

**A MATHEMATICAL PROGRAMMING BASED MODEL AND ALGORITHM FOR A TWO STAGE
PRODUCTION PROCESS**

by

Muralidharan S. Kodialam

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Industrial Engineering and Operations Research

APPROVED:

Hanif D. Sherali, Ph.D, Chairman

Subhash C. Sarin, Ph.D

Faiz A. Al-Khayyal, Ph.D

August, 1987

Blacksburg, Virginia

Abstract

The research conducted in this thesis is concerned with the study of a two-stage production process existing at the Standard Register's (previously Burroughs Corporation) Paper Products Division plant in Rocky Mount, Va. . The objective is to develop a mathematical programming based model and algorithm to allocate and sequence work orders in order to improve the plant's productivity and reduce the labor and material costs. The proposed algorithm is based on Lagrangian Relaxation and Benders' decomposition techniques, which exploit the inherent generalized assignment and travelling salesman problem substructures in the model. The algorithm is computationally intensive and generates (near) optimal solutions with a reasonable amount of effort. Heuristic algorithmic procedures for the generation of good solutions for large-sized problems is also proposed.

Acknowledgements

I would like to thank my major advisor Dr. Hanif D. Sherali who conceived the idea for this project. His support and advice both during my coursework and my research have been invaluable.

I owe many thanks to Dr. Subhash C. Sarin and Dr. Faiz A. Al-Khayyal for serving on my committee and for offering constructive criticism of the project and the manuscript.

Finally, I would like to thank my family and friends for their constant support during my stay at Virginia Tech.

Table of Contents

1.0	Introduction	1
2.0	The Existing System for Scheduling and Sequencing	6
3.0	System Data Specification	10
4.0	Mathematical formulation of the Problem	14
5.0	Related Models and Algorithms in the Literature	23
6.0	Proposed Solution Algorithm for the Single Stage Problem	31
6.1	Pre processing Stage	32
6.1.1	Singleton Assignments	32
6.1.2	Checking for Feasibility	32
6.1.3	Checking Against the Incumbent Solution	33
6.2	Lagrangian relaxation	33
6.3	Subgradient Algorithm	36

6.4	Further Tightening of the Relaxation	38
7.0	Formulation and Solution of the Allocation Sub-problem	40
7.1	Allocation Problem Formulation	40
7.2	Allocation Problem Linearization	43
7.3	Implicit Enumeration Algorithm for the Linearized Allocation Problem	48
7.3.1	Heuristic Solution	49
7.3.2	Lagrangian Relaxation of MIP	53
7.3.3	Updating the Strongest Surrogate Cut	55
7.3.4	Logical Tests at the Nodes	56
7.3.5	Calculation of Lower Bound LB1	57
7.3.6	Benders' cut Based on the Solution from LB1	58
7.3.7	Computation of Lower Bound LB2	58
7.3.8	Surrogation of [39] and [42] and LB3	59
7.3.9	Branching Variable Choice	60
8.0	The Sequencing Sub-problem	61
8.1	Implicit Enumeration Algorithm for the Sequencing Sub-problem	63
8.1.1	Heuristic solution	63
8.1.2	Lagrangian Relaxation at Node Zero	65
8.1.3	Logical Tests	65
8.1.4	Lower Bound Calculation	67
8.1.4.1	Subtour elimination	68
8.1.4.2	Due Date Violation Constraint Generation	70
9.0	Solving Large Problems via the Heuristic Algorithms	71
10.0	Results and Conclusions	73

10.1	Results from runs on actual plant data	73
10.2	Results from runs on randomly generated data	83
10.3	Conclusions and Extensions	87
Appendix A. Printing presses.		97
A.1.1.1	A note on the Goebels :	101
Appendix B. Factors relating completion time and wastage to the sequencing of the jobs.		103
Appendix C. Factors unaffected by sequencing but resulting in time and paper wastage.		106
Appendix D. Collators		108
Appendix E. Normal job characteristics and limitations at the collators.		112
Appendix F. Collator cylinder sizes and slotting arrangements.		116
F.1.1.1	Western gear system set cylinders.	116
F.1.1.2	Western gear continuous cylinders	116
F.1.1.3	Hamilton continuous cylinders	117
Appendix G. Factors relating completion times, paper wastage and sequencing		118
Appendix H. Doctoring machines		122
Appendix I. Current scheduling considerations at the production level.		123
Appendix J. Some observations on plant operation.		126

Bibliography	127
Vita	132

List of Tables

Table 1. Job Numbering Scheme	77
Table 2. Press Numbering Scheme	77
Table 3. Problems with Data Set 1	84
Table 4. Problems with Data Set 2	84
Table 5. Exact Algorithms - Data set 1.	89
Table 6. Exact Algorithms - Data set 2.	90
Table 7. Allocation Heuristic 2 - Overlapping Batch Exchange Sequencing Procedure - Data Set 1	91
Table 8. Allocation Heuristic 2 - Overlapping Batch Exchange Sequencing Procedure - Data Set 2	92
Table 9. Allocation Heuristic 2 - Pairwise Interchange Heuristic - Data Set 1 ...	93
Table 10. Allocation Heuristic 2 - Pairwise Interchange Heuristic - Data Set 2 ...	95
Table 11. Overall Press Characteristics	99
Table 12. Collator Characteristics 1	110
Table 13. Collator Characteristics 2	111

List of Illustrations

Figure 1. Overall Strategy	22
Figure 2. Allocation Heuristic 2	52
Figure 3. Implemented Schedule	78
Figure 4. Resequencing the Implemented Schedule	79
Figure 5. Proposed Reallocated and Resequenced Schedule	80
Figure 6. Alternative Load Balanced Schedule	81
Figure 7. Load Balanced Schedule on Three Machines	82

1.0 Introduction

The research conducted in this thesis deals with the study of the printing and collating processes at the Standard Register's (previously Burroughs Corporation) Paper Products Division plant in Rocky Mount, Va.. The objective is to develop a mathematical programming based model and algorithm to schedule and sequence work orders in order to improve the plant's productivity and reduce the labor and material costs.

The Rocky Mount plant produces two kinds of business forms - namely, system set and continuous forms. A system set form size is specified by the stub length and the 'opposite' dimension. Continuous form size is specified by the width and the stub length of the form, and these differ from the system set forms in that they have line holes on both edges of the form. The actual form layout and details on file holes, line holes, perforations, colors, printing and spacing requirements are specified by the customer, and are recorded in a graphical format known as the production copy.

Orders received at the plant are pre-processed by the editing, order entry and plate preparation departments, before being sent down to the production level. The plate preparation department is allowed a standard time of 5 days, unless an exceptional amount of art work is required. Some orders may be 'exact repeat' orders, which means that the negatives for the plates are available at a Standard Register plant (either at Rocky Mount or elsewhere). However, these are also allowed the standard

time in the plate preparation department. The remaining pre-processing operations take 3 - 5 days, which means that a job typically will take 8 - 10 days to get to the shop floor.

A proof requirement by a customer could delay the actual start of the production process by a few more days. A standard response time of 7 days is allowed for the customer to make any changes in the proof provided.

Non availability of material and an exceptionally heavy load in pre-processing are other factors which could delay the actual production of the forms.

The order in which the jobs are pre-processed is determined by the due date on the job. Currently, a 'normal' turn around schedule is 4 weeks. A 'rush' job would have a date specified on it. There are some jobs which have a flexible due date (specified by an 'o.k. by' date) and others have a 'not before' date specification.

A normal load is approximately 80 orders per week but the variation in the daily number of orders is substantial. By the end of the week 80 - 90% of the jobs are pre-processed.

The entire production planning is done by the WOEPP (Work Order Entry Planning and Processing) system. The WOEPP system is a code written in Basic. It takes as its input the complete job specifications. It also has access to data on the plant characteristics, such as technical specifications of the presses and collators and time and material standards.

The job characteristics and the information from this data base are processed into a detailed step by step output for the production process. Among other things, it outputs the primary (and secondary) press and collator for which the job is planned, the standard hours to be allowed for the job at the presses and collators, the standard allowable impressions, the set up times and details on printing and collating requirements, such as paper type and quantity, ink, perforation requirements and numbering and collating instructions .

The scheduling and sequencing of the jobs is based on changeover costs (in terms of paper wastage - standard allowable impressions, and time in standard hours), machine load and job suitability and delivery dates. More details on this are given later.

The plant operates on 3 daily shifts with a different set of operators for the presses and the collators. Each operator is usually assigned to a single machine and takes care of the entire run on that machine. More details on plant operations are given in Appendix H.

There are 10 presses at the Rocky Mount plant - 7 of these are Western gear and 3 are Goebel presses. All the presses can be used for printing either system set or continuous forms. The major characteristic of a press is its circumferential cylinder size. The Western gear presses are of fixed cylinder size (17" or 22") whereas the Goebels can take different cylinder sizes (17",21",24" or 26"). Normally, a form size must divide exactly into the cylinder size of the press on which it is printed. This is necessary to ensure an integral number of forms around a cylinder. However, some system set forms are run with waste that is trimmed at the guillotine.

Other characteristics like the number of litho colors, rubber imprinting, press numbering, folding capability and maximum width of run vary from one press to another. Details on these and their effect on job suitability are given in Appendix A. Their relation to sequencing, paper wastage and set up times is outlined in Appendix B.

After the printing, most jobs require collating. This arises out of a number of requirements :

- a) Multiple ply forms - that is, each ply of the form is on a different roll. The plys have to be ordered, collated and fastened together.
- b) Carbon between plys and perforation requirement through the carbon.
- c) Collator numbering - most of the numbering is done at the collators. Forms may require numbering through the carbon.
- d) Folding, unwinding or sheeting of the collated plys.

However, some orders are completed at the presses. These include the single ply forms which are printed, numbered, perforated and finally folded, rolled or cut into sheets at the presses themselves, before being sent to the bindery if binding is required.

The plant has 12 collators - 4 of these are continuous form and 8 are system set collators. Conversion from one to the other type is possible, but it is very time consuming and is usually avoided. The other major characteristic, which differentiates one collator from another, is the number of paper and carbon stations on a collator. This limits the number of plys that a particular collator can handle. Besides these two characteristics, most of the other characteristics of a collator are flexible (including the cylinder size). This results in a greater flexibility in scheduling at the collators,

as compared to the scheduling at the presses, unless a job requires some special features (for example - handling jumbo rewinds, MICR numbering, batch stacking). Details of collator characteristics and limitations are given in Appendix D. The manner in which these relate to scheduling, paper wastage and set up time is outlined in Appendix E.

In between the printing and collating, some jobs need to pass through a doctoring machine. At this machine, an operator accurately splices together small rolls which have been crudely spliced together at the presses. Details on this are given in Appendix F.

Before developing our approach to schedule and sequence work orders on presses and collators, we briefly describe the existing but obsolete WOEP system which performs the same function currently, and specify some system characteristics which are relevant to the development of the mathematical programming model.

2.0 The Existing System for Scheduling and Sequencing

WOEPP is a system currently in use for making rudimentary scheduling and sequencing planning decisions. The information available on the WOEPP algorithm is sparse. As such it was not possible to determine the exact nature of the WOEPP sequencing routine. In any case, the scheduler usually ignores or overrides the WOEPP decisions because of the inadequacy of the proposed decisions. The following factors appear to be considered by WOEPP in the sequencing process.

Sequencing is done on the plys of individual jobs. However, all plys of the same job are planned for the same press and changeover cost evaluations are based on the plys of the same job. The jobs are processed as they are entered into the WOEPP system, and assigned to the first available press which does not have a load exceeding twenty four hours.

The changeover labor and material costs for each job are calculated based on the changeover costs between the plys. These calculations are based on certain sequence *independent* standards (for example - litho ink color change is allowed 100 standard impressions at the presses). Similarly, time allowances are made at the presses and the collators. Hence, irrespective of the difficulty or the degree of changeover, a standard allowance of time and material is made for each factor in the changeover.

Based on the above costs and a preemptive prioritizing of some five identified factors, a least cost ply sequence is selected for each job. The five factors given below are in the order of priority used. The different priorities allotted to the factors help resolve the ties which are inevitable. Ties will occur because the allowances are independent of the difficulty or the degree of the changeover.

The five factors considered are :

- a) minimize stock loss - that is if the aggregate changeover material costs are equal, then select the lowest cost stock to minimize the large set up cost.
- b) minimize the sum of the plate and color change costs in terms of the standard allowable changeover impressions.
- c) minimize file hole punch changes.
- d) minimize color change.
- e) minimize plate change.

This scheme has been observed to produce lower costs than any other manual attempt. However, there are some noticeable omissions in this scheme, which show up in the actual scheduling at the production level. Some of these factors are outlined below.

- a) There is no job to job sequencing which involves a substantial cost in terms of changeover material and labor costs. The current algorithm makes no attempt to minimize an aggregate changeover cost over all the jobs.
- b) The lack of job to job sequencing at the presses could result in a diverse mix of jobs being available for collating in any given duration of time. This could result in a substantial increase in collator changeover costs. Hence, collator compatibility must be considered in the sequencing at the presses.

c) There is no cost factor associated with actual changeover times. It is not clear whether these are (if at all) directly or indirectly taken into account by the WOEPF sequencing.

d) The degree and difficulty of changeover is not considered in the cost evaluations. For example, if a perforation change only requires a removal of blades, it would be allotted the same allowances of time and paper wastage as a change requiring the removal and insertion of new blades.

e) Certain factors which appear to affect material and time allowances at the production level, have not been incorporated in the sequencing algorithm. For example, cross perforation changes affect time allowances substantially and jobs with similar cross perforation requirements are sequenced one behind the other (if other factors do not prevent this) at the production level (further details are given in Appendix B).

The rescheduling that is performed at the production level accounts somewhat for the above factors. This rescheduling is required to handle the dynamic situation on the shop floor. The scheduler at the production level is constantly aware of the current conditions on the shop floor. He uses his experience and judgement to minimize material and labor costs while meeting the due dates on the jobs. Some of these considerations are outlined in Appendices G and I.

Factor (a) is considered in the context of the scheduling model. The 'first come first served' approach to sequencing has been known to yield suboptimal results.

The scheduling model that is developed in this research tries to minimize the aggregate of the material and labor changeover costs over all the jobs to be scheduled. Moreover, collator changeover costs are indirectly taken into account for the press scheduling. This tends to ensure that jobs with similar collator characteristics will be

completed (at the presses) close to one another. This in turn permits them to be sequenced one behind the other on the same collator. Due dates, press and collator limitations are taken care of in the form of constraints which must be satisfied.

3.0 System Data Specification

The model outlined in the next chapter requires specific data on job and machine characteristics. The following is a summary of the data required and the probable sources of this data. The data from the WOEP system has been used as far as possible. As pointed out earlier, some additional scheduling considerations arise at the production level and these have been incorporated in the model. However, the WOEP system will not be able to supply this additional data without some modifications. We may need to generate this data externally for input to the model.

1) The variable j_k represents the job for which machine k has been initially set. This is specified in terms of the characteristics of the last job run on press k and would be required for each press available for scheduling.

The characteristics required are those that affect changeover costs and are listed below :

- a) number width and width of run.
- b) number of litho colors and specific litho colors
- c) backprinting
- d) file hole punching specifications
- e) line hole punching
 - setting across web
 - system set or continuous form
- f) rotary perf specifications

number of wheels

spacing of blades across web

type of blade - blade number

g) slit location

h) cross perforation specifications

number of blades on cylinder

spacing of blades

type of blade - blade number

i) sheet, fold or rewind requirements

j) rubber imprinting - face or back

k) press numbering and type

The WOEP system currently compares the characteristics of the plies of the same job. This will need modifications or external code to provide changeover costs between the last ply of the previous job on each machine and the second ply of the current job assignable to that machine.

2) J_k represents the set of jobs assignable to machine k. This is a subset of the schedulable jobs and comprises all of the jobs capable of being performed on machine k.

The following machine characteristics will determine this :

a) circumferential cylinder size (17",22",24",26")

b) maximum number of litho colors

c) press numbering capability

d) maximum width of press as compared to the recommended width of run for the job (based on order quantity).

e) specific machine characteristics (for example - line hole punching closer than a specified distance is not possible at some presses because of fixed punch rings).

With some modifications in the WOEPF code we should be able to obtain the list of jobs assignable to a certain machine. This follows because the WOEPF code must be identifying the machines suitable for a certain job before evaluating changeover costs for scheduling.

3) K_j represents the set of machines which can perform a given job j . This will have to be generated external (to WOEPF) using the information from (2) above.

4) The operation time of job j on machine k is represented by the variable t_{jk}^* . WOEPF gives an estimate of the run time allowed for a job based on the run speeds that it recommends. This value pertains to some particular machine for each job. Some modification or external code will be required to calculate the run time of job j on each machine to which it can be assigned (obtained from K_j).

5) The changeover time from job i to job j on machine k is represented by the variable t_{ijk}^* . WOEPF outputs a changeover time between plies of the same job for the machine that it selects for the job. External code will be required to obtain this for each machine in K_j , and also to output changeover times between jobs.

6) The cost of assigning job j to machine k is represented by the variable c_{jk}^* . This includes the labor and material costs involved. The cost of paper would include consideration of the total paper usage, cost of paper per yard and sequence independent wastage (dependent on job and machine characteristics). The cost of time

could be taken as an average labor cost per hour. However, it will have to be scaled to account for the weightage given to this factor relative to paper costs.

7) The cost of changeover from job i to job j on machine k is represented by the variable c_{ij}^k . This will depend on the cost of time required for the changeover and the paper wastage and time required for the changeover.

The variables in (5),(6) and (7) are estimated by WOEPP for the individual plys of a job. Some modifications will be needed to obtain the values for each press and on a job to job basis.

8) The due date for job j at the presses is represented by the variable d_j . Based on the overall due date, the algorithm will first draw up a line up for the collators. This will be used to estimate a latest completion time on the presses. This could be used for d_j . For the collator problem, the actual due date on the job (estimated by WOEPP) will be used.

4.0 Mathematical formulation of the Problem

In order to put the Two-Stage (Printer-Collator) problem in perspective, consider the following conceptualization and mathematical formulation of the problem. The fundamental and salient features of the joint printer-collator problem are represented below. Further embellishments and specification of additional constraints pertaining to particular applications may be incorporated in the model as desired.

Let $k = 1, \dots, m$ index the presses and $l = 1, \dots, n$ index the collators. Let j_k and $\bar{j}_l$ denote the initial job settings for printer k and collator l respectively. Let J_k and $\bar{J}_l$ denote the jobs which can be processed on press k and collator l respectively. As an inverse to the index sets J_k and $\bar{J}_l$ we will refer to the set K_j and L_j as being the sets of presses and collators to which job j can be assigned. The decision variables for the problem are defined as follows.

$$x_{ij}^* = \begin{cases} 1 & \text{if press } k \text{ operates on job } i \text{ followed by job } j \\ 0 & \text{otherwise} \end{cases}$$

for each $i, j \in J_k \cup \bar{J}_k, i \neq j, k = 1, \dots, m$.

[1]

$$\tilde{x}_{ij}^l = \begin{cases} 1 & \text{if collator } l \text{ operates on job } i \text{ followed by job } j \\ 0 & \text{otherwise} \end{cases}$$

for each $i, j \in \bar{J}_l \cup \bar{j}_l, i \neq j, l = 1, \dots, m$.

$$y_j^k = \begin{cases} 1 & \text{if job } j \text{ is assigned to press } k \\ 0 & \text{otherwise} \end{cases}$$

[2a]

$$\tilde{y}_j^l = \begin{cases} 1 & \text{if job } j \text{ is assigned to collator } l \\ 0 & \text{otherwise} \end{cases}$$

The first set of decision variables account for the sequencing decisions, and the second set account for the job-to-machine allocation decisions.

Additionally, to aid in the mathematical formulation, let us also define the (decision) variables

T_j = Completion time of job j on the presses.

[2b]

$\tilde{T}_j$ = Completion time of job j on the collators.

Below, we present the mathematical formulation of the joint press-collator problem. Note that each press / collator is conceived to begin with job $j_k / \bar{j}_l$, thread through a series of jobs and then close a loop by returning to the first job. The final return is only to indicate that the press/collator is done with all the jobs. Notationally, t_j^k and $\bar{t}_j^l$ denote the operating time of job j on press k and collator l respectively, c_j^k is the cost of assigning job j on press k , c_{ij}^k is the changeover cost from job i to job j on printer k , $\bar{c}_j^l$ and $\bar{c}_{il}^l$ are the corresponding costs on the collator, and $\bar{d}_j$ is the due date of job j .

We have also included load balancing constraints on the printers and the collators, requiring that the total operating time is less than or equal to some value R_k for the press k and $\bar{R}_l$ for collator l , for each $k = 1 \dots m$ and $l = 1 \dots n$.

The quantities $\tau_j^k = \text{Min}_{K \neq k} [t_j^k + T_{ik}]$, $\bar{\tau}_j^l = \text{Min}_{L \neq l} [\bar{t}_j^l + \bar{T}_{il}]$, $R_{ik} = \min \{R_k, d_i\}$ and $\bar{R}_{il} = \min \{\bar{R}_l, \bar{d}_i\}$ are dummy coefficients used to enforce the sequencing constraints.

Minimize

$$\sum_{k=1}^m \sum_{i \in J_k \cup j_k} \sum_{j \in J_k} c_{ij}^k x_{ij}^k + \sum_{k=1}^m \sum_{j \in J_k} c_j^k y_j^k + \sum_{l=1}^n \sum_{i \in J_l \cup j_l} \sum_{j \in J_l} \bar{c}_{ij}^l \bar{x}_{ij}^l + \sum_{l=1}^n \sum_{j \in J_l} \bar{c}_j^l \bar{y}_j^l$$

Subject to

PRINTER PROBLEM CONSTRAINTS

$$\begin{aligned} \sum_{k \in K_j} y_j^k &= 1 \quad \forall j \\ \sum_{j \neq p} x_{pj}^k - y_p^k &= 0 \quad \forall p \in J_k \cup j_k \quad \forall k \\ \sum_{i \neq p} x_{ip}^k - y_p^k &= 0 \quad \forall p \in J_k \cup j_k \quad \forall k \\ T_j &\geq T_i + t_{ij}^* x_{ij}^k + t_j^* y_j^k \\ &+ (1 - y_j^k) \tau_j^* + (x_{ij}^k - 1) R_{ik} \quad \forall \begin{matrix} i \in J_k \cup j_k \\ i \neq j \end{matrix} \quad \forall k \in K_j \quad \forall j \\ \sum_{j \in J_k} t_j^* y_j^k &\leq R_k \quad \forall k \end{aligned}$$

COLLATOR PROBLEM CONSTRAINTS

$$\begin{aligned}
 \sum_{l \in L_j} \tilde{y}'_l &= 1 \quad \forall j \\
 \sum_{p \neq j} \tilde{x}'_{pj} - \tilde{y}'_p &= 0 \quad \forall p \in \tilde{J}_I \cup \tilde{J}_I \quad \forall I \\
 + \sum_{p \neq j} \tilde{x}'_{ip} - \tilde{y}'_p &= 0 \quad \forall p \in \tilde{J}_I \cup \tilde{J}_I \quad \forall I \\
 \tilde{T}_I &\geq \tilde{T}_I + \tilde{t}_{ij} \tilde{x}'_{ij} + \tilde{t}_{ij} \tilde{y}'_j \\
 + (1 - \tilde{y}'_j) \tilde{\tau}'_j + (\tilde{x}'_{ij} - 1) \tilde{R}_{ij} &\quad \forall i \in \tilde{J}_I \cup \tilde{J}_I \quad \forall I \in L, \quad \forall j \\
 \sum_{j \in J_I} \tilde{t}_{ij} \tilde{y}'_j &\leq \tilde{R}_I \quad \forall I
 \end{aligned}$$

COUPLING CONSTRAINTS

$$\tau_j + \sum_i \tilde{t}_{ij} \tilde{y}'_j \leq \tilde{T}_j \leq \tilde{d}_j \quad \forall j$$

VARIABLE RESTRICTIONS

$$\begin{aligned}
 x^*_{ij}, \tilde{x}'_{ij}, y^*_j, \tilde{y}'_j &\text{ binary} \\
 \tau_j, \tilde{T}_j &\geq 0
 \end{aligned}$$

The two stages of the problem can be decoupled if the work in process inventory between the machines is high. In that case the coupling constraints will hold automatically, even if ignored, so long as some consideration is given to sequencing jobs on the presses in a manner which results in desirable job availability times for the collators. Ignoring these coupling constraints, the problem can be transformed into two similar problems, one for the printer and one for the collator. The press problem is described in detail below. The collator problem is identical to this problem and is not explicitly presented.

$$\text{minimize } \sum_{k=1}^m \sum_{i \in (J_k \cup J_k)} \sum_{j \in J_k} c_{ij}^k x_{ij}^k + \sum_{k=1}^m \sum_{j \in J_k} c_j^k y_j^k \quad [3]$$

subject to

$$\sum_{k \in K_j} y_j^k = 1 \quad \forall j \quad [4]$$

$$\sum_{\substack{j \in (J_k \cup J_k) \\ j \neq p}} x_{pj}^k = y_p^k \quad \forall p \in J_k \cup J_k, \forall k \quad [5]$$

$$\sum_{\substack{i \in (J_k \cup J_k) \\ i \neq p}} x_{ip}^k = y_p^k \quad \forall p \in J_k \cup J_k, \forall k \quad [6]$$

$$T_j \geq T_i + t_{ij}^k x_{ij}^k + t_j^k y_j^k + (1 - y_j^k) \tau_j^k + R_{ik}(x_{ij}^k - 1) \quad i \in (J_k \cup J_k), i \neq j, k \in K_j, \forall j \quad [7]$$

$$\sum_{j \in K_j} y_j^k [T_{j_k} + t_j^k] \leq T_j \leq d_j \quad \forall j \quad [8]$$

$$\sum_{j \in J_k} t_j^k y_j^k \leq R_k \quad [9]$$

$$x_{ij}^k \text{ binary for } i, j \in (J_k \cup j_k), i \neq j, \forall k \quad [10]$$

$$y_j^k \text{ binary for } j \in (J_k \cup j_k), \forall k$$

$$T_j \geq 0, \forall j$$

where R_k and τ_j^k are defined as before. Here, d_i is a due date implied for the presses by job availability requirements on the collators. (They may be derived by approximately solving the collator problem or by other considerations).

The above problem is a linear mixed-integer zero-one program with the principal variables being the binary variables [1], and with the other continuous variables [2] which can be treated as continuous (with the y variables bounded on $[0,1]$), being related to these variables. In this program, observe that constraint [4] says that each job j should be assigned to exactly one assignable machine. Constraint [5] says that if a job is assigned to machine k , it should be followed by exactly one other job (which may be a dummy job). Similarly, constraint [6] says that each job p assigned to machine k should be sequenced to follow exactly one other job.

The constraints [7] imply the subtour elimination constraints, which prohibit any machine k from involving itself with a cyclic sequence of jobs not involving the "dummy" job j_k . If there was such a sequence not involving job j_k these constraints would be violated. Hence, the subtour elimination constraints are not explicitly included in the formulation. They may be used (in a relaxed form) to provide a tighter

linear programming relaxation. The constraints [7] relate the finish times of the jobs. Note that the coefficients τ_j^k and R_{jk} in these constraints are used simply to ensure that these constraints are redundant whenever $x_{ij}^k = 0$ or $y_j^k = 0$. Constraints [8] are the due date restrictions, while constraints [9] are the load balancing requirements. The constraints [10] specify feasible variable values.

Note that the above problem has imbedded within itself an allocation structure based on the assignment of jobs to machines and a routing or Multiple Travelling Salesman Problem (MTSP) structure based on the loops or sequences of jobs which each machine threads through. This structure may be suitably exploited by any algorithm.

The overall strategy for the joint problem is to decompose it into a printer problem and a collator problem of the above type. The collator problem is solved first. Once jobs have been allotted to the different collators, the jobs are arranged in an Earliest Due Date (EDD) sequence. Arranging the jobs in an EDD sequence minimizes the maximum lateness of the jobs. If there exists a sequence in which all the jobs are on time, they are on time in the EDD sequence too. The start time of the jobs on the collators when arranged in an EDD sequence gives the implied due date of the jobs on the presses. This is taken as the due date for the jobs on the presses. Next, the press allocation and sequencing problems are solved. This completion time of the jobs on the presses gives the actual time at which processing can be started on the collators. The sequencing problem is solved on the collators taking into consideration these earliest starting times. The overall strategy is shown in Figure 1.

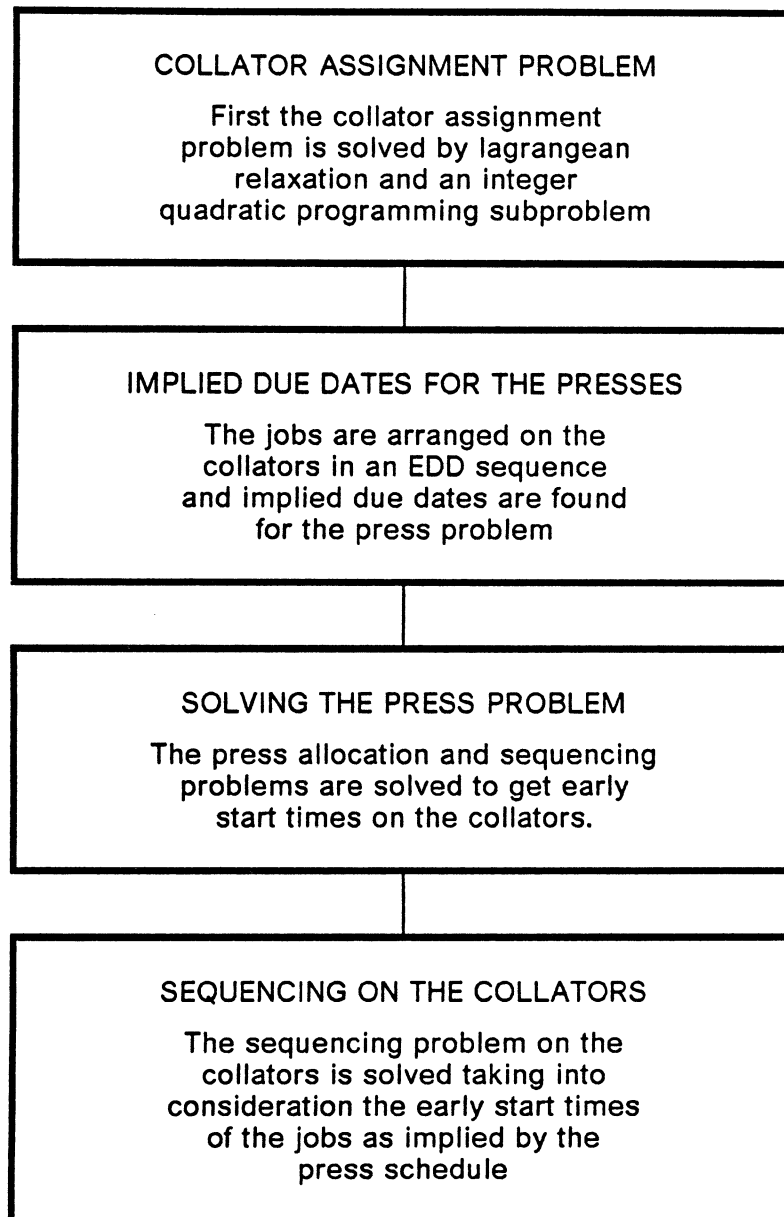


Figure 1. Overall Strategy

5.0 Related Models and Algorithms in the Literature

The general form of our model [3]-[10] falls within the class of Vehicle Routing Problems (VRP) with side constraints. The Vehicle Routing Problem is a generic name given to a whole class of problems involving the visiting of "customers" by "vehicles". Each customer requires a quantity of some product to be delivered by one of a number of vehicles which involves vehicles of given capacities. The cost is route dependent. The objective is to route the given set of vehicles through the customers so as to satisfy their demands while minimizing the total cost incurred. Side constraints can be in the form of capacities of the vehicles, maximum time permitted on any route, time-windows for servicing, and routing restrictions. Hence, these problems usually have the general assignment and Travelling Salesman Problem substructures. In our problem, the given set of machines are the vehicles. The jobs are the customers and a route is a sequence in which a machine processes the jobs allotted to it. The cost of a route is the sum of the changeover costs incurred in going from one job to another in the sequence. The side constraints are in the form of load balancing constraints, restrictions on the capacities of the machines, and due dates. Because of the combinatorial nature of VRP's, mostly heuristic procedures have been proposed for any realistically sized problem.

Some of the more successful heuristics developed are outlined below.

The solution strategies adopted for the VRP fall into one of the following general categories (Bodin,Golden,Assad and Ball, [1981]) :

- a) Cluster first - route second
- b) Route first - cluster second
- c) Savings/insertion scheme
- d) Improvement/exchange scheme
- e) Mathematical Programming based heuristics
- f) Interactive optimization procedures
- g) Exact procedures

The methods (a) - (d) use heuristic ideas while (e) and (f) combine exact procedures with heuristic ideas. A brief description of the ideas used in each of these methods is given below.

a) Cluster first - route second procedures cluster the demand nodes and/or arcs first and then design economical routes over each cluster. Various heuristic ideas have been used in the cluster and routing stages. Examples can be found in Gillett and Miller [1974], Gillett and Johnson [1976], and Karp [1977] for the standard single depot vehicle routing problem.

b) Route first - cluster second procedures first find a route (usually infeasible) through all the demand nodes. Next, the large route is partitioned into smaller feasible routes. Examples can be found in Levy et al. [1978], Newton and Thomas

[1969], Bodin and Berman [1979], Bodin and Kursh [1979],[1978], Stern and Dror [1979].

c) Savings or insertion procedures build a solution such that at each step of the procedure (up to the penultimate step) a comparison is made between the current configuration and improving alternate configurations. Either of these configurations may be infeasible, and feasibility is achieved in the concluding step. The improvement in the alternate configurations may be measured in terms of total cost or least cost insertions. Clarke and Wright [1964], Golden et al. [1977], Norback and Love [1977], Golden and Wong [1981], Hinson and Mulherkar [1975], have used variations of the above procedure.

e) Improvement or exchange procedures are branch exchange heuristics which always maintain feasibility and strive towards optimality. At each step an alteration to the current feasible solution leads to an improved feasible solution. These heuristics have been developed by Lin [1965], Lin and Kernighan [1973], Christofides and Eilon (extensions) [1972], Russel [1977] and Bodin and Sexton [1980].

f) Mathematical programming approaches exploit the two underlying structures in the VRP, namely the Travelling Salesman Problem and the Generalized Assignment Problem. These two problems have been studied extensively and powerful mathematical programming approaches for their solution already exist. Fisher and Jaikumar [1981] use heuristics which take advantage of the powerful solution methods which exist for the two subproblems. Christofides et al. [1981] and Stewart and Golden [1979] have used Lagrangean relaxation and Dynamic programming approaches to obtain lower bounds for some related problems.

g) Interactive optimization is a general purpose approach in which a high degree of human interaction is incorporated into the solution procedure. Some implementations can be found in Krolak, Felts, and Marble [1971], Krolak, Felts and Nelson [1972] and Cullen, Jarvis and Ratliff [1981].

h) Exact procedures include branch and bound and cutting plane algorithms. These have been used effectively on smaller problems with typically no side constraints. The largest problem solvable involves only 30 customers. Examples are procedures developed by Held and Karp [1970] [1971], Crowder and Padberg [1980] and Christofides et al. [1981]. The basic idea is to use a relaxation to obtain good lower bounds on the optimal objective value and to use this in a branch and bound context.

A related problem to the VRP but with fewer constraints is the Multiple Travelling Salesman Problem (MTSP). In its simplest formulation there are no capacity restrictions on the tours or vehicles, and all salesmen or vehicles are assumed to be identical. Hence, the MTSP with appropriate capacity and time-window side constraints is of the form of our problem.

Most of the MTSP solution procedures first convert the MTSP into a single Travelling Salesman Problem and then use TSP solution strategies on the transformed problem. The concept in obtaining an equivalence is to create M copies of the depot each connected to the other nodes exactly as was the original depot (i.e. with the same length arcs). The M nodes themselves are not connected (or may be connected with large cost arcs). Then a single TSP solution will never use an arc connecting two

copies of the depot. By subsequently coalescing the copies back into a single depot, the M required tours are obtained.

Examples of equivalent TSP formulations of the MTSP can be found in Eilon et al. [1971], Bellmore and Hong [1974], Orloff [1974], Svestka and Huckfeldt [1973], and Gorenstein [1977]. A MTSP with side constraints is solved as a TSP with the side constraints handled explicitly. This implies that the TSP solution procedure will start off at a feasible point and improving changes will be made only if they are feasible to the side constraints. An example of this can be found in Russell [1976]. A related problem of scheduling on parallel processors with quadratic changeover cost has been addressed in Geoffrion and Graves [1976].

As evident from the above discussion, the Travelling Salesman problem constitutes an important structure in our model. The Travelling Salesman problem has been dealt with in detail by Lawler et al. [1985]. This problem is well known to be NP complete Karp [1972], even with simplifying assumptions of triangle inequality or Euclidean distances (Garey et al. [1976], Papadimitriou [1977]). Hence, polynomially bounded exact algorithms are unlikely to exist. Some good exact algorithms have been proposed by Crowder and Padberg [1980], Held and Karp [1970] [1971], Little et al. [1963], Miliotis [1976], but they are limited to cases of less than 300 nodes. (However, some recent advances by Crowder, Padburg and Johnson [1983] involves solving sparse problems upto about 1000 nodes). Nonetheless, for general or dense larger problems, heuristic procedures are employed. An analytical comparison (worst case behavior) of these heuristics can be found in Rosenkrantz, et al.[1977]. A computational comparison can be found in Golden, et al. [1980] and Stewart [1973].

One very successful exact procedure for solving the TSP involves the use of a Lagrangean relaxation scheme along with a subgradient optimization in order to provide tight lower bounds in a branch and bound context. The use of different formulations and Lagrangean relaxations typifies the exact procedures developed for the TSP. Examples can be found in Balas and Christofides [1981] and Houck et al. [1980].

The heuristic procedures fall into three broad classes (Golden, et al. [1979]). These are tour construction, tour improvement and composite procedures.

a) Tour construction procedures generate a good quality tour from the distance matrix. Rosenkrantz et al. [1977] have tried a nearest neighbor procedure. Golden [1964] has proposed to use the Clarke and Wright [1964] savings algorithm in which, starting from any node, arcs are added into the tour according to the savings resulting from the addition. Insertion procedures have been investigated by Rosenkrantz et al. [1977]. Here, a subtour on k nodes is available at iteration k . Then the next node to join the subtour is selected before determining its position in the subtour. The position of this node in the subtour is itself decided using heuristic ideas (for example : nearest insertion, cheapest insertion, farthest insertion). Kim [1975] and Christofides [1976] have proposed techniques using minimal spanning trees of the TSP graph. The nearest merger method of Rosenkrantz et al. [1977] constructs a sequence $S(1), \dots, S(N)$ such that each $S(i)$ is a set of $N - i + 1$ disjoint subtours covering all the nodes. Then $S(i + 1)$ is obtained from $S(i)$ by merging the two subtours closest to one another. Karp [1977] presents a partitioning algorithm for the TSP in a plane. The key idea here is to partition a large rectangle into a number of smaller subrectangles and to solve a TSP over each subrectangle. The outcome is an Euler cycle which can be

easily transformed into a tour over all the nodes. Bartholdi and Platzman [1982] make use of space filling curves to heuristically construct good tours.

b) Tour improvement procedures attempt to find a better tour starting from an initial tour. The best known heuristics in this category are the branch exchange heuristics. Lin [1982] and Lin and Kernighan [1973] have introduced the 2-opt, 3-opt and the k-opt ($k \geq 3$) procedures. A k-opt tour is one in which no k-change can be made to further improve the solution value. A k-change is defined as the deletion of k branches in a tour and their replacement by k other branches to form a new improved tour. Rosenkrantz et al. [1977] have shown that a 2-opt tour can be almost twice the length of an optimal tour. Roughly, a 3-opt procedure requires n times the work that a 2-opt procedure needs, where n is the number of nodes. Hence, composite procedures are investigated which are less expensive than the 3-opt procedure but are almost as accurate.

c) Composite procedures start off by obtaining a good initial tour. Then a 2-opt procedure is used, followed by a 3-opt procedure. This has been found to be relatively fast computationally and gives excellent results. It uses the 3-opt procedure only once (saving on the most expensive step) as opposed to the use of the procedure in (a), where the 3-opt procedure would be used repeatedly until no additional improvement can be made. Variations on this have also been tried.

All the above procedures have been found useful for symmetric TSP's, although some can be applied to asymmetric TSP's. Specific algorithms for asymmetric TSP's have been proposed by Akl [1978]. Frieze [1978], Karp [1979] and Kanellakis and Papadimitriou [1980]. Akl's directed TSP approach finds a minimal directed spanning

graph of the TSP graph, adds arcs to this to transform it into an Eulerian graph, finds an Euler cycle in this graph and converts it into a Hamiltonian cycle (Bondi and Murty [1976]).

At this stage, user enforced sequences/ job positions in the sequence can be easily incorporated as in Chan and Graffe [1979]. Sensitivity analysis involving ' what if ' questions pertaining to enforced job sequences, changes in resources/manpower availability can be handled at this stage.

We now proceed to develop a specialized solution algorithm for the our problem. This algorithm exploits the generalized assignment and travelling salesman sub-structures in our model, but in a manner somewhat different from the above mentioned vehicle routing algorithms, which takes into consideration the specific nature of our problem. However, the proposed algorithm can readily be extended to solve more general types of Vehicle Routing Problems with side constraints.

6.0 Proposed Solution Algorithm for the Single Stage Problem

In this section we discuss the solution procedure for the printer or the collator stage of the problem. Initially a Lagrangian relaxation of the problem is solved by dualizing the allocation, the load balancing constraints and the subtour elimination constraints. A near optimal set of dual multipliers is obtained by using a subgradient based algorithm on the Lagrangian dual. The dualizing of the above constraints results in subproblems with a simple network structure which are solved using a minimum cost network flow programming algorithm. Based on this, an allocation subproblem is derived. Once the allocation has been determined, the sequencing on the machines involves the solution of Travelling Salesmen Problems with time-window side constraints, one for each of the machines. This is done using an implicit enumeration scheme with logical tests to fix variables and a Lagrangian Relaxation scheme to obtain lower bounds.

6.1 Pre processing Stage

We do the following processing of the given data in order to fix the variables a priori before starting the actual solution procedure.

6.1.1 Singleton Assignments

There might be some job (j) which can be processed on only one particular machine (k), in which case we set $y_j^k = 1$ for such combinations of jobs and machines.

6.1.2 Checking for Feasibility

If the sum of the processing times of the set of jobs (A) already allotted to a machine (k) and the processing time of some job (j) on the machine is greater than the maximum load permitted on the machine, then we cannot allocate job j to machine k.

$$\sum_{i \in A} t_i^k + t_j^k > R_k \text{ implies } y_j^k = 0$$

6.1.3 Checking Against the Incumbent Solution

If the sum of the fixed assignment cost on a machine (k), the changeover cost from job i to job j, and the allotment cost of job j to machine k is greater than the incumbent value and if job i has already been allotted on machine k then job j cannot be allotted on machine k. That is,

$$\begin{aligned} [\text{Fixed assignment cost}] + \max_{i \in A} \text{Min} \{ c_{ij}^k, c_{ji}^k \} \\ + c_j^k \geq \text{Incumbent value} \end{aligned}$$

implies $y_j^k = 0$.

Note that we assume that the triangle inequality holds for the changeover costs. Also, the second term above may be replaced by any greater lower bound on the changeover cost amongst allotted jobs, if available.

6.2 Lagrangian relaxation

After fixing some of the variables in the pre-processing stage, we formulate a Lagrangian dual for the problem by dualizing constraints [4], [7], and [9]. Since the so-called integrality property (Taha [1975]) holds, solving this Lagrangian dual problem is equivalent to solving the linear programming relaxation to the mixed integer programming problem. The subproblem obtained for this relaxation has a network structure, separable over the machine index k. These problems can be

solved easily using the Out Of Kilter algorithm, (Bazaraa and Jarvis [1979]) for example.

Hence, let us dualize the allotment constraint [4], the subtour breaking constraint [7], and the load balancing constraint [9]. associating dual variables α_j with [4], β_{ij}^k/R_{ik} with [7] and γ_k/R_k with [9]. The scaling of the equations is performed to obtain unity for the right hand sides of all the equations; this accelerates the convergence process. With this transformation, the objective function of the Lagrangian subproblem becomes

$$\sum_k \sum_{i \in J_k \cup J_k} \sum_{j \in J_k} M_{ij}^k + \sum_k \sum_{i \in J_k \cup J_k} \sum_{j \in J_k} N_{ij}^k x_{ij}^k + \sum_k \sum_{j \in J_k} P_j^k y_j^k + \sum_{j \in J_k} Q_j T_j$$

where

$$M_{ij}^k = \sum_j \alpha_j + \sum_j \sum_{k \in K_j} \sum_{\substack{i \in J_k \\ i \neq j}} \frac{\beta_{ij}^k (\tau_j^k - R_{ik})}{R_{ik}} + \sum_j \sum_{k \in K_j} (\tau_j^k - R_{ik} + T_{ik}) \frac{\beta_{ik}^k}{R_{ik}} - \sum_k \gamma_k \quad [11]$$

$$N_{ij}^k = \left[c_{ij}^k + \left(1 + \frac{t_{ij}^k}{R_{ik}} \right) \beta_{ij}^k \right] \quad [12]$$

$$P_j^k = \left[c_j^k - \alpha_j + \sum_{\substack{i \in J_k \cup J_k \\ i \neq j}} \frac{(t_j^k - \tau_j^k)}{R_{ik}} \beta_{ij}^k + \frac{t_j^k \gamma_k}{R_k} \right] \quad [13]$$

$$Q_j = \sum_{k \in K_j} \left[\sum_{\substack{i \in J_k \\ i \neq j}} \frac{\beta_{ji}^k}{R_{ik}} - \sum_{\substack{i \in J_k \cup J_k \\ i \neq j}} \frac{\beta_{ij}^k}{R_{ik}} \right] \quad [14]$$

The constraints in the Lagrangian subproblem are [5], [6], [8], and [10]. In these constraints, the variables T_j have a simple lower-upper bounding structure only, and do not interact with other constraints. Therefore we first look at the coefficient of T_j in the objective function of the Lagrangian subproblem. If the coefficient of some T_j is less than zero, T_j is set equal to d_j . If the coefficient of T_j is greater than or equal to zero then T_j is set at $\sum_{k \in K_j} [T_{j_k} + t_j^*] y_j^*$. This accordingly modifies the objective coefficient of y_j^* . Now, the remaining constraints have a network structure and are separable over each k . Mathematically, for each k we have

$$\sum_{j \neq p} x_{pj}^k - y_p^k = 0 \quad \text{for each } p \in J_k \cup j_k$$

$$- \sum_{i \neq p} x_{ip}^k + y_p^k = 0 \quad \text{for each } p \in J_k \cup j_k$$

These constraints are associated with a bipartite graph (Bondi and Murty [1976]) on a node set $(J_k \cup j_k) \times (J_k \cup j_k)$. Each x_{rs}^k has a +1 at node r and a -1 at node s ($s \neq r$). Similarly, the y_p^k have a +1 and a -1 between nodes p and p for each k . The objective function has separable coefficients over each k , which results in a minimum cost network flow problem for each machine. The resulting network problem in x and y is solved using the Out of Kilter algorithm.

Hence, the Lagrangian problem is to

Maximize $\{ \theta(\pi) : \alpha_j \text{ unrestricted}, \beta_{ij}^k \geq 0, \gamma_k \geq 0, \forall i,j,k \text{ where } \pi = (\alpha, \beta, \gamma) \}$

A near optimal solution to this problem is obtained using a subgradient optimization scheme described next.

6.3 Subgradient Algorithm

The concave Lagrangian dual objective function $\theta(\cdot)$ is maximized over the dual space using a conjugate subgradient optimization scheme which combines the ideas of Camerini, Fratta and Maffioli [1975] and Bazaraa and Sherali [1981].

Let us denote the k^{th} iterate by π^k . Let $P_D(\pi) = \bar{\pi}$ be the projection of π onto the feasible space D , where D is given by

$$D = \{ \pi = (\alpha, \beta, \gamma) : \alpha \text{ unrestricted}, \beta, \gamma \geq 0 \}$$

Let ξ^k denote a subgradient of $\theta(\cdot)$ at π^k . The components of ξ will correspond to the coefficient of α, β, γ in the Lagrangean objective function.

INITIALIZING THE ALGORITHM

1. An initial feasible starting set $\pi^0 = (\alpha^0, \beta^0, \gamma^0)$ is needed to start the algorithm. (We used $\pi^0 = 0$ – vector).
2. We need an upper bound on the value of $\theta(\pi)$ (This was obtained via a heuristic solution to the original problem).

3. We use the following parameters in the subgradient scheme.

N = Maximum number of iterations.

$\bar{v}$ = Number of consecutive failures before resetting to incumbent.

$\bar{r}$ = Number of iterations at which we reset to the incumbent and recalculate the step size.

The typical values for these parameters are $N = 300$, $\bar{v} = 10$ and $\bar{r} = 75$.

All counters are set to 0 before beginning the solution procedure. Starting with the initial solution π^0 at $k = 0$, at any general iteration k , let $\theta(\pi^k)$ denote the solution of the Lagrangean problem given a particular dual vector π^k . This value is obtained by solving the Minimum Cost Network Flow subproblem using $\pi^k = (\alpha^k, \beta^k, \gamma^k)$. This solution also yields a subgradient ξ^k of $\theta(\cdot)$ at π^k , based on the coefficients of π in the objective function of the Lagrangean subproblem. If the norm of the subgradient $\|\xi^k\|$ is less than a preset value of ϵ , we stop with π^k as a near optimal solution with value $\theta(\pi^k)$.

If this is not the case, we check to see if $\theta(\pi^k)$ is greater than the present incumbent value. If so, the incumbent solution is reset to $\pi^* = \pi^k$ and $\theta^* = \theta(\pi^k)$. To compute the next point π^{k+1} we have to calculate a direction and a step length. The direction d^k is calculated by taking a combination of the subgradient and the direction at the previous point.

$$d^k = \xi^k + \delta_k d^{k-1}$$

where

$$\delta_k = \frac{-\Delta d^{k-1} \xi^k}{\|d^{k-1}\|^2} \quad \text{if } d^{k-1} \xi^k < 0$$

$$\delta_k = 0 \quad \text{otherwise}$$

The step length is calculated based on Bazaraa and Sherali [1981], as

$$s = \left[\frac{\bar{\theta} - \theta(\pi^k)}{\|d^k\|^2} \right] \left[\frac{1 - 0.5(n-1)\bar{r}}{(N - \bar{r})} \right]$$

The step length is periodically updated and resetting to the incumbent is also periodically performed based on the counter thresholds $\bar{v}$ and $\bar{r}$ as in Bazaraa and Sherali [1981].

The vector might not be in the feasible region. Therefore the vector is finally projected onto the feasible region D and this gives the new iterate. The process now continues until either $\|\xi^k\| \leq \varepsilon$ or $k \geq N$

6.4 Further Tightening of the Relaxation

Suppose that the continuous relaxation has been solved to (near) optimality via the Lagrangian dual. Corresponding to the optimal Lagrange multipliers, examine the Lagrangian subproblem, and recall that this is separable for each machine k . Consider some machine k for which the solution to this subproblem gives a subtour involving jobs in a set Q . Then we can impose a constraint

$$\sum_{i \in Q} \sum_{\substack{j \in Q \\ j \neq i}} x_{ij}^k \leq |Q| - 1$$

for each subtour, for each machine k . Now, one can add these constraints to the problem, dualize these using appropriate multipliers and perform additional dual ascent steps using subgradient optimization in order to improve the lower bound, and also improve the potential of deriving good quality upper bounds because of the improved representation of the problem.

In order to accelerate this ascent process, one can initialize the dual multiplier associated with each of the above constraints as

$$\theta = \begin{cases} -\frac{\Delta}{|Q|} & \text{if } \Delta < 0 \\ 0 & \text{otherwise} \end{cases}$$

where Δ is the cost of the subtour in the final subproblem solution. This choice of multipliers results in an increase of $-\frac{\Delta}{|Q|} > 0$ in the subproblem solution value for the currently optimal solution when $\Delta < 0$, and hence provides a useful advanced start for the Lagrangian dual problem.

7.0 Formulation and Solution of the Allocation Sub-problem

The allocation sub-problem is formulated as an integer (0-1) quadratic programming problem with linear and quadratic cost terms in the objective. The quadratic term is associated with the changeover cost. The linear cost terms are to account for the allocation cost. The quadratic programming problem is solved using an implicit enumeration scheme. Logical tests are used to fix variables and an updating strategy combined with strong Benders' cuts are used to give good lower bounds.

7.1 Allocation Problem Formulation

The Lagrangian dual multipliers are now used to derive the coefficients of a Strongest Surrogate Benders' cut. Denoting $\bar{\beta}_{ij}^k$ to be the dual variables obtained with respect to [7] and $\bar{u}_p^k$ and $-\bar{v}_p^k$ to be those associated with [5] and [6] after solving for the variables T_j in the Lagrangian dual subproblem, we obtain the coefficient $\bar{c}_j^k$ of y_j^k in the Benders' cut $z \geq [Constant] + \sum_k \sum_{j \in J_k} \bar{c}_j^k y_j^k$, as follows:

$$\bar{c}_j^k = \left[c_j^k + \sum_{\substack{i \in J_k \cup J_k \\ i \neq j}} \frac{\bar{\beta}_{ij}^k}{R_{ik}} (t_j^k - \tau_j^k) + \bar{u}_j^k - \bar{v}_j^k + \bar{Q}_j (T_{j_k} + t_j^k) \right] \quad [15a]$$

where $\bar{Q}_j = 0$ if $Q_j \leq 0$ in [14] and is given by that Q_j in [14] corresponding to $\bar{\beta}_{ij}^k$, otherwise.

The next step is the heuristic formulation of the allocation subproblem. The sequencing constraints are relaxed in this formulation. Therefore we incorporate some information on the sequencing constraints in the problem via the coefficients [15a] in the first term of the objective function. Hence [15a] can be used as an indicator of the cost of assigning job j to machine k , given the allocation and sequencing constraints in the problem.

The second term in the objective function is an approximation of the actual allocation and sequencing costs. If there are p jobs on a machine, then there will be $(p - 1)$ changeover costs between adjacent jobs. If we use the term $\sum_{i,j \in J_k} c_{ij}^k y_i^k y_j^k \forall i < j$, for the total changeover cost, it considers $\frac{p(p - 1)}{2}$ changeover costs. If there are a total of N jobs to be allotted on m machines, then there will be an average of $\frac{N}{m}$ jobs on every machine. Therefore, we will have an average of $(N/m - 1)$ changeovers on every machine. The $\sum_{i,j \in J_k} c_{ij}^k y_i^k y_j^k \forall i < j$, term considers $(N/2m)(N/m - 1)$ changeover costs. Therefore it is weighted with $\frac{2m}{N}$ to make it somewhat compatible with the actual changeover cost.

The $\sum_{j \in J_k} c_j^k y_j^k$ term is the allocation cost of all jobs assigned to machine k . A convex combination of the two terms based on the parameter $\lambda \in [0,1]$ allows for varying the

relative importance given to each term in the objective function. The resulting quadratic programming allocation problem (AQP) is as follows.

AQP :

$$\text{Min } \lambda \sum_k \sum_{j \in J_k} \bar{c}_j^k y_j^k + (1 - \lambda) \sum_k \left[\frac{2m}{N} \sum_{\substack{l, j \in J_k \\ l < j}} c_{lj}^k y_l^k y_j^k + \sum_{j \in J_k} c_j^k y_j^k \right] \quad [15b]$$

subject to

$$\sum_{k \in K_j} y_j^k = 1 \quad \forall j = 1 \dots n \quad [16]$$

$$\sum_{j \in J_k} t_j^k y_j^k \leq R_k \quad \forall k \quad [17]$$

$$\sum_{\substack{j \in J_k \\ j < l}} t_j^k [y_j^k + y_l^k - 1] + t_l^k y_l^k \leq d_l y_l^k \quad \forall l \in J_k \quad \forall k \quad [18]$$

$$0 \leq y_j^k \leq 1 \quad y_j^k \text{ binary} \quad \forall j, k \quad [19]$$

(Above, we can assume that if $j \in J_k$ but $d_j > R_k$ then $y_j^k = 0$. Hence, J_k is assumed to contain only those assignable jobs j for which additionally, $d_j \leq R_k$). Constraints [16] and [17] are the same as in the original formulation. Constraint [16] is to ensure

that a job is assigned to one machine only and constraint [17] is the load balancing constraint. Constraint [18] is added to ensure feasibility of the job assignments with respect to the due dates. It assumes that the jobs assigned to a machine are arranged in an Earliest Due Date (EDD) sequence. (For convenience, it is assumed that the jobs are indexed in the order of their due dates). This minimizes the maximum lateness of the jobs on the machine. Hence, an allocation of jobs to a machine will have a sequence feasible to the due dates if and only if [18] holds. The sequencing stage of the algorithm looks for a better sequence on the machine.

7.2 Allocation Problem Linearization

The allocation subproblem is a quadratic programming problem which can be solved by the algorithm of Adams and Sherali (1986). It uses a linearization technique which is shown to yield a tighter continuous relaxation than is available through other methods. An implicit enumeration algorithm which uses Lagrangian relaxation, Benders' cutting planes, and local exploration is designed to exploit the strength of the linearization.

We formulate a problem which is equivalent to the quadratic programming problem in an integer sense but is tighter in a continuous sense. For this purpose we multiply [17] by y_i^* and $(1 - y_i^*)$ for each $i \in J_k$. and writing $y_i^* y_j^*$ as w_{ij}^* , we get

$$\sum_{\substack{j \in J_k \\ j < i}} t_j^k w_{ji}^k + \sum_{\substack{j \in J_k \\ j > i}} t_j^k w_{ij}^k + (t_i^k - R_k) y_i^k \leq 0 \quad \forall i \in J_k \quad \forall k \quad [20]$$

$$\sum_{\substack{j \in J_k \\ j < i}} t_j^k w_{ji}^k + \sum_{\substack{j \in J_k \\ j > i}} t_j^k w_{ij}^k + (t_i^k - R_k) y_i^k \geq \sum_{j \in J_k} t_j^k y_j^k - R_k \quad \forall i \in J_k \quad \forall k \quad [21]$$

Equations [20] and [21] imply [17] in the continuous sense.

An alternate way of writing [18] is

$$\sum_{\substack{j \in J_k \\ j < i}} t_j^k y_j^k + y_i^k \left[\sum_{\substack{j \in J_k \\ j \leq i}} t_j^k - d_i \right] - \sum_{\substack{j \in J_k \\ j < i}} t_j^k \leq 0 \quad \forall i \in J_k \quad \forall k \quad [22]$$

Multiplying each of the equations [22] by y_i^k and $(1 - y_i^k)$ for each $i \in J_k$, we obtain

$$\sum_{\substack{j \in J_k \\ j < i}} t_j^k w_{ji}^k + t_i^k y_i^k + \sum_{\substack{j \in J_k \\ i < j < l}} t_j^k w_{ij}^k + w_{il}^k \left[\sum_{\substack{j \in J_k \\ j \leq l}} t_j^k - d_l \right] - y_i^k \sum_{\substack{j \in J_k \\ j < i}} t_j^k \leq 0 \quad [23a]$$

and

$$\sum_{\substack{j \in J_k \\ j < i}} t_j^k w_{ji}^k + t_i^k y_i^k + \sum_{\substack{j \in J_k \\ i < j < l}} t_j^k w_{ij}^k + w_{il}^k \left[\sum_{\substack{j \in J_k \\ j \leq l}} t_j^k - d_l \right] - y_i^k \sum_{\substack{j \in J_k \\ j < i}} t_j^k \geq$$

$$\sum_{\substack{j \in J_k \\ j < i}} t_j^k y_j^k + y_i^k \left[\sum_{\substack{j \in J_k \\ j \leq l}} t_j^k - d_l \right] - \sum_{\substack{j \in J_k \\ j < i}} t_j^k \quad \forall i \in J_k, i \in J_k, \text{ with } i < l \quad \forall k \quad [23b]$$

and we obtain the following sets of constraints.

$$\sum_{\substack{j \in J_k \\ j < l}} t_j^k w_{jl}^k + w_{li}^k \left[\sum_{\substack{j \in J_k \\ j \leq l}} t_j^k - d_l \right] - y_i^k \sum_{\substack{j \in J_k \\ j < l}} t_j^k \leq 0 \quad [24a]$$

and

$$\sum_{\substack{j \in J_k \\ j < l}} t_j^k w_{jl}^k + w_{li}^k \left[\sum_{\substack{j \in J_k \\ j \leq l}} t_j^k - d_l \right] - y_i^k \sum_{\substack{j \in J_k \\ j < l}} t_j^k \geq$$

$$\sum_{\substack{j \in J_k \\ j < l}} t_j^k y_j^k + y_l^k \left[\sum_{\substack{j \in J_k \\ j \leq l}} t_j^k - d_l \right] - \sum_{\substack{j \in J_k \\ j < l}} t_j^k \quad \forall l \in J_k, \forall i \in J_k, \text{ with } i > l \quad \forall k \quad [24b]$$

and finally, we obtain the following sets of constraints as well:

$$\sum_{\substack{j \in J_k \\ j < l}} t_j^k w_{jl}^k + y_l^k [t_l^k - d_l] \leq 0 \quad [25a]$$

and

$$\sum_{\substack{j \in J_k \\ j < l}} t_j^k w_{jl}^k + y_l^k [t_l^k - d_l] \geq$$

$$\sum_{\substack{j \in J_k \\ j < l}} t_j^k y_j^k + y_l^k \left[\sum_{\substack{j \in J_k \\ j \leq l}} t_j^k - d_l \right] - \sum_{\substack{j \in J_k \\ j < l}} t_j^k \quad \forall l \in J_k \quad \forall k \quad [25b]$$

We enforce $w_{ij}^k = y_i/y_j^k$ with the following constraints.

$$w_{ij}^k \leq y_i^k \quad \forall i, j \in J_k, \quad i < j \quad \forall k \quad [26a]$$

$$w_{ij}^k \leq y_j^k \quad \forall i, j \in J_k, \quad i < j \quad \forall k \quad [26b]$$

$$w_{ij}^k \geq y_i^k + y_j^k - 1 \quad \forall i, j \in J_k, \quad i < j, \quad \forall k \quad [26c]$$

$$w_{ij}^k \geq 0 \quad \forall i, j \in J_k, \quad i < j \quad \forall k \quad [26d]$$

$$0 \leq y_i^k \leq 1 \quad \forall i, j \in J_k, \quad i < j \quad \forall k \quad [26e]$$

$$0 \leq y_j^k \leq 1 \quad \forall i, j \in J_k, \quad i < j \quad \forall k \quad [26f]$$

This gives the following equivalent mixed integer program MIP to the allocation problem [15]-[19].

MIP :

$$\text{Min} \quad \lambda \sum_k \sum_{j \in J_k} \bar{c}_j^k y_j^k + (1 - \lambda) \sum_k \left[\frac{2m}{N} \sum_{\substack{i, j \in J_k \\ i < j}} c_{ij}^k w_{ij}^k + \sum_{j \in J_k} c_j^k y_j^k \right]$$

Subject to

[16]

[27]

[20] and [21]

[28]

[23] , [24] and [25] [29]

[26] [30]

y binary [31]

The number of constraints in the MIP is as follows:

n constraints in [27]

$2 \sum_k |J_k|$ in [28]

$2 \sum_k |J_k| |J_k|$ in [29]

$\frac{3}{2} \sum_k [|J_k|^2 - |J_k|]$ in [30]

Note that the constraints [30] are simple variable bounding type of constraints and can be handled very easily in the solution algorithm. The following section discusses the solution procedure for the Quadratic Programming problem by implicit enumeration.

7.3 Implicit Enumeration Algorithm for the Linearized Allocation Problem

An implicit enumeration algorithm is used for the solution of the linearized problem MIP. In the implicit enumeration scheme we do a depth first tree enumeration on the y_j^* variables. The strategy at every node is to first fix some of the variables at 0/1 and or fathom the node based on the lower bound. If this is not possible, then we try and improve the incumbent solution and branch.

The implicit enumeration scheme is described below:

STEP 0A. Obtain a heuristic solution.

STEP 0B. Solve the continuous relaxation $\overline{\text{MIP}}$ of MIP using Lagrangian relaxation by dualizing [28] and [29] and [26b] and [26c] from [30]. Hence, derive a strongest surrogate Benders' cut and possibly update the heuristic incumbent solution from this solution.

The following steps are repeated at each node which is analyzed in the implicit enumeration scheme.

STEP 1. Derive an updated Lagrangean solution and corresponding strongest surrogate Benders' cut.

STEP 2. Perform logical tests to possibly fix variables at 0/1. If any additional variables are fixed return to STEP 1.

STEP 3. Compute a Lower Bound (LB1) based on the updated SS Benders' cut. Update the incumbent, if possible, from this LB1 calculation. If the lower bound is greater than the incumbent, fathom the node.

STEP 4. Derive a Benders' cut based on the solution from the LB1 computation from STEP 3. Perform logical tests on this cut, and if any variables are fixed at 0/1, return to STEP 1.

STEP 5. Calculate LB2 based on the Benders' cut. If the lower bound is greater than the incumbent then fathom this node.

STEP 6. Compute a surrogate of the (near) Strongest Surrogate and the newly derived Benders' cut. Perform logical tests on this cut. If any variables are fixed at 0/1, return to STEP 1. Based on this, derive a Lower Bound LB3. If this lower bound is greater than the incumbent solution then fathom the node. Also, update the incumbent solution from the derivation of LB3 if possible.

STEP 7. Choose a suitable branching variable and branch.

Details of the above steps are given below.

STEP 0A

7.3.1 Heuristic Solution

Two heuristics have been developed for the allocation problem. The first heuristic (Allocation Heuristic 1) operates as follows. To begin with, a list is prepared for each machine consisting of jobs whose cost of assignment is below a certain given value. The algorithm goes through the machines sequentially and allots a job from this list for the machine under consideration, after checking for feasibility with respect to due dates and load balancing constraints. The objective function for this assignment is

the same as the objective function of the quadratic programming algorithm. By the nature of the assignment scheme, a load balanced schedule results.

In the second heuristic (Allocation Heuristic 2) the cost of assigning a job on a machine is taken as the sum of the assignment cost for the particular job-machine combination and the average changeover cost between the job and the jobs already on the machine. This cost A_{ik} is hence given by

$$A_{ik} = (1 - \lambda) \left[c_i^k + \frac{\sum_{j \in J_k^+} \bar{c}_{ij}^k}{|J_k^+|} \right] + \lambda \bar{c}_i^k \quad \forall i, k$$

where,

$$\begin{aligned} \bar{c}_{ij}^k &= c_{ij}^k \quad \text{if } d_i < d_j \\ &= \frac{c_{ij}^k + c_{ji}^k}{2} \quad \text{if } d_i = d_j \end{aligned}$$

and $|J_k^+|$ is defined as in [37] (tentatively for this heuristic), for any partial solution constructed this far. The second term is the modified allocation cost as in [15a]. This cost takes the sequencing aspect also into consideration, by virtue of its derivation. Next a penalty cost P_j is computed for each job j as the difference between the cost A_{jl} on that machine (l) for which the total assignment cost for job j is a minimum, and the cost A_{jm} on that machine (m) which has the next higher assignment cost. This gives

$$P_j = A_{jl} - A_{jm}$$

The job (J) for which this penalty cost is the highest is selected and is assigned to that machine $k = K$ for which the cost A_{jk} is the lowest. Hence, we determine

$$J \in \text{Arg} \max_j P_j$$

and

$$K \in \text{Arg} \min_{k \in K_J} A_{Jk}$$

Before job J is assigned to machine K , it is checked for due date and machine load balancing constraint feasibility. Failing this, an assignment to the next cheaper machine is made. If the assignment is found feasible, then job J is assigned on machine K . The assignment cost of all the other jobs on machine K is updated as follows. If the assignment cost for some job i on machine K is A_{ik} before the assignment of job J on machine K , the updated cost A'_{ik} after the assignment is given by

$$A'_{ik} = A_{ik} - \left[\frac{T}{|J_k^+|} - \frac{T + \bar{c}_{ij}^k}{|J_k^+| + 1} \right] (1 - \lambda)$$

where

$$T = \left[\frac{A_{ik} - \lambda \bar{c}_i^k}{(1 - \lambda)} - c_i^k \right] |J_k^+|$$

J_k^+ is updated to $J_k^+ \cup J$ and job J is removed from the list of available jobs.

Load balancing can be incorporated in the algorithm by specifying a load balancing constraint. The schematic diagram of the algorithm is shown in Figure 2.

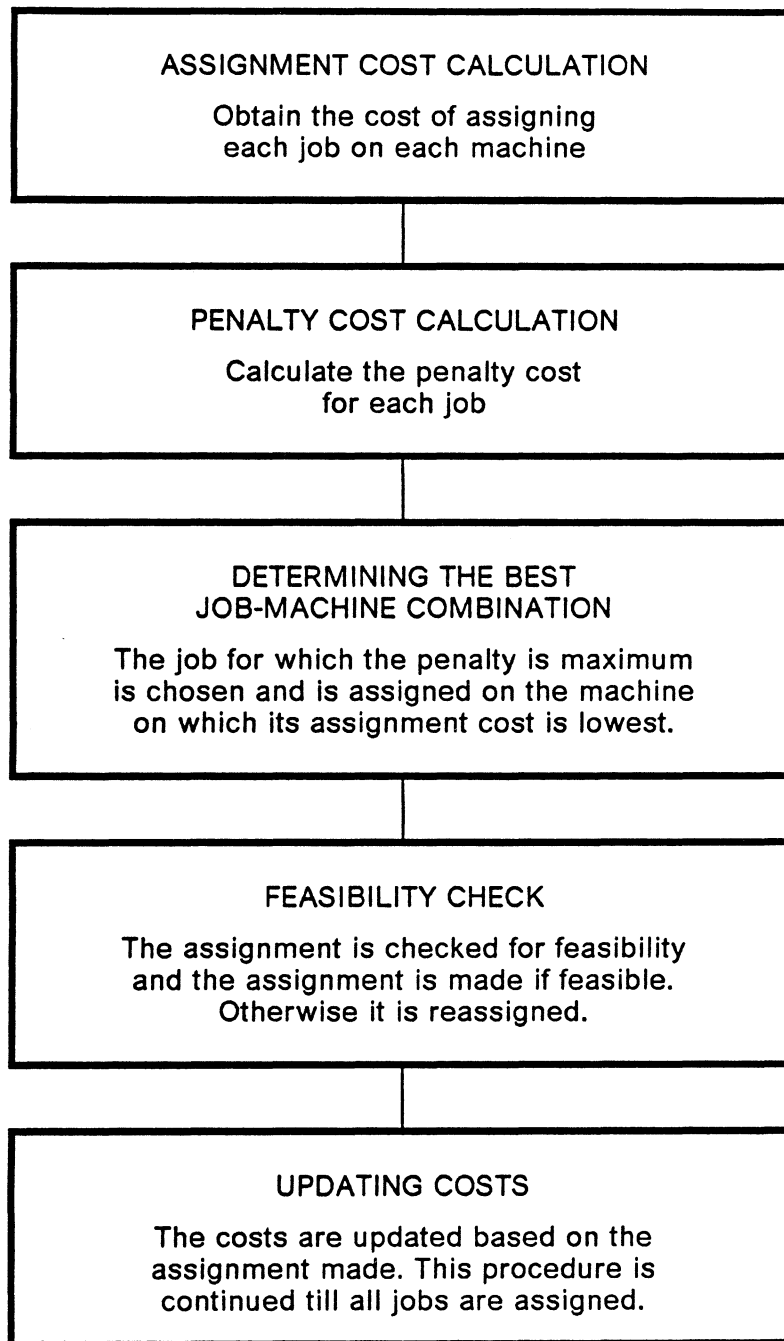


Figure 2. Allocation Heuristic 2

7.3.2 Lagrangian Relaxation of MIP

Dualizing [28], [29] and [26b] and [26c] from [30] gives a Lagrangian dual of maximizing a nonnegatively constrained, concave objective function. This can be optimized using the subgradient optimization procedure described in section 6.3. The Lagrangian subproblem in this case, which evaluates the Lagrangian objective function, takes the form:

$$\text{Min} \left[K + \sum_k \sum_{j \in J_k} F_j^k y_j^k + \sum_k \sum_{i \in J_k} \sum_{\substack{j \in J_k \\ i < j}} G_{ij}^k w_{ij}^k \right]$$

$$\sum_{k \in K_j} y_j^k = 1 \quad \forall j = 1 \dots n \quad [32]$$

$$0 \leq w_{ij}^k \leq y_i^k \quad \forall i, j \in J_k \text{ with } i < j \quad \forall k \quad [33]$$

$$y_j^k \text{ binary} \quad [34]$$

where K is a constant, and F_j^k and G_{ij}^k are appropriate coefficients in the Lagrangian function. This problem is equivalent to

$$\text{Min} \left[K + \sum_k \sum_{j \in J_k} F_j^k y_j^k + \sum_k \sum_{i \in J_k} \sum_{\substack{j \in J_k \\ i < j}} \text{Min}(0, G_{ij}^k) y_i^k \right] \quad [35a]$$

Subject to

$$[32] \text{ and } [34]$$

where we have used the fact that if G_{ij}^k is greater than or equal to 0, then w_{ij}^k will be 0 and if G_{ij}^k is less than 0 then w_{ij}^k will be equal to y_j^k in [33]. Equation [35a] can be rewritten as

$$\text{Min} \left[K + \sum_k \sum_{j \in J_k} \left[F_j^k + \sum_{\substack{l \in J_k \\ l > j}} \text{Min}(0, G_{jl}^k) \right] y_j^k \right] \quad [35b]$$

Subject to

$$[32] \text{ and } [34]$$

Note that constraint [34] can be equivalently replaced with

$$0 \leq y_j^k \leq 1$$

so that the subproblem is easy to solve. Once this Lagrangean dual is solved and the near optimal dual multipliers obtained are substituted in the Lagrangian function to derive the appropriate coefficients K^* , F^* and G^* in [35b] corresponding to the final subproblem, one can derive the Strongest Surrogate Benders' cut as

$$z \geq K^* + \sum_k \sum_{j \in J_k} \left[F_j^{*k} + \sum_{\substack{l \in J_k \\ l > j}} \text{Min}(0, G_{jl}^{*k}) \right] y_j^k \quad [36]$$

This Strongest Surrogate cut may be used to calculate the bounds in the implicit enumeration scheme.

STEP 1.

7.3.3 Updating the Strongest Surrogate Cut

Whenever some variables are fixed at 0/1, the cut [36] needs to be updated instead of being recalculated. For this purpose, define the following.

$$\begin{aligned}
 J_k^+ &= \{ j : y_j^k = 1 \text{ fixed} \} \\
 J_k^- &= \{ j : y_j^k = 0 \text{ fixed} \} \\
 J_k^0 &= \{ j : y_j^k \text{ is free} \}
 \end{aligned} \tag{37}$$

Note that:

If $y_j^k = 0$ then $w_{ij}^k = 0$ for $i < j$ and $w_{ji}^k = 0$ for $i > j$

If $y_j^k = 1$ then $w_{ij}^k = y_i^k$ for $i < j$ and $w_{ji}^k = y_i^k$ for $i > j$

Therefore the MIP can be reduced by fixing the variables that have been fixed at zero or one as well as fixing w_{ij}^k at y_j^k as the case may be. Equations [21] and [22] are written only for $i \in J_k^0$ and equations [23] - [25] are written for each $i \in J_k^+$ and each $i \in J_k^0$ and [26] is written for each $i, j \in J_k^0$, $i < j$ for each machine k .

We associate the same multipliers with these surviving constraints as with the original constraints which were used to derive [36], and we set up the Lagrangian relaxation subproblem as before but now with the updated coefficients $\hat{K}$, $\hat{F}$ and $\hat{G}$. Solving this subproblem updates the constraint [36] to yield

$$z \geq \hat{K} + \sum_k \sum_{j \in J_k^0} \left[\hat{F}_j^k + \sum_{\substack{i \in J_k^0 \\ i > j}} \text{Min}(0, \hat{G}_{ji}^k) \right] y_j^k \quad [39]$$

We use the above updated constraint for logical tests and bound calculations.

STEP 2

7.3.4 Logical Tests at the Nodes

The following logical tests are performed at each node.

1. Do standard logical tests based on the objective function using the updated Strongest Surrogate Benders' cut [39].
2. If for some job j , y_j^k is set equal to 1, then y_l^k is set equal to 0 for all $l \neq j$. Conversely, if [16] has only a single free variable for any j , that y_j^k is set at one. (If $y_j^k = 0$ for all $k \in K_j$, then the node is fathomed).
3. For each machine the following test is done to check if the inclusion of any job which can be assigned on the machine will violate the load balancing constraint. If J_k^+ is defined as in [37] then from [17],

$$\sum_{j \in J_k^+} t_j^k + t_p^k > R_k$$

then, $y_p^k = 0$ for each free y_p^k $p \in J_k$, $\forall k$.

4. The fourth logical test is to determine if the allocation of some job to a machine will result in the violation of the due date constraint. If the due date is violated then the problem is infeasible. Let J_k^+ denote the jobs already fixed on machine k . A job

p can be allotted on machine k if and only if all the jobs in the EDD sequence of $J_k^+ \cup p$ meet their due dates. If one of the due dates is violated in the EDD sequence of $J_k^+ \cup p$ then y_p^k is set at 0.

STEP 3.

7.3.5 Calculation of Lower Bound LB1

The lower bound LB1 is calculated as follows : Based on the updated version of the Benders' cut [39], compute

$$LB1 = \text{Min } \hat{K} + \sum_k \sum_{j \in J_k^0} \left[\hat{F}_j^k + \sum_{\substack{i \in J_k^0 \\ i > j}} \text{Min}(0, \hat{G}_{ji}^k) \right] y_j^k \quad [40]$$

Subject to

[16] over the free variables, and y_j^k binary

Let $\bar{y}$ solve [42]. Update the incumbent using $\bar{y}$, if possible.

STEP 4

7.3.6 Benders' cut Based on the Solution from LB1

Let $\bar{y}$ be the solution obtained from the LB1 computation in [40]. The following Benders' cut is constructed from this solution $\bar{y}$. Define the following :

$$\bar{J}_k = \{ (i,j) \in J_k \mid i < j : [\bar{y}_i^* + \bar{y}_j^* = 2] \text{ or } [\bar{y}_i^* + \bar{y}_j^* = 1]$$

and the i or j for which the y variable is 1 belongs to $J_k^+ \} \quad [41]$

Then the following valid Benders' cut may be derived as in Adams and Sherali [1986].

$$\begin{aligned} z \geq & \sum_k \sum_{j \in J_k} [\lambda \bar{c}_j^* + (1 - \lambda) c_j^*] y_j^* \\ & + \frac{2m}{N} (1 - \lambda) \sum_k \sum_{(i,j) \in \bar{J}_k} c_{ij}^k [y_i^k + y_j^k - 1] \end{aligned} \quad [42]$$

STEP 5.

7.3.7 Computation of Lower Bound LB2

The second lower bound is computed based on the Benders' cut [42] as follows.

$$\begin{aligned} LB2 = \text{Min } & \sum_k \sum_{j \in J_k} [\lambda \bar{c}_j^* + (1 - \lambda) c_j^*] y_j^* \\ & + \frac{2m}{N} (1 - \lambda) \sum_k \sum_{(i,j) \in \bar{J}_k} c_{ij}^k [y_i^k + y_j^k - 1] \end{aligned} \quad [43]$$

Subject to

$$\sum_{k \in K_j} y_j^k = 1 \quad \forall j = 1 \dots n$$

$$y_j^k \text{ binary } \forall j, k$$

STEP 6

7.3.8 Surrogation of [39] and [42] and LB3

A convex combination of the near Strongest Surrogate Benders' cut [39] and the Benders' cut [42] is derived based on the weight

$$\varphi = \frac{LB1}{LB1 + LB2}$$

Namely, the convex combination of [39] and [42] is obtained using weights of φ and $(1 - \varphi)$ respectively. Let the resulting equation be

$$z \geq K_\varphi + \sum_k \sum_{j \in J_k^0} c_j^{\varphi k} y_j^k \quad [44]$$

The third Lower Bound LB3 is now found as

$$LB3 = \text{Min } K_\varphi + \sum_k \sum_{j \in J_k^0} c_j^{\varphi k} y_j^k \quad [45]$$

Subject to

$$\sum_{k \in K_j} y_j^k = 1 \quad \forall j = 1 \dots n$$

$$y_j^k, \text{ binary } \forall j, k$$

If $\tilde{y}$ solves [45], update the incumbent solution in case $\tilde{y}$ is feasible and improving.

STEP 7

7.3.9 Branching Variable Choice

For branching, we pick the free job with the smallest $|K_j|$ cardinality. Each time some y_j^k is set at zero the value of $|K_j|$ is decreased by 1. In case of ties we pick the job j which has the smallest objective coefficient in [44] corresponding to that k for which $\bar{y}_j^k$ is 1 in the solution of [45]. Persisting ties in this case are broken arbitrarily. Depth first branching is done by fixing $\bar{y}_j^k = 1$

8.0 The Sequencing Sub-problem

After an allocation of jobs on machines is complete, the jobs on each individual machine have to be sequenced. This sequencing subproblem is separable over the machines, and is formulated as a travelling salesman problem with side constraints for each machine. The formulation of this problem is given below for any machine; hence, the machine superscript k has been suppressed.

$$\text{Min } \sum_i \sum_j c_{ij} x_{ij}$$

Subject to

$$\sum_i x_{ij} = 1 \quad \forall j \quad [46]$$

$$\sum_j x_{ij} = 1 \quad \forall i \quad [47]$$

$$T_j \geq T_i + (t_{ij} + t_j) x_{ij} + R_{ij} (x_{ij} - 1) \quad \forall i \neq j, \forall j \neq 1 \quad [48]$$

$$T_1 + t_j + \sum_{i \neq j} (t_i + t_{ij}) x_{ij} \leq T_j \leq d_j \quad [49]$$

$$0 \leq x_{ij} \leq 1 \quad x_{ij} \text{ binary} \quad [50]$$

where

$$R_{ij} = d_i - \min_{p \neq j} [T_1 + t_{1p} + t_{pj} + t_j] \quad \forall i, j \quad i \neq j \quad [51]$$

Constraints [46] and [47] are standard assignment type constraints. Constraint [48] is to ensure that the completion time of a job (j) on a machine is greater than or equal to the sum of the completion time of all the previous jobs on the machine, the changeover time and the processing time of the job. This is enforced if x_{ij} is equal to 1, ie. job i is processed just before job j. If x_{ij} is equal to zero then the constraint is essentially redundant by virtue of [51], assuming that the changeover costs satisfy the triangle inequality. Constraint [49] is a lower-upper bound type of constraint on the completion time of a job, based on its early finish and late finish times, respectively. Note that job 1 in the above formulation is assumed to be the one corresponding to the initial setting of the machine.

The above problem is solved using an implicit enumeration scheme. An initial upper bound is obtained using a heuristic algorithm. At every node logical tests are performed to fix variables. These tests are described in the following section. Then an assignment problem is solved over the free variables to get a lower bound at the node. The solution is checked for feasibility. If the solution is feasible and better than the current upper bound, then the upper bound is updated. The solution to the assignment problem may be infeasible because of the existence of subtours or because of due date violations. Those infeasibilities are used to further increase the lower bound using additional cutting planes. The schema of the algorithm is stated below, and details of the different steps of the algorithm are given subsequently.

8.1 *Implicit Enumeration Algorithm for the Sequencing Sub-problem*

STEP 0A. Obtain a heuristic incumbent solution.

STEP 0B. Solve the continuous relaxation of the problem via Lagrangian Duality. Hence derive a lower bound on the problem and a Strongest Surrogate Benders' cut.

STEP 1. At any general node in the branch and bound enumeration tree, first perform appropriate specialized logical tests in order to possibly fix some x_{ij} variables at 0/1.

STEP 2. Compute a lower bound for the node based on the Strongest Surrogate Benders' cut. Tighten this bound by updating the cut using additional valid inequalities. Fathom the node if the bound exceeds the incumbent solution value. Also, update the incumbent if any improving, feasible solution is found in this process.

STEP 3. Choose a branching variable and return to STEP 1.

STEP 0A

8.1.1 Heuristic solution

The heuristic algorithm (Pairwise Interchange Heuristic) used for solving the sequencing subproblem is a savings-improvement exchange type of algorithm. In this algorithm, we start off with a feasible schedule and jobs are switched pairwise to obtain improving, feasible solutions.

The jobs are initially arranged in an EDD sequence. If the due date of any job is violated, the heuristic fails. (Note that this will not occur if the allocation subproblem has passed on a feasible solution). If all the due dates are met, this schedule acts as the initial feasible schedule on which improvements are made. The savings in switching any pair of jobs is defined as the difference in cost between the total changeover cost in the existing line up of jobs and the total changeover cost after the pair of jobs are switched. A savings matrix is created whose (symmetric) entry (s_{ij}) is the savings obtained by switching jobs i and j . Let $p(j)$ and $s(j)$ denote the preceding and succeeding job corresponding to job j in the current sequence. If k and l are non adjacent jobs then the savings from switched jobs k and l will be

$$s_{kl} = c_{p(k)k} + c_{ks(k)} + c_{p(l)l} + c_{ls(l)} - (c_{p(k)l} + c_{ls(k)} + c_{p(l)k} + c_{ks(l)})$$

If k and l are adjacent jobs and $k = p(l)$, then the savings will be

$$s_{kl} = c_{p(k)k} + c_{kl} + c_{ls(l)} - (c_{p(k)l} + c_{lk} + c_{ks(l)})$$

The pair of jobs which gives the maximum exchange savings subject to the due date or load balancing constraints, is selected and the switch is made if this savings is positive. The savings matrix is updated and the same procedure is repeated until no more improving, feasible switches are possible.

STEP 0B

8.1.2 Lagrangian Relaxation at Node Zero

A Lagrangian relaxation is done at node 0 to determine a lower bound and a (near) Strongest Surrogate Benders' cut. Constraint (48) is dualized in the Lagrangian dual formulation. To solve the resulting subproblem, if the coefficient of T_j is greater than or equal to 0 then the value of T_j is set at $T_1 + t_j + \sum_{i \in J} (t_i + t_{ij}) x_{ij}$ and if it is less than 0 it is set at d_j . The resulting problem now has an assignment structure and can be solved easily. A set of (near) optimal dual multipliers are obtained using a subgradient optimization scheme as in Section 4.2.1. Using the (near) optimal dual multipliers, a (near) Strongest Surrogate Benders' cut is derived in terms of the x-variables in the same spirit as [36] and [39] above.

STEP 1

8.1.3 Logical Tests

The following logical tests are performed at each node possibly to fix variables at 0/1. These are repeated until no additional variables are fixed in one complete pass.

1. Perform the following objective based logical tests on the Strongest Surrogate Benders' cut. A weak lower bound can be calculated via a dual feasible solution for the assignment problem over [46],[47] and [50] using the Strongest Surrogate cut expression as the objective cost matrix. The minimum in each row is subtracted from

each row, and this is repeated with the columns to derive this dual solution. If this lower bound is greater than the upper bound then the node is fathomed. Furthermore, examining the highest cost in each row of the reduced assignment cost matrix, if setting the variable corresponding to this cost equal to 1 makes the lower bound greater than the upper bound, the variable is fixed at 0.

2. Check for singleton assignments. If x_{ij} has been set equal to 0 for all $j \neq k$, then set x_{ik} equal to 1.

3. Trace the string of fixed assignments from job 1. Let the last job in this string be job p . For all jobs j not in this string, the reverse string is traced from that job j . Let the first job in the reverse string be q . If setting x_{pq} equal to one violates the due date of any job in the string, then the node is fathomed. Further, if any job in the string has its finish time equal to its due date when x_{pq} is set to 1, then x_{pq} can be fixed equal to 1. Note that if x_{pq} is fixed equal to zero in the current partial solution, then the computed lower bound on the finish of each job can be further tightened and checked against the due date for possible fathoming.

4. Again trace the string from job 1. Let the last job in the string be p . We now trace all the other strings. Let the first job in a particular string be q . We try to introduce another job string starting at r and ending at s between p and q (possibly $r = s$). If setting x_{pr} equal to 1 and x_{sq} equal to one violates any of the due dates in the second string, both x_{pr} and x_{sq} are set equal to 0. (In this case we assume that the triangle inequality holds).

5. Arrange all strings in an EDD sequence, based on the due date of the last jobs in the strings. If this arrangement violates the due date of any of the last jobs in the strings, then the node is fathomed due to infeasibility.

STEP 2

8.1.4 Lower Bound Calculation

A lower bound is calculated at each node. First all the x_{ij} variables that are fixed at either 0 or 1 are removed from the problem by factoring them out as constants. Whenever a job lies on the string from job 1 to that job, the value of T_j for that job is known and is substituted in the formulation. The value of R_{ij} is updated as variables are fixed at 0/1. Constraint [46] is dualized with the optimal dual multipliers. This results in a problem which can be solved sequentially, first over the T_j variables and then over the x_{ij} variables. The value of T_j is set equal to $T_1 + t_j + \sum_{i \neq j} (t_i + t_{ij}) x_{ij}$ if the coefficient of T_j is greater than or equal to 0, and is set equal to d_j if it is less than 0. This results in an assignment problem over the x_{ij} variables. This is solved using the Out Of Kilter algorithm to give a lower bound on the node problem. If this lower bound is greater than the incumbent solution, then the node is fathomed. If the solution is feasible and better than the current incumbent solution, then the incumbent is updated. If the out of kilter algorithm does not give a feasible solution, then additional valid inequalities are used to increase the value of the lower bound at the node as follows:

The out of kilter algorithm need not give a feasible solution to the problem due to two reasons.

1. Existence of subtours in the optimal out of kilter solution.
2. The violation of the due date of one or more jobs in the optimal Out Of Kilter solution.

8.1.4.1 Subtour elimination

The optimal solution to the assignment problem gives a lower bound on the solution at the node. If this solution has subtours, the lower bound can be increased further. A dual ascent is done on the Lagrangian relaxation by introducing and dualizing additional valid inequalities. Suppose the out-of-kilter solution gives a subtour containing jobs $j_1, j_2, \dots, j_n$. The following constraint is introduced to break this subtours.

$$x_{j_1 j_2} + x_{j_2 j_3} + \dots + x_{j_{n-1} j_n} + x_{j_n j_1} \leq n - 1 \quad [52]$$

We can write one such constraint for every subtour. In order to determine the minimum increase in the objective value due to the introduction of this constraint can be obtained by solving the knapsack problem

$$\bar{K} + \text{Min} \sum_i \sum_j \bar{c}_{ij} x_{ij} \quad [53]$$

Subject to

$$x_{j_1 j_2} + x_{j_2 j_3} + \dots + x_{j_{n-1} j_n} + x_{j_n j_1} \leq n - 1$$

x binary

where $\bar{c}_{ij}$ are the reduced cost coefficients obtained from dualizing [46]-[49] using the current dual multipliers, and $\bar{K}$ is the associated constant. This problem can be rewritten in a different form by taking into consideration the fact that

$$\sum_i x_{ij} = 1 \quad \forall j$$

The subtour breaking constraint can then be rewritten as

$$\sum_{k \neq j_1} x_{kj_2} + \sum_{k \neq j_2} x_{kj_3} + \dots + \sum_{k \neq j_{n-1}} x_{kj_n} + \sum_{k \neq j_n} x_{kj_1} \geq 1 \quad [54]$$

The subtour breaking constraint can also be rewritten making use of the fact that

$$\sum_j x_{ij} = 1 \quad \forall i$$

The subtour breaking constraint then becomes

$$\sum_{k \neq j_2} x_{j_1k} + \sum_{k \neq j_3} x_{j_2k} + \dots + \sum_{k \neq j_{n-1}} x_{j_{n-1}k} + \sum_{k \neq j_1} x_{j_nk} \geq 1 \quad [55]$$

The two knapsack problems obtained by using [54] or [55] in lieu of [52] in [53] may be solved and the maximum of the two values gives a valid increase in the objective function due to the elimination of the subtour.

This is done for each of the subtours, and if only [54] or [55] is used so that the constraints for the different subtours remain separable, then the sum of the increase for each of the subtours gives the increase in the objective function due to the elimination of the subtours. An alternative tightening of the bound due to the breaking of the subtour is to pick the minimum from the row and column of each cell which forms part of the subtour. For each cell the maximum of these two values gives the increase in the objective function if this cell is removed from the subtour. This is done for each cell in the subtour and the minimum of these values gives the increase in objective function due to the breaking of the subtour. An equivalent interpretation of this is in terms of adding implied inequalities as above is evident.

8.1.4.2 Due Date Violation Constraint Generation

Suppose that the solution given by the out of kilter algorithm violates the due dates of some jobs, but does not contain any subtours. We trace the string of assignments from job 1 to the other jobs in the tour ordered as $j_1, j_2, \dots, j_n$ where $j_1 = 1$. Then, the following cut can be introduced into the problem.

$$x_{j_1/j_2} + x_{j_2/j_3} + \dots + x_{j_{n-1}/j_n} + x_{j_n/j_1} \leq n - 3$$

This cut makes certain that the same solution does not occur again, which is a necessary condition for a solution to satisfy the due date of a job. This is due to the fact that at least two jobs have to be interchanged in order to get a solution feasible to the due dates. If two jobs are interchanged, at least three arcs in the present solution will be changed. By the same reasoning as in the previous case, we obtain cuts [54] or [55], but with 3 instead of 1 in the right hand side. Hence taking the sum of the three minimum values from the cost matrix, aside from the cells corresponding to the above assignment tour, we obtain the resulting increase in the lower bound. If more than one subtour is considered, the largest increase over the subtours can be used as the required objective function penalty.

9.0 Solving Large Problems via the Heuristic Algorithms

The exact allocation and sequencing algorithms can be used for small sized problems. For medium and large sized problems a combination of exact and heuristic procedures can be used to obtain good solutions. This procedure is described below:

STEP 1:

Use the Allocation Heuristic 2 (Section 7.3.1) to give an allocation of the jobs to the machines. (Set the parameter λ to zero to eliminate the Lagrangean relaxation based cost term from consideration.)

STEP 2: (Overlapping Batch Exchange Sequencing Procedure)

Execute this step if the number of jobs on each machine is less than about 15-20 jobs. For larger problems skip this step and go directly to STEP 3.

Order the jobs allocated on every machine in an EDD sequence. Use the exact sequencing algorithm to arrange the first 10 jobs from the EDD sequence. Take the last 5 jobs from this arranged sequence and jobs 11 to 15 from the EDD sequence, and run the exact sequencing algorithm on this batch of jobs, keeping the first job of the batch fixed.

Repeat this process of executing the exact sequencing algorithm in an overlapping fashion till all the jobs have been sequenced. (The last batch over which the exact

sequencing algorithm is solved might involve less than 10 jobs. The number of overlapping jobs can be varied between about 3 and 7 depending on the size of problem being solved.)

STEP 3:

If STEP 2 has been executed, then use the output from STEP 2 as the starting solution for the Pairwise Interchange Heuristic (Section 8.1.1) to give an improved feasible schedule.

If STEP 2 has been skipped, then execute the Pairwise Interchange Heuristic using the EDD sequence as the initial sequence.

10.0 Results and Conclusions

In this chapter, we present sample results of our experience with the allocation and sequencing algorithm. The code for the algorithms was written in Pascal. The testing of the algorithms was done on an IBM 3090 computer. The testing was done in two phases. In the first phase, the algorithms were tested on actual data collected on the shop floor at the Standard Register plant. In the second phase, the algorithms were tested on randomly generated problems.

10.1 Results from runs on actual plant data

The first phase of testing was done with the data collected at the presses at the Standard Register plant over a period of two weeks. The data included the job number, time of entry of the job into the system, processing time of the job, due date of the job, press on which the job was assigned, collator on which the job was assigned, alternate presses and collator on which the jobs could be assigned, job characteristics such as color of form, perforation requirements, paper type and quality, numbering, collating instructions etc. Information was also provided on the expected wastage of paper during the changeover between jobs having different characteristics. This information was provided as the number of standard impressions wasted during the changeover. To implement the algorithm, the

changeover cost used was the weighted average of the paper wastage cost and a cost equivalent of the changeover time.

There were a total of 245 jobs over the two week period on all the presses. The allocation / sequencing algorithms were tested on the set of 77 jobs that were assigned to the 22" presses during the two week period. No other jobs could have been assigned to the 22" presses during these 2 weeks and these 77 jobs could not have been assigned to any other type of press. Therefore the allocation and sequencing of these 77 jobs is a problem which can be solved separately.

These 77 jobs were scheduled on 4 presses. The actual cost incurred in the scheduling of these jobs was calculated based on the estimated changeover cost between pairs of jobs. (Since the runs typify the remainder of the problem the presses and the collators, we present only these results for illustrative purposes.)

The following runs were conducted :

1. Resequencing the actual schedule

In the first run, the allocation of jobs on the 4 presses was assumed to be the same as in the actual implemented schedule. The jobs were resequenced on the presses using the Pairwise Interchange Heuristic. This resulted in a new schedule whose cost was about 4.3 percent lower than the implemented schedule.

2. Rescheduling and Resequencing the Implemented Schedule

The 77 jobs were allocated and sequenced on the 4 presses using Allocation Heuristic 2 and the Pairwise Interchange Heuristic respectively. The jobs were first allocated to the 4 presses and then sequenced thereon. Since the load was not balanced in the actual schedule, Allocation Heuristic 2 was applied with the load

balancing constraint relaxed. This resulted in a schedule whose cost was about 5.5 percent lower than the implemented schedule.

3. Rescheduling and Resequencing with Load Balancing

The load balancing constraint was enforced and the jobs were reallocated and resequenced via the Allocation Heuristic 2 and the Pairwise Interchange Heuristic on the four presses. This was also repeated using only three of the presses. This gave an indication of the estimated paper wastage if load balancing requirements were also enforced on the system, with three or four machines in operation.

Summary of Plant Problem Analysis

Our implementation results using the allocation and sequencing algorithm show the potential of obtaining upto 6 percent reduction in total labor/production time and paper wastage costs. This savings has been estimated on a set of problems originating in a slack-time period of the year at the plant, where the system capacity comfortably exceeds the job load. However, during the heavy demand periods, when several more jobs are to be processed by the plant, and due date considerations become tight, the human scheduler can be expected to deteriorate in his performance because of the highly combinatorial nature of information that needs to be processed in order to make the scheduling and sequencing decisions. This would be accentuated while considering wider time windows. Under such circumstances the proposed algorithm is more likely to produce a solution which gives higher cost savings while satisfying the due date restrictions, than is the human scheduler. This is a very important consideration for the management which places the highest priority on customer satisfaction.

The list of jobs and presses for this schedule (Tables 1 and 2), the implemented schedule (Figure 3) and the schedules from the runs (Figures 4,5,6 and 7) are shown in the following pages.

**Table 1. Job Numbering Scheme
(TWO WEEK HORIZON, 22" PRESS JOBS)**

#USED	ACTUAL NUMBER
1	598534
2	596439
3	598450
4	599904
5	599930
6	597874
7	598930
8	598435
9	599102
10	599359
11	600033
12	599920
13	597858
14	599281
15	600199
16	600355
17	569121
18	890482
19	598591
20	600215
21	599268
22	601601
23	601494
24	601551
25	602450
26	601692
27	601684

#USED	ACTUAL NUMBER
28	601700
29	579351
30	538977
31	538985
32	597544
33	597551
34	597569
35	597528
36	597536
37	600173
38	596876
39	596868
40	596850
41	596843
42	596353
43	596827
44	596819
45	601528
46	601015
47	601130
48	601148
49	601023
50	589617
51	598856
52	599730
53	600496
54	600082

#USED	ACTUAL NUMBER
55	600058
56	600066
57	600181
58	602070
59	584054
60	595068
61	597379
62	597395
63	597312
64	597288
65	597304
66	597270
67	597361
68	597353
69	597346
70	597338
71	597320
72	597387
73	597296
74	600157
75	590572
76	597379
77	601833

**Table 2. Press Numbering Scheme
(22" PRESSES)**

# USED	ACTUAL PRESS NUMBER
1	22
2	23
3	24
4	27

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	21	22	23	24	25	26	27	28
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 1

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 2

54	55	56	57	58
----	----	----	----	----

MACHINE 3

59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 4

TOTAL COST = 30465 Standard impression units

Figure 3. Implemented Schedule

2	12	6	19	17	18	3	8	21	9	7	5	1	4	16	10	11	13	14	15	20	22	23	24	25	26	28	27
---	----	---	----	----	----	---	---	----	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 1

38	39	40	41	42	43	44	45	37	35	34	36	32	33	31	29	30	51	52	53	50	47	46	49	48
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 2

54	55	56	57	58
----	----	----	----	----

MACHINE 3

74	60	75	63	77	65	66	67	68	69	70	71	72	73	61	62	64	76	59
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 4

TOTAL COST =29125 Standard Impression Units

Figure 4. Resequencing the Implemented Schedule

54	37	18	38	39	40	41	42	43	44	17	60	32	33	34	35	36	29	30	31	1	56	59	65	66	5	55	19	57	20	75	48	22	24	23	27	58	28	26	25
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	----	----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 1

2	6	45	21	12	8	9	3	50	72	73	7	67	68	69	70	71	16	62	63	64	61	10	11	13	14	4	15	76	49	46	47
---	---	----	----	----	---	---	---	----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	----	----	----	----	----

MACHINE 2

75	77
----	----

MACHINE 3

53	52	51
----	----	----

MACHINE 4

TOTAL COST = 28770 Standard Impression Units

Figure 5. Proposed Reallocated and Resequenced Schedule

39	42	60	8	35	19	72	30	65	69	52	25	56	16	10	14	49	77	46
----	----	----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 1

41	44	23	3	33	36	76	37	1	66	13	63	11	4	55	31	74	26	58	27
----	----	----	---	----	----	----	----	---	----	----	----	----	---	----	----	----	----	----	----

MACHINE 2

45	12	17	40	43	75	32	34	70	67	7	62	2	73	24	29	15	57	20
----	----	----	----	----	----	----	----	----	----	---	----	---	----	----	----	----	----	----

MACHINE 3

18	6	21	23	38	51	50	9	53	59	71	68	64	61	48	5	54	47	28
----	---	----	----	----	----	----	---	----	----	----	----	----	----	----	---	----	----	----

MACHINE 4

TOTAL COST = 32030 Standard Impression Units

Figure 6. Alternative Load Balanced Schedule

21	39	42	24	2	7	62	31	68	63	73	71	56	11	52	59	16	13	14	58	49	46	54	47	77	28
----	----	----	----	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

MACHINE 1

17	23	12	75	40	43	44	9	32	34	35	76	19	30	1	65	67	69	70	4	25	74	20	15	57	26
----	----	----	----	----	----	----	---	----	----	----	----	----	----	---	----	----	----	----	---	----	----	----	----	----	----

MACHINE 2

18	6	60	45	38	41	22	51	3	50	8	33	36	29	37	72	66	61	64	5	55	10	53	48	27
----	---	----	----	----	----	----	----	---	----	---	----	----	----	----	----	----	----	----	---	----	----	----	----	----

MACHINE 3

TOTAL COST = 32030 Standard Impression Units

Figure 7. Load Balanced Schedule on Three Machines

10.2 Results from runs on randomly generated data

In the next phase of the testing of the exact as well as the heuristic algorithms, we used randomly generated data as the input to the algorithms. The data was made to conform to the type of data expected to be encountered at the plant. (The data was however scaled down proportionately). In data set 1, uniformly distributed random numbers were generated between 10-15 for the processing time, 1-5 for the allocation and changeover cost and 1-5 for the changeover time. The due date was between 30-120 for small sized problems (upto 15 jobs on 3 machines), between 80-300 for medium sized problems (upto 60 jobs on 4 machines), and between 500-1500 for large problems (upto 200 jobs on 5 machines). In order to test the effect of a change in the range of the cost data on the algorithm, the changeover costs and the allocation costs were varied between 1-15 in data set 2. All other data was generated in the same fashion as in data set 1. The problems generated with data set 1 and data set 2 are shown in Tables 3 and 4 respectively.

The following combination of the exact and heuristic algorithms were tested.

- Solving the allocation and sequencing subproblem exactly.
- Solving the allocation subproblem with Allocation Heuristic 2 and the sequencing subproblem with the Overlapping Batch Exchange Sequencing Procedure (Chapter 9, STEP 2).
- Solving the allocation subproblem with Allocation Heuristic 2 and the sequencing subproblem with the Pairwise Interchange Heuristic.

Table 3. Problems with Data Set 1

PROBLEM #	# OF JOBS	# OF MACHINES
1	10	2
2	12	2
3	15	2
4	15	3
5	30	2
6	30	3
7	50	3
8	60	4
9	100	5
10	150	5
11	200	5

Table 4. Problems with Data Set 2

PROBLEM #	# OF JOBS	# OF MACHINES
1	10	2
2	12	2
3	15	2
4	30	2
5	30	3
6	50	3
7	60	4
8	100	5
9	150	5
10	200	5

A time limit of 5 minutes of execution was put on all the problems. Problems of the size which might conceivably be faced on the shop floor were solved.

1. Exact Allocation and Sequencing (Tables 5,6)

The exact allocation and sequencing algorithms were applied only to smaller sized problems. First, a Lagrangian-Subgradient scheme was used to get a lower bound on the problem. A maximum of 10 iterations of the subgradient algorithm was allowed. This gave a lower bound usually within 15 percent of the optimal solution value for data set 1, and a lower bound within 60 percent of the optimal solution for data set 2. Next, the allocation subproblem was solved. The Lagrangian-Subgradient scheme, which gives a lower bound on the allocation subproblem was allowed a maximum of 10 iterations and typically came to within 50 percent of the optimal solution in all the cases. The computation time for the exact allocation algorithm increases rapidly with problem size (product of the number of jobs and the number of machines) and is the limiting factor in the solution of larger sized problems.

In the case of the sequencing subproblem the subgradient scheme based on the Lagrangean relaxation was also allowed a maximum of 10 iterations and it gave a lower bound on the optimal solution value within 20 percent in the case of data set 1. The lower bound was poor in the case of data set 2. The overall time taken to solve a problem exactly increases rapidly with the problem size and the largest problem that was solved exactly within the total time bound of 5 minutes for the allocation and sequencing subproblems was 15 jobs on 3 machines. The gap between the lower bound and the optimal solution of all the problems increases with the problem size and with the variance in the costs as shown.

2. Allocation Heuristic 2 with Overlapping Batch Exchange Sequencing Procedure (Tables 7,8)

The allocation subproblem cannot be solved exactly for larger sized problems. The sequencing subproblem can be solved optimally with reasonable effort for about 10 jobs. The computational effort increases rapidly beyond this size. This is the motivation for solving the allocation subproblem heuristically and the sequencing subproblem as a sequence of overlapping exact sequencing problems (Overlapping Batch Exchange Sequencing Procedure).

The solution from the Overlapping Batch Exchange Sequencing Procedure was used as the initial sequence for the Pairwise Interchange Procedure to find improving feasible solutions. In all the problems examined, the Pairwise Interchange Procedure did not find an improved solution. For data set 1, the sequencing subproblem picked the optimal solution in all but one of the problems. This can be verified from the fact that the minimum changeover cost between any pair of jobs is 1 unit and the optimal sequence involving n jobs is at best $(n-1)$ units. The solution time is both data dependent as well as problem size dependent, and is largely governed by the time it takes to solve the exact sequencing problems.

3. Allocation Heuristic 2 with Pairwise Interchange Heuristic (Tables 9,10)

For large sized problems the time taken for the sequencing subproblem can be prohibitive due to the large number of exact sequencing problems solved in the Overlapping Batch Exchange Sequencing Procedure. Therefore, for large sized problems, we use Allocation Heuristic 2 for scheduling jobs on the machines and the Pairwise Interchange Heuristic for sequencing the jobs on individual machines. It can be seen that in all the problems solved with data set 1 this solution is within 5 percent

of the solution given by the Overlapping Batch Exchange Sequencing Procedure and the time for solution is on an average about 3 percent of the time taken by the Overlapping Batch Exchange Sequencing Procedure. Problems up to the size which can be expected on the shop floor, (200 jobs on 5 machines) can be solved easily using this heuristic.

10.3 Conclusions and Extensions

In this study, we have addressed the problem of scheduling and sequencing jobs on machines in a two-stage production process. A detailed study of such system, namely, the paper products division of Standard Register at Rocky Mount, Va., was conducted. The salient features and system characteristics which affect the scheduling and sequencing decisions, and the associated time, labor, and material costs were identified. Based on this, a mathematical model of the system was constructed. This mathematical model was decoupled in the framework of a conceptual algorithm which interacts separately with the two production stages, but passes on the information of the solution of each stage to the other in an interactive fashion, in order to implicitly recouple the two stages. A detailed algorithm based on Lagrangian Duality and Benders' Decomposition along with implicit enumeration methods was developed to solve the single stage problem. Two heuristic procedures were developed for the generation of good solutions for larger sized problems.

From a study of the results it can be seen that for problems which involve less than about 15 jobs on 2 or 3 machines, the exact allocation and sequencing algorithms can

be used. Allocation Heuristic 2 along with the Overlapping Batch Exchange Sequencing Procedure can be used to solve problems which have less than about 15 jobs on each machine. Allocation Heuristic 2 in conjunction with the Pairwise Interchange Heuristic can be used to solve larger sized problems, and certainly any problem which might conceivably be faced on the shop floor at the present plant.

This algorithm can be extended to a n -stage process where the stages can be decoupled. In this case the n^{th} stage allocation problem is solved first. The jobs are arranged in an EDD sequence on each of the machines in the n^{th} stage. The start time of the jobs in the n^{th} stage will act as the implied due date for the $n-1^{\text{st}}$ stage. This process is continued till the first stage is reached. The allocation and sequencing problem is solved exactly for the first stage problem with implied due dates. The allocation and sequencing problems are now solved forward from the first stage onwards taking the time availability of the jobs into consideration.

If the two stage problem cannot be decoupled then based on the dual multipliers associated with the solution to the continuous relaxation of the problem, the coupling constraints can be dualized. The resulting problem after dualizing these coupling constraints is the decoupled two stage problem whose solution methodology has been addressed in this thesis. The heuristic proposed herein may then be adapted, with appropriate modifications for trying to achieve feasibility. Details of such a scheme, and the investigation of alternative procedures is suggested for future research.

Table 5. Exact Algorithm - Data Set 1

PROB. #	# OF JOBS	# OF MACH.	OVERALL LAGR. RELXN. LOWER BOUND		ALLOCATION SUBPROBLEM				SEQUENCING SUBPROBLEM				TOTAL FINAL SOLN. (COST)	TOTAL TIME (SECS)
			SOLN. (COST)	TIME (SECS)	LOWER BOUND (COST)	AQP SOLN. (COST)	NODES	TIME (SECS)	LOWER BOUND (COST)	SOLN. (COST)	NODES	TIME (SECS)		
1	10	2	18	2.28	14.40	21.76	65	2.48	4	5	2	0.86	20	5.84
									2	3	1	0.22		
2	12	2	29.36	3.97	21.90	37.44	101	5.22	4	5	2	0.87	32	11.04
									4	5	2	0.98		
3	15	2	33.34	9.18	24.28	45.13	255	17.18	6	7	2	3.24	41	31.01
									5	6	2	1.41		
4	15	3	28.00	11.16	15.45	29.71	2357	267.37	4	8	45	23.81	36	304.55
									3	4	1	0.84		
									2	3	4	1.17		

All lower bounds are based on 10 iterations of the appropriate Lagrangian Relaxation.

Table 6. Exact Algorithm - Data Set 2

PROB. #	# OF JOBS	# OF MACH.	OVERALL LAGR. RELXN. LOWER BOUND		ALLOCATION SUBPROBLEM				SEQUENCING SUBPROBLEM				TOTAL FINAL SOLN. (COST)	TOTAL TIME (SECS)
			SOLN. (COST)	TIME (SECS)	LOWER BOUND (COST)	AQP SOLN. (COST)	NODES	TIME (SECS)	LOWER BOUND (COST)	SOLN. (COST)	NODES	TIME (SECS)		
1	10	2	39.50	1.88	27.98	54.68	74	2.91	8	14	2	1.62	63	6.09
									2	13	4	0.57		
2	12	2	73.21	2.69	54.27	97.90	175	8.56	6	22	102	28.96	116	48.82
									4	11	28	8.61		
3	15	2	67.15	6.14	63.67	109.99	351	23.15	6	46	101	63.88	152	141.16
									5	27	98	47.99		

All lower bounds are based on 10 iterations of the appropriate Lagrangian Relaxation.

Table 7. Allocation Heuristic 2 with Overlapping Batch Exchange Sequencing Procedure - Data Set 1.

PROB. #	# OF JOBS	# OF MACH.	ALLOC. HEUR.		OVERLAPPING SEQUENCING PROCEDURE					PAIRWISE INTERCHANGES			FINAL SOLN. (COST)	TOTAL TIME (SECS)
			TIME (SECS)	SOLN. (COST)	# OF JOBS	EDD SOLN. (COST)	# OF OVERL. PROBS.	TIME (SECS)	SOLN. (COST)	# OF INTER- CHNGS	TIME (SECS)	SOLN. (COST)		
5	30	2	0.30	65	15	14	2	8.11	14	0	0.08	14	94	16.62
					15	33	2	8.05	15	0	0.08	15		
6	30	3	0.35	75	9	16	1	2.82	8	0	0.08	8	102	17.59
					9	15	1	4.63	8	0	0.07	8		
					12	22	2	9.57	11	0	0.07	11		
7	50	3	0.64	97	16	22	2	25.45	15	0	0.28	15	144	85.12
					16	29	2	7.36	15	0	0.27	15		
					18	38	3	50.83	17	0	0.29	17		
8	60	4	1.07	130	19	29	3	13.25	18	0	0.34	18	186	53.50
					12	20	2	7.34	11	0	0.30	11		
					14	25	2	22.32	13	0	0.30	13		
					15	21	2	8.26	14	0	0.29	14		

Table 8. Allocation Heuristic 2 with Overlapping Batch Exchange Sequencing Procedure - Data Set 2

PROB. #	# OF JOBS	# OF MACH.	ALLOC. HEUR.		OVERLAPPING SEQUENCING PROCEDURE					PAIRWISE INTERCHANGES			FINAL SOLN. (COST)	TOTAL TIME (SECS)
			TIME (SECS)	SOLN. (COST)	# OF JOBS	EDD SOLN. (COST)	# OF OVERL. PROBS.	TIME (SECS)	SOLN. (COST)	# OF INTER- CHNGS	TIME (SECS)	SOLN. (COST)		
4	30	2	0.27	200	14	62	2	1.99	29	0	0.12	29	271	4.13
					16	105	2	1.87	42	0	0.14	42		
5	30	3	0.32	240	12	48	2	8.48	19	0	0.09	19	280	182.65
					7	21	1	1.94	6	0	0.09	6		
					11	59	1	171.91	15	0	0.09	15		
6	50	3	0.64	315	16	119	2	26.30	28	0	0.28	28	407	79.34
					17	129	2	27.94	33	0	0.30	33		
					17	80	2	24.46	31	0	0.30	31		

Table 9. Allocation Heuristic 2 with Pairwise Interchange Heuristic - Data Set 1

PROB. #	# OF JOBS	# OF MACH.	ALLOC. HEUR.		SEQUENCING HEURISTIC					FINAL SOLN. (COST)	TOTAL TIME (SECS)
			TIME (SECS)	SOLN. (COST)	# OF JOBS	EDD SOLN. (COST)	TIME (SECS)	# OF INTER- CHANGES	SOLN (COST)		
5	30	2	0.30	65	15	24	0.09	6	15	106	0.42
					15	33	0.10	29	26		
6	30	3	0.35	75	9	16	0.08	4	8	103	0.61
					9	15	0.08	3	8		
					12	22	0.10	8	12		
7	50	3	0.64	97	16	22	0.29	4	16	146	1.56
					16	29	0.30	9	15		
					18	38	0.33	28	18		
8	60	4	1.07	130	19	29	0.35	8	18	189	2.38
					12	20	0.30	2	11		
					14	25	0.33	2	16		
					15	21	0.33	5	14		
9	100	5	4.04	215	24	43	1.69	45	25	314	11.49
					14	20	0.99	6	13		
					18	30	1.56	11	17		
					22	46	1.60	8	21		
					22	38	1.61	15	23		

Table 9. (Continued)

10	150	5	12.33	341	25	46	3.44	24	25	497	30.33
					29	48	3.61	28	30		
					31	50	3.64	28	32		
					32	63	3.64	32	31		
					33	70	3.70	45	38		
11	200	5	24.44	405	45	74	4.22	44	44	660	45.09
					41	81	4.17	10	40		
					41	78	4.12	10	40		
					40	71	4.13	9	39		
					33	54	4.01	29	32		

Table 10. Allocation Heuristic 2 with Pairwise Interchange Heuristic - Data Set 2

PROB. #	# OF JOBS	# OF MACH.	ALLOC. HEUR.		SEQUENCING HEURISTIC					FINAL SOLN. (COST)	TOTAL TIME (SECS)
			TIME (SECS)	SOLN. (COST)	# OF JOBS	EDD SOLN. (COST)	TIME (SECS)	# OF INTER CHNGS.	SOLN (COST)		
4	30	2	0.27	200	14	62	0.89	7	22	264	2.30
					16	105	1.14	42	42		
5	30	3	0.32	240	12	48	0.10	6	22	286	0.58
					7	21	0.07	2	7		
					11	59	0.09	4	17		
6	50	3	0.64	315	16	119	0.30	6	36	430	1.69
					17	129	0.32	14	42		
					17	80	0.33	19	37		
7	60	4	0.79	449	17	98	0.34	10	37	566	2.09
					15	66	0.32	5	22		
					14	92	0.32	13	26		
					14	84	0.32	10	32		
8	100	5	4.12	723	19	93	1.57	15	28	914	12.10
					15	66	0.32	5	22		
					21	132	1.60	25	30		
					21	132	1.68	24	72		
					16	107	1.56	12	27		

Table 10. (Continued)

9	150	5	12.33	1023	33	114	3.59	27	78	1380	30.31
					29	170	3.58	18	99		
					31	167	3.64	28	67		
					28	143	3.55	37	59		
					29	168	3.62	28	54		
10	200	5	24.58	1403	36	189	4.12	35	58	1698	45.54
					46	291	4.34	17	66		
					39	205	4.15	18	52		
					46	273	4.33	45	70		
					33	183	4.02	12	49		

Appendix A. Printing presses.

There are 10 presses at the Rocky Mount plant - 7 Western gear and 3 Goebbel presses.

A summary of their overall capabilities is shown in Table 11.

Besides the overall capabilities listed in Table 1, there are other characteristics which make a job suitable for a particular press . These are listed below :

Litho Unit

Within a litho unit differentiation is based on 'face only' or 'face and back' printing capability. In all the presses the first litho unit (encountered by the paper feed) is capable of reversing the paper and backprinting. The remaining are capable of only face printing.

Rubber imprint unit

This permits face or back printing with inserts mounted on acetate carriers.:

File hole punch units

The file hole punch units suitable for a certain job are determined by the following characteristics :

- a) number of shafts and number of rings per shaft.
- b) rings fixed or segmented.
- c) pattern of punch holes (A through R / 2 & 4 around).
- d) minimum distance across the web on each shaft.
- e) minimum center to center distance with web using 2 shafts.
- f) minimum center to center across web using 2 shafts.

Table 11. Overall Press Characteristics

cylinder size (inches)	press number	num. of litho colors	rubber imprint units	capable of press numbering	capable of folding	max. width (inches)
17	70	2	1	yes	yes	26
	71	2	1	no	sheeter	26
	72	3	1	yes	sheeter	26
22	22	2	1	no	no	19
	23	3	1	yes	sheeter	26
	24	2	1	yes	yes	19
	27	3	1	yes	yes	33 1/2
GOEBBELS	given below	see report	1	yes	yes	20
17	74/75/76					
21	11/12/13					
24	40/41/46					
26	65/66/67					

Line hole punch units

The characteristics of line hole punch units which determine their suitability for a particular line hole requirement are given below :

- a) number of rings on shaft (only one shaft per press).
- b) single row or double row of rings.
- c) punch hole diameter and spacing on each ring.
- d) minimum distance between rings.
- e) maximum center to center distance between outer rings.

Cross perfs

Cross perforation requirements and capabilities of the presses depend on the following factors :

- a) number of shafts.
- b) number around. (ex : for a 22 inch repeat with perfs at 5 1/4 inches require 4 blades)

All Western gear presses have 2 shafts each with 4 and 6 around.

For the Goebbels , depending on the inserted cylinder size, the number of shafts with 4 / 6 / 8 around are available. Two of the Goebbels (s.no. 1272 , 1291) can take 1 cross perf shaft each, the third can take 2 cross perf shafts.

Slitter and rotary perfs

The rotary perforations and slitter capabilities which differ from one press to another are given below :

- a) number of shafts.

- b) minimum center to center distance across web on one shaft.
- c) minimum distance across web using 2 shafts.
- d) skip perf requirement - minimum and maximum length of perf and use of a full shaft.

Rewind / folder / sheeter units

Either one of these units is attached as the final stage at the printing presses. They differ from one press to another in the following characteristics :

- a) kind(s) of units available for a press
- b) rewind units - max. number of rolls wide, max. rewind diameter and width
- c) folder units - fold length, maximum web width for fold
- d) sheeter - maximum web width, cut off length

For the Western gear presses

- 1) all sheeters have a max web width = 26" and cut off size = 17" .
- 2) press no. 70 folds at 8 1/2" with a max width = 26".
no. 27 folds at 11" with a max width = 33 1/2".

A.1.1.1 A note on the Goebels :

The press numbers assigned to the 3 Goebels depend on the size of the cylinder used with it. (ex. for 17" cylinder inserts the three Goebels are numbered 74 / 75 / 76)

The number of litho units, rubber units, numbering units, cross perf cylinders , file hole punch rings and folds are specified according to the cylinder size to be inserted in the Goebel.

There is only 1 Goebel with a drying unit on it. This has to be used for jobs requiring ink drying. (ex. AMEX jobs)

Each of the Goebels has 4 units which will accept up to four inserts of the same size in any order.

Appendix B. Factors relating completion time and wastage to the sequencing of the jobs.

Webbing of paper through the press.

This is dependent on the following job characteristics :

- a) number of litho colors required
- b) backprinting requirement
- c) number wide
- d) rubber imprint / press numbering requirement - face/back
- e) rewind roll / fold / sheet

This means that if factors (a) - (e) are the same for any 2 jobs (and if the 2 nd job can be run on the same press as the 1 st job), the second job will not require rewetting but can be simply taped on to the end of the first job.

Webbing affects the completion time of a job and paper wastage.

Perforation requirements.

There are three kinds of perforations that a job could require - skip perforations, rotary and cross perforations.

a) *Skip perms* require a large set up time and result in a substantial paper wastage in setting up. Hence if a press has been set up for a skip perf pattern , sequencing

jobs with the same perf pattern on this press will save a lot of time and paper. However there is usually a small number of such jobs.

b) *Rotary perfs* differ from one job to another in the number of wheels required, type of perf blade and the spacing of wheels across the web. These operations involving the set up of rotary perfs take up a small amount of time and paper.

c) *Cross perfs* on jobs may differ in their spacing, number of perfs and type of cross perf. Change of the type of cross perf requires changing the blade(s). This will involve time for removal of old blades, insertion of new blades and alignment. Alignment causes paper wastage. An increase / decrease in the number of cross perfs requires addition/ removal of blades. This consumes time and also paper if there is an addition of blades involved. A change in the spacing of the perforations on the form will require the addition / removal of blades.

All settings are done visually by the operator. The plate has markings at the appropriate places for perforations and the operator visually tries to place the blades to perforate at these positions. Changing a perf blade could take up to 15 - 20 minutes.

Ink color change

For each litho unit ink color change requires scraping off the old ink , cleaning the ink holder using chemicals and filling in the new ink. The extent of cleaning is dependent on the color change involved. For a change to standard black, a crude cleanup is sufficient. This takes about 10 minutes. A thorough clean up could take upto half an hour.

Cleaning up *does not* involve paper wastage. However getting the correct flow of ink to match the specified color on the copy is a trial and error procedure and involves some paper wastage.

File hole and line hole punch units

File hole punch units require setting up of the punch and spacing on the paper. Its proper registration onto the paper would require some time and paper wastage.

For line hole punch units, differences between consecutive jobs would be based on form dimensions and the form type. (system set or continuous, with the former having 1 edge and the latter 2 edges with line holes).

Plate change

Zeroing in the plate onto the paper will take up time and some paper wastage. Markings on the roller (holding the plate) facilitates this task.:

Rubber unit / numbering unit change

Time and paper wastage are incurred by a change in either of these units, or a position change of a unit, or an additional back printing requirement, i.e., face or back registration required.: In addition to these factors, paperweight changes requiring folder and sheet length adjustments, and close register jobs (OCR) incur time and paper wastage.

Appendix C. Factors unaffected by sequencing but resulting in time and paper wastage.

There are some factors which result in time and paper wastage but are independent of the sequencing of the jobs. These are outlined below :

a) Unwind roll change

b) Rewind roll change

c) Through length changes - this is a change in the length of the paper as it is pulled through the press. This is crucial , since any increase in the length of the paper will adversely affect collating.

However this is purely a function of the quality of the paper , speeds on the press and the paper tension maintained.

Press speed is left to the operator's judgement (although the spec sheet does specify a standard) - his considerations are usually avoiding straking of ink, ripping of paper and getting close enough to the specified standard.

d) Job O.K. stop/ restart - some jobs may require stopping of the job to obtain the supervisor's O.K. before going ahead.

e) Paper mayrun out due to greater wastage than planned for – results in the job being taken off the press if more paper is not immediately available.

f) Rush jobs - may be externally or internally generated. These will require overriding all time and wastage costs on other jobs - their completion within a specified time

may require set up changeovers and sequence disruption resulting in time and paper wastage.

g) Machine problems / breakdowns.

h) Butt rolls - a paper roll with a diameter less than a specified size is discarded as a butt roll. Sometimes butt rolls are spliced together for use on short jobs.

Appendix D. Collators

There are 12 collators at the Rocky Mount plant - 4 continuous form and 8 system set collators.

A summary of their overall characteristics / limitations is given in Tables 12 and 13.

The collators are different from the presses in that they can be set up for almost any job. Hence, there are few characteristics which govern the suitability of a job for a certain collator. The two major fixed characteristics which limit job suitability to a certain collator(s) are :

1) system set / continuous form - This is a semi - fixed characteristic since changing a s/s collator into a c/f collator (& vice versa) is possible but takes a substantial amount of time and is usually avoided.

2) number of plys on the job to be collated - The maximum number of paper stations on a collator limits the suitability of a collator for a certain job. (usually for jobs with more than 5 plys)

Collators capable of handling more than 5 plys are :

6 plys - all c/f collators and s/s # 10,60,80

8 plys - c/f # 81,82 & s/s # 10,80

10 plys - s/s # 10

Size change minimization is an important factor in the sequencing at the collators. The volume of waste at the collators is less but the value per pound is considerably higher.

Table 12. Collator Characteristics 1

number	conventional numbering units	num. of paper stations	maximum roll diameter (inches)	number of carbon stations	maximum carbon diameter (inches)	capable of punching
Continuous form Collators						
61	1	6	40	5	24	no
62	0	6	24	5	18	no
81	1	8	40	7	24	no
82	1	8	24	7	18	no
System Set						
10	2	10	24	9	18	yes
20	1	2	24	1	18	no
42	1	4	24	3	18	yes
43	2	4	24	3	18	no
44 & 45	2	4	40	3	24	no
60	1		24	5	18	no
80	2	8	24	7	18	no

Table 13. Collator Characteristics 2.

number	capable of rotary perfs	capable of crimping	num. of glue systems	conversion from / to s/s / c/f	maximum form width (inches)
Continuous form Collators					
61	yes	yes	1	yes	20
62	yes	yes	1		19
81	yes	yes	1		20
82	yes	yes	2		18
System Set					
10	yes	no	1	yes	19 1/2
20	no	no	1		19
42	no	no	1		18 or 19
43	no	no	1		18 or 19
44 & 45	no	no	1		20
60	no	no	1		19 1/2
80	no	no	1		19

Appendix E. Normal job characteristics and limitations at the collators.

The normal job characteristics of a collator are the specifications of the kinds of jobs normally assigned to that collator. The limitations outlined are either due to technical constraints or operational problems. These are outlined against the corresponding collator number.

- 61** used for 17" continuous
 jumbo paper or carbon rolls can be used
 any other size can be run except 26" multiples
 folding at less than 8 1/2" is not recommended
 maximum fold length is 14"

- 62** used for 24" continuous
 can be used for 17",21",22" multiples
 can be converted to system set - can handle 26" s/s
 numbering machine has been removed
 fold length between 7" and 14"
 small rewinds only
 no trim if converted to system set

- 81** used for 22" continuous

can collate any size except 26" multiples
jumbo paper or carbon rolls can be used
recommended for folding length $\leq 8 \frac{1}{2}"$
recommended folding width $\leq 7"$

82 normally 22" multiple continuous jobs
can handle multiples of 17" and 21"
small rewinds only
no numbering for 21" multiples
(has 2 units each of 17" and 22" numbering units)
no stub trim
a dryer unit addition has resulted in this collator
handling most continuous glue work.
a 24" head will be available for this collator.

10 about half the jobs are $4 \frac{1}{4}"$, $5 \frac{2}{3}"$, $8 \frac{1}{2}"$
multiples and the remaining are 6" multiples.
recommended for all crash imprints
has file hole (17" & 22") and MR 516 (22") punching

20 4.8" system set work usually but can handle
any system set work.
used for 2 ply guest checks
this collator is a good emergency backup for
collator numbers 42 & 43.

42 normally $4 \frac{1}{4}"$ system set work with carbon less
transfer paper because the carbon does not run
well on the batcher stacker.

minimum stub size = 3.4"

file hole punching is possible on this collator

17" system set work only

modification for 22" and 24" work is planned

43

normally 4.8" system set work with carbon less

transfer paper - can be used for other sizes too.

can take upto 2 jumbo rolls of carbonless transfer paper

can crash imprint

use only carbonless transfer paper because carbon does not run well on the batcher stacker (has been consequently removed)

cut off size between 4" and 12" (to be increased to 17" after modifications)

44

currently runs 3" AMEX forms but can handle any

45

size of system set work

has MICR and conventional numbering capabilities

jumbo paper and carbon can be used

folding length between 7" and 14"

45 now runs PMO almost exclusively

60

can handle any size system set work but usually

set up to run 3 4/10" or 8 1/2" system set work

no trim system (2 portable trim systems can be used)

spot gluing over 3 ply not recommended

80

can handle any size system set work but usually

set up for 17" or 22" work

the collator is to be used for 24" system set work

with the addition of a new numbering unit.

Appendix F. Collator cylinder sizes and slotting arrangements.

F.1.1.1 Western gear system set cylinders.

size	quantity	slotted for (inches)
17	3	17, 8 1/2, 5 2/3, 4 1/4
	2	17, 8 1/2, 3 4/10
	1	17, 8 1/2, 5 2/3, 2 5/6
21	1	21, 10 1/2, 7, 3 1/2
22	3	22, 11, 7 1/3, 5 1/2, 3 2/3
	1	22, 11, 5 1/2, 2 3/4 - (PMO head)
24	2	24, 4.8
	1	24, 12, 8, 6, 4
	1	24, 12, 6, 3 -(AMEX head)
26	1	26, 13, 6 1/2, 3 1/4

F.1.1.2 Western gear continuous cylinders

16	1	8, 4
17	4	17, 8 1/2, 5 2/3, 4 1/4
21	1	10 1/2, 7, 3 1/2
22	4	22, 11, 7 1/3, 5 1/2, 3 2/3
24	1	24, 12, 8, 4
	1	24, 12, 6, 3

F.1.1.3 Hamilton continuous cylinders

17	1	8 1/2, 5 2/3, 4 1/4
22	1	11, 7 1/3, 5 1/2, 3 2/3
28	1	14, 7, 3 1/2

Appendix G. Factors relating completion times, paper wastage and sequencing

c/f - denotes a factor relevant only in collating continuous forms.

s/s - indicates a factor relevant only to system set forms.

Cut off size (s/s)

A change of cut off size may involve the change of a cylinder or only the addition / removal of blades.

For s/s collators, cylinders with the same head size but different slotting arrangements are available. Hence, even if the head size (of the currently used cylinder in the planned collator) is a multiple of the form size , the cylinder may have to be changed to obtain a suitable cut off slot. This will take up a significant amount of time and involve some paper wastage.

The cylinder sizes, quantity and slotting arrangements are given in Appendix F.

Cross perforation requirement (c/f)

When perforations are required through the carbon, they have to be done at the collators.

The only variations from one job to another are in the head size and type of perf blade.

- a) Head size - all c/f cylinders of the same head size have *identical* slotting arrangements. Hence if for example a 17" head is on a collator , a cylinder change would be required only if the new job does not divide exactly into 17".
- b) type of perf pattern - this would require the removal of old blades, the insertion of the new blades and the proper registration of these new blades. This results in some time and paper wastage.

Number of plys

A reduction in the number of plys on a job in sequence does not take up too much time. However, an increase requires rewetting of the additional plys. To ensure smooth registration and paper flow with the additional plys some time and paper wastage will be incurred.

In case the number of plys in a job is the same as on the preceding job in the sequence, the new job is simply taped on to the end of the preceding job. This saves webbing time and paper wastage.

Width adjustment

The pin feed line has to be adjusted to fit the line feed holes. This takes up time and paper wastage.:

Gluing systems

- a) Spot / line gluing - a changeover from one to the other would require a change of the glue mounting head(s). Some time and paper wastage is involved in getting the glue units restarted and properly registered.

b) Inboard / outboard gluing - inboard is the gear side and outboard is the operator side. Inboard / outboard gluing depends on whether the job is head pull / tail pull of the presses.

c) Gluing pattern - the plys between which the form is glued vary according to a carbon to ply / ply to ply gluing.

Any gluing pattern change (for a job in sequence) will require rewetting of the affected plys and may require restarting some other gluing heads. The glue usually dries up in the head and the glue lines.

Hence, obtaining a smooth flow from a head which has not been used for a long time takes up a lot of time and paper wastage.

For pattern carbon which has a stub with pin feed holes and which has to be treated like a paper roll for collating, some gluing patterns may require mounting of the carbon rolls on paper spindles.

Fold length (c/f)

Any change in this requires time and paper wastage.:

Numbering requirements

A change in any of the following factors will require a change in numbering machine(s).

1) Factors specifying numbering machines :

- a) Straight / convex numbering - most continuous forms require straight numbering.
- b) Skip 2/4/6 .. - skip 2 would mean the use of 2 machines , one with odd numbers the other with even numbers.

c) Form size - This affects the skip # and the number of machines to be mounted on the cylinder.

d) Forward / backward numbering - This is determined by tail / head pull off the press as specified by prep.

e) Odd / even machine - for example skip 4 will have 2 odd and 2 even machines & skip 5 will need 5 odd numbering machines.

2) Ink color change for the numbering - Typically wash up and reinking would take a 1/2 hour and some paper would be wasted in proper ink registration.

note - all machines have the same number / letter sizes.

machines are either already available (if standard type) or are easily prepared within the plant.

MICR numbering is possible only on 2 collators (# 44 & 45) which are currently set up to run AMEX and PMO jobs.

Crimping unit (c/f)

Setting up the crimp position will take up some time and paper wastage.

Appendix H. Doctoring machines

There are 2 doctoring machines in the plant. They are used for

- a) cutting carbon width to the required size.
- b) removing splices from rolls.

When small rolls of paper are run through a press, they are spliced one behind the other to fill out the required length of paper. These splicing positions are marked with tags.

At the doctoring machine an operator removes these tags and splices the paper *accurately* , so that the roll can be run as one continuous roll at the collators. If this is not done , this job would have to be stopped at the collator (as a tag appears), cut off , rewebbed and run again for the remaining portion of the job.

Although several jobs require 'doctoring' there is no bottleneck at this stage and jobs are always at the collators when required.

Appendix I. Current scheduling considerations at the production level.

The daily routine is as follows :

- 1) The job tickets are received at about 10 a.m. every day.
- 2) The list of new tickets and the unfinished jobs till that day are reviewed and a new schedule is drawn up for these.
- 3) The normal daily load is 25 - 30 jobs.

The scheduling routine at the production level is as follows :

a) First group the jobs by size - Three major groups result

17" jobs

22" jobs

Goebbel jobs (24",26")

(17",21")

b) Within a group arrange the jobs by due dates. Consideration is given to both the overall and the press due dates but the planning is based on the overall due date.

c) Look for job similarities, and sequence similar jobs one after the other. More details on this are given below.

d) Look at collator status - if for example there is very little work on the continuous collators, then push continuous jobs forward.

The job similarities examined for sequencing considerations are identified as follows :-

- 1) Subgroup each job size according to the number of litho colors.

job size	# of litho colors	suitable presses
----------	-------------------	------------------

17"	2	70,71,(72)
-----	---	------------

	3	72
--	---	----

22"	2	21,22,24,(23,27)
-----	---	------------------

	3	23,27
--	---	-------

24",26" Choose from the 3 Goebbels depending on
their current set up .(1/2/3 litho units)

- 2) Further grouping in each of the above categories based on s/s or c/f specifications,in order to avoid line hole changes.
- 3) Consider run time of jobs - shift duration and availability of operators.
- 4) Consider 1 wide / 2 wide runs - to avoid rewebbing.
- 5) Consider perf requirement :
 - a) similarity in cross perf pattern & type involving no change in blades.
 - b) reduction in the # of perfs - not too many blades need to be replaced.
 - c) skip perf - special consideration because of the set up time involved.
- 6) Consider rewind / fold / sheet requirement.
- 7) Consider press numbering.

8) Consider total load on a machine - related to job run times.

Appendix J. Some observations on plant operation.

a) The plant operates on 3 daily shifts :

shift	# press operators	# collator operators
A	3	4
B	10	10
C	6	6

About 80 % - 100 % attendance over 3 shifts.

b) A maximum of 6 - 8 breakdowns per day

- some are minor and can be fixed in minutes.

- Goebels have a lot of breakdowns.

c) The Goebels are mostly set at 24" unless there is a particularly heavy load of 17" work. This is because there are a lot of 24" & 17" jobs.

d) Most of the numbering is done at the collators.

e) The standards on the specifications sheet are tight. Scheduling is based on about 1.5 times the run time specified on the spec sheet.

The machines specified on the spec sheet are not binding but are used more as guidelines.

A particularly heavy load would be about 50 standard hours per press.

Bibliography

1. Adams, W.P. and Sherali, H.D., "A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems," **Management Science**, 32 (10), 1274-1290 (1986)
2. Akl, S., "The Minimal Directed Spanning Graph for Combinatorial Optimization," **The Australian Computer Journal**, 12 (4), 132 - 136 (1980).
3. Assad, A., "Analytic Models in Rail Transportation : An Annotated Bibliography," **INFOR**, 19 (1), 59-80 (February 1981).
4. Bartholdi, J.J. and Platzman, L.K., "An $O(n \log n)$ Planar Travelling Salesman Heuristic Based on Space Filling Curves," **Operations Research Letters**, 1 (4), 121-125 (1982).
5. Balas, E. and Christofides, N., "A Restricted Lagrangean Approach to the Travelling Salesman Problem," **Mathematical Programming**, 21, 19-46 (1981).
6. Bazaraa, M.S. and Sherali, H.D., "On the Choice of Step Size in Subgradient Optimization," **EJOR**, 7, 380-388 (1981).
7. Bazaraa, M.S. and Jarvis, J.J., **Linear Programming and Network Flows**, New York Wiley, 1979.
8. Bellmore, M. and Hong, S., "Transformation of Multi-Salesman Problem to the Standard Travelling Salesman Problem," **Journal of the Association of Computing Machinery**, 21, 500-504 (1974).
9. Bodin, L. and Berman, L., "Routing and Scheduling of School Buses by Computer," **Transportation Science**, 13 (2), 113-129 (1979).
10. Bodin, L., Golden, B., Assad, A. and Ball, M., "The State of the Art in the Routing and Scheduling of Vehicles and Crews," College of Business and Management, University of Maryland, College Park, Maryland, Report # UMTA/BMGT/MSS # 81-001 (1981).
11. Bodin, L. and Kursh, S., "A Detailed Description of a Street Sweeper Routing and Scheduling System," **Computers and Operations Research**, 6, 181-198 (1979).

12. Bodin, L. and Kursh, S., "A Computer-Assisted System for the Routing and Scheduling of Street Sweepers," **Operations Research**, 26 (4), 525-537 (1978).
13. Bodin, L. and Sexton, T., "The Subscriber Dial-a-Ride Problem : The Baltimore Maryland Benchmark," ORSA/TIMS Meeting, Colorado Springs (November,1980).
14. Bondi, J.A. and Murty, U.S.R, **Graph Theory with Applications**, New York American Elsevier Publishing Company, 1976.
15. Camerini, P.M., Fratta, L. and Maffioli, F., "On Improving Relaxation Methods by Modified Gradient Techniques," **Math. Prog. Study** 3, 26-34 (1975).
16. Chan A.W. and Udo Graffe P.W., "An Interactive Sequencing Aid For Printing Presses," **Computers and Industrial Engineering** 321-325 (1979)
17. Christofides, N., "The Vehicle Routing Problem," **Rev. Frans. Res. Oper.** 10, 55-70 (1976)
18. Christofides, N., "Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem," Report 388, Graduate School of Industrial Administration, Carnegie Mellon University (February 1976).
19. Christofides, N. and Eilon, S., "Algorithms for Large Scale Travelling Salesman Problems," **Operational Research Quarterly**, 23, 511 (1972).
20. Christofides, N., Mingozzi, A. and Toth, P., "Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations," **Mathematical Programming** 20, 255-282 (1981).
21. Clarke, G. and Wright, W., "Scheduling of Vehicles from A Central Depot to A Number of Delivery Points," **Operations Research**, 12, 568-581 (1964).
22. Crowder, H. and Padberg, M., "Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality," **Management Science**, 26 (5), 495-509 (1980).
23. Crowder, H., Johnson, E.L., and Padberg, M., "Solving Large Scale Zero-One Linear Programming Problems," **Operations Research**, 31, 803-834, (1983).
24. Cullen, F., Jarvis, J. and Ratliff, H., "Set Partitionong Based Heuristics for Interactive Routing," **Networks**, 11 (2), 125-144 (1981).
25. Eilon, S., Watson-Gandy, C. and Christofides, N., **Distribution Management : Mathematical Modeling and Practical Analysis**, Hafner Publishing Co., New York (1971).
26. Fisher, M. and Jaikumar, R., "A Generalized Assignment Heuristic for Vehicle Routing," **Networks**, 11 (2), 109-124 (1981).

27. Frieze, A., "Worst-case Analysis of Algorithms for Travelling Salesman Problems," **Operations Research Verfahren**, 32, 94-112 (1978).
28. Garey, M., Graham, R. and Johnson, D., "Some NP-Complete Geometric Problems," **Proc. 8th SIGACT Symp. on the Theory of Computing**, 10-22 (1976).
29. Geoffrion A.M. and Graves G.W., "Scheduling Parallel Production Lines with Changeover Costs: Practical Application of a Quadratic Assignment / LP Approach," **Operations Research** 595-610 (1976)
30. Gillett, B. and Johnson, T., "Multi-terminal Vehicle-Dispatch Algorithm," **Omega** 4, 711-718 (1976).
31. Gillett, B. and Miller, L., "A Heuristic Algorithm for the Vehicle Dispatching problem," **Operations Research**, 22, 340-349 (1974).
32. Golden, B., Bodin, L., Doyle, T. and Stewart, W., "Approximate Travelling Salesman Algorithms," **Operations Research**, 28 (3), 694-711 (May-June 1980).
33. Golden, B., Magnanti, T. and Nguyen, H., "Implementing Vehicle Routing Algorithms," **Networks**, 7, 113-148 (1977).
34. Golden, B. and Wong, R., "Capacitated Arc Routing Problems," **Networks**, 11 (3), 305-318 (1981).
35. Held, M. and Karp, R., "The Travelling Salesman Problem and Minimum Spanning Trees," **Operations Research**, 18, 1138-1162 (1970).
36. Held, M. and Karp, R., "The Travelling Salesman Problem and Minimum Spanning Trees Part II," **Mathematical Programming**, 1, 6-25 (1971).
37. Hinson, J. and Mulherkar, S., "Improvements to the Clarke and Wright Algorithm as Applied to an Airline Scheduling Problem," Technical Report, Federal Express Corp. (1975).
38. Houck, D., Picard, J., Queyranne, M. and Vemuganti, R., "The Travelling Salesman Problem and Shortest n-Paths," **Opsearch**, 17, 93-109 (1980).
39. Kanellakis, P. and Papadimitriou, C., "Local Search for the Assymmetric Travelling Salesman problem," **Operations Research**, 28 (5), 1086-1099 (1980).
40. Karp, R., "A Patching Algorithm for the Nonsymmetric Travelling Salesman Problem," **SIAM J. Comput.**, 8, 561-573 (1979).
41. Karp, R., "Probabilistic Analysis of Partitioning Algorithms for the Travelling Salesman Problem in the Plane," **Mathematics of Operations Research**, 2, 209-244 (1977).

42. Karp, R., "Reducibility among Combinatorial Problems," **Complexity of Computer Computations**, (R. Miler and J. Thatcher, eds.), 85-104, Plenum Press, New York (1972).
43. Kim, C., "A Minimal Spanning Tree and Approximate Tours for a Travelling Salesman," Comp. Sci. Tech. Report, University of Maryland (1975).
44. Krolak, P., Felts, W. and Marble, G., "A Man-Machine Approach Toward Solving the Travelling Salesman Problem," **Comm. ACM**, 14, 327-334 (1971).
45. Krolak, P., Felts, W. and Nelson, J., "A Man-Machine Approach Toward Solving the Generalized Truck Dispatching Problem," **Transportation Science**, 6, p. 149 (1972).
46. Lawler, E.L., Lenstra, J.K., Rinooy Kan, A.H.G., Shmoys, D.B., eds., **The Travelling Salesman Problem : A Guided Tour of Cominatorial Optimization** , Chichester [West Sussex]; New York Wiley, 1985
47. Leipala, T., "On the Solutions of Stochastic Travelling Salesman Problems," **European Journal of Operational Research**, 2 (4), 291-297 (1978).
48. Levy, L., Golden, B. and Asssad, A., "Optimal Fleet Size and Mix," submitted for publication.
49. Lin, S., "Computer Solutions of the Travelling Salesman Problem," **Bell System Technical Journal**, 44, 2245-2269 (1965).
50. Lin, S. and Kernighan, B., "An Effective Heuristic Algorithm for the Travelling Salesman Problem," **Operations Research** 21, 498-516 (1973).
51. Little, J., Murty, K, Sweeney, D. and Karel, C., "An Algortihm for the Travelling Salesman Problem," **Operations Research**, 11 (6), 972-989 (1963).
52. Miliotis, P., "Integer Programming Approaches to the Travelling Salesman Problem," **Mathematical Programming**, 10, 367-378 (1976).
53. Newton, R. and Thomas, W., "Design of School Bus Routes by Computer," **Socio-Economic Planning Sciences**, 3, 75-85 (1969).
54. Norback, J. and Love, R., "Geometric Approaches to Solving the Travelling Salesman Problem," **Management Science**, 23, 1208-1223 (1977).
55. Orloff, C., "Routing a Fleet of M Vehicles To/From A Central Facility," **Networks**, 4, 147-162 (1974).
56. Papadimitriou, C., "The Euclidean Travelling Salesman Problem is NP-Complete," **Theoretical Computer Science**, 4 (3), 237-244 (1977).

57. Rosenkrantz, D., Stearns, R. and Lewis, P., "An Analysis of Several Heuristics for the Travelling Salesman Problem," **SIAM J. Computing**, 6, 563-581 (1977).
58. Rubin, J., "A Technique for the Solution of Massive Set Covering Problems with Application to Airline Crew Scheduling," **Transportation Science**, 7, 34-48 (1973).
59. Russell, R., "An Effective Heuristic for the M-Tour Travelling Salesman Problem with Some Side Conditions," **Operations Research**, 25 (3), 517-524 (1977).
60. Stern, H. and Dror, M., "Routing Electric Meter Readers," **Computers and Operations Research**, 6, 209-223 (1979).
61. Stewart, W. and Golden, B., "A Vehicle Routing Algorithm Based on Generalized Lagrange Multipliers," **Proceedings of AIDS 1979 Annual Convention** (L. Moore, K. Monroe, B. Taylor, eds.), New Orleans, 108-110 (1979).
62. Svestka, J. and Huckfeldt, V., "Computational Experience with an M-Salesmen Travelling Salesman Algorithm," **Management Science**, 19 (7), 790-799 (1973).
63. Taha, H.A., **Integer Programming Theory Applications and Computations**, New York Academic Press, 1975

**The vita has been removed from
the scanned document**