" capability, increasing the urgency of completing processing prior to a perceived infeasibility occurring. The increased priority increases linearly within the critical interval.

Minimum = 0 h, maximum = 999 h, default = 24 h

3. Maximum operation duration
This value defines the maximum processing time each operation can receive before the operation's relative priority amongst its peers must be reassessed. This prevents a long operation continuing to monopolize a machine if a higher priority operation becomes available after the operation currently being processed was scheduled.

Minimum = 0 h, maximum = 999 h, default = 1 h

4. On-line operation priority
This value increases the priority of operations currently being processed to reflect the advantage of continuing processing without interruption. This is effectively an additional setup time which the user may adjust as required to define the differential priority a waiting operation must have to interrupt an operation being processed.

Minimum = 0 h, maximum = 999 h, default = 4 h

5. Non-Preemptable operation priority
This value ensures that all non-preemptable operations continue being processed until they are completed or their non-preemptable status changes. Default = 500 h.

6. Critical operation priority
This value ensures that critical jobs receive a higher priority than all non-critical jobs, but a lower than that assigned to non-preemptable jobs. Default = 250 h

7. Sibling job switch
This value reduces the priority of an operation waiting to be scheduled if a sibling operation is already being processed and both siblings cannot be processed simultaneously. This avoids the inefficiency associated with stopping one operation to start another to deliver the same product to the same offtaker.

Minimum = -12 h, maximum = 0 h, default = 0 h

8. Job Average Capacity
If active, this flag ensures that the average job processing capacity is used to estimate the processing time of all operations derived from that job. Without this adjustment, decision rules based upon minimum slack or tardiness give a higher priority to the sibling operation being processed on the pipeline with the slowest pumping rate. This anomaly may prevent a job from being processed as quickly as possible because the algorithm would choose the slower job, due to its higher priority under these conditions.

2.4.3 Job Scheduling Rule Selection

Each job passing through the screening phase is ranked to control the order in which jobs are selected for scheduling. Ranked jobs are queued in a common queue and then sorted in descending order of priority.

Each rule is activated by setting the selection flag to 1 and deactivated by setting the flag to 0. Multiple rules may be selected at once and their relative importance specified by the magnitude of the selection flag. Hence, setting the Shortest Processing Time flag to 10 and the Earliest Due Date flag to 1, would result in jobs being ranked based on both processing time and due date, with processing time being 10 times more important than due date.

In ranking jobs the value of each of the indices corresponding to the active selection rules are computed. Each of these indices are then scaled from 0 to 1 before the weighting factors are applied and the values aggregated. For example, all of the shortest processing time indices are scaled from 0 (longest, lowest priority) to 1 (shortest, highest priority), and likewise for the earliest due date indices, from 0 (most distant, lowest priority) to 1 (closest, highest priority). These scaled values are then multiplied by their relative weights and added together to define the parameter used to rank the jobs during the sorting phase.

The different scheduling rules are as follows.

1. **Work Based**

Shortest Job Processing Time
The processing time remaining is the job volume remaining divided by the processing rate of the machine upon which the job will be processed. If the job is already being processed it is the volume of the job remaining divided by the total processing rate of the machine or machines on which the job is currently being processed. The highest priority is given to the job with the shortest processing time.

Longest Job Processing Time
The highest priority is given to the job with the longest processing time (i.e. the job with the most work remaining to be completed).

Longest Operation Processing Time
The highest priority is given to the job with the longest next operation processing time. The duration of the next operation is limited by either offtaker or pipeline availability or the amount of work remaining. In this later case the operation processing time is equal to the job processing time.

2. **Due Time Based**

Earliest Due Date
The highest priority is given to the job with the earliest due date.

3. **Tardiness Based**

Slack Time
In this scheme each operation is ranked using a measure based upon its slack time. This slack time is the difference between the available

processing time (between the current time and the due time) and the remaining job processing time. Small positive values indicate an urgency to commence processing the job. Negative values indicate that the job cannot be completed before the due time, i.e. the job will be tardy.

In this algorithm the raw slack time is adjusted to increase the priority of jobs already being processed to reduce the number of setups. Thus any job currently being processed is not displaced by a new job unless the priority of the new job justifies the physical effort and capacity loss inherent in the setup.

Modified Slack Time
If the slack time of a job reduces to a critical level the priority of the job can be increased to encourage processing and completion of the job before tardiness occurs.

Float Time
This parameter is analogous the the slack time, but is measured against the dead line, as opposed to the due time.

Weighted Tardiness
The highest priority is given to the operation with the highest weighted tardiness. The tardiness is based upon an estimate of the earliest possible completion time of the job, recognizing any relevant offtaker or machine downtime.

Weighted Tardiness Delta
In case the operation fails to start during the current time period the next possible starting time is estimated. The highest priority is given to the operation with the highest weighted tardiness increment that will result if the operation does not start immediately. The tardiness is based upon an estimate of the next possible completion time of the job.

Apparent Tardiness Cost
Operations are ranked in order of the Apparent Tardiness Cost index (Young [2]). The index may be tuned by adjusting the value of a look-ahead parameter which scales the slack time according to the expected number of competing jobs.

Cost Over Time
Operations are ranked in order of the Cost Over Time index (Young [2]). The index may be tuned by adjusting the value of a parameter which relates the processing time to the expected waiting time.

<u>Other</u>

First In First Out (FIFO)
Priorities are assigned based upon the order in which the job becomes available (i.e. the job ready time).

Random
Priorities are assigned randomly. This option is used principally as a base or control to enable comparison of the other ranking methods during experimental investigation.

Load Levelling

The machine load is defined as the number of operations competing for the machine at any particular time. Hence this provides a measure of the probability that one particular operation will be successful in seizing the machine. It makes sense to hold an operation which may be processed when the machine is lightly loaded, i.e. during periods when other operations are unable to be processed, until that time arises. This avoids the problem of processing such operations during heavily loaded periods and then leaving the machine idle due to lack of jobs later on.

For each machine the number of operations that could possible require processing is estimated for each time period. Operations are excluded if their parent job has already finished, of if their offtaker or machine is unavailable.

This information is used to estimate the average machine load during the period required to process each job to completion, for each possible starting time between the current time and the end time. The minimum of these average machine loads is then found.

The highest priority is given to the operation with the highest minimum average machine load. The greater the value of this minimum the less flexibility the operation has to be processed during lightly loaded periods.

This value may also be thought of as measuring the "price" that a particular operation is willing to "pay" in order to use the machine. An operation which has a lower minimum average load will be prepared to wait for a more lightly loaded period, when it can use the machine for a lower fee.

2.4.4 Backtracking Parameters

These parameters constrain the computational requirements of the backtracking routine.

1. Maximum Backtracking Passes

   This value places an upper limit on the number of backtracking passes possible before program termination. This prevents the program continuing if looping should occur (this has not been observed in practice and is not expected to occur). More importantly, it can control the amount of computation which occurs and hence force the program to terminate with the best solution found within a reasonable time frame. Solution times on the large mainframe on which this program runs are very fast (of the order of seconds) and hence controlling computation is not a major issue. However, this may or may not be necessary when the program is run on a PC.

   Maximum = 60, recommended = 50

2. Maximum Consecutive Passes

   This value places an upper limit on the maximum number of consecutive passes without improvement which are allowed before program execution is stopped. This stops the algorithm from continuing if a better solution cannot be found within a reasonable amount of computation (and time) from the previous best solution.

Maximum = 10, recommended = 10

3. Maximum Time Increment
This value limits the maximum period between scheduling decisions. That is, it controls the maximum time period a particular pattern is allowed to run before the algorithm reassesses job and pattern priorities and reschedules activities.

Minimum = 1 h, maximum = 99 h, default = 1 h

2.4.5 Simulation Parameters

These parameters are only used when simulation experiments are required.

1. Simulation Run Flag
If active, this flag indicates that a simulation run is required. Job input data is drawn from in-built distributions and the job input data contained in the JOBDATA file is ignored.

2. Sample Size
This controls the number of simulated schedules generated during each simulation experiment, to enable meaningful statistics to be drawn.

3. Schedule Horizon
As in normal runs, this controls the number of days the schedule covers.

Minimum = 1 day, maximum = 7 days

4. Schedule Start Day
This controls the day of the week the schedule commences from. Normally this is set to Tuesday, the principal starting day for manually prepared schedules. However, the starting day can be selected at random if desired. In this situation, the starting day is drawn from a uniform distribution of seven days.

0=random, 1=Tuesday, 2=Wednesday, etc.

5. Pipeline Load.
In simulation runs, jobs volumes are selected from a distribution until some specified estimated pipeline load is achieved. In this manner, the performance of the algorithm can be investigated at various load levels.

0=random, >1 = % load (e.g. 65 = 65 % load)

6. Job Ready Times
If active, this flag ensures that job ready times are clustered at the product batch release times, as described in Attachment 2. If inactive, the ready time of all jobs is set equal to the schedule starting time.

7. Due Date Tightness
Three options are possible. First, the due time can be offset from the job ready time by a multiple of the nominal job processing time. The nominal job processing time equals the job volume divided by the average processing rate for the particular offtaker

/ product combination. This allows specific due date tightness scenarios to be simulated. Second, the due dates can be offset randomly from the ready time. Third, the due date of all jobs is set equal to the schedule horizon.

8. Deadline Tightness
Three options are possible. First, the deadline can be offset from the job ready time by a multiple of the nominal job processing time. The nominal job processing time equals the job volume divided by the average processing rate for the particular offtaker / product combination. This allows specific due date tightness scenarios to be simulated. Second, the deadline can be offset randomly from the due time. Third, no deadline is used.

9. Random Number Seed
This is the seed (initial value) used to start the pseudo-random number generation sequence.

10. Remark
This is a comment line to allow the user to identify the simulation run.

Section 3 PROGRAM EXECUTION

3.1 Mainframe Application

The APTS program operates within the Technical system environment as part of the ISPF (Interactive System Productivity Facility) application. This operates under TSO (Time Sharing Option) on the MVS (Multiple Virtual Storage) system. Full instructions for the operation of the technical system is contained in Doran [1]. Access to the technical system is through ISPF option M.G. The menu options to edit input data, run the program and browse and print reports are self explanatory.

3.2 PC Application

This enhancement will be developed when required.

Section 4 PROGRAM OUTPUT

4.1 Reports

The main types of reports are summarized in Section 2. As discussed, these are grouped into two files which may be browsed or printed. The primary output file can be browsed or printed from the Technical System panel. The secondary output file, which is seldom required, can be browsed or printed through standard ISPF panels.

4.2 Analysis

To be developed.

4.3 Tuning

To be developed.

REFERENCES

1. Dorin R, "MOA Technical System Guide", Mobil Oil Australia internal publication, August (1985).

2. Young J A, "Algorithms for Pipeline Transfer Scheduling", Master of Applied Science thesis, Department of Mathematics and Operations Research, Victoria University of Technology, Footscray Campus, December (1992).

Attachment 1 TARDINESS WEIGHTS

The tardiness weighting factors are based on current demurrage rates incurred by the late completion of each class of jobs.

High Priority
These jobs are typically destined for marine liftings and hence, if late, cause the delay of a vessel.

Approximate vessel demurrage rate = $A 36 thousand/day

= $A 1500 /hour

Medium Priority
These jobs are typically destined for road liftings and hence, if delayed, cause the delay of a road tanker.

Approximate truck demurrage rate = $A 40 /hour

Low Priority
Jobs which fall into this class are those whose delayed completion causes no significant financial penalty. These are assigned a nominal demurrage cost of $A 1/hour to ensure that their tardiness is recognized by the objective function and minimized accordingly.

Attachment 2 EXAMPLE INPUT

2.1 System Configuration File (NETWORK)

```
DSPRT   921122-145606-SUN DSNAME M995092.TECH.DATA(APTSLINE)

TITLE  : HYPOTHETICAL REFINERY
PIPELINES            4.
PRODUCTS            15.
OFFTAKERS            9.
*
SUCTION              PL1      PL2      PL3      PL4      SOM
*
        NO.          -1.        2.       3.      -1.       4.
*
LINECLEARS           PL1      PL2      PL3      PL4      SOM
*
        PRM            1.                          1.
        CPR            1.                          1.
        ULP            2.                          2.
        SUP            2.                          2.
        AVG            2.                          2.
        ABL            2.                          2.
        JET                     1.                          1.
        JP4                     1.
        LHO                     1.
        ADO                     1.       1.
        NZA                     1.       1.
        TSA                     1.       1.
        IDO                              1.
        IFO                              1.
        LSF                              1.
*
OFFTAKERS            PL1      PL2      PL3      PL4      SOM
*
        MOB          -1.        2.       3.      -1.
        ESS          -1.        2.       3.      -1.
        CTX            1.       1.       1.       1.
        AMP            1.       2.       2.       1.
        BPA            1.       1.
        BPN            1.       1.       1.
        BPC            1.       1.
        SHL            1.       1.
        SOM                                                  1.
*
RATES                PL1      PL2      PL3      PL4      SOM
*
        MOB          175.     190.     160.     205.
        ESS          175.     190.     160.     205.
        CTX          175.     190.     160.     205.
        AMP          175.     190.     160.     205.
        BPA          175.     190.
        BPN          175.     190.     160.
        BPC          175.     190.
        SHL          175.     190.
        SOM                                                200.
```

2.2 Availability Window File (WINDOW)

```
DSPRT   921122-145626-SUN DSNAME M995092.TECH.DATA(APTSWIND)

*
PLN DAY   0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
PL1
     SUN  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     MON  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     TUE  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     WED  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     THU  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     FRI  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     SAT  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*
PLN DAY   0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
PL2
     SUN  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     MON  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     TUE  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     WED  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     THU  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     FRI  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     SAT  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*
PLN DAY   0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
PL3
     SUN  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     MON  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     TUE  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     WED  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     THU  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     FRI  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     SAT  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*
PLN DAY   0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
PL4
     SUN  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     MON  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     TUE  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     WED  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     THU  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     FRI  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     SAT  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*
PLN DAY   0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
SOM
     SUN  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     MON  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     TUE  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     WED  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     THU  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     FRI  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
     SAT  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
*
OFT DAY   0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2
```

```
                0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
MOB
     SUN        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     MON        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     TUE        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     WED        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     THU        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     FRI        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     SAT        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
*
OFT  DAY        0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  2  2  2  2
                0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
ESS
     SUN
     MON                       1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     TUE                       1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     WED                       1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     THU                       1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     FRI                       1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     SAT
*
OFT  DAY        0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  2  2  2  2
                0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
CTX
     SUN                                                                          1
     MON        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     TUE        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     WED        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     THU        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     FRI        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     SAT        1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
*
OFT  DAY        0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  2  2  2  2
                0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
AMP
     SUN
     MON                       1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     TUE                       1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     WED                       1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     THU                       1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     FRI                       1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2
     SAT
*
OFT  DAY        0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  2  2  2  2
                0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
BPA
     SUN                          1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
     MON        2  2  2           1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
     TUE        2  2  2           1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
     WED        2  2  2           1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
     THU        2  2  2           1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
     FRI        2  2  2           1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
     SAT        2  2  2           1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
*
OFT  DAY        0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  2  2  2  2
                0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
BPN
```

```
     SUN                                 1  1  1  1  1  1  1  1
     MON                                 1  1  1  1  1  1  1  1  2  2  2  2
     TUE                                 1  1  1  1  1  1  1  1  2  2  2  2
     WED                                 1  1  1  1  1  1  1  1  2  2  2  2
     THU                                 1  1  1  1  1  1  1  1  2  2  2  2
     FRI                                 1  1  1  1  1  1  1  1  2  2  2  2
     SAT                                 1  1  1  1  1  1  1  1
*
OFT  DAY  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  2  2  2  2
          0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
BPC
     SUN                                 1  1  1  1  1  1  1  1
     MON                                 1  1  1  1  1  1  1  1  2  2  2  2
     TUE                                 1  1  1  1  1  1  1  1  2  2  2  2
     WED                                 1  1  1  1  1  1  1  1  2  2  2  2
     THU                                 1  1  1  1  1  1  1  1  2  2  2  2
     FRI                                 1  1  1  1  1  1  1  1  2  2  2  2
     SAT                                 1  1  1  1  1  1  1  1
*
OFT  DAY  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  2  2  2  2
          0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
SHL
     SUN  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     MON  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     TUE  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     WED  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     THU  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     FRI  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     SAT  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
*
OFT  DAY  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  2  2  2  2
          0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
SOM
     SUN  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     MON  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     TUE  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     WED  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     THU  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     FRI  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
     SAT  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

2.3 Job Definition File (JOBDATA)

```
DSPRT   921122-145546-SUN DSNAME M995092.TECH.DATA(APTSJOBS)

REMARK: JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE
*         DAY        DATE     MNTH     MAXD     NDAY
DATES  :  TUE         30.       3.      31.       5.
*
DETAILS:OFT    PRD    BCH    TNK    PTY      VOL      RDY      DUE      DEAD
*
                ULP                  PL4       .        .        .         .
                ADO                  PL3       .        .        .         .
          CTX   ULP    P10    ???    PL1      3.5    29.22     1.16       .
          MOB   JET    J20           PL2      5.5    29.18     1.23       .
          ESS   PRM    P08                    3.0    30.05     1.00       .
          CTX   ADO    A52                    3.5    30.00    31.12       .
          AMP   JET    J21                    1.5    30.12     1.05       .
          SHL   JET    J21                    2.0    30.12     1.12       .
          MOB   PRM    P08                    2.3    30.05     1.00       .
          MOB   ULP                           4.0    30.03     1.17       .
          MOB   ADO                           2.7    30.07     1.15       .
          ESS   ADO                           1.8    30.07     1.15       .
          CTX   PRM                           2.0    30.01     1.12       .
          CTX   CPR                           2.3    30.00     1.23     .
          AMP   CPR                           2.0    30.00     1.23     .
          BPN   PRM                           1.5    30.12     1.23     .
          BPN   ULP                           0.5    30.03    30.23      1.12
*         SOM   JET                           8.0    30.01     1.23     .
```

2.4 Run Definition File (RUNDATA)

```
DSPRT   921122-145816-SUN DSNAME M995092.TECH.DATA(APTSUSER)

*
REPORTS                      (1=Y,0=N)
*
PIPELINE SYSTEM DATA              1.
JOB INPUT DATA                    1.
PIPELINE STATUS                   1.
JOB SCREENING                     1.
JOB QUEUES                        1.      (1=SORTED,2=UNSORTED+SORTED)
PATTERN GENERATION                1.
PATTERN QUEUES                    1.      (1=SORTED,2=UNSORTED+SORTED)
PIPELINE SCHEDULE                 1.      (1=BEST,2=BETTER,3=ALL)
JOB SUMMARY                       1.      (1=BEST,2=BETTER,3=ALL)
AVAILABILITY SUMMARY              1.      (1=BEST,2=BETTER,3=ALL)
PIPELINE UTILIZATION              1.
BACKTRACK SUMMARY                 1.
TARDINESS PROFILE                 1.
PERFORMANCE GRAPH                 0.
TARDINESS VS UTILIZ.              0.
VOL REMAIN VS UTILIZ.             0.
JOBS/VOLUME VS SAMPLE             0.
SIMULATION STATISTICS             0.
*
OBJ FUNCTION WEIGHTING
*
WEIGHTED TARDINESS                1.
SETUP COUNT                       0.
MAKESPAN                          0.
OFFTAKER OVERTIME                 0.
*
JOB PRIORITY WEIGHTING       ($/H)
*
HIGH                           1500.
MEDIUM                           40.
LOW                               1.
*
PATTERN PRIORITY WEIGHTING
*
VOLUME TRANSFERRED                0.              0.
PATTERN DURATION                  0.              0.
UTILIZATION                     100.            100.
GENERATION ORDER                  1.              1.
AVERAGE JOB VALUE                 0.              0.
*
JOB PREEMPTION TUNING        (HOURS)
*
SIBLING WORK REMAINING            4.              4.
JOB WORK REMAINING                6.              6.
MINIMUM WORK REMAINING            4.              4.
CRITICAL: DUE TIME               12.             12.
CRITICAL: DEADLINE               24.             24.
*
JOB PRIORITY ADJUSTMENT      (HOURS)
*
NEW JOB                           1.              1.
OLD JOB (ALREADY ONLINE)          4.              4.
NON-PREEMPTABLE JOB             500.            500.
CRITICAL JOB                    250.            250.
```

```
SIBLING JOB SWITCH          -250.          -250.
JOB AVERAGE CAPACITY           0.       (1=Y,0=N)
*
JOB SCHEDULING RULE SELECTION  (1=Y,0=N)
*
RANDOM                         0.
FIRST-COME-FIRST-SERVED        0.
SHORTEST PROC TIME             1.
LONGEST PROC TIME              0.
LONGEST OPERATION              0.
EARLIEST DUE DATE              0.
MODIFIED DUE DATE              0.
MINIMUM SLACK (DUE DATE)       0.
MINIMUM SLACK (DEADLINE)       0.
MODIFIED SLACK                 0.
APPARENT TARDINESS COST        0.
COST OVER TIME                 0.
WEIGHTED TARDINESS             0.
TARDINESS DELTA                0.
LOAD LEVELLING                 0.
*
JOB SCHEDULING RULE PARAMETERS
*
APPARENT TARD COST K           3.0            3.0
COST OVER TIME K               3.0            2.0
COST OVER TIME B               1.0            2.0
*
HEURISTIC SELECTION
*
PIPELINE IDLING                0.
LOOK AHEAD                     0.
*
BACKTRACKING PARAMETERS
*
MAX BACKTRACKING PASSES       50.
MAX CONSECUTIVE PASSES        10.
MAXIMUM TIME INCREMENT         1.
*
SIMULATION
*
SIMULATION RUN                 0.      (1=Y,0=N)
SAMPLE SIZE                   40.      (100 MAX)
SCHEDULE HORIZON (DAYS)        7.      (1<=HORIZON<=7)
SCHEDULE START DAY             1.      (0=RANDOM,1=TUE,2=WED,ETC)
PIPELINE LOAD                 80.      (0=RANDOM,>1=% LOAD)
READY TIMES                    1.      (0=NO,1=RANDOM)
DUE DATE TIGHTNESS           300.      (0=NO,1=RANDOM,>1=% TIGHTNESS)
DEADLINE TIGHTNESS             0.      (0=NO,1=RANDOM,>1=% TIGHTNESS)
RANDOM NUMBER SEED             5.      (65539.0)
REMARK: 09 NOV 92    ATC (K*B=3.0)      DUE.T=300, UTZ=80, PAT UTZ
```

Attachment 3 EXAMPLE OUTPUT

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

CASE: 1 PASS: 1 VALUE: 1520 *** P I P E L I N E S C H E D U L E ***

PIPE LINE	TUE 30 MAR 4	8	12	16	20	24	WED 31 MAR 4	8	12	16	20	24	THU 1 APR 4	8	12	16	20	24	FRI 2 APR 4	8	12	16	20	24	SAT 3 APR 4	8	12	16	20	24
PL1	< 03	10	17	#	09		16						05			# 14														
	<CTX	MOB	BPN	#	MOB		BPN						ESS			# CTX														
	<ULP	ULP	ULP	#	PRM		PRM						PRM			# CPR														
180	<900>	1600	500	#	2300		1600						500			# 900>														
PL4		10	03	#	13		05			## 03			#05#	15		14														
		MOB	CTX	#	CTX		ESS			## CTX			#SS#	AMP		CTX														
		ULP	ULP	#	PRM		PRM			## ULP			#RM#	CPR		CPR														
210		2500	1000>	#	2100		2100	>		##1200			#40#	2100		1400														
PL2	< 04	07	08		04		08					# 06																		
	< MOB	AMP	SHL		MOB		SHL					# CTX																		
	< JET	JET	JET		JET		JET					# ADO																		
190	< 2100	> 1500	1000>		2300		1100					# 2100																		
PL3		06	11		12																									
		CTX	MOB		ESS																									
		ADO	ADO		ADO																									
160		1400	> 2700		1800																									
+STEP+			C			C			C		C		C		C			C			X									

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

CASE: 1 PASS: 1 VALUE: 1520 *** T R A N S F E R S U M M A R Y ***

JOB NUMBER	OFFT NAME	PROD NAME	BATCH NUMBER	TANK NUMBER	PRIOR LEVEL	READY TIME	DUE TIME	DEAD LINE	START TIME	FINISH TIME	TARD-INESS	VOL REQD	VOL DELIV	VOL REM	PARCEL COUNT
						DD.HH	DD.HH	DD.HH	DD.HH	DD.HH	H	ML	ML	ML	
1		ULP	EXC		PL4	0.00	0.00	0.00	0.00	0.00		0.0	0.0	0.0	
2		ADO	EXC		PL3	0.00	0.00	0.00	0.00	0.00		0.0	0.0	0.0	
3	CTX	ULP	P10	???	PL1	29.22	1.16	0.00	29.22	1.02		3.5	3.5	0.0	3
4	MOB	JET	J20		PL2	29.18	1.23	0.00	29.18	31.12		5.5	5.5	0.0	2
5	ESS	PRM	P08			30.05	1.00	0.00	31.07	1.08	8	3.0	3.0	0.0	2
6	CTX	ADO	A52			30.00	31.12	0.00	30.06	1.14	26	3.5	3.5	0.0	2
7	AMP	JET	J21			30.12	1.05	0.00	30.12	30.19		1.5	1.5	0.0	1
8	SHL	JET	J21			30.12	1.12	0.00	30.20	31.18		2.0	2.0	0.0	2
9	MOB	PRM	P08			30.05	1.00	0.00	30.20	31.09		2.3	2.3	0.0	1
10	MOB	ULP				30.03	1.17	0.00	30.03	30.14		4.0	4.0	0.0	1
11	MOB	ADO				30.07	1.15	0.00	30.15	31.07		2.7	2.7	0.0	1
12	ESS	ADO				30.07	1.15	0.00	31.08	31.18		1.8	1.8	0.0	1
13	CTX	PRM				30.01	1.12	0.00	30.20	31.06		2.0	2.0	0.0	1
14	CTX	CPR				30.00	1.23	0.00	1.15	2.03	4	2.3	2.3	0.0	2
15	AMP	CPR				30.00	1.23	0.00	1.09	1.19		2.0	2.0	0.0	1
16	BPN	PRM				30.12	1.23	0.00	31.10	31.18		1.5	1.5	0.0	1
17	BPN	ULP				30.03	30.23	1.12	30.15	30.17		0.5	0.5	0.0	1
TOTAL											38	38.1	38.1	0.0	16

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

CASE: 1 PASS: 1 VALUE: 1520 *** A V A I L A B I L I T Y S U M M A R Y ***

PIPE	TUE 30 MAR						WED 31 MAR						THU 1 APR						FRI 2 APR						SAT 3 APR					
LINE	4	8	12	16	20	24	4	8	12	16	20	24	4	8	12	16	20	24	4	8	12	16	20	24	4	8	12	16	20	24
PL1																														
PL4																														
PL2																														
PL3																														
MOB																														
ESS	XXXXX						0000XXXXXXXXX						$$00XXXXXXXXX						0000XXXXXXXXX						0000XXXXXXXXXXXXXXXXXXXXXXXXXXX					
CTX																									0000XXXXXXXXX					
AMP	XXXXXX						0000XXXXXXX						0000XXXXXXX						0000XXXXXXX						0000XXXXXXXXXXXXXXXXXXXXXXXXX					
BPA	000XXXXX						0000XXXXX						0000XXXXX						0000XXXXX						0000XXXXX					X
BPN	XXXXXXXX						$000XXXXXXXXXXXX						$$00XXXXXXXXXXX						0000XXXXXXXXXXX						0000XXXXXXXXXXX					XXXXXXXX
BPC	XXXXXXXX						0000XXXXXXXXXXX						0000XXXXXXXXXXX						0000XXXXXXXXXXX						0000XXXXXXXXXXX					XXXXXXXX
SHL																														
SOM																														

X=UNAVAILABLE 0=AVAILABLE ON OVERTIME $=OVERTIME USED

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

CASE: 1 PASS: 1 VALUE: 1520 *** P I P E L I N E U T I L I Z A T I O N ***

PIPE LINE	TRAN TIME	SETUP TIME	IDLE TIME	CONTN TIME	DOWN TIME	SPARE TIME	TOTAL TIME	UTILN	ONLINE RATE	AVG RATE	IDLE CAPY	SPARE CAPY	OPERN COUNT	SETUP COUNT	CONTN COUNT
	H	H	H	H	H	H	H	%	KL/H	KL/H	ML	ML			
PL1	47	2	4	22		45	120	39	175	68	1	83	7	2	4
PL4	62	5	1	7		45	120	51	205	105		97	8	4	4
PL2	53	1	1	7		58	120	44	190	83		115	6	1	1
PL3	37		3	22		58	120	30	160	49		97	3		3
TOTAL	199	8	9	58		206	480	42	185		2	387	24	7	12

REPORT SYSTEM *** P I P E L I N E S Y S T E M R E P O R T ***

1. SCHEDULING PERIOD

	DAY	MONTH	DATE	WDAY	SDAY
1	TUE	MAR	30	3	6
2	WED	MAR	31	4	7
3	THU	APR	1	5	1
4	FRI	APR	2	6	2
5	SAT	APR	3	7	3
6	SUN	APR	4	1	4
7	MON	APR	5	2	5

2. SUCTION PIPELINE LINEUPS

	PIPELINE			
	PL1	PL2	PL3	PL4
P/L	-1	2	3	-1

3. LINECLEAR TIMES H

PRODUCT	PIPELINE			
	PL1	PL2	PL3	PL4
PRM	1			1
CPR	1			1
ULP	2			2
SUP	2			2
AVG	2			2
ABL	2			2
JET		1		
JP4		1		
LHO		1		
ADO		1	1	
NZA		1	1	
TSA		1	1	
IDO			1	
IFO			1	
LSF			1	

4. OFFTAKER LINEUPS

OFFTAKER	PIPELINE PL1	PL2	PL3	PL4
MOB	-1	2	3	-1
ESS	-6	7	8	-6
CTX	11	11	11	11
AMP	16	17	17	16
BPA	21	21		
BPN	26	26	26	
BPC	31	31		
SHL	36	36		
SOM				

5. PUMPING RATES M3/H

OFFTAKER	PIPELINE PL1	PL2	PL3	PL4
MOB	175	190	160	205
ESS	175	190	160	205
CTX	175	190	160	205
AMP	175	190	160	205
BPA	175	190		
BPN	175	190	160	
BPC	175	190		
SHL	175	190		
SOM				

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

REPORT JOBINP *** T R A N S F E R I N P U T D E T A I L S ***

JOB NUMBER	OFFT NAME	PROD NAME	BATCH NUMBER	TANK NUMBER	PRIOR LEVEL	VOL REQD	READY TIME	DUE TIME	DEAD LINE	OFFT NUMBER	PROD NUMBER	READY TIME	DUE TIME	DEAD LINE
						ML	DD.HH	DD.HH	DD.HH			H	H	H
1		ULP			PL4	0.0	0.00	0.00	0.00	10	3	0	0	0
2		ADO			PL3	0.0	0.00	0.00	0.00	10	10	0	0	0
3	CTX	ULP	P10	???	PL1	3.5	29.22	1.16	0.00	3	3	-2	64	0
4	MOB	JET	J20		PL2	5.5	29.18	1.23	0.00	1	7	-6	71	0
5	ESS	PRM	P08			3.0	30.05	1.00	0.00	2	1	5	48	0
6	CTX	ADO	A52			3.5	30.00	31.12	0.00	3	10	0	36	0
7	AMP	JET	J21			1.5	30.12	1.05	0.00	4	7	12	53	0
8	SHL	JET	J21			2.0	30.12	1.12	0.00	8	7	12	60	0
9	MOB	PRM	P08			2.3	30.05	1.00	0.00	1	1	5	48	0
10	MOB	ULP				4.0	30.03	1.17	0.00	1	3	3	65	0
11	MOB	ADO				2.7	30.07	1.15	0.00	1	10	7	63	0
12	ESS	ADO				1.8	30.07	1.15	0.00	2	10	7	63	0
13	CTX	PRM				2.0	30.01	1.12	0.00	3	1	1	60	0
14	CTX	CPR				2.3	30.00	1.23	0.00	3	2	0	71	0
15	AMP	CPR				2.0	30.00	1.23	0.00	4	2	0	71	0
16	BPN	PRM				1.5	30.12	1.23	0.00	6	1	12	71	0
17	BPN	ULP				0.5	30.03	30.23	1.12	6	3	3	23	60

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

REPORT OPNINP *** J O B I N P U T D E T A I L S ***

JOB NUMBER	TRNSFR NUMBER	OFFT NUMBER	PROD NUMBER	MAIN P/L	OFFT P/L	SUCT P/L	PUMP RATE M3/H	TIME REQD H
1	3	3	3	1	11	-1	175	18
2	3	3	3	4	11	-1	205	17
3	4	1	7	2	2	2	190	23
4	5	2	1	1	-6	-1	175	17
5	5	2	1	4	-6	-1	205	15
6	6	3	10	2	11	2	190	18
7	6	3	10	3	11	3	160	22
8	7	4	7	2	17	2	190	8
9	8	8	7	2	36	2	190	11
10	9	1	1	1	-1	-1	175	13
11	9	1	1	4	-1	-1	205	11
12	10	1	3	1	-1	-1	175	23
13	10	1	3	4	-1	-1	205	20
14	11	1	10	2	2	2	190	14
15	11	1	10	3	3	3	160	17
16	12	2	10	2	7	2	190	9
17	12	2	10	3	8	3	160	11
18	13	3	1	1	11	-1	175	11
19	13	3	1	4	11	-1	205	10
20	14	3	2	1	11	-1	175	13
21	14	3	2	4	11	-1	205	11
22	15	4	2	1	16	-1	175	11
23	15	4	2	4	16	-1	205	10
24	16	6	1	1	26	-1	175	9
25	17	6	3	1	26	-1	175	3
TOTAL							185	345

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

CASE: 1 PASS: 1 TIME: 12 *** P I P E L I N E S T A T U S ***

P/L	CURRT JOB	NONPRE JOB	SETUP TIME	MIN TIME	CURRT PROD	LAST PROD	OFFT P/L	SUCT P/L	JOB STATUS	P/L
			H	H						
PL1		10			ULP	ULP	-1	-1	ANAF	PL1
PL2	4				JET	JET	2	2	ANMT	PL2
PL3	6				ADO	ADO	11	3	ANMT	PL3
PL4		10			ULP	ULP	-1	-1	ANAF	PL4

CASE: 1 PASS: 1 TIME: 12 *** J O B C O N T E N T I O N T E S T I N G ***

OPN #	JOB #	VOL REM	SETUP TIME	MIN TIME	JOB TIME	SLACK TIME	OTHER JOBS	SORT VALUE	STATUS	STOP FLAG	NEXT START	OPN #
		KL	H	H	H	H	*100				H	
1	3	2275		4	13				PPLN	1		1
2	3	2275		4	11				PPLN	1		2
3	4	2270			12	47		-30	ANMT			3
4	5	3000	1	5	18				PSPL	1		4
5	5	3000	1	5	15				PSPL	1		5
6	6	2540	1	5	14	10		85	NCDU			6
7	6	2540			16	8			ANMT			7
8	7	1500		4	7	21		31				8
9	8	2000		4	10	38		54				9
10	9	2300	1	5	14				PSPL	1		10
11	9	2300	1	5	12				PSPL	1		11
12	10	1105			3	50		-99	ANAF			12
13	10	1105			3	50		-99	ANAF			13
14	11	2699	1	5	15	36		92				14
15	11	2699		4	16	35		100				15
16	12	1800	1	5	10	23		54				16
17	12	1800		4	11	22		62				17
18	13	2000	1	5	12				PSPL	1		18
19	13	2000	1	5	10				PSPL	1		19
20	14	2300	1	5	14				PSPL	1		20
21	14	2300	1	5	12				PSPL	1		21
22	15	2000	1	5	12				PSPL	1		22
23	15	2000	1	5	10				PSPL	1		23
24	16	1500	1	5	9				PSPL	1		24
25	17	500		2	2				PPLN	1		25

CASE: 1 PASS: 1 TIME: 12 *** S O R T E D J O B Q U E U E ***

COUNT	1	2	3	4	5	6	7	8	9	10	11
JOB	10	10	4	6	7	12	8	12	6	11	11
P/L	PL4	PL1	PL2	PL3	PL2	PL2	PL2	PL3	PL2	PL2	PL3

VALUE -99 -99 -99 0 31 54 54 62 85 92 100

CASE: 1 PASS: 1 TIME: 12 *** P A T T E R N G E N E R A T I O N ***

TRIAL #	OPN #	JOB #	OFFT #	PROD #	MAIN P/L	OFFT P/L	SUCT P/L	MAIN JOB	OFFT JOB	SUCT JOB	STATUS	OPN #
1	13	10	MOB	ULP	PL4	1	1	10	10	10	PL4	13
1	12	10	MOB	ULP	PL1	1	1	10	10	10	PL1	12
1	3	4	MOB	JET	PL2	2	2	4	4	4	PL2	3
1	7	6	CTX	ADO	PL3	11	3	6	6	6	PL3	7
2	8	7	AMP	JET	PL2	17	2	7	7	7	PL2	8
2	7	6	CTX	ADO	PL3	11	3	6	6	6	PL3	7
2	3	4	MOB	JET	PL2	2	2	7		7	SPLN	3
2	12	10	MOB	ULP	PL1	1	1	10	10	10	PL1	12
2	13	10	MOB	ULP	PL4	1	1	10	10	10	PL4	13
3	16	12	ESS	ADO	PL2	7	2	12	12	12	PL2	16
3	8	7	AMP	JET	PL2	17	2	12		12	SPLN	8
3	7	6	CTX	ADO	PL3	11	3	6	6	6	PL3	7
3	3	4	MOB	JET	PL2	2	2	12		12	SPLN	3
3	12	10	MOB	ULP	PL1	1	1	10	10	10	PL1	12
3	13	10	MOB	ULP	PL4	1	1	10	10	10	PL4	13
4	9	8	SHL	JET	PL2	36	2	8	8	8	PL2	9
4	16	12	ESS	ADO	PL2	7	2	8		8	SPLN	16
4	8	7	AMP	JET	PL2	17	2	8		8	SPLN	8
4	7	6	CTX	ADO	PL3	11	3	6	6	6	PL3	7
4	3	4	MOB	JET	PL2	2	2	8		8	SPLN	3
4	12	10	MOB	ULP	PL1	1	1	10	10	10	PL1	12
4	13	10	MOB	ULP	PL4	1	1	10	10	10	PL4	13
5	17	12	ESS	ADO	PL3	8	3	12	12	12	PL3	17
5	9	8	SHL	JET	PL2	36	2	8	8	8	PL2	9
5	16	12	ESS	ADO	PL2	7	2	8		8	SPLN	16
5	8	7	AMP	JET	PL2	17	2	8		8	SPLN	8
5	7	6	CTX	ADO	PL3	11	3	12		12	SPLN	7
5	3	4	MOB	JET	PL2	2	2	8		8	SPLN	3
5	12	10	MOB	ULP	PL1	1	1	10	10	10	PL1	12
5	13	10	MOB	ULP	PL4	1	1	10	10	10	PL4	13
6	6	6	CTX	ADO	PL2	11	2	6	6	6	PL2	6
6	17	12	ESS	ADO	PL3	8	3	12	12	12	PL3	17
6	9	8	SHL	JET	PL2	36	2	6		6	SPLN	9
6	16	12	ESS	ADO	PL2	7	2	6		6	SPLN	16
6	8	7	AMP	JET	PL2	17	2	6		6	SPLN	8
6	7	6	CTX	ADO	PL3	11	3	12	6	12	SPLN	7
6	3	4	MOB	JET	PL2	2	2	6		6	SPLN	3
6	12	10	MOB	ULP	PL1	1	1	10	10	10	PL1	12
6	13	10	MOB	ULP	PL4	1	1	10	10	10	PL4	13
7	14	11	MOB	ADO	PL2	2	2	11	11	11	PL2	14
7	6	6	CTX	ADO	PL2	11	2	11		11	SPLN	6
7	17	12	ESS	ADO	PL3	8	3	12	12	12	PL3	17
7	9	8	SHL	JET	PL2	36	2	11		11	SPLN	9
7	16	12	ESS	ADO	PL2	7	2	11		11	SPLN	16
7	8	7	AMP	JET	PL2	17	2	11		11	SPLN	8

7	7	6	CTX	ADO	PL3	11	3	12		12	SPLN	7
7	3	4	MOB	JET	PL2	2	2	11	11	11	SPLN	3
7	12	10	MOB	ULP	PL1	1	1	10	10	10	PL1	12
7	13	10	MOB	ULP	PL4	1	1	10	10	10	PL4	13

CASE: 1 PASS: 1 TIME: 12 *** S O R T E D P A T T E R N S ***

TRIAL	VOLUM	LNGTH	VALUE	PL1	PL2	PL3	PL4
2	12	3		10	7	6	10
1	12	3		10	4	6	10
4	12	3		10	8	6	10
5	12	3	-1	10	8	12	10
3	11	3	-99	10	12	6	10
7	11	3	-100	10	11	12	10
6	11	3	-100	10	6	12	10

CASE: 1 PASS: 1 TIME: 12 H JOBS CHOSEN: 10 7 6 10 0 TOLD: 11 TNEW: 12 TINC: 1
CASE: 1 PASS: 1 TIME: 75 *** S O R T E D P A T T E R N S ***

TRIAL	VOLUM	LNGTH	VALUE	PL1	PL2	PL3	PL4
1		1	-1	58	58	58	14

CASE: 1 PASS: 1 TIME: 75 H JOBS CHOSEN: 58 58 58 14 0 TOLD: 74 TNEW: 75 TINC: 1

A CASE 1 PASS 1 STOPPED AT 75 H BECAUSE ALL JOBS FINISHED

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

CASE: 1 PASS: 1 VALUE: 1520 *** B A C K T R A C K S U M M A R Y ***

CURRENT BACKTRACK PASS	1
OBJECTIVE VALUE, CURRENT PASS	1520
OBJECTIVE VALUE, PREVIOUS BEST	999999

LATEST PASS BETTER THAN PREVIOUS BEST: UPDATING SCHEDULE

OLD BACKTRACK POINT	1
OLD BACKTRACK TIME	0
OLD BACKTRACK PATTERN	0
NEW BACKTRACK POINTS IDENTIFIED	6
TOTAL BACKTRACK POINTS AVAILABLE	7

BACKTRACK POINT	1	2	3	4	5	6	7
BACKTRACK TIME	0	31	40	43	51	57	69
TARDINESS	0	0	120	240	640	1120	1360
PATTERN COUNT	0	8	3	6	4	5	2
LAST PATTERN USED	0	1	1	1	1	1	1

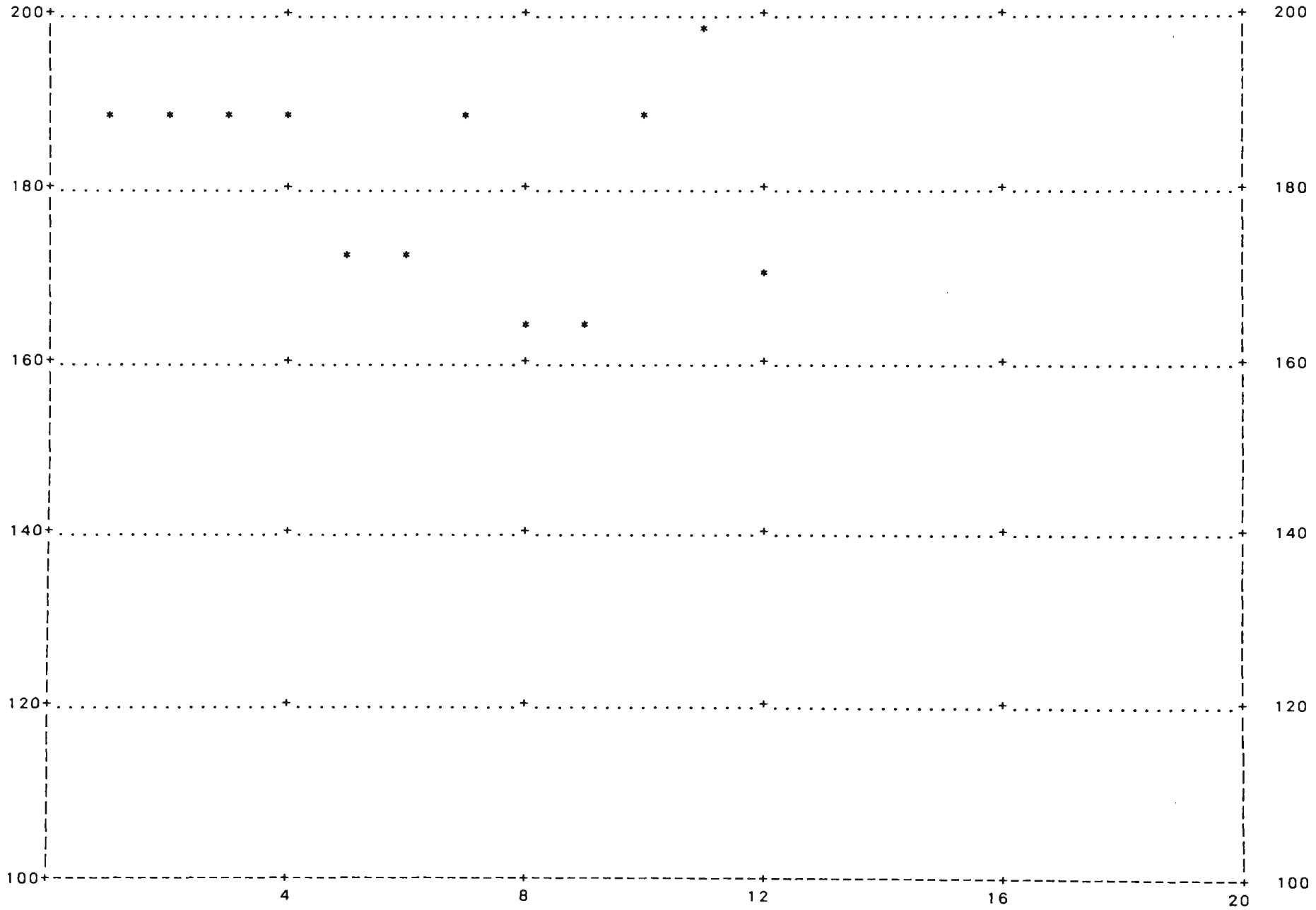
NEW BACKTRACK POINT	7
NEW BACKTRACK TIME	69
NEW BACKTRACK PATTERN	2

PIPELINE	PL1	PL2	PL3	PL4
JOB NUMBER	14	58	58	57

CASE: 1 PASS: 2 TIME: 69 H JOBS CHOSEN: 14 58 58 57 0 TOLD: 68 TNEW: 69 TINC: 1

HYPOTHETICAL REFINERY JAYOUNG, 22 NOV 92, USER GUIDE EXAMPLE

REPORT 0 *** P E R F O R M A N C E H I S T O R Y ***



*** S I M U L A T I O N S T A T I S T I C S ***

CASE	TOTAL JOBS	UNFIN JOBS	PASSES	MAKE-SPAN H	FIRST VALUE K$	BEST VALUE K$	TOTAL TARDI H	MAX TARDI H	NORM. WTAR $	VOL RQD ML	VOL REM ML	SETUP COUNT	P/L UTILN %	CASE
1	45.	11.	42.	168.	32.0	31.3	783.	115.	13.8	102.5	33.0	20.	59.	1
2	36.	8.	12.	168.	22.0	22.0	550.	123.	12.4	103.5	28.7	14.	63.	2
3	47.	11.	19.	168.	34.4	34.2	854.	115.	14.6	106.5	27.8	24.	67.	3
4	44.	19.	7.	168.	57.4	57.2	1429.	155.	24.2	102.5	46.0	11.	48.	4
5	40.	19.	33.	168.	59.1	57.5	1438.	123.	33.0	104.5	55.2	7.	42.	5
6	37.	11.	36.	168.	40.6	35.9	898.	115.	17.0	102.0	28.0	11.	62.	6
7	47.	15.	31.	168.	41.4	36.4	909.	135.	13.7	106.5	40.8	11.	56.	7
8	30.	13.	2.	168.	46.0	46.0	1151.	135.	29.6	102.0	40.5	21.	52.	8
9	43.	21.	12.	168.	72.5	72.5	1812.	123.	31.4	101.0	46.5	9.	47.	9
10	44.	20.	10.	168.	70.1	69.9	1748.	131.	26.2	103.0	39.1	13.	54.	10
11	40.	20.	8.	168.	43.4	43.4	1084.	135.	25.0	104.5	67.3	5.	32.	11
12	39.	20.	8.	168.	54.4	54.4	1360.	155.	21.9	103.5	45.5	9.	49.	12
13	40.	13.	15.	168.	52.7	50.6	1266.	147.	20.6	107.0	33.2	16.	62.	13
14	44.	28.	14.	168.	98.2	98.2	2455.	155.	41.0	103.5	64.0	13.	34.	14
15	44.	20.	12.	168.	39.4	39.4	986.	123.	17.2	105.0	59.0	10.	39.	15
16	38.	15.	20.	168.	44.2	43.9	1098.	147.	19.9	107.0	38.1	12.	57.	16
17	44.	16.	5.	168.	50.0	50.0	1251.	155.	18.8	103.5	35.3	11.	58.	17
18	46.	24.	12.	168.	67.6	67.6	1689.	135.	26.2	108.5	55.6	8.	44.	18
19	39.	19.	21.	168.	34.7	34.4	860.	143.	17.6	107.5	48.1	10.	49.	19
20	42.	20.	19.	168.	58.2	58.1	1453.	131.	25.9	101.0	48.5	10.	44.	20
21	43.	24.	1.	168.	70.7	70.7	1768.	143.	33.2	103.0	59.0	14.	38.	21
22	45.	17.	12.	168.	66.4	66.4	1659.	155.	27.2	100.5	40.8	20.	52.	22
23	43.	18.	8.	168.	50.4	50.4	1261.	155.	21.9	103.0	44.0	16.	50.	23
24	39.	19.	5.	168.	57.8	57.8	1445.	127.	29.2	102.0	51.5	6.	43.	24
25	45.	15.	24.	168.	55.0	54.0	1350.	147.	22.3	103.5	47.5	16.	47.	25
26	43.	13.	19.	168.	49.4	49.4	1235.	135.	18.5	109.5	38.7	10.	60.	26
27	49.	18.	12.	168.	50.9	50.9	1272.	135.	18.0	104.5	43.0	9.	53.	27
28	42.	14.	18.	168.	36.0	35.8	894.	135.	13.6	104.5	34.0	5.	59.	28
29	48.	18.	29.	168.	69.4	65.8	1645.	143.	23.6	104.0	43.5	10.	51.	29
30	39.	24.	8.	168.	62.2	60.3	1507.	143.	33.6	108.0	77.5	12.	26.	30
31	48.	12.	51.	168.	51.5	47.4	1186.	147.	16.3	105.0	28.0	12.	65.	31
32	47.	14.	21.	168.	42.4	42.2	1055.	143.	17.1	102.5	30.2	18.	61.	32
33	38.	21.	2.	168.	72.5	72.5	1812.	155.	30.1	102.0	48.0	12.	45.	33
34	44.	18.	8.	168.	62.5	62.5	1562.	143.	27.3	107.0	43.2	10.	54.	34
35	43.	20.	19.	168.	57.4	57.4	1434.	143.	24.5	104.0	44.1	8.	50.	35
36	40.	19.	11.	168.	58.0	58.0	1451.	155.	25.5	103.5	53.0	11.	44.	36
37	39.	17.	12.	168.	55.5	52.5	1312.	143.	22.6	114.0	48.4	10.	56.	37
38	37.	19.	13.	168.	47.1	46.9	1173.	147.	24.0	104.0	57.2	5.	39.	38
39	43.	16.	20.	168.	42.0	38.6	965.	131.	16.5	108.0	47.6	8.	50.	39
40	38.	19.	10.	168.	45.0	45.0	1124.	115.	16.0	106.5	44.5	14.	51.	40
AVERAGE	42.0	17.4	16.0	168.0	53.0	52.2	1304.6	138.4	22.8	104.6	45.1	11.8	50.2	
STD DEV	3.9	4.2	10.9	0.0	14.2	14.4	360.5	12.7	6.6	2.7	11.2	4.4	9.3	
MINIMUM	30.0	8.0	1.0	168.0	22.0	22.0	550.0	115.0	12.4	100.5	27.8	5.0	26.5	
MAXIMUM	49.0	28.0	51.0	168.0	98.2	98.2	2455.0	155.0	41.0	114.0	77.5	24.0	66.6	