71-27,647

SULLENBERGER, Alfred Gale, 1940-
AN INVESTIGATION OF PROBABILISTIC DISPATCHING PROCEDURES FOR MINIMIZING PENALTY COSTS IN A JOB-SHOP.

The University of Oklahoma, D.Engr., 1971
Engineering, industrial

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

AN INVESTIGATION OF PROBABILISTIC DISPATCHING PROCEDURES FOR MINIMIZING PENALTY COSTS IN A JOB-SHOP

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF ENGINEERING

BY

ALFRED GALE SULLENBERGER

Norman, Oklahoma

1971

AN INVESTIGATION OF PROBABILISTIC DISPATCHING

PROCEDURES FOR MINIMIZING PENALTY

COSTS IN A JOB-SHOP

APPROVED BY

DISSERTATION COMMITTEE

ACKNOWLEDGMENTS

This dissertation would not have been possible without the assistance and cooperation of a large number of people, all of whom cannot possibly be mentioned here. However, I particularly want to acknowledge Professor B. L. Foote, my committee chairman, whose encouragement and advice was indispensable to this research. Also, Professor J. E. Shamblin of Oklahoma State University assisted me in defining the problem in the initial investigations. North American Rockwell provided me an opportunity to study their Tulsa, Oklahoma facility which was very important to my understanding of the problem. Computer Congenerics Corporation provided me with unlimited use of its computer which allowed me to finish the simulations more rapidly than would otherwise have been possible. Mrs. E. Porterfield typed this dissertation with an attention to detail and an efficiency that was very important to the timely completion of this research.

Although my wife, Laura, did not contribute directly to this dissertation, it would not have been possible without her sacrifices and encouragement, both during the preparation of this dissertation and throughout all of my graduate studies.

TABLE OF CONTENTS

LIST OF TABLES

Table		Page

LIST OF ILLUSTRATIONS

Illustration | Page

AN INVESTIGATION OF PROBABILISTIC DISPATCHING PROCEDURES FOR MINIMIZING PENALTY COSTS IN A JOB-SHOP

CHAPTER I

THE JOB-SHOP PROBLEM

If it is assumed that a job is a single piece of work and a facility is a machine or group of similar machines, then a job-shop may be defined. Rowe (48) defined a job-shop by contrasting it to a flow-shop or production-shop in which all jobs are processed by the same facilities in the same order. Sisson (49) characterized a job-shop by the fact that the sequence of operations performed on any one job is independent of the sequence required for any other job. Conway and Maxwell (12) characterized a job-shop as a network of queues. Gere (18) stated in his Ph.D. thesis that a job-shop connotes a large number of jobs with diverse routings which compete for time on common machine facilities. Although all of these definitions are compatible, Gere's definition will be accepted for the purposes of this paper.

In a job-shop, the jobs to be processed usually have certain attributes associated with them. For Job i these attributes would include a due-date, d_i, which is the time at which Job i should be completed; a time available, r_i, which is the earliest time at which processing may begin on the job; a routing which is the sequence of facilities that must process a job; and the expected processing-time, P_{ij}, of each Job i on each Facility j in its routing. For most job-shop investigations, the facilities are assumed, for simplicity, to consist of a single machine.

The job-shop problem is to determine the sequence in which the available jobs should be processed at each facility so as to optimize some measure of effectiveness of the schedule generated by the sequencing decisions made at each facility. There is no single measure of schedule effectiveness which is accepted as best for the job-shop problem. Many measures of effectiveness for job-shop problems have been suggested and used to make comparative evaluations of schedules generated by the various techniques of making sequencing decisions in a job-shop. Most of the measures of effectiveness for job-shop schedules which have been utilized are a function of the following parameters (13):

1. C_i, the completion-time of Job i.
2. F_i, the flow-time of Job i.

3. L_i , the lateness of Job i.

4. T_i , the tardiness of Job i.

5. E_i , the earliness of Job i.

Lateness, tardiness, and earliness are three different ways of comparing the actual completion-time of a job with its due-date. Lateness is the algebraic difference of completion-time and the due-date of each job, regardless of the sign of the difference. Tardiness is the length of time a job finishes after its due-date and considers only those jobs that are completed after their due-dates. Earliness is the length of time a job finishes before its due-date and considers only those jobs completed before their due-dates. In nearly all of the work published on the job-shop scheduling problem, very simple measures of effectiveness have been used. These are usually mean completion-time, mean flow-time, maximum flow-time, mean lateness, maximum lateness, mean tardiness, or maximum tardiness.

Many techniques have been proposed and investigated for optimizing these measures of effectiveness of a job-shop schedule.

These techniques include:

1. Total enumeration and comparative evaluation of all feasible schedules.
2. The generation of a sample of feasible schedules and the comparative evaluation of these schedules.

3. The solution of the job-shop problem by formulating it as an integer programming problem.
4. The application of branch-and-bound techniques to the job-shop problem.
5. The application of graphical techniques to the job-shop problem.
6. The use of priority sequencing rules.

CHAPTER II

PREVIOUS INVESTIGATIONS

Almost all of the investigations that have been made to date were concerned with a highly restrictive problem called a simple job-shop problem. These restrictions are (13):

1. Each machine is continuously available for assignment.
2. Jobs are strictly-ordered sequences of operations, without assembly or partition.
3. Each operation can be performed by only one machine in the shop.
4. There is only one machine of each type in the shop.
5. Once an operation is started on a machine it must be completed before another operation can begin on that machine.
6. A job may be in process on at most one machine at a time.
7. Each machine can handle at most one job at a time.

A special case of the simple job-shop problem is the shop with only one facility or machine. Smith (51) showed in 1956 that the mean flow-time for the n-job, 1-machine problem is minimized by the shortest processing time first (SPT) priority sequencing rule. Jackson (28) showed in 1955 that for n jobs and 1 machine, both maximum job lateness and maximum job tardiness are minimized by sequencing the jobs in order of nondecreasing due-dates. For n jobs and 1 machine, the minimum job lateness and the minimum job tardiness are maximized by sequencing jobs in the order of nondecreasing slack-time (13). If F_i is defined to be the flow-time of Job i through the facility, U_i is defined to be a weighting factor for Job i, and P_i is the required processing-time of Job i, then $\sum_{i=1}^{n} U_i F_i$, the total weighted flow-time of all jobs, is minimized by sequencing the jobs so that the ratio of the processing-time divided by the weighting factor, P_i/U_i, is nondecreasing (51). Also, mean weighted lateness and mean weighted waiting-time are minimized by P_i/U_i ratio sequencing (39).

In 1964 Lawler (35) published a dynamic programming formulation of the n-job, 1-machine scheduling problem where each job has associated with it a deferral cost which is monotonically nondecreasing with time. Lawler stated that this method may be used to solve problems with up to approximately 15 jobs. The solution of problems with more than 15 jobs requires a prohibitive amount of computation.

In 1968 an algorithm was published by Moore (42) which sequences n-jobs through a single facility so as to minimize the number of late jobs. This algorithm is computationally feasible for a large number of jobs. The algorithm was extended to produce a schedule which minimized the maximum deferral cost when each job has associated with it a continuous monotone nondecreasing deferral cost function. It is interesting to note that in January of 1970, William L. Maxwell published a paper (38) which developed Moore's algorithm from an integer programming formulation of the n-job, 1-machine scheduling problem.

Another special case of the job-shop problem for which a solution is known is the n-job and 2-machine flow-shop problem. Johnson (33) and (34), Jackson (29), Mitten (40), and others have investigated this problem. Johnson (33) gave an algorithm for sequencing n-jobs, all simultaneously available, in a 2-machine flow-shop so as to minimize the maximum flow-time. Johnson also considered a special 3-machine case in which the second machine is completely dominated by either the first or third machine. For this special 3-machine case, an algorithm is given which minimizes the maximum flow-time. This paper is perhaps the best known and most referenced in the literature on sequencing. Jackson (29) has shown that the 2-machine job-shop problem with the restriction that each job can have at most two operations can be solved by a generalization of

Johnson's 2-machine flow-shop algorithm. This is the only true job-shop problem for which a simple algorithm for a global solution is known. A graphical procedure for the 2-job flow-shop problem was given by Akers (1). This same approach has been applied to the 2-job job-shop problem by Akers and Firedman (2) and discussed more completely and formally by Hardgrave and Nemhauser (23). The graphical procedure is practical for only very small problems.

Approximate methods for the sequencing of n-jobs through a flow-shop with m-machines have been developed by Palmer (46) and Campbell, Dudek, and Smith (7). The objective of the Palmer algorithm is to minimize the total time required to complete all jobs. This is accomplished by computing what Palmer called a slope index which is used to make sequencing decisions. This algorithm gives priority to those jobs having the strongest tendency to progress from short process times to long process times in the sequence of processes. The Campbell, Dudek, and Smith Algorithm is an extension of the Johnson Algorithm for the n-job, 2-machine flow-shop (33). This method requires a reasonable amount of calculation and can be used to solve large problems.

A general algorithm for the solution of the n-job, m-machine flow-shop problem has been given by Smith and Dudek (50). The algorithm is a partially enumerative combinatorial procedure for minimizing flow-time. Computational experience with this algorithm indicates that problems with greater than 10 jobs require excessive computation time precluding finding optimal solutions (7).

Some special cases of the job-shop scheduling problem can be modeled as integer programming problems. The pioneering work in this area was done by Bowman (5), Wagner (53), and Manne (37). All three of these researchers have presented a somewhat restricted integer programming formulation of the job-shop scheduling problem. However, due to the large size of the integer programming problems which resulted from the proposed formulations for even a small job-shop problem and the limitations of algorithms for the solution of integer programming problems, this work has not led to a practical method for solving job-shop problems. As an example of the size of the resulting integer programming problem, a job-shop with 4 machines and 10 jobs requires 220 variables and 390 restraint equations (13). One significant research effort has employed integer programming for the solution of scheduling problems. This study by Wagner and Story (52) and Wagner and Giglio (21) was for the 3-machine flow-shop with six jobs. It was found in this investigation that the number of iterations required for the solution of these problems by integer programming often was of the same order of magnitude as the total number of permutations, 720, of the jobs.

An integer programming formulation of the job-shop scheduling problem which appears to be an improvement upon previously suggested models has been made by Pritsker, Watters, and Wolfe (47). Their model is a zero-one integer programming formulation which is more general than previous models, and it provides some computational advantages over previous models by reducing the number of variables and equations required for the formulation. However, models such as this will not become a practical means for obtaining optimal solutions to scheduling problems until considerable improvement is made in the algorithms available for the solution of integer programming problems.

Branch-and-bound techniques for obtaining solutions to machine scheduling problems have been suggested by a number of researchers. Ignall and Schrage (27) suggested a method which they applied to flow-shop problems with 2 and 3 machines. Brooks and White (6) and Greenberg (22) have presented similar techniques for the implicit enumeration of all feasible solutions of a job-shop scheduling problem. All of the above studies reached similar conclusions concerning the feasibility of branch-and-bound techniques for problems of realistic size. However, branch-and-bound techniques can be used to solve scheduling problems of very small dimension. As an example, the largest problem that Greenburg (22) attempted consisted of 3 jobs and 2 machines. A series of nineteen linear programming problems had to be solved to obtain the solution to this scheduling problem.

A more recent branch-and-bound method for the solution of machine scheduling problems has been proposed by Charlton and Death (8). They discussed the underlying graphical nature of machine-scheduling problems and proposed a branch-and-bound method based upon the graphical nature of the problem. Due-dates were handled by constraints. The method was called general. However, it fails to produce a feasible schedule if it is not possible to satisfy all due-date constraints which is the usual case in practical scheduling problems. The solution was presented for a 2-job, 2-machine scheduling problem where the objective was to minimize maximum flow-time. Charlton and Death concluded that the method was not feasible for problems of a stochastic nature or for deterministic problems of a practical size. They suggested that alternative methods such as simulation must be used for these type problems.

The possibility of solving the job-shop problem using an analogue computer was investigated by Zimmermann and Pfaffenzeller (54). The 3-machine, 3-job problem was investigated and it was concluded that the computational feasibility of larger problems is doubtful.

Still another approach to solving the job-shop problem is to generate all possible schedules. However, this is impossible because there are infinitely many schedules for the job-shop problem since idle-time can be inserted into any given schedule in infinitely many ways.

It is not necessary to generate all feasible schedules though because there are sets of schedules which dominate all other schedules. For the infinite set of all schedules that possess identically the same ordering of the operations on each machine, there is a unique schedule called a "semi-active schedule" (45) which dominates all other schedules in the set for any regular measure of performance. A "regular measure of performance" is a value to be minimized that can be expressed as a function of the job completion-times and which increases only if at least one of the job completion-times increases. Thus, only a finite number of schedules must be considered because there is only one semiactive schedule for each ordering of the operations, and there are, of course, only a finite number of orderings. Unfortunately the number is still very large so that solution by exhaustive enumeration and comparison is not feasible even for relatively small problems and large computers. A slight reduction in the number of schedules which must be examined when solving a job-shop problem by exhaustive enumeration is possible by restricting attention to active schedules. An "active" schedule is a schedule in which it is not possible to perform a left-shift on any operation (20). A "left-shift" of an operation is any decrease in the time at which the operation starts that does not require an increase in the starting-time of any other operation. It is sufficient to consider only the set of active schedules which is usually a small and proper subset of the set of semiactive schedules. However, there is still a very large number. A third subset of all possible schedules is the set of nondelay schedules. A "nondelay" schedule is a schedule such that there is no instance in which a job is delayed when the machine that is to process the next operation is available and idle (45). Nondelay schedules are a subset of the active schedules, but not a dominating subset. It is not true that for every problem there is an optimal schedule among the nondelay schedules. However, nondelay schedules are easy to generate and there is strong empirical evidence that nondelay schedules are on the whole better than the remainder of the active schedules (13). Therefore, if a sampling approach is taken to solving the job-shop problem such as Heller (24) used, it may be advantageous to sample from the nondelay schedules rather than from the active schedules even though the infinitesimal probability of an optimal solution may be forfeited.

The properties of active and nondelay schedules were studied by Bakhru and Rao (4). They used a set of 11 problems with 10 jobs and 6 facilities each. Each job had exactly 6 operations, one on each machine, in a randomly determined order. For each of these 11 problems a random sample of 50 active schedules and a random sample of 50 nondelay schedules were generated. Using either mean flow-time or maximum flow-time as measures of effectiveness, the nondelay schedules were better than the active schedules. This suggested that if a "good" schedule is to be found by selecting the best of a sample of randomly generated schedules, it would be more efficient to sample from the population of nondelay schedules.

Active and nondelay schedules have also been investigated by Jeremiah, Lalchandani, and Schrage (32). In this study, instead of generating a sample of schedules at random, scheduleable operations were selected according to some attribute of the jobs and/or machines. The results of this study provided additional evidence that nondelay schedules are superior to active schedules.

The area of investigation that has received the most attention for the simple n-job, m-machine job-shop problem is that of experimental investigation. With the advent of the digital computer, it became possible to simulate the jobs, machines, and scheduling procedures of a job-shop. Some of the earliest investigators to recognize the potential of the digital computer and to use it as a tool for investigating the job-shop were Jackson, Nelson, and Rowe at UCLA (30) and (31) and Baker and Dzielinski at IBM (3). Since then there have been a large number of experimental investigations of the job-shop conducted. The best known of these and the most extensive are the works of R. W. Conway (10) and (11), E. LeGrande (36), and Y. R. Nanot (43).

Most of the investigations conducted have compared the relative effectiveness of various dispatching rules by means of a job-shop simulation. A computer program that produces a schedule by the use of a dispatching procedure is called a "job-shop simulation" (13). A "dispatching procedure" is a method which includes a "priority" mechanism, a numerical attribute of a job, to select a job to be processed next from among the jobs competing for machine assignment (13). Some researchers define "priority sequencing rule" to have the same meaning as "dispatching procedure." The dispatching procedures which have been investigated to date include but are not limited to:

1. Random.
2. First-come, first-served.
3. Last-come, first-served.
4. Shortest processing-time for imminent operation (SPT).
5. Least slack where slack is defined as the due-date minus remaining expected processing-time minus current date.
6. Least slack/remaining operation (S/O).
7. Truncated SPT: jobs which have become critical take precedence.
8. Linear combination of SPT and S/O.
9. Look ahead: select job which will go to shortest queue.

10. Fewest remaining operations.

11. Longest imminent operation.

12. Least work remaining.

13. Most work remaining.

14. Weighted random combinations of rules.

Most of this work has focused on minimizing the make-span of all jobs. That is, there have been a number of attempts to determine which rule minimizes the mean flow-time of all jobs or minimizes the maximum flow-time of all jobs or minimizes the mean number of jobs in the shop. Not nearly as much work has been done on meeting due-dates. Conway's investigations (10), (11) considered a job-shop with the following characteristics:

1. Nine machine groups each having a single machine.
2. Poisson job arrival process.
3. Exponential service times with a mean of 1.0.
4. Shop utilization level of approximately 90%.
5. Random job routing.
6. 10,000 jobs.

Conway (10) compared many dispatching rules and combinations of rules and determined that when due-dates were not considered, the shortest-processing-time rule dominated the other rules. He also concluded that there is no basis to believe that highly precise estimates of processing-time are required for scheduling purposes. These investigations showed that the shortest-processing-time rule is the best rule known for minimizing the mean flow-time of jobs through a simple job-shop. The shortest-processing-time rule minimizes the mean of the completion-time distribution of the jobs. This study also showed that good results could be obtained by a linear combination of the shortest-processing-time rule and a look ahead rule. The least slack/remaining operation rule was found to minimize the number of jobs tardy (jobs with positive lateness). Also a linear combination of shortest-processing-time and least slack/remaining operation was very effective at minimizing the number of jobs tardy.

Conway (11) compared the performance of various priority dispatching rules for meeting due-dates. He concluded that the performance of a job-shop in meeting due-dates is a function of how the due-dates were assigned as well as the dispatching rule used. He concluded that generally the least slack/remaining operation rule is the best of the standard due-date rules although for very heavy shop loads, the shortest-processing-time rule performed as well as the least slack/remaining operation rule. In general the least slack/remaining operation rule was found to produce schedules which exhibited the smallest value of lateness variance. Also the least slack/remaining operation rule minimized the number of jobs tardy.

LeGrande of Hughes Aircraft also performed extensive investigations of the job-shop and the performance of various sequencing rules using actual operating data in a computer simulation (36). One hundred and fifteen machine groups and 3000 jobs were simulated. The utilization level was approximately 60-70 per cent. LeGrande's study confirmed most of the results of Conway's investigations.

Nanot (43) compared priority sequencing rules by simulating job-shops with 2, 4, and 8 machine groups each with a single machine. He assumed Poisson job arrivals and exponential service times. Shop utilization varied from 60 to 95%. Due-dates were not considered. The results of this study are consistent with those of Conway and LeGrande.

A modified shortest-processing-time rule was investigated by Eilon and Cotterill (15) in which each machine had two queues. One queue consisted only of jobs with zero or negative slack. The second queue contained jobs with positive slack. The queue with zero and negative slack jobs had priority over the other queue. Both queues were sequenced using the shortest-processing-time rule. This procedure provided slightly better results than the shortest-processing-time rule without the modification.

The use of expediting has been simulated by Hottenstein (26). Expediting was found to increase the average time a job spends in the shop and reduce the variability of shop performance.

Probabilistic dispatching, in which some attribute of the machine or job is used to randomize the selection of the next job to be processed from the set of all scheduleuleable jobs, has been investigated by Jeremiah, Lalchandani, and Schrage (32) and Nugent (45). With some randomness in the process of selecting the next job to be scheduled, different schedules will be generated when the same selection process is repeatedly applied to the same problem. The results of these studies indicate that probabilistic dispatching generates slightly better schedules for minimizing maximum flow-time or minimizing mean flow-time than does deterministic dispatching, but the improvement is not very great and considerable effort is required.

Heuristic procedures have been employed in studies by Fisher and Thompson (17) and by Crabill (14). Both studies were rather limited. The former employed probabilistic learning combinations of simple job-shop sequencing rules while the latter used a job-at-a-time adjusting procedure to improve flow-times. Gere (19) also employed several adjusting procedures which depended on conditions of the machines and jobs. Conclusions drawn were that an unbiased random combination of scheduling rules is usually somewhat better than any of them taken separately, learning is possible, and combinations of rules are usually more effective than the individual rules employed separately.

CHAPTER III

RESEARCH PROGRAM

Outline of Program

Since no method is currently available which will find optional solutions to job-shop scheduling problems which are of reasonable dimension, there exists a real need for a method of generating "good" schedules. Shortest-processing-time and least slack/remaining operation sequencing rules and certain combinations of these and other simple rules have been found to be fairly effective for such things as minimizing mean flow-time or number of jobs late. However, there is room for improvement here. Also there is a need for a method which produces good schedules that considers not only due-dates but also the cost of delay for the jobs being processed. It is common in defense contracts to have an incentive clause which specifies a bonus for early completion of a job and/or a penalty cost for late completion. Since these costs are a function of the shop schedule, they can be reduced by improving the schedule. There are relatively few published papers on the problem of sequencing jobs in a job-shop so as to complete as many jobs as possible by the specified due-dates and only a fraction of these papers consider the delay costs of the jobs to be processed. Some papers that do consider the cost of delay of the jobs being processed as well as due-dates have already been identified. These were McNaughton (39), Lawler (35), Moore (42), Maxwell (38), and Pritsker, Watters, and Wolfe (47). Two additional papers of interest have been published by Fabrycky and Shamblin (16) and Holt (25).

Fabrycky and Shamblin (16) have developed the only dispatching procedure published to date which does not assume that either deterministic flow-times or processing-times are known for each job and each facility which processes the job. This dispatching procedure is a function of job due-dates and flow-time distributions for the facilities. Each facility is assumed to have a flow-time distribution which is normally distributed with a known mean and standard deviation. Penalty and bonus costs are not considered by the dispatching rule. When two or more jobs are competing for the same facility, this dispatching rule assigns highest priority to the job which has the lowest probability of being completed by its due-date. This dispatching rule was compared to the first-come, first-served rule and found to produce better schedules. However, the first-come, first-served rule has been shown to compare unfavorably with the shortest-processing-time, least slack/remaining operation, and other rules which are among the better rules available. Therefore, an evaluation of this rule should include a comparison of this rule with one of the more effective rules which considers due-dates such as least slack/remaining operation.

Holt (25) suggested that the cost of delaying jobs held in queue while the selected job is being processed should be considered. That is, for each job in a queue available for processing, compute the total delay costs for all other jobs in the queue if delayed for the processing-time of the job being considered. Then select that job with the smallest total delay cost. If C_{ij} is the total cost of delay caused by processing Job i on Machine j, P_{ij} is the processing-time of Job i on Machine j, W_i is the cost per unit time for delaying Job i, and q_j is the set of all jobs available for processing on Machine j, then

$$C_{ij} = P_{ij} \sum_{\substack{k \subset q_j \\ k \neq i}} W_k$$

A cost estimate such as this could be computed for each of the jobs available for processing when a machine becomes available and this cost could be used as a priority number where the lowest number is given preference. This type of rule has not been implemented or tested experimentally nor has a good method for computing W_k been developed, but some interesting properties are apparent. If the processing-time of a job is very short, it would not delay the remaining jobs very long and would have a low total cost of delay. Therefore, some similarities should exist between this rule and the shortest-processing-time rule. However, if a job had a very high delay cost, the total cost of delay would be minimized by processing this job immediately and delaying the other jobs. Therefore, it appears that a rule of this type would exhibit some of the properties necessary for a dispatching procedure which generates "good" schedules.

The works of Fabrycky and Shamblin (16) and Holt (25) are of interest because although they have not provided a feasible method of generating "good" schedules for a job-shop where due-dates and delay costs should be considered, they have definitely taken a step in that direction. Of particular significance is the use of a flow-time probability distribution by Fabrycky and Shamblin to compute the probability of a job being completed by its due-date and the suggestion by Holt that sequencing rules might well consider the cost of delaying competing jobs for the processing-time of the job being processed first.

In this dissertation, three stochastic dispatching procedures are proposed which are designed to sequence jobs in a manner which tends to minimize the total cost of processing all jobs to completion. Costs which are considered by the sequencing rules are penalty cost functions for tardiness and bonus functions for earliness. Penalty and bonus functions are assumed to be linear for the purposes of this paper. However, any monotonically nondecreasing functions can be used with these dispatching procedures.

All dispatching procedures developed assume that the processing-time distribution of each machine is known. Processing-time was chosen as a basis for the development of these procedures instead of flow-time because flow-time is sensitive to shop loading which is frequently very unstable in a true job-shop. Since processing-time is not a function of shop loading, historical processing-time data can be used with much greater confidence than can historical flow-time data. Any distribution can be used for processing-times. However, it is assumed for the purposes of this paper that the processing-time distributions are approximately normal.

One of the dispatching procedures developed is based on an expected savings formulation while the other procedures are based on expected cost formulations. The expected savings dispatching procedure was compared to the expected cost procedures on small job-shop problems with manual simulations, and the expected savings procedure was found to be much weaker in its ability to discriminate among competing jobs than the expected cost procedures. Therefore, simulation programs were developed for the experimental testing of only the expected cost dispatching procedures.

Dispatching procedures such as the ones presented in this paper must in some way be verified as to their ability to generate "good" schedules. If the schedules produced by these dispatching procedures could be compared to known optimal solutions of a number of problems of reasonable size and complexity, then this would demonstrate the effectiveness of these procedures. However, if optimal solutions to these problems were readily available for comparison, then there would be no need for the development of the dispatching procedures as the problem would already be solved. While optimal solutions to problems of reasonable size are not easily found, optimal solutions can be found for very small restricted problems by either total enumeration of all feasible schedules or the use of mathematical programming techniques. Because some small problems can be solved through the use of integer programming techniques, the job-shop scheduling problem with penalty and bonus costs is formulated as an integer programming problem. However, in order to do this, the assumption must be made that the processing-time of a job on a machine is known and is deterministic. This of course is an unrealistic assumption but a necessary one if the problem is to be formulated as an integer programming problem. There are no computationally feasible stochastic integer programming techniques currently available.

The primary method of verifying the validity of the expected cost dispatching procedures is experimental verification. This is the only practical means of testing the performance of the dispatching rules on a number of problems of reasonable size. Simulation programs were developed which simulate the activity of a job-shop which uses the expected cost dispatching procedures to determine the order of processing for the competing jobs at each facility in the shop. Also, a simulation program was developed which simulates the performance of the job-shop using the least slack/remaining operation rule to make sequencing decisions. The least slack/remaining operation rule was chosen because it has been shown to be the best performing of the simple due-date rules, and it has been extensively simulated and compared to many other dispatching rules by a large number of researchers (10), (11), (36), and (15). It should be pointed out that the least slack/remaining operation rule assumes known deterministic processing-times and known due-dates. It does not consider the distribution of processing-times nor penalty or bonus costs. Since there is no rule currently available which does consider these things, the least slack/remaining operation rule is the best choice because it has demonstrated good performance and is well known. If the expected cost dispatching rules exhibit an ability to generate schedules which are significantly better than schedules generated by the least slack/remaining operation rule in terms of the total cost of processing all jobs to completion, then the effectiveness of the expected cost dispatching rules will be verified.

In addition to the verification of the expected cost dispatching procedures by comparison to optimal solutions of very small problems and by comparison to the least slack/remaining operation rule by means of simulations for larger problems, one of the expected cost rules is proven to be optimal in some special cases. This does not imply that this expected cost procedure produces optimal solutions for all problems. However, it does indicate that the procedure generates schedules which are likely to be optimal.

Although the primary purpose of this paper is the development and verification of effective probabilistic sequencing rules for minimizing the total costs of processing all jobs to completion in a job-shop, the simulations developed are general enough to handle a wide variety of problems without the usual simplifying assumptions of the simple job-shop. Such flexible simulations have not previously been available, and they can be an extremely valuable management and research tool.

Integer Programming Formulation

An integer programming formulation of a simple job-shop problem with due-dates and linear penalty cost functions is presented because integer programming techniques can be used to generate optimal schedules for small job-shop problems with certain simplifying assumptions. The notation used will follow Manne (37). The assumptions made in this formulation are:

1. Jobs are strictly ordered sequences of operations.
2. Assembly is not allowed.
3. Partition is not allowed.
4. Each operation can be performed by only one facility in a shop.
5. Once an operation is started on a facility, it must be processed through to completion.
6. A job may be in process on at most one facility at a time.
7. Each job has a due-date associated with it.
8. Each job has associated with it a penalty cost for delay directly proportional to the tardiness of the job.
9. The processing-time of each job on all applicable facilities is deterministic and known.
10. A facility may process only one job at a time.
11. All jobs are immediately available for processing.

12. All facilities are continuously available.

In order to formulate this problem, the following parameters and variables are defined:

Parameters

P_{ik} = Processing-time of Job i on Machine k.

r_{ijk} = 1 if Job i has its j-th operation on Machine k.

r_{ijk} = 0 otherwise.

d_i = Due-date of Job i.

M = A very large constant.

m = Number of machines.

n = Number of jobs.

W_i = Penalty cost in dollars per day for tardiness of Job i.

Variables

T_{ik} = Starting time of Job i on Machine k.

Y_{ijk} = 1 if Job i precedes Job j on Machine k (not necessarily directly).

Y_{ijk} = 0 otherwise.

The constraints can now be written.

Noninterference constraints:

$$(M+P_{jk})\, Y_{ijk} + (T_{ik} - T_{jk}) \geq P_{jk} \tag{1a}$$

Constraint (1a) prevents Jobs i and j from being processed simultaneously on Machine k if Job j precedes Job i.

The number of inequalities is $m \cdot \binom{n}{2} = m\frac{n(n-1)}{2}$.

$$(M+P_{ik})(1-Y_{ijk}) + (T_{jk}-T_{ik}) \geq P_{ik} \qquad (1b)$$

Constraint (1b) prevents Jobs i and j from being processed simultaneously on Machine k if Job i precedes Job j.

The number of inequalities is $m \cdot \binom{n}{2} = m\frac{n(n-1)}{2}$.

Because jobs are strictly ordered sequences of operations, there must be operation precedent constraints.

The starting-time of the j-th operation on the i-th job can be written as:

$$\sum_{k=1}^{m} r_{ijk}\, T_{ik} = r_{ij1}\, T_{i1} + r_{ij2}\, T_{i2} + \cdots + r_{ijm}\, T_{im}$$

The processing-time of the j-th operation on the i-th job can be written as:

$$\sum_{k=1}^{m} r_{ijk}\, P_{ik} = r_{ij1}\, P_{i1} + r_{ij2}\, P_{i2} + \cdots + r_{ijm}\, P_{im}$$

For all but the last operation on a job, the starting-time plus the processing-time must be less than or equal to the starting-time of the next operation.

$$\sum_{k=1}^{m} r_{ijk}\, T_{ik} + \sum_{k=1}^{m} r_{ijk}\, P_{ik} \leq \sum_{k=1}^{m} r_{i,j+1,k}\, T_{ik} \qquad (2)$$

The number of inequalities is $(m-1)n$.

Constraints on Variable T_{ik} :

$$T_{ik} \geq 0 \qquad (3)$$

The number of inequalities is $m \cdot n$.

Constraints on Variable Y_{ijk} :

$$Y_{ijk} = 0 \text{ or } 1 \qquad (4)$$

The number of inequalities is $m \cdot \binom{n}{2} = m\frac{n(n-1)}{2}$.

The solution to this problem must satisfy constraints (1), (2), (3), and (4) above.

The objective is to minimize the total penalty cost of processing all jobs to completion. To write an objective function for total penalty cost, let C_i be the completion-time of Job i, and for simplicity of notation, assume that each job is processed by exactly m-machines. The jobs must be processed by each machine. However, any ordering of the operations is possible.

C_i can be written as the starting-time of the last operation on Job i plus the processing-time of the last operation on Job i. The last operation will always be the m-th operation if each job is processed by all machines.

The starting-time of the last operation can be written as:

$$\sum_{k=1}^{m} r_{imk} T_{ik} = r_{im1} T_{i1} + r_{im2} T_{i2} + \cdots + r_{imm} T_{im}$$

The processing-time of the last operation can be written as:

$$\sum_{k=1}^{m} r_{imk} P_{ik} = r_{im1} P_{i1} + r_{im2} P_{i2} + \dots + r_{imm} P_{im}$$

therefore,

$$C_i = \sum_{k=1}^{m} r_{imk} T_{ik} + \sum_{k=1}^{m} r_{imk} P_{ik}$$

$$\text{for} \quad i = 1, 2, \dots, n$$

Let I_i = tardiness of Job i.

Then $I_i = 0$ if $C_i \leq d_i$

$I_i = C_i - d_i$ if $C_i > d_i$

Therefore, I_i can be written as:

$I_i = \text{Maximum}\ (0,\ C_i - d_i)$

Z, the total penalty for all jobs is:

$$Z = \sum_{i=1}^{n} I_i W_i$$

$$Z = \sum_{i=1}^{n} [\max(0, C_i - d_i)]\ W_i \qquad (5)$$

and the objective is to minimize Z subject to constraints (1), (2), (3), and (4).

Most integer programming algorithms are for the solution of integer programming problems with linear constraints and linear objective functions. Therefore, it is undesirable to leave the objective function in the non-linear form shown in equation (5). It is possible through the use of additional constraints to replace equation (5) with an equivalent linear objective function. To do this, require that:

$$I_i \geq 0 \qquad \text{for all } i \tag{6}$$

There are n such inequalities.

Also require that:

$$I_i \geq C_i - d_i \qquad \text{for all } i$$

which can be rewritten as

$$I_i \geq \sum_{k=1}^{m} r_{imk}(T_{ik} + P_{ik}) - d_i \tag{7}$$

for all i. There are n such inequalities.

With the addition of inequalities (6) and (7), the objective function becomes:

$$\text{Minimize} \quad Z = \sum_{i=1}^{n} I_i W_i \tag{8}$$

Therefore, the linear integer programming formulation of the simple job-shop problem with due-dates and linear penalty cost functions is the minimization of the objective function in equation (8) subject to constraints (1), (2), (3), (4), (6), and (7). If a simple job-shop problem with 20 jobs and

6 facilities is formulated in this manner, the formulation would require 2420 variables and 7000 constraints. A problem of this size cannot be solved by currently available integer programming techniques. Only a few seconds of computer time are required to generate feasible schedules for problems with 20 jobs and 6 machines if simulation techniques are used.

The integer programming formulation presented here can be extended to incorporate bonus costs by rewriting the objective function so as to include the bonus. Equation (5) then becomes:

$$Z = \sum_{i=1}^{n} [\max(0, C_i - d_i)]W_i - \sum_{i=1}^{n} [\max(0, d_i - C_i)]B_i \qquad (5')$$

where B_i is the bonus per unit of time that Job i is completed early. In order to rewrite this objective function as a linear function, the following theorem will be needed.

Theorem: maximum $(0,-A) = A -$ maximum $(0,A)$

Proof: To prove this theorem, three cases must be considered:

$A < 0, \quad A = 0, \quad A > 0$.

1. $A < 0$: maximum $(0,-A) = A$

$A -$ maximum $(0,A) = A - 0 = A$

Therefore, maximum $(0,-A) = A$ $-$ maximum $(0,A)$

2. $A = 0$: maximum $(0,-A) = 0$

A - maximum $(0,A) = 0 - 0 = 0$

Therefore, maximum $(0,-A) = A$ - maximum $(0,A)$

3. $A > 0$: maximum $(0,-A) = 0$

A - maximum $(0,A) = A - A = 0$

Therefore, maximum $(0,-A) = A$ - maximum $(0,A)$

Using this theorem, Equation (5') may be rewritten:

$$Z = \sum_{i=1}^{n} [\max(0, C_i - d_i)] \, W_i - \sum_{i=1}^{n} [(C_i - d_i) - \max(0, C_i - d_i)] \, B_i$$

$$Z = \sum_{i=1}^{n} (W_i + B_i)\max(0, C_i - d_i) - \sum_{i=1}^{n} B_i(C_i - d_i)$$

The objective function can now be written as:

$$\text{Minimize } Z = \sum_{i=1}^{n} (W_i + B_i) I_i - \sum_{i=1}^{n} B_i(C_i - d_i) \qquad (8')$$

if constraints (6) and (7) are still required. Thus the linear integer programming formulation of the simple job-shop problem with penalty and bonus cost functions is the minimization of (8') subject to constraints (1), (2), (3), (4), (6), and (7).

The assumption that each job must be processed by all m machines can also be relaxed. This is done by:

1. Setting $P_{ik} = 0$ if Job i is not processed by Machine k.
2. Letting m_i be the number of operations on Job i and restricting the range of the second subscript on r_{ijk} so that $j = 1, 2, \ldots, m_i$.

Dispatching Procedures

Three stochastic dispatching procedures for scheduling jobs in a job-shop so as to minimize the total cost of processing all jobs to completion have been developed. All three of these dispatching procedures assume known processing-time distributions for each shop facility. All procedures have as parameters the due-dates of the jobs, the penalty costs for tardiness, the bonus payments for earliness, the required technological processing sequence for each job, and the processing-time distributions for the shop facilities. For the purposes of this dissertation, it is assumed that the processing-time distributions are approximately normal.

One of the dispatching procedures investigated uses an expected savings formulation to determine processing priorities for competing jobs. The other dispatching procedures investigated use expected cost formulations for the determination of processing priorities. The objective of the expected savings dispatching procedure is to sequence competing jobs in a manner which will maximize expected savings for the sequence and in so doing minimize the total cost of processing all jobs to completion. The objective of the expected cost procedures is to sequence competing jobs in a way which minimizes expected costs for the sequence and in so doing minimizes the total cost of processing all jobs to completion.

Expected Savings Formulation

This procedure for the dispatching of competing jobs is based upon the difference between two expected cost calculations. Whenever a facility of the shop becomes available for processing, the expected savings per day for processing each job immediately and not delaying it is computed for all jobs competing for that facility. The jobs are assigned to the facility for processing in the order of non-increasing expected savings. The job with the greatest expected savings would be processed first. Each time a facility completes the processing of a job and is available for processing another job, the expected savings for the immediate processing of each competing job is recomputed. This is necessary because in a real shop or in a simulation time is constantly changing, and expected savings is a function of time. Also, new jobs may have joined the facility's queue, and expected savings must be computed for the new arrivals as well as the jobs which were previously delayed.

In order to compute the expected savings per day which would result if Job i were processed immediately instead of being delayed for one day, let

M_i = {Set of machines in the technological processing sequence of Job i which have not yet processed Job i.}

$\overline{X}_{ij}$ = Expected processing-time of Job i on Machine j.

S_{ij} = Standard deviation of the processing-time of Job i on Machine j.

d_i = Due date of Job i.

P_i = Penalty cost per day for tardiness of Job i.

B_i = Bonus per day for earliness of Job i.

$\overline{\overline{X}}_i$ = Total expected processing-time of Job i.

S_i^P = Pooled standard deviation of total processing-time of Job i.

α_D = Probability of a job being D days tardy.

θ_D = Probability of a job being D days early.

EC_i = Expected cost of processing Job i immediately.

EC_i' = Expected cost of processing Job i if it is delayed for one day.

ΔEC_i = Expected savings per day for processing Job i immediately instead of delaying it.

K_D = $\dfrac{X - \overline{\overline{X}}}{S^P}$, the standard normal deviate for a job D days early or tardy.

γ_D = $\int_{-\infty}^{K_D} f(x)\,dx$ where $f(x)$ is the standard normal distribution with a mean of 0.0 and a standard deviation of 1.0.

The expected cost formulation for Job i is

$$EC_i = P_i \sum_{D=1}^{\infty} \alpha_D D - B_i \sum_{D=1}^{\infty} \theta_D D$$

which is the penalty per day tardy multiplied by the expected number of days tardy minus the bonus per day early times the expected number of days early.

Expected savings per day may now be defined as:

$$\Delta EC_i = EC_i' - EC_i$$

which is the expected cost of Job i if processing is delayed for one day minus the expected cost of Job i if processing begins immediately.

ΔEC_i is computed for each job competing for a facility available for the processing of these jobs. The job with the greatest expected savings per day, ΔEC_i , is processed first. This is done for each shop facility as the facilities complete jobs already in progress and become available for the processing of additional jobs.

As an example of how this procedure is applied to an actual problem, consider a machine scheduling problem with two jobs and two machines. Technological ordering requires that both jobs must be processed by both machines in the order 1-2. For simplicity assume that the day the jobs become available for processing is day one.

TABLE 1

JOB CHARACTERISTICS

Job	Machine Sequence	B_i	P_i	d_i
1	1 - 2	10	20	6
2	1 - 2	5	10	5

TABLE 2

PROCESSING-TIME DISTRIBUTIONS

Job	$\overline{X}_{i1}$	S_{i1}	$\overline{X}_{i2}$	S_{i2}
1	2	.707	1	.707
2	3	.707	2	.707

At the beginning of this manual simulation both Job 1 and Job 2 are competing for processing on Machine 1. There are no jobs competing for processing at this time on Machine 2. Therefore, Machine 2 will initially be idle. To make a decision as to which job should be processed first on Machine 1, ΔEC_1 and ΔEC_2 are computed. If $\Delta EC_1 > \Delta EC_2$, then Job 1 is processed first since it offers the greatest savings. If $\Delta EC_1 < \Delta EC_2$, then Job 2 is processed first. If $\Delta EC_1 = \Delta EC_2$, the job to be processed first can be chosen randomly or on some other basis such as earliest due-date.

In order to compute ΔEC_1 and ΔEC_2, the total expected processing-time of processing each job to completion and the pooled standard deviation of the total processing-time distribution for each job must be computed.

$$\bar{\bar{X}}_i = \sum_{j \in M_i} \bar{X}_{ij} \quad \text{and} \quad S_i^P = \sqrt{\sum_{j \in M_i} S_{ij}^2}$$

Therefore,

$$\bar{\bar{X}}_1 = \bar{X}_{11} + \bar{X}_{12} = 2 + 1 = 3$$

$$\bar{\bar{X}}_2 = \bar{X}_{21} + \bar{X}_{22} = 3 + 2 = 5$$

$$S_1^P = \sqrt{S_{11}^2 + S_{12}^2} = \sqrt{(.707)^2 + (.707)^2} = 1.0$$

$$S_2{}^P = \sqrt{S_{21}^2+S_{22}^2} = \sqrt{(.707)^2 + (.707)^2} = 1.0$$

Recall that $\Delta EC_1 = EC_1{}' - EC_1$

and $$EC_1 = P_1 \sum_{D=1}^{\infty} \alpha_D D - B_1 \sum_{D=1}^{\infty} \theta_D D$$

To determine EC_1 the expected number of days tardy,

$$\sum_{D=1}^{\infty} \alpha_D D ,$$

and the expected number of days early,

$$\sum_{D=1}^{\infty} \theta_D D ,$$

if Job 1 is processed immediately must first be computed.

TABLE 3

EXPECTED NUMBER OF DAYS TARDY FOR
JOB 1 IF PROCESSED IMMEDIATELY

D	$K_D = \frac{X-\bar{\bar{X}}}{S^P}$	$\gamma_D = \int_{-\infty}^{K_D} f(x)dx$	$\alpha_D = \gamma_D - \gamma_{D-1}$	$\alpha_D D$
0	$\frac{6-3}{1} = 3$	.999	---	---
1	$\frac{7-3}{1} = 4$	1.000	.001	.001

$$\sum_{D=1}^{\infty} \alpha_D D = .001$$

TABLE 4

EXPECTED NUMBER OF DAYS EARLY FOR JOB 1 IF PROCESSED IMMEDIATELY

D	$K_D = \frac{X-\overline{\overline{X}}}{S^P}$	$\gamma_D = \int_{-\infty}^{K_D} f(x)dx$	$\theta_D = \gamma_D - \gamma_{D+1}$	$\theta_D D$
0	$\frac{6-3}{1} = 3$	.999	.022	.000
1	$\frac{5-3}{1} = 2$	.977	.136	.136
2	$\frac{4-3}{1} = 1$	.841	.341	.682
3	$\frac{3-3}{1} = 0$	.500	.341	1.023
4	$\frac{2-3}{1} = -1$	.159	.136	.544
5	$\frac{1-3}{1} = -2$	.023	.022	.110
6	$\frac{0-3}{1} = -3$	.001	.001	.006
7	$\frac{-1-3}{1} = -4$	.000	---	---

$$\sum_{D=1}^{\infty} \theta_D D = 2.501$$

The expected cost of processing Job 1 immediately, EC_1, can now be computed.

$$EC_1 = P_1 \sum_{D=1}^{\infty} \alpha_D D - B_1 \sum_{D=1}^{\infty} \theta_D D$$

$$= (20)(.001) - (10)(2.501)$$

$$= -24.99$$

To determine EC_1', the expected number of days tardy and the expected number of days early if Job 1 is delayed for one day must first be computed. This is done in the same manner that these computations were performed for the immediate processing of Job 1 except that the time until the due-date is reduced by one day.

TABLE 5

EXPECTED NUMBER OF DAYS TARDY FOR JOB 1 IF DELAYED FOR ONE DAY

D	K_D	γ_D	α_D	$\alpha_D D$
0	2	.977	---	---
1	3	.999	.022	.022
2	4	1.000	.001	.002

$$\sum_{D=1}^{\infty} \alpha_D D = .024$$

TABLE 6

EXPECTED NUMBER OF DAYS EARLY FOR JOB 1 IF DELAYED FOR ONE DAY

D	K_D	γ_D	θ_D	$\theta_D D$
0	2	.977	.136	.000
1	1	.841	.341	.341
2	0	.500	.341	.682
3	-1	.159	.136	.408
4	-2	.023	.022	.088
5	-3	.001	.001	.005
6	-4	.000	---	---

$$\sum_{D=1}^{\infty} \theta_D D = 1.524$$

The expected cost of processing Job 1 if delayed for one day, EC_1', can now be computed.

$$EC_1' = P_1 \sum_{D=1}^{\infty} \alpha_D D - B_1 \sum_{D=1}^{\infty} \theta_D D$$

$$= (20)(.024) - (10)(1.524)$$

$$= -14.76$$

Therefore, the expected savings per day for not delaying Job 1 is:

$$\Delta EC_1 = EC_1' - EC_1$$

$$= -14.76 - (-24.99)$$

$$= +10.23$$

ΔEC_2 is computed in the same manner as ΔEC_1 except that the characteristics of Job 2 are substituted for those of Job 1 in the computations.

$$\Delta EC_2 = EC_2' - EC_2$$

$$EC_2 = P_2 \sum_{D=1}^{\infty} \alpha_D D - B_2 \sum_{D=1}^{\infty} \theta_D D$$

TABLE 7

EXPECTED NUMBER OF DAYS TARDY FOR JOB 2 IF PROCESSED IMMEDIATELY

D	K_D	γ_D	α_D	$\alpha_D D$
0	0	.500	---	---
1	1	.841	.341	.341
2	2	.977	.136	.272
3	3	.999	.022	.066
4	4	1.000	.001	.004

$$\sum_{D=1}^{\infty} \alpha_D D = .683$$

TABLE 8

EXPECTED NUMBER OF DAYS EARLY FOR JOB 2 IF PROCESSED IMMEDIATELY

D	K_D	γ_D	θ_D	$\theta_D D$
0	0	.500	.341	.000
1	-1	.159	.136	.136
2	-2	.023	.022	.044
3	-3	.001	.001	.003
4	-4	.000	---	---

$$\sum_{D=1}^{\infty} \theta_D D = .183$$

The expected cost of processing Job 2 immediately is:

$$EC_2 = P_2 \sum_{D=1}^{\infty} \alpha_D D - B_2 \sum_{D=1}^{\infty} \theta_D D$$

$$= (10)(.683) - (5)(.183)$$

$$= 5.915$$

The computation of EC_2' proceeds in the same manner as the computation of EC_2 except that the time until the due-date is reduced by one day.

TABLE 9

EXPECTED NUMBER OF DAYS TARDY FOR JOB 2 IF DELAYED FOR ONE DAY

D	K_D	γ_D	α_D	$\alpha_D D$
0	-1	.159	---	---
1	0	.500	.341	.341
2	1	.841	.341	.682
3	2	.977	.136	.408
4	3	.999	.022	.088
5	4	1.000	.001	.005

$$\sum_{D=1}^{\infty} \alpha_D D = 1.524$$

TABLE 10

EXPECTED NUMBER OF DAYS EARLY FOR JOB 2 IF DELAYED FOR ONE DAY

D	K_D	γ_D	θ_D	$\theta_D D$
0	-1	.159	.136	.000
1	-2	.023	.022	.022
2	-3	.001	.001	.002
3	-4	.000	---	---

$$\sum_{D=1}^{\infty} \theta_D D = .024$$

The expected cost of processing Job 2 if delayed for one day is:

$$EC_2' = P_2 \sum_{D=1}^{\infty} \alpha_D D - B_2 \sum_{D=1}^{\infty} \theta_D D$$

$$= (10)(1.524) - (5)(.024)$$

$$= 15.12$$

Therefore, the expected savings per day for not delaying Job 2 is:

$$\Delta EC_2 = EC_2' - EC_2$$

$$= 15.12 - 5.915$$

$$= 9.205$$

Since $10.23 > 9.205$, $\Delta EC_1 > \Delta EC_2$ and the processing of Job 1 immediately and the delay of Job 2 will result in greater expected savings than the immediate processing of Job 2 and the delay of Job 1. Therefore, Jobs 1 and 2 would be processed by Machine 1 in the order 1-2. This delays Job 2 for the processing-time of Job 1. Because of the small size of this problem, the decision to process Jobs 1 and 2 in the sequence 1-2 on Machine 1 determines a unique schedule. No other sequencing decisions are necessary.

To evaluate this schedule is not a simple undertaking since the processing-times of the jobs are stochastic.

One way would be to formulate the total expected costs of all feasible schedules for this problem and compare them. Although this might be possible for this 2-job, 2-machine problem, it would not be a practical method of evaluating the schedules of more complex problems. To enumerate all feasible schedules and compute their total penalty costs, it is assumed that a job is processed by a machine for a period of time equal to the mean of the processing-time distribution. This same assumption will be made later so that expected cost schedules can be compared to least slack/remaining operation schedules.

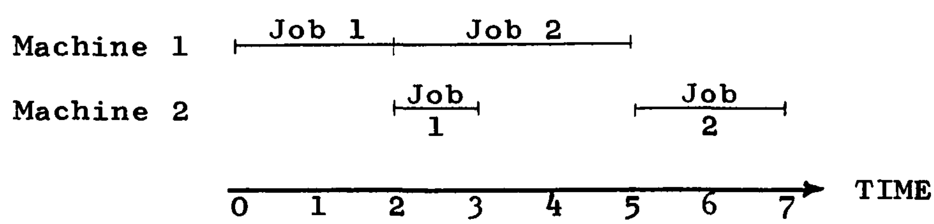

Illustration 1.--Expected Savings Schedule

Job 1 is processed for two days on Machine 1. Then at the completion of Job 1 on Machine 1, Job 2 begins processing on Machine 1, and Job 1 begins processing on Machine 2. At the end of Day 3, Job 1 completes processing on Machine 2, and since Job 1 was not due until Day 6, it is completed three days early. Job 2 completes processing on Machine 1 at the end of Day 5. It then begins processing on Machine 2 which has been idle for two days. Job 2 completes its processing on Day 7 which is two days tardy since it was due on Day 5. Therefore, the total actual cost of this schedule is:

$$
\begin{aligned}
TC = {} & P_1 \text{ (Number of days Job 1 is tardy)} \\
& -B_1 \text{ (Number of days Job 1 is early)} \\
& +P_2 \text{ (Number of days Job 2 is tardy)} \\
& -B_2 \text{ (Number of days Job 2 is early)}
\end{aligned}
$$

$$TC = (20)(0) - (10)(3) + (10)(2) - (5)(0)$$

$$TC = -10$$

It is easy to demonstrate that this is an optimal schedule for this problem. This is done by total enumeration of all feasible schedules. That schedule which has the smallest total actual cost is the optimal schedule. For this problem there are only two feasible schedules which must be considered. One of these schedules is the one obtained by the application of the expected savings dispatching procedure which has a total actual cost of -10. The other schedule is obtained by processing the jobs in the sequence 2-1 on Machine 1. This uniquely determines the remainder of the schedule This schedule is

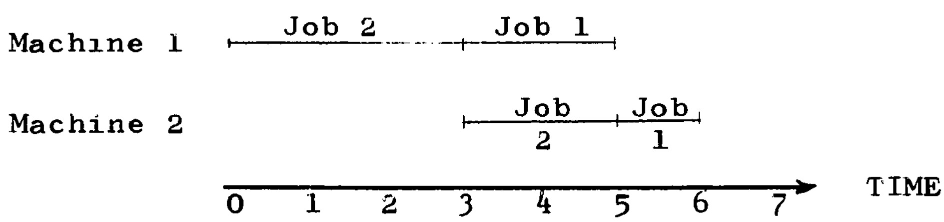

Illustration 2.--Alternate Schedule

Job 2 is completed on Day 5 which is its due-date, and Job 1 is completed on Day 6 which is its due-date. Since both jobs are completed exactly on time, there is no penalty or bonus cost. Therefore, the total actual cost of this schedule is zero.

$$TC = 0$$

However, the schedule generated by the expected savings procedure had a total actual cost of -10. $-10 < 0$. Therefore, the schedule obtained by the expected savings procedure is optimal for this problem. It is interesting to note that the optimal schedule had one job tardy while the non-optimal schedule completed both jobs exactly on time. This is because bonus and penalty costs were considered and the objective is the minimization of these costs, not the maximization of the number of jobs completed on time.

Expected Cost Formulation 1(EC)

This is an expected cost dispatching procedure which is designed to sequence competing jobs at each facility in the shop so as to minimize the total cost of processing all jobs to completion. This procedure computes the expected cost of processing to completion each job competing for an available facility under the assumption that the job will be processed to completion without any delays. Jobs are then sequenced for processing in the order of non-increasing expected costs. Each time a shop facility becomes available for assignment, the expected cost procedure is applied to the jobs available for processing on that facility. Jobs are assigned to the available facility in the order of non-increasing expected costs.

This procedure was chosen because it appears to have characteristics not unlike the earliest due-date and least slack sequencing rules except that this procedure is probabilistic and the value of the jobs is weighted by penalty and bonus costs for tardiness and earliness. The expected costs are computed in the same manner as in the expected savings dispatching procedure.

As an example of the application of this procedure to a small problem consider a machine scheduling problem with 3 jobs and 1 machine. The problem is to determine the best sequence to process the jobs in order to minimize the total cost of processing all jobs to completion. Assume that the day the jobs become available for processing is Day 1.

TABLE 11

JOB CHARACTERISTICS

Job	Machine Sequence	B_i	P_i	d_i
1	1	0	2	5
2	1	0	5	3
3	1	0	3	4

TABLE 12

PROCESSING-TIME DISTRIBUTIONS

Job	$\bar{\bar{X}}_{i1}$	S_{i1}
1	3	1
2	3	1
3	3	1

At the beginning of this manual simulation all three jobs are available for processing on Machine 1, and Machine 1 is available. To determine the sequence that these jobs should be processed, compute the expected cost of processing each job:

$$EC_i = P_i \sum_{D=1}^{\infty} \alpha_D D - B_i \sum_{D=1}^{\infty} \theta_D D$$

since there is only one machine, $\bar{\bar{X}}_i = \bar{X}_{i1}$ and $S_i^P = S_{i1}$.

Since B_i is zero for each job, there is no need to compute the expected number of days early,

$$\sum_{D=1}^{\infty} \theta_D D .$$

TABLE 13

EXPECTED NUMBER OF DAYS TARDY FOR JOB 1 IF PROCESSED IMMEDIATELY

D	$K_D = \frac{X-\overline{\overline{X}}}{S^P}$	$\gamma_D = \int_{-\infty}^{K_D} f(x)dx$	$\alpha_D = \gamma_D - \gamma_{D-1}$	$\alpha_D D$
0	$\frac{5-3}{1} = 2$	.977	---	---
1	$\frac{6-3}{1} = 3$	.999	.022	.022
2	$\frac{7-3}{1} = 4$	1.000	.001	.002

$$\sum_{D=1}^{\infty} \alpha_D D = .024$$

$$EC_1 = P_1 \sum_{D=1}^{\infty} \alpha_D D = (2)(.024) = .048$$

TABLE 14

EXPECTED NUMBER OF DAYS TARDY FOR JOB 2 IF PROCESSED IMMEDIATELY

D	K_D	γ_D	α_D	$\alpha_D D$
0	0	.500	---	---
1	1	.841	.341	.341
2	2	.977	.136	.272
3	3	.999	.022	.066
4	4	1.000	.001	.004

$$\sum_{D=1}^{\infty} \alpha_D D = .683$$

$$EC_2 = P_2 \sum_{D=1}^{\infty} \alpha_D D = (5)(.683) = 3.415$$

TABLE 15

EXPECTED NUMBER OF DAYS TARDY FOR JOB 3 IF PROCESSED IMMEDIATELY

D	K_D	γ_D	α_D	$\alpha_D D$
0	1	.841	---	---
1	2	.977	.136	.136
2	3	.999	.022	.044
3	4	1.000	.001	.003

$$\sum_{D=1}^{\infty} \alpha_D D = .183$$

$$EC_3 = P_3 \sum_{D=1}^{\infty} \alpha_D D = (3)(.183) = .549$$

$$3.415 > .549 > .024$$

$$EC_2 > EC_3 > EC_1$$

Because Job 2 has the highest expected cost, it is processed first. If Job 2 is processed first, it would be completed on Day 3. At this time a decision would have to be made as to which of the two remaining jobs to process next. Another expected cost computation would be made for Jobs 3 and 1. If any other jobs had joined the queue, an expected cost computation would be made for them also. In order to compute the expected costs the time until the due-dates must be reduced by three days, the expected processing-time of Job 2.

TABLE 16

EXPECTED NUMBER OF DAYS TARDY FOR JOB 1 IF DELAYED FOR THREE DAYS

D	K_D	γ_D	α_D	$\alpha_D D$
0	$\frac{2-3}{1} = -1$	.159	---	---
1	$\frac{3-3}{1} = 0$	.500	.341	.341
2	$\frac{4-3}{1} = 1$	.841	.341	.682
3	$\frac{5-3}{1} = 2$	.977	.136	.408
4	$\frac{6-3}{1} = 3$	.999	.022	.088
5	$\frac{7-3}{1} = 4$	1.000	.001	.005

$$\sum_{D=1}^{\infty} \alpha_D D = 1.524$$

$$EC_1 = (2)(1.524) = 3.048$$

TABLE 17

EXPECTED NUMBER OF DAYS TARDY FOR JOB 3 IF DELAYED FOR THREE DAYS

D	K_D	γ_D	α_D	$\alpha_D D$
0	$\frac{1-3}{1} = -2$	.023	---	---
1	$\frac{2-3}{1} = -1$	.159	.136	.136
2	$\frac{3-3}{1} = 0$	.500	.341	.682
3	$\frac{4-3}{1} = 1$	.841	.341	1.023
4	$\frac{5-3}{1} = 2$	.977	.136	.544
5	$\frac{6-3}{1} = 3$	.999	.022	.110
6	$\frac{7-3}{1} = 4$	1.000	.001	.006

$$\sum_{D=1}^{\infty} \alpha_D D = 2.501$$

$$EC_3 = (3)(2.501) = 7.503$$

$$7.503 > 3.048$$

$$EC_3 > EC_1$$

Because Job 3 has the higher expected cost, it is processed next. It would be completed on Day 6. Job 1 would then be processed beginning on Day 7. It would be completed on Day 9. No expected cost computation is necessary to assign Job 1 to Machine 1 on Day 7, because it has no competition for the facility. Therefore, the sequence for processing the jobs given by the expected cost dispatching procedure is 2-3-1.

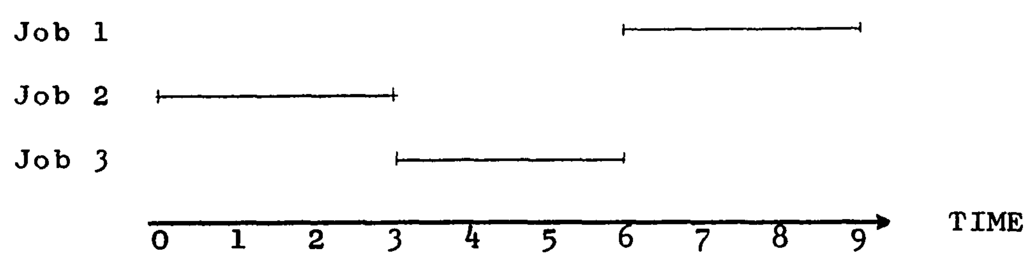

Illustration 3.--Gantt Chart for 2-3-1 Schedule

For this schedule, Job 2 is completed on Day 3 with no cost since it was due on Day 3. Job 3 is completed on Day 6 at a cost of (3)(2)=6 since it was due on Day 4. Job 1 is completed on Day 9 at a cost of (2)(4)=8 since it was due on Day 5. Therefore, the total cost of the 2-3-1 schedule is 0+6+8=14.

There are five other possible sequences for processing these three jobs. To determine if the 2-3-1 sequence is optimal, these schedules may be enumerated and a total cost computed for each of them. Then that schedule with the smallest total cost is the optimal schedule.

TABLE 18

FEASIBLE SEQUENCES AND THEIR COSTS

Sequence	Cost
1-2-3	30
1-3-2	36
2-1-3	17
2-3-1	14
3-1-2	32
3-2-1	23

For this problem the optimal processing sequence is 2-3-1 which has an actual cost of 14. This is the same sequence that was generated by the expected cost procedure.

Expected Cost Formulation 2 (TEC)

This expected cost dispatching procedure is based on the same expected cost formulation as the previously discussed expected cost procedure. However, this procedure considers both the expected cost of the job which is to be immediately processed and the expected costs of those jobs which must be delayed for at least the processing-time of the job to be processed immediately. A procedure which takes into account all of these expected costs will be computationally less efficient than the two procedures already discussed. However, the magnitude of the computations is still very reasonable if a digital computer is used, and significantly improved schedules are produced.

To define the manner in which the expected cost computations are made for this procedure, let $J = \{$set of all jobs available for processing on the facility being considered$\}$. Also, let TEC_i equal the expected cost of processing Job i immediately, plus the sum of the expected costs of processing all other jobs competing for this facility if delayed for the expected processing-time of Job i.

As before the expected cost formulation is:

$$EC_i = P_i \sum_{D=1}^{\infty} \alpha_D D - B_i \sum_{D=1}^{\infty} \theta_D D$$

An equation can now be written for TEC_i .

$$TEC_i = EC_i + \sum_{\substack{r \in J \\ r \neq i}} EC_r'$$

where EC_i is the expected cost of processing Job i immediately on the facility under consideration. EC_r' is the expected cost of processing Job r if delayed for the expected processing-time of Job i on the facility for which this calculation is being made. EC_r' is summed for all jobs competing for processing except Job i which is being considered for immediate processing.

In order to apply this dispatching procedure to a job-shop problem, TEC_i is computed for all $i \in J$. Then the jobs are assigned to the available facility in the order of nondecreasing TEC_i. That is, the job with the smallest TEC_i is the first to be processed. This procedure is applied to all the facilities of the shop. Each time a facility completes processing a job and becomes available for the processing of another job, this total expected cost dispatching procedure is used to determine which job should be processed. This continues until all jobs have been processed to completion.

It should be noted that for all the dispatching rules if there is only one job competing for a facility when that facility becomes available, then that job must be assigned to the facility and no total expected cost calculation is necessary. Also, if there are no jobs competing for a facility when it becomes available, the facility remains idle until a job arrives for processing.

To illustrate the application of this dispatching rule the same example problem will be solved as was used to illustrate the first expected cost dispatching rule. This example simplifies the hand calculations since the bonus per day is zero for all jobs. This eliminates the need for computing the expected number of days early. Also, all jobs have the same processing-time distributions. However, the penalty costs and due-dates are random. There is only one machine in this shop, and all three jobs are processed by this machine. The machine can process only one job at a time.

To determine the sequence in which the jobs should be processed, TEC_i will be computed for i = 1,2,3, and the job with the smallest TEC_i will be processed first. Then, when this job is completed TEC_i will be computed for the remaining two jobs, and the job with the smaller TEC_i will be processed next. At the completion of this job the remaining job will be processed, and the schedule will be complete. If the problem contained a larger number of jobs and machines, the algorithm would be applied in the same manner at each machine as it becomes available for processing. The jobs would move from facility to facility according to their required technological processing sequence as a job completed processing on one facility and become available for processing on another.

TABLE 19

PROBLEM CHARACTERISTICS

Job	Machine Sequence	B_i	P_i	d_i	Processing Time Distributions $\bar{X}_{i1}$	S_{i1}
1	1	0	2	5	3	1
2	1	0	5	3	3	1
3	1	0	3	4	3	1

Determination of TEC_1 :

$$TEC_1 = EC_1 + EC_2' + EC_3'$$

In Expected Cost Formulation 1, EC_1 , the expected cost of Job 1 if processed immediately, was found to be .048.

$$EC_1 = .048$$

EC_2' and EC_3' , the expected costs of processing Jobs 2 and 3, if delayed for the expected processing-time of Job 1, must be computed.

$$EC_2' = P_2 \sum_{D=1}^{\infty} \alpha_D D - B_2 \sum_{D=1}^{\infty} \theta_D D$$

where $B_2 = 0$ and the probabilities are computed assuming that Job 2 has been delayed for 3 days.

TABLE 20

EXPECTED NUMBER OF DAYS TARDY FOR
JOB 2 IF DELAYED FOR THREE DAYS

D	K_D	γ_D	α_D	$\alpha_D D$
0	$\frac{0-3}{1} = -3$	.001	---	---
1	$\frac{1-3}{1} = -2$	.023	.022	.022
2	$\frac{2-3}{1} = -1$	.159	.136	.272
3	$\frac{3-3}{1} = 0$	.500	.341	1.023
4	$\frac{4-3}{1} = 1$	.841	.341	1.364
5	$\frac{5-3}{1} = 2$	.977	.136	.680
6	$\frac{6-3}{1} = 3$	.999	.022	.132
7	$\frac{7-3}{1} = 4$	1.000	.001	.007

$$\sum_{D=1}^{\infty} \alpha_D D = 3.500$$

$$EC_2' = P_2 \sum_{D=1}^{\infty} \alpha_D D = (5)(3.5) = 17.5$$

TABLE 21

EXPECTED NUMBER OF DAYS TARDY FOR JOB 3 IF DELAYED FOR THREE DAYS

D	K_D	γ_D	α_D	$\alpha_D D$
0	-2	.023	---	---
1	-1	.159	.136	.136
2	0	.500	.341	.682
3	1	.841	.341	1.023
4	2	.977	.136	.544
5	3	.999	.022	.110
6	4	1.000	.001	.006

$$\sum_{D=1}^{\infty} \alpha_D D = 2.501$$

$$EC_3' = P_3 \sum_{D=1}^{\infty} \alpha_D D = (3)(2.501) = 7.503$$

$$TEC_1 = .048 + 17.5 + 7.503 = 25.051$$

Determination of TEC_2 :

$$TEC_2 = EC_2 + EC_1' + EC_3'$$

In Expected Cost Formulation 1, EC_2 , the expected cost of Job 2 if processed immediately was found to be 3.415.

$$EC_2 = 3.415$$

EC_1' and EC_3' , the expected cost of processing Jobs 1 and 3 if delayed for the expected processing-time of Job 2, must be computed.

TABLE 22

EXPECTED NUMBER OF DAYS TARDY FOR JOB 1 IF DELAYED FOR THREE DAYS

D	K_D	γ_D	α_D	$\alpha_D D$
0	-1	.159	---	---
1	0	.500	.341	.341
2	1	.841	.341	.682
3	2	.977	.136	.408
4	3	.999	.022	.088
5	4	1.000	.001	.005

$$\sum_{D=1}^{\infty} \alpha_D D = 1.524$$

$$EC_1' = P_1 \sum_{D=1}^{\infty} \alpha_D D = (2)(1.524) = 3.048$$

Since the expected processing-times of Job 1 and Job 2 are equal, the expected cost of Job 3 if delayed for the expected processing-time of Job 2 will be equal to the expected cost of Job 3 if delayed for the expected processing-time of Job 1.

$$EC_3' = 7.503$$

$$TEC_2 = 3.415 + 3.048 + 7.503$$

$$= 13.966$$

Determination of TEC_3 :

$$TEC_3 = EC_3 + EC_1' + EC_2'$$

In Expected Cost Formulation 1, EC_3 , the expected cost of Job 3 if processed immediately was found to be .549.

$$EC_3 = .549$$

Since the expected processing-times of Jobs 1, 2, and 3 are all equal to 3, EC_1' is equal to the value of EC_1' found in the computation of TEC_2 , and EC_2' is equal to the value of EC_2' found in the computation of TEC_1 .

$$EC_1' = 3.048$$

$$EC_2' = 17.5$$

$$TEC_3 = .549 + 3.048 + 17.5$$

$$= 21.097$$

$$13.966 < 21.097 < 25.051$$

$$TEC_2 < TEC_3 < TEC_1$$

Because Job 2 has the lowest total expected cost, it would be processed first. If Job 2 is processed first, it would be completed on Day 3. At this time a decision has to be made as to which of the two remaining jobs to process next. Therefore, another total expected cost computation must be made for Jobs 1 and 3. The job with the smaller TEC would be processed beginning on Day 4.

Determination of TEC_1 :

$$TEC_1 = EC_1 + EC_3'$$

In this case, EC_1 is the same as the EC_1 computed in the previous TEC_2 computation.

$$EC_1 = 3.048$$

EC_3' is the expected cost of Job 3 if delayed for the expected processing-time of Job 1 which began processing on Day 4.

TABLE 23

EXPECTED NUMBER OF DAYS TARDY FOR
JOB 3 IF DELAYED FOR SIX DAYS

D	K_D	γ_D	α_D	$\alpha_D D$
0	-5	.000	---	---
1	-4	.000	.000	.000
2	-3	.001	.001	.002
3	-2	.023	.022	.066
4	-1	.159	.136	.544
5	0	.500	.341	1.705
6	1	.841	.341	2.046
7	2	.977	.136	.952
8	3	.999	.022	.176
9	4	1.000	.001	.009

$$\sum_{D=1}^{\infty} \alpha_D D = 5.500$$

$$EC_3' = P_3 \sum_{D=1}^{\infty} \alpha_D D = (3)(5.500) = 16.5$$

$$TEC_1 = 3.048 + 16.5 = 19.548$$

Determination of TEC_3 :

$$TEC_3 = EC_3 + EC_1'$$

In this case, EC_3 is the same as the EC_3 computed in the first TEC_1 computation.

$$EC_3 = 7.503$$

EC_1' is the expected cost of Job 1 if delayed for the expected processing-time of Job 3 which began processing on Day 4.

TABLE 24

EXPECTED NUMBER OF DAYS TARDY FOR JOB 1 IF DELAYED FOR SIX DAYS

D	K_D	γ_D	α_D	$\alpha_D D$
0	-4	.000	---	---
1	-3	.001	.001	.001
2	-2	.023	.002	.004
3	-1	.159	.136	.408
4	0	.500	.341	1.364
5	1	.841	.341	1.705
6	2	.977	.136	.816
7	3	.999	.022	.154
8	4	1.000	.001	.008

$$\sum_{D=1}^{\infty} \alpha_D D = 4.460$$

$$EC_1' = P_1 \sum_{D=1}^{\infty} \alpha_D D = (2)(4.460) = 8.920$$

$$TEC_3 = 7.503 + 8.921$$

$$= 16.424$$

$$16.424 < 19.548$$

$$TEC_3 < TEC_1$$

Since Job 3 has the smaller total expected cost, it would be processed before Job 1. The processing of Job 1 would begin as soon as the processing of Job 3 is completed. The total expected cost schedule then is to process the jobs in the sequence 2-3-1. It was shown in the illustration of the first expected cost procedure that the 2-3-1 sequence is optimal for this problem with a minimum total cost of 14. Therefore, the total expected cost procedure also generated an optimal schedule for this problem.

Simulation Programs

Simulation programs[1] have been developed for the expected cost stochastic dispatching procedures and the deterministic least slack/remaining operation dispatching procedure. All of the simulation programs accept the same input data although not all of the input data is used by each of the dispatching procedures. The simulations developed all have the same assumptions, capabilities, and

[1]The programs, flow charts, sample data, and sample output are available in the University of Oklahoma School of Industrial Engineering Research Report TS-71-1.

limitations except for the dispatching procedure employed to determine the sequence in which the queue of jobs competing for each facility is to be processed.

These simulations were developed to provide a means of comparing the expected cost dispatching procedures presented in this paper to the least slack/remaining operation procedure which is well known. The least slack/remaining operation procedure has been simulated by several researchers (10), (11), (36), and (15) and has been found to be one of the better of the due-date rules. Since the least slack/remaining operation rule has been compared to a great many other dispatching procedures, a comparison of the performance of the expected cost procedures to the performance of the least slack/remaining operation rule provides a means of verifying the validity of the proposed expected cost procedures.

The least slack/remaining operation rule is used to determine the sequence in which competing jobs should be processed. The rule requires known deterministic processing-times and a due-date for all jobs. To use the rule, the following computation is made for all competing jobs:

$$\frac{\text{Slack}}{\text{Operation}} = \frac{\text{Due-Date} - \left(\begin{array}{c}\text{Processing-times of the}\\ \text{Remaining Operations}\end{array}\right)}{\text{Number of Remaining Operations}}$$

Then the jobs are sequenced for processing in the order of nondecreasing slack/remaining operation.

To develop sequencing rules and simulations, certain assumptions must be made about the jobs to be processed and the facilities of the shop. For each job the following assumptions are made:

1. Jobs are strictly ordered sequences of operations.
2. Assembly is allowed.
3. Partition is not allowed.
4. Each operation can be performed by only one facility in the shop.
5. Once an operation is started on a facility, it must be processed through to completion without interruption.
6. A job may be in process on at most one facility at a time.
7. Each job has a due-date associated with it.
8. Each job has associated with it a penalty cost directly proportional to the tardiness of the job.
9. Each job has associated with it a bonus which is directly proportional to the earliness of the job.
10. For each job, an estimate of the expected processing-time of the job on all applicable facilities is assumed to be known.

11. For each job, the processing cost per unit of time for the job on all applicable facilities is assumed to be known.
12. For each job, the proportion of the capacity of the facility required for the processing of that job is assumed known.
13. A job may already be in process at the beginning of the simulation.
14. A job may be delayed for processing until some predetermined time.

For each facility the following assumptions are made:

1. A facility may, depending upon its capacity limitations, process jobs simultaneously.
2. Each facility is continuously available for assignment.
3. For each facility, the processing-time distribution is assumed to be normal with known mean and standard deviation.

If the above assumptions for jobs and facilities are compared with those stated on page 5 of this paper for the simple job-shop, it is readily seen that the simulations developed are much more general than those reported in the literature. This is desirable because then the simulation programs can be effectively utilized as a management tool as well as a means of evaluating and comparing the performance of various sequencing rules.

Until recently there was really no need for a simulation which would allow the manager of a job-shop to accurately simulate his particular shop because the information required to construct an accurate simulation was not available and was too expensive to collect. However, today many companies either have or are in the process of developing comprehensive management information systems. This is due in no small part to the availability of large scale random access storage devices at a reasonable cost and to the recent development of terminal devices for data collection which are easy to operate, flexible, and inexpensive.

If a shop is operating under a given dispatching procedure, then the shop may be simulated to predict the kind of performance that can be expected under existing conditions. If the predicted performance is not satisfactory, the manager may then simulate various possible solution alternatives such as additional facilities, additional shifts, overtime, or subcontracting. In this way, the manager can compare the cost and performance of these alternatives. Thus comprehensive simulations such as these can be a valuable management tool.

Experimental Results

To verify the validity of the proposed expected cost dispatching procedure, thirty-eight machine scheduling problems were simulated using the expected cost dispatching procedures and the least slack/remaining operation procedure to sequence the jobs. Ten flow-shop problems and twenty-eight job-shop problems were simulated. The number of jobs ranged from 3 to 20, and the number of facilities ranged from 1 to 6. All jobs had a required technological sequence in which they had to be processed by the facilities, a due-date, and a linear penalty cost function for tardiness. All facilities had a processing-time distribution that was assumed to be approximately normal with known mean and variance.

Three job-shop simulations were used to determine the relative cost performance of the two expected cost dispatching procedures and the least slack/remaining operation dispatching procedure. One job-shop simulation which will be referred to as the EC simulation simulates the behavior of a job-shop which uses the expected cost formulation 1 dispatching procedure to sequence competing jobs. This procedure processes that job first which has the highest expected cost if processed to completion without delay. The definition of expected cost for this procedure is:

$$EC_i = P_i \sum_{D=1}^{\infty} \alpha_D D - B_i \sum_{D=1}^{\infty} \theta_D D$$

A second job-shop simulation which will be referred to as the TEC simulation simulates the performance of a job-shop which uses the expected cost formulation 2 dispatching procedure to sequence competing jobs. This procedure processes that job first which has the least total expected cost where total expected cost is defined as:

$$TEC_i = EC_i + \sum_{\substack{r \in J \\ r \neq i}} EC_r,$$

The third job-shop simulation which will be referred to as the S/O simulation simulates the activities of a job-shop in which the least slack/remaining operation dispatching procedure is used to determine the processing sequence of the jobs. This procedure processes that job first which has the least slack/remaining operation which is defined as:

$$\frac{\text{Slack}}{\text{Operation}} = \frac{\text{Due-Date} - \left(\begin{array}{c}\text{Processing-Times of Remaining} \\ \text{Operations}\end{array}\right)}{\text{Number of Remaining Operations}}$$

Of the ten flow-shop problems simulated, three had 3 jobs and 1 machine; six had 3 jobs and 2 machines; and one had 6 jobs and 2 machines. For the problems with 3 jobs and 1 machine, the processing-time distribution for that machine had a mean of 2.0 and a standard deviation of 0.5. For the problems with 3 jobs and 2 machines, Machine 1 had a processing-time distribution with a mean of 1.0 and a standard deviation of 0.3, and Machine 2 had a distribution with a mean of 2.0 and a standard deviation of 0.6. For the problem with 6 jobs and 2 machines, both Machine 1 and Machine 2 had a processing time distribution with a mean of 2.0 and a standard deviation of 0.6.

For these flow-shop problems the schedules generated by the EC , TEC , and S/O simulations were compared to each other and to the optimum schedule for each problem. The optimum schedule which minimizes the total penalty cost of all jobs was found by the total enumeration of all feasible schedules and the computation of total penalty costs for each schedule with the assumption that each job requires an amount of time for processing on a machine which is equal to the mean of that machine's processing-time distribution. It was found that the TEC simulation produced an optimal schedule for all of the flow-shop problems. The EC simulation produced schedules for which the mean total penalty costs were 26% greater than the mean total penalty costs for the TEC schedules, and the S/O simulation produced schedules for which the mean total penalty costs were 61% greater than the mean total penalty costs of the TEC schedules and 28% greater than the mean total penalty costs of the EC schedules. A summary of the normalized costs of the schedules generated by the three dispatching procedures is given in Table 25.

TABLE 25

FLOW-SHOP PROBLEMS

Problem	Jobs	Facilities	Normalized Costs		
			TEC	EC	S/O
3105	3	1	1.00	1.25	1.25
2	3	1	1.00	1.00	1.00
3	3	1	1.00	1.80	2.20
3201	3	2	1.00	1.00	1.00
3202	3	2	1.00	1.00	2.00
3203	3	2	1.00	1.05	1.09
3204	3	2	1.00	1.22	1.45
3205	3	2	1.00	2.00	2.00
3206	3	2	1.00	1.00	1.00
3212	6	2	1.00		3.06
	Mean Costs		1.00	1.26	1.61

A total of twenty-eight job-shop problems were simulated. These were true job-shop problems in that the required technological processing sequences of the jobs were random. Twenty-seven of these problems had 6 jobs and 3 machines, and one problem had 20 jobs and 6 machines. The first nineteen problems in Table 29 had the processing-time distributions in Table 26.

TABLE 26

PROCESSING-TIME DISTRIBUTIONS

Machine	$\bar{X}$	S
1	1.0	0.3
2	2.0	0.6
3	3.0	0.9

The next six problems in Table 29 had the processing-time distributions shown in Table 27.

TABLE 27

PROCESSING-TIME DISTRIBUTIONS

Machine	$\bar{X}$	S
1	2.0	0.6
2	2.0	0.6
3	2.0	0.6

The last problem in Table 29, problem number 5000, had the processing-time distributions shown in Table 28.

TABLE 28

PROCESSING-TIME DISTRIBUTIONS

Machine	$\overline{X}$	S
1	1.35	0.59
2	1.89	0.99
3	2.47	1.37
4	1.40	0.66
5	2.27	1.18
6	3.10	1.23

Three of these 6 x 3 job-shop problems had equal due-dates and equal penalties; three problems had equal due-dates and random penalties; three problems had random due-dates and equal penalties; and the remainder of the problems had random due-dates and random penalties. The expected cost dispatching procedures produced schedules with substantially lower costs than the least slack/remaining operation schedules except in the case of equal due-dates and equal penalties. Without any difference in due-dates or penalties, the performance of the expected cost dispatching procedures was equal to the performance of the least slack/remaining operation procedure. These problems are numbers 5, 9, and 109 in Table 29.

Optimal schedules for the 6-job and 3-machine job-shop problems investigated here are not known nor is there a practical means available for finding the optimal solutions. Therefore, the evaluation of the performance of the TEC and the EC simulations is made by comparing the costs of the TEC and the EC schedules to the costs of the S/O schedules. This does not demonstrate optimality of these techniques, but it does demonstrate that substantial improvements over the S/O schedules are obtained by using one of the expected cost procedures. An examination of the summary of the normalized schedule costs for the 6-job, 3-machine problems in Table 29 reveals that the mean total cost of the S/O schedules are 51% greater than the mean total costs for the TEC schedules and 15% greater than the mean total costs for the EC schedules. A detailed analysis of the 20-job, 6-machine problem is found in Table 30. For this problem the S/O schedule has a 53% higher penalty cost, three times as many jobs tardy, and a 13% greater total flow time than the TEC schedule. The total number of days tardy for all jobs is the same for both the S/O and the TEC schedules.

TABLE 29

JOB-SHOP PROBLEMS

Problem	Jobs	Facilities	Normalized Costs		
			TEC	EC	S/O
5	6	3	1.00	1.00	1.00
9	6	3	1.00	1.00	1.00
109	6	3	1.00	1.00	1.00
10	6	3	1.00	2.19	3.50
6	6	3	1.00	1.81	3.12
210	6	3	1.00	1.33	1.61
7	6	3	1.00	1.00	1.00
11	6	3	1.00	1.00	1.00
211	6	3	1.00	1.07	1.07
12	6	3	1.00	1.27	1.00
112	6	3	1.00	1.35	1.27
213	6	3	1.00	1.00	1.74
312	6	3	1.00	1.35	1.73
412	6	3	1.00	1.33	1.57
512	6	3	1.00	1.08	1.01
612	6	3	1.00	1.00	1.57
712	6	3	1.00	1.07	1.25
812	6	3	1.00	1.32	1.32
912	6	3	1.00	1.18	1.86
2212	6	3	1.00	1.46	2.46

TABLE 29--Continued

Problem	Jobs	Facilities	Normalized Costs TEC	EC	S/O
2312	6	3	1.00	1.65	1.65
2412	6	3	1.00	1.25	1.25
2612	6	3	1.00	1.53	1.76
2712	6	3	1.00	.98	1.06
2812	6	3	1.00	1.02	1.02
5000	20	6	1.00		1.53
	Mean Costs		1.00	1.31	1.51

TABLE 30

ANALYSIS OF 20-JOB, 6-FACILITY PROBLEM

			S/O		TEC	
Job	Due-Date	Penalty Cost	Days Tardy	Penalty Cost	Days Tardy	Penalty Cost
1	20	125	5	625		
2	28	125	1	125		
3	20	125	6	750		
4	12	125				
5	16	50				
6	20	50	3	150		
7	20	50			16	800
8	4	125				
9	4	125				
10	12	125				
11	4	125				
12	8	200				
13	16	200				
14	20	200	4	800		
15	28	125	3	375		
16	12	125			4	500
17	20	125	1	125	9	1125
18	16	125	5	625		
19	8	125				
20	28	125	1	125		
Total Penalty Cost			3700		2425	
Total Days Tardy			29		29	
Total Jobs Tardy			9		3	
Total Flow-Time			330		292	

Two 6 x 3 job-shop problems not included in Table 29 were found where the TEC procedure did not produce schedules with costs equal to or less than the S/O procedure. However, this was for only two out of the thirty-eight problems which were simulated, and it was found that schedules equal to or better than the S/O schedules could be obtained from the TEC procedure simply by increasing the standard deviation of the processing-time distributions. Therefore, the TEC procedure is somewhat sensitive to the standard deviation of the processing-time distributions. This sensitivity seems small and not too important since even when an inferior schedule was produced by the TEC procedure there was less than a 10% difference in the total actual costs of the TEC and the S/O schedule.

From the experimental investigation of the TEC and the EC procedures for sequencing jobs, it is obvious that these are very effective dispatching procedures. In fact the S/O procedure produces job-shop schedules that have 51% higher mean costs than the TEC procedure and 15% higher mean costs than the EC procedure. The S/O procedure requires less computation than either the EC or the TEC procedures, but the improvement in schedules is sufficient to justify the additional computational effort. Also, in an actual application of dispatching procedures such as these, the priorities would probably be computed using a digital computer and provided to the shop. The computer time required for the computation of the expected cost priorities for the problems simulated was insignificant.

Optimality in Special Cases

It can be shown that in some special cases of the job-shop problem the TEC dispatching procedure always produces an optimal schedule. An optimal schedule is defined as a feasible schedule for which the true expected total penalty cost is less than or equal to the true expected total penalty costs of all feasible schedules. To demonstrate the optimality of the TEC procedure, a special case of the job-shop problem with 2 jobs and 1 machine is considered. The proof consists of computing the true expected value of the total penalty costs for all feasible schedules, determining the schedule which has minimum expected cost and sufficient conditions for this minimum, and proving that the TEC procedure will always produce the schedule with minimum true expected total penalty cost under these conditions.

Special Case I

Consider a job-shop problem with 2 jobs and 1 machine where the processing-time distribution of the machine is approximately normally distributed with a mean $\overline{X}$ and a standard deviation S. The jobs are both immediately available for processing, and each job has a due-date and a penalty cost per day tardy. Let both jobs have the same due-date, d , and let the penalty cost per day tardy for Job 1 be greater than the penalty cost per day tardy for Job 2, $P_1 > P_2$. The two jobs may be processed either in the sequence 1-2 or in the sequence 2-1. These are the only feasible schedules for this problem. Therefore, the sequence with the smaller expected value of the total penalty cost for all jobs is optimal for this problem.

Proof that Sequence 1-2 is Optimal.--The job that is processed first, regardless of which job it is, will have a processing-time distribution with mean $\bar{X}$ and standard deviation S . The processing-time distribution of the job that is processed last will have a mean:

$$\bar{X} + \bar{X} = 2\bar{X}$$

and a standard deviation:

$$\sqrt{S^2 + S^2} = \sqrt{2}\, S .$$

The expected value of the total penalty costs for the jobs processed in the sequence 1-2 is:

$$EC_{1-2} = EC_1 + EC_2'$$

where

$$EC_1 = P_1 \sum_{D=1}^{\infty} D\,\alpha_D$$

$$EC_2' = P_2 \sum_{D=1}^{\infty} D\,\alpha_D'$$

and

$$\alpha_D = \int_{d+D-1}^{d+D} f(X;\ \overline{X},\ S^2)\ dx \ ;\ \alpha_D' = \int_{d+D-1}^{d+D} f(X;\ 2\overline{X},\ 2S^2)\ dx$$

with D defined as the number of days tardy for a job. Therefore, the expected value of the total penalty costs for the processing sequence 1-2 is:

$$EC_{1-2} = P_1 \sum_{D=1}^{\infty} D\,\alpha_D + P_2 \sum_{D=1}^{\infty} D\,\alpha_D'$$

$$EC_{1-2} = P_1 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ \overline{X},\ S^2)\ dx$$

$$+ P_2 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ 2\overline{X},\ 2S^2)\ dx$$

In the same manner the expected value of the total penalty costs of the sequence 2-1 is:

$$EC_{2-1} = EC_2 + EC_1'$$

$$EC_{2-1} = P_2 \sum_{D=1}^{\infty} D\,\alpha_D + P_1 \sum_{D=1}^{\infty} D\,\alpha_D'$$

$$EC_{2-1} = P_2 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ \overline{X},\ S^2)\ dx$$

$$+ P_1 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ 2\overline{X},\ 2S^2)\ dx$$

If $EC_{1-2} \leq EC_{2-1}$, then the sequence 1-2 is optimal for this problem.

$$EC_{1-2} \leq EC_{2-1}$$

$$P_1 \sum_{D=1}^{\infty} D\alpha_D + P_2 \sum_{D=1}^{\infty} D\alpha_D' \leq P_2 \sum_{D=1}^{\infty} D\alpha_D + P_1 \sum_{D=1}^{\infty} D\alpha_D'$$

$$(P_1 - P_2) \sum_{D=1}^{\infty} D\alpha_D \leq (P_1 - P_2) \sum_{D=1}^{\infty} D\alpha_D' \qquad (10)$$

It was assumed that $P_1 > P_2$, therefore $P_1 - P_2 > 0$ and inequality (10) becomes:

$$\sum_{D=1}^{\infty} D\alpha_D \leq \sum_{D=1}^{\infty} D\alpha_D'$$

which when expanded gives:

$$\alpha_1 + 2\alpha_2 + 3\alpha_3 + \ldots \leq \alpha_1' + 2\alpha_2' + 3\alpha_3' + \ldots \qquad (11)$$

This inequality can be rewritten as:

$$\sum_{D=1}^{\infty}\alpha_D + \sum_{D=2}^{\infty}\alpha_D + \sum_{D=3}^{\infty}\alpha_D + \dots \leq \sum_{D=1}^{\infty}\alpha_D{}' + \sum_{D=2}^{\infty}\alpha_D{}' + \sum_{D=3}^{\infty}\alpha_D{}' + \dots \qquad (12)$$

Inequality (12) can be shown to be true by showing that the following inequalities are all true:

$$\sum_{D=1}^{\infty} \alpha_D \leq \sum_{D=1}^{\infty} \alpha_D{}' \qquad (13.1)$$

$$\sum_{D=2}^{\infty} \alpha_D \leq \sum_{D=2}^{\infty} \alpha_D{}' \qquad (13.2)$$

$$\sum_{D=3}^{\infty} \alpha_D \leq \sum_{D=3}^{\infty} \alpha_D{}' \qquad (13.3)$$

$$\vdots \qquad \vdots$$

If (13.1), (13.2), (13.3), ... can all be shown to be true, then this implies that $EC_{1-2} \leq EC_{2-1}$ and schedule 1-2 is optimal.

α_D and $\alpha_D{}'$ are both functions which have parameters d, $\overline{X}$, and S . The validity of (13.1), (13.2), (13.3), ... depends on the relative values of these parameters. Since it will be shown that the TEC procedure always produces the 1-2 sequence for this special case of the job-shop problem, sufficient conditions will be determined for the optimality of the 1-2 processing sequence. This is done by requiring that (13.1), (13.2), (13.3), ... are all true and determining the conditions which are sufficient to prove that (13.1), (13.2), (13.3), ... are true. These same conditions then will be sufficient for schedule 1-2 to be optimal.

Rewrite inequalities (13.1), (13.2), (13.3), ... and require that they are true.

$$\int_{d}^{\infty} f(X;\ \overline{X},\ S^2)\ dx \ \leq\ \int_{d}^{\infty} f(X;\ 2\overline{X},\ 2S^2)\ dx \qquad (14.1)$$

$$\int_{d+1}^{\infty} f(X;\ \overline{X},\ S^2)\ dx \ \leq\ \int_{d+1}^{\infty} f(X;\ 2\overline{X},\ 2S^2)\ dx \qquad (14.2)$$

$$\int_{d+2}^{\infty} f(X;\ \overline{X},\ S^2)\ dx \ \leq\ \int_{d+2}^{\infty} f(X;\ 2\overline{X},\ 2S^2)\ dx \qquad (14.3)$$

$$\vdots \qquad\qquad\qquad \vdots$$

The standard normal deviate of the lower limit of the left integral of (14.1), (14.2), (14.3), ... is:

$$Z_1 = \frac{d+k-\overline{X}}{S} \quad \text{for} \quad k = 0,\ 1,\ 2,\ 3,\ \ldots$$

The standard normal deviate of the lower limit of the right integral is:

$$Z_2 = \frac{d+k-2\bar{X}}{\sqrt{2}\,S} \quad \text{for} \quad k = 0, 1, 2, 3, \ldots$$

If $Z_1 \geq Z_2$ for $k = 0, 1, 2, 3, \ldots$, this implies that (14.1), (14.2), (14.3), ... are true. Require that:

$$Z_1 \geq Z_2$$

$$\frac{d+k-\bar{X}}{S} \geq \frac{d+k-2\bar{X}}{\sqrt{2}\,S} \quad \text{for} \quad k = 0, 1, 2, \ldots$$

multiply both sides by $\sqrt{2}\,S$.

$$\sqrt{2}\,d + \sqrt{2}\,k - \sqrt{2}\,\bar{X} \geq d + k - 2\,\bar{X} \quad \text{for} \quad k = 0, 1, 2, \ldots \tag{15}$$

It is obvious that:

$$\sqrt{2}\,k \geq k \quad \text{for} \quad k = 0, 1, 2, \ldots \tag{15.1}$$

Therefore, (15) will be true if it is required that:

$$\sqrt{2}\,d - \sqrt{2}\,\bar{X} \geq d - 2\,\bar{X} \tag{15.2}$$

$$\sqrt{2}\,d - d \geq \sqrt{2}\,\bar{X} - 2\,\bar{X}$$

$$(\sqrt{2} - 1)\,d \geq (\sqrt{2} - 2)\,\bar{X}$$

$$d \geq \frac{(\sqrt{2} - 2)}{(\sqrt{2} - 1)}\,\bar{X}$$

Multiply by $\frac{\sqrt{2}}{\sqrt{2}}$.

$$d \geq \left(\frac{2 - 2\sqrt{2}}{2 - \sqrt{2}}\right) \overline{X}$$

$$d \geq \left[\frac{2(1 - \sqrt{2})}{\sqrt{2}(\sqrt{2} - 1)}\right] \overline{X}$$

$$d \geq \left[\frac{2(1 - \sqrt{2})}{-\sqrt{2}(1 - \sqrt{2})}\right] \overline{X}$$

$$d \geq -\sqrt{2}\,\overline{X}$$

Therefore, inequality (15.2) will always be true if it is required that:

$$d \geq -\sqrt{2}\,\overline{X}$$

It has already been shown that:

$$\sqrt{2}\,k \geq k \quad \text{for} \quad k = 0, 1, 2, \ldots \tag{15.1}$$

Therefore, if inequalities (15.1) and (15.2) are added together the result is inequality (15):

$$\sqrt{2}\,d + \sqrt{2}\,k - \sqrt{2}\,\overline{X} \geq d + k - 2\,\overline{X} \tag{15}$$

$$\text{for} \quad k = 0, 1, 2, \ldots \text{ and } \quad d \geq -\sqrt{2}\,\overline{X}$$

Therefore:

$$Z_1 \geq Z_2 \quad \text{for} \quad k = 0, 1, 2, \ldots$$

$$\text{and} \quad d \geq -\sqrt{2}\,\overline{X}$$

$$\int_{d+k}^{\infty} f(X;\ \overline{X},\ S^2)\ dx \ \leq \ \int_{d+k}^{\infty} f(X;\ 2\overline{X},\ 2S^2)\ dx$$

$$\text{for} \quad k = 0,\ 1,\ 2,\ \ldots \quad \text{and} \quad d \geq -\sqrt{2}\ \overline{X}$$

Therefore inequalities (14.1), (14.2), (14.3), ... are all true, which implies that:

$$EC_{1-2} \leq EC_{2-1} \quad \text{for} \quad d \geq -\sqrt{2}\ \overline{X}$$

and sequence 1-2 is optimal for this special case since it minimizes the expected value of the total penalty cost.

If $d \ngeq -\sqrt{2}\ \overline{X}$, then it is not necessarily true that $EC_{1-2} \leq EC_{2-1}$. The requirement that $d \geq -\sqrt{2}\ \overline{X}$ is a sufficient condition for $EC_{1-2} \leq EC_{2-1}$. Necessary conditions have not been determined.

<u>Proof that the TEC Procedure is Optimal</u>.--It has been shown that for Special Case I the processing sequence 1-2 is optimal if $d \geq -\sqrt{2}\ \overline{X}$. The TEC procedure can be proven to be optimal by showing that it always produces the 1-2 processing sequence for Special Case I when $d \geq -\sqrt{2}\ \overline{X}$.

The TEC procedure does not use the true expected cost of processing all jobs to completion to determine priorities for the jobs. Instead, it uses an approximation to the true expected cost. This is done because the true expected cost of processing all jobs to completion in a job-shop with two or more facilities is too complicated to be feasible for use in a dispatching procedure. For the TEC dispatching procedure to produce optimal schedules, it is not necessary that the procedure use true expected costs to determine the sequence in which the jobs will be processed. As long as the TEC procedure produces the same processing sequence that would have been obtained had the true expected cost been used, the TEC dispatching procedure is optimal.

The approximate total expected cost of the processing sequence 1-2 as defined by the TEC procedure is:

$$TEC_{1-2} = P_1 \sum_{D=1}^{\infty} \alpha_D D + P_2 \sum_{D=1}^{\infty} \alpha_D' D$$

$$TEC_{1-2} = P_1 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ \overline{X},\ S^2)\ dx$$

$$+ P_2 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ 2\overline{X},\ S^2)\ dx$$

The total expected cost for the 2-1 processing sequence as defined by the TEC procedure is:

$$TEC_{2-1} = P_2 \sum_{D=1}^{\infty} D\alpha_D + P_1 \sum_{D=1}^{\infty} D\alpha_D'$$

$$TEC_{2-1} = P_2 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ \overline{X},\ S^2)\ dx$$

$$+ P_1 \sum_{D=1}^{\infty} D \int_{d+D-1}^{d+D} f(X;\ 2\overline{X},\ S^2)\ dx$$

If $TEC_{1-2} < TEC_{2-1}$ for $d \geq -\sqrt{2}\ \overline{X}$ then the TEC procedure will always produce an optimal schedule for a problem with the assumptions of Special Case I.

$$TEC_{1-2} < TEC_{2-1}$$

$$P_1 \sum_{D=1}^{\infty} D\alpha_D + P_2 \sum_{D=1}^{\infty} D\alpha_D' < P_2 \sum_{D=1}^{\infty} D\alpha_D + P_1 \sum_{D=1}^{\infty} D\alpha_D'$$

$$(P_1 - P_2) \sum_{D=1}^{\infty} D\alpha_D < (P_1 - P_2) \sum_{D=1}^{\infty} D\alpha_D'$$

It was assumed that $P_1 > P_2$, therefore $P_1 - P_2 > 0$.

$$\sum_{D=1}^{\infty} D\alpha_D < \sum_{D=1}^{\infty} D\alpha_D'$$

which when expanded gives:

$$\alpha_1 + 2\alpha_2 + 3\alpha_3 + \ldots < \alpha_1' + 2\alpha_2' + 3\alpha_3' + \ldots$$

This inequality can be rewritten as:

$$\sum_{D=1}^{\infty}\alpha_D + \sum_{D=2}^{\infty}\alpha_D + \sum_{D=3}^{\infty}\alpha_D + \ldots < \sum_{D=1}^{\infty}\alpha_{D'} + \sum_{D=2}^{\infty}\alpha_{D'} + \sum_{D=3}^{\infty}\alpha_{D'} + \ldots \tag{18}$$

Inequality (18) can be shown to be true by showing that the following inequalities are all true:

$$\sum_{D=1}^{\infty}\alpha_D < \sum_{D=1}^{\infty}\alpha_{D'} \tag{19.1}$$

$$\sum_{D=2}^{\infty}\alpha_D < \sum_{D=2}^{\infty}\alpha_{D'} \tag{19.2}$$

$$\sum_{D=3}^{\infty}\alpha_D < \sum_{D=3}^{\infty}\alpha_{D'} \tag{19.3}$$

$$\begin{array}{cc} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{array}$$

If (19.1), (19.2), (19.3), ... can all be shown to be true, then this implies that $TEC_{1-2} < TEC_{2-1}$ and proves the TEC procedure to be optimal for this special case of the job-shop problem.

Inequalities (19.1), (19.2), (19.3), ... can be rewritten as:

$$\int_{d}^{\infty} f(X;\ \overline{X},\ S^2)\ dx < \int_{d}^{\infty} f(X;\ 2\overline{X},\ S^2)\ dx \qquad (20.1)$$

$$\int_{d+1}^{\infty} f(X;\ \overline{X},\ x^2)\ dx < \int_{d+1}^{\infty} f(X;\ 2\overline{X},\ S^2)\ dx \qquad (20.2)$$

$$\int_{d+2}^{\infty} f(X;\ \overline{X},\ S^2)\ dx < \int_{d+2}^{\infty} f(X;\ 2\overline{X},\ S^2)\ dx \qquad (20.3)$$

$$\vdots \qquad\qquad \vdots$$

Show that (20.1), (20.2), (20.3), ... are true. The standard normal deviate of the lower limit of the left integral is:

$$Z_1 = \frac{d+k-\overline{X}}{S} \quad \text{for} \quad k = 0,\ 1,\ 2,\ \ldots$$

The standard normal deviate of the lower limit of the right integral is:

$$Z_2 = \frac{d+k-2\overline{X}}{S} \quad \text{for} \quad k = 0,\ 1,\ 2,\ \ldots$$

If $Z_1 > Z_2$ for $k = 0, 1, 2, \ldots$ this implies that (20.1), (20.2), (20.3), ... are true.

$$Z_1 > Z_2 \quad \text{for} \quad k = 0, 1, 2, \ldots$$

$$\frac{d + k - \overline{X}}{S} > \frac{d + k - 2\overline{X}}{S} \quad \text{for} \quad k = 0, 1, 2, \ldots$$

Multiply both sides by S .

$$d + k - \overline{X} > d + k - 2\overline{X} \quad \text{for} \quad k = 0, 1, 2, \ldots$$

Add $\overline{X} - d - k$ to both sides.

$$0 > -\overline{X} \quad \text{for} \quad k = 0, 1, 2, \ldots$$

Multiply both sides by -1 .

$$\overline{X} > 0 \quad \text{for} \quad k = 0, 1, 2, \ldots$$

$\overline{X}$ is the mean of the processing time distribution for the machine in this special case. Jobs cannot be processed in a negative amount of time or in zero time. Therefore:

$$\overline{X} > 0 \quad \text{for} \quad k = 0, 1, 2, \ldots$$

and

$$Z_1 > Z_2 \quad \text{for} \quad k = 0, 1, 2, \ldots$$

which implies that:

$$\int_{d+k}^{\infty} f(X;\overline{X},S^2)dx < \int_{d+k}^{\infty} f(X;2\overline{X},S^2)\, dx \quad \text{for} \quad k = 0, 1, 2, \ldots$$

and inequalities (20.1), (20.2), (20.3), ... are true.

Therefore, this implies that:

$$TEC_{1-2} < TEC_{2-1}$$

and the TEC procedure always produces an optimal sequence for Special Case I for $d \geq -\sqrt{2}\,\bar{X}$.

Special Case II

The conditions of Special Case II are the same as Special Case I except for due-dates and penalties. Consider a job-shop problem with 2 jobs and 1 machine where the processing-time distribution of the machine is approximately normally distributed with a mean $\bar{X}$ and a standard deviation S . The jobs are both immediately available for processing, and each job has a due-date and a penalty cost per day tardy. Let the due-date of Job 1 be earlier than the due-date of Job 2, $d_1 < d_2$, and let the two jobs have equal penalty costs, P . The two jobs may be processed either in the sequence 1-2 or the sequence 2-1. The sequence with the smaller expected value of total penalty cost for all jobs is optimal for this problem.

Proof that Sequence 1-2 is Optimal.--The job that is processed first will have a processing-time distribution with mean $\bar{X}$ and standard deviation S . The processing-time distribution of the job that is processed last will have a mean $2\bar{X}$ and a standard deviation $\sqrt{2}\,S$.

Let

$$f(x) = f(X;\ \overline{X},\ S^2)$$

$$f'(x) = f(X;\ 2\overline{X},\ 2S^2)$$

The expected value of the total penalty cost for all jobs processed in the sequence 1-2 is:

$$EC_{1-2} = EC_1 + EC_2'$$

$$EC_{1-2} = P\sum_{D=1}^{\infty} D\alpha_D + P\sum_{D=1}^{\infty} D\alpha_D'$$

$$EC_{1-2} = P\sum_{D=1}^{\infty} D\int_{d_1+D-1}^{d_1+D} f(x)dx + P\sum_{D=1}^{\infty} D\int_{d_2+D-1}^{d_2+D} f'(x)dx$$

$$EC_{1-2} = P\Bigg[\int_{d_1}^{d_1+1} f(x)dx + 2\int_{d_1+1}^{d_1+2} f(x)dx + 3\int_{d_1+2}^{d_1+3} f(x)dx + \ldots$$

$$+ \int_{d_2}^{d_2+1} f'(x)dx + 2\int_{d_2+1}^{d_2+2} f'(x)dx + 3\int_{d_2+2}^{d_2+3} f'(x)dx + \ldots\Bigg]$$

$$EC_{1-2} = P\Bigg[\int_{d_1}^{\infty} f(x)dx + \int_{d_1+1}^{\infty} f(x)dx + \int_{d_1+2}^{\infty} f(x)dx + \ldots$$

$$+ \int_{d_2}^{\infty} f'(x)dx + \int_{d_2+1}^{\infty} f'(x)dx + \int_{d_2+2}^{\infty} f'(x)dx + \ldots\Bigg]$$

The expected value of the total penalty cost for all jobs processed in the sequence 2-1 is:

$$EC_{2-1} = EC_2 + EC_1'$$

$$EC_{2-1} = P\sum_{D=1}^{\infty} D\alpha_D + P\sum_{D=1}^{\infty} D\alpha_D'$$

$$EC_{2-1} = P\sum_{D=1}^{\infty} D\int_{d_2+D-1}^{d_2+D} f(x)dx + P\sum_{D=1}^{\infty} D\int_{d_1+D-1}^{d_1+D} f'(x)dx$$

$$EC_{2-1} = P\left[\int_{d_2}^{d_2+1} f(x)dx + 2\int_{d_2+1}^{d_2+2} f(x)dx + 3\int_{d_2+2}^{d_2+3} f(x)dx + \cdots\right.$$

$$\left. + \int_{d_1}^{d_1+1} f'(x)dx + 2\int_{d_1+1}^{d_1+2} f'(x)dx + 3\int_{d_1+2}^{d_1+3} f'(x)dx + \cdots\right]$$

$$EC_{2-1} = P\left[\int_{d_2}^{\infty} f(x)\ dx + \int_{d_2+1}^{\infty} f(x)\ dx + \int_{d_2+2}^{\infty} f(x)\ dx + \cdots\right.$$

$$\left. + \int_{d_1}^{\infty} f'(x)dx + \int_{d_1+1}^{\infty} f'(x)dx + \int_{d_1+2}^{\infty} f'(x)dx + \cdots\right]$$

If $EC_{1-2} \leq EC_{2-1}$, then the sequence 1-2 is optimal for this problem.

$$EC_{1-2} \leq EC_{2-1}$$

$$EC_{1-2} - EC_{2-1} \leq 0 \tag{30}$$

Since it was assumed that $d_1 < d_2$,

$$\int_{d_1}^{\infty} f(x)dx - \int_{d_2}^{\infty} f(x)dx = \int_{d_1}^{d_2} f(x)dx$$

$$\int_{d_1+1}^{\infty} f(x)dx - \int_{d_2+1}^{\infty} f(x)dx = \int_{d_1+1}^{d_2+1} f(x)dx$$

$$\begin{array}{ccc} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{array}$$

Also,

$$\int_{d_2}^{\infty} f'(x)dx - \int_{d_1}^{\infty} f'(x)dx = - \int_{d_1}^{d_2} f'(x)dx$$

$$\int_{d_2+1}^{\infty} f'(x)dx - \int_{d_1+1}^{\infty} f'(x)dx = - \int_{d_1+1}^{d_2+1} f'(x)dx$$

$$\begin{array}{ccc} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{array}$$

The penalty cost is never negative, $P \geq 0$. Therefore (30) becomes:

$$\int_{d_1}^{d_2} f'(x) + \int_{d_1+1}^{d_2+1} f(x)dx + \int_{d_1+2}^{d_2+2} f(x)dx + \ldots$$

$$- \int_{d_1}^{d_2} f'(x)dx - \int_{d_1+1}^{d_2+1} f'(x)dx - \int_{d_1+2}^{d_2+2} f'(x)dx - \ldots \leq 0 \quad (31)$$

Inequality (31) will be true if the following inequalities are all true:

$$\int_{d_1}^{d_2} f(x)dx - \int_{d_1}^{d_2} f'(x)dx \leq 0 \qquad (32.1)$$

$$\int_{d_1+1}^{d_2+1} f(x)dx - \int_{d_1+1}^{d_2+1} f'(x)dx \leq 0 \qquad (32.2)$$

$$\int_{d_1+2}^{d_2+2} f(x)dx - \int_{d_1+2}^{d_2+2} f'(x)dx \leq 0 \qquad (32.3)$$

$$\begin{array}{cc} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{array}$$

If $f(x) \leq f'(x)$ for all $d_1+k \leq X \leq d_2+k$ where $k=0$, 1, 2, ... , then inequalities (32.1), (32.2), (32.3), ... will all be true. This would imply that $EC_{1-2} \leq EC_{2-1}$ and sequence 1-2 is optimal. Therefore require that:

$$f(x) \leq f'(x) \tag{33}$$

and determine conditions which are sufficient for this inequality to be true. Inequality (33) may be rewritten as:

$$\frac{1}{S\sqrt{2\pi}}\, e^{\frac{-(X-\overline{X})^2}{2S^2}} \leq \frac{1}{\sqrt{2}\ S\sqrt{2\pi}}\, e^{\frac{-(X-2\overline{X})^2}{2(2S^2)}}$$

Multiply by $\sqrt{2}\ S\sqrt{2\pi}$.

$$\sqrt{2}\ e^{\frac{-(X-\overline{X})^2}{2S^2}} \leq e^{\frac{-(X-2\overline{X})^2}{4S^2}}$$

Take the natural logarithm,

$$\ln \sqrt{2} + \left[\frac{-(X-\overline{X})^2}{2S^2}\right] \leq \frac{-(X-2\overline{X})^2}{4S^2}$$

Multiply by $4S^2$.

$$4S^2 \ln \sqrt{2} - 2(X^2 - 2X\overline{X} + \overline{X}^2) \leq - (X^2 - 4X\overline{X} + 4\overline{X}^2)$$

$$4S^2 \ln \sqrt{2} - 2X^2 + 4X\overline{X} - 2\overline{X}^2 \leq - X^2 + 4X\overline{X} - 4\overline{X}^2$$

$$X^2 \geq 4S^2 \ln \sqrt{2} + 2\overline{X}^2$$

$$X \geq +\sqrt{4S^2 \ln \sqrt{2} + 2\overline{X}^2}$$

Therefore $f(x) \leq f'(x)$ if it is required that:

$$X \geq +\sqrt{4S^2 \ln \sqrt{2} + 2\overline{X}^2} \qquad (34)$$

for $\quad d_1 + k \leq X \leq d_2 + k$ where $k=0,1,2, \ldots$

The smallest value that X may take on is d_1. Therefore, (34) will always be true if it is required that:

$$d_1 \geq +\sqrt{4S^2 \ln \sqrt{2} + 2\overline{X}^2} \qquad (35)$$

Therefore,

$$f(x) \leq f'(x)$$

where X is subject to restriction (35).

This implies that inequalities (32.1), (32.2), (32.3), ... are all true and that:

$$EC_{1-2} \leq EC_{2-1}$$

for $\quad d_1 \geq +\sqrt{4S^2 \ln \sqrt{2} + 2\overline{X}^2} \qquad (35)$

Therefore an optimal sequence for processing the jobs is 1-2 if condition (35) is satisfied. If condition (35) is not satisfied, no statement can be made about the optimality of sequence 1-2.

<u>Proof that the TEC Procedure is Optimal</u>.--It has been shown that for Special Case II the processing sequence 1-2 is optimal if:

$$d_1 \geq + \sqrt{4S^2 \ln \sqrt{2} + 2\overline{X}^2}$$

The TEC procedure can be proven to be optimal by showing that it always produces sequence 1-2 for this restriction. Let

$$f(x) = f(X;\ \overline{X},\ S^2)$$

$$f'(x) = f(X;\ 2\overline{X},\ S^2)$$

Then the approximate total expected cost as defined by the TEC procedure is:

$$TEC_{1-2} = P \sum_{D=1}^{\infty} \int_{d_1+D-1}^{d_1+D} f(x)dx + P \sum_{D=1}^{\infty} \int_{d_2+D-1}^{d_2+D} f'(x)dx$$

$$TEC_{2-1} = P \sum_{D=1}^{\infty} \int_{d_2+D-1}^{d_2+D} f(x)dx + P \sum_{D=1}^{\infty} \int_{d_1+D-1}^{d_1+D} f'(x)dx$$

For this special case, the formulation of TEC is exactly the same as for EC except that $f'(x)$ has a variance of S^2 instead of $2S^2$. Therefore, if inequality (33) can be shown to be strictly true for:

$$f'(x) = f(X;\ 2\overline{X},\ S^2)$$

$$f(x) = f(X;\ \overline{X},\ S^2) \quad ,$$

this will imply that:

$$TEC_{1-2} < TEC_{2-1}$$

and that the TEC procedure is optimal since it always produces sequence 1-2.

Rewrite inequality (33) as a strict inequality and prove that it is still true under condition (35).

$$f(x) < f'(x) \qquad (33')$$

$$\frac{1}{S\sqrt{2\pi}} e^{\frac{-(X-\overline{X})^2}{2S^2}} < \frac{1}{S\sqrt{2\pi}} e^{\frac{-(X-2\overline{X})^2}{2S^2}}$$

Multiply both sides by $S\sqrt{2\pi}$ and take the natural logarithm of both sides.

$$-\frac{(X-\overline{X})^2}{2S^2} < -\frac{(X-2\overline{X})^2}{2S^2}$$

Multiply by $2S^2$.

$$-(X^2-2X\overline{X}+\overline{X}^2) < -(X^2-4X\overline{X}+4\overline{X}^2)$$

$$-X^2+2X\overline{X}-\overline{X}^2 < -X^2+4X\overline{X}-4\overline{X}^2$$

$$0 < 2X\overline{X} - 3\overline{X}^2$$

$\overline{X} > 0$ since processing time is always greater than zero.

Therefore,

$$0 < 2X - 3\overline{X}$$

$$2X > 3\overline{X}$$

$$X > \tfrac{3}{2}\overline{X}$$

Therefore, $f(x) < f'(x)$ if it is required that:

$$X > \frac{3}{2}\overline{X} \tag{36}$$

for $d_1+k \leq X \leq d_2+k$ where $k=0, 1, 2, \ldots$

The smallest value that X may take on is d_1. Therefore, (36) will always be true if it is required that:

$$d_1 > \frac{3}{2}\overline{X} \tag{37}$$

Therefore, if conditions (35) and (37) are both satisfied, the TEC procedure will always produce an optimal sequence. That is,

$$EC_{1-2} \leq EC_{2-1}$$

and

$$TEC_{1-2} < TEC_{2-1}$$

for

$$d_1 \geq + \sqrt{4S^2 \ln \sqrt{2} + 2\overline{X}^2} \tag{35}$$

$$d_1 > \frac{3}{2}\overline{X} \tag{37}$$

Note that if $S^2 \approx 0$, condition (35) reduces to

$$d_1 \geq \sqrt{2}\,\overline{X} .$$

Therefore, the lower limit of (35) is approximately the same as the lower limit of (37) when $S^2 \approx 0$. Very few problems would occur where one of these restraints on d_1 was satisfied but not the other.

Optimality in Less Restricted Problems

No general proof of optimality was found for the 2-job, 1-machine problem. In order to prove that the TEC procedure always produces optimal schedules, certain conditions had to be imposed on the problem. By requiring that the two jobs have equal due-dates and unequal penalty cost functions, the TEC dispatching procedure was shown to be optimal provided that:

$$d \geq - \sqrt{2}\,\bar{X} \quad .$$

By requiring that the two jobs have equal penalty costs and unequal due-dates, the TEC dispatching procedure was shown to be optimal if:

$$d_1 \geq + \sqrt{4S^2 \ln \sqrt{2} + 2\bar{X}^2}$$

and

$$d_1 > \frac{3}{2}\bar{X} \quad .$$

Optimality of the TEC procedure was not shown when the two jobs have both unequal due-dates and unequal penalty cost functions.

These proofs were not extended to the more general cases where the number of jobs, n, is greater than two or the number of machines, m, is greater than one. If these proofs were extended to the n-job, 1-machine problem, it would require the computation of the true total expected penalty costs for $n!$ possible processing sequences to determine the optimum schedule. Then it would be necessary to show that the TEC procedure always produces this schedule. For $n > 2$, this seems to be impractical due to the complexity of the expected cost computations and the large number of possible schedules.

The problem with more than one machine is more complex than the n-job, 1-machine problem. For example, when $m=2$, the total number of technological orderings of the jobs is $(4)^n$, and the number of feasible schedules is even larger.

Proofs of optimality for the job-shop problem with two machines have been previously found only in very special cases of the deterministic job-shop problem (29), (33), and (42).

An alternate proof of the 2-job, 1-machine problem of Special Case I was developed. It was proved that the TEC dispatching procedure will always give an optimal sequence provided that $\bar{X}$ is sufficiently large with respect to S. This implies that there may be some desirable relationship between $\bar{X}$ and S which will assure optimal or near optimal solutions to larger problems. This was not thoroughly investigated. In addition, these proofs for the 2-job, 1-machine problems can be made if the processing-time distribution is an exponential distribution instead of a normal distribution. Other processing-time distributions were not investigated.

CHAPTER IV

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS FOR ADDITIONAL RESEARCH

A survey of the vast amount of literature which has been published on the job-shop machine scheduling problem reveals that almost all research in this area has ignored the true stochastic nature of the problem. Invariably researchers have assumed that the processing-time of each job on each machine is deterministic and precisely known. This assumption has no basis whatever. In a real job-shop much of the work is one-of-a-kind and has never been done before. Even when items are to be processed in lots, the lots are typically very small. In either case, precise estimates of processing-time are almost never available, and the use of processing-time distributions which can be determined from past history would provide a much more accurate representation of a real job-shop.

Three dispatching procedures based on the probabilistic nature of the processing-times in a job-shop have been investigated by this paper. Each of these procedures considers the processing-time distributions of the machines in the shop and attempts to sequence the jobs in such a manner that the total penalty and bonus costs for not meeting due-dates is minimized. Simulation programs were developed for the EC , TEC , and S/O dispatching procedures. These simulations are very general and allow the simulation of more realistic problems than have been previously reported.

Both flow-shop and job-shop problems were simulated using the TEC , EC , and S/O procedures. The TEC procedure was demonstrated to be the most effective of the procedures investigated. The mean cost of the S/O schedules was 51% greater than the mean cost of the TEC schedules and 15% greater than the mean cost of the EC schedules. Optimal schedules were found by the TEC procedure for all of the flow-shop problems simulated, and the TEC procedure produced schedules with costs equal to or less than those produced by the S/O procedure for every job-shop problem. Also, the TEC procedure was proven to be optimal for some special cases of the job-shop problem with 2 jobs and 1 machine.

In addition to the experimental results reported in this paper, a small number of problems were simulated using a Monte Carlo simulation where the time that a job is processed by a machine is a random sample drawn from the processing-time distribution of that machine. Ten replications of each problem were made. The S/O simulation was also provided the processing-times generated by the Monte

Carlo simulation. The distribution of Monte Carlo total penalty costs were compared for the TEC and the S/O simulations and were found to be consistent with the findings of this dissertation.

Because the job-shop problem is so often viewed as a deterministic problem, it has often been formulated as an integer programming problem. Manne's integer programming formulation of the job-shop problem (37) has been extended to include an objective function which minimizes total penalty and bonus costs. This is significant since other integer programming formulations use constraints to require that jobs are completed by their due-dates. These formulations will not produce a feasible solution if it is not possible to meet all of the due-dates. A schedule which has some late jobs could be the schedule with minimum costs. The integer programming formulation presented in this dissertation will find the minimum cost schedule even if due-dates are not met.

The job-shop problem is so complex that no one to date has had any success in obtaining a general solution to the problem. However, additional research into the development of approximate solutions still seems promising. It may be possible to develop stochastic procedures which can be proven to be optimal for a more general problem than the 2-job, 1-machine problem. This should be investigated. There is a relationship between $\overline{X}$ and S which insures optimality of the TEC dispatching procedure for certain problems. Additional analytical work on the nature of this relationship would be in order. An experiment could be designed to use analysis of variance techniques to determine the nature of this relationship if it cannot be determined analytically. Also, an extensive Monte Carlo simulation investigation of probability-based dispatching procedures should be conducted. It is recommended that future research in this area use a processing-time distribution such as the beta distribution instead of the normal distribution or the Poisson distribution.

An interesting application of the techniques investigated in this paper would be the scheduling of jobs on a digital computer. In many ways this problem is similar to the job-shop problem. Many jobs are competing for limited resources. The core storage is limited; the number of jobs that the central processing unit can process simultaneously is limited; and the number of input-out units is also limited. Processing-times are not precisely known, and some jobs are more important than others. The requirement for the facilities of the computer varies from job to job. Thus, this problem is definitely related to the job-shop problem, and similar scheduling techniques should be effective.

LIST OF REFERENCES

1. Akers, S. B. Jr. "A Graphical Approach to Production Scheduling Problems," Operations Research 4, No. 2, April 1956.

2. Akers, S. B. Jr., and Friedman, J. "A Non-Numerical Approach to Production Scheduling Problems," Operations Research 3, No. 4, November 1955.

3. Baker, C. T., and Dzielinski, B. P. "Simulation of a Simplified Job Shop," IBM Business Systems Research Memorandum, August 1, 1958. Also, Management Science 6, No. 3, April 1960.

4. Bakhru, A. N., and Rao, M. R. "An Experimental Investigation of Job-Shop Scheduling." Research Report, Department of Industrial Engineering, Cornell University, 1964.

5. Bowman, E. H. "The Schedule-Sequencing Problem," Operations Research 7, No. 5, September 1959.

6. Brooks, G. H., and White, C. R. "An Algorithm for Finding Optimal or Near-Optimal Solutions to the Production Scheduling Problem," Journal of Industrial Engineering 16, No. 1, January 1965.

7. Campbell, H. G.; Dudek, Richard A.; and Smith, M. L. "Algorithm for the n-Job, m-Machine Sequencing Problem," Management Science 16, No. 10, June 1970.

8. Charlton, John M., and Death, Carl C. "A Method of Solution for General Machine-Scheduling Problems," Operations Research 18, No. 4, July-August 1970.

9. Conway, R. W. "An Experimental Investigation of Priority Assignment in a Job-Shop," Rand Corp. Memorandum RM-3789-PR, February 1964.

10. Conway, R. W. "Priority Dispatching and Work-In-Process Inventory in a Job-Shop," Journal of Industrial Engineering 16, No. 2, March 1965.

11. Conway, R. W. "Priority Dispatching and Job Lateness in a Job-Shop," Journal of Industrial Engineering 16, No. 4, July 1965.

12. Conway, R. W., and Maxwell, W. L. "Network Dispatching by the Shortest Operation Discipline," Operations Research 10, No. 1, February 1962. Also Industrial Scheduling. Edited by J. F. Muth and G. L. Thompson. Englewood Cliffs, New Jersey: Prentice-Hall, 1963, Chapter 17.

13. Conway, R. W.; Maxwell, W. L.; and Miller, L. W. Theory of Scheduling. Addison-Wesley Publishing Company, 1967.

14. Crabill, T. B. "A Lower-Bound Approach to the Scheduling Problem." Research Report, Department of Industrial Engineering, Cornell University, 1964.

15. Eilon, S., and Cotterill, D. J. "A Modified SI Rule in Job-Shop Scheduling," International Journal of Production Research 7, No. 2, April 1969.

16. Fabrycky, W. J., and Shamblin, J. E. "A Probability Based Sequencing Algorithm," Journal of Industrial Engineering 18, No. 6, June 1966.

17. Fischer, H., and Thompson, G. L. "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules," Industrial Scheduling. Edited by J. F. Muth and G. L. Thompson. Englewood Cliffs, New Jersey: Prentice-Hall, 1963, Chapter 15.

18. Gere, W. "A Heuristic Approach to Job-Shop Scheduling." Ph.D. thesis, Carnegie Institute of Technology, 1962.

19. Gere, W. "Heuristics in Job-Shop Scheduling," Management Science 13, No. 3, November 1966.

20. Giffler, B., and Thompson, G. L. "Algorithms for Solving Production-Scheduling Problems," Operations Research 8, No. 4, July 1960.

21. Giglio, R. J., and Wagner, H. M. "Approximate Solutions to the Three-Machine Scheduling Problem," Operations Research 12, No. 2, March 1964.

22. Greenberg, Harold H. "A Branch-Bound Solution to the General Scheduling Problem," Operations Research 16, No. 2, March-April 1968.

23. Hardgrave, W. W., and Nemhauser, G. "A Geometric Model and Graphical Algorithm for a Sequencing Problem," Operations Research 11, No. 6, November 1963.

24. Heller, J. "Some Numerical Experiments for an MXJ Flow-Shop and its Decision-Theoretical Aspects," Operations Research 8, No. 2, March 1960.

25. Holt, C. C. "Priority Rules for Minimizing the Cost of Queues in Machine Scheduling," Industrial Scheduling. Edited by J. F. Muth and G. L. Thompson. Englewood Cliffs, New Jersey: Prentice-Hall, 1963, Chapter 6.

26. Hottenstein, Michael P. "A Simulation Study of Expediting on a Job-Shop," Productions and Inventory Management 10, No. 2, 1969.

27. Ignall, E., and Schrage, L. "Application of the Branch-and-Bound Technique to Some Flow-Shop Scheduling Problems," Operations Research 13, No. 3, May 1965.

28. Jackson, J. R. "Scheduling a Production Line to Minimize Maximum Tardiness." Research Report 43, Management Sciences Research Project, UCLA, January 1955.

29. Jackson, J. R. "An Extension of Johnson's Results on Job-Lot Scheduling," Naval Research Logistics Quarterly 3, No. 3, September 1956.

30. Jackson, J. R. "Simulation Research on Job-Shop Production," Naval Research Logistics Quarterly 4, No. 4, December 1957.

31. Jackson, J. R., and Nelson, R. T. "SWAC Computations for Some mXn Scheduling Problems," Journal of Association for Computing Machinery 4, No. 4, 1957.

32. Jeremiah, B.; Lalchandani, A.; and Schrage, L. "Heuristic Rules Toward Optimal Scheduling." Research Report, Department of Industrial Engineering, Cornell University, 1964.

33. Johnson, S. M. "Optimal Two- and Three-Stage Production Schedules with Setup Times Included," Naval Research Logistics Quarterly 1, No. 1, March 1954. Also, Industrial Scheduling. Edited by J. F. Muth and G. L. Thompson. Englewood Cliffs, New Jersey: Prentice-Hall, 1963, Chapter 2.

34. Johnson, S. M. "Discussion: Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science 5, No. 3, April 1959.

35. Lawler, E. "On Scheduling Problems with Deferral Costs," Management Science 11, No. 2, November 1964.

36. LeGrande, E. "The Development of a Factory Simulation Using Actual Operating Data," Management Technology 3, No. 1, May 1963.

37. Manne, A. S. "On the Job-Shop Scheduling Problem," Operations Research 8, No. 2, March 1960.

38. Maxwell, W. L. "On Sequencing n Jobs on One Machine to Minimize the Number of Late Jobs," Management Science 16, No. 5, January 1970.

39. McNaughton, R. "Scheduling with Deadlines and Loss Functions," Management Science 6, No. 1, October 1959.

40. Mitten, L. G. "Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science 5, No. 3, April 1959.

41. Moodie, C. L., and Roberts, S. D. "Experiments with Priority Dispatching Rules in a Parallel Processor Shop," International Journal of Production Research 6, No. 4, 1968.

42. Moore, J. Michael. "An n Job on One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," Management Science 16, No. 1, September 1969.

43. Nanot, Y. R. "An Experimental Investigation and Comparative Evaluation of Priority Disciplines in Job-Shop-Like Queuing Networks." Research Report 87, Management Sciences Research Project, UCLA, December 1963.

44. Nelson, R. T. "A Simulation Study and Analysis of a Two-Station, Waiting-Line Network Model." Research Report 91, Management Sciences Research Project, January 1965.

45. Nugent, C. E. "On Sampling Approaches to the Solution of the n-by-m Static Sequencing Problem." Ph.D. thesis, Cornell University, September 1964.

46. Palmer, D. S. "Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time -- A Quick Method of Obtaining a Near Optimum," Operational Research Quarterly 16, No. 1, 1965.

47. Pritsker, A. A. B.; Watters, L. J.; and Wolfe, P. M. "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," Management Science 16, No. 1, September 1969.

48. Rowe, A. J. "Toward a Theory of Scheduling," Journal of Industrial Engineering 11, No. 2, March 1960.

49. Sisson, R. L. "Methods of Sequencing in Job-Shops -- A Review," Operations Research 7, No. 1, January-February 1959.

50. Smith, R. D., and Dudek, R. A. "A General Algorithm for Solutions of the n-Job, m-Machine Sequencing Problem of the Flow-Shop," Operations Research 15, No. 1, January-February 1967.

51. Smith, W. E. "Various Optimizers for Single-State Production," Naval Research Logistics Quarterly 3, No. 1, March 1956.

52. Story, A. E., and Wagner, H. M. "Computational Experience with Integer Programming for Job-Shop Scheduling," Industrial Scheduling. Edited by J. F. Muth and G. L. Thompson. Englewood Cliffs, New Jersey: Prentice-Hall, 1963, Chapter 14.

53. Wagner, H. M. "An Integer Linear-Programming Model for Machine Scheduling," Naval Research Logistics Quarterly 6, No. 2, June 1959.

54. Zimmermann, W., and Pfaffenzeller, D. "Simultaneous Determination of Production and Sequencing in Industry," Ablauf-und Planungsforschung 8, No. 1, 1967.