APPLICATION OF CRITICAL PATH TECHNIQUES TO PRODUCTION PLANNING

by

RICARDO DA ROCHA AZEVEDO

Engenheiro Mecanico

Instituto Tecnologico de Aeronautica

1965

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1968

Approved by:

L. E. Grosh

Major Professor

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 PRESENT STATE OF PRODUCTION PLANNING

We propose to study in this report the "state of the art" of production planning in firms manufacturing large and complex products. We will see that the advent of critical path techniques has opened new horizons in production planning procedures. Methods recently developed using these techniques make possible large savings in in-process inventories, delay penalties, and indirect costs. An analysis of the planning system as a whole, including all relevant costs, show how those savings can be achieved.

Before entering into the merit of these techniques, though, the problem should be stated, and some concepts clarified.

1.1.1 Job-Shop Production

Consider a factory organized around a job-shop type of production. This type of production is used for shipyards, large turbines and generators, material-handling equipment, paper machines, marine engines, etc. The characteristics of this type of production are:

I. The product is usually very large, both in physical size and in monetary value; it usually takes a long time to be built, and ties up huge amounts of resources (men, money, machines).

II. There is a heavy engineering content in the product; each hydraulic turbine, for instance, has to be designed in accordance with the specific characteristics of the fall or dam where it will be installed.

III. The product is custom-made. Due to the technical and functional requirements, each product is designed around the specifications set by the client; an order for a heavy over-head crane, for instance, will specify not only the desired lifting capacity, (tonnage) of the crane (or of each one of its hoists), but also length, speed, minimum free height, weight, hoisting speeds, safety measures, materials, etc. Frequently delivery dates are also specified, along with technical requirements. This usually happens when the equipment is just a component of a larger project.

IV. Considering the above mentioned characteristics, it can be seen why this type of production is essentially non-repetitive. It is the rule, not the exception, never to have two orders exactly alike all through the life of the firm. If one thinks, for instance, of a turbine manufacturer, he may receive an order for three or four turbines exactly alike for a specific hydroelectric power plant; but it is highly improbable that, in the future, another hydroelectric plant exactly equal to the first will ever be built again; and it is thus equally improbable that the firm will ever run across an order for a turbine exactly equal to the first three ones.

Job-shop production deals then basically with projects. Projects, in a broad sense, are complex, non-repetitive jobs. The construction of a factory, the building of a large weapons system, are projects. The housewife, planning a formal dinner, is involved in a project; so is the Army, when devising a new missile system, or NASA, planning to send a man to the moon.

It should then be seen that production planning problems in a factory organized for job-shop production are only part of the larger problems of Project Management. This report will then concentrate on the application of critical path techniques, the latest development in the discipline of Project Management, to the more specific problems of production planning in a job-shop manufacturing firm. In the production planning problems, we will further focus attention on the scheduling of the fabrication operations. The whole manufacturing project involves several steps:

- designing the product;
- ordering materials and components;
- fabrication of the parts in the machine shop or welding shop;
- assembling;
- testing;
- disassembling and shipping;
- final assembling on the site.

We can see then that the scheduling of the fabrication is but one of the several sub-problems in the general planning and scheduling problem. It is one of the most crucial; it is also the step where the use of modern planning techniques can be most profitable in this type of production organization.

Let us then review the techniques presently used to solve the scheduling problem in machine shops.

1.1.2 Traditional Scheduling.

The traditional scheduling procedures have been based on several variations of Gantt charts (12) and on a clerical procedure of posting operation and route sheets. This procedure is divided into two (and sometimes three) levels: the project as a whole (including all steps, not only fabrication) was scheduled and controlled by means of a Gantt Progress Chart (see Fig. 1), in what is called (rather improperly) "long-range planning." The fabrication step, in the shop, constitutes the "intermediate and short-range planning," and relies on Gantt Load Charts and clerical posting.

1.1.2.1 Long-range Planning.

The Gantt Progress Chart is the oldest type of Gantt chart. Several key completion dates (modernly called "milestones") are set for the project, and the progress of the operation is recorded periodically in the chart (as a full line, or a contrastingly-colored string). Comparing the full line with the milestone, it can be readily seen which projects are late (thus needing expediting), and what step is causing the delay.

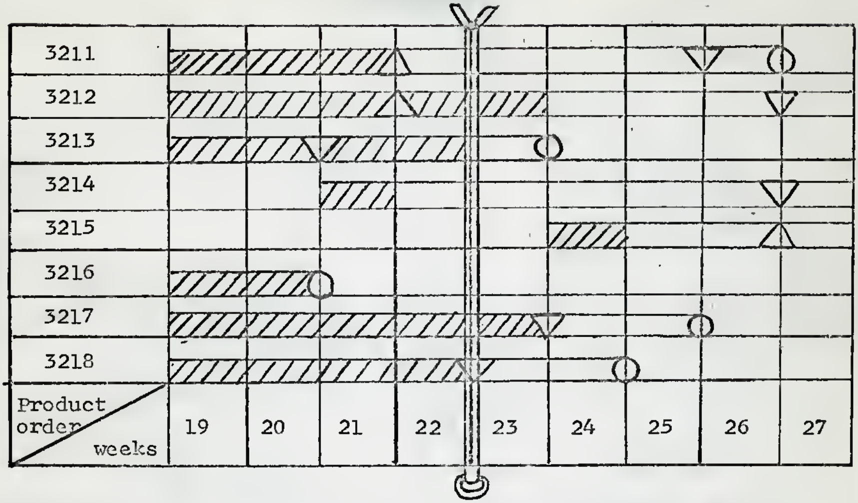


FIGURE 1.--Detail of a Produc-Trol board (a visual display of Gantt Progress Chart)

As an illustration, in Fig. 1., it can be seen that, as of week 22, orders 3211 and 3214 are late; 3212, 3215, 3217 and 3218 are ahead of schedule; 3213 is on schedule, and 3216 has been closed (delivered).

1.1.2.2 Intermediate-range Planning.

There are several possible ways of scheduling production on the intermediate and short-range level. Usually each company develops a procedure appropriate for its particular needs. But all methods traditionally rely on some variation of Gantt Load Charts and clerical posting. The method to be developed here is fairly typical of this type of production, and still is largely used today. As we shall see later, it is markedly inefficient when compared to some more modern methods.

This method separates the intermediate-range from the short-range scheduling steps; its cornerstone is the concept of a delay-unit. A delay-unit is a specified unit of time, chosen by management; only one operation is scheduled per delay-unit in each part. The most usual delay-units are one week (one operation being then scheduled per week), or three days (two operations per week). The choice of the delay-unit depends on the average operations time in the machine shop. The tools used by this method are Machine-group Loading Charts and Route Sheets (or, alternatively, Flow Charts of the fabrication).

Initially, the planner receives from the Industrial Engineering Department, for each new order, a Route Sheet for each part of the project; these Route Sheets contain the sequence of operations to be done on the part, and the time standards for each operation. The planner also obtains, from the long-range planning, the scheduled date for the end of fabrication (or end of assembly). From these dates he then works backwards in time, establishing dates for completion of each sub-assembly, for each part in each sub-assembly, for each operation in each part and, finally, for the delivery of materials necessary for each part.

As an ideal illustration of the routine, see Fig. 1. The planner knows (considering the whole project fabrication flow-chart, of which Fig. 2 is but a detail) the scheduled completion date of operation 45, "assembly of the subcomponents into the final sub-assembly:" it is at the end of week 19. As this operation takes a week, the two subcomponents have to be ready at the end of week 18; so, operations 44 and 43 have to be scheduled in week 18 (or earlier). Operation 42, then, can be scheduled at week 17, and so on. In this way, all operations for every part, and the material deliveries, are scheduled in the shop.

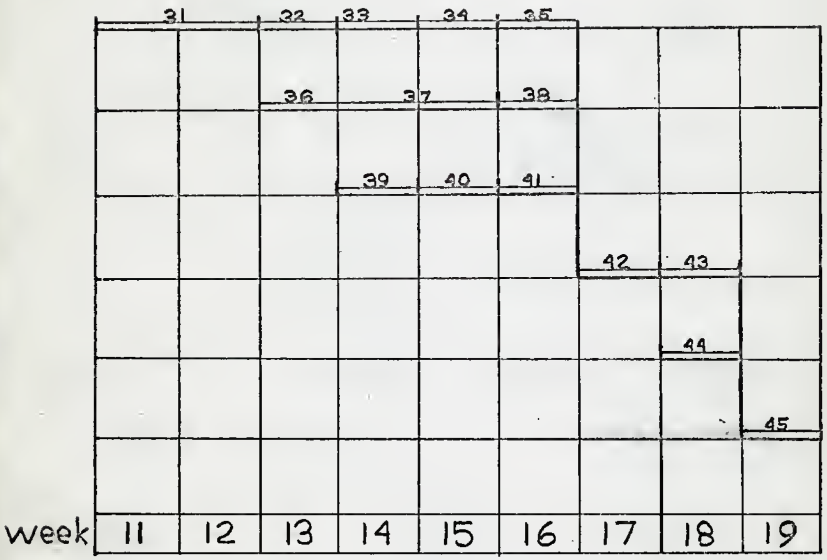


FIGURE 2.--Flowchart for an hypothetical sub-assembly.

The routine just shown is, of course, a great simplification of the actual routine. In practice, the scheduling has to be done both in the Route Sheet (or flowchart) by posting of dates and, simultaneously, by loading the posted operation into a Load Chart.

The Load Chart is another variation of Gantt charts. As used in the production planning department, it shows the future workload, in machine hours, for each group of machines.

In Fig. 3, the detail of a typical Load Chart is shown for two machine groups. The lathe group has four machines, with a capacity of 400 hours per week (in two shifts, each shift with 50 hours per week); the milling machines are a group of six machines, also on two shifts, thus with a weekly capacity of 600 hours.

As an illustration of the actual scheduling process, consider that the planner is trying to schedule operation 34 in week 15, as shown in the ideal schedule of Fig. 2. Suppose that operation 34 is made on a lathe and that, for week 15, the lathe group is already fully loaded (that is, more than 400 hours of lathe operations have already been scheduled for that week), as shown in Fig. 3. Looking at the Load Chart, the planner realizes that week 15 is already fully loaded, but that the operation could be done either in week 13 or in week 16, both not yet fully loaded (see Fig. 3). If he decides on week 13, he posts the date ("week 13") on the Route Sheet for operation 34 and immediately loads in the Load Chart the group lathes week 13 with the standard lathe hours for operation 34. He then proceeds in a similar manner to schedule operation 33. By juggling judiciously with the dates and the weekly loads, the experienced planner then defines a week for each one of the hundreds of operations in the process.

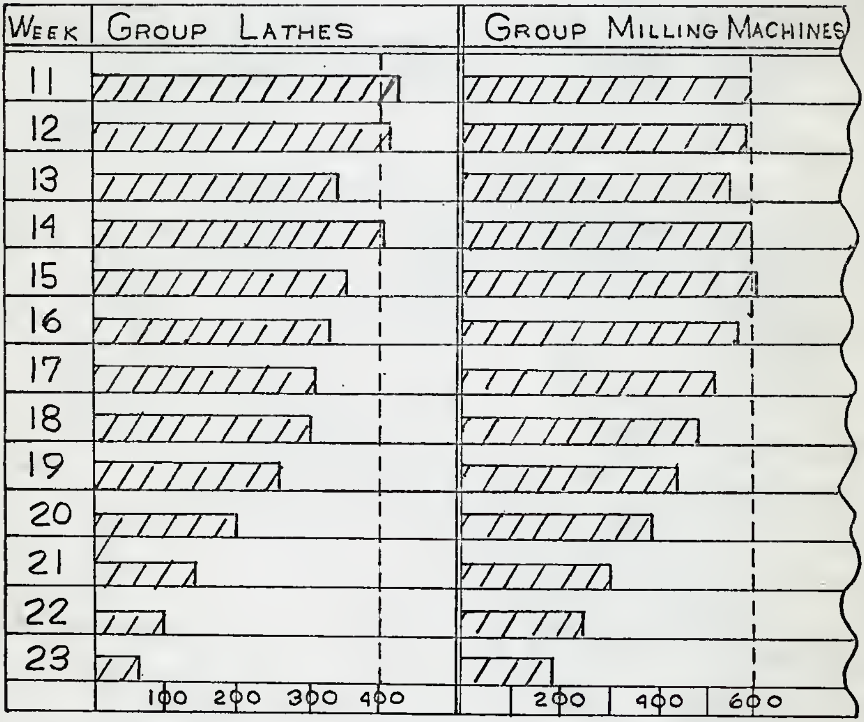


FIGURE 3.--Detail of a load chart.

1.1.2.3 Short-range Planning.

Short-range planning deals with setting of the exact moment, inside the week, when each operation will begin for each machine, inside the group of machines. This planning problem, much simpler than the intermediate-range, can be delegated to the group foremen. He receives every week from the production planning department the list of operations to be made for next week in each part; the foreman then decides the exact loading sequence of each machine, and controls it using more detailed Progress Charts (one line for each machine). The exact sequence is not important for the planner, as long as all operations scheduled for one week are completed in that week.

1.1.3 <u>Evaluation of the Traditional Scheduling Procedure</u>.

The whole system can function smoothly once a judicious amount of slack is introduced by the planner. When operations are delayed, or materials do not arrive in time, the planner can reschedule the project juggling with this slack.

There are two grave drawbacks, though. They are caused by this unnecessary introduction of slack, and by the policy of scheduling one operation per week. Let us analyze in more detail those two disadvantages.

1.1.3.1 First Disadvantage: Introduction of Excessive Slack.

Nothing assures the planner that the project is at all feasible, that is, that the desired completion dates could be respected.

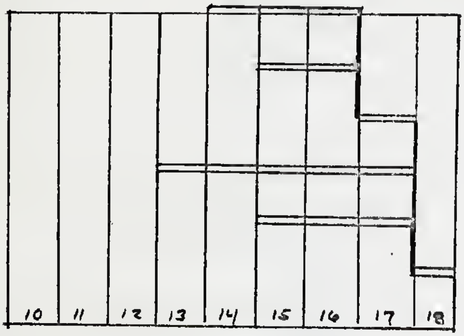
As an illustration of this statement, let us come back to the example mentioned in 1.1.3.2, the scheduling of operation 34 (see again Fig. 2). In the same way as this operation could not be scheduled on week 15, as desired, it may happen that it could not have been scheduled in any earlier week. All weeks preceding week 15 could be fully loaded, so operation 34 would have to be scheduled at some other time. Anyway, if operation 34 is scheduled to some week later than 15, it is obvious that operation 35, that is to be done after 34, will not be executed in week 16, as scheduled; all subsequent operations will thus have to be rescheduled. This means that whenever he schedules an operation, the planner runs the risk of having to begin all over again, rescheduling most of the already scheduled operations.

To get around this obviously cumbersome procedure, the planner introduces slack in its scheduling, that is, he purposefully and systematically leaves slack (idle) weeks between operations; he then does not schedule always successive operations in successive weeks. If trouble develops later, the planner can juggle with these slack weeks, and does not have to reschedule the whole project.

Most of the slack weeks introduced will not be used in trouble-shooting scheduling impossibilities, though. These remaining slacks are wasteful, for they increase the average fabrication span of the parts, increasing then the optimal (minimum possible) amount of capital tied up in work-in-process inventory. This problem of the minimization of work-in-process inventory will be further studied in 1.2. For the moment, it is enough to notice that the amount of capital tied up in work-in-process is inversely proportional to the average fabrication span of the pieces. The fabrication span of a piece is the time elapsed between the instant the piece enters the machine shop ("bought" either from the supplier or from the warehouse) and the instant it leaves the shop ("sold" either to the client, as a part of the final product, or to another department, such as Final Assembly, as a sub-component). Ideally, to minimize in-process inventory, the manufacturer would prefer to buy the raw materials as late as possible, to fabricate the product in the shortest possible time, and to ship it to the client (receiving the bill) as soon as possible. See Fig. 4.

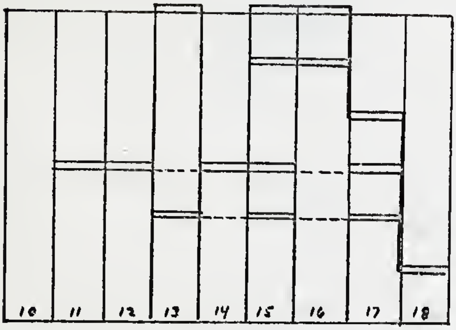
Notice that this idea of the fabrication span is independent of delaying the project. An increase in the fabrication span will only delay the project completion time if the delivery dates of the materials are fixed. Fig. 4 shows two schedules, one optimal, another sub-optimal, for the same project; the average fabrication span of the sub-optimal schedule is larger than the span for the optimal schedule, but the completion time (week 18) is equal for the two schedules. The difference is that the materials for some of the parts in the sub-optimal schedule must be bought earlier (or, brought earlier into the shop).

We can see then, that the introduction of unused slack in the scheduling by the planner has the disadvantage of increasing the capital tied up in the in-process inventory.



Optimal scheduling: each part enters the machine shop as late as possible.

Fabrication span (maximum) = 18 - 12 = 6 weeks



Sub-optimal scheduling: some parts enter the shop before they have to:

Fabrication span (maxmum) = 18 - 10 = 8 weeks.

FIGURE 4.--Optimal and sub-optimal scheduling.

1.1.3.2 Second Disadvantage: Only One Operation per Week.

The second drawback of this traditional scheduling method lies in the simplifying policy of scheduling only one operation per week for each part. This policy forces the execution time of the operation to be one week, regardless of the actual completion time estimated by the Industrial Engineering Department.

If, other things being equal, the planner wants to minimize the average fabrication span of the pieces, he would try to schedule all operations "back-to-back", that is, as soon as one operation in one part ends, the part should be transported to another machine, fixed and the second operation should begin. Ideally, then, if a part requires five different operations, in five different machines, each operation with three hours of standard time, this part could be completed in 5 x 3 = 15 hours of continuous work (less than a day). This state of affairs is, of course, practically impossible, for it would require that all machines be idle, waiting anxiously to begin work on this one part. We have to reconcile our objective of minimizing inventory with other objectives such as leveling the machine load.

Anyway, fifteen hours would be the minimum possible fabrication span for this part. Note that, in the traditional scheduling procedure, the fabrication span for this part would be five weeks, (one week for each operation) or twenty-five working days, as compared to one day for the minimum-time schedule. Obviously, the optimal (minimum feasible) fabrication span lies between one day and twenty-five days. As long as we do not know this optimum, it is difficult to have an objective measure of the inefficiencies introduced by either one of these drawbacks. But we can measure by several ways the average fabrication span for two different actual scheduling procedures, and so recognize the one that is best (nearest to the optimum).

Künzi studied the problem of the measurement of the inefficiencies introduced by the policy of one operation per week. He mentions (7) some experiments done with actual scheduling systems; when the "delay-unit" was changed from one week (six working days) to half a week (three days), the work-in-process inventory decreased considerably in the machine shop studied, although he does not mention specific figures. In Brown Boveri, in Switzerland, where Künzi did his experiments, after the good results obtained with the decrease in the delay unit, management tried a further decrease, from three days to 1.5 days (four delay-units per week). The results were bad, and the machine shop returned to the three day delay-unit, considered to be the best. It seems that the flexibility left to the foreman in the Short-Range scheduling was insufficient, and so excessive delays and idle machine times ensued.

Possibly this "optimum" delay-unit was optimum only for the machine shop in question, and could be a different value for other machine shops. Intuitively, this "optimum" would possibly be related to the average operations time in the machine shop: if a large percentage of the operations take two days to complete, it is obviously useless to try and use a delay-unit of less than three days; probably one week (six working days, in Switzerland) would be better. On the other hand, if the average machining time for each part is one hour only, a delay-unit as small as one day might be enough.

Künzi (7) has also introduced the idea of measuring the actual efficiency of the Production Planning Department by a variation of the traditional Work Sampling technique.

He made several Work Sampling studies of the parts in a heavy machine shop at Brown Boveri; results are given in terms of "productive times," percentage and "improductive times."

Productive times as defined by Künzi cover three types of activities for the part:

1. The part is being cut by the tool bit;
2. The part is waiting in the machine while the operator is fixing it or adjusting the set-up.
3. The part is being transported to or from the machine.

These three "activities" of the part were considered to be essential to all machine operations, and could not be done away with; they were then called "productive."

Improductive times were times when, for one reason or another, the part was not being worked on as it should; those times were considered to be wasteful, and measures should be taken to eliminate them. They comprised:

1. The part is not being worked on because the machine where it should be worked on next is busy; the part is then in the machine queue.

2. The part is not being worked on for miscellaneous reasons, such as technical problems, or the operator is "busy" making a social call on a friend in another machine, or the machine is going through preventive maintenance, etc.

The results obtained in some of Künzi's studies (7) are shown in Table 1. Azevedo and Lauretti (2) verified some of Künzi's results in a similar study.

	Künzi I	Künzi II	Künzi III	Azevedo
Productive Times (%)	34.3	42.6	8	9.7
Waiting in the Queue (%)	37.1	28.2	-	61.3
Improductive Miscellaneous (%)	28.6	29.2	-	28.0

TABLE 1.--Summary of several work sampling of parts in heavy machine shops.

The first two studies of Künzi were made after an extensive reorganization of the scheduling practices in a heavy machine shop manufacturing steam turbines; the "delay unit" used under the conditions studied was three days. He considered that the productive time percentage found under these conditions (34.3% and 42.6%) were exceptionally good. He mentions only partial results in his third study (productive percentage = 8%), and adds that this percentage is more usual in machine shops with inefficient scheduling procedures. Azevedo and Lauretti (2) studied another heavy machine shop, manufacturing hydraulic turbines and large overhead cranes. They obtained results very similar to Künzi's for "improductive time - miscellaneous," and productive times very low (9.7%), more comparable to the figure mentioned in Künzi's third study. The "delay unit" used at the time of Azevedo's study was one week (six working days).

We can roughly equate the waiting time in a machine queue for the part to the efficiency of the scheduling procedures. The larger those times, the lower the scheduling efficiency, other things being equal (such as general workload). It is a fair assumption, then, that the strikingly different results obtained by Künzi and Azevedo for the productive times (under two very similar scheduling procedures) could to a large extent be explained by the difference in the "delay units" used under the two situations.

These results are useful to give an idea as to the amount of inefficiency introduced by the policy of one operation per "delay unit": with the "delay unit" being smaller by half, the waiting time of the parts decreased radically (from around 60% to around 33%), under two very similar scheduling procedures (and average operation times).

The purpose of both Künzi's and Azevedo's studies was not specifically to measure the inefficiencies of the scheduling procedures; their larger aim was to develop, through Work Sampling, a technique of analysis of job-shop type of production that would produce the same results as the traditional methods studies for batch or continuous production. The figures of Table I are then only a by-product of these studies, but they show the possibilities of this approach to the measurement of scheduling inefficiencies.

Another way to measure the efficiencies of scheduling systems would be through simulation of the systems; some studies of scheduling problems through simulation are mentioned in 1.1.5, although in a rather different context.

1.1.3.3 Comments.

Traditional scheduling techniques have been used for a long time in industry, and still are. They have the advantage of producing schedules that are very good as to minimization of idle machine time. On the other hand, their main disadvantages are not minimizing work-in-process inventories, and being rather cumbersome.

After World War II, these techniques began to be challenged by production planning men in search for total optimization of the system. These researchers were not satisfied either with the sub-optimization as to in-process inventories, or with the expenses in clerical work necessary to keep the traditional system going. Analytical and heuristic techniques intended to eliminate these disadvantages have been presented and applied in practical situations with varied success. Let us review some of these techniques.

1.1.4 <u>The Classical Scheduling Problem</u>.

One of the most precise statements of the scheduling problem is due to Giffler (4). He says:

> "There are N jobs and M machines (or facilities). Each job must be processed in a specified order by some subset of the M machines. Given the time to process each job on each machine (and assuming that the processing of a job on a machine must be performed to its completion before either the job can advance to another machine or the machine can start another job), in what sequence should the jobs be processed by each machine if the time to complete all jobs is to be minimized?"

As we can see, this formulation of the scheduling problem, that directed a large part of the research done in scheduling in the last ten years, is more narrow than the scheduling problem that is solved by the traditional method; the classical scheduling problem considers jobs (what we called "parts") independently of each other; it does not take into account the relationships between parts. As an example of this relationship, consider the three jobs (parts) with the sequence of operations shown in the flowchart in Fig. 5. Part I has to pass through operations A, B and C; part 2 has to go through D and E; then, parts 1 and 2 are assembled together (operation F) into part 3, that goes through operations F, G and H.

The classical scheduling problem, as stated above, would not take into consideration the constraint that requires parts 1 and 2 to be completed before work can begin on part 3; the hypothesis on the scheduling problem are that all jobs can begin simultaneously.

Note also that the optimization criterion for the problem as stated by Giffler is: minimize total completion times of all jobs, subjected to the constraints of having only M machines, and that only one job at a time can be worked in one machine. This optimization criterion is not necessarily the most appropriate in practical situations, as shall be discussed in 1.2.

1.1.5 Proposed Solutions to the Classical Scheduling Problem.

When the potential of computers began to be realized, in the early 50's, it was believed that the scheduling problem could be solved by complete enumeration. The computer was thought to be capable of trying all possible schedules, and select the best one. Soon it was realized, that the number of possible alternatives was so enormous that it would take even very fast computers centruies to solve problems of usual sizes.

The disillusionment with this "brute force" philosophy led the way into looking for analytical solutions of the scheduling problem. Johnson (6) published a paper in 1954 that seemed as if it might trigger a breakthrough. He developed a fairly simple numerical method for solving the special case of the classical scheduling problem of having N jobs and only two machines, but in vain; no analytical solution has been found for more than three machines.

Akers and Friedman (1) studied the general N x M problem, and developed logical tests that would eliminate inconsistent schedules and also some obviously non-optimal schedules. Gifler, Thompson and Van Ness (5) devised a systematic method of generating all feasible schedules, based on some of Aker's ideas; these methods seemed to decrease very much the number of possible alternatives for the general problem. A return to the "brute force" approach (complete enumeration) was then tried, using this systematic method of generating feasible schedules: but the number of feasible alternatives was still astromical. Giffler, Thompson and Van Ness (5) did then the next best thing; they modified their program to generate a random sample of the feasible schedules and selected the best of these as an "approximate" solution. This was a Monte Carlo solution in that the sample, if it were increased indefinitely would produce a schedule whose probability of being optimal would steadily increase to one.

With this work, the road to simulation was open. There were two possible ways by which simulation could solve the scheduling problem:

1. It could certainly give an "approximate" solution to the problem, that is, a feasible schedule that had an acceptably small completion time; the amount of computer time needed to solve the the problem periodically in a factory would depend on the precision desired (on how near the optimum schedule we want our solution to be). This approach, although feasible, does not seem to be efficient enough to do practical everyday scheduling in the factory.

2. Simulation could be used to try to discover new empirical principles of scheduling. In this way, intuitive or analytical reasonings could be tested, hoping they would turn out to be the exact solution to the problem. Once this solution was discovered, it could be used in practical situations (without simulation).

In this context, then, a large amount of work on the scheduling problem was done using simulation. This work was directed towards discovering efficient loading rules, that is, rules that would decide on which of the parts, waiting in the queue of a machine, should be loaded first on the machine. If an efficient loading rule were discovered, even not being the exact solution, it was hoped that it could improve sufficiently the simulation procedure so as to be used in practical factory situations.

Several such loading rules were tested by simulation; Moore and Wilson (10) mention twenty-four of them, and summarized the results obtained with these rules, or combinations of them.

Another interesting study was the one by Thompson and Fischer (3); they tried a routine of combining several loading rules in a probabilistic learning way. In their routine, the computer itself decided, based on results obtained with previous schedules, the rule that seemed to be the best for each problem. This was done by modifying the probabilities of choosing each rule at each scheduling step. The probabilities of choosing the rules that, when used in previous schedules, produced a good schedule, were increased after each schedule; similarly, the rules that when used did not produce good schedules had their probabilities decreased.

The above mentioned simulation studies established two main results. The first is that the use of appropriate loading rules can generate efficient schedules. The second is that, up to now, few important analytical results have been achieved through simulation, after a decade of trying.

As researchers began to despair of simulation as the tool to solve the scheduling problems, a drift back to mathematical formulations of the problem ensued.

In fact, one such mathematical formulation seems to have achieved a definite breakthrough in solving part of the scheduling problem. This formulation is the one used in critical path techniques. These techniques are, in the opinion of Giffler (4) the most promising approach to attack the classic scheduling problem.

Several methods of adapting critical path techniques to solve production planning problems have been proposed. The purpose of this report is to survey those methods and evaluate them under the light of improvements over the traditional scheduling techniques.

1.2 THE PROBLEM OF OPTIMIZATION

Before studying in detail the different solution to the scheduling problem, it is time now to state more precisely the problem.

Let us remember then that we are studying production planning methods; the production control function is a study in itself, and will not be mentioned from now on. Also, in planning, we are going to restrict ourselves to the scheduling phase; the problems of product "explosion," choosing of the sequence of the operations and of materials, manufacturing methods, and time estimates will not be touched.

Another point is that we are concentrating also on one specific type of production, job shop production, as described in 1.1. In this type of production, we are further narrowing ourselves to the fabrication stage of job shop production; or, more specifically, to machine shop scheduling.

Four types of cost are involved in the problem. They are:

1. Cost of idle machine time.
2. Cost of in-process inventories.
3. Cost of delays in project completion.
4. Cost of operating the system (systemic costs).

Let us study more closely each one of these costs, and see what are its components for the traditional scheduling procedure.

1.2.1 Cost of Idle Machine Time.

The equipment in machine shops is usually a major capital expenditure, and as such is a fixed cost; depreciation has to be paid for it even if the equipment is not being used. One of the chief aims of production management is then to utilize fully the available equipment. Equipment cannot be "hired" or "fired" in the same manner as men; so the approach usually used is to consider equipment availability as a "constraint" in the scheduling problem, and "equipment utilization" a variable to be maximized. Or, to put it another way, one of management's objectives in production planning is to *minimize idle machine time*. This is then the first "objective function" to be optimized in scheduling.

1.2.2 Cost of In-Process Inventories.

We are more used to hear the word in-process inventories in connection to continuous or batch production; we know it is an expense, and as thus should be minimized. In job shop scheduling, it is an expense too; but its importance has not quite been grasped yet.

The value of in-process inventories comprises all materials that entered the machine shop from other shops or to the client. In a medium-sized machine shop, there are thousands of parts in fabrication at the same time, and the value of these inventories is sizeable.

The cost of these parts can be viewed in three ways:

1. The opportunity cost of the capital tied up in the

parts. Although it varies with industries, usually this cost varies from 5% to 15% of the capital tied up in inventories per year. If we consider the large sums invested, we can have an idea of how important can be a decrease in these inventories.

2. The opportunity cost of the space occupied by those parts. This cost varies widely from industry to industry, but can have considerable importance in a machine shop working near its full capacity; parts clogger up all available space, overflow into aisles, are lost in the machine shop, make transportation and even movement of the workers difficult, increase safety risks, etc. Usually management responds to this chaos by expansion of physical facilities. This often is not the best solution, for even a modification in the scheduling system can sometimes decrease drastically the number of parts in the machine shop, making the expansion unnecessary.

3. The decreased liquidity of the company. Although this is not strictly a production cost, and is certainly not a measurable cost, it is at least an intangible factor to be considered. If too much capital is tied up unnecessarily in the machine shop, the financial manager will find it difficult to find extra sources of funds for working capital financing (with a consequent increase in the cost of the capital).

These inventory costs are not clearly visible to management, for the concept of "opportunity costs" is usually hard to understand; consequently, they have been ignored largely in the design of scheduling systems. The result is that nowadays large opportunities for savings are possible by minimizing work-in-process inventories.

These costs are related to the average manufacturing span of the products. The larger the average manufacturing span, the larger the inventories and its cost will be. To understand this, consider the example of six turbines to be manufactured in a machine shop (partially, of course). Suppose that each turbine requires 10,000 machine-hours of work to completion, and that the machine shop capacity is 20,000 machine-hours per month. Two extreme cases can occur:

1. We can make each turbine in thirty days, or two turbines per month; it would take us three months to finish all turbines. The manufacturing span is then thirty days, and we will have only two turbines at a time in the machine shop.

2. We can make all six turbines at the same time; it will take us equally three months to complete all turbines, but now the manufacturing span is ninety days (three months) for each turbine; the inventory will be three times (six turbines instead of two at the same time during the three months) the one in the first case.

We see then that by scheduling the turbines in two different manufacturing spans, we can vary the inventory by 300%. Note that if we schedule the turbines the second way, the opportunity costs of the extra capital tied up during the three months will be equal to:

$$\begin{pmatrix}\text{opportunity}\\ \text{costs}\end{pmatrix} = \frac{(6 - 2) \cdot 3}{12} \cdot (A \cdot B)$$

where A = Cost of one turbine

B = Opportunity cost of the capital

Supposing $B = 12\%$, the opportunity cost is then 12% of the value of one turbine. Or, in another way, the sale of the turbines will bring (in three months) a revenue of 6A or 2A per month; and the opportunity cost is $0.12 \cdot A$, or $\frac{0.12\,A}{2A} = 6\%$ of the monthly costs. This is an extreme case, of course, but the order of magintude of the possible savings (6% of the monthly production costs) gives an idea of how important it is to minimize manufacturing spans in scheduling.

1.2.3 Cost of Delays in Project Completion.

In machine shop operations, there usually are penalties to be paid to the client for each day or month of delay in delivery after the contractual delivery date. Management is nowadays very much aware of these costs, and usually strives to minimize or, if possible, eliminate these eosts. Note, though, that very often there arc no premiums for early delivery; so it is useless to try to hurry up all projects, or to try to minimize completion times. Management should then strive to begin all projects as late as possible (so as to minimize manufacturing spans), and to finish up manufacture at exactly the due date. Finishing early, inventory costs will increase; finishing late, delay penalties will ensue. Note though that when we talk of project delays, we mean "planned delays," not actual delays. Planned delays are delays incurred because of conflicts during the planning stage, without expediting. All planning is made without considering expediting, using only normal planning routines. Planned delays can turn out to be not actual delays, if expediting is introduced. But the exceptional measures to be taken when expediting is necessary are costly, and should then be minimized. The minimization of planned delays involves both minimization of costs caused by delay penalties and costs caused by expediting. It does not include costs caused by expediting an operation that had no planning delay but got late, as this is a control cost and not a planning cost.

1.2.4 Systemic Costs.

The scheduling system itself may cost a sizeable sum. These systemic costs include not only the salaries of all men involved in the scheduling process, but also the average cost of the mix-ups, of the foreman and supervisor's headaches when a mistake in the scheduling of an operation causes an unexpected, unplanned delay, and the average cost of ensuing necessary expediting. Sizeable savings can also be made in this category of costs by increasing scheduling efficiency. The objective to be pursued here is then the minimization of the systemic (operating) costs.

1.2.5 Optimization.

Summing up, then, there are four different and sometimes contradictory objectives when optimization for a scheduling system is sought:

1. minimization of idle machine time;
2. minimization of in-process inventories;
3. minimization of planned delay penalties;
4. minimization of systemic costs.

With the problem thus stated, we can proceed to study the latest scheduling techniques; in chapter IV we will come back and study how do these techniques compare with the traditional scheduling methods under these four criteria.

To study the scheduling techniques using critical path methods, we will first review the basic concepts and applications of critical path techniques in general, in chapter II; only then, in chapter III, can we look more thoroughly into the scheduling systems derived from these techniques.

CHAPTER II

CRITICAL PATH TECHNIQUES

2.1 GENERAL DESCRIPTION

Critical Path Techniques deal with the problem of Project Management. Project Management, defined by contrast with Production Management, involves the management of only one project; in this sense, a project is a major non-repetitive job; no previously executed job was ever exactly equal, and so the manager cannot rely on a previous plan; he has to use his previous experience on similar projects, applying his judgement to the particular conditions of the project at hand.

Until a few years ago, project management relied chiefly on Gantt-chart type techniques to plan the project; these techniques, useful in repetitive Production Management problems, were markedly ineffective when applied to one-shot, complex projects; their basic drawbacks were not showing explicitly enough the inter-relationships that exist between the several tasks or activities of the project, and not separating planning from scheduling.

In 1958-1959, Critical Path Techniques were introduced. Basically, they involve a graphical portrayal of the interrelationships among elements of a project, and arithmetic procedures which identify the relative importance of each element. The objectives of these techniques are, in a broad sense, to develop an optimal (workable) plan of the activities that make up a project.

Rarely in the past has a new technique attracted so much interest and been so quickly adopted by so many. Several government agencies, American and foreign, specify its use in all subcontractor's bids; likewise, in the construction, chemical and aerospace industries, Critical Path Techniques are already firmly established; manufacturing firms in general use it largely for planning Research and Development projects, equipment overhaul, facilities design and construction, etc. An impressive amount of research has been done in the field: roughly three years after the initial papers were published, articles on the subject, either technical or descriptive, were counted in the hundreds; nowadays, less than ten years after the initial research was done, the author is aware of at least nineteen books dealing specifically with these techniques, and about 300 articles and papers.

This tremendous success is due to two marked advantages these techniques have over the old Gantt-chart: they introduce logical discipline in the planning, scheduling and control of projects; by formally distinguishing the planning from the scheduling functions, they help the planner concentrate his attention on each phase.

The use of Critical Path Techniques causes a sharp increase in the project planning cost, of course; but this increase is more than justified by savings made through concentrating the planner's efforts only on the activities lying along the critical path, and avoiding unnecessary expenditures such as across-the-board overtime.

There are several developments and ramifications of the basic PERT/CPM arithmetic procedure; techniques dealing with the Resource Allocation Problem, Time/cost Tradeoffs, the uncertainty in estimating activity-time, and cost-control procedures will be mentioned as we go along.

2.2 HISTORY

Studies made in the fifties by Marshall and Meckling (8, see 220) of the RAND Corporation, and Peck and Scherer (see 220) of the Harvard Business School indicate that, at the time, management techniques used in the planning and control of projects (techniques based primarily on Gantt-chart variations) were a dismal failure; studying large engineering-oriented projects, it was found that, in the average, final costs were 320% of the original cost estimates for governmental projects, and 170% for commercial projects; the actual delivery times were respectively, 136% and 140%. If we consider that the time span of the typical project studied is measured in years, and cost, in millions, the need for a better technique for the planning and control of large engineering-oriented projects was evident.

Before the advent of PERT/CPM techniques, two other techniques were presented and used successfully (to a degree) in the planning of projects; they included some ideas that were afterwards formally incorporated in PERT/CPM techniques, but in a small and unrelated way; these two techniques were: The Line of Balance Technology (18), developed by Fouch in the Goodyear Company in 1941, and used largely by the Navy Bureau of Aeronautics during World War II; and the Milestone Method, developed by the Navy after World War II. The Line of Balance Technology is still very useful today, complementing network techniques. The Milestone

Method, a development of Gantt-charts, was used largely before the advent of PERT/CPM, but abandoned after it.

Both PERT and CPM arrived in the industrial scene at about the same time, and as essentially independent developments. CPM (Critical Path Method) was developed by Morgan R. Walker of the Engineering Services Division of I.E. du Pont de Nemours Company, and James E. Kelley, of Remington Rand, in 1956-1957. Their work was revised in early 1957 by a larger group, the UNIVAC Applications Research Center, under the direction of Dr. John W. Mauchly. It was first tested at Du Pont by March, 1958, in the planning of the shutdown of a plant for overhaul and maintenance. The test was a success, the shutdown time having been decreased by using CPM from 125 hours to 72 hours, with large savings in costs. The Kelley-Walker arrow diagram and method of calculating the longest or critical path through it are the core of the Critical Path Techniques, and have not suffered any major modification. Originally Kelley developed the method of Time/cost Tradeoff, or expediting a project for minimum costs. Walker and Kelley joined Mauchly Associates in 1958, and have since had an important role in promoting CPM in industry and developing advanced techniques. The original papers of Kelley and Walker (54) and the works of Fulkerson (49) and Clark (47) are basic in this field.

PERT was developed by a research team of the Special Projects Office of the Navy Bureau of Ordinance, because of the recognition of Admiral R. F. Raborn of the need of a new planning and control system for the Polaris missile program. The team, composed of Navy SPO personnel, Lockheed, and consulting firm of Booz, Allen and Hamilton, issued a report in July 1958, containing the basic ideas of PERT (originally Program Evaluation Research Task, afterwards changed to Program Evaluation and Review Technique). The results of the application of PERT on the Polaris program were largely publicized by the Navy, and it has been stated that it decreased completion of the program by two years. D. G. Malcom, J. H. Roseboom, C. E. Clark and W. Fazar, all of the original Navy-sponsored research team, were the authors of the first publicly published paper on PERT (68), in the September, 1959 issue of Operations Research.

PERT was originally time-oriented, as opposed to CPM, that was both time and cost-oriented; PERT includes the treatment of uncertainty in the estimation of the activity times, as opposed to the deterministic approach of CPM; but for these differences, PERT and CPM are one and the same thing. PERT-minded techniques are used typically in the control of large government Research and Development contracts, and CPM-minded techniques, in the construction and chemical industries. The modern tendency nowadays is to use the best of both techniques under the general name of Critical Path Techniques.

Since the original works on PERT and CPM, several management systems and computer programs have been written, these being modifications of either PERT or CPM. These derived techniques will be discussed briefly in 2.4.

2.3 CONCEPTS - BASIC ALGORITHM

The PERT/CPM algoritim's basic idea is to separate the planning from the scheduling phase. The planning phase includes drawing the network and estimating activity times; the scheduling phase includes the arithmetic computations to fix starting and completion times for each activity and for the whole project. Note that those two phases, although separate formally, are not completely unrelated; it is usual, in practical applications, to replan a project after the scheduling phase, if the result of the scheduling is an unacceptable completion time.

2.3.1 Drawing the Network.

The core of PERT/CPM techniques is the graphical representation of the plan as a network. The project should be analysed by an expert, all individual activities that make up the project identified, and the dependence relationships between activities correctly pin pointed. With a complete list of all activities and interdependencies, the network is then drawn.

There are three equivalent ways of drawing networks: the "activity on arrow" system, the "activity on node" system, and the "event" system. The first method is the most widely used, thus we shall use it.

Let us then define some terms:

1. An activity is any portion of a project which conforms to the following statement: it cannot begin until certain other

activities are completed. Activities are graphically represented by arrows in the network.

2. An event is the beginning or ending point of an activity. If an event represents the joint completion of more than one activity, it is called a "meger" event (a "sink", in network vocabulary); if it represents the joint initiation of more than one activity, it is called a "burst" event (or a "fountain"). An event is often represented graphically by a numbered circle.

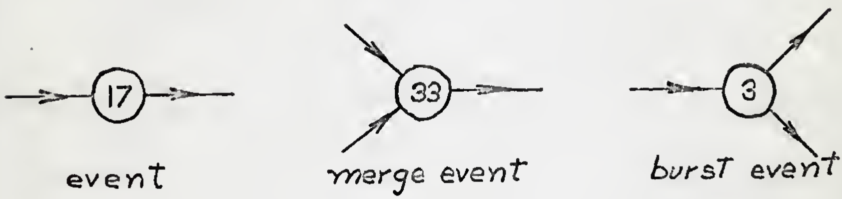


FIGURE 5.--Graphical representation of Events.

3. A network is a graphical representation of a project plan showing the interrelationships of the various activities.

4. A "dummy" activity is an arrow merely representing a dependency of one activity upon another; dummy activities have a zero time estimate, and are represented by dashed-line arrows.

There are two sets of rules for drawing networks; some of them are basic to the network logic, and some are imposed for computational reasons.

Basic rules:

R1. Before an activity may begin, all activities preceding it must be completed.

R2. Length or direction of arrows have no significance whatsoever; arrows imply only logical precedence.

R3. No more than one activity directly can connect two events.

R4. No two events in the network may have the same number.

R5. Networks should have only one initial event (with no predecessor) and only one terminal event (with no successor).

As an example of network drawing, let us consider a project to plan and conduct a market survey (*). Listing the component activities, we have:

Activity	Code	Depends on
Study purpose of survey	A	*
Hire data-collecting personnel	B	A
Design survey questionaire	C	A
Train personnel	D	B,C
Select households to be surveyed	E	C
Make survey and analyze results	F	D,E

TABLE 2.--Market survey-activities

(*) example from reference 221.

The network representing this project would be:

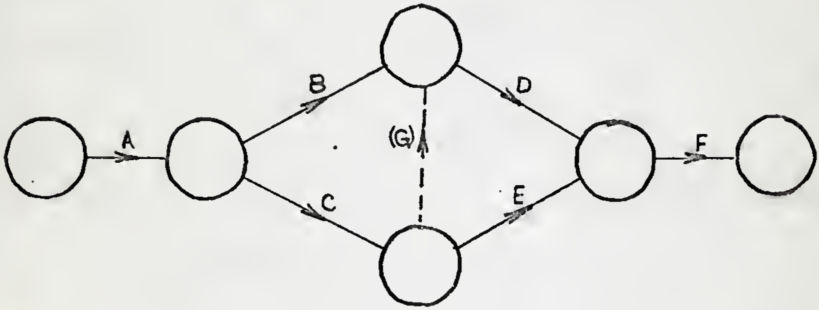


FIGURE 6.--Market Survey-network.

2.3.2 Time Estimates and Level of Detail.

After drawing the initial network, the next step is to estimate the time necessary for each activity. This time should be measured in working days (or weeks, or hours depending on the desired level of detail). These time estimates should be made by the personnel in charge of the project, based on previous experience. One important point is not to let the estimate be influenced by scheduling considerations; such reasoning as "I think that activity F would take three days, but as I know from previous experience that activity D, that precedes it, usually is late one day, I will estimate the duration of F as taking four days" is basically wrong, for it mixes scheduling with planning considerations; if activity D is late, it will be seen afterwards, in the scheduling phase.

The method of time estimation is one of the major differences between PERT and CPM. CPM historically was first used in maintenance projects, where time estimation is done with a relatively low degree of uncertainty; it considers then only one estimate. PERT, on the other hand, was used initially with large Research and Development projects such as the Polaris project, where there is a high degree of uncertainty as to activity duration. PERT then evolved three estimates for each activity: an "optimistic" estimate, a "pessimistic" estimate, and a "most probable" estimate. From these three estimates, an average expected duration is computed. Both methods furnish one overall estimate, (either a deterministic one, as in CPM, or an average one, as in PERT); only one estimate of the activity duration will then be considered at present. The PERT-type estimation will be discussed in more detail in 2.6.

In general, during the time estimation phase, the planner finds errors in the network. This would happen if the time estimate for any one activity, based on the network, seems to be too large, or too small, contradicting common sense. The cause may have been an improper, too coarse or too fine, subdivision of the project into activities. It is often found that a rearrangement of the activities and a redrawing of the network will produce a better representation of the logical work sequence. After correcting the network, the planner then estimates again the times for the new activities. In this sense, then, the two phases (drawing the network and activity time estimation) are interrelated.

An important point should be raised about the proper level of detail for the activities. A project plan can be made in several levels of detail. As an example, in the above cited Market Survey project, seven activities were identified. The plan could have specified, though, only two activities; "plan the survey" and "make the survey." This low level of detail would probably produce too coarse a schedule to be of any use whatsoever. On the other hand, the network could have been refined into several dozens of activities, by subdividing each of the seven activities in their component sub-activities.

These refinements probably would not improve the resulting schedule so much as to be worth the additional planning effort. At first, in the initial networks, it is hard for the planner to identify the proper level of detail to be used in the plan; but after the initial projects it becomes easier and easier.

2.3.3 The Scheduling Phase - The Computational Algorithm.

Once the planner has all the activities and its interdependencies defined, the network drawn and the duration of each activity estimated, the planning phase is complete, and he proceeds to the scheduling phase. The scheduling phase uses the concept of the critical path through a network to determine the optimal schedule for all activities (and, consequently, for the project). The algorithm used to find the critical path is made of a forward pass through the network, and of a backward pass. The following nomenclature will be used:

t = single estimate of mean activity duration time.

T_e = earliest event occurence time.

T_l = latest allowable event occurence time.

ES = earliest activity start time.

EF = earliest activity finish time.

LS = latest allowable activity start time.

LF = latest allowable activity finish time.

S = total activity slack (float).

F = free activity slack (float).

The purpose of the forward pass is to compute the earliest start and finish times (ES and EF) for each activity in the project on an elapsed working day basis. To compute these times, we use three rules:

1. Set T_e (earliest occurrence time) of the (single) initial event as zero.

2. Assuming that each activity begins as early as possible, we set, for each activity:

$$ES = T_e \text{ (for the predecessor event)}$$

$$EF = ES + t$$

3. Whenever two activities converge upon an event (merge event), the later date (<u>larger</u>) is selected as T_e (for the merge event) since the merge event cannot be said to be achieved until the latest of its preceding activities is complete. Or:

T_e = largest of $(EF_1, EF_2, \ldots, EF_n)$ for an event with n merging activities.

If n = 1, then T_e = EF simply.

As an example of the computations of the forward pass, let us recall the project of the Market Survey. Let us assume the following estimated times for each activity:

Activity	Depends on	Time (days)
A	-	2
B	A	5
C	A	2
D	B,C	5
E	C	3
F	D,E	10

TABLE 3.--Market survey - time estimates.

The network is the same as drawn in 2.3.1, with the addition of the numbering of the events:

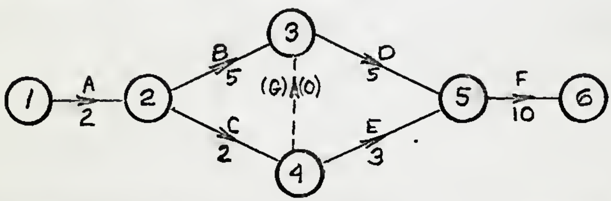


FIGURE 7.--Market survey-network with time estimates.

Using Rule 1, we set $T_e(1) = 0$, for event 1.

Using Rule 2, for activity A ($1 \rightarrow 2$), we have

$$ES(A) = T_e(1) = 0$$

$$EF(A) = ES(A)+t = 0+2 = 2$$

Using Rule 3,

$$T_e(2) = EF(A) = 2$$

Using Rule 2, for activity C (2-4):

$$ES(C) = T_e(2) = 2$$

$$EF(C) = ES(C)+t = 2+2 = 4$$

Using Rule 3,

$$T_e(4) = EF(C) = 4$$

Similarly, we calculate:

$$ES(B) = 2$$

$$EF(B) = 2+5 = 7$$

$$ES(G) = 4$$

$$EF(G) = 4+0 = 4$$

Using Rule 3, now, to find T_e for event 3:

$$T_e(3) = \text{largest of } [EF(G), EF(B)] = \text{largest of } (4,7)$$

$$T_e(3) = 7$$

Similarly

$$ES(D) = T_e(3) = 7$$

$$EF(D) = 7+5 = 12$$

$$ES(E) = T_e(4) = 4$$

$$EF(E) = 4+3 = 7$$

$$T_e(5) = \text{largest of } [EF(D), EF(E)] = \text{largest of } (12,7) = 12$$

$ES(F) = 12$

$EF(F) = 12+10 = 22$

$T_e(6) = 22$

At the end of the forward pass, the planner has then ES and EF for all activities and T_e for all events in the network. He proceeds then to the backward pass, to find similarly all latest allowable start and finish times (LS and LF) for each activity.

He follows three rules, similar to the ones used for the forward pass:

1. Set T_l (latest allowable occurrence time) for the (single) last event in the network equal to the earliest occurrence time computed in the forward pass.

   $T_l = T_e$ (for the terminal event)

   This rule imposes that the terminal event will occur at its earliest expected time, so as to minimize the completion time of the program.

2. Assuming that the activities will start as late as possible without increasing the total time to complete the project, we set:

   $LF = T_l$

   $LS = LF - t$

3. The latest allowable occurrence time for an event (T_l) is the smallest of the latest allowable start times (LS) of the activities bursting from the event in question, for

an event must occur before any succeeding activities begin. So:

T_l = smallest of (LS_1, LS_2, ..., LS_n) for an event with n bursting activities.

Again, obviously when n = 1, T_l = LS.

As an example of the backward pass computation, we have for the Market Survey:

Using Rule 1, for the terminal event 6:

$T_e = T_l(6) = 22$

Using Rule 2:

$LF(F) = T_l(6) = 22$

$LS(F) = LF(F) - t = 22-10 = 12$

Using Rule 3:

$T_l(5) = [LS(F) = 12]$

Similarly:

$LF(D) = 12$

$LS(D) = 12-5 = 7$

$T_l(3) = LS(D) = 7$

$LF(G) = 7$

$LS(G) = 7-0 = 7$

$LF(E) = 12$

$LS(E) = 12-3 = 9$

Using Rule 3, now, for the bursting event 4:

$T_l(4)$ = smallest of $[LS(E), LS(G)]$ = smallest of (9,7) = 7

Similarly:

$LF(B) = 7$

$LS(B) = 7-5 = 2$

$LF(C) = 7$

$LS(C) = 7-2 = 5$

$T_1(2)$ = smallest of $[LS(B), LS(C)]$ = smallest of $(2,5)$ $= 2$

$LF(A) = T_1(2) = 2$

$LS(A) = 2-2 = 0$

$T_1(1) = 0$

After having, in the backward pass, determined LS and LF for each activity and T_1 for each event in the network, the planner computes, for each activity, the total activity slack (S) and the free activity slack (SF).

The total activity slack is the amount of time that the activity completion time can be delayed without affecting the earliest start or occurrence time of any activity or event in the network critical path. More rigorously:

Definition: Total Activity Slack on an activity is equal to the latest allowable time of the activity's successor event minus the ealiest finish time of the activity in question:

$$S = T_1 - EF$$

Definition: Free activity slack is equal to the earliest expected time of the activity's successor event minus the earliest finish time of the activity in question: $S_f = T_e - EF$.

It is equal to the amount of time that the activity completion time can be delayed without affecting the earliest start time of any other activity in the network.

Definition: The Critical Path through a network is the path (sequence of activities) with zero total slack; or, the sequence of activities for which $S = 0$.

The concept of critical path is the most important in PERT/CPM techniques. Each activity on the critical path must receive priority on the scheduling and special attention on the control phase, for if any of these activities are delayed by one single day, the whole project will be delayed by one day; activities not on the critical path can afford delays in scheduling and completion without increasing the completion time for the whole project (as long as the delay is not greater than its total slack).

This is then the algorithm used in PERT/CPM techniques to identify the critical path. To have a formal schedule for the whole project, the planner now transforms the dates of the network (expressed in elapsed working days since the beginning of the project) to calendar days, and has the schedule of the critical path activities. He can schedule activities not on the critical path whenever he wants to, inside the slack; chapter III studies the methods of doing this in an optimal way, considering the problem of resource allocation.

Table 4 summarizes the results for the Market Survey project.

Events	T_e	T_1	Activities	t	ES	EF	LS	LF	S
1	0	0	A (1 - 2)	2	0	2	0	2	0
2	2	2	B (2 - 3)	5	2	7	2	7	0
3	7	7	C (2 - 4)	2	2	4	5	7	3
4	4	7	G (4 - 3)	0	4	4	7	7	3
5	12	12	D (3 - 5)	5	7	12	7	12	0
6	22	22	E (4 - 5)	3	4	7	9	12	0
			F (5 - 6)	10	12	22	12	22	0

TABLE 4.--Market survey - computations.

This algorithm lends itself nicely to computer operations, although, for small and medium-sized networks, the planner will get faster results by doing the computation by hand. If for control purposes the network has to be recomputed several times as the project execution proceeds, then it is better to use a computer for medium and large networks.

There are several refinements of this basic algorithm. In some of them, completion dates can be imposed externally, as input; the critical path is then redefined as the path with less total slack, and the slacks can be negative, zero or positive. Also, a slight modification in the algorithm makes possible the introduction of multiple initial and terminal events in the network, relaxing rule 5 in 2.3.1.

As another example of the computational algorithm, consider the following project:

Activity	Duration t	Depends on	Activity	Duration t	Depends on
A	8	-	G	4	C
B	12	A	H	7	C
C	10	A	I	8	F,H
D	16	C	J	3	E,G
E	8	B,D	K	2	I,J
F	12	B,D			

TABLE 5.--Example of a project. --Activities, time estimates.

The network would be, for this project:

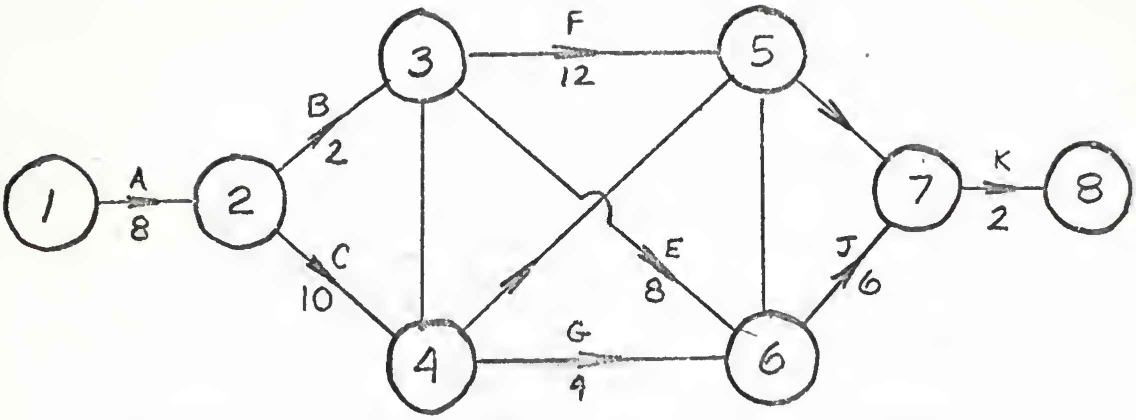


FIGURE 8.--Network for project given in Table 5.

The results are summarized below.

Events	T_e	T_1	Activities	t	ES	EF	LS	LF	S
1	0	0	A (1 -2)	8	0	8	0	8	0
2	8	8	B (2 -3)	2	8	10	32	34	24
3	34	34	C (2 -4)	10	8	18	8	18	0
4	18	18	D (4 -3)	16	18	34	18	34	0
5	46	46	E (3 -6)	8	34	42	43	51	9
6	42	51	F (3 -5)	12	34	46	34	46	0
7	54	54	G (4 -6)	4	18	22	47	51	29
8	56	56	H (4 -5)	7	18	25	39	46	21
			I (5 -7)	8	46	54	46	54	0
			J (6 -7)	3	42	45	51	54	9
			K (7 -8)	2	54	56	54	56	0

TABLE 6.--Computations of the project given in Table 5

The critical path is through A-C-D-F-I-K, and the minimum project duration is 56.

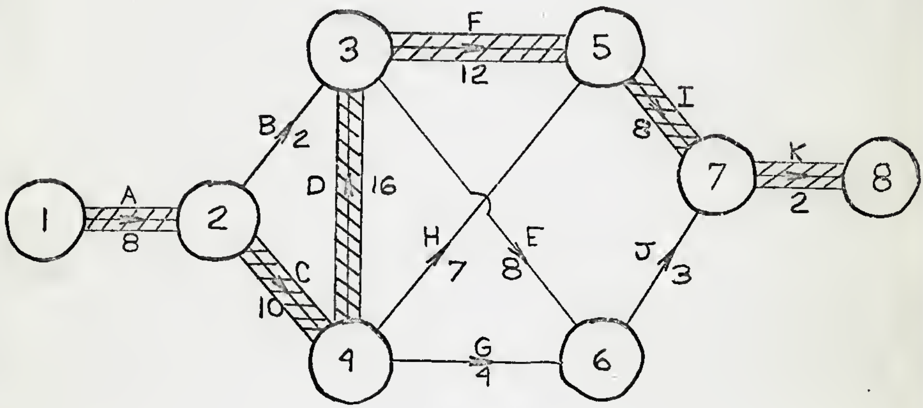


FIGURE 9.--Network of the project given in Table 5, showing critical path.

2.4 CLASSIFICATION OF DERIVED TECHNIQUES

The basic algorithm and concepts of critical path techniques having already been seen, we can proceed to review the developments of these basic ideas. These developments have been directed mainly into four areas:

1. Time/cost tradeoff methods,
2. PERT directed techniques,
3. Resource allocation techniques,
4. General network theory developments.

Time/cost tradeoff techniques are one of the first developments. Their objective is the "compression" of project completion times by allocation of additional resources at minimum cost. They will be studied in 2.5.

PERT - related techniques deal mainly with the problem of uncertainty in activity time estimation, and how this uncertainty will cause uncertainty in planned project completion. This topic will be studied in 2.6.

Resource Allocation techniques deal with the problems of leveling manpower requirements during project life, and of scheduling several projects under stated resource constraints. As this topic is of the utmost importance to machine shop scheduling, it will be studied in depth in chapter III. All methods the author is aware of will be studied and evaluated under the light of the general scheduling problem.

General network techniques are a broad field, and will not be covered specifically in this report. It is related to circuits and network theory in electrical engineering, besides linear and and dynamic programming. The subject is covered quite well in Archibald and Villoria (211) and also by Hillier and Lieberman (229). The works of Elmaghraby (131) and Prisker and Happ (178), are important in this field.

2.5 TIME/COST TRADEOFF

For many projects, some or all activities can be accelerated at the expense of greater direct cost for such activities. When this is so, there are many different ways that activity durations can be selected so that project completion times of the resulting schedules are all equal. However, each schedule would imply a different value of total project direct cost. This chapter treats the different methods that have been devised in Critical Path Techniques to optimize this problem, that is, for any given project duration (or, for any given project acceleration), determining the least costly schedule. Those methods differ basically in the assumptions made about the form of the activity direct cost - duration relationship.

The basic method was devised by Kelley (54) in his original work at Du Pont, and was presented in the paper where he introduced CPM. He developed a parametric linear programming formulation for the problem, and used the Ford - Fulkerson network-flow algorithm to obtain the project cost curve. In a separate article, originating slightly after the first Kelley article, Fulkerson (49) also presents a network-flow solution for the problem.

Both Kelley and Fulkerson made the following assumptions:

1. The "true" time-direct cost relationship of a typical project activity is a continuous, convex function. (Fig. 10)

2. The various functions (for the various activities) are independent.

3. An accurate linear or piecewise-linear approximation to the "true" convex function can be made for each activity.

The data required for this linear approximation consists of two pairs of cost-time estimates for each activity: one pair for the "normal" activity duration and its associate cost, and another pair for an accelerated ("crash") duration and cost. This implies a continuous linear relationship between duration and cost.

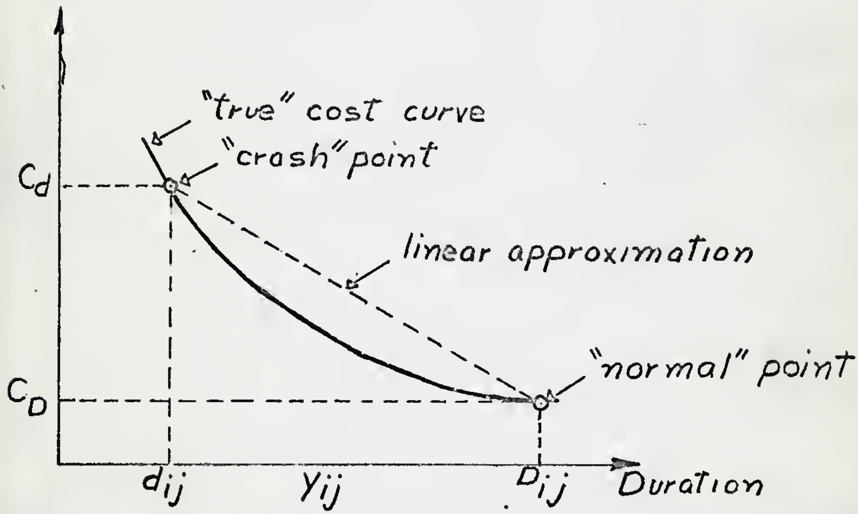


FIGURE 10.--Linear approximation of a convex time/cost curve for activity (i,j).

To define then the linear programming problem, we build an objective function (project direct cost) to be minimized, the independent variables being the activity durations, within the limits defined by normal and crash points.

In Fig. 11, we have, vor activity (i, j), if

$$0 \leq d_{i,j} \leq y_{i,j} \leq D_{i,j} \text{ then } C_{i,j} \text{ the cost of}$$

activity (i,j), is given by

$$C_{i,j} = b_{i,j} - a_{i,j}\,(y_{i,j})$$

where $a_{i,j}$ and $b_{i,j} \leq 0$. The project direct cost, TC, of any feasible schedule is given by

$$TC = \sum_{i,j} (b_{i,j} - a_{i,j} y_{i,j})$$

and the primal linear programming formulation is to minimize TC, that is maximize $\sum_{i,j} a_{i,j}\, y_{i,j}$ subject to the following set of constraints:

$$T_i + y_{i,j} - T_j \leq 0, \text{ all } (i,j)$$

$$d_{i,j} \leq y_{i,j} \leq D_{i,j}, \text{ all } (i,j)$$

$$-T_1 + T_n \leq \lambda .$$

where T_k = earliest expected time of event k; and the project is constrained to start at time 0 and end at some time λ (parameter).

T_k are (unknown) variables, and are not included in the objective function for their cost coeficient is zero; their role in this formulation is merely to insure that the scheduled values of $y_{i,j}$ are feasible from the standpoint of network logic, and that the project duration does not exceed λ.

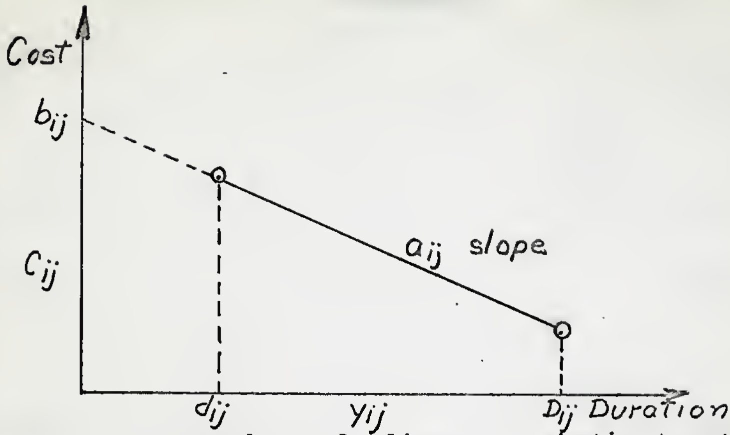


FIGURE 11.--Nomenclature for linear approximation to activity time/cost curve.

With the problem thus defined, it could be solved, using the simplex algorithm, for each λ; with the minimum cost schedule for each project duration λ, we could plot a curve of minimum project cost-duration:

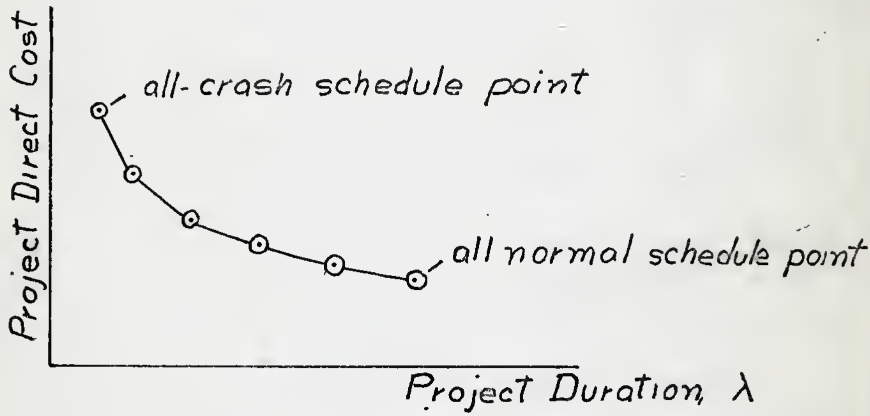


FIGURE 12.--Minimum project cost-duration curve.

Or, if instead of wanting to find the minumum cost schedule for a given project duration, we want to find the project duration with minimum total costs, we can superimpose on the project direct cost-time curve the indirect cost:

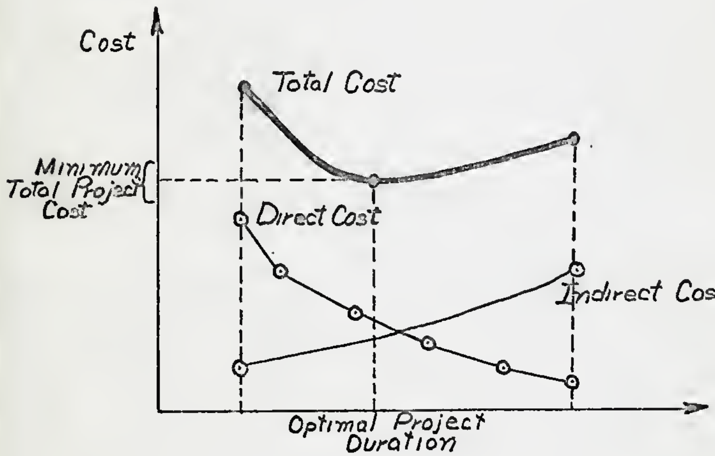


FIGURE 13.--Minimum total cost-duration curve.

Although, this linear programming problem could be solved using the simplex algorithm, such a procedure would be very inefficient, for the number of constraints could be very large (three for each activity in the network). Kelley and Fulkerson tackled the problem in a different way: they formulated the dual of the above primal LP problem, and used the Ford-Fulkerson network-flow algorithm to solve it. Intuitively, the procedure goes as follows: an all-normal schedule and cost are computed letting

$y_{ij} = D_{ij}$ for every activity (i,j). Then the procedure forces a reduction in project completion time (λ) by expediting those critical-path activities possessing the smallest cost/time slopes. That is, not all critical activities are expedited at once. As the project completion time is reduced, different activities become critical and there is a change in the selection of the activities to be expedited. The procedure is repeated until the minimum-duration project schedule is achieved (that is, with all critical activity $y_{ij} = d_{ij}$). The result of the procedure is a project cost curve such as the one shown in Fig. 13.

This CPM procedure is then a rigorous and efficient computational algorithm, which has been programmed for various computers and is available as part of several standard CPM and PERT routines, at least one of which will handle up to 75,000 activities. For a summary of some of the available routines, see Philips (89). Its chief drawbacks are the rather stringent assumptions about continuity, convexity and about the linearization. Clark, using a very similar conceptual approach to the problem, presented an alternative technique (47).

Several methods have been proposed aiming to relax the restrictive assumptions made by Kelley and Fulkerson in the CPM procedure; they are intended to handle nonconvex activity functions as well as discrete time/cost points. One such approach is described in the DOD/NASA Guide PERT/Cost (57), a similar one by Alpert and Orkland (43) and more recently, with additional refinements, by Moder and Phillips (221). The general approach in each case is the same: only discrete time/cost points for each activity are used for feasible data (see Fig. 14), instead of continuous, straight-line estimates.

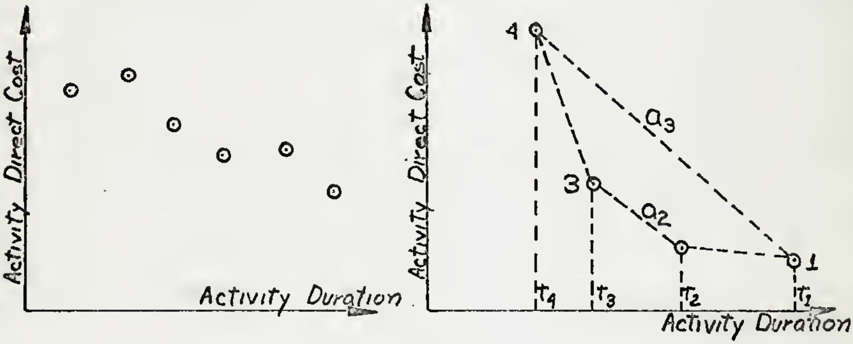


FIGURE 14.--Discrete time/cost points for the activity curve.

For each activity in the critical path, the discrete points are connected by line segments drawn, at each iteration, between a given scheduled point and all the other points. Augmentations of resources are then made to the critical path activity having the least (absolute value of) slope, which is equivalent to buying time where it is cheapest. As each augmentation is made, a new schedule is computed.

Even though the assumptions about the activity functions are much less restrictive than the Kelley-Fulkerson assumptions, this method does not give all possible minimum-cost project duration reduction, as the network-flow solution does. In this sense, it is not an optimal procedure, although it may produce sufficiently accurate answers in practical situations.

A different approach, also designed to handle unrestrained activity fluctuations, is the integer linear programming technique offered by Meyer and Schaffer (56). They start with the primal LP formulation given earlier, then modify it to handle various types of activity time/cost functions; the problem is then solved using integer programming methods.

The disadvantages of this method is that it requires more computational effort than the network-flow equation, for the number of variables and constraint equations increases very rapidly with network size. The result is that even with the largest computers, networks of no more than fifty activities can be handled. It is possible to handle larger networks by decomposing them into subnetworks, but the procedure is complicated and time consuming; the integer LP approach is not then practical for the majority of problems, and should only be used when very costly activities are involved, or other considerations justify the extra effort required.

Several other approaches have been used, each of which hinge upon assumptions about the activity direct cost-duration relationshop. The works of Jewell (53) and Berman (45) are examples of interesting approaches; a good overall discussion of these methods is found in (24).

Finally, some methods have been recently proposed to reduce the total effort in using the Fulkerson network-flow algorithm, such as in (51, 52, 58 and 59). Up to now, though, the Kelley-Fulkerson method is still the most widely used, as very little research has been done on the actual form of the activity cost/time function in practical projects, and on comparing and evaluating these several alternative techniques by simulation.

2.6 UNCERTAINTY (PERT)

2.6.1 Introduction.

In large research and development projects such as the polaris missile project, the planner faces a rather different problem from the one that has been discussed up to now. The basic difference is that chance plays a much larger part in the completion time of the activities. Activities in such large projects often depend on man's creative ability, and on undefined technical difficulties; consequently, the estimate of their length must be an uncertain one. Such an approach as the deterministic estimation of activity duration time would be wholly inadequate for such activities.

The PERT algorithm (68) then presents a method by which these large uncertainties in activity time estimation can be handled, in the context of network theory.

2.6.2 Conventional PERT Procedure.

PERT handles uncertainty by assuming that the duration of an activity is a beta-distributed (Fig. 15). The probability density function of the beta-distribution is

$$f(t) = K.(t-a)^{\alpha} (b-t)^{\beta}$$

PERT uses two time estimates (the "optimistic" time and the "pessimistic" time) to specify a and b.

The optimistic time (a) is the shortest time in which the activity could ever be completed; the <u>pessimistic time</u> (b) is the longest time the activity could ever take to complete (barring "acts of God").

A third estimate, m, the <u>most likely time</u>, is also obtained. The value of m is the mode of the distribution, and this value is used to determine the two parameters α and β.

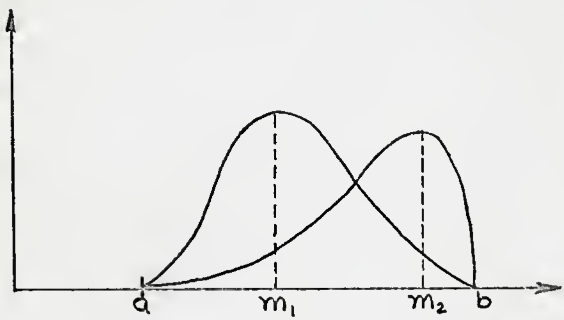


FIGURE 15.--Beta-distributions.

Certain cares must be taken in the estimation of these three parameters for each activity, so that they will be independent both of each other and of scheduling considerations. Also, it should be noted that these estimates use the basic assumption that the task will be done only with the resources originally allocated to them, that is, no expediting of the activities is going to be considered.

Once the planner has three estimates for each activity, a beta-distribution has then been defined for each activity, and the next step is to calculate the mean and the variance of the distribution.

The PERT system approximates the mean (t_e) of the distribution, and the variance (V_t) as:

$$t_e = \frac{(a + 4m + b)}{6} \quad \text{, and}$$

$$V_t = \left(\frac{b - a}{6}\right)^2$$

The formula for the mean implies giving four times more weight to the mode than to the extremes; it is a linear approximation of the exact solution, which includes finding the roots of cubic equations.

The assumption that the standard deviation is one sixth of the range is based on the Tchebicheff inequality; it states that at least 89% of <u>any</u> distribution lies within three standard deviations of the mean. For the normal distribution, the figure is 99.1%.

Having the mean and the variance of the times for each activity, the planner then proceeds to determine the mean and variance of the whole project, (or, to put it another way, to find the probability of meeting a scheduled completion date for the project). In this step, the PERT procedure uses the Central Limit Theorem, which states that the sum of <u>n</u> independent random variables (the times for each activity) is a random variable (the completion time for the whole project) having the shape of a normal distribution; the mean of the new random variable is the sum of the means of the independent variables, and the variance is the sum of the variances.

Based on this Central Limit Theorem, then, PERT treats the means of the individual activity times deterministically and goes through the computations described in 2.3 (forward and backward passes) to determine the critical path. Once the critical path is found, the mean project completion time is taken as the sum of the mean times for the critical path activities, and its variance, as the sum of the variances of the critical path activities. When there are two or more critical paths converging on one event, with different variances, the largest is taken as the event variance.

Using this procedure, the planner then finds the mean and variance of the total completion times of the project; with these two, using a table of the normal distribution, he can predict the probability of meeting a stated completion deadline.

As an illustration of this PERT procedure, consider the network shown below.* The numbers for each activity are a, m and b.

---

*Example taken from Moder and Phillips, p. 208.

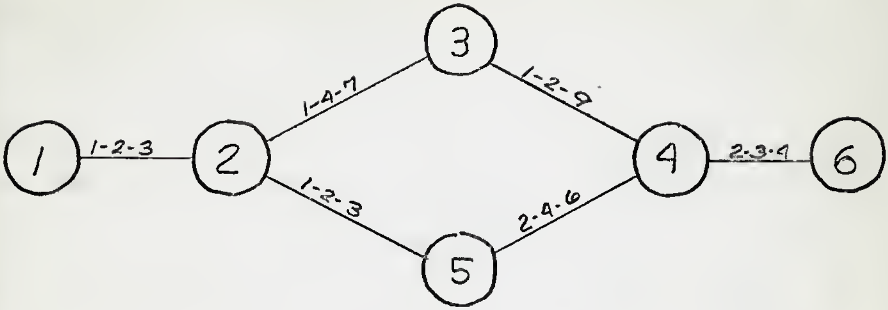


FIGURE 16.--Example.

Using the two expressions for t_e and V_t, we get:

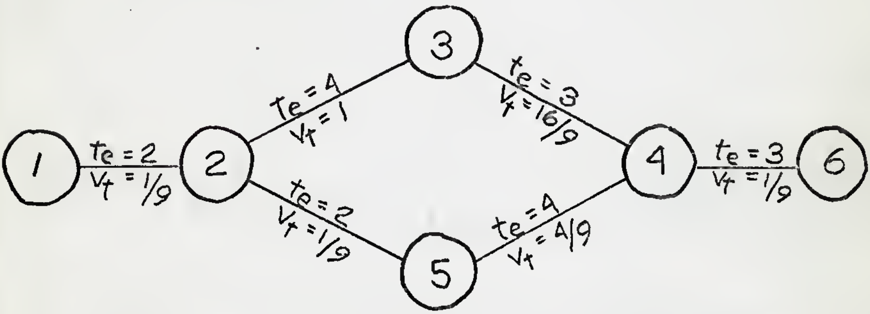


FIGURE 17.--Example of Figure 16, with PERT-estimated means and variations.

It can be readily seen, even without using the formal forward and backward computations, that the critical path is through events 1, 2, 3, 4, and 6. The completion time for the project (mean) is then 2 + 4 + 3 + 3 = 12; as to the variances:

V_t (event 1) = 0

V_t (event 2) = 0 + 1/9 = 1/9

V_t (event 3) = 1/9 + 9/9 = 10/9

V_t (event 5) = 1/9 + 1/9 = 2/9

V_t (event 4) = 10/9 + 16/9 = 26/9

V_t (event 6) = 26/9 + 1/9 = 27/9 = 3

Note that event 4 is a merge event; its variance is computed along the critical path, or 1-2-3-4, and not along 1-2-5-4.

For the whole project, then, we have:

$t_e = 12, \quad V_t = 3$

The probability that the project would be completed by time 12 is then 50%; by time 14, is 88% ($z = \frac{14-12}{1.73} = 1.16$; for $z = 1.16$, in the normal tables, we find $p = 0.88$). Note, thus, that these probabilities are the probabilities that the project be completed <u>without expediting</u>; if expediting is considered, then the probabilities would be much larger. Also, in this particular example, the Central Limit Theorem (that presupposes an infinite sequence of activities) does not hold very well, for we have a sequence with only four activities. But we hope that it helped to make clear how PERT is used in complex networks.

This concept of the probability of completing the project by a scheduled date can be easily extended to include the probability of reaching any event (milestone) in a scheduled date (see Moder and Phillips). (221)

2.6.3 <u>Analytical Study of PERT Assumptions</u>.

The PERT assumptions will be studied in two steps: first, the study of the possible activity-based errors; then, the study of the network configuration-errors.

2.6.3.1 Activity-Based Errors.

The assumptions made in the PERT procedure to calculate the mean t_e and the variance V_t of each activity can lead to three distinct kinds of errors:

E-1. The true activity time distribution is not known; it can be said to be unimodal, continuous, and that it touches the abscisses in two non-negative points, and that is all; to say that the distribution is a beta distribution, when it can be otherwise, introduces a first error in the values of t_e and V_t. Mac Crimmon and Ryavec (69) calculated that the maximum possible error introduced by this assumption can be, for t_e, 33%; in the more usual cases, where the mode is near 1/2, the worst error is around 11%. Lukaszewicz (67) stated that this last value is wrong, and that even with the mode near 1/2, the error can be as large as 25%. For V_t, an error of 17% is possible.

E-2. A second type of error is introduced by the approximate expressions for the calculation of t_e and V_t. That is, even if the activity times were exactly a beta distribution, the use of the approximate expressions would introduce an error. Mac Crimmon and Ryavec (69) calculated that, for extreme cases, these errors can be as large as 33% for the mean and would be reduced to 4% and 7%, respectively, in most cases.

E-3. A third type of error is the error in the estimation of a, b and m. Even if the distribution were exactly a beta distribution, and if t_e and V_t were calculated using the rigorous formulae for the beta-distribution (and not the approximate ones), there could be errors in t_e and V_t caused by errors in the estimation of a, b and m. Mac Crimmon and Ryavec calculated that errors of 10% to 20% in the estimation of these parameters could cause an absolute error of 30% in the value of the mean, and 15% in V_t.

It has to be said that since many of the cases considered -- although theoretically possible -- are rather extreme, these three errors could, probably, be reduced to perhaps 5% in the mean, and 10% in the variance. Also, as the errors can be positive or negative, some degree of cancelation is expected to occur when all activities are combined in the network.

Even allowing for these facts, these errors still seem very large, and can cast considerable doubt on the validity of the whole PERT procedure. Mac Crimmon and Ryavec (69) even suggested that if, instead of the beta-distribution, a simple triangular distribution were considered, the planner would be better off, for the E-1 and E-3 errors would stay on the same levels, and E-2 errors would be zero, for the parameters of the triangular distribution can be readily and rigorously determined.

Moder and Phillips (221) proposed a variation of the conventional PERT method tending to decrease E-1 and E-3 errors. Their method is based on defining a and b as percentiles of the beta distribution, rather than extremes.

Still another assumption made in the PERT procedure is related to the measure of skewness of the distribution. Even though Malcom et al. (68), in the first publicly published PERT paper, stated that "no assumption is made about the position of m relative to a and b," Grubbs (78) indicated that the assumptions about the beta distributions are very restrictive, implicating a coefficient of skew of either ± 0.707 or zero. Donaldson (65) proposed a method to avoid this inconsistency and relax this "hidden" PERT assumption; his method involves an estimate of the mean, instead of the mode (as in the conventional PERT procedure) and also assumes that the beta distributions are tangent to the x - axis at a and b. Coon (64) showed how Donaldson's method could be extended to the case where one has only an estimate of the mode, as in the conventional PERT method, and not of the mean,

2.6.3.2 Network Configuration - Based Errors.

Even if the values for t_e and V_t for each activity were free of any error, the PERT procedures still would introduce an error in the determination of the total project mean and variance. This error is caused by the assumption that the mean and variance of the critical path are the mean and variance of the project. This assumption makes the PERT value of the project completion mean time to be smaller than the true mean, that is, the PERT value is always biased optimistically; the variance is also biased, but in both directions.

To explain this fact, it should be remembered that, once the mean and variance of the individual activity times are calculated, the PERT procedure gives up its stochastic approach and proceeds to determine the critical path, using a deterministic approach; it takes the mean and variance of the project completion time as being the sum of the mean and variance of only the activities on the critical path. This is not true, for the mean and variance of the project is always greater than the mean and variance of the critical path (considering its expected values), as shown by Mac Crimmon and Ryavec (69). In fact, not only the critical path, but all other paths through the network have a probability of turning out to be the longest path, after the project is completed; and this probability should be taken into account when of the calculation of the mean and variance of the whole project. The "true longest path" through the network is the path with the longest actual completion time; this path can sometimes be the critical path, but it can also be any other path; and, as the longest path duration is obviously always greater or equal to the duration of the critical path, the expected true mean of the project completion time is always greater than the PERT-found mean.

This error in the PERT-calculated project mean and variance varies with the network configuration; the more paths in parallel in the network, the larger it will be. Also, if many non-critical paths each have a duration approximately equal to the duration of the critical path, this error will tend to be large; conversely, the more slack there is in each of the non-critical times, the smaller will be the error.

Several approaches have been proposed to cope with this error. Clark (63) introduces a correction for this error; his procedure seems to be feasible, though rather cumbersome, but no computer programs seem to have included it yet. Mac Crimmon and Ryavec (69) suggest that the whole PERT approach should be modified, with the substitution of the conventional "critical path" concept for a "critical activity" concept; his ideas have been further developed by Welsh (77), including the concept of "super-critical arcs"; Welsh further suggests types of algorithms that could be used to solve the project scheduling problem thus stated.

2.6.4 Comments

There seems to be a considerable amount of disagreement on the problem of uncertainty. The development of new ideas has been rapid, and there has not been enough time for all of the conclusions to be tested; noticeably, simulation is very useful in testing several alternate methods; an insufficient amount of work has been done in this area.

One of these simulations, made by Van Slyke (74) indicated rather surprisingly that the probability of meeting a scheduled completion time was given by the conventional PERT method with a fair degree of accuracy, in spite of all its assumptions. This could be explained by compensation of activity-based errors, and by the special network configuration. It would be interesting to know if this accuracy would hold for different network configurations.

In conclusion, as the basic PERT assumptions are being more and more challenged on theoretical grounds, the PERT statistical approach is disappearing from many PERT output reports, although it enjoyed great popularity shortly after its inception. If this approach, for which there is a real need, is going to stage a comeback or not will depend on the development of some basic improvements on the method, even if at the cost of simplicity.

CHAPTER III

RESOURCE ALLOCATION

3.1 IMPORTANCE IN PRODUCTION PLANNING

The objectives to be reached in optimizing machine shop scheduling under job shop production are, as shown in 1.2:

1. Minimization of idle machine time,
2. Minimization of in-process inventories,
3. Minimization of delay penalties,
4. Minimization of systemic costs.

Traditional scheduling procedures, based on Gantt charts and clerical posting (as described in 1.1.2) do fairly well as to objectives 1 (minimization of idle machine time) and not so bad as to 4 (minimization of systemic costs). They are very bad as objectives 2 and 3 (minimization of in-process inventories and and delay penalties) are concerned.

New methods that could produce schedules nearer to objectives 2 and 3 have been searched intensively, as exposed in 1.1.5. Of these, Critical Path Techniques seem to be the most promising, for they allow for the first time the (partial) attainment of objectives 2 and 3 in a logical and systematic way.

Critical path techniques do not fare so well in general towards objective number 4: by disciplining the planning process, they may (or may not) increase planning costs. Mechanization and data processing often can keep the systemic costs under an acceptable level. As to objective 4, then, critical path techniques do not do badly.

Considering now objective 1 (minimization of idle machine time), we see that critical path techniques do very badly indeed (at least the ones described in chapter II). They give attention only to minimization in-process inventories and delay penalties, and consider that resources are freely available to do the tasks whenever they are scheduled. This approach was initially developed to help in the planning and control of <u>one project</u> only. Specifically, as was seen in 2.2., PERT was developed to help the planning and control of the Polaris project by the Navy, and CPM, to plan the overhaul and maintenance of chemical plants at Du Pont. In these projects, minimization of project completion was of the utmost importance, and resources were made available in the necessary amount, regardless of their level.

The scheduling problem in machine shops involves several simultaneous projects, instead of only one; resources (men and machine time) are not unlimited, although they can be varied to a small extent. The results of the first tentative applications of PERT and CPM to multi-project scheduling in machine shops were then unsatisfactory, for they produced resource utilization profiles that were completely unnacceptable for everyday scheduling; for instance, for a certain machine tool, 2,000 machine hours might be scheduled for one month, and zero for the next two or three months. This state of affairs was unacceptable to machine shop conditions, for objective number 1 (minimization of idle machine times) is of great importance here.

Several tentative adaptations of basic critical path methods have been tried, to solve the resource allocation problem in factory scheduling. Some of these methods are of a very sophisticated nature, and give very good results <u>under all four of the scheduling objectives</u>. All the methods the author is aware of will be surveyed in this chapter with a brief explanation of their mechanism, wherever possible.

Two sharply differing general approaches have been followed in tackling the resource allocation problem. They are Resource Leveling, to be studied in 3.2, and Scheduling under Stated Resource Constraints, to be covered in 3.3. An evaluation of these techniques is made in 3.4.

In resource leveling, the problem is stated in terms of minimization of resource level variation under project completion time constraints. To put it another way, given a stated (minimum) project completion time (obtained using the basic PERT/CPM algorithm), what is the best possible schedule of project activities so as to minimize ("level") variations in the resource profile? The constraint here is then <u>time</u>, and the objective function is expressed in terms of <u>resources</u>.

Under resource constrained techniques, the approach is the opposite. Given resource constraints, what is the best possible schedule of project activities so as to minimize project completion times? The constraint here is then *resources*, and the objective function is expressed in terms of *time*.

As will be seen in 3.4, the second approach (resource constraint techniques) is the more realistic in machine shop scheduling. As a final note, several time/cost tradeoff methods have been published under the title of "resource allocation." In the sense used in this report, resource allocation deals only with either leveling techniques or resource constraint techniques.

3.2 LEVELING TECHNIQUES

3.2.1 The Burgess-Killibrew Method.

The Burgess-Killibrew method (21, 221) is a systematic approach to the one-resource leveling problem. The method consists essentially of comparing alternate schedules obtained by sequentially moving, in time, slack activities and computing the resulting profile. The measure of effectiveness used for comparison of schedules is the sum of the squares of the resource requirements. This measure has the property of becoming smaller as the variation in resource requirements from time-unit to time-unit becomes smaller.

The procedure to be followed is:

Step 1: List the activities so that arrow head numbers are in ascending order; when two activities have the same arrow head number, list them by ascending tail numbers. Schedule all activities to begin at their earliest starting times. Prepare a bar chart for the activities showing their total and free slack (see Tab. 8).

Step 2: Starting with the last activity in the list, schedule it to give the lowest total sum of squares of resource requirements for each time unit. This is done by moving the activity one time unit at a time to the right (inside the slack) and computing the sum of the squares, until the minimum is found.

If more than one schedule gives the same total sum of squares, choose the one in which the activity begins as late as possible.

Step 3: Holding the last activity fixed, repeat Step 2 on the next to the last activity in the list, taking advantage of any slack that may have been available to it by the rescheduling on Step 2.

Step 4: Continue Step 3 until the first activity in the list has been considered; this completes the first rescheduling cycle.

Step 5: Repeat Steps 2 through 4 until no further reduction in the sum of squares is possible. Note that only movements of an activity to the right (schedule later) are permissible under this scheme.

For an application of this procedure, consider the following network (taken from Moder and Phillips, ref. 221).

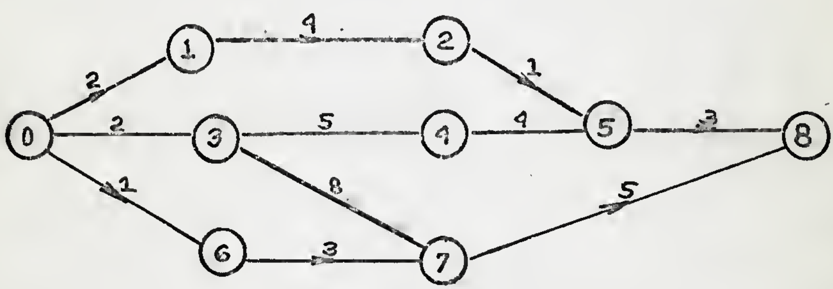


FIGURE 18.--Project network with durations.

Duration	Activity	ES	EF	LS	LF	S	S_f	Event	T_e	T_l
2	0-1	0	2	5	7	5	0	0	0	0
4	1-2	2	6	7	11	5	0	1	2	7
1	2-5	6	7	11	12	5	4	2	6	11
2	0-3	0	2	0	2	0	0	3	2	2
5	3-4	2	7	3	8	1	0	4	7	8
4	4-5	7	11	8	12	1	0	5	11	12
1	0-6	0	1	6	7	6	0	6	1	7
3	6-7	1	4	7	10	6	6	7	10	10
8	3-7	2	10	2	10	0	0	8	15	15
5	7-8	10	15	10	15	0	0			
3	5-8	11	14	12	15	1	1			

TABLE 7.--Determination of the critical path.

The critical path is through activities 0-3, 3-7, 7-8, as can be quickly verified in Tab. 7.

Going through Step 1, we have Tab. 8; assume that the crew requirements (resource X) for each activity are as stated in the table.

Notice that the activities are listed in Tab. 8 by their arrow head numbers, as specified in Step 1. All activities have been scheduled on their earliest start dates. The column "scheduled" refers to the final schedule; at this stage, only the activities on the critical path (0 - 3, 3 - 7 and 7 - 8) are already scheduled, for they cannot be moved without increasing the total completion time for the project; these critical path activities are then "fixed," that is, scheduled. The bar chart in Tab. 8 portrays the schedule after Step 1: the crisscrossed bars indicate critical path activities.

Duration	Scheduled		Earliest		Slack		Crew Requirements	Activity No.
	Start	Finish	Start	Finish	Total	Free		
2	-	-	0	2	5	0	3	0-1
4	-	-	2	6	5	0	-	1-2
2	0	2	0	2	CRITICAL		6	0-3
5	-	-	2	7	1	0	-	3-4
1	-	-	6	7	5	4	4	2-5
4	-	-	7	11	1	0	2	4-5
1	-	-	0	1	6	0	4	0-6
8	2	10	2	10	CRITICAL		4	3-7
3	-	-	1	4	6	6	5	6-7
3	-	-	11	14	1	1	-	5-8
5	10	15	10	15	CRITICAL		2	7-8

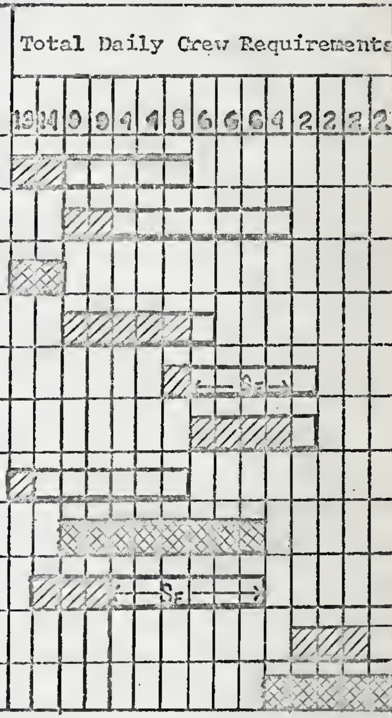


TABLE 8.--Determination of the critical path.

Going to Step 2, we try to move the last activity in the list (7-8); but this activity is already scheduled, being on the critical path, and cannot be moved.

Repeating Step 2, we try to move 5-8. This activity has crew requirements of zero for this resource, so wherever we schedule it, the sum of the squares of this resource remains unchanged. According to Step 3, then, we move it to the right one day, to get as much slack as possible in all preceding activities.

Repeating Step 2, we arrive to activity 6-7. It is currently scheduled from day 1 to day 3, and the current resource requirements are given for each day on top of the bar chart. So the current sum of squares is:

$$\ldots.\ 13^2 + 14^2 + 9^2 + 9^2 + 4^2 + 4^2 + 8^2 + \ldots. = 763$$

If we try to schedule it one day later, (day 2-4), we have for the sum of squares:

$$13^2 + (14 - 5)^2 + 9^2 + 9^2 + (4 + 5)^2 + 4^2 + 8^2 + \ldots. =$$

$$13^2 + 9^2 + 9^2 + 9^2 + 9^2 + 4^2 + 8^2 + \ldots = 713.$$

Note that the term for the first day (13^2) does not change, nor do all the others after the fourth day; instead of computing every time all the terms, let us then take into account only the "end" effects. The above computation would then become:

$$14^2 + 9^2 + 9^2 + 4^2 = 374 \quad \text{(current schedule)}$$

$$9^2 + 9^2 + 9^2 + 9^2 = 324 \quad \text{(new schedule)}$$

So it is profitable to move 6-7 one day to the right (days 2-4), for such move will decrease the sum of squares by 50 (763 -

713 = 374 - 324 = 50). But this may not be the optimum schedule for 6-7 yet; let us try to move it one more day to the right (day 3-5):

$9^2 + 9^2 + 9^2 + 4^2 = 259$ (current schedule)

$4^2 + 9^2 + 9^2 + 9^2 = 259$ (new schedule)

So, according with the rules in Step 3, it is profitable to move one more day to the right. (day 3-5).

Trying to move it one more day (to day 4-6), we have:

$9^2 + 9^2 + 9^2 + 8^2 = 307$ (current schedule)

$4^2 + 9^2 + 9^2 + 13^2 = 347$ (new schedule)

So this last movement is unwelcome, for it would increase the total sum of squares by 40. Trying to schedule 6-7 later still, similarly we discover that the sum keeps on increasing. The optimum schedule for 6-7 is then in days 3-5, and we decreased the sum of squares by 50.

Tab. 9 shows the final schedule obtained by repeating this procedure. We can follow the scheduling changes by noting the changes in each line of the table giving the crew requirements and sum of the squares. Line 1 gives the initial requirements, with all activities scheduled as early as possible, exactly as in Tab. 9.. Line 2 shows the situation after changing 6-7 two days to the right. After scheduling 6-7, we proceed to schedule 3-7 similarly, and so on. With two complete passes through the list of activities, (cycles), the sum of squares does not decrease anymore, and we stop.

A summary of the steps involved in both rescheduling cycles is given Tab. 10.

Total Crew Requirements															Sum of Squares	Line No.	Cycle No.
6	6	7	7												691	8	II
		~~8~~		8											665	7	
			~~8~~	~~8~~		9	9								665	6	
						~~9~~					8				665	5	
							~~9~~				~~8~~				665	4	I
~~9~~		~~8~~													673	3	
	~~9~~	~~8~~		~~9~~	~~9~~										713	2	
~~13~~	~~14~~	~~9~~	~~9~~	~~8~~	~~8~~	~~8~~	~~6~~	~~6~~	6	4	2	2	2	2	763	1	

Duration	Scheduled Start	Scheduled Finish	Earliest Start	Earliest Finish	Slack Total	Slack Free	Crew Requirements	Activity No.
2	2	4	0	2	5	0	3	0-1
4	Ψ	Ψ	2	6	5	0	-	1-2
2	0	2	0	2	CRITICAL		6	0-3
5	Ψ	Ψ	2	7	1	0	-	3-4
1	⌗	⌗	6	7	5	4	4	2-6
4	8	12	7	11	1	0	2	4-5
1	4	5	0	1	6	0	4	0-6
9	2	10	2	10	CRITICAL		4	3-7
3	2	6	1	4	6	6	5	6-7
3	12	15	11	14	1	1	-	5-8
5	10	15	10	15	CRITICAL		2	7-8

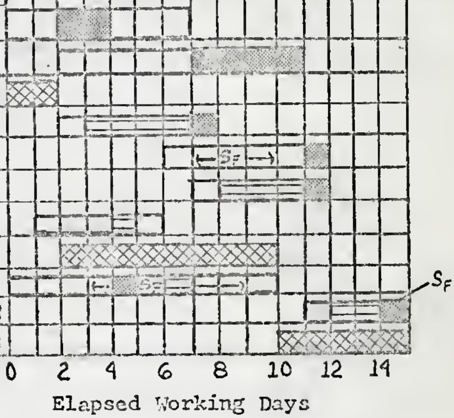


TABLE 9.--Final schedule, computations.

Ψ These activities, which do not require any of the resource under consideration, have some flexibility remaining in their schedules.

⌗ This activity could be scheduled on 10-11, without changing the total sum of squares.

Line in Table	Cycle	Activity Schedule Change	Sum of Squares
1	I	all activities at their earliest times	763
-	I	activity 5-8 from 11-14 to 12-15	-
2	I	activity 6-7 from 1-4 to 3-6	713
3	I	activity 0-6 from 0-1 to 2-3	673
4	I	activity 4-5 from 7-11 to 8-12	665
5	I	activity 2-5 from 6-7 to 11-12	665
-	I	activity 3-4 from 2-7 to 3-8	-
-	I	activity 1-2 from 2-6 to 7-11	-
6	II	activity 6-7 from 3-6 to 5-8	665
7	II	activity 0-6 from 2-3 to 4-5	665
8	II	activity 0-1 from 0-2 to 2-4	641

TABLE 10.--Summary of steps in scheduling the activities in Tab. 9.

The final crew requirements are as follows: 6,6,7,7,8,9,9,9, 6,6,4,8,2,2,2. The final sum of squares is 641, with a total decrease of 763 - 641 = 122.

After we have the final crew requirements, a number of additional adjustments might be made, to take into account factors not considered in the basic scheduling procedure. We could move activity 2-5 back from day 12 to 11 so that the final crew requirements will taper off in a more desirable manner, i.e.,

... 6,6,8,4,2,2,2 instead of ... 6,6,4,8,2,2,2. Note that the total sum of squares is the same.

This is then the Burgess and Killibrew method. It does not necessarily produce optimal results, for it depends on the numbers assigned to the activities on the network, and on the ordering of the activities in Step 1. If we arranged the activities by ascending arrow head numbers, but by decreasing (instead of increasing) arrow tail numbers, we might get a different final result (possibly better). Or, if we had assigned the event numbers in a different way to the network, we might get still other results. Burgess mentions that, if the resource being leveled is critical (very expensive), several different orderings of activities and numbering of events should be tried, and the best final solution found should be adopted.

Also, the method has been demonstrated in leveling only one resource. It can be extended to the leveling of more than one resource, though; but we would have to assign a system of priorities to the resources being leveled and level them one at a time. As an example, if after scheduling for leveling the resource X as in illustration, we wanted to level another resource Y, we would be able to move only activities 1-2, 3-4, and 2-5. The remaining activities are already fixed by the consideration of the most critical resource.

Burgess and Killibrew present also an application of the above procedure to projects which contain groups (cycles) of activities that are repeated a number of times. (21, 221) They give also a computer program to execute the procedure. (21)

3.2.2 Levy, Thompson and Wiest Method.

Levy, Thompson and Wiest (32) presented a method basically similar to the Burgess method, but enlarging it to include the leveling of several types of resources simultaneously, and of several projects.

3.2.2.1 Explanation of the Method.

The problem studied was the leveling of crew requirements in the several shops of a shipyard. Each "project" is then the building of a ship, and each shop, with its specialized crews, corresponds to a different type of resource to be allocated.

The method considers one project (ship) at a time. One simplified project would then be (as an illustration) the sequence of activities portrayed in the network of Fig. 19. The activities and their crew requirements are given in Tab. 11. Note that two shops (types of resources) are included.

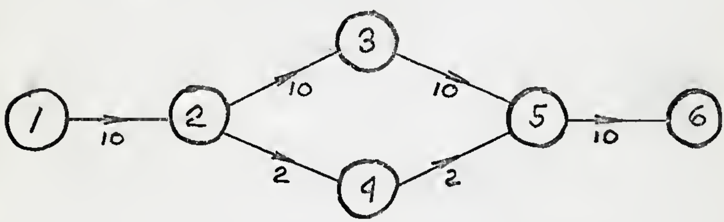


FIGURE 19.--Network with durations.

The basic idea of the method is to schedule first all activities at their earliest start dates, and then shift them to the right for leveling. In this sense, then, the basic idea is similar to the Burgess method.

Activity	Duration	Shop	Crew Requirement
1-2	10	1	10
2-3	10	1	10
3-5	10	2	10
2-4	2	2	10
4-5	2	1	10
5-6	10	1	10

TABLE 11.--Activities and durations for the network of Fig. 19.

There are two distinct parts in the method. The first part consists in trying to level crew requirements in all shops simultaneously. In this sense, then, it tries to level two or more resource profiles at the same time. The second part consists in doing further leveling, but on one resource (shop) at a time, beginning with the most expensive resource (the shop with the highest wages for the crew members).

In the first part, then, all activities of the project are scheduled in their earliest start dates. The manpower requirements for each shop (resource profiles) are then plotted.

In our example, the resource profiles in this step would be as in Fig. 20.

The maximum crew requirements in shop 1 are 20 men, and in shop 2, 10 men.

The next step is to set "trigger levels" one unit below the maximum crew requirements in all shops. In shop 1, then, the "trigger level" would be placed first at 19, and in shop 2, at 9.

The shops and the days where the trigger levels are exceeded are then studied to see if the activities that caused the peak can be rescheduled at some later date. This reallocation can be obviously done for the activities having available slack only. Between those activities, the program chooses one at random and reschedules it at some later date.

The program then recalculates the earliest start times for all activities that are affected by the rescheduling of the chosen activity, and new resource profiles are plotted. Possibly new peaks will develop, either in the same type of resource, or in some other; the program then reschedules another activity, and plots another profile, and so on. This rescheduling continues until all peaks are below the trigger levels in all shops (for all types of resources).

When this feasible schedule is then obtained, the program tries to obtain another one even better, and to do so lowers again the trigger levels in all shops to one unit below the maximum crew requirements in the shop. The rescheduling begins all over again, until another feasible schedule is found with the new trigger levels; then the trigger levels are once again set one unit below the maximum resource requirements and the rescheduling begins again, and so on.

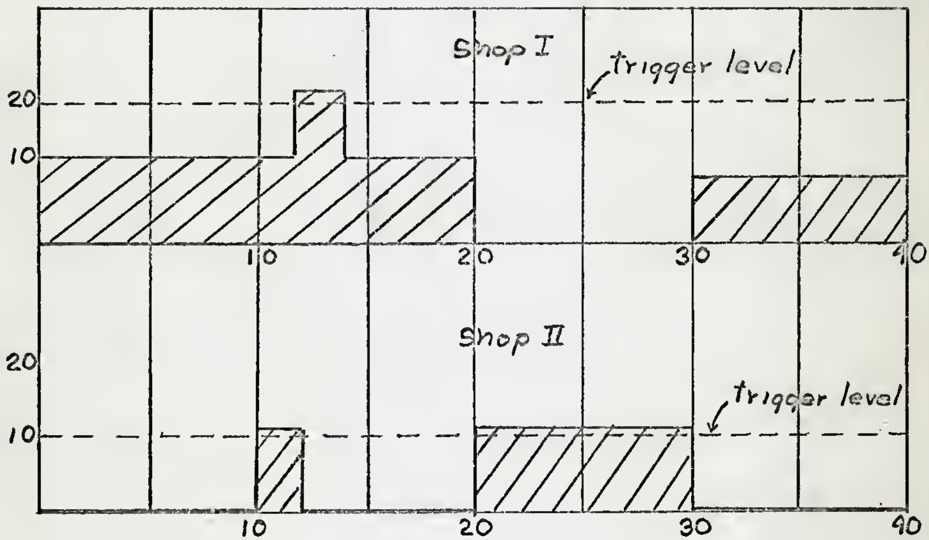


FIGURE 20.--Initial crew requirements for network of Fig. 11.

This process stops when no possible juggling of activities produces a feasible schedule; at least one peak in one shop is always higher than the trigger level; or then no activity contributing to the peak still has any slack available. In any case, the last feasible schedule is then recalled, with the corresponding trigger levels, and the first part of the program ends.

To illustrate this procedure with our simplified example, let us come back to Fig. 20. Let us suppose that the program would choose first the peak on days 12-14 in shop 1 (notice that it could equally well choose the peaks in shop 2). Analyzing the activities contributing to this peak (2-3 and 4-5), 2-3 is on the critical path (1-2-3-5-6) and so does not have slack; the only alternative is then to reschedule 4-5. Let us suppose that we reschedule it for days 23-25. The new resource profile is then shown in Fig. 21.

The peak in shop 1 has then been eliminated in Fig. 21. The maximum resource requirements for any day in shop 1 is now 10 days, well below the trigger level (19).

The program would then recalculate the earliest start times and available slack for all activities depending on 4-5; there is none (for 5-6 is on the critical path and so is already fixed). Proceeding to the next peak, we could try to level the peak on days 10-12, on shop 2, or on days 20 -30, also in shop 2. But there is no possible way in which we can level those peaks so they will be lower than the trigger level. So at this step the program recalls the last feasible schedule (the initial one, shown in Fig. 20 and stops.

Although this particular example is too simple to show the power of the procedure (actually in this case we did have no leveling whatsoever), we hope it served to illustrate how the method would work.

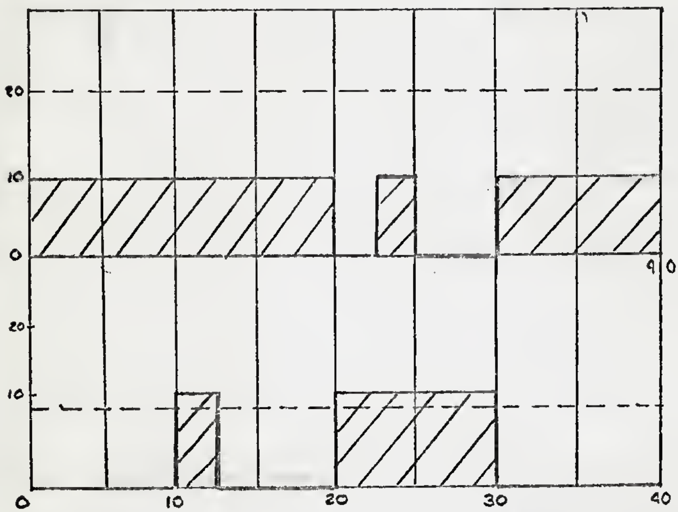


FIGURE 21.--First rescheduling - Resource profile.

There are several interesting details in this procedure. One is the use of a random number generator both to choose between the activities with available slack on a peak, and to choose the specific day when the chosen activity will be rescheduled. The program associates different probabilities to different activigies. More specifically, the probability of choosing an activity in the list is set as being inversely proportional to the manpower requirements in the activity; the probability of scheduling an activity in any specific day wherein the slack is equal for every day. Levy, Thompson and Wiest (32) mention also the possibility of using probabilistic learning techniques in assigning probabilities to the available schedules. All these ideas are obviously refinements, influenced by the work done by Thompson in simulating loading rules when trying to solve the general scheduling problem [see chapter 1.1.5, also (5) and (4)].

After obtaining a feasible schedule (hopefully already half-leveled), the program begins part II. In this part, it uses a similar procedure to level the resource profiles in each shop, (one shop at a time). We begin by the most expensive resource (the shop with the highest wages for the crew). A trigger level is set for the shop, activities are rescheduled, a feasible schedule is found, the starting times and slack for the subsequent activities are recalculated, the resource profile is replotted, a new trigger level is set, and so on. The procedure continues until no further rescheduling eliminates a peak, or no activity in a peak has any slack left. The program goes on then to level the second most expensive resource (shop) in a similar way.

The final output is a plot of the leveled resource requirements. Improvements of 30% to 60% on the maxiumu crew requirements have been reported with this method (32). This is very important, for usually crew sizes are dependent upon maximum manpower requirements: a shipyard does not hire and fire people everyday as the requirements change, but has a stable crew large enough to cover the maximum requirements.

To take into account more than one project, the program adds each new project's resource requirements onto the cumulative totals of the already leveled projects, and proceeds to level this new project in the same way.

3.2.2.2 Evaluation of the Method and Comparison with the Burgess Method.

The adaptation of the Levy, Thompson and Wiest method to the scheduling problem on a machine shop can be easily imagined. The "projects" (ships) would be the products to be manufactured; the "resources" (shops) would be each key machine or machine group; the "resource requirements" would be measured in machine hours rather than in manpower.

The great disadvantage of the adaptation of the method to factory scheduling situations is the basic disadvantage of all leveling methods: in a factory, there are stated constraints on machine hours available per week; the leveling procedures do not see resource requirements as stated constraints, but as an objective function to be minimized (leveled). This basic difference of approach makes any such leveling method, including such a nice one as the Levy method, rather useless for factory scheduling, although it seems to be promising in shipyard scheduling.

Another disadvantage of the method is that, in the same way as the Burgess method, it does not produce an optimal schedule. Notice that, in our illustrative example, if we had begun by trying to level the peaks in shop II before the peak in shop I, the program would have stopped right up there, realizing it was impossible to level the peaks in shop II, and not even trying to level the peak in shop I. Similar occurrences and suboptimizations (in a much larger scale) would happen to larger and more complex projects.

In a way, this method, though much more general and ambitious than the Burgess method, is less efficient than it. At least the Burgess method had the appearence of being systematic; the Levy method, trusting the rescheduling of peak activities to a random number generator, investigates far less possible schedules than the other method; in random rescheduling, for instance, an activity three days after the peak occurred, it fails to investigate the possibilities of scheduling it one and two days after the peak, as the Burgess method would have done. Although it is difficult to make an "a priori" prediction, without experimental evidence from simulation studies, this randomness introduced by the Levi method would seem to make it then less efficient than the Burgess method.

The possibility of being able to level several projects is no great advantage over the Burgess method, since anyway they are leveled one project at a time. The same procedure could easily be followed with the Burgess method.

The possibility of scheduling more than one type of resource at a time in part 1 seems to be more difficult to evaluate. Intuitively, it seems to be an advantage over the Burgess method, as it would take into account interrelationships between different shops (resources); but this advantage seems to be annulled by the one-resource-at-a-time leveling in part II, and this casts considerable doubt on the soundness of the whole idea of dividing the program into two parts. If in the end resources are going to be leveled independently, there does not seem to be any point in beginning by leveling them together.

The final decision, though, on the relative efficiency of the two methods can only be made after careful comparison of the schedules resulting from the two methods for a large number of projects.

Notice, by the way, that the second part of the method is essentially the same as the Burgess method, treating one resource at a time, in order of priorities (cost of resources).

Another interesting point of comparison is the way the objective function is built, that is, what is meant by leveling in the two methods. To the Burgess method, leveling is minimizing the sum of the squares of the resource profile, that is, preventing sharp variations in day to day resource requirements. To the Levy method, it is decreasing, through the use of trigger levels, the "ceiling" or the maximum resource requirements for any day, and not caring about abrupt changes of level from day - to - day.

Probably in a large construction project, employing less skilled labor, as workers can be hired and laid-off according to work requirements, the Burgess approach would be better. In a shipyard, with more skilled labor and larger Union influence, it is mandatory to conserve at all times the same number of workers in each shop and skill; consequently, the Levy approach would probably be better. In a factory situation both approaches would be wrong, as was seen, for failing to take into consideration fixed constraints on available resources.

Another interesting idea used in the Levy method is the distinction it makes between the minimum total project duration, as obtained through the critical path calculation, and the maximum permissible duration. The Levy method does not use the minimum project duration as a constraint, as the Burgess method does; instead, its duration constraint is based on the delivery dates for the project. As an illustration, a product (project) that that could be made in two months, and has to be delivered five months from now would be scheduled by the Burgess method in months 4 and 5 from now; in the Levy method, it would be spread all over the five available months. The Levy method would then be more flexible, and would produce more "leveled" schedules, for it could reschedule even the critical path activities; but it would have to pay for this flexibility by having a much larger in-process inventory.

Finally, it is interesting to notice that Davis (24) in his otherwise excellent survey on resource allocation techniques, seems not to have fully understood the Levy method; he consistently mixes up the concepts of "shop" and "project" rather badly.

3.2.3 The de Witte Method.

De Witte (25) describes a computerized manpower-leveling procedure developed at Hughes Aircraft Company. Like the Burgess routine, it is designed to minimize manpower fluctuation by adjusting the start times of project activities having slack. However, the measure of effectiveness of minimization is absolute magnitude of flucuation from a calculated project mean level of resource usage.

Basically, the method consists of partitioning the resource profile into specially-derived intervals and then sequentially leveling each interval, revising early start times of the following activities where necessary. Output from the computer may be obtained in histogram form.

The problem is split into subproblems, and slack in various activities is systematically reduced until all starting dates are precisely fixed.

The division into subproblems is achieved by finding "critical intervals". These intervals are time intervals in the duration of the project where either the maximum possible load is smaller than the mean level, or the minimum possible load exceeds the mean level.

To find those critical intervals, the program uses an algorithm involving the concepts of upper and lower envelopes of all permissible manpower distributions. The upper envelope consists of the locus of points representing the maximum possible loading of each unit of time within the duration of the project. The lower or irreducible envelope similarly is the locus of points representing the minimum possible manpower loading. The upper and lower envelope calculation is done in subroutines using the values of the earliest starts and latest finish times of each operation.

The critical intervals are found as the intervals where the upper envelope falls below the mean manpower level. In Fig. 22 for example, any of the intervals 1-2, 4-5, 5-6 and 6-7 can be considered as critical intervals.

FIGURE 22.--Histogram: Upper and Lower Envelopes.

As an illustration, in interval 1-2, the lower envelope exceeds the mean level; so there will then always be a "peak" in 1-2, and to level we want to lower this peak the most we can; beginning with any schedule, we try to reschedule all activities out of 1-2, either in the preceding interval (0-1) or in the succeeding interval (2-8), trying to reach the lower envelope in 1-2. In doing thus, we have to recalculate the earliest start times and latest finish times of all sequentially related operations. After this recalculation, we have a new resource profile, with new upper and lower envelopes, and new critical intervals; we again try to "clear" the critical intervals, and so on. In the end all slack will be eliminated, and we will have the final schedule. The program then prints a list of activities with their scheduled start times and the resource profile (histogram).

This is then a brief description of the De Witte method. Although the method is simple in concept, its subroutines are lengthy and intricate, and make it more cumbersome than the Burgess method. Note that this method is also heuristic, and so does not assure us of an optimum (just like the Burgess method).

3.3 RESOURCE CONSTRAINED TECHNIQUES

As was mentioned in 3.1, the second basic type of resource allocation problem is the problem of minimizing project completion times, subject to stated resource constraints. This statement of the problem, as we will see, is more representative of everyday factory scheduling than the leveling approach. Let us study then the major methods that have been proposed to solve this problem.

3.3.1 The Kelley Serial Method.

Kelley (30) proposes a method that is basically an extension of the Burgess leveling technique studied in 3.2.1.

The idea is to try to schedule the project as in the Burgess technique; if the resource constraints are sufficiently high, a feasible schedule might be produced. If they are not, the critical path activities have to be rescheduled, causing an increase in the total project completion time.

The exact steps in the routine would be:

Step 1. List the activities with activity arrow head numbers increasing. In case of ties, list in order of increasing total slack.

Step 2. Check to see if any individual activity requirements exceed the total availability for each resource. If that happens, there is no feasible schedule, and the project has to be replanned (or resource constraints increased).

Step 3. Starting with the first activity in the list and working down the list, schedule each activity as early as possible. In making those schedules, the following rules are followed:

a. The earliest time at which we may consider scheduling the start of an activity is the latest of the finish times of the activities immediately preceeding the one in question. Since the activities are listed in order of precedence, all predecessors of the activities in question will already have been scheduled, and will be found above it in the activity list.

b. Having the earliest start time for an activity, we attempt to schedule it to start at that time. If the required resources are unavailable at that time, the start is delayed to the earliest feasible start time - the earliest time at which resources are available. Of course, when a job is scheduled, the resources available to subsequent jobs in the list are reduced by the amount and type allocated to it.

Note that for each attempt, if resources are not available, all the activities competing for the resource in question and having slack available should be rescheduled within the limits of their total slack, if this will permit scheduling the activity in question.

This is the Kelley serial method. As originally presented by Kelley, it had several refinements designed to increase its flexibility.

One such refinement was the consideration that some activities might not have to be performed continuously, but could be split in several periods. This splitting of activities increased greatly the flexibility of the method; consider for instance the attempt to schedule an activity that would take three days to be completed, when the resources are available only every other day, for the next ten days. If we split the activity to be performed in three different days, it could be completed on the sixth day; if we do not split it, we could only begin it after the tenth day.

Unfortunately, this refinement is useless in machine shop operations, for company policies always state that once a part is loaded into a machine, it will only leave the machine after the operation has been completed. This policy is caused by the high set-up costs in machine shops. In this sense, then, machine shop operations can never be split, and so this refinement has not been considered in the description of the method.

Another refinement is repeating the process for different listings of the activities, and then taking the best feasible schedule found as the solution. The listing used at first (increasing arrow head numbers and, in case of ties, increasing total slack) is only one of the several listings possible, and will not necessarily produce the best schedule. Other possible listings could be to list activities, in case of ties, by increasing arrow tail numbers (as in the Burgess method), or by increasing dollar values of the jobs, duration of the jobs, etc. Although this repeating could lead to a shortest schedule, it is only practical when the resource limitations are quite tight.

Still another refinement is to consider the concept of crew requirement thresholds. If an activity had planned to take three welders and last for three days, that choice of three welders and three days in usually rather arbitrary; if only two welders are available, possibly the foreman would start anyway with only the two welders, the third being added whenever possible. Of course, the duration would be increased.

The method then takes into account this fact by establishing an arbitrary minimum threshold for crew requirements; for example, if a threshold of 80% is accepted and the job requires five riggers but only four are available, we would start the job. If only three riggers were available, we would delay the job.

This refinement seems again useless for our problem (machine shop scheduling). In machine shops, the resource constraint is not men, but machine hours; and the concept of beginning jobs with less men than planned does not have correspondence in machine hours; you cannot begin a job without having all the machine hours (time) available. This fact of the resource being men (as in the example of the riggers)in one case, and time (machine hours) in the other case makes the refinements useless for our scheduling problem being studied, and so this refinement has not been included in the description of the method.

Two further refinements are mentioned in the method, and both seem to be quite useful (for a change) for our scheduling problem. One is the concept of craft requirements thresholds, and the other, the concept of the start delay threshold.

Usually, the resource constraints are not fixed rigidly as described in the method, but are elastic. The number of riggers available might be fifteen, but if projects get constantly delayed because of lack of riggers, it can be increased to seventeen or eighteen. The number of machine hours available per week, with only one shift, is likewise not necessarily fifty hours per week; if necessary, overtime work can be authorized, increasing this availability to, say, sixty hours.

A threshold of 20% can then be included on the method; the jobs scheduled taking advantage of this extra availability are indexed, for it may be possible to replan these jobs later so an overload will not occur.

The second concept, the concept of a start delay threshold, is useful if we consider the fact that long jobs, requiring a large amount of resources, can be delayed indefinitely by the method. A threshold N, which tells how much we are willing to delay the start of a job is then introduced. If we find that the start will be delayed more than N days, we schedule it for its earliest start time, regardless of how resource availabilities are exceeded. The jobs that violated the start delay threshold are indexed for further reference; the results of using such a device can provide necessary argument for obtaining more resources (namely, increasing the overtime threshold or beginning work in a second shift).

Regardless of the refinements, we can see that basically Kelley's serial method is but an extension of the Burgess method to the resource constraint problem. The chief difference is not calculating the sum of squares as in leveling. The method has then the same disadvantages of the Burgess method, chiefly not necessarily producing an optimal schedule. Furthermore, although Moder and Phillips (221) report satisfactory results on its use, no attempt to use it in machine shop operations, without the flexibility of the extra refinements, has yet been reported; it is possible that the schedules thus produced will be unsatisfactory.

3.3.2 Parallel Methods: the Brooks Method.

Kelley (30) makes a distinction between parallel and serial methods. Serial methods would consist of a listing of the activities, and of working down the list scheduling each activity. Parallel methods would consist of defining, at each time t, a subset Q(t) of the set P(t) of activities that can be scheduled at time t. The subset Q(t) is then scheduled, until resource constraints become active or some job is completed; then P(t) and Q(t) are redefined, and the process continues.

The Brooks method is an example of a parallel method. It is described by Moder and Phillips (221) and attributed to G. H. Brooks of Purdue University. It requires just one pass through the list of project activities. The method is stated below in seven steps.

Step 1. Arrange the activities in a linear array such that the maximum remaining path length (MRPL) is decreasing in magnitude. MRPL (activity x-y) = T_e(terminal event) - T_e(event x) - S (activity x-y). Also indicate, for each activity, its duration time and its resource requirements, as shown in Tab. 13.

Step 2. Check the resource requirement of each individual activity to see that none of them exceeds the maximum availability of the resource in question.

Step 3. Establish the first "decision set," defined in general to be the set of all unscheduled activities whose predecessor events have all occurred (in time). The first decision set will consist of all activities "bursting" from the initial network event. As activities enter the decision set, record the current value of "time now," or T, in the row, Time of Entering Decision Set; for the initial set this time will be zero.

Step 4. Initially set T equal to zero, and set the total availability of the constraining resource, C, equal to the specified maximum availability.

Step 5. In general, choose from the current decision set the activity with the largest MRPL which, because of Step 1, will be on the left hand side of the decision set; in case of ties, choose any one. Now subtract the resource requirements for this activity from C, and call the remainder C'. If $C \geq 0$, enter T for the activity in question in the row titled, Activity Scheduled Start Time; enter T plus the activity's duration time in a row titled, Activity Scheduled Finish Time, and delete this activity from the decision set. If $C' \geq 0$ add the resource requirement for this activity back to C', and repeat Step 5 for the activity with the next larger MRLP. When the current decision set has been completely examined, go to Step 6.

Step 6. If T is equal to the largest number in row titled, Activity Scheduled Finish Time, terminate the algorithm; note, T in this case gives the project duration time. If T is not equal to the largest number in the row titled, Activity Scheduled Finish Time, increase T to the next largest number in this row.

Step 7. For each activity scheduled to finish at the new T, determine all successor activities that now have all of their predecessor events completed, and add these activities to those remaining in the current decision set. Also, for each activity scheduled to finish at the new T, add its released resource requirements to C'.

The above algorithm has been applied by Moder and Phillips (221) to the network problem shown in Fig.18. The results given in Table 12 were obtained. The detailed application of the algorithm for the case of a maximum crew availability of seven is given below (only partially).

Maximum Number of Crews Available	Duration of Total Project (days)
9 or more	15
8	18
7	19
6	21
less than 6	not feasible.

TABLE 12.--Results from Moder and Phillips.

Step 1. From Figures 18 and 8 the required linear array is constructed. The first four heading lines of Tab. 13 are covered by this step.

Activity number	03	37	34	02	06	12	67	45	78	25	58
Duration time, t	2	8	5	2	1	4	3	4	5	1	3
MRPL	15	13	12	10	9	8	8	7	5	4	3
Crews required	6	4	0	3	4	0	5	2	2	4	0
Time of entering decision set	0	2	2	0	0	4	11	7	14	8	15
Activity scheduled start time	0	2	2	2	10	4	11	7	14	14	15
Activity scheduled finish time	2	10	7	4	11	8	14	11	19	15	18
End of cycle number	0	1	12	3	44	5	6	7	8	9	10
Resources available, C	$\cancel{7}\cancel{1}$	$\cancel{7}\cancel{3}\cancel{0}$	3	$\cancel{3}\cancel{1}$	$\cancel{1}$	$\cancel{5}\cancel{1}$	$\cancel{7}\cancel{2}$	$\cancel{7}\cancel{5}\cancel{1}$	5	$\cancel{5}$	7
Time now, T	0	2	4	7	8	10	11	14	15	18	19
Members of decision set	03*	37	06	06	06	06	67	78	58	—	
	01	34	12	45	25	25	25	25			
	06	01									
		06									

TABLE 13.--Application of Brooks' Algorithm to problem given in Fig. 18.

(*) Circle denotes those activities that are scheduled during following cycle, e.g., activity 03 was scheduled during the first cycle and was circled in the preceding or 0 cycle column.

Step 2. Resource availability is feasible, since no single activity requires more than seven crews.

Step 3. The first decision set is established, as denoted by entry "0" in the row titled, Time of Entering Decision Set. These activities, i.e., 03, 01 and 06, are also listed at the bottom of Tab. 13 under the End of Cycle Number 0 column.

Step 4. Set $T = 0$ and $C = 7$.

<u>Start Cycle 1</u>

Step 5. The activity in the decision set with the largest MRPL = 15 is activity 03. The number of crew required is 6, so $C' = 7 - 6 = 1$, and $1 > 0$. Time now, $T = 0$, is entered in row, Activity Scheduled Start Time, for activity 03, and its scheduled finish time of $T + t = 0 + 2 = 2$, is entered in the following row. The activity in the decision set with the next MRPL (10 days) is 01. $C' = 1 - 3 = -2 \angle 0$. Activity 01 cannot therefore be scheduled. C is restored, $C = -2 + 3 = 1$. The Activity in the decision set with the next largest MRPL is 06 (9 days). $C' = 1-4 = -3 \angle 0$. Activity 06 cannot be scheduled. C is again restored, $C = -3 + 4 = 1$. The decision set for $T = 0$ has now been completely examined.

Step 6. $T = 0$ is not equal to the largest number in the row Activity Scheduled Finish Time; hence, set $T = 2$, the next largest number in this row.

Step 7. At T = 2, there is one activity in the array with entry equal to T, that is, activity 03. Its resource requirement is 6; thus, the new value of C = 1 + 6 = 7. Activities 37 and 34 can now enter the decision set, as all predecessor events have been completed for these activities. These activities, along with activities 01 and 06, form the decision set for the next cycle.

<u>End Cycle 1</u>

Step 5. The events 37, 34, 01, and 06 are considered in turn, in the same way as in the previous application of Step 5. This time, 37, 34, and 01 can be scheduled, and removed from the decision set. The resulting final C will equal 0. Note that 34 required none of the constraining resource and could be scheduled immediately.

Step 6. T is not equal to the largest number in the Activity Scheduled Finish Time row; hence, it is increased to the next larger number, 4.

Step 7. At T = 4, activity 01 is completed. Its resource requirements, 3, are added to the current value of C to get the new C = 0 + 3 = 3. Activity 12 can now enter the new decision set.

<u>End Cycle 2</u>

Step 5. Activities 06 and 12 are considered in turn. 06 and cannot be scheduled since C' = 3-4 = -1. C is restored

to 3. Activity 12 can be scheduled and removed from the decision set, since $C' = 3 - 0 = 3$.

Step 6. New $T = 7$.

Step 7. At $T = 7$, activity 34 is complete. C is unchanged, as 34 did not use the constraining resource. Activity 45 enters the decision set.

And so on. A complete working of this example can be found in (221), Appendix 6-1.

This method will give the best results obtainable in the first pass, and in some cases will produce a shorter duration schedule than the Kelley routines. For the case of maximum crew availability of 6 and 7, for example, the Brooks method gives project durations one day less than the Kelley method.

3.3.3 DART (Daily Automatic Rescheduling Technique).

DART is an integrated scheduling, dispatching and control system used operationally by the Directorate of Maintenance of the Air Force Logistics Command at Kelly Air Force Base, Texas. This Directorate is engaged in the repair and modification of Air Force equipment, chiefly the B-52 aircraft. In 1965, ninety aircraft were modified, with expenditure of 2,600,000 man-hours. Each B-52 can need 18,000 different operations, of which 9,500 are done on each individual aircraft, consuming 60,000 man-hours.

The system, as described by Marchbanks (33) schedules daily the operations to be done on the next day, trying to "minimize the in-work flow time (project duration) of the aircraft and maximize the utilization of time-consumed production resources within flow-time constraints" (33) In fact, the system does not maximize utilization of resources, but follows the pattern of the resource constrained scheduling problems; it tries to minimize duration, subject to stated resource constraints. Basically, the scheduling process used is a variation of the Kelley method; in this sense, then, only an approximation to the optimum schedule is found (sec 3.3.1).

DART is a fascinating integrated system. It is a pity that we have to restrict ourselves only to its scheduling phase, leaving aside its planning, dispatching and control characteristics. The resource constraints considered are labor (in several different skills) and work area. The labor availabilities are calculated daily by a section of the program, taking into account employee vacations, inter-work center transfers, etc. Work area refers to the area in the aircraft that has to be worked on; it is a constraint in the sense that not more than a certain number of people can work simultaneously inside the pilot cockpit, or on a wing; there would not be physical space for more workers in cockpit or on the wing.

Equipment or tools are not considered as constraints. They are left to what is called "supportability planning by Maintenance managers."

The objective function is the project duration of all aircraft. A priority is assigned to aircrafts, depending on its completion progress; different aircrafts are assigned a "daily scheduling factor" which is the quotient of hours remaining to project completion date by number of hours required to complete the remaining work. This factor, for example, gives an aircraft that is one day behind schedule, and has only ten days left to its project completion date, a greater priority than an aircraft that is one day behind schedule and has twenty days left to its project completion date. Each project duration is tentatively minimized then, in order of daily scheduling factor.

The scheduling process begins with the daily updating of each network, taking into account operations completed during that day. The critical path through each network is then computed, from the initial operations (operations with all predecessors completed) to the final event in the network. Each operation in each project is assigned thus an earliest and a latest start date.

All operations on each project (aircraft) are then sorted and listed by latest start times, and separated into three classes:

Class one, latest start times smaller than sixteen hours; these operations should then be scheduled to start (if possible) in the next day, within the two eight-hour shifts.

Class two, operations with latest start times greater than 16 and smaller than 32 hours; these operations should be scheduled until the end of the second day.

Class three, operations with latest start time greater than 32 hours.

All aircraft are processed in order of their priority. An attempt is made to assign aircraft area and available skills to all Class one operations on each aircraft first, in order of priority (aircrafts with the most critical daily scheduling factors are considered first). All aircraft are then processed again in the same order of priority and an attempt is made to assign area and skills to Class two operations. The aircraft are processed a third time for Class three operations.

The actual scheduling method used is the Kelley method; Each operation is tentatively scheduled for its earliest start time. If resources are not available, its start is postponed until resources are available. Any operation that cannot be scheduled on a particualr shift, due to area or skills not available, is moved to the next shift and an attempt is again made to schedule the operation.

The chief output of the system is the daily schedule, a listing of all required operations that are scheduled today and forecasted to be scheduled tomorrow. Operation cards and several reports for management are by-products of the system.

The main difference between the DART and the Kelley techniques is the listing of the operations by increasing order of latest start time within each project, instead of by increasing arrow head numbers. This feature of DART seems to be an advantage over the basic Kelley method, as it causes scheduling of the most critical, that is, with less total slack activities first. It is more complicated, though, for it involves periodic recalculation of the earliest start times of all successor operations in the network when each operation is scheduled.

Although the system has not been operational long several benefits already are said to have been achieved. The method is claimed to have increased production effectiveness and decreased the time required to modify each aircraft, without increasing the cost of overhead support; no figures are reported to substantiate these qualitative claims. Also no mention is made of the method DART substituted, so no basis of comparison can be made (it does not tell who DART is better than).

The chief importance of DART is having established that resource allocation methods based on critical path techniques are feasible and profitable in day-to-day scheduling operations; it can be easily adapted to machine-shop conditions by simple substitution of the constraining resources (critical machines, instead of work area and labor skills).

3.3.4 Proprietary Programs: RAMPS and RSPM.

3.3.4.1 Introduction.

Two sophisticated comprehensive multi-project scheduling programs are also available today. They are RAMPS (31, 38, 213, 223 and 225) and RSPM (225). RAMPS stands for Resource Allocation and Multi-project Scheduling, and was developed by Du Pont and CEIR, Inc. RSPM stands for Resource Planning and Scheduling Method, and was developed by Mauchly Associates. Both programs are proprietary, but networks can be processed on a service center basis.

The objectives of both programs are:

1. Meet project completion dates, or minimize overruns.
2. Respect resource availabilities.
3. Minimize idle resources.

O'Brien (225) mentions that the Automotive Safety Foundation in Washington, D. C. conducted a comparative test of the two methods and obtained essentially the same results for the sample network computed. RAMPS can handle up to 700 activities and is run on an IBM 7090; RSPM can handle up to 1600 activities and is run on an IBM 1620.

The computational algorithms utilized in both programs have not been published. It is safe to assume, though, that both programs are heuristic and so do not necessarily produce an optimal schedule, and that both algorithms are based on variations of the Kelley or Brooks procedure. We shall study more closely the RAMPS program.

3.3.4.2 Description of RAMPS.

The RAMPS system is based on conventional network logic for each project, together with input information pertaining to each project activity, to each over-all project, to the common pool of available resources, and finally to the over-all scheduling objectives. First, for each activity, three sets of input data are included as follows: one for normal time operation, one for speedup, and one for slowdown. Each of these three sets of data include the resources required to perform the activity (the resource utilization efficiency may be different for each set of data), the corresponding activity performance time, and the cost of interrupting (splitting) an activity once it has begun. At the project level, the input information includes the start date, desired completion date, and dollar-penalty rate for delay of completion, or, as an alternative, a project priority rating. With regards to resource availabilities, the input information must give, for each time period, the normal costs and available number of units and their cost, which may be made available through overtime and subcontracting. Finally, scheduling objectives must be stated in terms of relative importance (weights) of minimizing idle resources, meeting project completion dates, avoidance of activity interruption (splitting), maximizing the number of activities scheduled concurrently, etc. After the basic scheduling computations are made using all normal times, this program progresses through the network, the activities being time scheduled as long as resources are available. If the available resources are not sufficient, the various feasible combinations of allocations are evaluated, and the best combination is chosen. The rules used in this choice reflect the relative weights given to the various scheduling objectives. There are two main outputs of this program; one gives the individual activity costs and resources, summarized by projects, and the other gives the resources used by type and by time period summarized over all of the projects. A study of these outputs usually suggests certain changes in the inputs that will bring the former more in line with desired objectives, whatever they may be. A few such computer runs will, in most cases, lead to an acceptable master schedule, which is updated periodically to accommodate changes in current plans, cancellation and completion of current projects, and the introduction of new projects.

3.3.4.3 Analysis of RAMPS

As was previously mentioned, details of the RAMPS computational algorithm have not been published. It is possible, though to imagine modifications of the Kelley algorithm, for instance, that would produce essentially the same results as RAMPS. It is possible then that RAMPS (and RPSM) use similar ideas.

RAMPS could use the same basic idea of the Kelley method, namely, with the projects listed by priority, schedule each one (in order of priority) at the earliest start time of its activities;

when conflicts develop, alternatives are analyzed and a solution is chosen. Or then it could use the parallel approach of the Brooks method, listing all activities (of all projects) whose predecessor events have occurred, and then scheduling them systematically at the earliest possible time, according to some ordering, until conflicts develop.

The most interesting and novel of RAMPS characteristics is the choice of management objectives, or the possibility of the use of control factors to assign different weights to different objectives. The six control factors, mentioned in RAMPS are:

1. Total Float.
2. Free Float.
3. Look Ahead.
4. Work Continuity.
5. Number of Jobs.
6. Idle Resources.

Let us study how the programmer can, by assigning different weights to these objectives, make the final schedule optimal in relation to different objectives.

1. Total Float.

The Total Float factor is intended primarily to place special emphasis on scheduling those activities with little or no total float. The smaller the activity total float, then, the highest its priority. This priority tends to <u>minimize project</u>

completion time, for activities on the critical path have the smallest total float (zero, if the due date is the same as the terminal event).

Note that this objective (minimization of project completion time) is not always what management strives for. In fact, if a project takes three months to be completed, but only has to be finished five months from now (its due date is five months from now), it is useless to try to minimize project completion times, completing the project three months from now. Only in certain types of contracts, where there might be heavy penalties for delays or incentives for early delivery, is this objective important; in most usual cases, as long as the due date is respected, management should not worry very much about minimizing completion times.

2. Free Float.

Assigning priorities to activities in inverse order of their free floats has basically the same objective of the Total Float: to minimize project completion times. Free Float, as defined in 2.3.3, is the number of days an activity can be delayed without delaying any other successor activity; total float was defined as the number of days an activity can be delayed without delaying project completion. Consider an activity with large total float but with no free float; if we delay it within its total float, we have to delay the successor activities; but the total float of these successor activities will decrease, and they might cause bottlenecks later on when they are going to be scheduled. By assigning priorities to activities by inverse order of free float we are then preventing future bottlenecks and thus helping minimize project completion times.

Both the total float and the free float of an activity are then indicative of the "criticality" of the activity. If the minimization of project completion time is of paramount importance, activities should be scheduled _in the order of their criticality_. This can then be easily accomplished, in the Kelley method, by sorting activities by their total float and, in case of ties, by free float, instead of by arrow head numbers. This sorting would automatically condition the program to minimize completion times of all projects.

3. Look Ahead.

The look ahead feature of RAMPS is intended as a guard to avoid bottlenecks, or scheduling conflicts. It consists in the assigning of priority to those activities upon whose completion many activities are waiting. Consider two activities, A and B, with the same total and free float, but A with 10 successor activities, and B with 2 successor activities. If we want to prevent future bottlenecks, we had better schedule A first; there are ten possibilities of a bottleneck if we delay it, and only two if we delay B. So activity A should have a higher priority than B.

To assign priorities this way, it is enough to use the same trick used in the Brooks method, that is, listing activities in the order of their MRPL (Maximum Remaining Path Length). Going systematically down the list, we are automatically scheduling first activities whose delay could cause bottlenecks.

Note that this objective (avoiding bottlenecks) can be more crucial than minimizing project completion times, in the case of projects with delay penalties but not premium on early delivery. What management wants, in this case, is to deliver all projects in time; it does not want early completion of all projects, for it would bring no advantage; it prefers avoidance of bottlenecks, for this would minimize delays in project completion.

4. Work Continuity.

RAMPS allows for the possibility that some activities can be interrupted without extra costs, and some others cannot. The Work Continuity factor expresses how much we want to avoid activity interruptions. If we give a zero weight to this factor, all activities will be vulnerable to interruption; if we give a high weight to this factor, and associate "interruption penalties" to each activity, activities with low penalties will be more vulnerable to interruption than the ones with high penalties.

The Work Continuity factor could be introduced in a Kelley method. Step 3 of the Kelley method (see 3.3.1) gives the rules to be followed once conflicts develop (that is, when resource constraints make it impossible to schedule one activity within its total float). The original Kelley method had some more rules, in Step 3, to try to split activities, to slow them down, and to speed them up (223). These rules are arranged sequentially in the following way:

Step 3.

1. $m = m + 1$.

2. Try to schedule activity m within its total float. If you can, go to 1;

3. Try to reschedule some other conflicting activities within their total float. If this makes it possible to schedule activity m, schedule it and go to 1.

4. Try to slow down activity m. If this solves the conflict, go to 1.

5. Try to hurry it up.

6. Try to schedule the activity by splitting it. If this solves the conflict, go to 1.

7. Try to slow down, hurry up and split all other conflicting activities. If this makes it possible to schedule activity m, schedule it and go to 1; if not, schedule it at the first possible spot even outside of its total float; the project will have to be delayed. Go to 1.

We can see then that all possibilities are tried sequentially; if one fails, the next possibility is tried, until all possibilities have failed; then, the activity is scheduled anyhow, even if outside of its total float.

One way to introduce the Work Continuity factor in the program is to <u>change the order in which the rules are to be consulted according to the weight assigned to the factor</u>.

To explain this idea, let us call K_1 the weight assigned to the Work Continuity factor, and K_2 the interruption penalty of activity m. Note that $K_1 \leq 1$. Multiply $K_1 \times K_2 = K_3$. Now the order in which the list is going to be consulted can <u>change according to K_3</u>.

If K_3 is zero, or smaller than Q_1 (and so the activity can be split), from 3 go to 6, and then to 4. That means we try to split activities before trying to slow them down or hurry them up. Similarly, in 7, try first to split all other activities.

If $Q_1 \leq K_3 \leq Q_2$, from 4 go to 6, and then to 5. That means we try to slow down the activity first, then to split it, and then to hurry it up. Do the same thing on 7.

If $K_3 \geq Q_3$, skip 6 entirely. That means that we are not going to try to split activity m at all, as it is too expensive to do so. Note that this does not mean that splitting conflicting activities should not be tried in 7.

According to this scheme, then, we are changing the order in which the rules are to be consulted according to the weight assigned to the Work Continuity factor. This change in the order will change the frequency with which the possibility of splitting is tried. If splitting is tried first, then in the long run a large percentage of activities will be split. If splitting is tried only as a last resort, this percentage will decrease, for all activities that could be scheduled either by splitting, hurrying up and slowing down are not going to be split. If splitting is not tried at all, activities with $K_3 \leq Q_3$ will never be split (that is, their percentage of splitting is zero).

This is then one possible way in which the Work Continuity factor can be introduced in the Kelley method to do the same tricks RAMPS is said to do. Several variations of this idea are possible; it can be equally used in the Brooks method.

Unfortunately, for the specific application we are thinking of, that is, machine shop scheduling, this characteristic of RAMPS is useless: activities (operations) are almost never split in machine shops, due to the high set-up costs. So all this nice scheme is useless in our particular application, although it is certainly useful in a more general case.

Note also that "minimizing splitting of activities" is the objective to be attained by the Work Continuity factor.

5. Number of Jobs factor.

Sometimes one of management's objectives is to maximize the number of jobs being worked on simultaneously. This objective has two consequences:

1. It tends to minimize idle resources, for it increases

the manufacturing span of all projects and so gives the system flexibility.

2. It tends to decrease possibilities of delays in project completion, because of the added flexibility.

There are two ways to introduce this factor in the Kelley algorithm. They are:

1. List activities according to activity times, instead of according to total float; or then list them by total float (or some other ordering) and, in case of ties, by activity times. The effect of this listing is the same as the effect of using the SIO (Shortest Imminent Operation) loading rule in simulating job shop scheduling (see 4, 10; also, 1.1.6).

2. Try to slow down all operations; this can be done in a scheme similar to the one mentioned for the Work Continuity factor; the frequency of trying the slow down possibility can be changed by changing the order in which the list is consulted in Step 3 of the Kelley method. When we slow down all operations, obviously more operations will be allowed to be scheduled at the same time.

Note that this management objective (maximization of number of jobs being worked on) is quite opposite to minimization of completion times of all projects. To minimize completion times, all jobs should have the smallest possible manufacturing span, and so only a few jobs can be worked on at the same time. The effect of maximization of number of jobs is an increase in the manufacturing span of all projects, with consequent increase in work-in-process inventories and related cost.

6. Idle Resources factor.

This factor gives weight to the minimization of idle resources as a management objective. This objective is also opposite to minimizing project completion times.

The factor can be included in the Kelley method either by trying one of the leveling techniques mentioned in 3.2, each time you schedule each activity, or then by trying to hurry up all activities. This second solution could be done using a scheme similar to the ones suggested for the Work Continuity and Number of Jobs factors.

These are then some ways in which some of RAMPS characteristics can be achieved with modifications of the Kelley method. Note that we are not claiming that these are the ways used by the RAMPS program; they are simply ways in which RAMPS' tricks could be done.

These uses of alternative management objectives are the most glamorous of RAMPS characteristics. Actually, several of these extra-tricks are quite useless; the splitting possibility, as was said, is useless in pure job shop scheduling; (although it might be useful in batch production scheduling). The slow down and hurry up possibilities are equally useless, for you cannot increase or decrease the speed of machining operations. In a welding shop, if you allocate more (or less) resources (welders) to a job, you can speed up or slow down its duration;

in a machining operation, the pace is set by the machine, and the concept of slow down or hurry up does not apply; machines are long range capital investments and its purchase is usually outside the planner's authority.

With the elimination of these characteristics, we can see that RAMPS is nothing more than an extension of the Kelley or Brooks method to multi-project scheduling. The basic difference is that, instead of trying to schedule one project at a time, RAMPS collects all activities for all projects, sorts the activities of all projects by some criteria, and goes down the list. In this sense, then, RAMPS (the mysterious undisclosed proprietary program) is just as good for machine shop scheduling as the DART program that has been mentioned in 3.3.3 (and published).

Obviously, the great advantage of RAMPS is to have been the first program to face multi-project scheduling as a general problem that could be optimized. Note, though, that it is only a sub-optimization, in the same sense that the use of loading rules by the foreman (4, 10) produces an "approximate" solution. But RAMPS was the first program to prove that large scale computer scheduling was feasible and advantageous in manufacturing operations.

3.3.5 <u>The McGee and Markarian Method</u>.

Another multi-project approach is described by McGee and Markarian (34). Their model is based on a time/cost tradeoff formulation of CPM type described in 2.5. Two sets of cost-time input data are required for each activity - a "minimum essential effort" and "crash effort" manpower (cost), each with associate time estimates. A linear function is assumed to exist over the interval between these two points.

All projects are scheduled initially using the "minimum essential" resource allocations. If, for this schedule, manpower constraints are exceeded for any time interval, slack activities are rescheduled in an attempt to stay within the constraints. If this causes project completion dates to be delayed, then manpower increments are allocated to the minimum-cost activities on the critical path of the project that is most late. This incremental allowance is done successively until all due dates are met or manpower is fully allocated.

Although the basic idea is a nice one, again this method is of no earthly use in machine shop scheduling, for the same reason that all time/cost tradeoff methods are useless: the concept of hurrying up activities by allocating more men (resources) in a "crash" effort has no meaning for machining operations. Of course we could buy more machines, if necessary; but this is done only on the long run, never in day-to-day scheduling procedures, as it is with men and other resources.

A useful point in the method is its assigning of priorities to activities and projects through the concept of the most overdue project. S=(project completion - desired date) is computed. Then the project with the maximum value of S is determined, and the first manpower allocation made to this project. This idea is quite useful, but it has been superseded by a better idea along the same lines, the one used by DART (see 3.3.3), of listing activities by latest start times.

3.3.6. Assembly Line Balancing Methods.

A completely different approach to the scheduling problem under resource constraints has been the one used by Wilson (42) and Moodie and Mandeville (37). They showed that the problem is analogous to the general Assembly Line Balancing problem and so all the algorithms and methods used to solve the Assembly Line Balancing problem can be used to solve the "dual" of the scheduling problem.

Moodie and Mandeville (37) present a very neat solution to the dual of a scheduling problem using Bowman's integer programming solution to the Assembly Line Balance problem. They began with a network with 7 activities and transformed it in an integer programming problem with 5 sets of constraints and 103 variables, and solved it. The conclusion is obviously that this is an academic rather than an economical procedure for solving the resource balancing problem. The application of this procedure might not be feasible, even if more efficient integer programming algorithms were available, because the task of writing the objective function and constraint equations is in itself a formidable one.

Moodie and Mandeville (37) show still that their exact approach can be extended to multi-project scheduling, although the problem becomes still more enormous. They then show that the use of heuristic approaches to Assembly Line Balancing can be equally used to scheduling problems; they mention the results obtained by the use of one such heuristic method, the Moodie and Young method, on a small sized problem.

Although this Assembly Line Balancing approach opens some future possibilities for the discovery of exact methods to solve the scheduling problem, for the present it is impractical, and so only of conceptual value; as to the heuristic methods, once we are not reaching an exact optimum anyway, there does not seem to be any advantage in going in a roundabout way when you can go on a straight way; the heuristic solutions to the Assembly Line Balance problem are not at all so much simpler than heuristic methods to solve the resource scheduling problem that they would be worth the extra work (and computing time).

3.4 EVALUATION OF THE LATEST TECHNIQUES

Reviewing all techniques that have been published for the resource allocation problem, it can be seen that none of them present an exact solution; all of them are heuristic, giving only good approximate solutions. No breakthrough has then been made yet as to an exact mathematical solutions. But several techniques give good workable solutions, and are then quite satisfactory.

We can realize why no exact solution has yet been found when we look back at the optimization problem. Four different and contrary objectives have to be reached. Ideally, the objective function could be expressed in terms of costs, and optimized as a (integer) linear programming problem; but this approach does not seem promising, because of the large number of constraints and variables necessary, and of the difficulty of obtaining all of the cost coefficients for all the factors. Moodie and Mandeville's (36) approach has been along these lines, and his results indicate how enormous the size of such an integer programming solution would be.

The various leveling techniques have a grave setback, as was seen: they are quite efficient (some more than others) at solving the problems they propose to solve; only the problem they propose to solve is not the problem we want solved, that is, the machine scheduling problem. As they do not treat resources as constraints (but variables to be minimized), their results might not be feasible most of the time. A scheduling system might possibly be designed based on leveling techniques, but it would have to include several trial-and-error procedures; they have then a marked disadvantage as compared to resource-constrained techniques.

As to resource constrained techniques, DART, RAMPS and RSPM are tested, established techniques; they use multi-project, multi-resource approaches, and are ready to be used with modifications. Management can then decide whether he wants to buy the system (RAMPS and RSPM) or design a system of its own, based on the DART, Kelley or Brooks methods.

We see then that the resource-constrained methods use a mix of basic critical path ideas (that is, trying to schedule activities within their slack) with loading rules. Loading rules can be used either in the initial ordering of all jobs in all projects, or in choosing between a set of possibilities (as in the Brooks method), or when conflicts develop. The following loading rules (decision rules) can be used:

1. Latest start times
2. Arrow head numbers
3. Arrow tail numbers
4. Total slack
5. Free slack
6. Maximum remaining path length
7. Shortest imminent operation.

Juggling with these loading rules under the basic Kelley (serial) or Brooks (parallel) algorithm, a good multi-project, multi-resource scheduling system can be designed according to the management objectives desired. Flexibility can be added, if necessary, by use of some of Kelley's or RAMPS' mechanisms.

Summing up, then, it is possible to design your own critical path-based scheduling system that will produce feasible solutions.

CHAPTER IV

CONCLUSIONS

4.1 APPLICATION OF CRITICAL PATH TECHNIQUES TO PRODUCTION PLANNING

Up to now, we have seen two basic types of solutions to the scheduling problem. One is the traditional way: it is based on Gantt charts and clerical posting. The other is critical path-based techniques. Before a final comparison is made, let us investigate other possible solutions.

One solution that has been used is to plan with pure critical path techniques for important projects only. Sometimes, of the fifty projects a machine shop is working on simultaneously, two or three loom out as being much more important than the others, either by their high manufacturing costs or by their steep delay penalties. PERT or CPM is then applied to those three projects, regardless of resource constraints. Usually, anyway, the constraints would not be reached at any time period with only three projects in fifty; and, if they are, it is feasible to replan the three projects so as to respect the constraints. These projects are then minimized towards manufacturing span and project delays. The next step is then to schedule all other projects through some variation of the traditional method, thus trying to level resource requirements, that is, minimize idle machine times. All scheduling objectives can (see 1.2) thus be at least partially reached: idle resources are "minimized" (after the three important projects have been scheduled) by scheduling the majority of projects under the traditional way: as this traditional method is very good as to minimizing idle resources, objective 1 is at least partially reached. Objective 2 is not so well reached, unless the three projects are so large in machine hours as to dominate all others; but objective 3 is reached fairly well, for the projects with most steep penalties are indeed planned separately, through PERT and CPM. As to systemic costs, they are in between the costs for critical path-based techniques and those for the traditional methods.

Another possible solution, largely used today in machine shops, is to computerize the traditional method. This computer approach is almost as good as the traditional method as to idle resources, although it has a little less flexibility if the delay-unit is decreased and may decrease systemic costs. It may also show large improvements as to objective 2, minimizing in-process inventories, for it will decrease substantially the average manufacturing span. This is so because memory and speed capabilities of the computer are greater than the human's, and so the machine does not have to rely so much on preventive slack as does the human planner; also, these capabilities make it possible for it to decrease the delay unit from one week or three days, to one day or even one shift; most of the waste in manufacturing spans typical of the traditional method is thus eliminated. This method is largely used today; there are several variations, either using critical path ideas, or numbering systems, or loading rules.

Having briefly reviewed those two existing methods, let us proceed to compare all four basic systems through grading them against the four objectives. Note that those basic methods are not the only possible approaches; other different ideas are also used, as are mixtures of the four; but they seem to be the most largely used.

		Objective				
No.	Type	1	2	3	4	G.P.A.
I	Traditional Method	A	D	C	C	2.25
II	Computers + Traditional Method	B	B	D	B	2.50
III	Mixture of Pure CPM/PERT with Traditional Procedures	B	C	B	C	2.50
IV	Resource Allocation Procedures (Based on Critical Path Techniques)	B	A	A	B	3.50

TABLE 14.--Comparison of several methods of scheduling under four scheduling objectives. A = 4, B = 3, C = 2, D = 1.

As to objective 1, the traditional method (1) is the best; the extra slack obtained through sacrificing manufacturing span makes the scheduling procedure very flexible. This flexibility is partly lost in II, with the decrease in extra slack; III is better than II, for it uses the traditional method in the majority of projects, thus keeping its flexibility. The flexibility of IV is not due to slack, but to the possibility of analyzing bottlenecks and rescheduling of all activities contributing to it.

As to objective 2, the traditional method (I) does very badly, as was shown in 1.1. Method III is also rather bad, for it uses the traditional method in most projects; only the two or three priority projects are "optimized" through the basic critical path algorithm. Computers (II) are visibly better than (I), due to the great decrease in extra slack. Critical Path Techniques are very good, for through the concept of float they "optimize" completion times.

As to objective 3, method I does not analyze all conflicting activities when a bottleneck occurs; so it relies on its extra slack to prevent completion delays. Computers (II) would neither have the extra slack (if the delay-units are decreased) nor the capacity of analyzing (and rescheduling) conflicting activities in a bottleneck, so they do badly under objective 3. III is good, for it schedules very carefully the projects with high delay penalties; IV is the best, for through analysis of all conflicting activities in all projects, it has more power to solve conflicts. In this sense, IV includes in the planning phase characteristics of the expediting process.

Systemic costs vary with the size of the machine shop. In a medium-to-large machine shop, computers (II) would cost usually less than the manual posting procedure(I). As the bulk of the activities would be scheduled manually in III, its systemic costs are not very low either. Computer time requirements of IV are larger than II.

Obviously all those ratings are qualitative only, and so rather gross estimates; small variations in the techniques might improve drastically its rating. Note also the relative importance of the objectives can vary markedly from machine shop to machine shop; thus, if objective 1 is of the utmost importance, the traditional scheduling system still might be the best; or, in small machine shops, the use of computers would increase very much systemic costs; method III or I would then be recommended.

A general representation of the problem would require assigning weights to all objectives, or then just giving the cost (or "value") of each objective. The difficulties and uncertainties involved in such an analysis are obvious; it is recommended that the objectives be evaluated in each situation and the systems proposed rated under the relevant objectives, (if possible, through simulation). In a very gross way, though, we can see that critical path techniques do very well on the whole in the comparison; but for objective 1, they do as well or better than all the others; if we consider all objectives to have the same importance (and consequently the same weights), the "Grade Point Average" of IV is noticeably higher than all the others.

We can then state the following conclusions:

1. No exact and feasible solution has yet been found for the scheduling problem through critical path concepts.

2. Good approximate solutions are available today. One of them is the use of critical path techniques.

3. These critical path-based, resource allocation-oriented

techniques compare favorably with other existing techniques in most cases.

4. The greatest difficulty in the optimization of the scheduling is to state the problem and assign weights to the objectives. Once this is done, simulation could then be used to rate proposed methods under the objectives.

4.2 FUTURE PERSPECTIVES

There is always a possibility that mathematical methods will be developed to really optimize scheduling systems in machine shops. The greatest difficulty, as was seen in 4.1, is to find appropriate variables so that an objective function can be stated in terms of cost, taking then all four objectives (types of costs) into account. At present, some methods optimize one or another objective, but no method optimizes exactly all the objectives. Integer programming solutions can then be developed to solve the general problem, although present algorithms are too cumbersome to solve practical-sized problems.

Another important obstacle to the discovery of exact optimization methods is that they are not really necessary. The present heuristic techniques give very good approximate solutions, and thus management is not striving very hard towards exact solutions.

As to the heuristic solutions, it was seen that several are already used operationally today. It is also possible to build a "do it yourself" kit for computer programmers, with all available gadgets (loading rules and other tricks) so that the planner can choose the ones that are most appropriate for his particular problem.

The next step to be done is to test the existing programs under simulation, to check, objective by objective, how they do compare exactly. Having ratings established precisely, and assigning weights to objectives according to particular conditions in each machine shop, a general method of designing scheduling systems can then be evolved.

Another work that would be helpful in building this general scheduling design technology would be a survey and analysis of the relative importance given to the four objectives by management in actual machine shops. Such a survey might show that some of these objectives are not considered to be important enough to be included in a general objective function.

CHAPTER V

BIBLIOGRAPHY

Production Scheduling

1. Akers, S. B., and Friedman, J., "A Non-Numerical Approach to Production Scheduling Problems", Operations Research, Vol. 3, 1955.

2. Azevedo, R. da R., and Lauretti, R., "Um Ensaio em Consultoria: Aplicacao do Medodo do Ratio-Delay", Graduation Thesis; Instituto Tecnologico de Aeronautica, S. Jose dos Campos, 1965.

3. Fischer, H., and Thompson, G. L., "Probabilistic Learning Combinations of Local Job-shop Scheduling Rules", Industrial Scheduling, p. 225.

4. Giffler, Bernard, "Simulation Models in Production Scheduling and Inventory Control", Production and Inventory Management, Vol. 7, No. 3, July 1966.

5. Giffler, Bernard., Thompson, G. L., and Van Ness, B., "Numerical Experience with the Linear and Monte Carlo Algoritims for Solving Production Scheduling Problems," Industrial Scheduling.

6. Johnson, S. M., "Optimal Two-and Three-State Production Schedules with Set-up Times Included," Naval Research Logistics Quarterly, Vol 1., 1954.

7. Künzi, C., "Contribution a L'Organization d'une Fabrication de Gros Ensembles Unitaires", These De Doctorat, Ecole Polytechnique Federal, Zurich; Imprimerie Leemann SA, Zurich 1961

8. Marshall, A. W. and Meckling, W. H., "Predictability of the Costs, Time and Success of Development," RAND Corporation, Report, December 1959, p. 14 and 20-22.

9. Merton, J. P., and Scherer, F. M., "The Weapons Aquisition Process: An Economic Analysis" Division of Research, Graduate School of Business Administration, Harvard University, Cambridge, Mass. 1962, p. 22 and 433.

10. Moore, J. M., and Wilson, R. C., "A Review of Simulation Research in Job-Shop Scheduling," Production and Inventory Management, January 1967, Vol. 8, No. 1.

11. Reid, M., "Job-Shop Production Control on Computer," Control Engineer, Vol. 8, No. 1, January 1961, pp. 107-109.

12. Mrs. Wallace Clark, "The Gantt Chart", Industrial Engineering Handbook, 2nd. Edition, McGraw-Hill Book Company, Inc., New York, 1963.

Line of Balance Technology

13. Catalano, A. J., "Line of Balance Shows Production Problems," Eletronics, Vol. 35, No. 35, August 31, 1962; pp. 66, 68.

14. Curtis, B. D., "Performance Index and Overall Status from Line of Balance," ASME-Paper 62-WA -109 for meeting November 25-30, 1962

15. Fleischer, G. A., "Linha de Controle - Um Metodo Grafico para o Planejamento da Producao," Instituto Tecnologico de Aeronautica, S. Jose dos Campos, 1964.

16. Gehringer, A. C., "Line of Balance," The Armed Forces Comptroller, June 1961.

17. "The Line of Balance as Applied to Production Analysis-Two Case Studies: Assembly Line Production and a Job-Shop Operation," Office of Naval Material, Department of the Navy, Washington 25, D. C.

18. Anon., "Line of Balance Technology," Office of Naval Material, U. S. Navy (NAVEXOSP 1851 - Rev. 4-62), Washington, D. C., April 1962.

Critical Path Techniques: History

19. Kane, J. S., "Origin of CPM and PERT," Hofstra University Yearbook of Business, op. cit.

Critical Path Techniques: Resource Allocation

20. Blair, R. J., "Critical Path Resources Simulation and Scheduling," IEEE-Transactions on Engineering Management, Vol. Em-10, No. 3, September 1963, pp. 100-103.

21. Burgess, A. R. and Killebrew, J. B., "Variation in Activity Level on a Cyclical Arrow Diagram," Journal of Industrial Engineering, Vol. 13, No. 2, March-April 1962.

22. American Institute of Industrial Engineering, Proceedings of the 15Th Annual Conference and Convention, New York, N. Y., 1964.

23. RAMPS Training Text and User's Guide, CEIR, Inc., New York, 1962.

24. Davis, E. W., "Resource Allocation in Project Network Models-Survey, "Journal of Industrial Engineering, Vol. 17, No. 4, April 1966, p. 177-188.

25. De Witte, L., "Manpower Leveling of PERT Networks," Data Processing for Science/Engineering, March-April, 1964.

26. Dike, S. H., "Project Scheduling with Resource Constraints," IEEE-Transactions on Engineering Management, Vol. EM-11, No. 4, December 1964, pp. 155-157.

27. Fey C. F., "Methods of Resource Allocations, "I and II, IBM, 1963, 1964.

28. Chare, P. M., "Optimal Resource Allocation in Activity Networks, "Central Library, Oklahoma State University, Stillwater, Oklahoma, August, 1965.

29. Johnson, E. R., "Optimum Allocation of Resources for Homogeneous Projects," ASME-Paper 64-PROD-16 for meeting April 20-24, 1965.

30. Kelley, J. E., Jr., "Scheduling Activities to Satisfy Constraints on Resources," Industrial Scheduling, op. cit.

31. Lambourn, S., "Resource Allocation and Multi-Project Scheduling (RAMPS)-New Tool in Planning and Control," Computer Journal, Vol. 5. No. 4, January 1963, p. 300-304.

32. Levy, F. K., Thompson, G. S., and Wiest, J. D., "Multi-ship, Multi-shop Workload Smoothing Program," Naval Research Logistics Quarterly, March 1963.

33. Marchbanks, J. L, "Daily Automatic Rescheduling Technique," Journal of Industrial Engineering, Vol, No. 3, March 1966, p. 119.

34. McGee A. A., and Markarian, M. D., "Optimum Allocation of Research/Engineering Manpower Within a Multi-Project Organizational Structure," IRE-Transactions on Engineering Management , Vol. Em-9, No. 3, September 1962, p. 104.

35. Mize, J. H., "An Investigation of Project Planning and Control Techniques," Main Library, Memorial Center, Purdue University, Lafayette, Indiana, January 1963.

36. Moodie, C. L. and Mandeville, D. E., "Project Resource Balancing by Assembly Line Balancing Techniques," Journal of Industrial Engineering, Vol. 17, No. 7, July 1966, p. 377.

37. Moshman, J., Johnson, J., and Larson, M., "RAMPS - Technique for Resource Allocation and Multi-Project Scheduling," Proceedings of the 1963 Spring Joint Computer Conference, American Federation of Information Processing Society (AFIPS), Vol 23, 1963, p. 17/

38. Roman, and Johnson, "On the Allocation of Common Physical Resources to Multiple Development Tasks," IEEE-Transactions on Engineering Management, Vol. Em-14, March 1967, No. 1, p. 16.

39. Verhines, D. R., "Optimum Scheduling of Limited Resources," Chemical Engineering Progress, Vol. 59, No. 3, March 1963, p. 657.

40. Wiest, J. E., "Some Properties of Schedules for Large Projects With Limited Resources," Operations Research, Vol. 12, p. 395, May 1964.

41. Wison, R. C., "Assembly Line Balancing and Resource Leveling" University of Michigan Engineering Summer Conference, Production and Inventory Control, 1964.

42. Rosenbloom, R. S., "Notes on the Development of Network Models for Resource Allocation in R & D Projects," IRE-Transactions on Engineering Management, Vol. EM-11, p. 58, June 1964.

<u>Critical Path Techniques - Time/Cost Trade-off</u>

43. Alpert, L., and Orkland, D. S., "A Time Resource Tradeoff Model for Aiding Management Decisions," Operations Research, Inc., Technical Paper No. 12, Silver Spring, Maryland, 1962.

44. Black, C. J., "An Algorithm for Resource Leveling in Project Networks," Unpublished paper, Department of Industrial Administration, Yale University, May 1965.

45. Berman, E. B., "Resource Allocations in a PERT Network Under Continuous Time/Cost Functions," Management Science, July, 1964.

46. Charnes, A., and Cooper, W. W., "A Network Interpretation and a Directed Sub-Dual Algorithm for Critical Path Scheduling," Journal of Industrial Engineering, 1962, 213.

47. Clark, C. E., "Optimum Allocation of Resources Among the Activities of a Network," Journal of Industrial Engineering, Vol. 12, No. 1, January-February 1961, p. 11.

48. Frantz, R. A., Jr., and Northern, L. B., "PERT/COST," Western Electric Engineer, Vol. 8, No. 3, July 1964, p. 24.

49. Fulkerson, D. R., "A Network Flow Computation for Project Cost Curves," Management Science, Vol. 7, 1961, p. 167.

50. Fulkerson, D. R., "Out-of-Kilter Method for Minimal-Cost Flow Problems," Journal of the Society of Industrial and Applied Mathematics, Vol. 9, No. 1, March 61, p. 18.

51. Fulkerson, D. R., "Scheduling in Project Networks," RAND Corporation Memo. RM-4137-PR, June 1964.

52. Handa, V. K., "Project Cost Curve Equivalent Linear Graphs," Paper presented at ORSA Meeting, Boston, May 1965.

53. Jewell, W. S., "Risk-Taking in Critical Path Analysis," Management Science, Vol. 11, No. 3, January 1965, p. 438.

54. Kelley, J. E., and Walker, M. R., "Critical Path Planning and Scheduling," 1959 Proceedings of the Eastern Joint Computer Conference.

55. Kelley J. E. Jr., "Critical Path Planning And Scheduling-Mathematical Basis," Operations Research, Vol. 9, No. 3, 1961.

56. Meyer, W. L., and Schaffer, R. L., "Extensions of the Critical Path Method Through the Application of Integer Programming," Department of Civil Engineering, University of Illinois, July 1963.

57. Anon, "DOD and NASA Guide, PERT/COST," Office of the Secretary of Defense and National Aeronautics and Space Administration, Washington, D. C., 1962.

58. Parikh, S. C., and Jewell, W. S., "Decomposition of Project Networks," Management Science, Vol. 11, No. 3, January 1965, p. 444.

59. Prager, W., "A Structural Method of Computing Project Cost Curves," Management Science, Vol. 9, p. 394, 1963.

60. Thompson, T. E., "Adjusting Network Plans with "PERT Slack Bonus"," Journal of Industrial Engineering, Vol. 17, No. 145.

Critical Path Techniques - Mathematical Basis of PERT

61. Barnby, J. G., "Applicability of PERT as a Management Tool," IRE -Transactions of Engineering Management, Vol. EM-9, No. 3, September 1962, p. 130

62. Charnes, A., "Critical Path Analysis via Chance Constrained and Stochastic Programming," Operations Research, May-June 1965, Vol. 13, No. 3, p. 382.

63. Clark , C. E., "The Greatest of a Finite Set of Random Variables," Operations Research, 9, p. 145, 1961.

64. Coon, H., "Note on William A. Donaldson's "The Estimation of the Mean and Variance of a PERT Activity Time"," Operations Research, May-June 1965, Vol. 13, No. 3, p. 386.

65. Donaldson, W. A., "The Estimation of the Mean and Variance of a PERT Activity Time," Operations Research, May-June 1965, Vol. 13, No. 3, p. 382.

66. Healy, T. L., "Activity Subdivision and PERT Probability Statements," Operations Research Vol. 9, No. 3, May-June 1961, p. 341.

67. Lukaszewicz, J., "On the Estimation of Errors Introduced by Standard Assumptions Concerning the Distribution of Activity Duration in PERT Calculations," Operations Research, March-April 1965, Vol. 13, No. 2, p. 326

68. Malcom, D. G., Roseboom, J. H., Clark, C. E., and Fazar, W., "Applications of a Technique for R and D Program Evaluation (PERT)," Operations Research, Vol. 7, No. 5, September-October 1959, p. 646.

69. MacCrimmon, K. R., and Ryavec, C. A., "Analytical Study of PERT Assumptions," Operations Research, Vol. 12, No. 1, January-February 1964, p. 16.

70. MacCrimmon, K. R., and Ryavec, C. A., "Analytical Study of the PERT Assumptions," RAND Corporation Memo RM-3408-PR, December 1962.

71. Murray, J. E., "Consideration of PERT Assumptions," IEEE - Transactions on Engineering Management, Vol. EM-10, No. 3, September 1963, p. 94.

72. Pocock, J. W., "PERT as Analytical Aid for Program Planning -Its Pay-off and Problems," Operations Research, Vol. 10, No. 6, November-December 1962.

73. Robinson, F. D., "Background of PERT Algorithm," Computer Journal, Vol. 5, No. 4, January 1963, p. 297.

74. Van Slyke, R. M., "Monte Carlo Methods and PERT Problems," Operations Research, Vol. 11, No. 5, September-October 1963, p. 839.

75. Walker, J. D., and Houry, E., "Comparison of Actual and Allocated Costs for Work Accomplished Using NASA PERT", IEEE-Transactions on Engineering Management, Vol. EM-a2, No. 3, September, 1965.

76. Welsh, J. A., "Errors Introduced By a PERT Assumption," Operations Research, January-February 1965, Vol. 13, No. 1, p. 141.

77. Welsh J. A., "Super-Critical Arcs of a PERT Network," Operations Research, Vol. 10, 1962, p. 912.

78. Grubbs, F. E., "Attempts to Validate Certain PERT Statistics," Operations Research, Vol. 10, 1962, p. 912.

79. Clark, C. E., "The PERT Model for the Distribution of an Activity Time," Operations Research, Vol. 10, 1962, p. 405.

Critical Path Techniques - Bibliographies

80. Anon., "A Bibliography of PERT and CPM," Industrielle Organization (February, 1963), Postfach, Zurick, Switzerland.

81. Dieckert, E. A., "Dissemination of PERT/CPM Information Within Metals Industry," Hofstra University Yearbook of Business, op. cit.

82. Dooling, T. A., "Dissemination of Critical Path Network Technique for Project Scheduling and Control in Construction Industry," Hofstra University Yearbook of Business, op. cit.

83. Fry, B. L., "Network-Type Management Control Systems Bibliography," Prepared for U. S. Air Force Project RAND, Memorandum RM-3074-PR, RAND Corporation, 1700 Main St., Santa Monica, California, February 1963.

84. Fry, B. L., "Selected References on PERT and Related Techniques," IEEE-Transactions on Engineering Management, Vol. EM-10, No. 3, September 1963, p. 150.

85. Lamb, R. J., "Management Network Techniques (CPM, PERT, etc.,) Appearing in Periodical Literature of Chemical and Petroleum Industries," Hofstra University Yearbook of Business, op. cit.

86. Lerda-Olberg, S., "Bibliography on Network-Based Project-Planning and Control Techniques: 1962-1965," Operations Research, September-October 1966, Bol. 14, No. 5, p. 925.

87. Meyer, J. W., "Dissemination of Network Techniques for Project Schedule and Control Through Electronic Industry Trade Journals," Hofstra University Yearbook of Business, op. cit.

88. Murray, A. R., "CPM/PERT - Dissemination via Aerospace Periodicals," Hofstra University Yearbook of Business, op. cit.

89. Phillips, C. R., "Fifteen Key Features of Computer Programs for CPM and PERT," Journal of Industrial Engineering, Vol. 15, No. 1, January-February 1964, p. 14.

90. Poletti, G., "Diffusion of Network Techniques Throughout Government Publications," IEEE-Transactions on Engineering Management, Vol. EM-11, No. 1, March 1964, p. 43.

91. Sobczac, T. V., "Network Planning - A Bibliography," Journal of Industrial Engineering, Vol. 13, No. 6, November-December 1962.

92. Spanier, A., "Dissemination of Network Techniques by Professional Societies," Hofstra University Yearbook of Business, op. cit.

93. Warner, W. C., "Dissemination of PERT and Critical Path Information in General Business Periodicals," Hofstra University Yearbook of Business, op. cit.

94. Zinghini, F. J., "Monographic and Text Literature," Hofstra University Yearbook of Business, op. cit.

95. Zinghini, F. J., "Network Scheduling and Control Systems-Bibliography Through December 1962," Hofstra University Yearbook of Business, op. cit.

<u>Critical Path Techniques - Foreign Publications</u>

96. Angelini, A. M., "Analisi e Sintesi dei Programmi - Un Procedimento Grafico di Valutazioni Analitica i Sintetica i di Aggiornamiento dei Programmi Construtici," Energia Electrica, vol. 40, No. 12, December 1963, p. 941; Vol. 41, No. 5, May 1964, p. 335.

97. Dreger, W., "Netzplantechnik - eine mathematische Planungsmethode," Foerdern u Heben, Vol. 15, No. 5, May 1965, p. 377.

98. Dreger, W., "Verfeinerung und Weiterentwicklung der Netzplantechnik," Foerdern u Heben, Vol. 16, No. 5, May 1966, p. 415.

99. Eugen, Y., and Baudouin, G., "Deux Cas Concrets d'Application de la Methode PERT," Automatisme, Vol. 10. No. 10, October 1965, p. 406

100. Gewald, K., and Kasper, K., "Erfahrungen bei der Anwendung von Projektplanungsmethoden (Netzplantechnik)," Elecktronische Rechenaulagen, Vol. 7, No. 3, June 1965, p. 147.

101. Mattion, A. B., "El Metodo del Camino Critico - una Experiencia en la Programacion de Obras," Ingenieria (Buenos Aires), Vol. 68, No. 989, June-August 1964, p. 9.

102. Rodriguez Gomez, S. "Sistema PERT para Plancacion y control de Proyectos," ATCP, Vol. 4, No. 1, January-February 1964, p. 27.

103. Rybashov, V. V. and Dudnikov, Y. Y., "Reshenie zadachi minimizatsii stoimosti razrabotking analogovykh modelykh," Akademiya Nauk SSSR, Izvestiya, Tekhnicheskaya Kibernetika, No. 4, July-August 1964, p. 84.

104. Vilenkin, S. Y., "Determination of Maximum time (critical path time) Distribution," Automatika i Telemekhanika, Vol. 26, No. 7, July 1965, p. 1235; English Translation in Automation and Remote Control, Vol. 26, No. 7, July 1965, p. 1233.

105. Westry, C., "Het samenstellen van tijdschema's m.b.v. "eleck tronische rekenmachines," Ingeniur, Vol. 74, No. 30, July 27, 1962, P. B185.

106. Wille, H., Gewald, K, and Weber, H. D., "Netzplanmodelle fuer die Planung von Projekten," Elektronische Rechenaulagen, Vol. 6, No. 6, December 1964, p. 277.

5.8. <u>Critical Path Techniques - Miscellaneous</u>

107. Anon., "Critical Path Techniques-Communications Shorthand for Management," Steel, Vol. 151 No. 21, November 19, 1962, p. 74.

108. Anon., "Outline of CPM and PERT" Factory and Plant, Vol. 51, No. 11, June 1964, p. 40.

109. Aras, T. M., Surkis, J., "PERT and CPM Techniques in Project Management," ASCE - Proceedings, Vol. 90, (Journal of Construction Management), No. CO1, March 1964, paper 3823, p. 1.

110. Astrop, A. W., "Critical Path Analysis," Machinery (London), Vol. 104, No. 2677, March 4, 1964, p. 523.

111. Athan, P., McCord, M., "RITE Scans PERT," Aerospace Management, Vol. 5, No. 4, April 1962.

112. Baldwin, T., "PERT - New Management Technique," AEI Engineering, October 1963, p. 58.

113. Baker, B. N., "Making PERT Work," Space/Aeronautics, Vol. 37, No. 3, March 1962, p. 58.

114. Barker, R. H., "Critical Path Network Analyzer," Engineer, Vol. 219, No. 5697, April 2, 1965, p. 595.

115. Beckwith, R. E., "Cost Control Extension of PERT System," IRE - Transactions on Engineering Management," Vol. EM-9, No. 4, December 1962, p. 147.

116. Boulanger, D. G., "Managing PERT Program," Machine Design, Vol. 34, No. 27, November 22, 1962, p. 130.

117. Boverie, R. T., "Practicalities of PERT," IEEE - Transactions on Engineering Management, Vol. EM-10, No. 1, March 1963, p. 3.

118. Bradford, L. R., "Controlling Progress of Work in Design Engineering Department," ASME - Paper 66-MD-62 for meeting May 9-12, 1966.

119. Brooke, E. R., "Project Planning," Machine Design, Vol. 33, No. 16, August 3, 1961, p. 76.

120. Calica, A. B., "Fabrication and Assembly Operations - 5," IBM Systems Journal, Vol. 4, No. 3, 1965, p. 224.

121. Christensen, B. M., "Network Models for Production Scheduling," Machine Design, Vol. 34, No. 11, 12, 13, 15, 16, 17, May 10 1962, p. 114. May 24, p. 173, June 7, p. 132, June 21, p. 155, July 5 p. 105, July 19, p. 136.

122. Carson, J. M., "Critical Path Method Gives Better Project Control," British Chemical Engineering, Vol. 10, No. 4, April 1965, p. 248.

123. Christensen, B. M., "How to Take Guesswork out of Project Planning," Iron Age, Vol. 188, No. 5, August 3, 1961, p. 67.

124. Clark, T., Board, N. F., Hincks, H. A., "Keeping Production on Schedule," Iron Age, Vol. 195, No. 3, January 21, 1965, p. 54.

125. Cosinuke, W., "Critical-Path Technique for Planning and Scheduling," Chemical Engineering, Vol. 69, No. 13, June 25, 1962, p. 113.

126. Cummins, J. J., "Current Applications of Network Techniques (CPM/ PERT) for Project Scheduling and Control," Hofstra University Yearbook of Business, op. cit.

127. Davis, K., "Role of Project Management in Scientific Manufacturing," IRE - Transactions on Engineering Management, Vol. EM-9, No. 3, September 1962, p. 122.

128. Davis, G. B., "Network Techniques and Accounting with Illustration," National Association of Accountants Bulletin, Vol. 44, No. 9, section1, May 1963, p. 11.

129. Deangeli, "Hand Performed Computations for Network Control," Journal of Industrial Engineering, Vol. 18, No. 2,

130. Digman, "PERT/LOB: Life-Cycle Technique," Journal of Industrial Engineering, Vol. 18, No. 2, February 1967, p. 155.

131. Elmaghraby, S. E., "Algebra for Analysis of Generalized Activity Networks," Management Science, Vol. 10, No. 3.

132. Fine, B., and Whattingham, D., "Planning Large Scale Contract Work," Engineering, Vol. 196, No. 5078, August 16, 1963, p. 220.

133. Fischer, C., and Nemhauser, G. L., "Multicycle Project Planning," Journal of Industrial Engineering, Vol. 18, April 1967, No. 4, p. 278.

134. Ford, J. W., "Project Planning Techniques," Chemical and Process Engineering, Vol. 43, No. 11, November 1962, p. 613.

135. Frantz, R. A. Jr., Nothern L. B., "Introduction to PERT," Western Electric Engineer, Vol. 43, No. 4, October 1963, p. 38.

136. Frantz, R. A. Jr., and Nothern, L. B., "Can PERT Improve your R and D Planning," Machinery (N. Y.), Vol. 70, No. 10, June 1964, p. 117.

137. Fry, B. I., "SCANS-System Description and Comparison With PERT," IRE-Transactions on Engineering Management, Vol. EM-9, No. 3, September 1962, p. 122.

138. Fry, B. L., "How to Use SCANS," Aerospace Management, Vol. 5, No. 11, November 1962, p. 26.

139. Getz, C. W., "PERT in Perspective," SAE-Paper 557 A for Meeting September 10-13, 1962.

140. Gleason, W. J., "First Five Years of Critical-Path Method," ASCE-Proceedings, Vol. 90, (Journal of Construction Division), No. CO1, March 1964, paper 3832, p. 27.

141. Greene, D. E., "Critical Path Scheduling," Production Engineer, Vol. 42, No. 12, December 1963, p. 736.

142. Halcomb, J. L., "Use of PERT in Product Design," ASME-Paper 63-MD-13 for meeting May s0-23, 1963.

143. Harben, J. F. C., "More Accurate Cost Estimating and Time Scheduling," ASME-Paper 62-MD-18 for meeting April 30-May 3, 1962.

144. Healy, B. W., "Network Control Systems," Hofstra University Yearbook of Business, op. cit.

145. Hill, L. S. "Some Cost Accounting Problems in PERT/Cost," Journal of Industrial Engineering, Vol. 17, No. 2, February 1966, p. 87.

146. Hochwald, W., Miller, B. U., Ashcraft, W. D., "SCOPE - (Schedule, Cost, Performance) Planning and Control System," IEEE - Transactions on Engineering Management, Vol. EM-13, No. 1, March 1966, p. 2.

147. Iannone, A., "How Avco Put MOST in Minuteman," Aerospace Management, Vol. 6, No. 5, May 1963, p. 21.

148. Jewell, W. S., "Divisible Activities in Critical Path Analysis," Operations Research, September-October 1965, Vol. 13, No. 5, p. 747.

149. Jodka, J., "PERT-Recent Control Concept," National Association of Accountants Bulletin, Vol. 43, No. 5, sec. 1, January 1962, p. 81.

150. Johnson, E. J., "PERT/PMD - Project Monitoring Device," IEEE-Transactions on Engineering Management, Vol. EM-11, No. 2, June 1964, p. 81.

151. Jonas, D. L., "Reducing Equipment Needs with CPM," Western Construction, Vol. 39, No. 11, November 1964, p. 64.

152. Kadet, J., Frank, B. H., "PERT for Engineer," IEEE Spectrum, Vol. 1, No. 11, November 1964, p. 131.

153. Kedrowsky, G. V., and Vaughan, B. W., "SYNCOM Event Schedule Networks-Best of PERT," IEEE - Transactions on Engineering Management, Vol. EM-10, No. 3, September 1963, p. 104.

154. Kelley, J. E. Jr., Wilson, I. D. and Berman, H., "Using Critical Path Programming," Automation, Vol. 9, No. 11, November 1962, p. 90.

155. Kelton, G., ""PERT and Automation Philosophy,"Research Management, Vol. 7, No. 1, January 1964, p. 55.

156. Kochanski, S. L., "Critical Path Techniques - Appraisal," Engineer, Vol. 215, No. 5592, March 29, 1963, p. 559.

157. Kochanski, S. L., "Critical Paths and True Productivity," Engineering, Vol. 198, No. 5124, July 3, 1964, p. 2.

158. Loeber, N. C., "PERT for Small Projects," Machine Design, Vol. 34, No. 25, October 25, 1962, p. 134.

159. Love C. W., "Critical-Path Scheduling Simplified," Chemical Engineering, Vol. 69, No. 25, December 10, 1962, p. 170.

160. Magnus, H. A., "Simplified Network Planning Techniques," Machine Design, Vol. 38, No. 1. January 6, 1966, p. 102.

161. Martin, J. J., "Distribution of Time through Directed Acyclic Network," Operations Research, Vol. 13, No. 1, January-February 1965, p. 46.

162. Martino, R. L., and Popper, H., "How Useful Are Critical-Path Methods," Chemical Engineering, Vol. 71, No. 19, September 14, 1964, p. 220.

163. Martino, R. L., "How Critical-Path Scheduling Works," Canadian Chemical Processing, Vol. 44, No. 2, February 1960, p. 38.

164. Matthews, R., "How to Implement PERT costs," Aerospace Management, Vol. 6, No. 10, October 1963, p. 18.

165. Mattozzi, M., and Lipinski, F., "New Approach to Project Scheduling - Control Operation Technique," Chemical Engineering Progress, Vol. 59, No. 3, March 1963, p. 657.

166. Mayhuch, J. O., "On Mathematical Theory of Schedules," Management Science, Vol. 11, No. 2, November 1964, p. 289.

167. Maynes, W., "What's Wrong with PERT," Aerospace Management, Vol. 5, No. 4, April 1962, p. 20.

168. McMullan, V. J., and Coxan, D. J., "Project Control by Computer," A.T.E.J., Vol. 21, No. 3, July 1965.

169. Meyer, W. L., "PROPS-Improved CPM Technique for Project Planning and Control," SAE-Paper S427 (Central Illinois Section) for meeting April 6-7, 1965.

170. Morrisreg, G. L., "Management Control - NAMDA Plus Computer Equals On-Time Deliveries," Tool and Manufacturing Engineering, Vol. 65., Book 4, paper SP65-21, 1965.

171. Morrisroe, G. L., "NAMDA - New Method of Computerized Management of Major Ballistic Missile Modification Programs," Journal of Industrial Engineering, Vol. 16, No. 4, July-August 1965, p. 274.

172. Molieri, J. T., "Control and Monitoring Techniques in ACCD-C Engineering Department," IEEE Int. Convention Record, Vol. 11, Part 10, 1963, p. 59.

173. Newham, Hudson et at., " A PERT Control Center for Management of Major Ballistic Missile Modification Programs," Journal of Industrial Engineering, Vol. 16, No. 4, July-August 1965, p. 274.

174. O'Brien, J. J., "Project Scheduling - Second Generation," Machine Design, Vol. 37, No. 10, April 29, 1965, p. 172.

175. Odom, R. G., and Blystone, E. E., "Case Study of CPM in Manufacturing Situation," Journal of Industrial Engineering, Vol. 15, No. 6, November-December 1964, p. 306.

176. Ohlinger, L. A., "New Scientific Management Technique (WSP-AC3 Study)," Aerospace Engineering, Vol. 20, No. 11, November 1961, p. 10.

177. Pearlman, J., "Engineering Program Planning and Control Through the Use of PERT," IRE-Transactions on Engineering Management, Vol. EM-7, No. 4, December 1960, p. 125.

178. Pristker, and Happ, "GERT: Graphical Evaluation and Review Technique - Part I. Fundamentals," Journal of Industrial Engineering, Vol 17, No. 5, May 1966, p. 267, "Part II-Probabilistic and Industrial Engineering Applications," Journal of Industrial Engineering, Vol. 17, No. 6, June 1966, p. 293.

179. Prostick, J. M., "Network Integration," IEEE-Transactions on Engineering Management, Vol. EM-10, No. 2, June 1963, p. 65.

180. Rensen, J. S., "PERT as Value Engineering Tool," Journal of Value Engineering, No. 9-62-1, September, 1962, p. 14.

181. Riggs, J. L. and Inoue, M. S., "Critical Path Scheduling: Tableau Method," Proceedings of the Seventeenth Annual Institute Conference and Convention of AIIE.

182. Rogers, J. J., "Use of Critical Path Scheduling on Complex Jobs," ASTME 33rd Annual meeting- Collected Papers, Vol. 65, book 4, paper SP65-21, 1965.

183. Roper, D. E., "Critical Path Scheduling," Journal of Industrial Engineering, Vol. 15, No. 2, March-April 1964, p. 51.

184. Sadow, R. M., "How PERT was Used in Managing X-20 (Dyna-Soar) Program," IEEE-Transactions on Engineering Management, Vol. EM-11, No. 4, December 1964, p. 138.

185. Sando, F. A., "CPM - What Factors Determine its Success," Architectural Record, Vol. 135, No. 4, 5, April 1964, p. 211, May 1964, p. 202.

186. Sayer, J. S., Kelley, J. E. Jr., and Walker, M. R., "Critical Path Scheduling," Factory, Vol. 118, No. 7, July 1960, p. 74.

187. Schaidt, L., "Program Management Techniques at Martin Orlando," IEEE-Transactions on Engineering Management, Vol. EM-10, No. 3, September 1963, p. 124.

188. Senecal, T. L., and Sadow, R. M., "PERT in Dyna-Soar," Aerospace Management, Vol. 5, No. 12, December 1962, p. 18.

189. Sobczak, T. V., "Look at Network Planning," IRE-Transactions on Engineering Management, Vol. EM-9, No. 3, September 1962, p. 113.

190. Stagg, G. W., Palmer, A. H., and Veldon, C. A., "PERT Schedules Manpower," Electrical World," Vol. 158, No. 5, July 30 1962, p. 36.

191. Stahl, N., "Relating Organizational Needs to Improved Engineering Information Flow - Applying Network Principles to Flow of Engineering, Vol. 67, No. 24, November 28, 1960, p. 148.

192. Steinfeld, R. C., "Critical-Path Saves Time and Money," Chemical Engineering, Vol. 67, No. 24, November 28, 1960, p. 148.

193. Szuprowicz, B. O., "Choosing Critical-Path Scheduling Program," Engineering News-Record, Vol. 172, No. 25, June 18, 1964, p. 78.

194. Szuprowicz, B. O., "When PERT is Mandatory," Data Processing Magazine, 7, 6, (June 1965), p. 38.

195. Swim, R. K., "Graphic PERT/Cost Milestone Reporting," SAE-Paper 576 D for meeting October 8-12, 1962.

196. Taylor, R. B., "Hoffman Gets HEPP," Aerospace Management, Vol. 5, No. 7, July 1962, p. 31.

197. Tebo, L. M., "PERT - Technique for Management," SAE-Paper 557D for meeting September 10-13, 1962.

198. Viehmann, N. J., "PERT Control of Industrial Projects," Automations, Vol. No. 9, Septmeber 1963, p. 50.

199. Viehmann, N. J., "PERT as Means for Control in Multi-Projects Systems," Western Electric Engineer, Vol. 7, No. 4, October 1963, p. 43.

200. Waldron, A. J., "Use of PERT in Scheduling Manufacturing Operation," SAE-Paper 658A for National Production Meeting, Cleveland, Ohio, March 12-13, 1963.

201. Walton, H. "Critical-Path Method-How to Start," Chemical and Process Engineering, Vol. 45, No. 4, April 1964, p. 183.

202. Ward, T., "New Look at Manufacturing Startup, Planning and Control," ASME-Paper 65-WA/MGT-1 for meeting November 7-11, 1965.

203. Whittet, K. R., "Critical Path Programming," Metal Industry, Vol. 104, No. 2, January 9, 1964, p. 50.

204. Whittet, D. R., "Critical Path Programming," Metal Industry, Vol. 104, No. 5614, August 30, 1963, p. 345.

205. Wiest, H. D., "Some Properties of Schedules for Large Projects with Limited Resources," Operations Research, Vol. 12, No. 3, May-June 1964, p. 395.

206. Williams, R., "Solutions to Some Problems in Planning and Control of First Article Cost and Schedules," SAE-Paper 576A for meeting October 8-12, 1962.

207. Williamson, D. J., "Application of Critical Path Planning to Plant Design and Construction," Western Electric Engineer, Vol. 7, No. 4, October 1963, p. 46.

208. Wood, K. R., and Zagorski, H. J., "How to Schedule by Computer - Use MAPS," Aerospace Management, Vol. 5, No. 2, February 1962, p. 36.

209. Wookey, S. J., "Putting CPM/PERT to Work in Factory," Factory and Plant, Vol. 51, No. 12, July 1964, p. 26.

<u>Critical Path Techniques-BOOKS</u>

210. Antill, J. M., and Woodhead, R. W., <u>Critical Path Methods in Construction Practice</u>," John Wiley, New York, 1965.

211. Archibald, R. D., and Villoria, R. I., "Network-based Management Systems (PERT/CPM)," John Wiley, New York, 1967.

212. Bock, R. H., and Holstein, W. K., "Production Planning, Operation and Control: Text and Readings," C. E. Merrill Books, Columbus, Ohio, 1963.

213. Dooley, A. R., et al., "Casebooks in Production Management: Operations Planning and Control," John Wiley, New York, 1964.

214. Evarts, H. F., "Introduction to PERT," Allyn and Bacon, Inc.

215. "Hofstra University Yearbook of Business-Network Schedules and Control Systems (CPM/PERT)," Vol. 2, January 1964.

216. Levin, R. I. and Kilkpatrick, C. A., "Planning and Control with PERT/CPM," McGraw-Hill, New York, 1966.

217. Lockyer K. G., "An Introduction to Critical Path Analysis," Pitman Pub. Corp., New York, 1964.

218. Lowe, C. W.,"Critical Path Analysis By Bar Chart," Business Publications Limited, London, 1966.

219. Martino, R. L., "Project Management and Control - Vol. 1, Finding the Critical Path," and "Project Management and Control - Vol. 2, Applied Operational Planning," New York, American Management Association, 1964.

220. Miller, R. W., "Schedule, Cost and Profit Control with PERT", McGraw-Hill, New York, 1963.

221. Moder, J. J., and Phillips, C. R., "Project Management with CPM and PERT," New York, Reingold Publishing Corporation, 1964.

222. Muth, J. F., and Thompson, G. L., (editors), "Industrial Scheduling," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.

223. O'Brien, J. J., "CPM in Construction Management," McGraw-Hill Book Company, New York, 1965.

224. Riggs, J. L., and Heath, C. O., "Guide to Cost Reduction Through Critical Path Scheduling," Prentice-Hall, Inc., Englewood Cliffs, N. J., 1963.

225. Robertson, D. C, "Project Planning and Control - Simplified Critical Path Analysis," Heywood Books, London, 1967.

226. Schaffer, L. R., Ritter, J. B., and Meyer, W. L., "The Critical-Path Method," Mcgraw-Hill Book Company, New York, 1965.

227. Stilian, G. N., and others, "Pert - A New Management Planning and Control Technique," American Management Association, New York, 1962.

228. Maynard, H. B., (Editor-in-Chief),"Industrial Engineering Handbook," second edition, Mcgraw-Hill Book Company, Inc., New York, 1963.

229. Hillier, F. S. and Lieberman, G. J., " Introduction to Operations Research," Holden-Day, Inc., San Francisco, 1967.

APPLICATION OF CRITICAL PATH TECHNIQUES
TO PRODUCTION PLANNING

by

RICARDO DA ROCHA AZEVEDO

Engenheiro Mecanico

Instituto Tecnologico De Aeronautica

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of

the requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1963

This report studies the production planning problem for factories organized for job-shop production. In this type of production, products are custom-made, non-repetitive, large and complex. Under these conditions, the crucial step on the over-all planning process is machine shop scheduling.

The objectives to be met (or the variables to be "optimized," in a large sense) are:

1. Idle machine time (minimization)
2. In-process inventories (minimization)
3. Delay penalties (minimization)
4. Systemic costs (minimization)

The "traditional" method of solving this problem involves Gantt charts and clerical posting routine. This method gives fair results as to idle machine time (objective 1), but gives poor results under all other objectives.

Critical path techniques are one of the most successful and widely used Operations Research techniques invented in the last ten years. They consist of the graphical representation of a project as a network, and of algorithms to determine critical activities. The basic concepts involved are projects, network, slack and critical activity.

Basic critical path techniques are reviewed briefly in Chapter II. In Chapter III, all critical path-based scheduling methods that the author is aware of are discribed and analyzed. These techniques are shown to be markedly superior to all other presently used methods as to objectives 2 and 3, and are competitive as to objectives 1 and 4. As a whole, critical path-based methods such as RAMPS and DART are superior to (or at least, as good as) all other presently used methods. They are being used operationally in industry (although not yet on a large scale), and are tested and proven techniques.