This is to certify that the

dissertation entitled

PRODUCTION SCHEDULING OF CUTTING STOCK
IN A MULTI-PERIOD, MULTI-PROCESS FACILITY
WITH SEQUENCE DEPENDENT SETUPS

presented by

Michael P. D'Itri

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Business Administration

Paul A. Rubin

Major professor

Date June 29, 1994

PRODUCTION SCHEDULING OF CUTTING STOCK
IN A MULTI-PERIOD, MULTI-PROCESS FACILITY
WITH SEQUENCE DEPENDENT SETUPS

By

Michael Payne D'Itri

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Management

1994

ABSTRACT

PRODUCTION SCHEDULING OF CUTTING STOCK IN A MULTI-PERIOD, MULTI-PROCESS FACILITY WITH SEQUENCE DEPENDENT SETUPS

By

Michael P. D'Itri

This paper introduces an explicit formulation for sequence dependent production scheduling where the produced stock will be cut into finished products and a heuristic solution procedure that obtains good, but not necessarily optimal, solutions to a production scheduling problem common in the paper industry. The heuristic begins by adding additional constraints and a few binary variables that limit the number of decision variables likely to take fractional values when the formulation is solved with the integrality restrictions relaxed. Next, the heuristic proceeds to sequence production on each machine one period at time. Sequencing is accomplished with an iterative procedure that successively increases the number of transition variables required to be binary until a complete production sequence, for that period, is achieved. Once a production sequence is established for a period, the heuristic defines a range of integers describing possible production quantities for that period. Another mixed integer program is formulated that establishes integer values for the production in the period just sequenced and begins the sequencing for the next period.

This dissertation is dedicated to my grandparents
Payne and Catherine Ward

ACKNOWLEDGMENTS

I would like to thank the many individuals who, through their generous contributions of time and expertise, have made the completion of this dissertation possible. First and foremost, I would like to thank the chairman of my dissertation committee, Professor Paul Rubin. It has been an honor and a privilege to have such an accomplished scholar as a mentor, not only for this project but throughout my graduate career at Michigan State University. His insights, suggestions and patient tutoring have made the completion of this research possible. The other members of my dissertation committee, Professor Soumen Ghosh, Professor Gary Ragatz and Professor Shawnee Vickery have all made crucial contributions to this research as well as my development as a scholar.

Special appreciation is extended to Priya Campbell and the *Simpson Plainwell Paper Company*. Other important contributions to this research have been made by Mr. Michael Wach, Mr. Scott Twaroski, Dr. Lou Adler and Mr. David Rish. I also wish to thank the many members of the *Eli Broad Graduate School of Business* who have helped throughout my graduate career, most notably the management department secretarial staff and my fellow doctoral students.

I would also like to acknowledge the important contributions and sacrifices made by my wife Suzanne and children, Jason and Jessica, while I pursued my graduate degrees. I wish to thank my parents for their efforts and example in spurring my interest in advanced education. I am grateful for the support and encouragement offered by all of my family members throughout this process.

Finally, I would like to thank Professor John Hoagland, the members of the *Institute of Water Research* and *The Center for Remote Sensing* as well as the many others at Michigan State University for their support and camaraderie.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Paper manufacturers and other companies in the process industries are currently experiencing production pressures similar to those that batch or repetitive manufacturing managers have dealt with for the last two decades. Customers now expect suppliers to operate in just-in-time (JIT) environments with shorter delivery lead times even as they request higher quality and broader product lines. Better production scheduling can increase responsiveness to these new demands. Scheduling production to minimize waste enables management to streamline operations making them more efficient and profitable.

Efficient production sequencing is often an essential component of production scheduling in the process industries. The cost of configuring a manufacturing station can be independent of the sequence of products produced at the station. However, there are situations where the cost of setting up for manufacturing is sequence dependent. This is common in process industries and in the paper industry in particular, where residual materials from prior manufacturing can affect the current production.

When the produced materials are cut into finished products, the production scheduling problem becomes more complex, a situation common in the paper and process industries. An optimal solution to a stock cutting problem determines the best way to cut lengths or shapes from a stock of raw materials. The raw materials have value, and the objective is to minimize the amount of raw materials used while satisfying demands for each length or shape. A wide range of production technologies such as textiles, paper and steel would benefit from more effective production scheduling techniques that consider a subsequent stock cutting problem. In many situations that require the solution of a stock cutting problem, the composition of the available stock is beyond the control of the decision maker. For example, in the lumber industry the sizes of the trees dictate the composition and dimensions of the raw material, the stock set the cutting priorities are based on. In contrast, manufactured products offer the decision maker more control over the composition of the cutting stock.

Although there are substantial bodies of literature addressing both sequence dependent scheduling and stock cutting problems, to date there has not been a formulation to address these concerns simultaneously. This situation is common in the paper industry where paper machines produce stock of varying widths that will be cut into finished products. The cost of changing the type of paper is dependent on both the current and planned production.

EXAMPLE PROBLEM

The relationship between production scheduling and stock cutting is illustrated by the following example. For the purpose of simplicity, the example assumes all demand is for one roll of each type of paper in each requested width. A scheduler plans to produce two types of paper, A and B, where paper A is required in a roll that is 100" wide and paper B is needed in widths of 50" and 60" rolls. The company has two paper machines, 1 and 2, with the capability of producing widths of 60" and 100" respectively. At the beginning of the planning horizon both are producing paper C. Table 1-1 describes the cost, in dollars, of switching between papers on each machine, where the rows show the paper currently under production, and the columns describe the status of the machine after the switch. The incidence matrix shows the cost of that transition.

Table 1-1: Machine Dependence for Sequence Dependent Production Scheduling

	Machine 1			Machine 2		
Paper	A	B	C	A	B	C
A	0	4	6	0	4	6
B	3	0	9	6	0	12
C	2	9	0	2	11	0

Trim loss can be assumed to be $1 per wasted linear inch. There are several approaches to scheduling these orders.

A first approach might be to have both orders scheduled on machine 2 leaving machine 1 free to process other orders. This would require switching to product A and then to product B. The transition cost would be 2 + 4 = $6. The trim loss for paper A would be zero, and the trim loss for the first reel of paper B would be 50" and the second 40". Making the total cost of producing both orders on machine 2 96 dollars.

Producing paper B on machine 1 and paper A on machine 2 generates a total cost of $21. In this situation the transition costs would be higher, $11, while the cutting expenses would be lower, $10.

This simple example illustrates the benefits of considering the sequence dependent transition costs and trimming costs simultaneously. As in many manufacturing situations, production capability in the paper industry depends on both the type of machinery available and the mix of products demanded. Setup times for products are sequence dependent, indicating opportunities for more efficient operations through better production scheduling methods. Additional complexity arises when the output will be cut into finished products. Most paper mills have several machines with varying capabilities, the most obvious is different widths. Therefore, a production scheduler in a paper mill must consider many factors simultaneously when devising schedules for the production of cutting stock: due dates, sequence dependent setups, inventory levels, proportion of fiber recycled and the manufacturing and cutting capability of each machine.

A rolling planning horizon reflects the dynamic nature of most production planning systems where the product is demanded at varying intervals and plans are frequently revised.

According to Baker and Peterson (1979), a rolling schedule results from solving a multi-period production schedule and then implementing part of it. A rolling planning horizon introduces an additional level of flexibility, allowing the scheduler to build inventory levels to meet future demands or revise a production plan as more information becomes available.

PROBLEM STATEMENT

Baker (1974) contends that production scheduling should take place after three fundamental managerial decisions have been made: the product or service to be provided, the amount of product or service and the quantity of resources to be made available. After implementation of these fundamental managerial decisions, the scheduling problem should be solved by answering two questions: determining the allocation of resources to each task and the time when the task will be performed. This research seeks to answer these two questions for a situation in which manufactured materials are cut into products. The manufacturing environment is a multi-product facility that has several production lines with sequence dependent setups.

The objective of the research is to develop a method for devising schedules that meet customer time constraints and minimize the total cost of four components: production of the raw stock, trim waste from the stock cutting procedure, lost sales and the cost of scrapping excess product. The developed algorithm accomplishes this by first determining an efficient production sequence for each machine, and then setting the number of reels (form of the paper leaving the paper machines) to produce. Inventory levels for each period of the planning horizon as well as the quantity and period that unsatisfied demand will occur are also specified by the algorithm.

Final products, rolls, are cut from the reels of paper according to available cutting patterns. A cutting pattern represents one possible method of dividing a reel into rolls. For this research, only non-dominated patterns are considered.

A cutting pattern is dominant if no other pattern exists where it is possible to cut more rolls in any one width while producing an equal or greater number of rolls in all other widths cut from the reel. For example, if paper is demanded in 50" rolls and 20" rolls and the paper machine has a width of 100" there are several combinations of useful widths that could be cut from the 100" reel. For instance two 50" rolls could be cut, or one 50" roll and two 20" rolls. Both patterns would be dominant because it is not possible to have a pattern that will produce more rolls in one of the widths while producing as many in the other width of interest. A pattern where there was one 50" roll and one 20" roll cut would be dominated by the pattern that produces one 50" roll and two 20" rolls.

DESCRIPTION OF THE MANUFACTURING ENVIRONMENT

Stock cutting together with production scheduling is most commonly encountered in process industries.

Process Industries

Process industries such as paper, food and chemical manufacturing can be defined by two important facets, what they produce and the nature of the material flow in the process (Taylor, 1979). Manufacturing facilities in process industries are characterized by large capital expenditures that require high levels of utilization to achieve economic viability. Furthermore, products must meet a host of quality measures. Equipment operating under steady state conditions, which typically occur well after a start-up or transition period, are most likely to produce a consistent product. Achieving this steady state frequently requires considerable time and expense. Therefore, such facilities usually operate twenty-four hours per day, year round, except for scheduled and unscheduled maintenance.

Process industries share some common manufacturing characteristics. Raw materials represent nearly 80% of the cost of production. Capital investment is high, and profit margins are narrow, making efficient production crucial to achieving profitable operations (Rice & Norback, 1987).

Other features also differentiate process industries from those involved in assembly or fabrication. One is an inverted bill of materials. Unlike repetitive manufacturing, where final products require many components, the process industries produce large numbers of final products from relatively few raw materials. Typically, products are sold at several stages in the production process, manufacturing equipment is highly specialized and delivery lead times are short (Nelson, 1983).

This research deals primarily with the paper industry, which is representative of process industries. The production scheduling procedures developed here should be applicable to an assortment of manufacturing situations in other process industries.

Paper Production

Paper is sold as reels, rolls, or cut sheets. Reels are the direct result of converting pulp to paper. They are usually the width of the paper machine. Their defining characteristics are the type of paper, the basis weight and quality grade. The type is often a brand name and denotes a paper of a specific color, composition and texture. Basis weight is a measure of the mass of the paper, the weight of 500 sheets. Quality grades can be expressed by many measures such as defects per unit of surface area, gloss, and discoloration. Rolls are the result of slitting the reels width-wise immediately after the production of the reel. For the purposes of this research the slitting procedure is assumed to be unconstrained since it is fast compared with other operations, and any number of rolls can be cut from one reel of paper.

Procedures following the slitting operation are called conversion. Examples are placing an embossing mark on the paper, finishing one or both sides of the paper and producing cut sheets. Cut sheets are made by cutting the roll width-wise as it is unrolled. The result is rectangular sheets that are usually trimmed to precise dimensions. All cuts, whether lengthwise or width-wise, are from edge to edge and are called "guillotine" in the stock cutting literature.

Typically, orders are for a specified tonnage of paper of a particular type, cut dimensions, basis weight and grade. For this research, requests for cut sheets will be specified in terms of the number of rolls of paper required to produce the order. The scheduling algorithm will then determine the quantity of reels needed and the most efficient schedule for producing and then cutting them into rolls.

In the paper mill used as a setting for this research, four primary raw materials, Northern and Southern hardwoods and softwoods, are blended to form several types of paper pulp. The facility has three paper machines, and any given order could be produced on one or more of them. The type of paper produced influences the production rate, and a paper machine can produce approximately one reel every forty-five minutes. The output of reels can be characterized by the type of paper and other parameters such as the basis weight and width. In addition, each kind of paper may be one of several quality grades as well as different colors. Reels of paper weighing approximately 7000 pounds are usually produced to the maximum width of the machine. Narrower reels can be produced, but the practice represents lost production capacity and is avoided.

After the rolls are produced, manufacturing ceases to be continuous, and the process is more characteristic of batch production, making it possible to schedule subsequent operations independently. Therefore, this is an appropriate point to segment the process for analysis. This is the method used in all prior work on the paper trim problem known to this researcher.

The effectiveness of a given schedule can be defined by the total cost of producing the demanded material. Although the effect of inadequate management on two of the primary costs, over- and under-production, is obvious, poor management of the transitions between papers or the creation of a large amount of trim waste (the result of material left over after reels are cut into rolls) can have other implications. As a rule, the greater the similarity in the types of paper, the shorter the time required to convert from one paper to another of acceptable quality. Paper waste, whether from low quality or trimmings, called "broke," is recycled. Minimizing the amount of broke has several advantages. Most important, broke represents lost production capacity. Second, paper pulp is nearly 99% water, and the energy required to remove it is substantial. Finally, excessive recycling lowers the quality of the paper as the fibers break down from repeated reprocessing. Although this feature would be difficult to capture in a formulation, it does show that better scheduling procedures may increase product quality in some circumstances.

Paper is manufactured in both make-to-stock, an approach where firms attempt to meet customer orders from inventory, and make-to-order, a strategy where production is scheduled to meet specific customer orders. The most common types of paper, such as the brown wrapper that becomes grocery bags, are commodities and are usually produced using a make-to-stock strategy. On the other hand, specialty papers such as facsimile or computer paper are normally produced in make-to-order planning systems because of the wide range of production specifications.

Although some orders are shipped from inventory, it is more common to have production assigned upon customer request. This is also the case for much of the make-to-stock items where an intermediary will frequently order in large volumes and perform part of the warehousing and distribution function. Market conditions dictate that paper will be shipped on time, or the customer is not obligated to accept the order. Therefore, backordering is unusual for much of the paper industry.

IMPORTANCE OF THE PROBLEM

The variety of manufacturing environments in the paper industry offers many opportunities to demonstrate substantial improvements in production efficiency through this research. The high levels of production in the paper industry translate small (1 or 2 percent) increases in efficiency into substantial cost savings (Haessler, 1988). Paper sales in the United States were estimated at $42 billion in 1980 and represent nearly 4 percent of the gross national product (Noble, 1973).

Other companies in the process industries must schedule cutting production. These businesses account for nearly 50 percent of all production in the United States (Novitsky, 1983), suggesting that this research might have broad applicability.

Representatives of the paper and steel industries have described a distinct shift in management emphasis in recent years. Advances in automation have made achieving the required technical specifications for the product nearly routine. Therefore, the focus of managerial attention has shifted to scheduling and control issues. The paper industry is moving toward a long term goal of a completely automated production facility, called mill-wide control. Higher levels of automation will make it possible to implement schedule changes much more effectively (Routledge, 1988).

Although this research is aimed at designing production planning tools for the plant level, another important application would be as a component of a larger model describing the entire production planning process. This larger model could be used to address business questions at the corporate level. These questions might include when and which machines should be shut down for maintenance or which facility should be assigned a given job. For these purposes the algorithm could be used in a Hax-Meal hierarchical production planning framework (Hax & Meal, 1975).

Most paper is produced by large companies with several manufacturing facilities. Firms currently apply a hierarchical planning framework to allocate jobs first to individual facilities and then to the appropriate machines. A scheduling algorithm describing job allocations at the plant level would play an important part in developing an effective company-wide production planning procedure. The scheduling algorithm also could be used to schedule maintenance or vacations and to establish appropriate policies for responses to changing market conditions.

DESCRIPTION OF SUBSEQUENT CHAPTERS

The following is a brief description of each of the subsequent chapters.

Chapter 2: Literature Review

This chapter provides a review of previous related research. The primary focus of the review is the two bodies of literature that provide the foundation for this research, the stock cutting problem and sequence dependent production scheduling. Literature describing the process industries with particular emphasis on production scheduling and the paper industry is also reviewed.

Chapter 3: Conceptual Framework and Formulation

This chapter describes the manufacturing environment and the assumptions used in the formulation. An explicit formulation for scheduling cutting stock in a multi-period, multi-process facility with sequence dependent setups is presented.

Chapter 4: Heuristic Solution Procedure

Chapter 4 presents a heuristic procedure for selecting a near-optimal production plan. The heuristic is an iterative procedure that sequences all the machines for one period and then specifies the number of reels of paper to produce.

Chapter 5: Model Calibration and Benchmark Procedure

In this chapter the problem generator, used to create test problems, is described, as is the procedure used to determine a bound on the optimal cost of each problem.

Chapter 6: Computational Results

Three groups of problems were solved. The first set of problems was used to develop the heuristic. The second demonstrates the heuristic's performance on an independent set of problems, while the third set of problems shows that the heuristic is practical for life size problems. Results obtained using the heuristic are compared to bounds provided by the benchmarking procedure. In addition, several methods to reduce the problem size are introduced in Chapter 6.

Chapter 7: Conclusions and Future Directions

The final chapter of this document summarizes the results and contributions of this research and outlines several avenues for future research.

CHAPTER 2: LITERATURE REVIEW

Substantial bodies of literature describe production scheduling with sequence dependent setups and stock cutting problems independently. Research addressing situations in which these problems must be considered simultaneously is lacking. Three important characteristics should be noted. First, less complex production scheduling procedures that do not consider the cutting problem have been shown to be NP-Hard (Hsu, 1983). Second, obtaining an optimal solution to a stock cutting problem may require generating an inordinate number of cutting schedules. Finally, due to difficulty associated with solving these kinds of problems, both bodies of literature rely heavily on heuristic solution procedures.

SEQUENCE DEPENDENT PRODUCTION SCHEDULING

Among the first authors to review the production scheduling literature were Day and Hottenstein (1970) who provided a classification of the research describing it in terms of a three part classification scheme: the number of parts composing a job, the production factors possessed by the shop, and the nature of the availability of jobs for processing. The authors go on to describe sequencing methods such as combinatorial approaches, general mathematical programming, heuristic procedures, and Monte Carlo sampling.

In their recent review of the literature for parallel machine scheduling, Cheng and Sin (1990) characterized scheduling problems using four parameters: size of the job

set; number of machines; a system parameter, which gives information about the job characteristics and the performance criteria used to determine the better schedules. The literature base described by Cheng and Sin does not include investigations of parallel machines that are not identical, as is the case for this research. Although paper production facilities can have identical processors, similar machines with overlapping capabilities are more common. For instance, Machine A can produce paper with basis weights varying between 20 and 40 pounds. Machine B can produce paper of between 15 and 30 pounds. In this situation 25 pound paper could be produced on either machine.

The detailed description of the relevant portion of the production scheduling literature base will begin by subdividing the research into two categories, single stage and multi-stage production processes. Because this research deals with single stage production processes we analyze it in greater detail by partitioning the literature still further by defining two subgroups, single and parallel machine scheduling procedures.

Glassey (1968) and Mitsumori (1972) are among several early writers who considered single stage, single product production environments without changeover times. LaRobardier and Filak (1972) then extended this work by including production switch-over costs and inventory holding costs. Driscoll and Emmons (1977) modified the formulation proposed by LaRobardier and Filak for production scheduling on a paper-making machine. In this formulation inventory costs were ignored, additional bounds were added, and a backward-time dynamic programming approach was applied to solve the problem. The solution procedure minimized the total changeover cost while meeting customer demand schedules.

Lockett and Muhlemann (1972) researched the single machine problem in which the setup time was a function of the prior job and introduced several heuristics for solving the problem. Barnes and Vanston (1981) addressed the scheduling of jobs on a continuously available machine in which the individual jobs had linear delay penalties as well as sequence dependencies.

As pointed out by Cheng and Sin (1990), the vast majority of research on parallel machines has assumed that the processors are identical. There are large bodies of research considering both sequence dependent and non-sequence dependent scheduling on identical parallel processors.

Production scheduling on parallel processors with independent tasks was addressed by Rothkopf (1966). The author derived a procedure for determining optimal schedules for a single machine and a dynamic programming approach to be applied to an important class of scheduling problems involving parallel processors. A branch-and-bound procedure was developed by Elmaghraby and Park (1974) to schedule jobs on several identical machines. This model assumed that all jobs are available to be processed at time zero and that there are no a priori precedence relationships between the jobs. Barnes and Brennan (1977) improve the Elmaghraby and Park algorithm by proposing three refinements to the algorithm. These refinements, a bounding procedure and two methods of establishing natural precedence relationships, are combined in a more efficient fathoming procedure.

Dearing and Henderson (1984) used a mixed integer linear program to describe a scheduling problem in the weaving industry and solved it by relaxing the integer requirements. Frendewey and Sumichrast (1988) extended this work with a formulation that considered setup costs, lost production and overtime costs. Through an alternative problem structure, the authors reduced the number of variables that are required to take integer values, making it possible to solve problems of a realistic size.

Oliff and Burch (1985) implement a production planning procedure that considers sequence dependencies in the context of hierarchical production planning. Their approach uses the production switching rule developed by Mellichamp and Love (1978) to determine aggregate production levels. A heuristic is then applied to sequence production within the constraints imposed by the aggregate production plan. In related research, Burch, Oliff and Sumichrast (1987) propose a pair of heuristics in a hierarchical context to derive schedules for the production of fiberglass cloth.

In one of the earliest articles dealing with production sequencing in the process industries Prabhakar (1974) described a manufacturing environment in which chemicals were produced in reactors with subsequent processing. The author presented a mixed integer program designed to determine production quantities for parallel processors. The formulation also considers sequence dependency and limitations on the storage of intermediate products.

Geoffrion and Graves (1976) propose two versions of the quadratic assignment method to solve the problem introduced by Prabhakar. The first approach is the direct application of a quadratic method, while the second involves applying a linear program to augment the procedure. Joint application of the quadratic assignment method in conjunction with linear programming proved to be more effective for solving the general formulation of this type of scheduling problem. The works by Geoffrion, Graves and Prabhakar are complete descriptions of production sequencing in the process industries but do not consider a linkage with a cutting problem.

Smith-Daniels and Smith-Daniels (1986) introduced a lot-sizing and sequencing formulation for packaging lines in the process industries. In this formulation the authors minimize the sum of inventory holding, back-order, setup and tear-down costs while considering sequence dependencies of items within a part family. This capacity-constrained formulation assumes

that one part family is produced on each packaging line in any period. Later, Smith-Daniels (1988) introduced a heuristic procedure to solve the problem.

In an important extension of the Economic Lot Scheduling Problem, Singh and Foster (1987) considered situations with sequence dependent setup costs for a single machine. They proposed a heuristic procedure that breaks the problem into three stages. Although the approach is for a single-machine facility, an application is described where the algorithm is embedded in a multi-machine, multi-product environment.

An alternative method of modeling dynamic production planning problems with sequence dependent setup costs is by redefining the sequencing variables and adding a set of logical variables (Bruvold and Evans, 1985). This reduces the number of 0-1 variables, increasing the likelihood that the problem can be solved with conventional mixed integer program solution procedures. Models incorporating this approach can easily be extended to consider alternative objective functions or losses due to start-up or shut-down.

THE STOCK CUTTING PROBLEM

Several characteristics can be used to describe the stock cutting literature. These are: the dimensions of the problem, the solution procedure (heuristic or optimizing), and whether the cuts are guillotine (edge to edge). Dyckhoff (1990) recently described cutting and packing problems and the connection that exists between them.

Although references to cutting and packing problems date from the late 1930s and early 1940s, the first research to propose an effective algorithm for solving larger problems was introduced by Gilmore and Gomory (1961). This linear programming based approach describes an efficient method of solving the stock cutting problem by limiting the number of variables using auxiliary problems. In a companion article Gilmore and Gomory (1963) consider three practical limitations on the problem, the number of cutting knives, machine balancing concerns and multiple orders for several machines. Gilmore and Gomory (1965) also considered higher dimension problems that can be decomposed into a series of guillotine cuts performed in stages. More recent research on the two-phase approach has incorporated an automatic sequential search procedure to obtain a compromise between computational cost and trim loss (Ferreira, Soeiro, Neves, & Fonseca e Castro, 1990).

The computational effort can be lessened at the expense of larger trim loss by reducing the number of cutting patterns considered. Several authors besides Gilmore and Gomory have introduced procedures to select better cutting patterns. Haessler (1975) developed a format for a one-dimensional trim problem that uses a fixed charge to control the size of the problem by limiting the number of cutting patterns considered for the basis. This mechanism optimizes the tradeoff between minimizing trim waste and computation effort. Pierce (1964)

proposed the selection of dominant schedules to be considered for the solution basis.

Stock cutting questions dealing explicitly with the paper industry include work by Pierce, Johns (1966), and Haessler (1977). Pierce addressed the paper trim problem for both single and multiple machines and proposed heuristic solution procedures. Pierce's formulation minimizes the variable production costs while disregarding the capacity constraints and sequence dependence of paper production. Johns suggested heuristic procedures for solving the paper trim problem while Haessler described the single machine roll trim problem.

Haessler (1988) proposes two heuristics to solve the roll trim problem. The first involves relaxing the integer requirement for the pattern usage. Then additional patterns can be selected for entry into the basis by considering the reduced cost of a nonbasic pattern. This procedure reduces trim loss by increasing the number of cutting patterns considered for the basis. The second approach proposed by Haessler is a sequential method where new cutting patterns are added according to a selection criterion. The two approaches work at cross purposes; the linear programming based approach offers more opportunities to lower trim loss by increasing the number of patterns considered, while the sequential approach, by contrast, minimizes the number of schedules at the expense of increased trim loss. The author uses several examples to illustrate where and how each approach, or a combination of

the two, would be applied in actual practice. Sweeney and Haessler (1990) propose solution methods for one-dimensional cutting problems with rolls of multiple quality grades.

SUMMARY AND CONCLUSIONS

The literature bases are extensive for both the stock cutting and production scheduling problems. However, all prior researchers have viewed the problems separately, while in many manufacturing situations, and the paper industry in particular, they are clearly linked. Therefore, the most important contribution of this research is the linkage of these two problems in the context of paper production.

CHAPTER 3: CONCEPTUAL FRAMEWORK AND FORMULATION

Existing formulations of the paper trim problem describe the number of reels of paper scheduled to be produced on each machine as well as how the reels should be cut into rolls to meet customer demand. This research extends the existing formulations to consider multiple products and the associated sequence dependency of production in a time-phased environment.

ASSUMPTIONS OF THE MODEL

The analytical model presented describes the interaction between the production schedule and the stock cutting decisions with the objective of optimizing the use of production facilities and raw materials. While the current model is designed for the paper industry, the basic concepts can be applied to other industries as well. The decision tools developed in this research will enable managers to specify more precisely when and to which product a machine should switch in order to utilize the production facilities most efficiently while meeting customer demands.

The first important assumption is that production scheduling on the paper machines can be modeled independently from the preceding pulp production and subsequent conversion processes. Based on operations at an active mill, pulp production process can be assumed to have sufficient capacity and flexibility to meet the needs of most production schedules. Subsequent processing of the reels of paper, the

conversion process, is effectively decoupled from the production scheduling on the paper-making machines because its requirements can be specified in terms of rolls of paper. For all practical purposes, the conversion process is treated as another customer, specifying time phased demands that are inputs for the model. This is consistent with work done by earlier researchers.

The formulation assumes that paper is cut immediately after production, and that this process is not bound by capacity. The slitting operation, conversion of reels to rolls, is accomplished by rewinding the paper with knife blades placed at measured intervals in the paper's path. The time required to perform this procedure is negligible, compared with the time required to produce the paper. The model further assumes that paper type is not a function of the machine it is produced on.

There are two other features of the paper industry that play an important part in the formulation of the model. First, paper machines operate continuously and require that the transition to the following paper begin when the production goals for the current paper are completed. Second, backordering is not an acceptable practice in the industry and demands are required to be delivered on their requested due dates.

MODEL DESCRIPTION

The formulation determines the number of reels of paper of each type to be cut according to each pattern, the order in which the reels should be produced on each machine in the facility, the level of inventories throughout the horizon and the composition of lost sales. This is done in a way that minimizes the sum of production cost, the cost of over-production, the cost of lost sales and the transition costs incurred as machines are switched between types of paper.

Constraints on the decision maker are: the specific capabilities of each machine, particularly the width; the time available for production on each machine; and the demand requirements. The formulation must also recognize the need to specify a unique, uninterrupted sequence of transitions between types of papers on each machine over the entire planning horizon.

MODEL FORMULATION

The objective of the formulation is to minimize the total of the production and cutting costs, transition costs, the cost of over-production and the cost of lost sales. This objective is met while subject to two major classes of constraints. The first deals with the usual requirements and limitations of meeting demand with inventory and production subject to capacity limitations. The second class of constraints assure that the sequencing relationships are met.

The following formulation is organized in three parts: definitions, an explicit statement of the formulation, and then a detailed explanation of the objective function and each constraint in the formulation.

Definitions

Indices:

i	labels the state (type of paper currently produced) of the machines, $i = 0,1,\ldots,N$, where $i = 0$ indicates the communicating state that each machine must visit at the beginning of a period and N is the number of products produced;
j	labels the state (type of paper) to which a machine will be switched, ($j = 0,1,\ldots,N$), where $j = 0$ is the communicating state that each machine must visit at the end of each period;
k	indexes the widths that a paper of a given type may be demanded by customers, $k = 1,2,\ldots,K_i$;
m	enumerates the machines, $m = 1,2,\ldots,M$, and M is the number of machines in the facility;
q	indexes the cutting patterns for machine m and product i, $q = 1,2,\ldots,Q_{im}$;
t	enumerates the planning periods, $t = 1,2,\ldots,T$, where T is the number of planning periods;

Parameters:

b_{mt}	the time available on machine m in period t;
c_{im}	the time required to produce a reel of product i on machine m;
C_{imq}	the sum of production cost and trim waste, for a reel of paper of type i produced on machine m and cut according to pattern q;
D_{ikt}	the demand for rolls of paper of width k, type i in period t;
e_{ijm}	the time required to switch from paper i to paper j on machine m;

E_{ijm} the transition cost for changing from paper i to paper j on machine m;

H a large number;

l_{ikt} the cost of lost sales, per roll, for paper of type i, width k in period t;

L_m the width of paper-making machine m;

P_{ikmq} the number of rolls of width k to be cut from each reel using pattern q on machine m when producing product i, subject to the condition

$$\sum_{i=1}^{N}\sum_{k=1}^{K_i}\sum_{q=1}^{Q_{im}} W_{ik}P_{ikmq} \le L_m \qquad m = 1,2,\ldots,M \tag{3-1}$$

r_{ik} the recycling cost for a roll of paper of type i and width k;

S_{m0} the initial slack (unused time) on each machine at the beginning of the planning horizon, assumed to be zero;

V_{i0m0} the initial conditions for each machine, where V_{ijmt} is a binary variable indicating a transition from paper i to paper j;

W_{ik} the k^{th} physical width (in inches) of paper type i;

Variables:

d_{imt} a real number associated with state i on machine m for period t, used to prevent sub-touring;

G the total cost of the production schedule;

I_{ikt} the inventory, in rolls, of width k, product type i at the end of period t;

R_{ikt} the number of rolls of width k of product i produced in period t;

S_{mt} the slack (accumulated unused time) on machine m at the end of period t;

V_{ijmt} a zero one variable showing a switch between paper types i and j ($i \neq j$) on machine m in period t;

X_{imqt} the number of reels of type i produced on machine m and then cut according to pattern q in period t;

Z_{ikt} the demand, in rolls, of width k and type i not satisfied in each period (lost sales).

<u>Formulation</u>

Minimize:

$$G=\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{q=1}^{Q_{im}} C_{imq}X_{imqt} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{m=1}^{M} E_{ijm}V_{ijmt} + \sum_{i=1}^{N}\sum_{k=1}^{K_i} r_{ik}I_{ikT} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{k=1}^{K_i} l_{ikt}\, Z_{ikt} \tag{3-2}$$

Subject to:

$$\sum_{m=1}^{M}\sum_{q=1}^{Q_{im}} P_{ikmq}X_{imqt} - R_{ikt} = 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ k = 1,2,\ldots,K_i \\ t = 1,2,\ldots,T \end{array} \tag{3-3}$$

$$\sum_{i=1}^{N}\sum_{q=1}^{Q_{im}} c_{im}X_{imqt} + \sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} e_{ijm}V_{ijmt} - S_{m(t-1)} + S_{mt} = b_{mt} \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \tag{3-4}$$

$$I_{ik(t-1)} + R_{ikt} - I_{ikt} + Z_{ikt} = D_{ikt} \qquad \begin{array}{l} i = 1,2,\ldots,N \\ k = 1,2,\ldots,K_i \\ t = 1,2,\ldots,T \end{array} \tag{3-5}$$

$$H\sum_{i=0}^{N} V_{ijmt} - \sum_{q=1}^{Q_{jm}} X_{jmqt} \geq 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \tag{3-6}$$

$$\sum_{i=1}^{N} V_{i0mt} = 1 \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}7)$$

$$\sum_{j=1}^{N} V_{0jmt} = 1 \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}8)$$

$$V_{i0m(t-1)} - V_{0imt} = 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}9)$$

$$\sum_{i=0}^{N} V_{ijmt} - \sum_{i=0}^{N} V_{jimt} = 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}10)$$

$$d_{imt} - d_{jmt} + (N+1)\, V_{ijmt} \leq N \qquad \begin{array}{l} i = 0,1,\ldots,N \\ j = 0,1,\ldots,N \\ j \neq i \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}11)$$

$$\begin{array}{l} Z_{ikt},\ X_{imqt},\ S_{mt}, \\ R_{ikt},\ I_{ikt},\ d_{imt}, \quad \geq 0 \end{array} \qquad \begin{array}{l} i = 1,2,\ldots,N \\ k = 1,2,\ldots,K_i \\ m = 1,2,\ldots,M \\ q = 1,2,\ldots,Q_{im} \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}12)$$

$$X_{imqt} \qquad \text{integer } \forall\ i,m,q, \text{ and } t \qquad (3\text{-}13)$$

$$V_{ijmt} \qquad 0 \text{ or } 1\ \forall\ i,j,m, \text{ and } t \qquad (3\text{-}14)$$

EXPLANATION OF THE FORMULATION

The following section is divided into two parts, a description of the objective function and a detailed explanation of the function of each of the constraint sets.

Objective Function

Minimize:

$$G = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{q=1}^{Q_{im}} C_{imq}X_{imqt} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{m=1}^{M} E_{ijm}V_{ijmt} + \sum_{i=1}^{N}\sum_{k=1}^{K_i} r_{ik}I_{ikT} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{k=1}^{K_i} l_{ikt}\, Z_{ikt} \qquad (3\text{-}2)$$

The objective function has four major components: production and cutting charges, setup charges, the cost of lost sales and a recycling charge for any excess rolls produced. The possibility of producing extra rolls results from limiting the available cutting patterns to those that are non-dominated. Although the formulation stipulates that the rolls would accumulate until the final period, in actual practice extra production would be recycled as it is produced.

Constraints

Conversion of reels to rolls in each period:

$$\sum_{m=1}^{M}\sum_{q=1}^{Q_{im}} P_{ikmq}X_{imqt} - R_{ikt} = 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ k = 1,2,\ldots,K_i \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}3)$$

The first set of constraints, equation (3-3), relates the number of reels of paper, X_{imqt}, of type i cut according to

schedule q in period t on machine m to the number of rolls, R_{ikt}, of width k of paper type i produced during period t, where K_i is the number of widths paper i is demanded in. The number of rolls of width k cut for product i on machine m must be selected such that the total of the roll widths does not exceed the width of the paper machine. Each cutting pattern has elements P_{ikmq} (the number of rolls of width k cut from a reel of type i produced on machine m and slit according to pattern q). Cutting patterns are generated such that

$$\sum_{i=1}^{N}\sum_{k=1}^{K_i}\sum_{q=1}^{Q_{im}} W_{ik}P_{ikmq} \leq L_m \qquad m = 1,2,\ldots,M \qquad (3\text{-}1)$$

where W_{ik} is the dimension of the k^{th} width for product i and L_m is the width of paper machine m.

Machine capacities:

$$\sum_{i=1}^{N}\sum_{q=1}^{Q_{im}} c_{im}X_{imqt} + \sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} e_{ijm}V_{ijmt} - S_{m(t-1)} + S_{mt} = b_{mt} \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}4)$$

Equation (3-4) describes the capacity constraints for each machine in each period. Machine capacity not required for the current period can be reassigned through the slack variables, S_{mt}, to the following period. The time required to switch from paper type i to type j on machine m is e_{ijm}, and the time to produce a reel of paper type i on machine m is c_{im}.

The total amount of time available on machine m in period t is b_{mt} (usually 24 hours).

Production, inventory, and demand balance:

$$I_{ik(t-1)} + R_{ikt} - I_{ikt} + Z_{ikt} = D_{ikt} \qquad \begin{array}{l} i = 1,2,\ldots,N \\ k = 1,2,\ldots,K_i \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}5)$$

Equation (3-5) describes the relationship between demand, production and inventory for each of the roll widths for each type of paper.

Setup before paper production can begin:

$$H\sum_{i=0}^{N} V_{ijmt} - \sum_{q=1}^{Q_{jm}} X_{jmqt} \geq 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}6)$$

The i index shows the type of paper for which a given machine is configured. The zero state indicates a null state used to transmit the status of the machines at the end of the prior period to the current period. The formulation relies on the assumption that the ending condition of each machine in period t is identical to the beginning condition of the same machine in the next period. To achieve this there must be a theoretical transition to an artificial ending state (0) for each machine in each period, (V_{i0mt} = 1). In addition, each machine must begin each period by leaving the initial state, (V_{0jmt} = 1). These requirements are enforced by equations (3-7) and (3-9).

Required transition to the ending state:

$$\sum_{i=1}^{N} V_{i0mt} = 1 \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}7)$$

Equation (3-7) ensures that there is a transition to the ending state for each machine in each period. Equation set (3-8), included earlier for clarity, is omitted when solving the model. It ensures a transition out of the initial state and is made redundant by equation (3-9).

Continuity of machine state:

$$V_{i0m(t-1)} - V_{0imt} = 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}9)$$

Equation (3-9) ensures that each machine begins each period producing the same type of paper it was producing at the end of the previous period.

Setup restrictions:

$$\sum_{i=0}^{N} V_{ijmt} - \sum_{i=0}^{N} V_{jimt} = 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}10)$$

Constraint set 3-10 requires that if there is a transition into a state on a machine in a period, then there must be a transition out of that same state.

Sub-tour restrictions:

$$d_{imt} - d_{jmt} + (N+1)\, V_{ijmt} \le N \qquad \begin{array}{l} i = 0,1,\ldots,N \\ j = 0,1,\ldots,N \\ j \ne i \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}11)$$

The Tucker (1960) constraints, equation set (3-11), assure that each state (paper to be produced) for each machine in each period will be assigned a unique real value, d_{imt}. Through this requirement there will be one complete production sequence for each machine in each period. This sequence will begin and end with the null state, and make one visit to each state in which there will be production.

Equations (3-12), (3-13) and (3-14) define the domains of the variables. It is important to note that the inventory and lost sales variables, although defined as continuous, will be integer in any solution through additivity and the integrality requirements of X_{imqt}.

Equation (3-15), not shown in the complete formulation, is an auxiliary set of restrictions that can be used to limit the number of changes on each machine at the paper mill by requiring that a minimum number of reels, g, are produced if there is a transition to a new paper.

$$\sum_{q=1}^{Q_{imt}} X_{imqt} - g\delta_{imt} + gV_{i0mt} \geq 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}15)$$

The binary indicator variables, δ_{imt}, represent the decision to switch to paper type i on machine m in period t. A strict implementation of this policy is difficult because of situations where production overlaps periods. Relaxing the minimum production amount for the last paper produced in the period is a practical way to meet this type of managerial priority with the current formulation.

SUMMARY

Chapter 3 presents the formulation for the stock cutting problem in conjunction with sequence dependent production scheduling within the specific context of a facility producing specialty papers. The objective is to minimize the sum of production, cutting, transition, lost sales and over-production costs, subject to demand, capacity and sequencing constraints. Paper-making machines operate continuously, requiring that the transition to the next paper scheduled begin immediately after production of the current paper ceases. This is addressed through the slack variables and a null production state. The formulation, as presented, is a rigorous description of the production planning problem that confronts managers in the paper industry. However, the large number of integer and binary variables make conventional solution procedures impractical. The following chapter describes an effective heuristic solution procedure.

CHAPTER 4: HEURISTIC SOLUTION PROCEDURE

This description of the heuristic procedure used to obtain solutions for the formulation presented in Chapter 3 is presented in four sections. The first is an introduction describing the operation of the heuristic in broad terms. The next two sections describe in detail the two major steps of the heuristic, determining the production sequence and then specifying the number of reels to produce. The fourth section summarizes the chapter and presents important conclusions.

INTRODUCTION

The heuristic, shown in Figure 4-1, is an iterative procedure that produces a production plan by adding constraints and a limited number of binary variables to the formulation described in Chapter 3. After the additions, the heuristic proceeds by repeatedly solving the enhanced formulation with the binary requirements enforced on selected transition variables and the domain restricted on a few of the integer variables, X_{imqt}, used to determine the production quantities.

Enforcement of the integrality requirements is based on two major objectives. The first is determination of a suitable production sequence, that the different papers will be produced on each machine. Second, after the production sequence is set, production quantities can be specified.

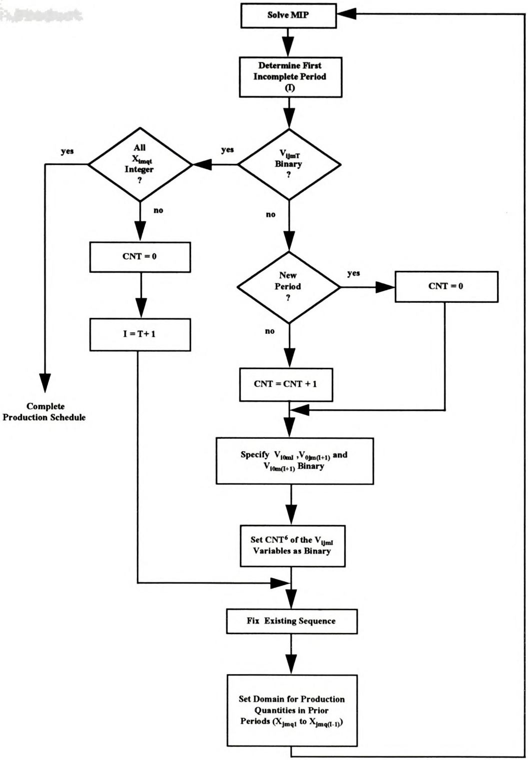

Figure 4-1: Heuristic Procedure

PRODUCTION SEQUENCING

Production sequencing is accomplished by determining the last period with an incomplete production sequence on one of the machines and then selectively enforcing some binary requirements for variables describing transitions on machines in the incomplete period. Once enough of the transition variables, V_{ijmt}, are specified as binary (or are fixed), to determine a production sequence for the period, the heuristic then sets the production quantities. Although the procedure for setting the production quantities is straightforward and will be described later, it is necessary now to describe some additional details associated with determining the production sequence.

Figure 4-1 shows the sequence of steps used to solve problems using the heuristic procedure. Initially the formulation is solved with all of the integrality restrictions relaxed. Subsequently, selected sets of the transition and production variables are forced to take integer values requiring the solution of a Mixed Integer Program (MIP) at the start of each iteration.

The results from the solved MIP are then tested to determine the first period with an incomplete production sequence, period I, a situation where some transition variables in the period are not binary. Completely binary transition variables guarantee, through the Tucker constraints, a unique and uninterrupted production sequence

for all machines in periods prior to period I. Transition variables for those periods are fixed at their current values. Transition variables for period I that are a part of completed sequences are fixed up to but not including the transition to the null state at the end of the period. This leaves the possibility of adding another paper to the sequence during the next iteration.

Three mechanisms are incorporated in the formulation to encourage the complete sequencing of period I in the next iteration. Specifying all variables describing the transition from the endpoints of the completed sequences is the first and most important mechanism. Second, all transitions to the null state at the end of period I are required to be binary.

A final mechanism assures that period I will be correctly sequenced. This is done by rapidly increasing the number of transition variables in period I required to be binary. Each time there is an attempt to sequence the same period the counter, CNT, is incremented. This count is then raised to the sixth power to determine the number of transition variables, in addition to those already described, that will be specified binary.

Transitions out of the null state in period I+1 are also required to be binary. This assures that once period I is sequenced, there will be continuity between the ending state of each machine in period I and the initial states of the machines in period I+1.

Several mechanisms are employed to assist the formulation in completing the production sequence in a given period. The first is the addition of binary variables, δ_{jmt}. Their purpose is to show whether there will be production of paper j on machine m in period t. The delta variables work with an additional constraint set, (4-1), in two ways.

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} V_{ijmt} - \delta_{jmt} = 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (4\text{-}1)$$

First, if there is a transition to paper j these constraints require that the sum of all transition variables describing entrances to state j will equal one. This is an important property because it forces the formulation to recognize, in some sense, one complete setup. Second, the constraints require that transition variables for papers that will not be produced are set to zero.

Solving this formulation with the binary requirements of the V_{ijmt} variables relaxed frequently produces a valid production sequence, however, this is not guaranteed. Incomplete sequences can occur when the transition variables are allowed to take fractional values. This results in solutions that suggest several simultaneous partial transitions to the next paper scheduled on a machine. Figures

4-2 and 4-3 give examples of solutions with and without complete sequences for a single period on one machine.

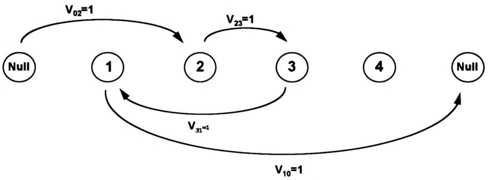

Figure 4-2: Solution With a Valid Sequence

Figure 4-2 is a complete production sequence, with a clear ordering of the papers, beginning in the null state and then entering state 2, proceeding to paper 3, then paper 1 and finally returning to the null state. In the example where the transition variables are allowed to take fractional values,

Figure 4-3, it is not possible to establish the production sequence.

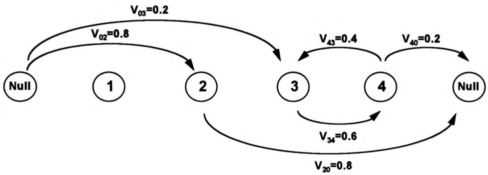

Figure 4-3: Solution With an Invalid Sequence

When transition variables do take fractional values the formulation must be constrained further by stipulating that the variables describing the transitions to the null state are binary. However, there is an additional practical consideration that further limits the number of transition variables that take fractional values.

As first pointed out in Chapter 3, the transition to different types of papers can be costly and may introduce difficulties in achieving consistent quality. For this and other managerial reasons, many production planners would prefer to produce at least a minimum number of reels, g, once

production begins. This operating priority is carried out with a set of constraints of the form:

$$\sum_{q=1}^{Q_{imt}} X_{imqt} - g\delta_{imt} + gV_{i0mt} \geq 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}15)$$

In this research the minimum number of reels of production for each setup was set at four. This addition to the formulation plays an important role in the performance of the heuristic for several reasons. First, the minimum production size limits the number of papers produced on a machine in a period to three or four. Second, higher production volumes in constraint set (3-6)

$$H\sum_{i=0}^{N} V_{ijmt} - \sum_{q=1}^{Q_{jm}} X_{jmqt} \geq 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}6)$$

will encourage each of the transition variables associated with paper types that are produced to take larger values. Once the values of the V_{ijmt} variables are large enough, the Tucker constraints, equation (3-11), become binding.

$$d_{imt} - d_{jmt} + (N+1)\,V_{ijmt} \leq N \qquad \begin{array}{l} i = 0,1,\ldots,N \\ j = 0,1,\ldots,N \\ j \neq i \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \qquad (3\text{-}11)$$

Finally, and most important, constraint set (3-15) allows the enumeration process to rule out sequences with small production quantities. In some cases this can assist the

fathoming of the branch-and-bound tree considerably with a minimal impact on solution quality.

It is useful to note that the minimum required production quantity, constraint set (3-15), provides enough structure to make it practical to require that the variables describing the decision to produce a paper on a machine in a period, δ_{jmt}, are binary for the period with the incomplete sequence. However, this requirement and the added constraints are not sufficient to guarantee that the production sequence will be complete through the current period. For this reason, it is necessary to restrict further the values of the transition variables, V_{ijmt}, in the incomplete period.

The most confining restriction is a requirement that transition variables whose solution values were equal to one in the prior iteration and are part of a complete sequence are set at one for all subsequent iterations, fixing established production sequences. An exception is made for variables describing transitions to the null state. These transitions are kept binary until the sequencing is complete on all machines for the current period, providing the heuristic with additional opportunities to schedule different papers on a machine by assuring an endpoint for each production sequence in the first incomplete period.

The next restriction stipulates that all transitions from the endpoints of the production sequence on each machine are binary, forcing a complete transition to the next paper or the

null state, should the solution procedure chose to do so. Three other sets of transition variables are also restricted to binary status. These are the required transition to the null state in the current period and the transitions from and to the null state in the subsequent period. These are important because they require continuity between periods and they constitute class 1 specially ordered sets (Beale & Tomlin, 1969) whose structure is exploited.

Allowing some transition variables to take real values can generate solutions with sub-tours. Sub-touring, shown in Figure 4-4, occurs when complete transition sequences can be devised that do not pass through the null state.

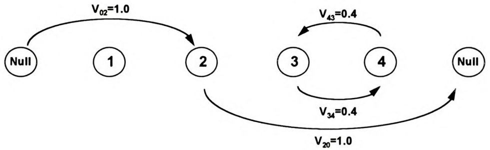

Figure 4-4: Sequence With Sub-Touring

Sub-touring is prevented by increasing the number of transition variables (besides the specially ordered sets and the required transition from the endpoints) required to take binary values on each successive attempt to sequence a period. For instance, the first time there is an attempt to sequence period one, no variables beyond the previously mentioned sets would be set binary. On subsequent attempts to sequence

period one, the number of transition variables specified as binary would be increased rapidly. This approach allows the procedure to establish the production sequence and then assure that no sub-touring has taken place. Once the sequencing, without sub-touring, is complete for the first period, the sequencing procedure can begin for the second and the production quantities can be set for the first.

SPECIFYING PRODUCTION QUANTITIES

As mentioned earlier, determination of the production quantities is straightforward. The approach is to establish an integer set of possible production quantities around the truncated X_{imqt} values found during the sequencing step of the heuristic. For this research the set of integers had a range spanning from two below to three above the truncated X_{imqt} values.

Two sets of variables are given ranges. The first set consists of any production variables for papers scheduled to be produced in the period immediately preceding the period currently being sequenced. The ranges of production quantities provide additional flexibility for the sequencing procedure while assuring that the integrality requirements will be met in the earlier period. Production quantities are fixed at their integer values for periods more than one period before the one undergoing sequencing.

In most situations the procedure just described would be sufficient to guarantee production in integer quantities. However, if more than one period happens to be sequenced in an iteration it would be possible to have fractional production variables in more than just the prior period. Therefore, ranges are also applied to any of the production variables in periods more than one period before the current period that take fractional values.

In situations where the total demand level is low and the bulk of the demand is required late in the horizon, the heuristic is faced with a host of identical solutions (from a cost perspective). Most production schedulers would prefer to keep the machines operating in the short term in hopes that additional work would come available later. This priority is implemented by attaching a small profit (when compared to the transition costs) to the slack variables, where that profit increases in the later planning periods. This incentive motivates the formulation to minimize slack in the earlier periods.

At modest levels of utilization the heuristic, as just described, solves most of the scheduling problems expeditiously. However in situations where there is the possibility of substantial slack early in the horizon or lost sales late in the horizon, an additional concession can be made to limit the number of alternative solutions and increase the performance of the heuristic procedure.

Large numbers of essentially equivalent solutions are generated because each lost sale has the same cost despite type or timing. As a practical matter, there is no indication that the decision to reject a sale occurs at the plant level. In situations where it is likely that a plant will be unable to meet all of its scheduled demand, steps are taken to reduce the demand level or change its timing. For this research, lost sales are an important part of the planning criteria and a mechanism that introduces minor differences in those costs can improve the solution procedure. In this case, an additional charge is attached to the cost of a lost sale, where the cost varies as a function of the paper type and time period. This provides the branch-and-bound procedure an additional mechanism to differentiate nodes for fathoming.

Although these enhancements were developed to increase the solution speed of the heuristic, they also assure that work is completed as soon as possible, making it obvious when the machines will be out of work. There might be situations where it would be useful to reverse the incentive structure, forcing slack time to the beginning of the planning horizon. Solutions generated this way would provide a conservative estimate of the earliest time that capacity can be made available for unscheduled orders.

SUMMARY AND CONCLUSIONS

Chapter 4 has described a practical heuristic solution procedure for obtaining good solutions to the formulation

presented in Chapter 3. The two part procedure seeks to determine a production sequence, the order of production of paper on each machine, by adding new papers to an existing production sequence one period at a time.

After the production sequence is established in a period, the second step of the heuristic specifies the number of reels of paper to produce in the periods up to and including the period sequenced most recently.

Although this procedure will produce a production schedule, it is not an optimizing procedure; consequently, the quality of those schedules cannot be guaranteed. The next chapter describes the procedure used to benchmark the performance of this heuristic procedure.

CHAPTER 5: MODEL CALIBRATION AND BENCHMARK PROCEDURE

As mentioned earlier, the production scheduling of stock that will be cut has applicability in a broad spectrum of manufacturing environments. While limiting the discussion to the paper industry diminishes that spectrum considerably, there are many different market and manufacturing environments in the paper industry alone. Within the paper industry the simplest manufacturing, from a production scheduling perspective, is the production of large volumes of essentially the same product, such as newsprint. Production is standardized, not only in terms of the type of paper produced but also with respect to the widths. This type of application requires machine widths of up to 300 inches, the largest machines in the industry. At the other end of the spectrum are plants that produce a wide assortment of colors and textures as in the production of construction paper. Plant operations in these facilities are more consistent with batch or semi-batch procedures where the machines are stopped between production runs so that they can be thoroughly cleaned before the production of the next paper begins.

A market that is experiencing considerable growth is specialty papers, such as facsimile, most types of writing or computer papers (uncoated freesheet), and the backings for pull away stickers (release paper). Plants that produce for these markets typically have smaller machines producing several different types of paper, and each paper can be

produced in several grades and/or basis weights. The problems used for testing the performance of the heuristic were generated based on manufacturing and marketing situations consistent with this type of production. The information used to specify distribution functions for the parameters of the test problems was obtained through interviews with individuals in the paper industry.

PROBLEM GENERATION

Problem generation begins by entering parameters that characterize the size of the problem. These are the number of periods (T), the number of machines (M), the number of distinct papers (N), and the level of expected demand. Based on this general description of the problem size, a complete manufacturing scenario is generated using probability distributions described below.

Generation of the cutting patterns for each machine depends on the capabilities of the machines, primarily the width, as well as the widths of paper required. For the specialty papers on which this research is based, the machines are toward the narrow end of the spectrum, 100 to 140 inches in width. Further, there is considerable diversity in the paper widths demanded; at a representative mill, the narrowest width demanded is 20 inches while the largest of the common widths is nearly 80 inches. Machine widths and paper widths are uniformly distributed between the largest and smallest width, and rounded to the nearest inch.

Cutting patterns and the associated cutting charges are generated by enumerating all possible patterns. While this is tedious, it is only necessary to do it once. For the purposes of the simulation all rolls in an order are for the same type and width of paper. Each kind of paper could be demanded in a maximum of four different widths over the planning horizon.

To limit the number of possible patterns to those that are most likely to be useful, only dominant patterns (defined in Chapter 1) are considered. Although this assumption will tend to cause over-production, the amount is small when compared to the total production volume, and in actual practice the cost of storing or disposing of excess production is quite small.

Simulated demand, measured in rolls of paper of a given width and type, is generated for each day of the planning horizon. The number of rolls demanded depends on two parameters, the largest number of orders expected in a day, and the maximum number of rolls required per order. All parameters are drawn from the appropriate uniform distributions.

Transition times and costs for each machine vary considerably depending on the degree of difference between the types of paper. Changing between basis weights is relatively easy and requires only a few minutes, while transitions between papers of different colors may require stopping the machine and completely rinsing it clear, a procedure that can

take several hours. This research assumes that several similar kinds of paper are produced, each in an assortment of basis weights. Based on similarities in the paper types, anticipated transition times are assumed to be uniformly distributed between 0.1 and 2.0 hours. The expected time required to produce a reel of paper is drawn from a uniform distribution over 0.5 to 1.0 hours. Similarly, the transition costs are uniformly distributed between $40 and $800. The production cost has three components, energy, fiber and labor. Trim waste and the recycling costs are adjusted to reflect the value of the fiber that can be recovered. Fiber represents one half the total cost, while labor and energy cost each represent one quarter (D. Rish, personal communication, August 10, 1993). Finally, the cost of a lost sale was set at a large arbitrary number.

BENCHMARK PROCEDURE

This research is based on operating conditions at two mills that produce a fairly wide assortment of specialty papers that are sold in bulk form and then used as a raw material in subsequent manufacturing operations. This type of paper mill may have three paper-making machines and will try to establish a production plan for the next two to four weeks. On this horizon, the production planner may have to schedule five distinct kinds of paper with five to seven different basis weights. Each type, paper kind and basis weight combination, can have as many as four different widths. Table

5-1 shows the dimension of a small formulation describing a production scheduling decision at the example mill.

Table 5-1: Representative Problem Size

Problem Parameter	
Planning horizon	14 days
Number of product types	10
Number of machines	3
Maximum number of widths per product type	4
Variable Type	
Cutting patterns, X_{imqt}	3206*
Transition, V_{ijmt}	4620*
Rolls of paper, R_{ikt}	434
Inventory, I_{ikt}	465
Lost sales, Z_{ikt}	434
Slack, S_{mt}	42
Sub-tour restriction, d_{imt}	504
Total number of variables	9705

* Integer or binary

Besides the 9705 variables described in Table 5-1, this formulation would have 6832 constraints. Although, by modern standards, a problem of this dimension might be characterized as a medium-sized linear program, the large number of integer and binary variables precludes any opportunity of obtaining an optimal solution within an acceptable amount of time. Therefore, much of this research effort has focused on finding an effective heuristic solution procedure.

The most appropriate mechanism for conclusively establishing the effectiveness of a heuristic procedure would be to compare the quality of the solution obtained using the

heuristic to the optimal solution. However, the computational cost of obtaining optimal solutions for relatively small problems can be prohibitive. Various methods were considered, most notably Lagrangean Relaxation (Fisher, 1981), to overcome the intractable nature of solving the formulation to optimality. Failure to find a relaxation that produced subproblems with an exploitable structure dictated the use of a more direct means to demonstrate the effectiveness of the heuristic procedure. This procedure entailed using branch-and-bound (Land and Doig, 1960) to solve the optimal formulation whose objective function is given by equation (3-1). The solution obtained from the heuristic was employed to assist the pruning of the enumeration tree. The performance of the branch-and-bound procedure was further enhanced by using specially ordered sets and a specified branching order.

Additional binary variables, δ_{imt}, whose purpose and usage are the same as in equation set (4-1), will not compromise the integrity of the formulation and offer opportunities to improve the fathoming of the enumeration tree. Branching on these variables first usually decreases the time required to find an optimal solution. In a sense, addressing these variables early in the fathoming process is analogous to a manager first deciding whether to produce, and then later deciding how much to produce. As mentioned in the description of the heuristic in Chapter 4, the entrances and exits to and from the null state constitute a specially ordered set whose

exploitation further improves the performance of the branch-and-bound procedure. It is interesting to note that the variables describing the switch to any given paper on each machine in each period are ordered sets also. However, empirical trials showed that only the sets describing the entrances and exits to and from the null state offered any significant change in computation effort.

The objective of the benchmark procedure is to provide a measure of the suboptimality of the solution obtained with the heuristic. The best method is to use the objective value obtained from the heuristic solution as an upper bound during the fathoming of an enumeration tree for the optimal formulation. If the enumeration tree is completely fathomed, the solution obtained using the heuristic is shown to be optimal; if the heuristic is suboptimal, then the actual optimal solution will be uncovered.

The computation effort required to fathom the enumeration tree using the heuristic's objective value made this approach impractical for most of the problems. The number of binary and integer variables, as well as the availability of alternate optima, generates enumeration trees that require considerable amounts of memory and exorbitant amounts of computer time to solve. Problems that have more than one optimal solution could require the algorithm to consider a large proportion of the enumeration tree. Further, if the alternative optima were from disparate parts of the tree, the

algorithm would have to consider more of the tree simultaneously. This may be the case when there is a capacity limitation in the last part of the planning horizon and the procedure is indifferent to which period the sale is lost in.

Given that the purpose of the benchmark procedure is to estimate the error associated with the heuristic solution, it is possible to use a series of proposed incumbents to establish a bound on the optimal solution. For the purposes of this discussion an incumbent is a bound on the actual objective used to dissect the search space and increase the chances that the pruning procedure will fathom the entire tree or encounter a feasible solution.

A binary search procedure was applied in which the first incumbent was a value halfway between the objective obtained from the relaxed formulation and the objective obtained from the heuristic. If the enumeration tree was completely fathomed without encountering a feasible solution, a lower bound was established on the optimal solution and another incumbent was generated halfway between this bound and the upper limit. The initial upper limit was the objective value of the heuristic. Upper limits were updated if one of two circumstances occurred: a better feasible solution was encountered, or an incumbent value was too large to prevent the enumeration tree from outgrowing the computer's capabilities. This binary search procedure continued until an incumbent was proven optimal, or the value of the largest

incumbent capable of fathoming the entire tree was established. From this incumbent, a maximum error was calculated for the objective found using the heuristic.

CHAPTER 6: COMPUTATIONAL RESULTS

The objectives of the research were to develop a solution procedure for problems of a practical size and demonstrate the quality of those solutions. For this purpose three classes of problems, Validation, Independent and Scale, were developed and solved.

INTRODUCTION

Each of the problem classes had a distinct function in proving the effectiveness of the heuristic procedure. The primary use of the Validation problems was to develop and validate the heuristic, while the Independent problems provided an unbiased evaluation of the heuristic's performance. Scale problems were designed to give some insight into how the computation cost would grow as the length of the planning horizon increases, and to show that the heuristic could be used to solve problems of a practical size.

The Validation and Independent problems each have three different sizes, (T), (M) and (P), with (T) the smallest. For each size there are three different utilization levels, low, medium and high. Each Scale problem has three machines, four products and is specified at the low utilization level making it possible to consider a broader range of values when evaluating the effect of increasing the length of the planning horizon on computation costs.

The next three sections provide descriptions of the Validation, Independent and Scale problems along with their

accompanying results. Three tables illustrate these descriptions. The fifth section of the chapter describes several procedures that could reduce the problem size and/or the computation effort. The chapter concludes with a summary of the important results.

VALIDATION PROBLEMS

The first class of problems, shown in Tables 6-V-1 through 6-V-3, was used to develop and validate the heuristic solution procedure. Parameters for this set of problems were selected to ease the coding and validation procedure and provide insights into how the heuristic would perform in diverse situations, such as large machine widths with small paper widths, which induces a large number of possible cutting patterns and substantial diversity in the manufacturing environment.

Description of the Tables

Three sets of three tables, nine in all, describe the problem sets. The first table in each set provides information about the size of the problems. The first five columns contain the name of the problem, the number of periods (T), the number of distinct types of paper (N), and the number of machines (M). Demand, the sixth column, is specified at three levels.

The best description of the problem size is found in the last two columns of the first table of each set, the number of

variables and columns necessary to describe the optimal formulation presented in Chapter 3.

Demand levels for the Validation problems are somewhat arbitrary, with the average demand per day for the medium case twice as large as for the low, and for the high demand case three times as large as for the low. Demand patterns for the validation problems were generated with limited regard for the machine widths. Although it is assured that there is a machine with sufficient width to produce the paper, it is possible to have lost sales in a wide paper while a narrow machine is idle.

All paper widths generated for the Independent and Scale problems are less than the widths of the narrowest machines. This requirement is consistent with the situation at the example mill and produces much more realistic production schedules, where, in actual practice, large mismatches between the market demands and the production environment would be addressed higher in the production planning hierarchy, before the paper mill is committed to delivering the order.

The Independent problems were generated for three different demand levels, where the low level of demand is designed to operate the machines at 50 percent of capacity, the medium demand level to operate at 70 percent of capacity, and the high level case to operate at 90 percent of capacity. Estimated utilization was the ratio of the expected number of reels demanded to the expected shop capacity (in reels per

day). Calculation of the expected shop capacity is straightforward, and is the product of the expected machine capacity (reels per hour), the length of the production day (twenty-four hours), the number of machines and the number of days in the planning horizon. Expected demand is the product of the expected number of orders per day, the expected number of rolls per order, the number of days in the planning horizon and the inverse of the estimated number of rolls per reel of paper. The rolls per reel factor is estimated as the ratio of the expected machine width (averaged across machines) to the average expected paper width (averaged across paper types).

Determining meaningful bounds for solutions obtained using a heuristic is always difficult in this type of research. The second table for each data set provides two comparison measures for the objective obtained using the heuristic. The first is a "Best Incumbent" category where the best feasible solution that is obtained either with the heuristic or optimizing procedure is recorded. The second category used to interpret the quality of the heuristic solution is the "Best Bound" column. This value represents the best bound, without regard to feasibility, achieved through the binary search procedure described in Chapter 5.

There are some important observations to be made about the use of these tables. First, for the purposes of this discussion, "incumbent" will refer to any feasible solution as

good as or better than the solution obtained using the heuristic procedure. If the solution obtained using the heuristic procedure is proven to be optimal, the values in all three columns of the second tables will be identical, and the value in the "Maximum Error" column will be zero. In situations where a feasible solution is better than the solution obtained using the heuristic and proven to be optimal, then values in the "Best Incumbent" and "Best Bound" columns will be identical and the maximum error will be calculated based on the value of the incumbent value. In situations where it is not possible to prove that either the incumbent or the heuristic solution value is optimal, the maximum error is calculated using the best bound.

Tables 6-V-3, 6-I-3 and 6-S-3 contain the objective values of each problem solved with the integrality restrictions relaxed, and the solution time required to solve the heuristic using an *Intel* 486/66 microprocessor. Solutions for the mixed integer programs in both the optimizing and heuristic procedures were solved using version 2.1 of the *CPLEX Mixed Integer Optimizer* (CPLEX Optimization Inc., 1993). The objective value quoted in the "Linear Relaxation Objective" is obtained by solving the formulation with all of the integrality requirements relaxed in the optimal formulation.

Table 6-V-1: Problem Characteristics Report
Validation Problems

Problem Name	T	N	M	Demand	Variables	Constraints
T-A-1	2	2	2	low	119	76
T-A-2	2	2	2	medium	104	72
T-A-3	2	2	2	high	83	68
M-A-2	3	3	3	low	345	261
M-A-2	3	3	3	medium	362	273
M-A-3	3	3	3	high	286	249
P-A-1	5	4	3	low	705	590
P-A-2	5	4	3	medium	655	590
P-A-3	5	4	3	high	782	610

Table 6-V-2: Performance Summary
Validation Problems

Problem Name	Best Incumbent	Best Bound	Heuristic Objective	Maximum Error(%)
T-A-1	40,826.1	40,826.1	40,826.1	0.00
T-A-2	33,899.1	33,899.8	34,120.9	0.65
T-A-3	40,647.2	40,647.2	40,674.2	0.07
M-A-1	none	171,245.2	171,597.6	0.21
M-A-2	953,024.9	950,080.1	953,401.8	0.35
M-A-3	727,453.2	727,453.2	728,265.3	0.11
P-A-1	501,052.4	501,052.4	501,052.4	0.00
P-A-2	531,733.0	531,733.0	532,423.2	0.13
P-A-3	none	1,637,565.0	1,645,936.0	0.51

Tables 6-V-2 and 6-V-3 show that both the heuristic and the optimizing procedure performed well on this type of problem. In addition, the solution speed for the heuristic is very good. Although these problems were not intended to give insights into the relationship between the problem size and the computation effort, the information in Tables 6-V-1 and 6-V-3 make it appear that computation effort is increasing in

nearly the same proportion as the problem size. It should be pointed out that the heuristic reads and writes several files, and a substantial portion of the time used to solve the problem is spent in these activities. The question of the relationship between computation effort and problem size will be addressed in the section dealing with the Scale problems.

Table 6-V-3: Problem Solution Report
Validation Problems

Problem Name	Linear Relaxation Objective	Best Bound Using CPLEX	Heuristic Solution Objective	Time
T-A-1	40,826.1	40,826.1	40,826.1	1
T-A-2	34,120.9	33,899.8	34,120.9	1
T-A-3	40,674.2	40,647.2	40,674.2	< 1
M-A-1	171,597.6	171,245.2	171,597.6	2
M-A-2	953,401.8	950,080.1	953,401.8	3
M-A-3	728,265.3	727,453.2	728,265.3	3
P-A-1	501,052.4	501,052.4	501,052.4	6
P-A-2	532,423.2	531,733.0	532,423.2	5
P-A-3	1,645,936.0	1,637,565.1	1,645,936.0	21

It should be noted that problems where the benchmark procedure demonstrated that it was not possible to obtain a schedule without lost sales were discarded. This procedure is consistent with current practices. The authority to reject a sale is not held at the plant level and when a situation does arise where it is obvious that the plant will not meet the demand requirements, arrangements are made to lessen the demand or reschedule it. The interesting question of determining which order to reject or reschedule is left for further research.

INDEPENDENT PROBLEMS

The second set of problems, whose characteristics are shown in Tables 6-I-1 through 6-I-3, demonstrate the performance of the heuristic with an independent set of problems. The number of machines in this class was kept at three, a number quite common in practice. The parameters for these problems were randomly generated using distribution functions based on the operating conditions at an actual mill. Detailed descriptions of the environment and the derivation of the parameters used in the generation of the problems are given in Chapter 5.

Table 6-I-1: Problem Characteristics Report
Independent Problems

Problem Name	T	N	M	Demand	Variables	Constraints
T-B-1	2	2	3	low	151	104
T-B-2	2	2	3	medium	154	108
T-B-3	2	2	3	high	168	108
M-B-1	3	3	3	low	306	243
M-B-2	3	3	3	medium	351	243
M-B-3	3	3	3	high	361	267
P-B-1	5	4	3	low	751	600
P-B-2	5	4	3	medium	908	620
P-B-3	5	4	3	high	892	610

Table 6-I-2: **Performance Summary Independent Problems**

Problem Name	Best Incumbent	Best Bound	Heuristic Objective	Maximum Error(%)
T-B-1	137,418.3	137,418.3	138,433.3	0.74
T-B-2	120,910.6	120,910.6	120,910.6	0.00
T-B-3	160,068.0	160,068.0	160,676.0	0.38
M-B-1	108,651.7	108,651.7	108,990.6	0.31
M-B-2	none	271,194.0	272,819.9	0.60
M-B-3	none	301,023.4	303,095.2	0.69
P-B-1	none	222,791.0	223,400.4	0.27
P-B-2	none	988,326.2	996,658.8	0.84
P-B-3	none	920,083.3	923,957.4	0.42

Table 6-I-3: Problem Solution Report Independent Problems

Problem Name	Linear Relaxation Objective	Best Bound Using CPLEX	Heuristic Solution Objective	Time*
T-B-1	136,836.2	137,418.3	138,433.3	1
T-B-2	120,532.6	120,910.6	120,910.6	< 1
T-B-3	159,825.4	160,068.0	160,676.0	1
M-B-1	107,760.2	108,651.7	108,990.6	2
M-B-2	270,687.3	271,194.0	272,819.9	3
M-B-3	300,174.6	301,023.4	303,095.2	3
P-B-1	221,960.3	222,791.0	223,400.4	3
P-B-2	987,529.3	988,321.2	996,658.8	35
P-B-3	919,253.8	920,083.3	923,957.4	13

* Time is in minutes on a 486/66.

The tables clearly show that the heuristic produces very good solutions on these problems. Further, these results are accomplished with minimal computation effort.

As mentioned earlier, when the production planner is facing the possibility of missing a sale, steps are taken to reschedule the demand requirements. This observation makes it

unlikely that the demand properties for the early part of the horizon are the same as for later in the horizon. These two features are addressed by lowering the maximum order size by two-thirds in the first period and one-third in the second.

SCALE PROBLEMS

To give some insight into how the heuristic procedure would perform as the problem size is increased, a third set of problems was developed. The Scale problems only consider the effect of an increasing planning horizon. For these problems the number of products is fixed at four and the number of machines is fixed at three. The following three tables summarize the solution results.

Table 6-S-1: Problem Characteristics Report
Scale Problems

Problem Name	T	N	M	Variables	Constraints
S-1	2	4	3	289	232
S-2	3	4	3	600	360
S-3	4	4	3	805	512
S-4	5	4	3	797	610
S-5	6	4	3	951	720
S-6	7	4	3	950	812
S-7	8	4	3	1311	992
S-8	9	4	3	1444	1098
S-9	10	4	3	1673	1240
S-10	11	4	3	1713	1276
S-11	12	4	3	2208	1512
S-12	13	4	3	1565	1482
S-13	14	4	3	2134	1652

Table 6-S-2: **Performance Summary Scale Problems**

Problem Name	Best Incumbent	Best Bound	Heuristic Objective	Maximum Error(%)
S-1	56,842.1	56,842.1	56,842.1	0.00
S-2	32,543.9	32,543.9	32,543.9	0.00
S-3	467,143.7	464,681.2	467,352.6	0.58
S-4	335,314.2	333,670.3	335,702.9	0.61
S-5	625,037.9	622,477.4	625,482.6	0.48
S-6	659,336.1	656,836.8	660,249.9	0.52
S-7	445,635.1	444,070.5	445,697.2	0.37
S-8	818,749.9	813,457.8	818,784.1	0.65
S-9	none	800,714.2	803,497.9	0.35
S-10	1,060,281.9	1,056,315.8	1,061,013.6	0.44
S-11	1,765,124.9	1,753,409.5	1,765,925.6	0.71
S-12	none	1,620,498.7	1,622,933.1	0.15
S-13	none	1,340,279.9	1,346,557.5	0.47

The results shown in Tables 6-S-1 and 6-S-3 give insights into how the computation burden increases as the planning horizon is lengthened. Although computing costs are rising much faster than the problem size it is clear that problems of a practical size can be solved using a modest platform. Table 6-S-2 presents the most encouraging results, showing that the solution quality remains very high even for the longest planning horizons.

Table 6-S-3: Problem Solution Report
Scale Problems

Problem Name	Linear Relaxation Objective	Best Bound Using CPLEX	Heuristic Solution Objective	Time*
S-1	56,062.7	56,842.1	56,842.1	2
S-2	32,156.7	32,543.9	32,543.9	2
S-3	463,575.8	464,681.2	467,352.6	5
S-4	332,567.0	333,670.3	335,702.9	9
S-5	621,054.4	622,477.4	625,482.6	7
S-6	655,073.4	656,836.8	660,249.9	22
S-7	442,758.3	444,070.5	445,697.2	7
S-8	812,033.2	813,457.8	818,784.1	30
S-9	800,040.1	800,714.2	803,497.9	25
S-10	1,055,102.4	1,056,315.8	1,061,013.6	17
S-11	1,752,656.6	1,753,409.5	1,765,925.6	97
S-12	1,619,417.4	1,620,498.7	1,622,933.1	15
S-13	1,339,188.4	1,340,279.9	1,346,557.5	66

* Time is in minutes on a 486/66.

Although it is appropriate to define the problem size in terms of the parameters presented in Table 6-V-1 and 6-I-1 (number of machines, number of days in the planning horizon and the number of paper types), it should be pointed out that problem size is also a function of the widths and number of different widths of the papers demanded and the widths of the paper-making machines. Wide machines producing a broad assortment of narrow papers will be prone to generating a large number of cutting patterns, increasing the problem size. The heuristic procedure significantly reduces the number of binary and integer variables in any one formulation. This is done by selectively relaxing integrality requirements and/or restricting the domain of the integer and binary variables. However, within any one of the subproblems in the heuristic

procedure, there are still enough binary and integer variables that solution times would be unacceptable even on a very powerful platform. The results of this research would have wider applicability if the size of the formulations could be reduced even slightly.

PROBLEM SIZE REDUCTION TECHNIQUES

When considering the possibility of solving problems of a practical size, there are at least two methods that could be used to condense the problems. Both approaches are based on aggregation methods.

Although single days are the preferred planning period, it has been suggested (S. Twaroski, personal communication, October 15, 1991) that there are situations where the planning increment could be extended to forty-eight hours. It is clear that in situations where this approach could be applied, the problem size could be reduced by 50%. A less dramatic but still useful approach would be to aggregate the weekend into a single planning period. This is appropriate in situations where the subsequent conversion processes are shut down over the weekend.

The second aggregation approach is based on the similar nature of many of the products produced. Although demand will be for a specific kind of paper in a specific basis weight, the most formidable transitions are those between kinds of paper. Once a machine is producing, it is a much simpler problem to change basis weights than it is to switch to an

entirely different kind of paper. At the example mill, the practice is to have a fairly definite schedule with respect to the kind of paper produced, while the timing of the production of the different basis weights is left open.

The production planner at the example mill prefers to schedule on a 21-day planning horizon, giving vendors adequate notice for required materials. Although the planner will consider twenty or more paper kind/basis weight combinations, there will only be approximately five distinct paper types. At first inspection, it might seem reasonable to aggregate demand for the basis weights. Then the production scheduling issue might be analogous to many hierarchical production planning situations where part-families (paper kinds) are scheduled first and then the individual part (basis weight) is scheduled within the time allocated for the particular part family. The issue of the cutting patterns confounds this approach when considering production scheduling in a paper mill. This is because the demands are for a specific weight and width for each kind of paper, and aggregating the basis weights would introduce ambiguity, at best, into the meaning of the cutting patterns. The following proposed modifications would make it possible to reformulate the problem with significantly fewer transition variables, the most important consideration for the most time-consuming step of the heuristic.

Grouping products into part families is another method that could be used to reduce the problem size. Paper kind, characterized by a name brand, *Kashmir Natural* for instance, will denote the name of the part family. The different basis weights represent the individual members of the part family. The production scheduling problem can now be reformulated, where the sequencing variables, V_{ijmt}, represent a switch between different part families rather than individual paper types. This requires introducing an index, f, to identify the individual members of the part families.

The production variables still describe the number of reels of a type of paper to produce on a machine m in a period t and then cut according to pattern q, but it must be recognized that the i index now represents the paper family. Production variables are now expressed with the form: X_{fimqt}.

Except for the i and j indexes and the N parameter, most of the variables and parameters used in the following formulation have the same meaning as when they were first presented in Chapter 3. Only new or different indices, parameters or variables will be described in the following definitions.

<u>Definitions</u>

Indices:

f enumerates the individual paper types in each part family, $f = 1,2,...,F_i$, where F_i is the number of paper types in part family i;

i labels the state (paper family) on the machines, $i = 0,1,\ldots,N$, where $i = 0$ indicates the communicating state that each machine must visit at the beginning of a period and N is the number of paper families produced;

j labels the state (paper family) to which a machine will be switched, ($j = 0,1,\ldots,N$), where $j = 0$ is the communicating state that each machine must visit at the end of each period;

k denotes the width of paper of a given type demanded by customers, $k = 1,2,\ldots,K_{fi}$;

q enumerates the cutting patterns for machine m and paper type f from family i, $q = 1,2,\ldots,Q_{fim}$;

Parameters:

c_{fim} the time required to produce a reel of product f from family i on machine m;

C_{fimq} the sum of production cost and trim waste, for a reel of paper of type f from family i produced on machine m and cut according to pattern q;

D_{fikt} the demand for paper of width k, type f, family i in period t;

e_{ijm} the time required to switch from paper family i to paper family j on machine m;

E_{ijm} the transition cost for changing from paper family i to paper family j on machine m;

l_{fikt} the cost of lost sales, per roll, for paper of type f in family i, width k in period t;

N the number of different paper families;

P_{fikmq} the number of rolls of width k to be cut from each reel using pattern q on machine m when producing product f from family i, subject to the condition

$$\sum_{i=1}^{N}\sum_{f=1}^{F_i}\sum_{k=1}^{K_{fi}}\sum_{q=1}^{Q_{fim}} w_{fik}P_{fikmq} \le L_m \qquad m = 1,2,\ldots,M \qquad (3\text{-}1)$$

r_{fik} the recycling cost for a roll of paper of type f, family i in width k;

V_{i0m0} the initial conditions for each of the machines, where V_{ijmt} is a binary variable indicating a transition from part family i to part family j;

W_{fik} the k^{th} physical width (in inches) of paper type f from family i;

Variables:

d_{imt} a real number associated with state i on machine m for period t, used to prevent sub-touring;

I_{fikt} the inventory of roll width k, product type f, family i at the end of period t;

R_{fikt} the number of rolls of width k of product f from family i produced in period t;

V_{ijmt} a zero one variable indicating a switch between states i and j ($i \neq j$) on machine m in period t;

X_{fimqt} the number of reels of type f from family i produced on machine m and then cut according to pattern q in period t;

Z_{fikt} the demand, in rolls, of width k and type f, family i not satisfied in each period.

Formulation

Minimize:

$$G=\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{f=1}^{F_i}\sum_{m=1}^{M}\sum_{q=1}^{Q_{fim}} C_{fimq}X_{fimqt} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{m=1}^{M} E_{ijm}V_{ijmt} + \sum_{i=1}^{N}\sum_{f=1}^{F_i}\sum_{k=1}^{K_{fi}} r_{fik}I_{fikT} + \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{f=1}^{F_i}\sum_{k=1}^{K_{fi}} l_{fikt}\, Z_{fikt} \tag{3-2}$$

Subject to:

$$\sum_{m=1}^{M}\sum_{q=1}^{Q_{fim}} P_{fikmq}X_{fimqt} - R_{fikt} = 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ f = 1,2,\ldots,F_i \\ k = 1,2,\ldots,K_{fi} \\ t = 1,2,\ldots,T \end{array} \tag{3-3}$$

$$\sum_{i=1}^{N}\sum_{f=1}^{F_i}\sum_{q=1}^{Q_{fim}} c_{fim}X_{fimqt} + \sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} e_{ijm}V_{ijmt} - S_{m(t-1)} + S_{mt} = b_{mt} \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \tag{3-4}$$

$$I_{fik(t-1)} + R_{fikt} - I_{fikt} + Z_{fikt} = D_{fikt} \qquad \begin{array}{l} i = 1,2,\ldots,N \\ f = 1,2,\ldots,F_i \\ k = 1,2,\ldots,K_{fi} \\ t = 1,2,\ldots,T \end{array} \tag{3-5}$$

$$H\sum_{i=0}^{N} V_{ijmt} - \sum_{f=1}^{F_j}\sum_{q=1}^{Q_{fjm}} X_{fjmqt} \geq 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \tag{3-6}$$

$$\sum_{i=1}^{N} V_{i0mt} = 1 \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \tag{3-7}$$

$$\sum_{j=1}^{N} V_{0jmt} = 1 \qquad \begin{array}{l} m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \quad (3\text{-}8)$$

$$V_{i0m(t-1)} - V_{0imt} = 0 \qquad \begin{array}{l} i = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \quad (3\text{-}9)$$

$$\sum_{i=0}^{N} V_{ijmt} - \sum_{i=0}^{N} V_{jimt} = 0 \qquad \begin{array}{l} j = 1,2,\ldots,N \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \quad (3\text{-}10)$$

$$d_{imt} - d_{jmt} + (N+1)\, V_{ijmt} \leq N \qquad \begin{array}{l} i = 0,1,\ldots,N \\ j = 0,1,\ldots,N \\ j \neq i \\ m = 1,2,\ldots,M \\ t = 1,2,\ldots,T \end{array} \quad (3\text{-}11)$$

$$\begin{array}{l} Z_{fikt},\ X_{fimqt},\ S_{mt}, \\ R_{fikt},\ I_{fikt},\ d_{jmt} \quad \geq 0 \end{array} \qquad \begin{array}{l} i = 1,2,\ldots,N \\ f = 1,2,\ldots,F_i \\ k = 1,2,\ldots,K_{fi} \\ m = 1,2,\ldots,M \\ q = 1,2,\ldots,Q_{fim} \end{array} \quad (3\text{-}12)$$

$$X_{fimqt} \qquad \textit{integer}\ \forall\ f,i,m,q,\ \textit{and}\ t \quad (3\text{-}13)$$

$$V_{ijmt} \qquad 0\ \textit{or}\ 1\ \forall\ i,j,m,\ \textit{and}\ t \quad (3\text{-}14)$$

The problem reduction technique just described has the potential to offer substantial improvements in the usability of the procedure by reducing the number of paper types (N) for which the transition variables must be generated. In one example mill this approach would reduce the number of types by nearly 80%.

Although the formulation does ignore transition costs between basis weights, these times are small compared with the effort required to switch between paper kinds. In facilities

where it is common to maintain some flexibility in the production schedule, this time is small enough that it would not compromise the scheduling process. This approach addresses the most time consuming and important aspect of the heuristic procedure, generation of the production sequence.

SUMMARY AND CONCLUSIONS

For problems where there is sufficient capacity to meet all of the demand requirements, it has been shown that the heuristic procedure can obtain good solutions with an acceptable computation effort. This chapter also suggests a problem size reduction technique that will increase the applicability of the heuristic. This technique relies on the part family concept, where the various basis weights represent the members of the paper brand family.

CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS

The heuristic developed here, together with the problem reduction techniques discussed earlier, is an efficient method of scheduling in environments where manufacturing is sequence dependent and produced materials are cut into finished products. Many businesses operate in this manner, typically with a large investment in both capital equipment and raw materials. This research provides production managers with a tool that can significantly improve profitability. While this is a significant contribution itself, it also suggests other areas for future research.

There are modifications that could improve the effectiveness of the current heuristic, either by decreasing the time it takes to obtain a good solution or, if considered useful, by forcing the heuristic to develop schedules more consistent with those obtained using an optimizing procedure. The most direct approach to lowering the over-production charge would be the selective consideration of dominated cutting patterns. Selective inclusion of more patterns in this application may not have a noticeable impact on solution time given the small amount of time the heuristic now directs to this effort and could reduce the amount of material recycled.

The most pressing speed issue is the large increase in computing cost as the heuristic is forced to reject some sales. As mentioned in Chapter 4, this problem is

particularly acute if demand is heavy late in the planning period.

A related, and very interesting, question is how to respond to situations where demand will not be satisfied. In an actual production environment, the scheduler would have a variety of options available to respond to schedule infeasibility. These might include negotiating with the customer to reschedule all or part of the order, shipping by an alternate carrier, diverting some work to another plant or, if it is an in-house delivery, allowing the order to be late under the assumption that the time could be made up later. It should be pointed out that many feasible schedules could be improved if the right constraint could be relaxed. In these cases, frequently, the biggest difficulty is deciding which constraint to relax.

Among the most obvious methods of relieving production pressure on one mill is the diversion of some work to another facility. This could be addressed by incorporating the current formulation into a comprehensive company-wide model that can consider several plants in dispersed geographic areas. This approach is appealing because it offers an opportunity to determine production schedules for each machine in the organization that are optimal not just with respect to each plant, but also with respect to the entire organization. It will also allow the company to investigate other issues that affect the production scheduling at the plant level, such

as the selection and timing of a machine shutdown for maintenance, engineering changes, or the management of excess production capacity. Finally, a company-wide model could be used to investigate plant expansion or modernization efforts.

Another interesting research question is the relationship between product quality and the production schedule. To a large extent this is implied in the changeover costs and times already considered in the model. However, this could be extended by incorporating the proportion of virgin materials as an additional set of constraints. Different papers require different ratios of virgin raw materials to recycled fiber. This would give the production scheduler two mechanisms to control the recycled material, the cutting patterns selected and product composition. Production could be scheduled in a way that best utilizes these components. This implementation could be used with various inventory policies to provide management with another tool for managing product quality.

In summary, the research meets its objective of developing an effective method of obtaining good production schedules for environments that face a stock cutting problem in conjunction with sequence dependent setups, and does so with a reasonable amount of computation effort. The results of this research have applicability in the paper industry as well as other, primarily process, industries at the plant level. Given the large production volumes associated with the process industries, the results of this research can achieve

substantial dollar savings through modest increases in production efficiency.

APPENDICES

APPENDIX A: EXAMPLE PROBLEM AND SOLUTION

PROBLEM (Prototype 3)

Random Number Seed: 32

The Number of Periods is: 5
The Number of Machines is: 3
Number of Types of Paper: 4

Maximum Machine Width: 300
Minimum Machine Width: 75

Maximum Paper Width: 300
Minimum Paper Width: 20

Expected Level of Utilization: high
Maximum Number of Orders/Day: 4
Minimum Number of Rolls/Order: 13
Maximum Number of Rolls/Order: 88

Variable Legend

Type of Variable	Number	Beg.	End
Cutting Patterns:	205	1	205
Roll variables:	50	206	255
Inventory Variables:	60	256	315
Rejected Rolls variables:	50	316	365
Machine Initialization Variables:	12	366	377
Transition Variables:	300	378	677
Slack Variables:	15	678	692
Tucker Variables:	90	693	782
Setups on Each Day:	60	783	842
Total Number of Variables			842

Constraint Legend

Type of Constraint	Number	Beg.	End
Reel to Roll Conversion	50	1	50
Capacity Constraints	15	51	65
Demand Constraints	50	66	115
Required Setups	60	116	175
Exit From the Null State	15	176	190
Matched Exit to Entrance	60	191	250
Exit from Each State Visited	60	251	310
Tucker Constraints	300	311	610
Required Trans. Before Prod.	60	611	670
Minimum Production	60	671	730
Total Number of Constraints			730

PROBLEM SPECIFICATIONS

Machine Number	Machine Width
1	136
2	214
3	207

Time Required to Produce a Reel of Paper (in hours)

Machine Number 1

paper(x)	Production Time
1	0.8
2	0.9
3	0.9
4	0.7

Machine Number 2

paper(x)	Production Time
1	0.9
2	0.6
3	0.7
4	0.6

Machine Number 3

paper(x)	Production Time
1	0.5
2	1.0
3	1.0
4	1.0

Cost of Transferring Between Papers (in $s)

Machine Number 1

paper(x)	paper(y)	Transition Cost
1	2	250.9
1	3	48.9
1	4	401.4
2	1	63.2
2	3	112.1
2	4	150.5
3	1	265.3
3	2	202.0
3	4	352.5
4	1	370.1
4	2	391.3
4	3	419.0

Machine Number 2

paper(x)	**paper(y)**	**Transition Cost**
1	2	216.7
1	3	362.8
1	4	550.0
2	1	94.3
2	3	181.8
2	4	369.0
3	1	356.7
3	2	262.4
3	4	187.2
4	1	279.3
4	2	185.0
4	3	366.8

Machine Number 3

paper(x)	paper(y)	Transition Cost
1	2	738.5
1	3	262.6
1	4	554.5
2	1	83.9
2	3	117.8
2	4	638.4
3	1	474.4
3	2	758.5
3	4	607.5
4	1	422.7
4	2	796.3
4	3	685.4

Time to Transfer Between Papers (in hours)

Machine Number 1

paper(x)	paper(y)	Transition Time
1	2	0.3
1	3	1.6
1	4	1.8
2	1	0.8
2	3	1.7
2	4	1.6
3	1	1.0
3	2	0.3
3	4	1.1
4	1	1.8
4	2	1.4
4	3	1.9

Machine Number 2

paper(x)	paper(y)	Transition Time
1	2	1.9
1	3	1.7
1	4	1.2
2	1	1.0
2	3	1.3
2	4	0.9
3	1	1.2
3	2	1.6
3	4	0.9
4	1	1.8
4	2	0.1
4	3	0.5

Machine Number 3

paper(x)	paper(y)	Transition Time
1	2	0.6
1	3	1.9
1	4	1.7
2	1	1.6
2	3	0.8
2	4	1.0
3	1	0.8
3	2	1.9
3	4	1.4
4	1	1.5
4	2	0.3
4	3	1.1

DEMAND SCHEDULE

Requirements for Day 1

Paper Type 1 is needed in the following width(s):

22 Rolls of Width 88 Inches

Paper Type 3 is needed in the following width(s):

17 Rolls of Width 41 Inches

Requirements for Day 2

Paper Type 2 is needed in the following width(s):

15 Rolls of Width 95 Inches
45 Rolls of Width 120 Inches

Paper Type 4 is needed in the following width(s):

25 Rolls of Width 129 Inches

Requirements for Day 3

Paper Type 2 is needed in the following width(s):

38 Rolls of Width 58 Inches
59 Rolls of Width 120 Inches

Requirements for Day 4

Paper Type 1 is needed in the following width(s):

135 Rolls of Width 88 Inches

Paper Type 3 is needed in the following width(s):

74 Rolls of Width 142 Inches

Requirements for Day 5

Paper Type 2 is needed in the following width(s):

118 Rolls of Width 95 Inches

Paper Type 3 is needed in the following width(s):

41 Rolls of Width 142 Inches

Paper Type 4 is needed in the following width(s):

78 Rolls of Width 129 Inches

Widths Demanded for Each Paper

Paper 1 Widths	88			
Paper 2 Widths	95	58	120	
Paper 3 Widths	41	107	142	56
Paper 4 Widths	129	131		

Initial Configuration

Machine 1: paper 2
Machine 2: paper 2
Machine 3: paper 4

CUTTING PATTERNS

Patterns for Machine 1 and Paper 1
Paper Widths (in inches):

88	Cost($)
1	2219.20

Patterns for Machine 1 and Paper 2
Paper Widths (in inches):

95	58	120	Cost($)
0	0	1	2680.00
0	2	0	2622.40
1	0	0	2320.00

Patterns for Machine 1 and Paper 3
Paper Widths (in inches):

41	107	142	56	Cost($)
0	0	0	2	2564.80
0	1	0	0	2492.80
1	0	0	1	2348.80
3	0	0	0	2723.20

Patterns for Machine 1 and Paper 4
Paper Widths (in inches):

129	131	Cost($)
0	1	2838.40
1	0	2809.60

Patterns for Machine 2 and Paper 1
Paper Widths (in inches):

88	Cost($)
2	4032.40

Patterns for Machine 2 and Paper 2
Paper Widths (in inches):

95	58	120	Cost($)
0	1	1	4061.20
0	3	0	4003.60
1	2	0	4536.40
2	0	0	4234.00

Patterns for Machine 2 and Paper 3
Paper Widths (in inches):

41	107	142	56	Cost($)
0	0	1	1	4349.20
0	2	0	0	4579.60
1	0	0	3	4507.60
1	0	1	0	4133.20
1	1	0	1	4435.60
2	0	0	2	4291.60
2	1	0	0	4219.60
3	0	0	1	4075.60
5	0	0	0	4450.00

Patterns for Machine 2 and Paper 4
Paper Widths (in inches):

129	131	Cost($)
0	1	3384.40
1	0	3355.60

Patterns for Machine 3 and Paper 1
Paper Widths (in inches):

88	Cost($)
2	3983.40

Patterns for Machine 3 and Paper 2
Paper Widths (in inches):

95	58	120	Cost($)
0	1	1	4012.20
0	3	0	3954.60
1	1	0	3652.20
2	0	0	4185.00

Patterns for Machine 3 and Paper 3
Paper Widths (in inches):

41	107	142	56	Cost($)
0	0	0	3	3868.20
0	0	1	1	4300.20
1	0	1	0	4084.20
1	1	0	1	4386.60
2	0	0	2	4242.60
2	1	0	0	4170.60
3	0	0	1	4026.60
5	0	0	0	4401.00

Patterns for Machine 3 and Paper 4
Paper Widths (in inches):

129	131	Cost($)
0	1	3335.40
1	0	3306.60

```
REEL PRODUCTION

Day  1

   Machine 1

       Paper Type  2 Reels   W1=  95      W2=  58      W3= 120
                   24          0            0            1

   Machine 2

       Paper Type  2 Reels   W1=  95      W2=  58      W3= 120
                   18          0            1            1
       Paper Type  3 Reels   W1=  41      W2= 107      W3= 142      W4=  56
                   17          1            0            1            0

   Machine 3

       Paper Type  1 Reels   W1=  88
                   11          2
       Paper Type  2 Reels   W1=  95      W2=  58      W3= 120
                    7          0            1            1
                    8          2            0            0

Day  2

   Machine 1

       Paper Type  4 Reels   W1= 129      W2= 131
                   35          1            0

   Machine 2

       Paper Type  3 Reels   W1=  41      W2= 107      W3= 142      W4=  56
                   32          0            0            1            1
                    2          1            0            1            0

   Machine 3

       Paper Type  1 Reels   W1=  88
                   51          2

Day  3

   Machine 1

       Paper Type  2 Reels   W1=  95      W2=  58      W3= 120
                   25          0            0            1

   Machine 2

       Paper Type  2 Reels   W1=  95      W2=  58      W3= 120
                   30          0            1            1
       Paper Type  3 Reels   W1=  41      W2= 107      W3= 142      W4=  56
                    5          0            0            1            1
                    1          1            0            1            0
```

```
Machine 3

     Paper Type  1 Reels   W1=  88
                  16          2
     Paper Type  3 Reels   W1=  41     W2= 107     W3= 142     W4=  56
                   8          0           0           1           1
                   6          1           0           1           0


Day  4

   Machine 1

     Paper Type  4 Reels   W1= 129     W2= 131
                  32          1           0

   Machine 2

     Paper Type  1 Reels   W1=  88
                   1          2
     Paper Type  3 Reels   W1=  41     W2= 107     W3= 142     W4=  56
                  30          0           0           1           1

   Machine 3

     Paper Type  2 Reels   W1=  95     W2=  58     W3= 120
                  19          2           0           0
     Paper Type  3 Reels   W1=  41     W2= 107     W3= 142     W4=  56
                   3          1           0           1           0

Day  5

   Machine 1

     Paper Type  4 Reels   W1= 129     W2= 131
                  32          1           0

   Machine 2

     Paper Type  2 Reels   W1=  95     W2=  58     W3= 120
                  16          2           0           0
     Paper Type  3 Reels   W1=  41     W2= 107     W3= 142     W4=  56
                  11          0           0           1           1
     Paper Type  4 Reels   W1= 129     W2= 131
                   4          1           0

   Machine 3

     Paper Type  2 Reels   W1=  95     W2=  58     W3= 120
                  24          2           0           0

ROLL RECORD

Day  1

    Paper Type  1  (in rolls)

           Width      Prod.     B.Inv.    E.Inv.     Dmd.      Rej.
            88         22          0         0        22         0
```

Paper Type 2 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
95	16	0	16	0	0
58	25	0	25	0	0
120	49	0	49	0	0

Paper Type 3 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
41	17	0	0	17	0
107	0	0	0	0	0
142	17	0	17	0	0
56	0	0	0	0	0

Paper Type 4 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
129	0	0	0	0	0
131	0	0	0	0	0

Day 2

Paper Type 1 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
88	102	0	102	0	0

Paper Type 2 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
95	0	16	1	15	0
58	0	25	25	0	0
120	0	49	4	45	0

Paper Type 3 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
41	2	0	2	0	0
107	0	0	0	0	0
142	34	17	51	0	0
56	32	0	32	0	0

Paper Type 4 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
129	35	0	10	25	0
131	0	0	0	0	0

Day 3

Paper Type 1 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
88	32	102	134	0	0

Paper Type 2 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
95	0	1	1	0	0
58	30	25	17	38	0
120	55	4	0	59	0

Paper Type 3 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
41	7	2	9	0	0
107	0	0	0	0	0
142	20	51	71	0	0
56	13	32	45	0	0

Paper Type 4 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
129	0	10	10	0	0
131	0	0	0	0	0

Day 4

Paper Type 1 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
88	2	134	1	135	0

Paper Type 2 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
95	38	1	39	0	0
58	0	17	17	0	0
120	0	0	0	0	0

Paper Type 3 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
41	3	9	12	0	0
107	0	0	0	0	0
142	33	71	30	74	0
56	30	45	75	0	0

Paper Type 4 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
129	32	10	42	0	0
131	0	0	0	0	0

Day 5

Paper Type 1 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
88	0	1	1	0	0

Paper Type 2 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
95	80	39	1	118	0
58	0	17	17	0	0
120	0	0	0	0	0

Paper Type 3 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
41	0	12	12	0	0
107	0	0	0	0	0
142	11	30	0	41	0
56	11	75	86	0	0

Paper Type 4 (in rolls)

Width	Prod.	B.Inv.	E.Inv.	Dmd.	Rej.
129	36	42	0	78	0
131	0	0	0	0	0

COST SUMMARY

Cost of Transitions:	4311.34
Cost of Production:	1734893.38
Cost of Lost Sales:	0.00
Value of Recyled Inven.:	-93268.80

Total Cost of the Schedule is: 1645936.00 Dollars

SLACK INFORMATION

Day 1

Accumulated Slack on Machine # 1 is: 0.8 Hours
Accumulated Slack on Machine # 2 is: 0.0 Hours
Accumulated Slack on Machine # 3 is: 1.6 Hours

Day 2

Accumulated Slack on Machine # 1 is: 0.3 Hours
Accumulated Slack on Machine # 2 is: 0.2 Hours
Accumulated Slack on Machine # 3 is: 0.1 Hours

Day 3

Accumulated Slack on Machine # 1 is: 0.4 Hours
Accumulated Slack on Machine # 2 is: 0.4 Hours
Accumulated Slack on Machine # 3 is: 0.2 Hours

Day 4

Accumulated Slack on Machine # 1 is: 0.4 Hours
Accumulated Slack on Machine # 2 is: 0.0 Hours
Accumulated Slack on Machine # 3 is: 0.3 Hours

Day 5

Accumulated Slack on Machine # 1 is: 2.0 Hours
Accumulated Slack on Machine # 2 is: 1.6 Hours
Accumulated Slack on Machine # 3 is: 0.3 Hours

PRODUCTION SEQUENCE

Day 1

Machine 1
Sequence is: 2 4

Machine 2
Sequence is: 2 3

Machine 3
Sequence is: 4 2 1

Day 2

Machine 1
Sequence is: 4

Machine 2
Sequence is: 3

Machine 3
Sequence is: 1

Day 3

Machine 1
Sequence is: 4 2

Machine 2
Sequence is: 3 2

Machine 3
Sequence is: 1 3

Day 4

Machine 1
Sequence is: 2 4

Machine 2
Sequence is: 2 3 1

Machine 3
Sequence is: 3 2

Day 5

Machine 1
Sequence is: 4

Machine 2
Sequence is: 1 3 4 2

Machine 3
Sequence is: 2

APPENDIX B: COMPUTER CODE

```
      PROGRAM EXX
      REAL MAXTT,MINTT,MAXPT,MINPT
      DOUBLE PRECISION IX1
      CHARACTER*12  NAMSTR
        INTEGER STATUS
      DIMENSION IWI(15,4)
      WRITE(*,19)
19    FORMAT(5(/),5X,'What is the name of the problem ?')
        NAMSTR = 'TEMP'
      WRITE(*,20)
 20   FORMAT(3X,'How many periods are there? ')
      READ(*,*) IT
      WRITE(*,21)
 21   FORMAT(3X,'How many products are there? ')
      READ(*,*) N
      WRITE(*,22)
 22   FORMAT(3X,'How many machines are there? ')
      READ(*,*) M
      WRITE(*,23)
 23   FORMAT(3X,'What is the level of demand? ',
     1 '  (1 = Low; 2 = Medium; 3 = High)')
      READ(*,*) IOPT3
      WRITE(*,52)
  52  FORMAT(3X,'Random Number Seed? ')
      READ(*,*) IX1
        NWMAX = 4
        DAYL = 24
        MAXMW = 300
        MINMW =  75
        MAXPW = 300
        MINPW = 20
        AMTCST = 14.4
        ALABOR = 7.0
        CUTCST = 7.0
        MAXTC = 2 * 400
        MINTC = 0.1 * 400
        CLOST = 140000.0
        OPEN(88,FILE='EXX.FLE')
        WRITE(88,550)IT,N,M,IOPT3,IX1,ITOYS
550     FORMAT(I2,/,I2,/,I2,/,I2,/,D15.8,/,I2)
        CLOSE(88)
        IPERIOD = 1
        ITERAT = -1
        OPEN(88,FILE='INSUR.FLE')
        WRITE(88,505)IPERIOD,ITERAT
505     FORMAT(5X,I3,5X,I3)
        CLOSE(88)
        MAXTT = 2.0
        MINTT = 0.1
        MAXPT = 1.0
        MINPT = 0.5
        Y = 500.0
        HPR = (MAXPT+MINPT)/2.0
        AVPW= (MAXPW+MINPW)/2.0
        AVMW= (MAXMW+MINMW)/2.0
        ROPRE = AVMW/AVPW
        MINSIZ = 8
        MINR = NINT((MINSIZ/HPR)*ROPRE)
        IF(IOPT3.EQ.1)THEN
        MAXO = 4
        UTIL = 0.3
             ENDIF
```

```
          IF(IOPT3.EQ.2)THEN
          MAXO = 4
          UTIL = 0.6
               ENDIF
          IF(IOPT3.EQ.3)THEN
          MAXO = 4
          UTIL = 0.9
               ENDIF
          MAXR = NINT(
     X    4.0*UTIL*(M/1.0)*24.0*(1/HPR)*
     X      (1.0/MAXO)*ROPRE) - MINR
       CALL PRO(IT,N,M,IOPT3,IX1,NWMAX,
     X DAYL,MAXO,MINR,MAXR,MAXMW,MINMW,MAXPW,MINPW,
     X MAXTC,MINTC,CUTCST,MAXTT,MINTT,
     X MAXPT,MINPT,IWI,NAMSTR,ALABOR,AMTCST)
          OPEN(88,FILE='DELTA.FLE')
          OPEN(90,FILE='DELTAP.FLE')
        DO 64, J=1, IT
        DO 64, J1=1, M
        DO 64, J2=1, N
          WRITE(88,5511)INUM
          WRITE(90,5511)INUM
5511      FORMAT(5X,I1)
64     CONTINUE
          CLOSE(88)
          CLOSE(90)
          OPEN(88,FILE='ICON.FLE')
          INUM= 0
          WRITE(88,5511)INUM
          CLOSE(88)
          IOPT1 = 1
          LOLIM = 0
          IUPLIM = 1
      CALL HETEST( IWI,
     X             Y,CLOST,
     X             FACTOR,LOLIM,IUPLIM,
     X             AMTCST)
        STOP
C            LAST CARD OF EXX
        END
C
      SUBROUTINE HETEST( IWI,
     X  Y,CLOST,
     X  FACTOR,LOLIM,IUPLIM,
     X  AMTCST )
        DIMENSION IIW(15),IQK(15,7),OBJ(15,15,7),
     X   TRANS(15,15,7),
     X  ANIT(7,15),OBJX(200,15,7),IDOT(4,200,7,15),CAP(15,7),
     X  IDMD(14,15,4), IV(0:15,16,7,0:14),MW(7),
     X  ININV(15,4),RHS(14,7)

        INTEGER IWI(15,4),INV(0:14,15,4),
     X  ROLLPR(14,15,4,2),
     X  FACTOR,FLGDAY
          REAL IPROD(200,15,7,14)
        CALL READIT(DAYL,IT,N,M,IIW,IQK,
     1  OBJ,TRANS,ANIT,OBJX,IDOT,CAP,IDMD,MW)
          DO 6, I=0,IT
          DO 6, L=1,M
          DO 6, J=0,N
          DO 6, K=1,N+1
          DO 6, K1=1,IQK(J,L)
```

```
          IF((J.GT.0).AND.(I.GT.0))
     X          IPROD(K1,J,L,I) = 0
          IV(J,K,L,I) = 0
6         CONTINUE
          OPEN(12,FILE='IV.TEM')
          DO 8, I=0,IT
          DO 8, I1=1,M
          DO 8, I2=0,N
          DO 8, I3=1,N+1
          IF(I2.EQ.I3)GOTO 8
          IF((I2.EQ.0).AND.(I3.EQ.N+1))GOTO 8
          ATEM = IV(I2,I3,I1,I)/1.0
          WRITE(12,4455)ATEM
4455      FORMAT(F8.5)
8         CONTINUE
          CLOSE(12)
          FACTOR = 4
        WRITE(*,1212)
1212    FORMAT(3X,'What is the FACTOR value ? ')
        READ(*,*) FACTOR
          OPEN(88,FILE='FACTOR.FLE')
          WRITE(88,9011)FACTOR
9011      FORMAT(I3)
          CLOSE(88)
          FLGDAY = 1
          MIPCNT = 0
50        CONTINUE
          MIPCNT = MIPCNT + 1
          DO 55, K=1, 14
          DO 55, I=1, 15
          DO 55, J=1,  4
          DO 55, L=1,  2
          ININV(I,J) = 0
          RHS(K,I) = 0.0
          ROLLPR(K,I,J,L) = 0
55        CONTINUE
          DO 57, I=0, IT
          DO 57, I1=1,N
          DO 57, I2=1,IIW(I1)
          INV(I,I1,I2) = 0
57        CONTINUE
          INCOMP = 0
        CALL IP(IV,IPROD,ININV,Y,CLOST,
     X  FACTOR,IWI,LOLIM,IUPLIM,
     X  AMTCST,INCOMP)
          RETURN
C       LAST CARD OF HETEST
          END
C$INCLUDE READ_WRT.FOR
C$INCLUDE PRO.FOR
C
        SUBROUTINE READIT(DAYL,IT,N,M,IIW,IQK,
     1  OBJ,TRANS,ANIT,OBJX,IDOT,CAP,IDMD,MW)
        DIMENSION IIW(15),IQK(15,7),
     1  OBJ(15,15,7), TRANS(15,15,7), ANIT(7,15),OBJX(200,15,7),
     2  IDOT(4,200,7,15),CAP(15,7),IDMD(14,15,4),MW(7)
          OPEN(96,FILE='PROB_ST.FLE')
        READ(96,5000)DAYL
 5000   FORMAT(F7.4)
        READ(96,5001) IT,N,M
 5001   FORMAT(I10)
        READ(96,5002) (IIW(I),I=1,N)
```

```
 5002     FORMAT(I2)
          DO 1,I=1,N
          READ(96,5003) (IQK(I,J),J=1,M)
 5003     FORMAT(I2)
   1      CONTINUE
          DO 2 I=1, M
          READ(96,5004) (ANIT(I,J),J=1,N)
 5004     FORMAT(F6.1)
   2      CONTINUE
          DO 3 I=1, N
          READ(96,5005) (CAP(I,J),J=1,M)
 5005     FORMAT(F6.3)
   3      CONTINUE
          DO 4, I=1, N
          DO 4, I1=1, M
          DO 4, I2=1, IQK(I,I1)
          READ(96,5006) (IDOT(J,I2,I1,I),J=1,4)
 5006     FORMAT(I2)
   4      CONTINUE
          DO 5, I=1, N
          DO 5, I1=1, N
          READ(96,5007) (OBJ(I,I1,J),J=1, M)
 5007     FORMAT(F8.2)
   5      CONTINUE
          DO 6, I=1, N
          DO 6, I1=1, M
          READ(96,5008) (OBJX(J,I,I1),J=1, IQK(I,I1))
 5008     FORMAT(F8.2)
   6      CONTINUE
          DO 7, I=1, N
          DO 7, I1=1, N
          READ(96,5009) (TRANS(I,I1,J),J=1, M)
 5009     FORMAT(F7.1)
   7      CONTINUE
          ICNT = 0
          DO 8, I=1, IT
          DO 8, I1= 1, N
          DO 8, I2=1, IIW(I1)
          ICNT = ICNT + 1
          READ(96,5010)IDMD(I,I1,I2)
 5010     FORMAT(I6)
    8     CONTINUE
            READ(96,5011)(MW(I),I=1,M)
5011        FORMAT(I5)
            CLOSE(96)
          RETURN
C          LAST CARD OF READIT
          END
C
          SUBROUTINE WRITIT(DAYL,IT,N,M,IIW,IQK,CAP,IDOT,
     X                     OBJ,OBJX,TRANS,IDMD,ANIT,MW)
          DIMENSION ANIT(7,15),OBJ(15,15,7),
     1    TRANS(15,15,7),OBJX(200,15,7),
     2    IDOT(4,200,7,15), CAP(15,7),IIW(15),
     3    IQK(15,7),MW(7),
     4    IWI(15,4),IDMD(14,15,4)
           OPEN(96,FILE='PROB_ST.FLE')
          WRITE(96,5000) DAYL
 5000     FORMAT(F7.4)
          WRITE(96,5001) IT,N,M
 5001     FORMAT(I10)
          WRITE(96,5002) (IIW(I),I=1,N)
```

```
 5002   FORMAT(I2)
        DO 1,I=1,N
        WRITE(96,5003) (IQK(I,J),J=1,M)
 5003   FORMAT(I2)
   1    CONTINUE
        DO 2 I=1, M
        WRITE(96,5004) (ANIT(I,J),J=1,N)
 5004   FORMAT(F6.1)
   2    CONTINUE
        DO 3 I=1, N
        WRITE(96,5005) (CAP(I,J),J=1,M)
 5005   FORMAT(F6.3)
   3    CONTINUE
        DO 4, I=1, N
        DO 4, I1=1, M
        DO 4, I2=1, IQK(I,I1)
        WRITE(96,5006) (IDOT(J,I2,I1,I),J=1,4)
 5006   FORMAT(I2)
   4    CONTINUE
        DO 5, I=1, N
        DO 5, I1=1, N
        WRITE(96,5007) (OBJ(I,I1,J),J=1, M)
 5007   FORMAT(F8.2)
   5    CONTINUE
        DO 6, I=1, N
        DO 6, I1=1, M
        WRITE(96,5008) (OBJX(J,I,I1),J=1, IQK(I,I1))
 5008   FORMAT(F8.2)
   6    CONTINUE
        DO 7, I=1, N
        DO 7, I1=1, N
        WRITE(96,5009) (TRANS(I,I1,J),J=1, M)
 5009   FORMAT(F7.1)
   7    CONTINUE
        DO 8, I=1, IT
        DO 8, I1=1, N
        DO 8, I2=1, IIW(I1)
        WRITE(96,5010)IDMD(I,I1,I2)
 5010   FORMAT(I6)
    8   CONTINUE
          WRITE(96,5011)(MW(I),I=1,M)
5011      FORMAT(I5)
        CLOSE(96)
        RETURN
C       LAST CARD OF WRITIT
        END
C
        SUBROUTINE PRO(IT,N,M,IOPT3,IX,NWMAX,
     X  DAYL,MAXO,MINR,MAXR,MAXMW,MINMW,MAXPW,MINPW,
     X  MAXTC,MINTC,CUTCST,MAXTT,MINTT,
     X  MAXPT,MINPT,IWI,NAMSTR,ALABOR,AMTCST)
        REAL MAXTT,MINTT,MAXPT,MINPT
        DOUBLE PRECISION  IX
        CHARACTER*12  NAMSTR
        DIMENSION ANIT(7,15),OBJ(15,15,7),
     1  TRANS(15,15,7),OBJX(200,15,7),
     2  IDOT(4,200,7,15), CAP(15,7),IIW(15),
     3  IQK(15,7),MW(7), IWI(15,4),IDMD(14,15,4)
          IXSAVE = IX
        CALL MACHW(M,IX,MW,MAXMW,MINMW)
        CALL PAPW(M,MW,IX,IIW,N,IWI,NWMAX,MAXPW,MINPW)
        CALL INIT1(ANIT,OBJ,TRANS,IQK,IIW,
```

```
     1     IDOT,IT,N,M,OBJX,IDMD)
          CALL DEMAND(IT,IIW,MAXO,MINR,MAXR,IX,N,
     X               NWMAX,IDMD)
          CALL PATRNS(N,M,IWI,MW,IDOT,IQK)
          CALL OBJF(OBJ,N,M,IX,MAXTC,MINTC)
          CALL MASSAG(OBJ,N,M)
          CALL OBJFX(OBJX,IDOT,MW,IQK,IWI,M,CUTCST,
     X               ALABOR,AMTCST)
          CALL TRANSF(TRANS,M,IX,MAXTT,MINTT)
          CALL CAPF(CAP,N,M,MAXPT,MINPT,IX)
          CALL ICOND(M,N,ANIT,IX)
          CALL SCREEN(M,MW,N,IIW,IT,IDMD,IWI,
     XOBJ,TRANS,OBJX,CAP,IQK,IDOT,ANIT)
             CALL REPORT(M,N,IIW,IT,
     X       IQK,MAXO,MINR,MAXR,MAXMW,MINMW,MAXPW,
     X       MINPW,IXSAVE,IOPT3,NAMSTR)
          CALL WRITIT(DAYL,IT,N,M,IIW,IQK,CAP,IDOT,
     X                 OBJ,OBJX,TRANS,IDMD,ANIT,MW)
          RETURN
          END

          SUBROUTINE INIT1(ANIT,OBJ,TRANS,IQK,IIW,
     1     IDOT,IT,N,M,OBJX,IDMD)
          DIMENSION ANIT(7,15),OBJ(15,15,7),
     1    TRANS(15,15,7),OBJX(200,15,7),
     2    IDOT(4,200,7,15), CAP(15,7),IIW(15),
     3    IQK(15,7),IDMD(14,15,4)
          DO 1, I=1, M
          DO 1, I1=1,N
          ANIT(I,I1) = 0.0
    1     CONTINUE
          DO 20, I=1, N
          DO 20, I1=1, N
          DO 20, I2=1, M
          OBJ(I,I1,I2) = 0.0
          TRANS(I,I1,I2) = 0.0
   20     CONTINUE
          DO 41, I=1, 4
          DO 41, I1=1, 200
          DO 41, I2=1, 7
          DO 41, I3=1, 15
          IDOT(I,I1,I2,I3) = 0
          OBJX(I1,I3,I2) = 0.0
          CAP(I3,I2) = 0.0
          IQK(I3,I2) = 0
   41     CONTINUE
          DO 48, I=1, IT
          DO 48, I1=1, N
          DO 48, I2=1, IIW(I1)
          IDMD(I,I1,I2) = 0.0
    48    CONTINUE
          RETURN
C          LAST CARD OF INIT1
          END

          SUBROUTINE MACHW(M,IX,MW,MAXMW,MINMW)
          DOUBLE PRECISION DRAND, IX
          DIMENSION MW(7)
            DO 5,I=1,10
            R = DRAND(IX)
5           CONTINUE
          DO 10, I=1, M
```

```
      R = DRAND(IX)
      MW(I) = (MAXMW-MINMW) * R + MINMW
 10   CONTINUE
      RETURN
C         LAST CARD OF MACHW
      END

      DOUBLE PRECISION FUNCTION DRAND(IX)
      DOUBLE PRECISION  A,P,IX,B15,B16,XHI,XALO,LEFTLO,FHI,K
      DATA A/16807.D0/,B15/32768.D0/,
     1 B16/65536.D0/,P/2147483647.D0/
      XHI=IX/B16
      XHI = XHI-DMOD(XHI,1.D0)
      XALO = (IX-XHI*B16)*A
      LEFTLO = XALO/B16
      LEFTLO = LEFTLO - DMOD(LEFTLO,1.D0)
      FHI = XHI*A+LEFTLO
      K=FHI/B15
      K = K-DMOD(K,1.D0)
      IX = (((XALO-LEFTLO*B16)-P)+(FHI-K*B15)*B16)+K
      IF(IX.LT.0.D0)IX=IX+P
      DRAND = IX*4.656612875D-10
      RETURN
C          LAST CARD OF DRAND
      END
C
      SUBROUTINE PAPW(M,MW,IX,IIW,N,IWI,NWMAX,MAXPW,MINPW)
      DOUBLE PRECISION DRAND, IX
      DIMENSION MW(7), IIW(15),IWI(15,4)
        DO 10,I=1,N
        DO 10,J=1,NWMAX
        IWI(I,J) = 0
10      CONTINUE
        DO 23,I=1,N
        R = DRAND(IX)
        DO 21, K=1,NWMAX
        IF((R.GT.(K-1.0)/(NWMAX+0.0))
     X.AND.(R.LE.K/(NWMAX+0.0)))THEN
        IIW(I) = K
        GOTO 22
        ENDIF
21      CONTINUE
22      CONTINUE
23      CONTINUE
      DO 30, I=1, N
      DO 30, I1=1, IIW(I)
 25   CONTINUE
      R = DRAND(IX)
      IWI(I,I1) = R * (MAXPW-MINPW) + MINPW
        DO 26, I2=1,I1-1
        IF(IWI(I,I2).EQ.IWI(I,I1))GOTO 25
26      CONTINUE
      DO 28, I2=1, M
      IF(IWI(I,I1).LE.MW(I2)) GOTO 29
 28   CONTINUE
      GOTO 25
 29   CONTINUE
 30   CONTINUE
      RETURN
C          LAST CARD OF PAPW
      END
C
```

```
      SUBROUTINE SCREEN(M,MW,N,IIW,IT,IDMD,IWI,
     X OBJ,TRANS,OBJX,CAP,IQK,IDOT,ANIT)
      DIMENSION ANIT(7,15),OBJ(15,15,7),
     1 TRANS(15,15,7),OBJX(200,15,7),
     2 IDOT(4,200,7,15), CAP(15,7),IIW(15),
     3 IQK(15,7),MW(7), IWI(15,4),IDMD(14,15,4)
        OPEN(44,FILE='SCREEN.FLE')
2       FORMAT(5X)
      WRITE(44,6000)
6000  FORMAT(20X,'      PROBLEM SPECIFICATIONS',/)
      WRITE(44,6001)
6001  FORMAT(15X,'Machine Number      Machine Width')
      DO 200, I=1, M
        WRITE(44,6002) I, MW(I)
6002    FORMAT(18X,I2,20X,I3)
200    CONTINUE
      WRITE(44,7015)
7015  FORMAT(5X,/,
     X'Time Required to Produce a Reel of Papers (in hours)',/)
        DO 210,I=1, M
        WRITE(44,7016)I
7016    FORMAT(5X, 'Machine Number',I2,/,
     X12X,'paper(x)',10X,'Production Time')
        DO 210,I1=1,N
        WRITE(44,7017)I1,CAP(I1,I)
7017    FORMAT(13X,I2,19X,F3.1)
210      CONTINUE
      WRITE(44,7001)
7001  FORMAT(/,15X,'Cost of Transferring Between Papers (in $s)')
        DO 230,I=1, M
        WRITE(44,7002)I
7002    FORMAT(5X, 'Machine Number',I2,/,
     X12X,'paper(x)',5X,'paper(y)',5X,'Transition Cost')
        DO 229,I1=1,N
        DO 229,I2=1,N
        IF(I1.NE.I2)THEN
        WRITE(44,7003)I1,I2,OBJ(I1,I2,I)
7003    FORMAT(14X,I1,12X,I2,14X,F6.1)
        ENDIF
229     CONTINUE
        WRITE(44,2)
230      CONTINUE
      WRITE(44,7010)
7010  FORMAT(15X,'Time to Transfer Between Papers (in hours)')
        DO 240,I=1, M
        WRITE(44,7011)I
7011    FORMAT(5X, 'Machine Number',I2,/,
     X12X,'paper(x)',5X,'paper(y)',5X,'Transition Time')
        DO 239,I1=1,N
        DO 239,I2=1,N
        IF(I1.NE.I2)THEN
        WRITE(44,7003)I1,I2,TRANS(I1,I2,I)
        ENDIF
239     CONTINUE
        WRITE(44,2)
240      CONTINUE
        WRITE(44,6003)
6003    FORMAT(25X,'DEMAND SCHEDULE',/)
        DO 300, J=1, IT
        WRITE(44,6007)J
6007    FORMAT(5X,'Requirements for Day ',I2,/)
        DO 300, I=1, N
```

```
            ITEM = 0
            DO 250, I4U=1,IIW(I)
            ITEM = IDMD(J,I,I4U) + ITEM
250         CONTINUE
            IF(ITEM.EQ.0)GOTO 299
            WRITE(44,6004) I
6004        FORMAT(10X,'Paper Type  ',I2,
     X      ' is needed in the following width(s):')
            DO 290, I1=1, IIW(I)
            IF(IDMD(J,I,I1).EQ.0)GOTO 289
            WRITE(44,6008)IDMD(J,I,I1),IWI(I,I1)
6008        FORMAT(15X,I4,' Rolls of Width ',I3,' Inches')
289          CONTINUE
290          CONTINUE
            WRITE(44,6009)
6009        FORMAT(5X)
299         CONTINUE
300          CONTINUE
            WRITE(44,9670)
9670        FORMAT(/,5X,'IIW =     (number of widths for each paper)',/)
            DO 302,I=1,N
            WRITE(44,9671)I,IIW(I)
9671        FORMAT(10X,'Paper type ',I2,'  Has ',I2,'  Widths')
302         CONTINUE
            WRITE(44,9672)
9672        FORMAT(/,5X,'IWI = ',/)
            DO 304,I=1,N
            WRITE(44,9673)I,(IWI(I,I1),I1=1,IIW(I))
9673        FORMAT(5X,'Paper ',I2,' Widths ',4(I4,4X))
304         CONTINUE
            WRITE(44,9674)
9674        FORMAT(/,5X,'INITIAL CONDITIONS')
            DO 306, I=1,M
            WRITE(44,9675)(ANIT(I,I1),I1=1,N)
9675        FORMAT(5X,'MACHINE   ',5(F5.2,3X))
306         CONTINUE
            WRITE(44,8000)M,N
8000        FORMAT(5X,'CUTTING PATTERNS:',/,
     X     15X, 'NUMBER OF MACHINES = M = ',I2,/,
     X     15X, 'NUMBER OF PRODUCTS = N = ',I2,/)
            WRITE(44,8011)
8011        FORMAT(5X,'NUMBER FOR ROLLS OF THE FOLLOWING WIDTHS:')
            DO 355, I=1,M
            DO 355, I1=1,N
            WRITE(44,9000)I,I1,MW(I)
9000        FORMAT(/,5X,'PATTERNS FOR MACHINE',I2,
     X      '  AND PAPER ',I2,//,
     X       15X,'Machine Width = ',I3,' inches',/,
     X       15X,'With Paper Widths (in inches):',/)
            WRITE(44,8012)(IWI(I1,I2),I2=1,IIW(I1))
8012        FORMAT(9X,I3,8X,I3,
     X      12X,I3,12X,I3)
            IF(IIW(I1).EQ.4)THEN
            DO 350,I3=1,IQK(I1,I)
            WRITE(44,9001)(IDOT(I4,I3,I,I1),I4=1,4),OBJX(I3,I1,I)
9001        FORMAT(9X,I2,9X,I2,13X,I2,13X,I2,7X,'COST = ',F7.2)
350         CONTINUE
            ENDIF
            IF(IIW(I1).EQ.3)THEN
            DO 351,I3=1,IQK(I1,I)
            WRITE(44,9002)(IDOT(I4,I3,I,I1),I4=1,3),OBJX(I3,I1,I)
9002        FORMAT(9X,I2,9X,I2,13X,I2,13X,2X,7X,'COST = ',F7.2)
```

```
351       CONTINUE
          ENDIF
          IF(IIW(I1).EQ.2)THEN
          DO 352,I3=1,IQK(I1,I)
          WRITE(44,9003)(IDOT(I4,I3,I,I1),I4=1,2),OBJX(I3,I1,I)
9003      FORMAT(9X,I2,9X,I2,13X,2X,13X,2X,7X,'COST = ',F7.2)
352       CONTINUE
          ENDIF
          IF(IIW(I1).EQ.1)THEN
          DO 353,I3=1,IQK(I1,I)
          WRITE(44,9004)(IDOT(I4,I3,I,I1),I4=1,1),OBJX(I3,I1,I)
9004      FORMAT(9X,I2,9X,2X,13X,2X,13X,2X,7X,'COST = ',F7.2)
353       CONTINUE
          ENDIF
355       CONTINUE
          CLOSE(44)
           RETURN
C              LAST CARD OF SCREEN
          END
C
          SUBROUTINE DEMAND(IT,IIW,MAXO,MINR,
     X      MAXR,IX,N,NWMAX,IDMD)
          DOUBLE PRECISION DRAND, IX
          DIMENSION IIW(15),IDMD(14,15,4)
          DO 10, I=1,IT
          DO 10, I1=1,N
          DO 10, I2=1,NWMAX
          IDMD(I,I1,I2) = 0
10        CONTINUE
          DO 100, I=1,IT
          R = DRAND(IX)
          DO 21, K=1,MAXO + 1
          IF((R.GT.(K-1.0)/(MAXO+1)).AND.(R.LE.K/(MAXO+1.0)))THEN
          NORDER = K-1
          IF((I.EQ.1).AND.(NORDER.GT.2))NORDER = 2
          IF((I.EQ.2).AND.(NORDER.GT.3))NORDER = 3
          GOTO 22
          ENDIF
21        CONTINUE
22        CONTINUE
          IF(NORDER.EQ.0)GOTO 100
          DO 50,  I1=1, NORDER
          R = DRAND(IX)
          DO 31, K=1,N
          IF((R.GT.(K-1.0)/(N)).AND.(R.LE.K/(N+0.0)))THEN
          IPICK  = K
          GOTO 32
          ENDIF
31        CONTINUE
32        CONTINUE
          R = DRAND(IX)
          IROLLS = NINT(((MAXR-MINR)/1.0)*R)+MINR
          IF(I.EQ.1)IROLLS = NINT(IROLLS/3.0)
          IF(I.EQ.2)IROLLS = NINT(IROLLS*(2.0/3.0))
          R = DRAND(IX)
          DO 46, K=1,IIW(IPICK)
          IF((R.GT.(K-1.0)/(IIW(IPICK)))
     X.AND.(R.LE.K/(IIW(IPICK))))THEN
          IWIDTH = K
          GOTO 47
          ENDIF
46        CONTINUE
```

```
47        CONTINUE
          IDMD(I,IPICK,IWIDTH) = IDMD(I,IPICK,IWIDTH) + IROLLS
50        CONTINUE
100       CONTINUE
          RETURN
C           LAST CARD OF DEMAND
          END
C
          SUBROUTINE OBJF(OBJ,N,M,IX,MAXTC,MINTC)
        DOUBLE PRECISION DRAND, IX
        DIMENSION OBJ(15,15,7)
          DO 100, I=1,M
          DO 100, I1=1,N
          DO 100, I2=1,N
          IF(I1.NE.I2)THEN
          R = DRAND(IX)
          OBJ(I1,I2,I) = R*(MAXTC-MINTC)+MINTC
          ENDIF
100       CONTINUE
               RETURN
C             LAST CARD OF OBJF
               END
C
          SUBROUTINE TRANSF(TRANS,M,IX,MAXTT,MINTT)
        REAL MAXTT,MINTT
        DOUBLE PRECISION DRAND, IX
        DIMENSION TRANS(15,15,7)
          DO 100, I=1,M
          DO 100, I1=1,15
          DO 100, I2=1,15
          IF(I1.NE.I2)THEN
          R = DRAND(IX)
          TRANS(I1,I2,I) = R*(MAXTT-MINTT)+MINTT
          ENDIF
100       CONTINUE
               RETURN
C             LAST CARD OF TRANSF.FOR
               END
C
          SUBROUTINE OBJFX(OBJX,IDOT,MW,IQK,IWI,
     X    M,CUTCST,ALABOR,AMTCST)
        DIMENSION OBJX(200,15,7),IDOT(4,200,7,15),
     1  IQK(15,7),MW(7), IWI(15,4)
          DO 1000, I=1, 15
          DO 1000, J=1, M
          DO 500,  K=1, IQK(I,J)
          LENGTH = 0
          DO 400, L=1,4
          LENGTH = LENGTH + IDOT(L,K,J,I)*IWI(I,L)
400       CONTINUE
          OBJX(K,I,J) = (MW(J)-LENGTH)*CUTCST
          OBJX(K,I,J) = OBJX(K,I,J) +
     X          (LENGTH/1.0)*(ALABOR+AMTCST)
          IF(OBJX(K,I,J).LT.0.0)THEN
          WRITE(*,5555)
5555      FORMAT(5X,'ERROR IN OBJFX.FOR',/,
     X5X,'PATTERN LONGER THAN THE WIDTH OF THE MACHINE')
          PAUSE
          ENDIF
500       CONTINUE
1000      CONTINUE
          RETURN
```

```
C        LAST CARD OF OBJFX
          END
C
          SUBROUTINE ICOND(M,N,ANIT,IX)
        DOUBLE PRECISION DRAND, IX
        DIMENSION ANIT(7,15)
          DO 100, J=1,M
          R = DRAND(IX)
          DO 21, K=1,N
          IF((R.GT.(K-1.0)/(N)).AND.(R.LE.(K)/(N)))THEN
          ANIT(J,K) = 1.0
          GOTO 22
          ENDIF
21        CONTINUE
22        CONTINUE
100       CONTINUE
          WRITE(44,9674)
9674      FORMAT(/,5X,'INITIAL CONDITIONS')

          RETURN
C              LAST CARD OF ICOND.FOR
          END
C
          SUBROUTINE CAPF(CAP,N,M,MAXPT,MINPT,IX)
        REAL MAXPT,MINPT
        DOUBLE PRECISION DRAND, IX
        DIMENSION CAP(15,7)
          DO 100, I=1,N
          DO 100, J=1,M
          R = DRAND(IX)
          CAP(I,J) = R*(MAXPT-MINPT) + MINPT
100       CONTINUE
          RETURN
C              LAST CARD OF CAPF.FOR
          END
C
C$INCLUDE REPORT.FOR
C$INCLUDE PATRNS.FOR
C$INCLUDE MASSAG.FOR
C
          SUBROUTINE REPORT(M,N,IIW,IT,
     X    IQK,MAXO,MINR,MAXR,MAXMW,MINMW,MAXPW,
     X    MINPW,IXSAVE,IOPT3,NAMSTR)
          CHARACTER*12 NAMSTR,UTIL
          DIMENSION  IIW(15),IQK(15,7)
          WRITE(*,555)
555       FORMAT(5X,'ENTERING REPORT')
          OPEN(43,FILE='REPORT.FLE')
          WRITE(43,1000)NAMSTR
1000      FORMAT(25X,'PROBLEM - ',A12)
          WRITE(43,1001)IXSAVE
1001      FORMAT(/,5X,'Random Number Seed = ',I3,/)
          WRITE(43,1010)IT,M,N
1010      FORMAT(5X,'The Number of Periods is:  ',I2,/,
     X           5X,'The Number of Machines is:  ',I1,/,
     X           5X,'Number of Types of Paper:  ',I2,/)
          WRITE(43,1002)MAXMW,MINMW,MAXPW,MINPW
1002      FORMAT(5X,'Maximum Machine Width = ',I4,/,
     X           5X,'Minimum Machine Width = ',I4,//,
     X           5X,'Maximum Paper Width   = ',I4,/,
     X           5X,'Minimum Paper Width   = ',I4)
          IF(IOPT3.EQ.1)UTIL = 'low'
```

```
      IF(IOPT3.EQ.2)UTIL = 'medium'
      IF(IOPT3.EQ.3)UTIL = 'high'
      WRITE(43,1003)UTIL,MAXO,MINR,MAXR
1003  FORMAT(/,5X,'Expected Level of Utilization = ',A12,/,
     X         5X,'Maximum Number of Orders/Day  = ',I3,/,
     X         5X,'Minimum Number of Rolls/Order = ',I3,/,
     X         5X,'Maximum Number of Rolls/Order = ',I3)
      CALL SIZE(IT,N,M,IQK,IIW)
      WRITE(43,1005)
1005  FORMAT(/,35X,'Constraint Legend')
      IBEG = 1
      IEND = 0
       WRITE(43,2000)
2000  FORMAT(/,5X,'Type of Constraint',15X,'Beginning ',
     X    10X,     'Ending ',/)
      DO 200, I=1,IT
      DO 200, I1=1,N
      DO 200, I2=1,IIW(I1)
      IEND = IEND+1
200   CONTINUE
      WRITE(43,2001)IBEG,IEND
2001  FORMAT(5X,'Reel to Roll Conversion',10X,I5,14X,I5)
      IBEG = IEND + 1
      DO 210, I=1,IT
      DO 210, I1=1,M
      IEND = IEND + 1
210   CONTINUE
      WRITE(43,2002)IBEG,IEND
2002  FORMAT(5X,'Capacity Constraints',13X,I5,14X,I5)
      IBEG = IEND + 1
      DO 230, I=1,IT
      DO 230, I1=1,N
      DO 230, I2=1,IIW(I1)
      IEND = IEND + 1
230   CONTINUE
      WRITE(43,2003)IBEG,IEND
2003  FORMAT(5X,'Demand Constraints',15X,I5,14X,I5)
      IBEG = IEND + 1
      DO 240, I=1,IT
      DO 240, I1=1,M
      DO 240, I2=1,N
      IEND = IEND + 1
240   CONTINUE
      WRITE(43,2004)IBEG,IEND
2004  FORMAT(5X,'Required Setups',18X,I5,14X,I5)
      IBEG = IEND + 1
      DO 250, I=1,IT
      DO 250, I1=1,M
      IEND = IEND + 1
250   CONTINUE
      WRITE(43,2005)IBEG,IEND
2005  FORMAT(5X,'Exit From the Null State ',8X,I5,14X,I5)
      IBEG = IEND + 1
      DO 260, I=1,IT
      DO 260, I1=1,M
      DO 260, I2=1,N
      IEND = IEND + 1
260   CONTINUE
      WRITE(43,2006)IBEG,IEND
2006  FORMAT(5X,'Matched Exit to Entrance',9X,I5,14X,I5)
      IBEG = IEND + 1
      DO 270, I=1,IT
```

```
        DO 270, I1=1,M
        DO 270, I2=1,N
        IEND = IEND + 1
270     CONTINUE
        WRITE(43,2007)IBEG,IEND
2007    FORMAT(5X,'Exit from Each State Visited',5X,I5,14X,I5)
        IBEG = IEND + 1
        DO 280, I=1,IT
        DO 280, I1=1,M
        DO 280, I2=0,N
        DO 280, I3=1,N+1
        IF(I2.NE.I3)THEN
        IF((I2.EQ.0).AND.(I3.EQ.N+1))GOTO 280
        IEND = IEND + 1
        ENDIF
280     CONTINUE
        WRITE(43,2008)IBEG,IEND
2008    FORMAT(5X,'Bread Crumb Constraints',10X,I5,14X,I5)
        IBEG = IEND + 1
       DO 285, I=1, IT
       DO 285, I1=1, M
       DO 285, I2=1, N
        IEND = IEND + 1
285    CONTINUE
        WRITE(43,2208)IBEG,IEND
2208    FORMAT(5X,'Required Trans. Before Prod.',
     X     5X,I5,14X,I5)
        IBEG = IEND + 1
        DO 345, I=1,IT
        DO 345,I1=1,M
        DO 345,I2=1,N
        IEND = IEND + 1
345      CONTINUE
        WRITE(43,2209)IBEG,IEND
2209    FORMAT(5X,'Minimum Production',15X,I5,14X,I5)
        WRITE(43,2009)IEND
2009    FORMAT(/,5X,'Total Number of Constraints',25X,I5)
        CLOSE(43)
        WRITE(*,556)
556     FORMAT(5X,'LEAVING REPORT')
        RETURN
C       LAST CARD OF REPORT
        END
C
        SUBROUTINE SIZE(IT,N,M,IQK,IIW)
        DIMENSION  IQK(15,7),IIW(15)
        WRITE(43,6000)
6000    FORMAT(/,5X,'For Integer Number of Reels',/,
     X          56X, '  Beg.   End',/)
        ICUT = 0
        DO 200, I=1,IT
        DO 200, I1=1,N
        DO 200,I2=1,M
        ICUT = ICUT + IQK(I1,I2)
200     CONTINUE
        IBG = 1
        IED = ICUT
        IROLLS = 0
        INV = 0
        DO 220,I=0,IT
        DO 220,I1=1,N
        IF(I.GT.0) IROLLS = IROLLS + IIW(I1)
```

```
          INV = INV + IIW(I1)
220       CONTINUE
          WRITE(43,5788)ICUT,IBG,IED
5788      FORMAT(5X,'Number of Cutting Patterns: ',17X,I4,
     X            3X,I4,3X,I4)
          IBG = IBG + ICUT
          IED = IED + IROLLS
          WRITE(43,6002)IROLLS,IBG,IED
6002      FORMAT(5X,'Number of Roll variables:',18X,I6,
     X            3X,I4,3X,I4)
          IBG = IBG + IROLLS
          IED = IED + INV
          WRITE(43,6003)INV,IBG, IED
6003      FORMAT(5X,'Number of Inventory Variables:',13X,I6,
     X            3X,I4,3X,I4)
          IBG = IBG + INV
          IED = IED + IROLLS
          WRITE(43,6004)IROLLS,IBG, IED
6004      FORMAT(5X,'Number of Rejected Rolls variables: ',7X,I6,
     X            3X,I4,3X,I4)
          IBG = IBG + IROLLS
          IED = IED + N*M
          WRITE(43,6005)N*M,IBG, IED
6005      FORMAT(5X,'Number of Machine Initialization Variables: ',I5,
     X            3X,I4,3X,I4)
          ITRAN = 0
        DO 630, I=1,IT
        DO 630, I1=1,M
        DO 630, I2=0,N
        DO 630, I3=1, N+1
        IF(I2.EQ.I3)GOTO 630
        IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 630
          ITRAN = ITRAN + 1
 630    CONTINUE
          IBG = IBG + N*M
          IED = IED + ITRAN
          WRITE(43,6006)ITRAN,IBG,IED
6006      FORMAT(5X,'Number of Transition Variables: ',12X,I5,
     X            3X,I4,3X,I4)
          IBG = IBG + ITRAN
          IED = IED + IT*M
          ISLACK = IT*M
          WRITE(43,6007) ISLACK,IBG,IED
6007      FORMAT(5X,'Number of Slack Variables:',18X,I5,
     X            3X,I4,3X,I4)
          ITOUR = 0
          DO 720, I=1, IT
          DO 720, I1=1, M
          DO 720, I2=0, N+1
          ITOUR = ITOUR + 1
  720     CONTINUE
          IBG = IBG + ISLACK
          IED = IED + ITOUR
          WRITE(43,6008)ITOUR,IBG,IED
6008      FORMAT(5X,'Number of Bread Crumb Variables:',12X,I5,
     X            3X,I4,3X,I4)
          IGCONS = 0
        DO 750, I=1, IT
        DO 750, I1=1, M
        DO 750, I2=1, N
        IGCONS = IGCONS + 1
  750   CONTINUE
```

```
         IBG = IBG + ITOUR
         IED = IED + IGCONS
         WRITE(43,6200)IGCONS,IBG,IED
6200     FORMAT(5X,'Number of Setups on Each Day:',15X,I5,
     X          3X,I4,3X,I4)
         ITOTAL = ICUT+2*IROLLS+IGCONS+
     X    INV + M*N+ITRAN+ISLACK+ITOUR
         WRITE(43,6009) ITOTAL
6009     FORMAT(/,5X,'Total Number of Variables: ',30X,I6)
C            LAST CARD OF SIZE
       END
C
         SUBROUTINE ELAPSE(IYEAR,IMONTH,IDAY, IHRS,IMINS,
     X                    ELHRS,ELMINS)
         INTEGER*2 IYEAR,IMONTH,IDAY,IHRS, IMINS,
     X                YEAR, MONTH, DAY, HRS,  MINS,
     X                SECS,HSECS
         INTEGER  ELHRS,ELMINS
         CALL GETDAT(YEAR,MONTH,DAY)
         CALL GETTIM( HRS, MINS, SECS, HSECS )
         IF(IYEAR.EQ.YEAR)THEN
         IF(IMONTH.EQ.MONTH)THEN
         IF(IDAY.EQ.DAY)THEN
         MINTOT= (HRS*60.0+MINS) - (IHRS*60.0+IMINS)
         CALL MNTOHS(MINTOT,ELHRS,ELMINS)
         RETURN
         ELSE
         MINTOT = 24.0*60.0-(IHRS*60.0+IMINS)
         MINTOT = (DAY-IDAY-1)*24.0*60.0 + MINTOT
         MINTOT = MINTOT+HRS*60.0+MINS
         CALL MNTOHS(MINTOT,ELHRS,ELMINS)
         RETURN
         ENDIF
         PAUSE'ELAPSE...CHANGE IN MONTH, ELAPSED TIME INVALID'
         ENDIF
         PAUSE'ELAPSE...CHANGE IN YEAR, ELAPSED TIME INVALID'
         ENDIF
         RETURN
C       LAST CARD OF ELAPSE
         END
C
         SUBROUTINE MNTOHS(MINTOT,ELHRS,ELMINS)
         INTEGER  ELHRS,ELMINS
         ELHRS = INT(MINTOT/60.0)
         ELMINS = INT(MINTOT-ELHRS*60.0)
         RETURN
C      LAST CARD OF MNTOHS
         END
C
         SUBROUTINE TIMING(ELHRS,ELMINS,EDHRS,EDMINS,
     X                       ETHRS,ETMINS,IYEAR,IMONTH,
     X                IDAY,IHRS, IMINS,MIPCNT)
         INTEGER*2 IHRS, IMINS, SECS, HSECS,
     X             LHRS, LMINS,
     X             IYEAR,IMONTH,IDAY,
     X             LYEAR,LMONTH,LDAY
         INTEGER  ELHRS,ELMINS,EDHRS,EDMINS,ETHRS,ETMINS
         OPEN(86,FILE='TIME.FLE')
         WRITE(86,53)
53       FORMAT(//,25X,'Solution Time Summary')
         WRITE(86,54)IYEAR,IMONTH,IDAY,IHRS, IMINS
54       FORMAT(///,5X,'Solution Procedure began on: ',/,
```

```
     X          5X,'Year:      ',I4,/,
     X          5X,'Month:     ',I2,/,
     X          5X,'Day:       ',I2,' at:',/,
     X          5X,'Hour:      ',I2,/,
     X          5X,'Minutes:   ',I2)
           WRITE(86,55)ELHRS,ELMINS
55         FORMAT(///,5X,'Time required to solve the LP: ',I3,
     X          ' Hours and ',I2,' Minutes')
           WRITE(86,56)EDHRS,EDMINS
56         FORMAT(/,5X,'Time required to solve dailies: ',I3,
     X          ' Hours and ',I2,' Minutes')
           WRITE(86,57)ETHRS,ETMINS
57         FORMAT(//,5X,'Total time to solve the heuristic: ',I3,
     X          ' Hours and ',I2,' Minutes')
           CALL GETDAT(LYEAR,LMONTH,LDAY)
           CALL GETTIM( LHRS, LMINS, SECS, HSECS )
           WRITE(86,58)LYEAR,LMONTH,LDAY,LHRS, LMINS
58         FORMAT(///,5X,'Solution Procedure ended on: ',/,
     X          5X,'Year:      ',I4,/,
     X          5X,'Month:     ',I2,/,
     X          5X,'Day:       ',I2,' at:',/,
     X          5X,'Hour:      ',I2,/,
     X          5X,'Minutes:   ',I2)
                WRITE(86,59)MIPCNT
 59        FORMAT(//,5X,'The large LP/MIP was solved:',I3,
     X           '  times')
           CLOSE(86)
C             LAST CARD OF TIMING
           END
C
           SUBROUTINE PATRNS(N,M,IWI,MW,IDOT,IQK)
           DIMENSION  IWI(15,4),MW(7),
     XIDOT(4,200,7,15),NROLLS(4),IQK(15,7)
           DO 1000, I=1,N
           DO 1000, J=1,M
           IQ = 0
           DO 5,L4=1,4
           NROLLS(L4) = 0
5          CONTINUE
           ITER1 = 0
           DO 10, L=1,4
           IF(IWI(I,L).EQ.0)GOTO 10
           ITER = MW(J)/(IWI(I,L) - 0.000001) + 1
           IF(ITER.GT.ITER1)ITER1 = ITER
10         CONTINUE
           ITER = ITER1
           DO 800, K1=0,ITER
           DO 700, K2=0,ITER
           DO 600, K3=0,ITER
           DO 500, K4=0,ITER
           ITOTAL = IWI(I,1)*K1+IWI(I,2)*K2+
     X  IWI(I,3)*K3+IWI(I,4)*K4
           IF(ITOTAL.GT.MW(J))GOTO 499
           NROLLS(1) = K1
           NROLLS(2) = K2
           NROLLS(3) = K3
           NROLLS(4) = K4
           CALL PATEST(NROLLS,I,J,IDOT,IQ)
499        CONTINUE
500        CONTINUE
600        CONTINUE
700        CONTINUE
```

```
800        CONTINUE
999        CONTINUE
           IQK(I,J) = IQ
1000       CONTINUE
           RETURN
C          LAST CARD OF PRTRNS
           END
C
           SUBROUTINE PATEST(NROLLS,I,J,IDOT,IQ)
           INTEGER Q
           DIMENSION NROLLS(4),IDOT(4,200,7,15)
           ITEM = 0
           DO 10, M1 = 1, 4
           ITEM = NROLLS(M1) + ITEM
10         CONTINUE
           IQTEM = IQ
           IF(ITEM.EQ.0)RETURN
20         CONTINUE
           IQ =IQTEM
           DO 200, Q=1, IQ
           IF((IDOT(1,Q,J,I).GE.NROLLS(1))
     X.AND.(IDOT(2,Q,J,I).GE.NROLLS(2))
     X.AND.(IDOT(3,Q,J,I).GE.NROLLS(3))
     X.AND.(IDOT(4,Q,J,I).GE.NROLLS(4)))RETURN
           IF((IDOT(1,Q,J,I).LE.NROLLS(1))
     X.AND.(IDOT(2,Q,J,I).LE.NROLLS(2))
     X.AND.(IDOT(3,Q,J,I).LE.NROLLS(3))
     X.AND.(IDOT(4,Q,J,I).LE.NROLLS(4)))THEN
           DO 185,IWQ=Q,IQ-1
           DO 184,L4=1,4
           IDOT(L4,IWQ,J,I) = IDOT(L4,IWQ+1,J,I)
184        CONTINUE
185        CONTINUE
           IQTEM = IQTEM - 1
           GOTO 20
           ENDIF
200        CONTINUE
           IF(IQ.GT.200)PAUSE'TOO MANY CUTTING PATTERNS'
           IQ = IQ + 1
           DO 220, L4=1,4
           IDOT(L4,IQ,J,I) = NROLLS(L4)
220        CONTINUE
           RETURN
C          LAST CARD OF PATEST
           END
C
           SUBROUTINE FINISH(REELPR,IT,M,N,IIW,
     X IQK,IWI,IDOT,ROLLPR,OBJ,IV,
     X IDMD,TRANS,CAP,OBJX,IOPT1,IPROD,CLOST,
     X SLACK,AMTCST )
         DIMENSION  IIW(15),IQK(15,7),OBJ(15,15,7), TRANS(15,15,7),
     X  OBJX(200,15,7),IDOT(4,200,7,15),CAP(15,7),
     X  IV(0:15,16,7,0:14),RHS(14,7),IDMD(14,15,4),SLACK(14,7)
        INTEGER REELPR(14,7,15,200),IWI(15,4),INV(0:14,15,4),
     X ROLLPR(14,15,4,2),SEQ(10)
           REAL IPROD(200,15,7,14)
           NWMAX = 4
           MAXN = 4
           WRITE(*,4545)
4545       FORMAT(5X,'ENTERING FINISH')
           DO 5,I=0,IT
           DO 5,I1=1,N
```

```
         DO 5,I2=1,IIW(I1)
         INV(I,I1,I2) = 0
5        CONTINUE
         DO 10, I=1,IT
         DO 10, I1=1,M
         DO 10, I2=1,N
         DO 10, I3=1,IQK(I2,I1)
         REELPR(I,I1,I2,I3)=REELPR(I,I1,I2,I3)+
     X     NINT(IPROD(I3,I2,I1,I))
10       CONTINUE
         OPEN(44,FILE='RESULTS.FLE')
         WRITE(44,4995)
4995     FORMAT(20X,'   REEL PRODUCTION')
         IF(IOPT1.EQ.0)THEN
         WRITE(44,4994)
4994     FORMAT(19X,'(optimizing procedure)')
         ENDIF
         IF(IOPT1.EQ.1)THEN
         WRITE(44,4993)
4993     FORMAT(20X,'(heuristic procedure)')
         ENDIF
         DO 101, I=1,IT
         WRITE(44,5000)I
5000     FORMAT(/,5X,'DAY ',I2)
         DO 100, I1=1, M
         WRITE(44,5001)I1
5001     FORMAT(10X,'MACHINE ',I1)
         DO 100, I2=1, N
         ITE1 = 1
         ITEM = 0
         DO 95, I5=1,IQK(I2,I1)
         IF(REELPR(I,I1,I2,I5).GT.0)ITEM=1
95       CONTINUE
         IF((ITEM.EQ.1).AND.(ITE1.EQ.1))THEN
         WRITE(44,5002)I2,(IWI(I2,I6),I6=1,4)
5002     FORMAT(7X,'PAPER TYPE ',I2,' REELS',
     X   '    W1= ',I3,
     X   '      W2= ',I3,'      W3= ',I3,'    W4= ',I3)
         ITE1 = 0
         ENDIF
         DO 100, I3=1, IQK(I2,I1)
         IF(REELPR(I,I1,I2,I3).GT.0)THEN
         WRITE(44,5003)REELPR(I,I1,I2,I3),
     X                (IDOT(I4,I3,I1,I2),I4=1,4)
5003     FORMAT(21X,I2,4(8X,I3))
         ENDIF
100      CONTINUE
101      CONTINUE
         DO 150, I=1,IT
         DO 150,I1=1,N
         DO 150,I2=1,IIW(I1)
         ROLLPR(I,I1,I2,1) = 0
         ROLLPR(I,I1,I2,2) = 0
         DO 150, J=1,IT
         DO 150,J1=1,M
         DO 150,J2=1,N
         DO 150,J3=1,IQK(J2,J1)
         IF((I.EQ.J).AND.(I1.EQ.J2))THEN
         ROLLPR(I,I1,I2,1)= ROLLPR(I,I1,I2,1)+
     X    REELPR(J,J1,J2,J3)*IDOT(I2,J3,J1,J2)
         ENDIF
150      CONTINUE
```

```
      DO 160,I=1,IT
      DO 160,I1=1,N
      DO 160,I2=1,IIW(I1)
      IENDIV=INV(I-1,I1,I2)-IDMD(I,I1,I2)+ROLLPR(I,I1,I2,1)
      IF(IENDIV.GE.0)THEN
      INV(I,I1,I2) = IENDIV
      ENDIF
      IF(IENDIV.LT.0)THEN
      INV(I,I1,I2) = 0
      ROLLPR(I,I1,I2,2) = -IENDIV
      ENDIF
160   CONTINUE
      WRITE(44,5999)
5999  FORMAT(/,24X,'ROLL RECORD',/)
      DO 300,I=1,IT
      WRITE(44,6000)I
6000  FORMAT(5X,'DAY ',I2)
      DO 300, I2=1,N
      WRITE(44,6002)I2
6002  FORMAT(10X,'PAPER TYPE ',I2,'  (in rolls)',/,14X,
     X'Width       Prod.      B.Inv.    E.Inv.      Dmd.       Rej.')
      DO 300,I3=1,IIW(I2)
      WRITE(44,6003)
     X  IWI(I2,I3), ROLLPR(I,I2,I3,1),
     X  INV((I-1),I2,I3),INV(I,I2,I3),IDMD(I,I2,I3),
     X  ROLLPR(I,I2,I3,2)
6003  FORMAT(14X,I3,8X,I3,7X,I3,8X,I3,7X,I3,7X,I3)
300   CONTINUE
      DO 375, I=1, IT
      DO 375, I1=1,M
      RHS(I,I1) = 0.0
      DO 375, I2=1,N
      DO 375,I3=1,N
      RHS(I,I1)=RHS(I,I1)+TRANS(I2,I3,I1)*IV(I2,I3,I1,I)
375   CONTINUE
      DO 377, I=1, IT
      DO 377, I1=1, M
      DO 377, I2=1, N
      DO 377, I3=1, IQK(I2,I1)
      RHS(I,I1)=RHS(I,I1)+CAP(I2,I1)*REELPR(I,I1,I2,I3)
377   CONTINUE
      CTRAN = 0.0
      CLOS = 0.0
      CPRO = 0.0
      COV = 0.0
      Z = 0.0
      DO 380, I=1, IT
      DO 380, I1=1,M
      DO 380, I2=0,N
      DO 380,I3=1,N+1
      Z = Z + OBJ(I2,I3,I1)*IV(I2,I3,I1,I)
      CTRAN = Z
380   CONTINUE
      DO 382, I=1, IT
      DO 382, I1=1, M
      DO 382, I2=1, N
      DO 382, I3=1, IQK(I2,I1)
       Z = Z + OBJX(I3,I2,I1)
     X *REELPR(I,I1,I2,I3)
382   CONTINUE
      CPRO = Z - CTRAN
      DO 384, I=1, IT
```

```
          DO 384, I2=1, N
          DO 384, I3=1, IIW(I2)
           Z = Z + CLOST * (ROLLPR(I,I2,I3,2)/1.0)
384       CONTINUE
          CLOS = Z - CTRAN - CPRO
          DO 386, I2=1, N
          DO 386, I3=1, IIW(I2)
          Z=Z-(IWI(I2,I3)*AMTCST*INV(IT,I2,I3))/1.0
386       CONTINUE
          COV = Z - CTRAN - CPRO - CLOS
          WRITE(44,5555)
5555      FORMAT(/,5X,'COST SUMMARY:',/)
          WRITE(44,6969)CTRAN,CPRO,CLOS,COV
6969      FORMAT(5X,'Cost of Transitions = ',F14.2,/,
     X           5X,'Cost of Production  = ',F14.2,/,
     X           5X,'Cost of Lost Sales  = ',F14.2,/,
     X           5X,'Cost of Over Prod.  = ',F14.2)
          WRITE(44,7005)Z
7005      FORMAT(//,10X,'Total Cost of the Schedule is: ',F14.2,
     X   '  Dollars',/)
          WRITE(44,7001)
7001      FORMAT(5X,'Slack Information')
          DO 400,I=1,IT
          WRITE(44,6000)I
          DO 400,I1=1,M
          ANUM = SLACK(I,I1)
          WRITE(44,7000)I1,ANUM
7000      FORMAT(5X,'Accumulated. Slack on Machine #',I2,
     X            ' is: ',F6.1,' Hours')
400       CONTINUE
          WRITE(44,2999)
2999      FORMAT(/,20X,'PRODUCTION SEQUENCE')
          DO 500,I=1,IT
          WRITE(44,3000)I
3000      FORMAT(/,5X,'DAY ',I2)
          DO 500,J=1,M
          WRITE(44,3001)J
3001      FORMAT(/,10X,'MACHINE ',I2)
          ICNT = 0
          IAM = 0
          DO 450, NO=1, MAXN
          DO 450, N1=1, N
          IF(IV(IAM,N1,J,I).EQ.1)THEN
          ICNT = ICNT + 1
          SEQ(ICNT) = N1
          IAM = N1
          ENDIF
450       CONTINUE
          WRITE(44,3002)(SEQ(N2),N2=1,ICNT)
3002      FORMAT(10X,'SEQUENCE IS: ',12(3X,I2))
500       CONTINUE
          CLOSE(44)
          RETURN
C         LAST CARD OF FINISH
          END
C
          SUBROUTINE MASSAG(OBJ,N,M)
          DIMENSION OBJ(15,15,7),TRAN(15,15)
          WRITE(*,55)
55        FORMAT(5X,'ENTERING MASSAG')
          OBJ(7,7,1) =       0.0
          SHORT = 0.0
```

```
          DO 200, I=1, M
125       CONTINUE
         DO 130, I3=1,N
         DO 130, I4=1,N
         TRAN(I3,I4) = OBJ(I3,I4,I)
130      CONTINUE
          DO 150, I1=1, N
          DO 150, I2=1, N
          IF(I1.EQ.I2)GOTO 150
          CALL DIJKST(TRAN,N,SHORT,I1,I2)
          IF(SHORT.LT.OBJ(I1,I2,I))THEN
               OBJ(I1,I2,I) = SHORT
               GOTO 125
               ENDIF
150       CONTINUE
200       CONTINUE
          RETURN
C         LAST CARD OF MASSAG
          END
C
          SUBROUTINE DIJKST(TRAN,N,SHORT,ISOUR,ISINK)
          DIMENSION DIST(15),LABEL(15),TRAN(15,15)
          DO 5, I=1,N
          DIST(I) = TRAN(ISOUR,I)
5         CONTINUE
          NMARK = ISOUR
          DO 10, I=1,N
          LABEL(I) = 0
10        CONTINUE
          LABEL(ISOUR) = 1
          DO 200, J=1,N
          MARK = 0
          TDIS = 1000000.0
          DO 150, J1=1,N
          IF((TDIS.GT.DIST(J1))
     X     .AND.(LABEL(J1).EQ.0))THEN
          MARK = J1
          TDIS = DIST(J1)
          ENDIF
150       CONTINUE
          IF(MARK.EQ.0)THEN
          PAUSE'ERROR IN DIJKST...1'
          ENDIF
          LABEL(MARK) = LABEL(NMARK) + 1
          IF(MARK.EQ.ISINK)GOTO 500
          NMARK = MARK
          DO 175, J3=1, N
          TEM = DIST(MARK) + TRAN(MARK,J3)
          IF((TEM.LT.DIST(J3))
     X    .AND.(LABEL(J3).EQ.0)) THEN
          DIST(J3) = TEM
          ENDIF
175       CONTINUE
200       CONTINUE
500       CONTINUE
          SHORT = DIST(ISINK)
          RETURN
C           LAST CARD OF DIJKST.FOR
          END
C
       SUBROUTINE HETEST( IWI,IOPT1,
     X  Y,CLOST,
```

```
     X  FACTOR,LOLIM,IUPLIM,
     X  AMTCST )
        DIMENSION IIW(15),IQK(15,7),OBJ(15,15,7),
     X    TRANS(15,15,7),
     X  ANIT(7,15),OBJX(200,15,7),IDOT(4,200,7,15),CAP(15,7),
     X  IDMD(14,15,4), IV(0:15,16,7,0:14),MW(7),
     X  ININV(15,4),RHS(14,7),SLACK(14,7)
        INTEGER REELPR(14,7,15,200),IWI(15,4),INV(0:14,15,4),
     X  ROLLPR(14,15,4,2),
     X  FACTOR,LOST(14,15,4,2),INCOMP
          REAL IPROD(200,15,7,14)
        CALL READIT(DAYL,IT,N,M,IIW,IQK,
     1  OBJ,TRANS,ANIT,OBJX,IDOT,CAP,IDMD,MW)
          DO 6, I=0,IT
          DO 6, L=1,M
          DO 6, J=0,N
          DO 6, K=1,N+1
          IV(J,K,L,I) = 0
6         CONTINUE
          INCOMP = IT+1
50        CONTINUE
          DO 55, K=1, 14
          DO 55, I=1, 15
          DO 55, J=1,  4
          DO 55, L=1,  2
          ININV(I,J) = 0
          RHS(K,I) = 0.0
          ROLLPR(K,I,J,L) = 0
55        CONTINUE
          DO 57, I=0, IT
          DO 57, I1=1,N
          DO 57, I2=1,IIW(I1)
          INV(I,I1,I2) = 0
57        CONTINUE
          CALL READZO(IT,N,M,
     X    IIW,IQK,INV,IPROD,IV,IFLG,LOST,
     X    SLACK,INCOMP )
        IF(IFLG.EQ.0)THEN
        CALL IP(IV,IPROD,ININV,Y,CLOST,
     X  FACTOR,IWI,LOLIM,IUPLIM,
     X  AMTCST,INCOMP)
          STOP
          ENDIF
          CALL FINISH(REELPR,IT,M,N,IIW,
     X IQK,IWI,IDOT,ROLLPR,OBJ,IV,
     X  IDMD,TRANS,CAP,OBJX,IOPT1,IPROD,CLOST,
     X  SLACK,AMTCST)
          RETURN
C       LAST CARD OF HETEST
          END
C
        SUBROUTINE READZO(IT,N,M,
     X  IIW,IQK,INV,IPROD,IV,IFLG,LOST,
     X  SLACK,INCOMP )
        DIMENSION  IIW(15), IQK(15,7),INV(0:14,15,4),
     X             TIV(0:15,16,7,0:14),SLACK(14,7),
     X              TIVT(0:15,16,7,0:14)
          INTEGER IV(0:15,16,7,0:14),
     X            LOST(14,15,4,2),
     X            DELTA(14,7,15),INCOMP
          CHARACTER*8 FIELD1,FIELD2,FIELD1A,FIELD2A
          CHARACTER*1  L(93)
```

```
           REAL IPROD(200,15,7,14),TDELTA(14,7,15)
           FIELD1 = '  NUMBER'
           FIELD2 = '  .....C'
           WRITE(*,9090)
9090       FORMAT(5X,'ENTERING READZO')
           DO 6, I=0,IT
           DO 6, L1=1,M
           DO 6, J=0,N
           DO 6, K=1,N+1
           TIVT(J,K,L1,I) = 0.0
6          CONTINUE
         DO 11, I0=1, IT
         DO 11, I1=1, N
         DO 11, I2=1, M
         DO 11, I3=1, IQK(I1,I2)
         IPROD(I3,I1,I2,I0) = 0.0
11       CONTINUE
 500     FORMAT(F7.2,5X,F10.2)
         OPEN(95,FILE='KRUS.TXT')
           ICNTT = 0
15         CONTINUE
           READ(95,1010)FIELD1A,FIELD2A,L
1010       FORMAT(A8,A8,93A1)
           IF(FIELD2A.NE.FIELD2)GOTO 15
           READ(95,1010)FIELD1A,FIELD2A,L
           EPSILN = 1/100000.0
           IFLGC = 0
         DO 20, I=1, IT
         DO 20, I1=1, N
         DO 20, I2=1, M
         DO 20, I3=1, IQK(I1,I2)
         CALL REDLIN(B)
           ITEM = NINT(B)
           ATEM = ABS(B-ITEM/1.0)
         IF(ATEM.GT.EPSILN)IFLGC = 1
         IPROD(I3,I1,I2,I) = B
 20      CONTINUE
         DO 40, I=1,IT
         DO 40, I1=1, N
         DO 40, I2=1, IIW(I1)
         CALL REDLIN(B)
  40     CONTINUE
         DO 50, I=0,IT
         DO 50, I1=1, N
         DO 50, I2=1, IIW(I1)
           CALL REDLIN(B)
           INV(I,I1,I2) = NINT(B)
 50        CONTINUE
           DO 60, I=1,IT
           DO 60, I1=1,N
           DO 60, I2=1,IIW(I1)
           CALL REDLIN(B)
           LOST(I,I1,I2,1) = NINT(B)
           LOST(I,I1,I2,2) = 0
 60        CONTINUE
           DO 65, I=1,M
           DO 65,I1=1,N
           CALL REDLIN(B)
             TIVT(I1,N+1,I,0) = B
             TIV(I1,N+1,I,0) = B
             IV(I1,N+1,I,0) = NINT(B)
 65        CONTINUE
```

```
          IFLGT = 0
4455      FORMAT(F8.5)
        OPEN(12,FILE='IV.TEM')
        DO 70, I=1,IT
        DO 70, I1=1,M
        DO 70, I2=0, N
        DO 70, I3=1, N+1
        IF(I2.EQ.I3)GOTO 70
        IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 70
            CALL REDLIN(B)
            READ(12,4455)TEMP
            IF((B.GT.0.0).AND.(B.LT.1.0))IFLGT=1
            TIVT(I2,I3,I1,I) = B
            IF((B.GT.0.0).OR.(TEMP.GT.0))
     X         TIV(I2,I3,I1,I) = 1.0
70      CONTINUE
          CLOSE(12)
        OPEN(12,FILE='IV.TEM')
        DO 71, I=1,IT
        DO 71, I1=1,M
        DO 71, I2=0, N
        DO 71, I3=1, N+1
        IF(I2.EQ.I3)GOTO 71
        IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 71
            WRITE(12,4455)TIV(I2,I3,I1,I)
71      CONTINUE
          CLOSE(12)
          DO 80, I=1,IT
          DO 80,I1=1,M
            CALL REDLIN(B)
          SLACK(I,I1) = B
80        CONTINUE
        DO 120, I=1, IT
        DO 120, I1=1, M
        DO 120, I2=0, N+1
            CALL REDLIN(B)
  120   CONTINUE
          IFGLD = 0
          OPEN(89,FILE='DELTAP.FLE')
        DO 150, I=1, IT
        DO 150, I1=1, M
        DO 150, I2=1, N
          CALL REDLIN(B)
          IF((B.GT.0.0).AND.(B.LT.1.0))IFGLD = 1
          IF((B.GT.0.0).AND.(B.LE.1.0))DELTA(I,I1,I2)=1
          TDELTA(I,I1,I2) = B
          READ(89,5511)INUM
5511      FORMAT(5X,I1)
          IF(INUM.EQ.1)DELTA(I,I1,I2)=1
  150   CONTINUE
          CLOSE(89)
        CLOSE(95)
          OPEN(89,FILE='DELTAP.FLE')
        DO 160, I=1, IT
        DO 160, I1=1, M
        DO 160, I2=1, N
          WRITE(89,5511)DELTA(I,I1,I2)
  160   CONTINUE
          CLOSE(89)
          IF(IFGLD.EQ.1)THEN
          CALL FLAGS(TIV,IV,TIVT,IT,N,M,IPROD,IQK,
     X                TDELTA,INCOMP)
```

```
              IFLG = 0
              CLOSE(12)
              RETURN
              ENDIF
         IF((IFLGC.EQ.0).AND.(IFLGT.EQ.0))THEN
         DO 171, I=1,M
         DO 171,I1=1,N
         IV(0,I1,I,0)=NINT(TIVT(0,I1,I,0))
171       CONTINUE
        DO 172, I=1,IT
        DO 172, I1=1,M
        DO 172, I2=0, N
        DO 172, I3=1, N+1
        IF(I2.NE.I3)THEN
        IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 172
          IV(I2,I3,I1,I)=NINT(TIVT(I2,I3,I1,I))
           ENDIF
172      CONTINUE
         OPEN(88,FILE='BINARY.FLE')
         WRITE(88,7723)
         CLOSE(88)
         OPEN(88,FILE='INTEGER.FLE')
         WRITE(88,7722)
7722     FORMAT(5X,'INTEGER.FLE')
         CLOSE(88)
         IFLG = 1
              RETURN
              ENDIF
         IF(IFLGT.EQ.0)THEN
         DO 175, I=1,M
         DO 175,I1=1,N
         IF(TIVT(0,I1,I,0).EQ.0)IV(0,I1,I,0)=3
         IF(TIVT(0,I1,I,0).EQ.1)IV(0,I1,I,0)=2
175       CONTINUE
        DO 180, I=1,IT
        DO 180, I1=1,M
        DO 180, I2=0, N
        DO 180, I3=1, N+1
        IF(I2.EQ.I3)GOTO 180
        IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 180
          IF(TIVT(I2,I3,I1,I).EQ.0)IV(I2,I3,I1,I)=3
          IF(TIVT(I2,I3,I1,I).EQ.1)IV(I2,I3,I1,I)=2
180      CONTINUE
         OPEN(88,FILE='BINARY.FLE')
         WRITE(88,7723)
7723     FORMAT(5X,'BINARY.FLE')
         CLOSE(88)
         OPEN(88,FILE='ICON.FLE')
         ICON = 2
         WRITE(88,5511)ICON
         CLOSE(88)
         IFLG = 0
         RETURN
         ENDIF
         CALL FLAGS(TIV,IV,TIVT,IT,N,M,IPROD,IQK,
     X              TDELTA,INCOMP)
         OPEN(88,FILE='ICON.FLE')
         ICON = 1
         WRITE(88,5511)ICON
         CLOSE(88)
         IFLG = 0
         RETURN
```

```
C         LAST CARD OF READZO
          END
C
          SUBROUTINE REDLIN(B)
          CHARACTER*1 L(64),C(13),LDIGIT(15)
          CHARACTER*8 FIELD(4)
          INTEGER INTG(13),MANT(13)
      DATA        LDIGIT /'0','1','2','3','4','5','6','7','8','9',
     X                   '+','-',' ','.','E'/
          READ(95,1010)FIELD,C,L
1010      FORMAT(4A8,13A1,64A1)
          IF(C(10).EQ.LDIGIT(15))THEN
               IF(C(11).EQ.LDIGIT(12))THEN
                    B = 0.0
                    RETURN
                    ENDIF
               PAUSE'PROBLEM IN REDLINE - E'
               STOP
               ENDIF
          IPOINT = 0
          DO 100,I=1,13
          IF(C(I).EQ.LDIGIT(14))THEN
              IPOINT = I
              GOTO 101
              ENDIF
100       CONTINUE
101       CONTINUE
          IF(IPOINT.EQ.0)THEN
          DO 200,I=1,13
            DO 195,I1=1,10
               IF(C(I).EQ.LDIGIT(I1))THEN
                    INTG(I) = I1-1
                    GOTO 200
                    ENDIF
195         CONTINUE
            IF(C(I).EQ.LDIGIT(13))THEN
               INTG(I) = 0
               GOTO 200
               ENDIF
          PAUSE'ERROR IN REDLINE - NON STANDARD INPUT'
200       CONTINUE
          B = 0.0
          DO 300,I=1,13
          B = B + INTG(I)*(10**(13-I))
300       CONTINUE
          ENDIF
          IF(IPOINT.NE.0)THEN
          DO 400,I=1,IPOINT-1
            DO 395,I1=1,10
               IF(C(I).EQ.LDIGIT(I1))THEN
                    INTG(I) = I1-1
                    GOTO 400
                    ENDIF
395         CONTINUE
            IF(C(I).EQ.LDIGIT(13))THEN
               INTG(I) = 0
               GOTO 400
               ENDIF
          PAUSE'ERROR IN REDLINE - NON STANDARD INPUT W'
400       CONTINUE
          DO 500,I=IPOINT+1,13
            DO 495,I1=1,10
```

```
             IF(C(I).EQ.LDIGIT(I1))THEN
                  MANT(I) = I1-1
                  GOTO 500
                  ENDIF
495         CONTINUE
            IF(C(I).EQ.LDIGIT(13))THEN
               MANT(I) = 0
               GOTO 500
               ENDIF
          PAUSE'ERROR IN REDLINE - NON STANDARD INPUT W'
500       CONTINUE
          B = 0.0
          DO 600,I=1,IPOINT - 1
          B = B + INTG(I)*(10**(IPOINT-1-I))
600       CONTINUE
          DO 700,I=IPOINT+1,13
          B = B + (MANT(I)*(0.1**(I-IPOINT)))/1.0
700       CONTINUE
          ENDIF
          RETURN
C            LAST CARD OF REDLIN
          END
C
C$INCLUDE FLAGS.FOR
C
          SUBROUTINE FLAGS(TIV,IV,TIVT,IT,N,M,IPROD,IQK,
     X                     TDELTA,INCOMP)
           DIMENSION TIV(0:15,16,7,0:14),IV(0:15,16,7,0:14),
     X                TIVT(0:15,16,7,0:14),IQK(15,7)
           INTEGER DELTA(14,7,15),MSTATUS(7)
           REAL IPROD(200,15,7,14),TDELTA(14,7,15)
           INTEGER COMPLE,STATUS,INCOMP
          WRITE(*,9090)
9090      FORMAT(5X,'ENTERING FLAGS')
          DO 10, I=0,IT
          DO 10, I1=1,M
          DO 10, I2=0,N
          DO 10, I3=1,N+1
          MSTATUS(I1) = 0
          IV(I2,I3,I1,I) = 0
10         CONTINUE
          INCOMP = 0
          DO 15, I=1,IT
          INCOMP = INCOMP + 1
          DO 15, I1=1,M
          DO 15, I2=0,N
          DO 15, I3=1,N+1
          IF(I2.EQ.I3)GOTO 15
          IF((I2.EQ.0).AND.(I3.EQ.N+1))GOTO 15
          IF((TIVT(I2,I3,I1,I).GT.0.0).AND.
     X       (TIVT(I2,I3,I1,I).LT.1.0))GOTO 16
15         CONTINUE
16         CONTINUE

          WRITE(*,5566)INCOMP
5566      FORMAT(5X,'INCOMP = ', I3)
          DO 300, I1=1,M
          I = 0
115       CONTINUE
          I = I + 1
          IBEG = 0
120       CONTINUE
```

```
      DO 200, J1=1,N+1
      IF(TIVT(IBEG,J1,I1,I).EQ.1.0)THEN
        IF(J1.LT.N+1)THEN
        IV(IBEG,J1,I1,I) = 2
        IBEG = J1
        GOTO 120
        ELSE
           IF(I.GE.INCOMP)GOTO 201
           IV(IBEG,J1,I1,I) = 2
           GOTO 115
        ENDIF
      ENDIF
200   CONTINUE
201   CONTINUE
      DO 250, J1=1,N+1
      IV(IBEG,J1,I1,I) = 1
250   CONTINUE
300   CONTINUE
      OPEN(88,FILE='INSUR.FLE')
      READ(88,505)IPERIOD,ITERAT
      CLOSE(88)
505   FORMAT(5X,I3,5X,I3)
      IF(IPERIOD.EQ.INCOMP)THEN
      ITERAT = ITERAT + 1
      ELSE
      ITERAT = -1
      IPERIOD = INCOMP
      ENDIF
      OPEN(88,FILE='INSUR.FLE')
      WRITE(88,505)IPERIOD,ITERAT
      CLOSE(88)
      ITERAT = ITERAT ** 6
      WRITE(*,7733)ITERAT
7733  FORMAT(5X,'ITERAT = ',I4)
      DO 350, I=1,INCOMP
      DO 350, I1=1,M
      DO 350, I2=0,N
      DO 350, I3=1,N+1
      IF(I2.EQ.I3)GOTO 350
      IF((I2.EQ.0).AND.(I3.EQ.N+1))GOTO 350
      IF(I.LT.INCOMP)THEN
      IF(IV(I2,I3,I1,I).EQ.0)IV(I2,I3,I1,I)= 3
      ENDIF
      IF((I.EQ.INCOMP).AND.(ITERAT.GT.0).AND.
     X    (IV(I2,I3,I1,I).EQ.0))THEN
      ITERAT = ITERAT - 1
      IV(I2,I3,I1,I)= 1
      ENDIF
350     CONTINUE
      IF(INCOMP.LE.IT)THEN
        DO 355, I1=1,M
        DO 354, I2=1,N
        IV(0,I2,I1,INCOMP+1)=1
        IV(I2,N+1,I1,INCOMP+1)=1
        IV(I2,N+1,I1,INCOMP)=1
354     CONTINUE
355     CONTINUE
      ENDIF
      OPEN(89,FILE='DELTA.FLE')
    DO 365, I=1, IT
    DO 365, I1=1, M
    DO 365, I2=1, N
```

```
          DELTA(I,I1,I2) = 0
          IF(I.EQ.INCOMP)THEN
           APROD = 0.0
           DO 360, J1=1,M
           DO 360, J2=1,IQK(I2,J1)
          APROD = APROD + IPROD(J2,I2,I1,I)
360       CONTINUE
          IF(APROD.GT.0.0)DELTA(I,I1,I2) = 1
          ENDIF
         WRITE(89,5511)  DELTA(I,I1,I2)
5511      FORMAT(5X,I1)
365       CONTINUE
          CLOSE(89)
          WRITE(*,8989)
8989      FORMAT(5X,'LEAVING FLAGS')
          RETURN
C        LAST CARD OF FLAGS
          END
C
        SUBROUTINE IP(IV,IPROD,ININV,Y,CLOST,
     X  FACTOR,IWI,LOLIM,IUPLIM,
     X  AMTCST,INCOMP)
        DIMENSION IQK(15,7),ITRAC(15,4),
     X  OBJ(15,15,7), TRANS(15,15,7), ANIT(7,15),OBJX(200,15,7),
     X  IDOT(4,200,7,15),CAP(15,7),IDMD(14,15,4),
     X  IV(0:15,16,7,0:14),
     X  MW(7),ININV(15,4),IWI(15,4)
          INTEGER FACTOR,INCOMP,STATUS
          REAL IPROD(200,15,7,14)
        COMMON ITRAK(15,4),IIW(15)
        OPEN(4, FILE='(C)PRINTER')
        OPEN(44,FILE='KRUS.MPS')
          WRITE(*,8989)
8989      FORMAT(5X,'ENTERING IP.FOR')
          TOL = 0.02
        CALL READIT(DAYL,IT,N,K,IIW,IQK,
     1  OBJ,TRANS,ANIT,OBJX,IDOT,CAP,IDMD,MW)
          NMAX = 6
        IC0 = 0
        IC1 = 0
        IC2 = 0
        IC3 = 0
        IR0 = 0
        IR1 = 0
        IR2 = 0
        IR3 = 0
        CALL WTRK(1, IR3, IR2, IR1, IR0)
          CALL ROWS(IIW,N,K,IT,
     X    IR3,IR2,IR1,IR0)
        CALL INIT(IT,N,K,IIW,ITRAC,IQK)
        IC0 = 0
        IC1 = 0
        IC2 = 0
        IC3 = 0
        OPEN(45,FILE='TEMP.PRN')
          CALL EXQKNT(IR3,IR2,IR1,IR0,
     XIC3,IC2,IC1,IC0,IQK,IT,N,K,ITRAC,
     XITRAK,IDOT,OBJX,IIW,CAP )
          CALL RCOEFF(IT,N,IIW,IR3,IR2,IR1,IR0,
     XIC3,IC2,IC1,IC0 )
          CALL INVEN(IR3,IR2,IR1,IR0,IC3,IC2,IC1,IC0,
     X    ITRAC,ITRAK,IIW,IT,N,
```

```
     X     IWI,AMTCST)
           ANUM = 1.0
           CALL RTRC(5,IC3,IC2,IC1,IC0,ITRAC)
           CALL RTRC(6,IR3,IR2,IR1,IR0,ITRAK)
           DO 227, I=1, K
           DO 227, I1=1, N
           CALL INCR(IC3, IC2, IC1, IC0)
           CALL INCR(IR3, IR2, IR1, IR0)
           WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0, ANUM
  35       FORMAT('     ',4(I1),'      ',4(I1),'      ',F10.1)
  227      CONTINUE
           CALL RTRC(1,IR3,IR2,IR1,IR0,ITRAK)
           CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
           DO 228, I=1,IT
           DO 228, I1=1,K
           DO 228, I2=0,N
           DO 228, I3=1, N+1
           IF(I2.NE.I3)THEN
           IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 228
           CALL INCR(IC3,IC2,IC1,IC0)
           IF(I2 .EQ. 0) GOTO 228
           IF(I3 .EQ. N+1) GOTO 228
           IF(OBJ(I2,I3,I1) .EQ. 0.0) GOTO 228
           WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,
     1     OBJ(I2,I3,I1)
           ENDIF
 228       CONTINUE
           CALL RTRC(2,IR3,IR2,IR1,IR0,ITRAK)
           DO 229, I5=1, IT
           DO 229, I6=1, K
           CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
           CALL INCR(IR3,IR2,IR1,IR0)
           DO 229, I=1,IT
           DO 229, I1=1,K
           DO 229, I2=0,N
           DO 229, I3=1, N+1
           IF(I2.NE.I3)THEN
           IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 229
           CALL INCR(IC3,IC2,IC1,IC0)
           IF(I2.EQ.0) GOTO 229
           IF(I3.EQ.N+1) GOTO 229
           IF(TRANS(I2,I3,I1) .EQ. 0.0) GOTO 229
           IF((I5 .EQ. I) .AND. (I6.EQ. I1))THEN
           WRITE(45,35)
     1     IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,TRANS(I2,I3,I1)
           ENDIF
           ENDIF
 229       CONTINUE
           CALL RTRC(4,IR3,IR2,IR1,IR0,ITRAK)
           DO 230, J=1, IT
           DO 230, J1=1, K
           DO 230, J2=1, N
           CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
           CALL INCR(IR3,IR2,IR1,IR0)
           DO 230, I=1,IT
           DO 230, I1=1,K
           DO 230, I2=0,N
           DO 230, I3=1, N+1
           IF(I2.NE.I3)THEN
           IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 230
           CALL INCR(IC3,IC2,IC1,IC0)
           IF((J.EQ.I).AND.(J1.EQ.I1).AND.(I3.EQ.J2))THEN
```

```
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,Y
      ENDIF
      ENDIF
230   CONTINUE
      CALL RTRC(5,IR3,IR2,IR1,IR0,ITRAK)
      ANUM = 1.0
      DO 231, J=1, IT
      DO 231, J1=1, K
      CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
      CALL INCR(IR3,IR2,IR1,IR0)
      DO 231, I=1,IT
      DO 231, I1=1,K
      DO 231, I2=0,N
      DO 231, I3=1, N+1
      IF(I2.NE.I3)THEN
      IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 231
      CALL INCR(IC3,IC2,IC1,IC0)
      IF(I3.NE.N+1)GOTO 231
      IF((J.EQ.I).AND.(J1.EQ.I1).AND.(I3.EQ.N+1))THEN
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
      ENDIF
      ENDIF
231   CONTINUE
      CALL RTRC(6,IR3,IR2,IR1,IR0,ITRAK)
      DO 232, J=1, IT
      DO 232, J1=1, K
      DO 232, J2=1, N
      CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
      CALL INCR(IR3,IR2,IR1,IR0)
      DO 232, I=1,IT
      DO 232, I1=1,K
      DO 232, I2=0,N
      DO 232, I3=1, N+1
      IF(I2.NE.I3)THEN
      IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 232
      CALL INCR(IC3,IC2,IC1,IC0)
      IF((J.EQ.I).AND.(J1.EQ.I1).AND.(J2.EQ.I3)
     1 .AND.(I2.EQ.0))THEN
      ANUM = -1.0
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
      ENDIF
      IF(((J-1).EQ.I).AND.(J1.EQ.I1).AND.
     1 (J2.EQ.I2).AND.(I3.EQ.(N+1)))THEN
      ANUM = 1.0
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
      ENDIF
      ENDIF
232   CONTINUE
      CALL RTRC(7,IR3,IR2,IR1,IR0,ITRAK)
      DO 233, J=1, IT
      DO 233, J1=1, K
      DO 233, J2=1, N
      CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
      CALL INCR(IR3,IR2,IR1,IR0)
      DO 233, I=1,IT
      DO 233, I1=1,K
      DO 233, I2=0,N
      DO 233, I3=1, N+1
```

```
      IF(I2.EQ.I3)GOTO 233
      IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 233
      CALL INCR(IC3,IC2,IC1,IC0)
      IF((J.EQ.I).AND.(J1.EQ.I1).AND.(I3.EQ.J2))THEN
      ANUM = -1.0
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
      ENDIF
      IF((J.EQ.I).AND.(J1.EQ.I1).AND.(I2.EQ.J2))THEN
      ANUM = 1.0
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
      ENDIF
 233  CONTINUE
      CALL RTRC(8,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
      DO 234, J=1, IT
      DO 234, J1=1, K
      DO 234, J2=0, N
      DO 234, J3=1, N+1
      IF(J2.EQ.J3)GOTO 234
      IF((J2.EQ.0).AND.(J3.EQ.N+1))GOTO 234
      CALL INCR(IR3,IR2,IR1,IR0)
      CALL INCR(IC3,IC2,IC1,IC0)
        IFLG = 0
        ANUM = N+1
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
 234  CONTINUE
      ANUM = CLOST
      CALL RTRC(1,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(4,IC3,IC2,IC1,IC0,ITRAC)
      DO 237, I=1, IT
      DO 237, I1=1, N
      DO 237, I2=1, IIW(I1)
      CALL INCR(IC3,IC2,IC1,IC0)
        ANUM = CLOST*(1+0.2*I)*(1+0.5*I1)
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
  237 CONTINUE
      CALL RTRC(7,IC3,IC2,IC1,IC0,ITRAC)
      CALL RTRC(2,IR3,IR2,IR1,IR0,ITRAK)
      ANUM =  1.0
      DO 265, I=1,IT
      DO 265, I1=1,K
      CALL INCR(IR3, IR2, IR1, IR0)
      CALL INCR(IC3, IC2, IC1, IC0)
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
  265 CONTINUE
      CALL RTRC(7,IC3,IC2,IC1,IC0,ITRAC)
      CALL RTRC(2,IR3,IR2,IR1,IR0,ITRAK)
      ANUM = -1.0
      DO 266, I=1,K
      CALL INCR(IR3, IR2, IR1, IR0)
  266 CONTINUE
      DO 267, I=1,IT-1
      DO 267, I1=1,K
      CALL INCR(IR3, IR2, IR1, IR0)
      CALL INCR(IC3, IC2, IC1, IC0)
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
  267 CONTINUE
      CALL RTRC(7,IC3,IC2,IC1,IC0,ITRAC)
```

```
      CALL RTRC(1,IR3,IR2,IR1,IR0,ITRAK)
      DO 270, I=1,IT
      DO 270, I1=1,K
        ANUM = I/2.0
      CALL INCR(IC3, IC2, IC1, IC0)
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
 270  CONTINUE
       CALL ITB1(IR3,IR2,IR1,IR0,ITRAK,
     1 IC3,IC2,IC1,IC0,ITRAC,IT,K,N)
      ANUM = 1.0
      CALL RTRC(3,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(4,IC3,IC2,IC1,IC0,ITRAC)
      DO 278, I=1, IT
      DO 278, I1=1, N
      DO 278, I2=1, IIW(I1)
      CALL INCR(IR3,IR2,IR1,IR0)
      CALL INCR(IC3,IC2,IC1,IC0)
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
 278  CONTINUE
      CALL RTRC(9,IR3,IR2,IR1,IR0,ITRAK)
      ANUM = 1
      DO 285, J=1, IT
      DO 285, J1=1, K
      DO 285, J2=1, N
      CALL INCR(IR3,IR2,IR1,IR0)
      CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
      DO 285, I=1, IT
      DO 285,I1=1,  K
      DO 285,I2=0, N
      DO 285,I3=1, N+1
      IF(I2.EQ.I3)GOTO 285
      IF((I2.EQ.0).AND.(I3.EQ.N+1))GOTO 285
      CALL INCR(IC3,IC2,IC1,IC0)
        IF(I2.EQ.0)GOTO 285
      IF((J.EQ.I).AND.(J1.EQ.I1).AND.(J2.EQ.I3))THEN
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
        ENDIF
 285  CONTINUE
      CALL RTRC(9,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(9,IC3,IC2,IC1,IC0,ITRAC)
        ANUM = -1.0
      DO 295, J=1, IT
      DO 295, J1=1, K
      DO 295, J2=1, N
      CALL INCR(IR3,IR2,IR1,IR0)
      CALL INCR(IC3,IC2,IC1,IC0)
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
 295  CONTINUE
      CALL RTRC(10,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(9,IC3,IC2,IC1,IC0,ITRAC)
        ANUM = -FACTOR/1.0
      DO 300, J=1, IT
      DO 300, J1=1, K
      DO 300, J2=1, N
      CALL INCR(IC3,IC2,IC1,IC0)
      CALL INCR(IR3,IR2,IR1,IR0)
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
 300  CONTINUE
        CALL RTRC(10,IR3,IR2,IR1,IR0,ITRAK)
```

```
      ANUM = 1.0
      DO 305, J=1, IT
      DO 305, J1=1, K
      DO 305, J2=1, N
      CALL INCR(IR3,IR2,IR1,IR0)
      CALL RTRC(1,IC3,IC2,IC1,IC0,ITRAC)
      DO 305, I=1, IT
      DO 305, I1=1, N
      DO 305, I2=1, K
      DO 305, I3=1, IQK(I1,I2)
      CALL INCR(IC3,IC2,IC1,IC0)
      IF((J.EQ.I).AND.(J1.EQ.I2).AND.(J2.EQ.I1))THEN
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
        ENDIF
305     CONTINUE
      CALL RTRC(10,IR3,IR2,IR1,IR0,ITRAK)
      ANUM = FACTOR/1.0
      DO 310, J=1, IT
      DO 310, J1=1, K
      DO 310, J2=1, N
      CALL INCR(IR3,IR2,IR1,IR0)
      CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
      DO 310, I=1, IT
      DO 310,I1=1,  K
      DO 310,I2=0, N
      DO 310,I3=1, N+1
      IF(I2.EQ.I3)GOTO 310
      IF((I2.EQ.0).AND.(I3.EQ.N+1))GOTO 310
      CALL INCR(IC3,IC2,IC1,IC0)
      IF((J.EQ.I).AND.(J1.EQ.I1).AND.(J2.EQ.I2)
     X     .AND.(I3.EQ.N+1))THEN
      WRITE(45,35)
     1 IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
        ENDIF
 310  CONTINUE
      ANUM = -1000000.0
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0, ANUM
      CLOSE(45)
      CALL SORT()
        OPEN(88,FILE='ICON.FLE')
        READ(88,5511)ICON
5511    FORMAT(5X,I1)
        CLOSE(88)
      WRITE(44,500)
  500 FORMAT('RHS')
      CALL RTRC(2,IR3,IR2,IR1,IR0,ITRAK)
      DO 520, I=1, IT
      DO 520, I1=1, K
      CALL INCR(IR3, IR2, IR1, IR0)
        ANUM = DAYL
      WRITE(44,600) IR3,IR2,IR1,IR0,ANUM
 600  FORMAT('    RHS       R',4(I1),'     ',F10.1)
 520  CONTINUE
      DO 530, I=1, IT
      DO 530, I1=1, N
      DO 530, I2=1, IIW(I1)
      ANUM = IDMD(I,I1,I2)
      CALL INCR(IR3, IR2, IR1, IR0)
      IF(ANUM .EQ. 0.0)GOTO 530
      WRITE(44,600) IR3,IR2,IR1,IR0,ANUM
 530  CONTINUE
```

```
      CALL RTRC(5,IR3,IR2,IR1,IR0,ITRAK)
      ANUM = 1.0
      DO 540, I=1, IT
      DO 540, I1=1, K
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,600) IR3,IR2,IR1,IR0,ANUM
 540  CONTINUE
      CALL RTRC(8,IR3,IR2,IR1,IR0,ITRAK)
      DO 550, I=1, IT
      DO 550, I1=1, K
      DO 550, I2=0, N
      DO 550, I3=1, N+1
      IF(I2.EQ.I3)GOTO 550
      IF((I2.EQ.0).AND.(I3.EQ.N+1))GOTO 550
      CALL INCR(IR3, IR2, IR1, IR0)
        IFLG = 1
        ANUM = N
        IF(ANUM.NE.0.0)
     X  WRITE(44,600) IR3,IR2,IR1,IR0,ANUM
 550  CONTINUE
      WRITE(44,601)
 601  FORMAT('BOUNDS')
        WRITE(*,3553)INCOMP
3553    FORMAT(5X,'INCOMP = ',I3)
      CALL RTRC(1,IC3,IC2,IC1,IC0,ITRAC)
      DO 560, I=1, IT
      DO 560, I1=1, N
      DO 560, I2=1, K
        IF(I.GE.INCOMP)STATUS = 0
        IF(I.LT.INCOMP-1)STATUS = 2
        IF(I.EQ.INCOMP-1)THEN
        DO 559, J=0,N
        IF((IV(J,I1,I2,I).EQ.2).OR.
     X  (IV(J,I1,I2,I).EQ.3))STATUS = 1
559     CONTINUE
        ENDIF
      DO 560, I3=1, IQK(I1,I2)
      CALL INCR(IC3, IC2, IC1, IC0)
        LOLIM = 2
        IUPLIM = 3
        IF(STATUS.EQ.1)THEN
        LOWER = INT(IPROD(I3,I1,I2,I)) - LOLIM
        IF(LOWER.LT.0)LOWER = 0
        IUPPER = INT(IPROD(I3,I1,I2,I))+IUPLIM
        WRITE(44,704) IC3,IC2,IC1,IC0,IUPPER
        WRITE(44,703) IC3,IC2,IC1,IC0,LOWER
        ENDIF
        IF(STATUS.EQ.2)THEN
        ERROR  = IPROD(I3,I1,I2,I)-INT(IPROD(I3,I1,I2,I))
        IF(ERROR.GT.TOL)THEN
        LOWER = INT(IPROD(I3,I1,I2,I)) - LOLIM
        IF(LOWER.LT.0)LOWER = 0
        IUPPER = INT(IPROD(I3,I1,I2,I))+IUPLIM
        WRITE(44,704) IC3,IC2,IC1,IC0,IUPPER
        WRITE(44,703) IC3,IC2,IC1,IC0,LOWER
        ELSE
        ANUM = NINT(IPROD(I3,I1,I2,I))
        WRITE(44,762) IC3,IC2,IC1,IC0,ANUM
        ENDIF
        ENDIF
 560    CONTINUE
      CALL RTRC(3,IC3,IC2,IC1,IC0,ITRAC)
```

```
      DO 622, I=1, N
      DO 622, J=1, IIW(I)
      CALL INCR(IC3,IC2,IC1,IC0)
      ANUM = ININV(I,J)/1.0
      WRITE(44,762) IC3,IC2,IC1,IC0,ANUM
622   CONTINUE
      CALL RTRC(5,IC3,IC2,IC1,IC0,ITRAC)
      DO 625, I=1, K
      DO 625, J=1, N
      CALL INCR(IC3,IC2,IC1,IC0)
        ANUM = ANIT(I,J)
      WRITE(44,762) IC3,IC2,IC1,IC0,ANUM
625   CONTINUE
      CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
      DO 635, I=1, IT
      DO 635, I1=1, K
      DO 635, I2=0, N
      DO 635, I3=1, N+1
      IF(I2.EQ.I3)GOTO 635
      IF((I2.EQ.0).AND.(I3.EQ.(N+1)))GOTO 635
      CALL INCR(IC3,IC2,IC1,IC0)
      IF(IV(I2,I3,I1,I).GT.0)THEN
      IF(IV(I2,I3,I1,I).EQ.1)
     X        WRITE(44,700)IC3,IC2,IC1,IC0
      IF(IV(I2,I3,I1,I).EQ.2)
     X        WRITE(44,705)IC3,IC2,IC1,IC0
      IF(IV(I2,I3,I1,I).EQ.3)
     X        WRITE(44,702)IC3,IC2,IC1,IC0
        ENDIF
  635 CONTINUE
        OPEN(88,FILE='DELTA.FLE')
      CALL RTRC(9,IC3,IC2,IC1,IC0,ITRAC)
      DO 640, J=1, IT
      DO 640, J1=1, K
      DO 640, J2=1, N
        READ(88,5511)IDELTA
      CALL INCR(IC3,IC2,IC1,IC0)
         IF(IDELTA.EQ.1) WRITE(44,700)IC3,IC2,IC1,IC0
         IF(IDELTA.EQ.2) WRITE(44,705)IC3,IC2,IC1,IC0
         IF(IDELTA.EQ.3) WRITE(44,702)IC3,IC2,IC1,IC0
640   CONTINUE
        CLOSE(88)
700   FORMAT(' BV BOUNDS    X',4(I1))
701   FORMAT(' UP BOUNDS    X',4(I1),12X,'1.0')
702   FORMAT(' FX BOUNDS    X',4(I1),12X,'0.0')
705   FORMAT(' FX BOUNDS    X',4(I1),12X,'1.0')
703   FORMAT(' LI BOUNDS    X',4(I1),12X,I3)
704   FORMAT(' UI BOUNDS    X',4(I1),12X,I3)
762   FORMAT(' FX BOUNDS    X',4(I1),10X,F5.1)
763   FORMAT(' LO BOUNDS    X',4(I1),10X,F5.1)
      WRITE(44,75)
75    FORMAT('ENDATA')
      CALL RTRC(10,IC3,IC2,IC1,IC0,ITRAC)
      ICC = IC3*1000+IC2*100+IC1*10+IC0
      CALL RTRC(13,IR3,IR2,IR1,IR0,ITRAK)
      IRC = 1000*IR3+100*IR2+10*IR1+IR0
      CALL SOS(IT,K,N,ITRAC)
      CALL ORD(IT,K,N,ITRAC,IIW,IQK)
      CLOSE(44)
      CLOSE(4)
        WRITE(*,9090)
9090    FORMAT(5X,'LEAVING IP.FOR')
```

```
      RETURN
C     LAST CARD OF IP.FOR
      END
C
        SUBROUTINE ROWS(IIW,N,K,IT,
     X  IR3,IR2,IR1,IR0)
        DIMENSION IIW(15)
      WRITE(44,5)
  5   FORMAT('NAME             IP   MODEL',/,'ROWS')
      WRITE(44,7)
  7  FORMAT('  N R0000')
      DO 10, J4=1, IT
      DO 10, I1=1, N
      DO 10, I2=1,IIW(I1)
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,21) IR3,IR2,IR1,IR0
  20  FORMAT('  L R',4(I1))
  21  FORMAT('  E R',4(I1))
  22  FORMAT('  G R',4(I1))
  10  CONTINUE
      CALL WTRK(2, IR3, IR2, IR1, IR0)
      DO 11, I=1,IT
      DO 11, I1=1,K
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,21) IR3,IR2,IR1,IR0
  11  CONTINUE
      CALL WTRK(3, IR3, IR2, IR1, IR0)
      DO 12, I=1, IT
      DO 12, I1=1, N
      DO 12, I2=1, IIW(I1)
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,21) IR3,IR2,IR1,IR0
  12  CONTINUE
      CALL WTRK(4, IR3, IR2, IR1, IR0)
      DO 13, I=1, IT
      DO 13, I1=1, K
      DO 13, I2=1, N
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,22) IR3,IR2,IR1,IR0
  13  CONTINUE
      CALL WTRK(5, IR3, IR2, IR1, IR0)
      DO 14, I=1, IT
      DO 14, I1=1, K
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,21) IR3,IR2,IR1,IR0
  14  CONTINUE
      CALL WTRK(6, IR3, IR2, IR1, IR0)
      DO 16, I=1, IT
      DO 16, I1=1, K
      DO 16, I2=1, N
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,21) IR3,IR2,IR1,IR0
  16  CONTINUE
      CALL WTRK(7, IR3, IR2, IR1, IR0)
      DO 17, I=1, IT
      DO 17, I1=1, K
      DO 17, I2=1, N
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,21) IR3,IR2,IR1,IR0
  17  CONTINUE
      CALL WTRK(8, IR3, IR2, IR1, IR0)
      DO 18, I=1, IT
```

```
      DO 18, I1=1, K
      DO 18, I2=0, N
      DO 18, I3=1, N+1
      IF (I2 .NE. I3) THEN
      IF ((I2 .EQ. 0) .AND. (I3 .EQ. N+1)) GOTO 18
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,20) IR3,IR2,IR1,IR0
      ENDIF
  18  CONTINUE
      CALL WTRK(9, IR3, IR2, IR1, IR0)
      DO 40, I=1, IT
      DO 40, I1=1, K
      DO 40, I2=1, N
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(44,21) IR3,IR2,IR1,IR0
  40  CONTINUE
      CALL WTRK(10, IR3, IR2, IR1, IR0)
        DO 45, I=1,IT
        DO 45,I1=1,K
        DO 45,I2=1,N
        CALL INCR(IR3, IR2, IR1, IR0)
        WRITE(44,22) IR3,IR2,IR1,IR0
45      CONTINUE
        CALL WTRK(11, IR3, IR2, IR1, IR0)
        RETURN
C         LAST CARD OF ROWS
        END
C
     SUBROUTINE EXQKNT(IR3,IR2,IR1,IR0,
    XIC3,IC2,IC1,IC0,IQK,IT,N,K,ITRAC,
    XITRAK,IDOT,OBJX,IIW,CAP )
     DIMENSION IQK(15,7),ITRAC(15,4),ITRAK(15,4),
    1 OBJX(200,15,7),IDOT(4,200,7,15),IIW(15),CAP(15,7)
      WRITE(44,30)
  30  FORMAT('COLUMNS')
  35  FORMAT('    ',4(I1),'     ',4(I1),'      ',F10.1)
      CALL RTRC(1,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(1,IC3,IC2,IC1,IC0,ITRAC)
      DO 50, I=1, IT
      DO 50, I1=1, N
      DO 50, I2=1, K
      DO 50, I3=1, IQK(I1,I2)
      IF(OBJX(I3,I1,I2).NE. 0.0)THEN
        CALL WKIT1(IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,
    1   OBJX(I3,I1,I2))
        ENDIF
      IF(OBJX(I3,I1,I2).EQ. 0.0)THEN
      CALL WKIT2(IC3,IC2,IC1,IC0)
      ENDIF
  50  CONTINUE
      CALL RTRC(1,IC3,IC2,IC1,IC0,ITRAC)
      CALL INCR(IR3,IR2,IR1,IR0)
      IR3S = IR3
      IR2S = IR2
      IR1S = IR1
      IR0S = IR0
      DO 52, I=1, IT
      DO 52, I1=1, N
      DO 52, I2=1, K
      DO 52, I3=1, IQK(I1,I2)
      IC3S = IC3
      IC2S = IC2
```

```
      IC1S = IC1
      IC0S = IC0
      DO 51, J0=1,IT
      DO 51, J=1, N
      DO 51, J1=1, IIW(J)
      IF((J.EQ.I1).AND.(J0.EQ.I))THEN
      ANUM = IDOT(J1,I3,I2,J)/1.0
      IF(IDOT(J1,I3,I2,J).NE. 0.0) THEN
      CALL WKIT1(IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,
     1 ANUM)
      IC3 = IC3S
      IC2 = IC2S
      IC1 = IC1S
      IC0 = IC0S
      ENDIF
      ENDIF
      CALL INCR(IR3,IR2,IR1,IR0)
51    CONTINUE
      CALL WKIT2(IC3,IC2,IC1,IC0)
      IR3 = IR3S
      IR2 = IR2S
      IR1 = IR1S
      IR0 = IR0S
52    CONTINUE
      CALL RTRC(2,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(1,IC3,IC2,IC1,IC0,ITRAC)
      CALL INCR(IR3,IR2,IR1,IR0)
      IR3S = IR3
      IR2S = IR2
      IR1S = IR1
      IR0S = IR0
      DO 55, I=1, IT
      DO 55, I1=1, N
      DO 55, I2=1, K
      DO 55, I3=1, IQK(I1,I2)
      IC3S = IC3
      IC2S = IC2
      IC1S = IC1
      IC0S = IC0
      DO 54, J=1, IT
      DO 54, J2=1, K
      IF((J.EQ.I).AND.(J2.EQ.I2))THEN
      IF(CAP(I1,I2).NE. 0.0) THEN
      CALL WKIT1(IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,
     1 CAP(I1,I2))
      IC3 = IC3S
      IC2 = IC2S
      IC1 = IC1S
      IC0 = IC0S
      ENDIF
      ENDIF
      CALL INCR(IR3,IR2,IR1,IR0)
54    CONTINUE
      CALL WKIT2(IC3,IC2,IC1,IC0)
      IR3 = IR3S
      IR2 = IR2S
      IR1 = IR1S
      IR0 = IR0S
55    CONTINUE
      CALL RTRC(4,IR3,IR2,IR1,IR0,ITRAK)
      ANUM = -1.0
      DO 60, J=1, IT
```

```
      DO 60, J1=1, K
      DO 60, J2=1, N
      CALL INCR(IR3,IR2,IR1,IR0)
      CALL RTRC(1,IC3,IC2,IC1,IC0,ITRAC)
      DO 60, I=1, IT
      DO 60, I1=1, N
      DO 60, I2=1, K
      DO 60, I3=1, IQK(I1,I2)
      IF((J.EQ.I).AND.(J1.EQ.I2).AND.(J2.EQ.I1))THEN
      CALL WKIT1(IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,
     1 ANUM)
      ELSE
      CALL WKIT2(IC3,IC2,IC1,IC0)
      ENDIF
 60   CONTINUE
        RETURN
C         LAST CARD OF EXQKNT
        END
C
      SUBROUTINE RCOEFF(IT,N,IIW,IR3,IR2,IR1,IR0,
     XIC3,IC2,IC1,IC0 )
        DIMENSION IIW(15)
      IDMD1 = 0
      DO 201, I=1,N
      IDMD1 = IDMD1 + IIW(I)
 201  CONTINUE
      ANUM = -1.0
      IFLG = 0
      DO 220, I=1, IT
      DO 220, I1=1, N
      DO 220, I2=1, IIW(I1)
      CALL INCR(IC3, IC2, IC1, IC0)
      IHJ1 = 0
      ANUM = -1.0
      IFLG = 0
      CALL RTRK(1, IR3, IR2, IR1, IR0)
      DO 202, J =1, IT
      DO 202, J1=1, N
      DO 202, J2=1, IIW(J1)
      CALL INCR(IR3, IR2, IR1, IR0)
      IF((I1.EQ.J1).AND.(I2.EQ.J2).AND.(J.EQ.I))THEN
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0, ANUM
 35   FORMAT('    ',4(I1),'       ',4(I1),'      ',F10.1)
      IFLG = 1
      ENDIF
      IF(IFLG .EQ. 1) GOTO 203
 202  CONTINUE
 203  CONTINUE
      IHJ1 = IHJ1 + 1
      ANUM = 1.0
      IFLG = 0
      CALL RTRK(3, IR3, IR2, IR1, IR0)
      DO 204, IH1=1, (IHJ1-1)*IDMD1
      CALL INCR(IR3, IR2, IR1, IR0)
 204  CONTINUE
      DO 205, J2=1, IT
      DO 205, J3=1, N
      DO 205, J4=1, IIW(J3)
      CALL INCR(IR3, IR2, IR1, IR0)
      IF((I.EQ.J2).AND.(I1.EQ.J3).AND.(I2.EQ.J4))THEN
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0, ANUM
      IFLG = 1
```

```
      ENDIF
      IF(IFLG .EQ. 1) GOTO 206
 205  CONTINUE
 206  CONTINUE
 220  CONTINUE
        RETURN
C       LAST CARD OF RCOEFF
        END
C
        SUBROUTINE INVEN(IR3,IR2,IR1,IR0,IC3,IC2,IC1,IC0,
     X  ITRAC,ITRAK,IIW,IT,N,
     X  IWI,AMTCST)
        DIMENSION IIW(15),ITRAC(15,4),ITRAK(15,4)
        INTEGER IWI(15,4)
 35   FORMAT('    ',4(I1),'     ',4(I1),'     ',F10.1)
      CALL RTRC(1,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(3,IC3,IC2,IC1,IC0,ITRAC)
      DO 136, I=0, IT
      DO 136, I1=1, N
      DO 136, I2=1, IIW(I1)
      CALL INCR(IC3,IC2, IC1,IC0)
      IF(I.EQ.IT)THEN
        ANUM = -(IWI(I1,I2)/1.0)*(AMTCST)
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
      ENDIF
 136  CONTINUE
      ANUM = 1.0
      CALL RTRC(3,IR3,IR2,IR1,IR0,ITRAK)
      CALL RTRC(3,IC3,IC2,IC1,IC0,ITRAC)
      IR = 3
      CALL SUB2(IC3,IC2,IC1,IC0,IT,N,ANUM,IR)
      ANUM = -1.0
      CALL RTRC(3,IC3,IC2,IC1,IC0,ITRAC)
      DO 226, I=1,N
      DO 226, I1=1,IIW(I)
      CALL INCR(IC3,IC2,IC1,IC0)
 226  CONTINUE
      CALL SUB2(IC3,IC2,IC1,IC0,IT,N,ANUM,IR)
        RETURN
C
        END
C
      SUBROUTINE WKIT1(I3,I2,I1,I0,N3,N2,N1,N0,TEMP)
      CALL INCR(I3, I2, I1, I0)
      WRITE(45,35) I3,I2,I1,I0,N3,N2,N1,N0, TEMP
 35   FORMAT('    ',4(I1),'     ',4(I1),'     ',F10.1)
      RETURN
      END
C
      SUBROUTINE  WKIT2(IC3,IC2,IC1,IC0)
      CALL INCR(IC3,IC2,IC1,IC0)
      RETURN
      END
C
      SUBROUTINE SORT()
      DIMENSION IR1(50000),IR2(50000),RA(50000)
      OPEN(45,FILE='TEMP.PRN')
      N = 0
      DO 10, I=1, 450000
      READ(45,27) IA,IB,C
27    FORMAT(3X,I6,3X,I6,3X,F11.1)
      IF(C.GT.-999999.0)THEN
```

```
            N = N + 1
            IF(N.EQ.50000)PAUSE'CONV....TOO MANY COEFFS.'
            IR1(I) = IA
            IR2(I) = IB
             RA(I) = C
          ELSE
            GOTO 11
          ENDIF
   10     CONTINUE
   11     CONTINUE
          CLOSE(45)
          CALL SUSS(N,IR1,IR2,RA)
              DO 130,I1=1,N
            CALL CONV(IR1(I1),IC3,IC2,IC1,IC0)
            CALL CONV(IR2(I1),IB3,IB2,IB1,IB0)
            WRITE(44,36) IC3,IC2,IC1,IC0,IB3,IB2,IB1,IB0,RA(I1)
  130         CONTINUE
   36         FORMAT('    X',4(I1),'      R',4(I1),'      ',F10.1)
            RETURN
            END
C
            SUBROUTINE CONV(IR,IC3,IC2,IC1,IC0)
               IF(IR.GT.9999)THEN
                 PAUSE'CONV....NUMBER >=10,000'
                 ENDIF
                 IC3=   IR/1000
                 IC2=  (IR-IC3*1000)/100
                 IC1= (IR-IC3*1000-IC2*100)/10
                 IC0= (IR-IC3*1000-IC2*100-IC1*10)
                 IRT = IC3*1000+IC2*100+IC1*10+IC0
                 IF(IR.NE.IRT)PAUSE'CONV....IR <> IRT'
                 RETURN
                 END
C
          SUBROUTINE SUSS(N,IR1,IR2,RA)
          DIMENSION IR1(50000),IR2(50000),RA(50000)
          L = N/2+1
          IR = N
  100     CONTINUE
          IF(L.GT.1)THEN
            L=L-1
            IRR1=IR1(L)
            IRR2=IR2(L)
            RRA=RA(L)
          ELSE
            IRR1=IR1(IR)
            IRR2=IR2(IR)
            RRA=RA(IR)
            IR1(IR)=IR1(1)
            IR2(IR)=IR2(1)
            RA(IR)= RA(1)
            IR=IR-1
            IF(IR.EQ.1)THEN
              IR1(1)=IRR1
              IR2(1)=IRR2
              RA(1)= RRA
              RETURN
            ENDIF
          ENDIF
          I=L
          J=L+L
  200     IF(J.LE.IR)THEN
```

```
          IF(J.LT.IR)THEN
            IF(IR1(J).LT.IR1(J+1))J=J+1
          ENDIF
          IF(IRR1.LT.IR1(J))THEN
            IR1(I)=IR1(J)
            IR2(I)=IR2(J)
            RA(I)= RA(J)
            I=J
            J=J+J
          ELSE
            J=IR+1
          ENDIF
        GOTO 200
        ENDIF
        IR1(I)=IRR1
        IR2(I)=IRR2
        RA(I)= RRA
        GOTO 100
         RETURN
        END
C
        SUBROUTINE RTRC(IR,IC3,IC2,IC1,IC0,ITRAC)
        DIMENSION ITRAC(15,4)
        IC3 = ITRAC(IR,1)
        IC2 = ITRAC(IR,2)
        IC1 = ITRAC(IR,3)
        IC0 = ITRAC(IR,4)
        RETURN
        END
C
        SUBROUTINE PNT(ITRAC)
        DIMENSION ITRAC(15,4)
        DO 10, I=1,12
        WRITE(*,*) (ITRAC(I,J),J=1,4)
   10   CONTINUE
        RETURN
C         LAST CARD OF PNT
        END
C
        SUBROUTINE INIT(IT,N,K,IIW,ITRAC,IQK)
        DIMENSION IIW(15), ITRAC(15,4),IQK(15,7),IC(4)

        DO 10, I=1,4
        IC(I) = 0
   10   CONTINUE
        IR = 1
        CALL QUICK(IR,IC,ITRAC)
        DO 20, I=1, IT
        DO 20, I1=1, N
        DO 20, I2=1, K
        DO 20, I3=1, IQK(I1,I2)
        CALL INCR(IC(4),IC(3),IC(2),IC(1))
20        CONTINUE
        IR=2
        CALL QUICK(IR,IC,ITRAC)
        DO 40, I=1,IT
        DO 40, I1=1, N
        DO 40, I2=1, IIW(I1)
        CALL INCR(IC(4),IC(3),IC(2),IC(1))
   40   CONTINUE
        IR = 3
        CALL QUICK(IR,IC,ITRAC)
```

```
      DO 60, I=0, IT
      DO 60, I1=1,N
      DO 60, I2=1, IIW(I1)
      CALL INCR(IC(4),IC(3),IC(2),IC(1))
 60   CONTINUE
      IR = 4
      CALL QUICK(IR,IC,ITRAC)
      DO 80, I=1, IT
      DO 80, I1=1, N
      DO 80, I2=1, IIW(I1)
        CALL INCR(IC(4),IC(3),IC(2),IC(1))
 80   CONTINUE
      IR = 5
      CALL QUICK(IR,IC,ITRAC)
      DO 90, I1=1,K
      DO 90, I2=1,N
      CALL INCR(IC(4),IC(3),IC(2),IC(1))
 90   CONTINUE
      IR = 6
      CALL QUICK(IR,IC,ITRAC)
      DO 100, I=1,IT
      DO 100, I1=1,K
      DO 100, I2=0,N
      DO 100, I3=1, N+1
      IF(I2.NE.I3)THEN
      IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 100
      CALL INCR(IC(4),IC(3),IC(2),IC(1))
      ENDIF
100   CONTINUE
      IR= 7
      CALL QUICK(IR,IC,ITRAC)
      DO 110, I=1, IT
      DO 110, I1=1, K
      CALL INCR(IC(4),IC(3),IC(2),IC(1))
110   CONTINUE
      IR = 8
      CALL QUICK(IR,IC,ITRAC)
      DO 120, I=1, IT
      DO 120, I1=1, K
      DO 120, I2=0, N+1
      CALL INCR(IC(4),IC(3),IC(2),IC(1))
 120  CONTINUE
      IR = 9
      CALL QUICK(IR,IC,ITRAC)
      DO 150, I=1, IT
      DO 150, I1=1, K
      DO 150, I2=1, N
      CALL INCR(IC(4),IC(3),IC(2),IC(1))
 150  CONTINUE
      IR = 10
      CALL QUICK(IR,IC,ITRAC)
        RETURN
C       LAST CARD OF INIT
      END
C
      SUBROUTINE QUICK(IR,IC,ITRAC)
      DIMENSION ITRAC(15,4),IC(4)
      ITRAC(IR,1) = IC(4)
      ITRAC(IR,2) = IC(3)
      ITRAC(IR,3) = IC(2)
      ITRAC(IR,4) = IC(1)
 10   CONTINUE
```

```
      RETURN
      END
C
      SUBROUTINE SUB2(IC3,IC2,IC1,IC0,IT,N,ANUM,IR)
      COMMON ITRAK(15,4),IIW(15)
      CALL RTRK(IR, IR3, IR2, IR1, IR0)
      DO 225, I=1, IT
      DO 225, I1=1, N
      DO 225, I2=1, IIW(I1)
      CALL INCR(IC3, IC2, IC1, IC0)
      CALL INCR(IR3, IR2, IR1, IR0)
      WRITE(45,35) IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0, ANUM
 35   FORMAT('    ',4(I1),'     ',4(I1),'     ',F10.1)
 225  CONTINUE
      RETURN
      END
C
      SUBROUTINE RTRK(I, IR3, IR2, IR1, IR0)
      COMMON ITRAK(15,4),IIW(15)
      IR3 = ITRAK(I,1)
      IR2 = ITRAK(I,2)
      IR1 = ITRAK(I,3)
      IR0 = ITRAK(I,4)
      RETURN
      END
      SUBROUTINE WTRK(I, IR3, IR2, IR1, IR0)
      COMMON ITRAK(15,4),IIW(15)
      ITRAK(I,1) = IR3
      ITRAK(I,2) = IR2
      ITRAK(I,3) = IR1
      ITRAK(I,4) = IR0
      RETURN
      END
      SUBROUTINE INCR(IR3, IR2, IR1, IR0)
      IR0 = IR0 + 1
      IF(IR0 .EQ. 10) THEN
        IR0 = 0
        IR1 = IR1 + 1
        IF(IR1 .EQ. 10) THEN
          IR1 = 0
          IR2 = IR2 + 1
          IF(IR2 .EQ. 10) THEN
            IR2 = 0
            IR3 = IR3 + 1
            ENDIF
            ENDIF
            ENDIF
      RETURN
      END
C
      SUBROUTINE ITB1(IR3,IR2,IR1,IR0,ITRAK,
     1 IC3,IC2,IC1,IC0,ITRAC,IT,K,N)
      DIMENSION ITRAK(15,4),ITRAC(15,4)
 35   FORMAT('    ',4(I1),'     ',4(I1),'     ',F10.1)
      CALL RTRC(8,IR3,IR2,IR1,IR0,ITRAK)
      DO 276, J=1, IT
      DO 276, J1=1, K
      DO 276, J2=0, N
      DO 276, J3=1, N+1
      IF((J2.EQ.0).AND.(J3.EQ.N+1)) GOTO 276
      IF(J2.EQ.J3) GOTO 276
      CALL RTRC(8,IC3,IC2,IC1,IC0,ITRAC)
```

```
        CALL INCR(IR3,IR2,IR1,IR0)
        DO 275, I=1, IT
        DO 275, I1=1, K
        DO 275, I2=0, N+1
        CALL INCR(IC3,IC2,IC1,IC0)
        IF((J.EQ.I).AND.(J1.EQ.I1).AND.(J2.EQ.I2))THEN
        ANUM = 1.0
        WRITE(45,35)
     1  IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
        ENDIF
        IF((J.EQ.I).AND.(J1.EQ.I1).AND.(J3.EQ.I2))THEN
        ANUM = -1.0
        WRITE(45,35)
     1  IC3,IC2,IC1,IC0,IR3,IR2,IR1,IR0,ANUM
        ENDIF
 275    CONTINUE
 276    CONTINUE
        RETURN
        END
C
C$INCLUDE SOS.FOR
C$INCLUDE ORD.FOR
C
        SUBROUTINE ORD(IT,M,N,ITRAC,IIW,IQK)
        INTEGER ITRAC(15,4),FLAG(15,15,14,7)
          DIMENSION IIW(15),IQK(15,7)
          DO 5,I=1,IT
          DO 5,I1=1,M
          DO 5,I2=1,N
          DO 5,I3=1,N
          FLAG(I2,I3,I,I1) = 0
5         CONTINUE
        OPEN(78,FILE='KRUS.ORD')
          WRITE(*,19)
19        FORMAT(5X,'ENTERING ORD.FOR')
        WRITE(78,20)
20      FORMAT('NAME')
21     FORMAT(1X,'UP X',4I1,17X,I5)
          CALL RTRC(10,IC3,IC2,IC1,IC0,ITRAC)
          INUM = IC3*1000+IC2*100+IC1*10+IC0 + 1
        CALL RTRC(9,IC3,IC2,IC1,IC0,ITRAC)
        DO 50, I=1, IT
        DO 50, I1=1, M
        DO 50, I2=1, N
          INUM = INUM - 1
        CALL INCR(IC3,IC2,IC1,IC0)
          WRITE(78,21)IC3,IC2,IC1,IC0,INUM
50      CONTINUE
        CALL RTRC(1,IC3,IC2,IC1,IC0,ITRAC)
        DO 60, I=1, IT
        DO 60, I1=1, N
        DO 60, I2=1, M
        DO 60, I3=1, IQK(I1,I2)
          INUM = INUM - 1
          CALL INCR(IC3, IC2, IC1, IC0)
          WRITE(78,21)IC3,IC2,IC1,IC0,INUM
60      CONTINUE
          CALL RTRC(4,IC3,IC2,IC1,IC0,ITRAC)
          DO 100, I=1,IT
          INUM = INUM - 1
          DO 100,I1=1,N
          DO 100,I2=1,IIW(I1)
```

```
          CALL INCR(IC3, IC2, IC1, IC0)
          WRITE(78,21)IC3,IC2,IC1,IC0,INUM
100       CONTINUE
          WRITE(78,835)
835        FORMAT('ENDATA')
          CLOSE(78)
          WRITE(*,18)
18        FORMAT(5X,'LEAVING ORD.FOR')
C
C         LAST CARD OF ORD.FOR
         RETURN
        END
C
        SUBROUTINE SOS(IT,M,N,ITRAC)
        INTEGER ITRAC(15,4),FLAG(15,15,14,7)
          DO 5,I=1,IT
          DO 5,I1=1,M
          DO 5,I2=1,N
          DO 5,I3=1,N
          FLAG(I2,I3,I,I1) = 0
5         CONTINUE
        OPEN(78,FILE='KRUS.SOS')
          WRITE(*,19)
19        FORMAT(5X,'ENTERING SOS.FOR')
        WRITE(78,20)
20      FORMAT('NAME')
21      FORMAT(1X,'S1',21X,I12)
22      FORMAT(1X,'S2',21X,I12)
23      FORMAT(1X,'S3',21X,I12)
31      FORMAT(4X,'X',4I1,15X,I12)
        CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
        DO 100, I=1,IT
          IWGHT = 10*N*M*(IT-I+1)
        DO 100, I1=1,M
          WRITE(78,21)IWGHT
        DO 100, I2=0,N
        DO 100, I3=1, N+1
        IF(I2.EQ.I3)GOTO 100
        IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 100
        CALL INCR(IC3, IC2, IC1, IC0)
          IF((I2.EQ.0).AND.(I3.NE.N+1))
     X    WRITE(78,31) IC3,IC2,IC1,IC0
 100    CONTINUE
        CALL RTRC(6,IC3,IC2,IC1,IC0,ITRAC)
        DO 200, I=1,IT
          IWGHT = 10*N*M*(IT-I+1)
        DO 200, I1=1,M
          WRITE(78,21)IWGHT
        DO 200, I2=0,N
        DO 200, I3=1, N+1
        IF(I2.EQ.I3)GOTO 200
        IF((I2.EQ.0) .AND. (I3.EQ.N+1)) GOTO 200
        CALL INCR(IC3, IC2, IC1, IC0)
          IF((I2.NE.0).AND.(I3.EQ.N+1))
     X    WRITE(78,31) IC3,IC2,IC1,IC0
 200    CONTINUE
          WRITE(78,835)
835        FORMAT('ENDATA')
          CLOSE(78)
          WRITE(*,18)
18        FORMAT(5X,'LEAVING SOS.FOR')
C
```

```
C          LAST CARD OF SOS.FOR
          RETURN
        END
C
```

REFERENCES

REFERENCES

Baker, K. R. (1974). *Introduction to Sequencing and Scheduling*. New York: John Wiley & Sons.

Baker, K. R., & Peterson, D. W. (1979). An Analytic Framework for Evaluating Rolling Schedules. *Management Science*, *25*, 341-351.

Barnes, J. W., & Brennan, J. J. (1977). An Improved Algorithm for Scheduling Jobs on Identical Machines. *AIIE Transactions*, *9*, 25-31.

Barnes, J. W., & Vanston, L. K. (1981). Scheduling Jobs with Linear Delay Penalties and Sequence Dependent Setup Costs. *Operations Research*, *29*, 146-160.

Beale, E., & Tomlin, J. (1969). Special Facilities in a General Mathematical Program System for Non-Convex Problems Using Ordered Sets of Variables. In J. Lawrence (Ed.), *Proceedings of the 5th International Conference on Operations Research*. Tavistock, London.

Bruvold, N. T., & Evans, J. R. (1985). Flexible Mixed-Integer Programming Formulations for Production Scheduling Problems. *IIE Transactions*, *17*, 2-7.

Burch, E. E., Oliff, M. D., & Sumichrast, R. T. (1987). Linking Level Requirements in Production Planning and Scheduling. *Production and Inventory Management*, *28*, 123-131.

CPLEX Mixed Integer Optimizer (1993). Incline Village, NV: CPLEX Optimization, Inc.

Cheng, T. C. E., & Sin, C. C. S. (1990). A State-of-the-Art Review of Parallel-Machine Scheduling Research. *European Journal of Operational Research*, *47*, 271-292.

Day, J., & Hottenstein, M. (1970, March). Review of Sequencing Research. *Naval Research Logistics Quarterly*, 118-146.

Dearing, P. M., & Henderson, R. A. (1984). Assigning Looms in a Textile Weaving Operation with Changeover Limitations. Production and Inventory Management, 25, 23-31.

Driscoll, W. C., & Emmons, H. (1977). Scheduling Production on One Machine with Changeover Costs. AIIE Transactions, 9(4), 388-395.

Dyckhoff, H. (1990). A Typology of Cutting and Packing Problems. European Journal of Operational Research, 44, 145-159.

Elmaghraby, S. E., & Park, S. H. (1974). Scheduling Jobs on a Number of Identical Machines. AIIE Transactions, 6, 1-13.

Frendewey, J. O., & Sumichrast, R. T. (1988). Scheduling Parallel Processors with Setup Cost and Resource Limitations. Decision Sciences, 19, 138-146.

Ferreira, S. J., Neves, A. M., & Fonseca e Castro, P. (1990). A Two-Phase Roll Cutting Problem. European Journal of Operational Research, 44, 185-196.

Fisher, M. L. (1981). The Lagrangean Relaxation Method for Solving Integer Programming Problems. Management Science, 27, 1-18.

Geoffrion, A. M., & Graves, G., W. (1976). Scheduling Parallel Production Lines with Changeover Costs: Practical Application of a Quadratic Assignment/LP Approach. Operations Research, 24, 595-610.

Gilmore, P. C., & Gomory, R. E. (1961). A Linear Programming Approach to the Cutting Stock Problem. Operations Research, 9, 848-859.

Gilmore, P. C., & Gomory, R. E. (1963). A Linear Programming Approach to the Cutting Stock Problem, Part II. Operations Research, 11, 863-888.

Gilmore, P. C., & Gomory, R. E. (1965). Multistage Cutting Stock Problems of Two and More Dimensions. Operations Research, 13, 94-120.

Glassey, C. R. (1968). Minimum Changeover Scheduling of Several Products on One Machine. Operations Research, 16(2), 342-352.

Haessler, R. W. (1975). Controlling Cutting Pattern Changes in One-Dimensional Trim Problems. Operations Research, 23, 483-493.

Haessler, R. W. (1977). Single-Machine Roll Trim Problems and Solution Procedures. Technical Association of the Pulp and Paper Industry (TAPPI), 59, 145-149.

Haessler, R. W. (1988). Selection and Design of Heuristic Procedures for Solving Roll Trim Problems. Management Science, 34, 245-257.

Hax, A. C., & Meal, H. C. (1975). Hierarchical Integration of Production Planning and Scheduling. In M. Geisler (Ed.), North Holland/TIMS, Studies in Management Sciences, Vol. 1, Logistics (53-69). New York: North Holland/American Elsevier.

Hsu, W. L. (1983). On the General Feasibility Test of Scheduling Lot Sizes for Several Products on One Machine. Management Science, 22, 93-105.

Johns, E. C. (1966). Heuristic Procedures for Solving the Paper Trim Problem. Chapter 20 in J. F. Pierce (Ed.), Operations Research and the Design of Management Information Systems, TAPPI STAP Series No. 4.

Land, A., & Doig A., (1960). An Automatic Method of Solving Discrete Programming Problems. Econometrica, 28, 497-520.

LaRobardier, L. M., & Filak, R. J., (1972, October). Dynamic Allocation of Manufacturing Inventory and Time. APICS International Conference Proceedings, 254-272.

Lockett, A. G., & Muhlemann, A. P., (1972). A Scheduling Problem Involving Sequence Dependent Changeover Times. Operations Research, 20, 895-902.

Mellichamp, J. M. & Love, R. M., (1978). Production Switching Heuristics for Aggregate Production Planning Problem. Management Science, 1242-1251.

Mitsumori, S., (1972, September). Optimal Production Scheduling of Multicommodity in Flow Line. IEEE Transactions on Systems, Man, and Cybernetics, SMC-2.

Nelson, N. S., (1983). MRP and Inventory and Production Control in Process Industries. Production and Inventory Management, Fourth Quarter, 15-22.

Noble, P., (1973). Marketing Guide to the Paper and Pulp Industry. Charles H. Kline & Co., Inc.

Novitsky, M. P., (1983) Process Industry - Survey Results, Process Industry - Where are You? Production and Inventory Management, First Quarter, 118-120.

Oliff, M. D., & Burch, E. E., (1985). Multi-Product Production Scheduling at Owens-Corning Fiberglass. Interfaces, 15, 25-34.

Prabhakar, T., (1974). A Production Scheduling Problem with Sequencing Considerations. Management Science, 21, 34-42.

Pierce, J. F., (1964). Some Large-Scale Production Scheduling Problems in the Paper Industry. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Rice, J. W., & Norback, J. P., (1987). Process Industries Planning Using Matrix Data Structures. Production and Inventory Management, Second Quarter, 15-23.

Rothkopf, M. H., (1966). Scheduling Independent Tasks on Parallel Processors. Management Science, 12, 437-446.

Routledge, T. C., (1988, October). Mill Management's Role in Millwide Automation. Paper Industry Management Association.

Singh, H., & Foster, B. J., (1987, March). Production Scheduling with Sequence Dependent Setup Costs. IIE Transactions, 43-49.

Smith-Daniels, V. L., & Smith-Daniels, D. E., (1986, September). A Mixed Integer Programming Model for Lot Sizing and Sequencing Packaging Lines in the Process Industries. IIE Transactions, 278-285.

Smith-Daniels, V. L., (1988, September). A Heuristic Procedure for Scheduling Packaging Lines in the Process Industries. IIE Transactions, 20, 295-305.

Sweeney, P. E., & Haessler, R. W., (1990). One-Dimensional Cutting Stock Decisions for Rolls with Multiple Quality Grades. European Journal of Operational Research, 44, 224-231.

Taylor, S., (1979). Production and Inventory Management in the Process Industries: A State of the Art Survey. Production and Inventory Management, First Quarter, 1-6.

Tucker, A., (1960). On Directed Graphs and Integer Programs. IBM Mathematical Research Project Technical Report, Princeton University.