

AD-A093 599

NAVAL POSTGRADUATE SCHOOL MONTEREY CA  
AN ALGORITHM FOR SCHEDULING A SHIP'S FORCE DURING A SHIPYARD OV--ETC(U)  
SEP 80 O KARADAG

F/G 5/1

UNCLASSIFIED

NL

1 of 1  
ADA  
093599

END

DATE

PLACED

IN

FILE

LEVEL II

2

NAVAL POSTGRADUATE SCHOOL  
Monterey, California

AD A 093599



DTIC  
SELECTED  
JAN 9 1981  
E

THESIS

AN ALGORITHM FOR SCHEDULING A SHIP'S FORCE  
DURING A SHIPYARD OVERHAUL

by

Osman Karadağ

September 1980

Thesis Advisor:

Ellen F. Roland

Approved for public release, distribution unlimited.

FILE COPY

81 1 08 030

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. HD 4093 599	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Algorithm for Scheduling a Ship's Force During a Shipyard Overhaul		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Osman Karadağ		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1980
		13. NUMBER OF PAGES 94
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Ship Overhaul Resource Allocation Combining Two Projects Scheduling		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis considers the combining of a ship overhaul project network with a shipyard project network to minimize possible scheduling conflicts during the ship overhaul process. The combined project is called the ship overhaul project network. A heuristic multi-pass scheduling algorithm is developed to combine the ship's and shipyard's project networks. The		



Approved for public release; distribution unlimited.

An Algorithm for Scheduling a Ship's Force  
During a Shipyard Overhaul

by

Osman Karadağ  
Lieutenant, Turkish Navy  
Turkish Naval Academy, 1974

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the  
NAVAL POSTGRADUATE SCHOOL  
September 1980

Author

O. Karadağ

Approved by:

Ellen F. Roland

Thesis Advisor

Alan W. Mc Masters

Second Reader

J. T. Howard, Acting

Chairman, Department of Operations Research

W. M. Woods

Dean of Information and Policy Sciences

## ABSTRACT

This thesis considers the combining of a ship overhaul project network with a shipyard project network to minimize possible scheduling conflicts during the ship overhaul process. The combined project is called the ship overhaul project network. A heuristic multi-pass scheduling algorithm is developed to combine the ship's and shipyard's project networks. The algorithm is tested with two sets of data. The work spaces on shipboard are considered as a common resource to both project networks, and the ship's complement as an uncommon but limited resource. Since the real-world problem is larger than the context of this thesis, some simplifying assumptions were made. The most important of these are the reliable communication link between ship and shipyard prior to and/or during the overhaul process; and sufficient knowledge about the shipyard project network as early as possible after overhaul initiation.

TABLE OF CONTENTS

I. INTRODUCTION -----10

II. BACKGROUND -----15

    A. NAVAL SHIPYARDS -----15

    B. SHIP'S FORCE -----16

    C. SHIP OVERHAUL PROCESS -----17

    D. APPROACHES TO SCHEDULING AND RESOURCE  
        ALLOCATION IN PROJECT NETWORKS -----19

III. DEVELOPMENT OF AN ALGORITHM -----23

    A. INTRODUCTION -----23

        1. Ship Overhaul Scheduling Problem -----24

        2. Divisible Activities -----25

    B. ASSUMPTIONS -----29

    C. THE CONCEPT OF INTERFACING BETWEEN NETWORKS ---31

    D. A PROPOSED ALGORITHM -----31

        1. Notation -----31

        2. General Description of the Algorithm -----33

        3. Algorithm -----34

IV. A COMPUTER PROGRAM FOR THE PROPOSED ALGORITHM ---44

    A. INTRODUCTION -----44

    B. DESCRIPTION OF THE PROGRAM -----45

        1. Main Program -----45

        2. Subroutines -----46

        3. Input -----47

        4. Possible Actions -----48

C.	RECOMMENDATIONS ON USING THE PROGRAM	-----	48
V.	TESTING THE PROPOSED ALGORITHM	-----	50
A.	GENERAL	-----	50
1.	Data	-----	50
B.	SCHEDULING RESULTS	-----	51
VI.	SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	-----	67
A.	SUMMARY AND CONCLUSIONS	-----	67
B.	RECOMMENDATIONS	-----	68
APPENDIX A:	VERBAL FLOWCHART	-----	71
APPENDIX B:	DESCRIPTION OF THE DATA CARDS	-----	75
COMPUTER PROGRAM:	COMBINING TWO PROJECT NETWORKS	-----	77
LIST OF REFERENCES		-----	91
INITIAL DISTRIBUTION LIST		-----	93

LIST OF TABLES

Table I	52
Table II	52
Table III	53
Table IV	56
Table V	61
Table VI	63
Table VII	66
Table VIII	66

LIST OF DRAWINGS

Figure 2.1	-----	18
Figure 3.1	-----	27
Figure 3.2	-----	28
Figure 3.3	-----	35
Figure 5.1	-----	58
Figure 5.2	-----	59

ACKNOWLEDGEMENT

The author wishes to acknowledge the many beneficial suggestions and comments of Lieutenant Ellen F. Roland. USN as thesis advisor and Professor Alan W. McMasters. Additionally, the author thanks Lieutenant James R. Howell, USN and Charles E. Sprenger, Long Beach Naval Shipyard for their time taken to expeditiously respond to questions from this author.

## I. INTRODUCTION

There are two parties working simultaneously on a ship, doing different but somewhat related jobs, during the ship overhaul process. One party is the shipyard personnel, the other is the ship's force complement.

These two parties have their own responsibilities during the ship overhaul process. Basically, the ship's force has the responsibility to accomplish the scheduled activities of the ship project network within the specified time limits without causing any delay in any activity of the shipyard project network. Additionally, the ship's force has the responsibility to carry on some military training requirements to avoid any degradation in the post-overhaul operational readiness of the ship. These two requirements must be balanced throughout the ship overhaul process by the ship management.

On the other hand, the shipyard personnel have the responsibility to accomplish the ship overhaul as completely and accurately as possible to support the post-overhaul operational readiness of the ship, and to minimize the total ship overhaul cost.

Under the present method of ship overhauling, the ship's management prepares a ship overhaul work request package and sends it to the shipyard 120 days prior to overhaul initiation. This time will be represented as A-120 where A is the date that the ship enters the shipyard. The work requirements

given in the work request package are generally incomplete and are not descriptive enough to permit accurate planning for repairs. Therefore, at A-90, a Pre-Overhaul Test and Inspection (POT&I) is conducted jointly by the shipyard and the ship's force to determine the material condition and define the repairs required.

The results of the POT&I is the Ship's Alteration and Repair Package (SARP) which defines the overhaul work that is necessary. The SARP is generated by the shipyard and describes all identified work to be performed during ship overhaul. The SARP consists of ship alteration requirements and ship repair requirements packages. The project network, created by the shipyard scheduling office, is based on this SARP.

Each ship scheduled for overhaul has a limited overhaul budget allocated by the Ship's Type Commander (TYCOM). The allocated budget may not be sufficient for the required jobs, so some requirements may not be accepted by the shipyard and TYCOM. In fact, negotiation may continue for the first couple of months after the overhaul process has started. Consequently, any unaccepted job requirements must, if possible, be undertaken by the ship's force in addition to their initially planned jobs. Therefore, the repair portion of the overhaul package is generally not fixed (the alteration portion is firmed before overhaul initiation) until after overhaul has started; and at least initially, this makes the shipyard project network unknown to the ship management.

Traditionally, the ship project network is created after the ship has entered the shipyard, somewhat independent from the shipyard project network. So during the first couple of months after the ship overhaul has started there is no formal schedule for the ship's force at all.

In addition to the uncertainty mentioned above, there are other factors that result in conflicts between the shipyard's and ship's project networks. They are:

1. The information flow between the ship and the shipyard prior to and/or during the overhaul, which is related to the equipment configuration and material condition of the ship, is not complete.

2. The ship's management is trained by the Planning and Engineering for Repairs and Alterations (PERA) representatives on board to make the time and resource requirements estimates for each individual job request and to prepare overhaul work packages. Due to ship's operational afloat schedule, it may not be possible to have access to the ship by PERA representatives whenever needed. Therefore, the work request package possibly would be prepared in limited time, and the estimates mentioned above may not be as accurate as desired. This, in turn, affects the duration of activities in question.

3. During the overhaul process some unforeseen factors, such as illness, cause fluctuations on ship's manpower availability. Additionally, some repair material and/or tools may not be available (or may not be sufficient in amount) when needed.

4. Quite often during the overhaul process, new job requirements are encountered which were not planned for. This means new activities are to be added to the present project network.

5. Funds, manpower, facilities, and materials required to accomplish the overhaul are not determined early enough or with sufficient accuracy. Therefore, the required resources cannot be well developed in advance of overhaul. Consequently, the detailed overhaul planning is often performed late and incompletely.

6. It is difficult to develop a single Ship Systems Definition and Index to be used by fleet commands, shipyards, and other related organizations to facilitate the gathering, indexing, analyzing, and communicating of overhaul data.

This list is not complete; however, the author believes that major sources of complication are included. During the ship overhaul process the situations cited above, with possibly many others, cause many scheduling conflicts between the two project networks. Past experience has shown that scheduling conflicts occur in two ways:

1. Those conflicts due to physical environment limitations. That is two separate activities using the same work space on shipboard at the same time, and

2. Those conflicts due to insufficient communication about activity precedence relationships between the ship's activities and activities conducted by the shipyard.

In return, these conflicts induce a great amount of personnel hour loss, and have an adverse impact on ship's personnel morale.

This thesis will propose an algorithm to decrease the amount of possible conflicts by combining the ship project network consistently with the shipyard project network. Specifically, given the shipyard project network and its work space requirements on board the ship by time, the algorithm attempts to create an effective schedule for the activities of the ship project network.

Chapter 2 provides a general background on the nature of Naval Shipyards, scheduling problems faced by ship's management, the ship overhaul process as a sequence of events, and resource allocation in project networks. Chapter 3 defines the problem in more detail and discusses reasonable assumptions followed by a proposed algorithm. In Chapter 4, a FORTRAN program is presented for the proposed algorithm. Chapter 5 gives information about two sets of data used to test the algorithm. The final chapter will discuss some conclusions and recommendations that can be made from the study.

## II. BACKGROUND

### A. NAVAL SHIPYARDS

Naval shipyards are industrial activities of a Navy. These highly complex organizations support the operational readiness of the Navy's warships and make significant contributions to a national economy, especially in developing countries. The shipyard mission consists of a wide range of operations from a single routine repair operation to a complicated full ship overhaul. These requirements necessitate employment of different skilled personnel as well as a variety of tools and machines [12].

Each shipyard consists of a number of shops in which some particular class of work is performed such as machining, electrical work, pipe fitting, sheet metal work, etc. Shipyard management is responsible for conducting several repair operations at the same time. Each ship overhaul is a project that requires different levels of services from different shops. Since each shop has a limited capacity to perform any given task, the projects cannot be conducted independently from one another. The main idea is the shared utilization of the various shops without causing any conflict.

The goals of the shipyard are: 1) to accomplish ship overhaul as completely and accurately as possible to support the post-overhaul operational readiness of the ship, 2) to increase the productivity of the shipyard's work force by adopting new

advances in Organizational Management and Ship Maintenance Management, 3) to eliminate unnecessary work and support functions.

The shipyard management is responsible for preparing the SARP, conducting the POT&I, allocating available resources, obtaining the required materials, and creating the ship overhaul project network. This project network is assumed to be given in this thesis.

For a given ship, the goals of the shipyard are to minimize the total project cost subject to the limited number of various shop personnel employed on the project activities, and to meet the project due date by utilizing its own resources.

#### B. SHIP'S FORCE

The scheduling problem faced by the ship's management is to minimize the possible scheduling conflicts between the ship project network and shipyard project network, and to utilize ship's personnel as efficiently as possible, without working overtime.

Employing ship personnel on ship overhaul or training them at shore installations, during ship overhaul, depends on criteria chosen by the decision maker. The common practice is to employ some personnel on ship-created projects, and to send others to training centers. The number of personnel being trained during ship overhaul may be increased by the minimization of possible conflicts between the two project networks.

The somewhat unknown aspect of the shipyard project network to the ship's management, at the beginning of the overhaul, makes it difficult to create the ship's project network consistently with the shipyard project network. Therefore, at the beginning of the overhaul, wasting resources is inevitable. The periodic rescheduling of activities during the overhaul process does not solve the problem at all. The importance of carefully prepared initial scheduling must be realized.

#### C. SHIP OVERHAUL PROCESS

The ship overhaul process of the United States Navy can be considered in three phases as shown in Figure 2.1 below.

In project management terminology these phases correspond to management decision, project planning, and project control, respectively.

Phase I starts with the Naval Sea Systems Command (NAVSEA) proposing the Navy's Warships Overhaul Schedules (NWOS) and continues with the revision by NAVSEA and the TYCOM, and final review by Chief of Naval Operations (CNO), and might be called the "Development of NWOS."

Phase II or the planning phase essentially includes the "Preparation for Overhaul" actions as follows [1]:

1. Development of the Ship Work Package
  - a. TYCOM tasks PERA to prepare SARP.
  - b. The shipyard conducts POT&I, prepares the SARP as directed by PERA, estimates costs, and submits the ship work package to TYCOM.

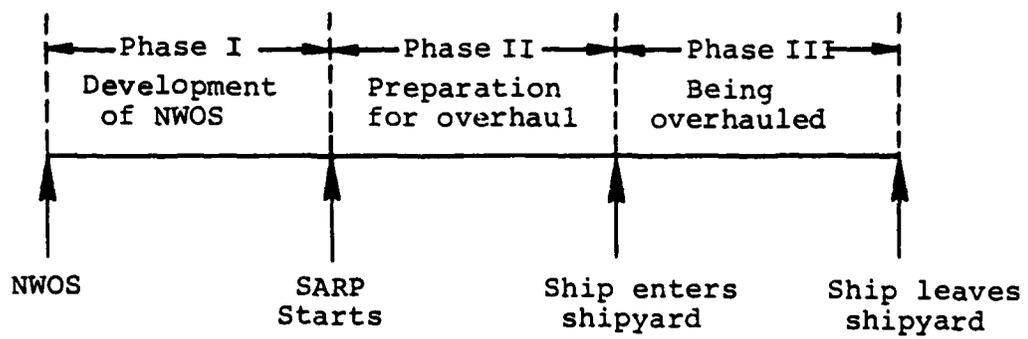


Figure 2.1

c. TYCOM screens the SARP and approves the ship work package.

d. The shipyard orders long lead time materials and prepares worker-oriented job orders.

2. Development of the Ship Alteration Package

a. The shipyard performs advance planning for specified alterations about a year in advance of overhaul.

b. The ship alteration work package is merged with the SARP.

c. The CNO funded alteration package is provided at A-180.

d. The CNO funded alteration package is modified during the prearrival conference, and the TYCOM funded alteration package is firmed.

The development of the ship work package and ship alteration package proceed along different routes, but are merged into the SARP before overhaul initiation.

Phase III is the actual ship overhaul at the shipyard and it starts when the ship enters the shipyard.

D. APPROACHES TO SCHEDULING AND RESOURCE ALLOCATION  
IN PROJECT NETWORKS

Scheduling problems in project networks vary in kind and in severity, depending upon the nature of the project and its organizational setting. In one extreme there might be just one dominant resource constraint, such as a compartment on

board ship which is so small that only one activity at a time can be performed. In this case, activities must be scheduled so that no more than one of them requiring the same environment occur at the same time. At the other extreme are complex projects requiring many resources.

Some activities in a project may compete for the same kind of limited resource simultaneously. For example, new technologies require skilled personnel as well as sophisticated tools. In addition, some kinds of resources, even if available, cannot be employed above a specified level. For example, it is possible that, only one crane can be used on a ship at a given time even if more cranes are available at the shipyard. Finally, physical or other considerations may impose limits on the employed resource level of some activities. For instance, in a ship's compartment it is not possible to employ more than a permitted amount of manpower to work on any activity in that environment.

The complexity of the scheduling problem, as pointed out above, makes it difficult to formulate and solve it as a mathematical programming problem. Linear programming and, recently, dynamic programming approaches have been attempted but only to those projects which have substantially fewer activities than those encountered in ship overhaul projects. Therefore, the general approach to such problems has been to employ heuristic programs. A heuristic (or rule of thumb) program essentially is a procedure for solving large combinatorial problems by adopting a set of rules which are believed

adequate to the problem. Heuristic programs, as discussed in detail by Wiest and Levy [15] may not lead to the optimal solution in some cases, but should generally lead to a feasible, hopefully near-optimal solution.

The existing algorithms (heuristic programs) for resource allocation in project networks take one of the following forms [15]:

1. Resource leveling programs, which try to reduce peak resource requirements and smooth out period-to-period assignments, within a constraint on project duration. These include Burgess' [2], and Wiest's [11] Unlimited Resource Leveling Procedures, and more recently Leachman's [10] Multiple Resource Leveling Procedure.

2. Resource allocation programs, which try to find the shortest project schedule by allocating available resources, such as Moder and Phillips's [9] Limited Resource Allocation Procedure.

Generalizations of the Limited Resource Allocation Procedure include Wiest's [14] SPAR-1 (Scheduling Programs for Allocating Resources), the procedures of McGee and Markarian [8], and Thesen [13] which are called RAMPS (Resource Allocation and Multi-Project Scheduling), and, finally, Davis and Heidorn [5], and Leachman [10] have developed Multiple Resource Constraints Scheduling Programs.

Recently, Holloway, Nelson, Vichit, and Suraphongschai [7] developed a heuristic project scheduling procedure which

is essentially a multi-pass procedure [6] based on problem decomposition. In their approach to the resource-constrained project scheduling, the problem is decomposed into single resource subproblems. Coordination across subproblems uses force signals created by violation of sets of constraints placed on each subproblem. The multi-pass procedure seemed to be particularly appropriate for the problem of combining two project networks with one common resource.

### III. DEVELOPMENT OF AN ALGORITHM FOR COMBINING TWO PROJECT NETWORKS

#### A. INTRODUCTION

The algorithm which will be developed in section III.D assumes that there are two project networks with one common resource requirement and that one of the project networks has been already scheduled. Specifically, it combines a ship project network with a shipyard project network, which has been already scheduled, in developing a ship overhaul project network.

The algorithm essentially is a multi-pass heuristic scheduling procedure based on problem decomposition into single resource subproblems [7].

The common constrained resource is the work space on board ship which is used by both shipyard's personnel and ship's force. The other constrained resource is the number of men in the ship's force. The ship's force is to be allocated not only to the industrial works, but also to the non-industrial works. The industrial work is defined as that work which is related to the ship overhaul activities whereas the non-industrial work is the military-oriented work, such as watch-standing and short-term training at shore installations.

The amount of resource applied to an activity is defined as the crew size assigned to that activity and the work space necessary to complete the activity. A work space might be a bulkhead, a portion of a deck or an entire compartment.

The ship project network has M activities and each activity has immediate predecessor activities; immediate successor activities; a duration; a work space in which the activity is to be performed; a maximum, minimum and normal unit resources level required each period as described by a duration-resource function; and a type characteristic, that is, splittable or non-splittable.

1. The Ship Overhaul Scheduling Problem

The scheduling problem faced by ship's management to be solved for effective ship overhaul may be summarized as follows:

Minimize Conflicts

- Subject to:
- 1) Work space availability on board ship
  - 2) Ship's crew size
  - 3) Due dates
  - 4) Activity precedence

In attempting to describe the work space constraint, it is proposed to establish a "work space availability" diagram. The work space availability diagram is a KXN matrix where the rows represent work spaces on board ship, and the columns represent overhaul periods. Thus, an element of this matrix describes the work space status as available or blocked period by period.

The ship's management must first identify the required work spaces involved in the ship and shipyard project networks. This would be followed by the blocking of those work

spaces which are required during each period by shipyard project network schedule. Then the scheduling begins on work space availability without considering ship's crew size. Finally, the crew size constraint will be incorporated. This last step is discussed in section III.D.3.

In the ship project network the activity precedence and due dates constraints are identified by determining EPST (Earliest Possible Start Time) and LPFT (Latest Possible Finish Time) for each activity from PERT calculations.

## 2. Divisible Activities

The basic idea in avoiding conflicts is to split, if possible, the activities into subactivities. Splitting activities is helpful for two reasons:

a. The divisibility of any activity is inherent to the activity nature, and splitting it makes the subactivities more homogeneous. By homogeneity it is meant that the required unit resources do not change by time. For instance, consider activity (i,j) which requires 10 days to be performed, and resources A and B. Resource A is needed, say, during the first three and last two days while resource B is needed for the whole activity duration. In this case, resource A might be a physical environment, such as a compartment on board ship in which activity (i,j) is to be started and finished while B might be the personnel doing that activity. Obviously, activity (i,j) is nonhomogeneous, since it does not need all required resources during the whole activity duration. When

activity (i,j) is split into three subactivities as shown in Figure 3.1 below, the three subactivities are homogeneous.

b. The common resource may not be available during the whole activity duration. That is, the work space required by a ship project activity on board ship may be blocked by a certain shipyard project activity for some periods. Splitting that ship project activity, if it is possible, simplifies scheduling the activity in question. For instance, consider activity (i,j) with  $t_1$  and  $t_2$  being EPST and LPFT, respectively. If, for some periods from  $t_1$  to  $t_2$ , the required work space by activity (i,j) is blocked by a shipyard project activity, then activity (i,j) must be split, if possible; otherwise it cannot be scheduled in that time interval. In Figure 3.2 below, the activity requires four periods to be scheduled. The longest available period is three units. Activity (i,j) cannot be scheduled to take advantage of these three units without splitting.

Divisibility of activities in a ship project network is feasible, since many of the activities are processed in parallel, and have large slacks. It is also feasible to delay portion(s) of some activities until some other time during the overhaul process. For example, the activity of painting a compartment could be terminated when one bulkhead is completed, and remaining bulkheads could be painted at some other time. But, splitting activities results in increased number of activities which in turn increases the problem dimensions and the computation time.

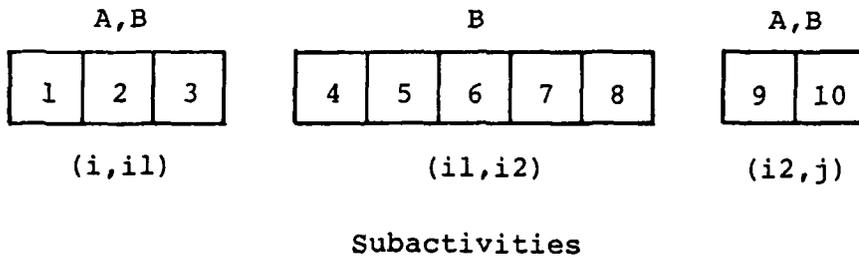
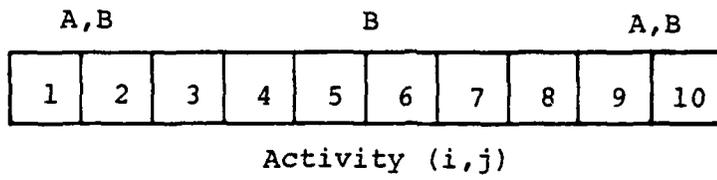
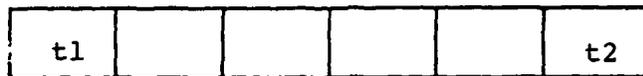
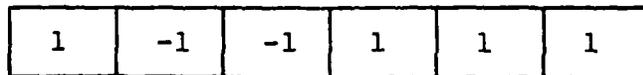


Figure 3.1



Activity (i,j)



Work space availability for activity (i,j)  
-1 : not available  
1 : available

Figure 3.2

## B. ASSUMPTIONS

In order to develop an algorithm for combining two project networks the following assumptions were made.

1. The ship and shipyard project networks are assumed to be valid networks. By valid network it is meant that the network does not contain impossible situations, such as existence of loops and redundant relationships between activities. Furthermore, it is assumed that the precedence relationships between activities of the ship project network are not unique. That is, there is possibility of changing precedence relations of networks.

2. The available manpower to be allocated to ship-created activities is limited to the available ship's force. Furthermore, the ship manpower is assumed to be homogeneous in the sense that each person can be assigned to any activity. Also, it is assumed that overtime work for the ship personnel is possible when needed.

3. The shipyard project network is assumed to be known completely at any given time; that is, the activities which are to be performed by the shipyard are already scheduled, even though it actually evolves with time. The ship project network is allowed to be updated regularly or whenever needed. For example, during the overhaul process a new job requirement may occur and necessitate updating of the ship project network.

4. The working environment (work space) on board ship such as compartments, engine room, galley, etc., are to be assumed resources which can be used by only one activity at a time. If, for instance, a device is to be removed from a particular compartment, then no other activity is permitted in that compartment until the removal activity has been completed. Furthermore, tools and materials needed for overhaul activities are assumed to be available whenever needed. Finally, each activity is assumed to be performed only in one work space.

5. The ship project network is assumed to be separable into subnetworks for managerial ease in controlling the whole project.

6. Each ship project network activity is assumed to be classifiable as splittable or non-splittable.

7. Time and resource estimates for each activity are assumed to be well defined. Some activities may be performed at other than the normal resource level. For these activities, activity duration is a linearly decreasing step-function of resources. The range of possible unit resource is described by a minimum level and a maximum level. A duration-resource function is developed over this interval. It is also assumed that the splitting of activities does not change the duration-resource function.

### C. THE CONCEPT OF INTERFACING BETWEEN NETWORKS

The ship project network naturally comprises many subnetworks. These subnetworks usually interface each other only at certain points in time. Thus, these subnetworks may be treated as gross project activities and interface points as events [3]. For management ease each subnetwork is usually made as self-contained as possible so that management control can be assigned to different officers. The interface points are controlled by top management (in the ship overhaul process, the top management is the ship commander) to avoid any conflict.

Each subnetwork is specified by its typical activities and certain "interface events." For various calculations within a particular subnetwork, these interface events are treated simply as activities of zero duration.

### D. A PROPOSED ALGORITHM

#### 1. Notation

The following notation is used throughout this chapter:

- $N$  = The total number of overhaul periods in day.  
 $R_n$  = The amount of manpower available in period  $n$ .  
 $K$  = The number of work spaces on shipboard.  
 $A$  = The set of activities in a ship project network.  
 $(i,j)$  = The activity starting at event (node)  $i$  and terminating at event  $j$ .  
 $d_{ij}$  = The duration of activity  $(i,j)$  with normal unit resource utilization.

- $\bar{d}_{ij}$  = The duration of activity (i,j) when minimum unit resources applied.
- $\underline{d}_{ij}$  = The duration of activity (i,j) when maximum unit resources applied.
- $r_{ij}$  = The normal manpower unit resource requirements by activity (i,j).
- $\bar{r}_{ij}$  = The maximum applicable manpower unit resource requirements to activity (i,j).
- $\underline{r}_{ij}$  = The minimum applicable manpower unit resource requirements to activity (i,j).
- $w_{ij}^n$  = The work space availability for activity (i,j) in period n. It is 1 if the work space is available and -1 if the work space is not available.
- $c_{ij}^n$  = The amount of manpower unit resource used by activity (i,j) in period n.
- $s_{ij}$  = The total slack for activity (i,j).
- $EST_{ij}$  = Earliest start time for activity (i,j).
- $LFT_{ij}$  = Latest finish time for activity (i,j).
- $EPST_{ij}$  = Earliest possible start time for activity (i,j).
- $LPFT_{ij}$  = Latest possible finish time for activity (i,j).
- $E_n$  = The set of activities which must all be performed by period n.
- $I_n$  = The set of activities which are in process during period n.
- $S$  = The set of scheduled activities.
- $U$  = The set of unscheduled activities.

## 2. General Description of the Algorithm

The algorithm consists of two phases with an input step (STEP 0). Each of these phases essentially corresponds to a single resource subproblem [7].

The first phase schedules the ship project network based on the common resource, which is the work space in this problem. Scheduling starts at the end of the ship project duration based on PERT and proceeds "backward" until first period of the project duration has been reached. Activities are scheduled at their EPSTs during this backward pass. The reason for using a backward pass is to insure that an activity of the ship project would be finished by its LPFT while not violating the work space requirements of the shipyard project activities.

The second phase computes the ship's manpower requirements by period. Next, a "residual" resource pool is computed, which might be defined as the difference between available amount of resources minus required amount of that resource, for each period. This residual resource pool can be either positive or negative.

Then, the algorithm searches for the period in which the residual resource pool is minimum and tries to increase that minimum level by delaying an activity, which is currently in process and has positive slack. The activity chosen is forwarded by at most its slack so that it finishes at its LPFT, unless the shifting does not cause a decrease in

previously investigated minimum level(s) further. If there is none, then that period is not to be considered any more, and no attempt will be made to decrease that minimum level further. This process continues until no eligible periods remain to be searched. This is essentially similar to the so-called "resource leveling procedure" [2], [10], [11].

A final "residual" manpower pool is computed at the end of Phase II and provides information about manpower availability for non-industrial work. In other words, it is a decision device for ship's management in two situations:

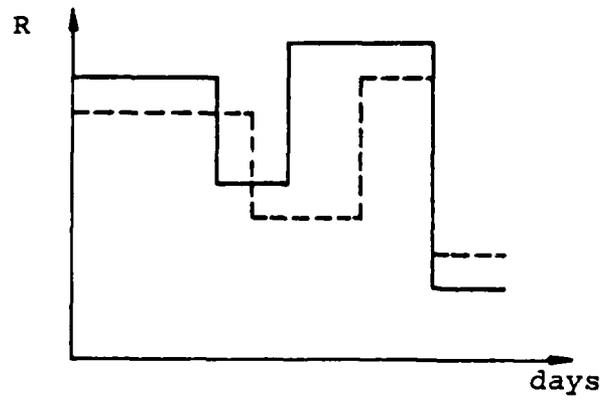
(a) A negative residual pool indicates the required amount of overtime work needed.

(b) A positive residual pool indicates the possibility of manpower availability for non-industrial work. In Figure 3.3a the solid line curve represents resource availabilities by period whereas the dotted line curve represents required resources or workload by period. The difference between these two curves is the residual resources by period and is shown in Figure 3.3b below.

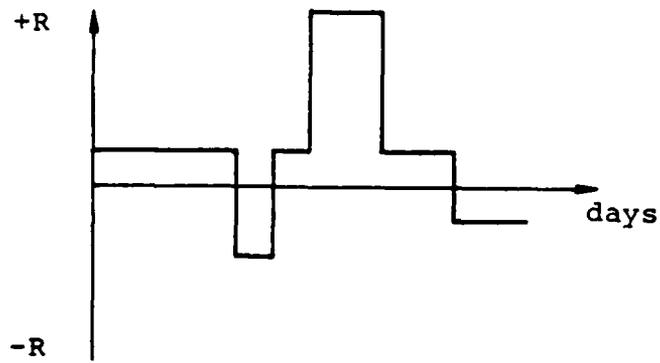
After the leveling process has completed, an attempt is made to schedule the unscheduled activities, if there are any, and the whole process restarts without STEP 0.

### 3. Algorithm

The ship overhaul scheduling procedure may be summarized as follows:



(a)



(b)

Figure 3.3

- a. Identify and list all of the activities which make up the ship project network.
- b. Determine duration-resource function for each activity.
- c. Determine precedence relationships among activities.
- d. Draw arrow diagram without any cycles, and redundancy [2].
- e. Perform PERT computations ignoring resource constraints.
- f. Perform the following algorithm to incorporate resource constraints.
- g. If there are no unscheduled activities, and the residual manpower pool is positive (or negative residual manpower, if any, can be offset by overtime work) for each period, then the problem is solved. Otherwise, the scheduling problem is not feasible.

STEP 0: Inputs to the algorithm.

1. Input parameters.
  - a. The number of overhaul periods.
  - b. The number of resources ( $=K+1$ ).
  - c. The number of activities in ship project network.
  - d. Decision for unscheduled activities. If it equals to 1, then the algorithm proceeds with scheduling of the common resource ignoring any unscheduled activity. If it

equals to 0, then the algorithm stops for revision of the precedence relations among project activities.

2. For each activity  $(i,j) \in A$ 
  - a. The ship project network PERT computations. That is, EPST, LPFT and slack.
  - b.  $d, r, \bar{r}, \underline{r}, w$ , slope of the duration-resource function, and type characteristic, that is, splittable or non-splittable. Input 1 for a splittable activity, and 0 for a non-splittable activity.
3. For each period  $n, n-1, \dots, N$ 
  - a.  $R_n$ , the amount of manpower available in period  $n$ .
  - b.  $S = \emptyset$  and  $U = \emptyset$

PHASE 1: Scheduling of the common resource.

STEP 1: Starting step.

- a.  $n=N$ , start to backward scheduling.
- b. Determine  $E_n$ , the set of activities for which  $LPFT=n$ .
- c.  $m = |E_n|$ , the number of activities in  $E_n$ .  
If  $m=0$ , go to STEP 2. Otherwise, order the activities in  $E_n$  according to the activity type. Non-splittable activities first. Within the same activity type, the activities are ordered in ascending order according to the activity slack.  
Go to STEP 3.

STEP 2: Time decrement step.

$n=n-1$ . If  $n=0$ , the scheduling is completed on common resource, go to PHASE 2. Otherwise, go to STEP 1b.

STEP 3: Scheduling on common resource.

a. Let  $l=1$ , take the first eligible activity from  $E_n$ .

b. Let  $(i,j)=E_n(l)$ , remove  $l$ th activity from  $E_n$  and identify it.

c. If  $w_{ij}^n = -1$  for some  $n \in [EPST_{ij}, LPFT_{ij}]$ , go to STEP 4.

If  $w_{ij}^n = 1, \forall n \in [EPST_{ij}, LPFT_{ij}]$ , work space is available, then activity  $(i,j)$  can be scheduled at normal resource utilization.

$t_f = EPST_{ij}$ ,  $t_f$  restricts the starting period of the activity  $(i,j)$ .

$t_l = LPFT_{ij}$ ,  $t_l$  restricts the terminating period of the activity  $(i,j)$ .

d. Update scheduled set and work space availability.

$$EST_{ij} = t_f$$

$$LFT_{ij} = t_l$$

$$S = S \cup (i,j)$$

$$w_{ij}^n = -1, \text{ blocked, and } c_{ij}^n = r_{ij}, \forall n \in [t_f, t_l]$$

e.  $l=l+1$ . If  $l > m$ , no more eligible activities to be scheduled, go to STEP 2. Otherwise, go to STEP 3b.

STEP 4: The required work space by activity (i,j) is not available for some periods in the interval of  $[EPST_{ij}, LPFT_{ij}]$

a. In this interval, let  $p'$  be the number of periods in which the work space needed by activity (i,j) is available, and  $p''$  be the largest number of consecutively available periods, i.e.,  $p'' \leq p'$ . Let  $t_f$  be the first period of this longest period and  $t_l$  be the last period of it. If  $p'' < d_{ij}$ , normal activity duration exceeds available periods, go to STEP 4b. If  $p'' \geq d_{ij}$ , available periods exceed normal activity duration, then activity (i,j) can be scheduled at normal resource utilization with its EPST and/or LPFT being changed.

b. If activity (i,j) is splittable, go to STEP 5. Otherwise, go to STEP 4c.

c. If  $p'' \geq d_{ij}$ , available periods exceed minimum activity duration, then the activity (i,j) can be scheduled by crashing it, that is, employing maximum unit resources.

$c_{ij}^n = \bar{r}_{ij}, \forall n \in [t_f, t_f + d_{ij}]$ ,  $d_{ij} = \underline{d}_{ij}$ , go to STEP 3d.

Otherwise, go to STEP 4d.

d. Find an activity which is already scheduled, it is in process during the current period, it has

positive slack, and it is performed in the same work space with the activity (i,j). If there is one, call this activity (i,j)' and delay this activity by its slack. Update the work space availability, go to STEP 3c. If there is none, the activity (i,j) cannot be scheduled, go to STEP 6.

STEP 5: Splitting the activity (i,j) into subactivities.

- a. Let  $s_p$  be the number of blocks of available periods. Split the activity (i,j) into  $s_p$  subactivities as described in section III.A.2.
- b. If  $p' < d_{ij}$ , total available periods less than normal activity duration, go to STEP 5d.

If  $p' \geq d_{ij}$ , the activity (i,j) can be scheduled at normal resource utilization by splitting it.

- c. If the required work space is available at period n, schedule the last subactivity at normal resource utilization, place the other subactivities in A, and update the duration-resource function,  $t_f = LPFT_{ij} - v$ , where v is the duration of the last subactivity.

$$t_1 = LPFT_{ij}$$

$$c_{ij}^n = r_{ij} \text{ and } w_{ij}^n = -1, \forall n \in [t_f, t_1]$$

$d_{ij} = v$ , go to STEP 3d. Otherwise, place  $s_p$  subactivities in A, go to STEP 3e.

- d. The activity (i,j) cannot be scheduled at normal resource utilization even after splitting it. If  $p' < \underline{d}_{ij}$ , total available periods less than minimum activity duration, then the activity (i,j) cannot be scheduled, go to STEP 6. Otherwise, the activity (i,j) can be scheduled at crashed level by splitting it.
- e. If the required work space is available at period n, schedule last subactivity by crashing it, place others in A.
- $$t_f = LPFT_{ij} - v$$
- $$t_l = LPFT_{ij}$$
- $$c_{ij}^n = \bar{r}_{ij}, \forall n \in [t_f, t_l]$$
- $d_{ij} = v$ , go to STEP 3d. Otherwise, place  $s_p$  subactivities in A, go to STEP 3e.

STEP 6: Activity (i,j) cannot be scheduled for the common resource with existing precedence relationships. A decision needs to be made to either

- a. Revise precedence relationship and/or remove some unimportant activities from the activity set A, and go to STEP 0, or
- b. Proceed with scheduling of the common resource with the activity (i,j) ignored. If so, go to STEP 3e.

PHASE 2: Scheduling of the manpower resource.

STEP 1: Computing resource loading.

- a. Let  $P$  be the set of periods for which the residual manpower cannot be increased further. Initially  $P$  is empty.
- b. Compute the residual manpower as  $y_n = R_n - \sum c_{ij}^n$ ,  
 $\forall (i,j) \in I_n, n=1, \dots, N$  and  $n \notin P$ .
- c. Let  $m$  be the number for which  
 $y_m = \min y_n$  for  $n=1, \dots, N$  and  $n \notin P$ .

STEP 2: Determine the set of activities which are eligible for shifting.

- a. Let  $I_m^S$  be the set of activities in  $I_m$  which have positive slack.
- b. If  $I_m^S = \emptyset$ , go to STEP 3. Otherwise  $P = P \cup m$ .  
STEP 3.
- c. If  $P$  contains whole project periods, then the solution is reached, go to STOP. Otherwise, go to STEP 1b.

STEP 3: Determine which activity is to be shifted and how much.

- a. Order the activities in  $I_m^S$  according to the slacks, in ascending order. Let  $k$  be the number of activities in  $I_m^S$ .
- b. Let  $a=1$
- c. Let  $(i,j)' = I_m^S(a)$ , the  $a$ th activity in  $I_m^S$  and let  $W$  be the number for which

$n \in [LFT_{ij}, LFT_{ij} + W]$  and  $n \notin P$ . The maximum possible value for  $W$  is the slack for activity  $(i, j)$ .

However, it may be less, because of blocked periods ( $n \in P$ ).

d. If  $W=0$ , go to STEP 3e. Otherwise, go to STEP 4.

e.  $a=a+1$ , if  $a > k$ , shifting cannot be made,  $P=P \cup m$ , go to STEP 1b. Otherwise, go to STEP 3c.

STEP 4: Shift the scheduled activity to release some uncommon resource.

a. Set  $EST_{ij}' = EST_{ij}' - W$ , and

$$LFT_{ij}' = LFT_{ij}' - W$$

b. Compute  $y_n = y_n - c_{ij}^n$ ,  $\forall n \in [EST_{ij}', EST_{ij}' + W - 1]$ , and

$$y_n = y_n + c_{ij}^n, \forall n \in [m - W + 1, m], \text{ go to}$$

STEP 1c.

#### IV. A COMPUTER PROGRAM FOR THE PROPOSED ALGORITHM

##### A. INTRODUCTION

A computer program, in FORTRAN IV, was written to implement the proposed algorithm. The program is composed of a main program and subroutines. It was originally written to combine a ship project network of up to 200 activities, requiring at most 30 work spaces for 200 or fewer overhaul periods with a shipyard project network, which is already scheduled. It can be, however, modified easily to accommodate any sized ship project network.

In the program the unit scheduling period is one day and the slope of the duration-resource function is a nonpositive integer. The integer assumption for the slope of the duration-resource function follows from the fact that the information about the entire project activities is summarized in an integer array. For more general linear duration-resource function it is necessary to define a real vector consisting of slopes of the duration-resource functions.

A portion of the inputs, EPST, LPFT and slack are obtained from a separate routine which does the PERT computations. For a moderate sized project of the order of 100 activities, these computations can be done by hand.

The subactivities, resulting from the splitting of any activity, are renumbered in the following way:

1. The subactivities are ordered in the descending order by their LPFT, and an index number is assigned to each of them.

2. For each subactivity, its index number multiplied by 1000 and added to the original activity number to produce the subactivity number.

For example, assume the activity (15,90) is split into 3 subactivities, the number for these subactivities would be (1015,1090), (2015,2090), and (3015,3090).

## B. DESCRIPTION OF THE PROGRAM

### 1. Main Program

The main program starts by calling the subroutine INPUT, which feeds the necessary inputs to the main program. Then, the activity set is ordered by LPFT of activities by calling the subroutine ORDER.

For each period, in the backward pass, the eligible activities for which LPFT equals to the current period are determined and ordered with respect to the two specified precedence criteria. They are:

- a. Activity type characteristic, and
- b. Activity slack.

Non-splittable activities precede splittable activities.

If two activities have the same type characteristic, then the one which has a smaller slack time takes precedence. For each eligible activity, the required work space is searched for the required period to figure out the work space status

by calling the subroutine WSC. Next, the four possible actions (normal, split, check back, and crash) are tried in one of the six different sequences (see section V.B) to schedule the activity in question at EPST of that activity.

When any activity is split and to be scheduled at separate time periods, the resulting subactivities are renumbered, the unscheduled subactivities are placed in the activity set, and the total slack is allocated to the last subactivity (last subactivity has the largest LPFT among the other subactivities).

The program continues by computing resource loading and residual resources. Then it searches for minimum residual periods. To increase the minimum level, an activity is selected from those activities which are in process at the current period and have positive slack. This activity is shifted forward by its slack (at most) to release some resources at peak demand period, unless the shifting does not cause to decrease previously investigated minimum level(s) further.

## 2. Subroutines

The main program calls four subroutines. They are: INPUT, ORDER, WSC, and SPLIT.

The subroutine INPUT provides the main program with necessary inputs and performs initializations.

The purpose of the subroutine ORDER is to order a given set of activities with respect to one or two specified

criterion(s). The number of criteria are transferred into the subroutine by the argument INDEX, and the specific criterion by arguments C1 (first criterion) and C2 (second criterion). C1 and C2 represent the column number from which the ordering is performed.

The subroutine WSC checks the required work space for required periods to figure out its availability. It counts available periods and/or the available number of consecutive period(s), and largest consecutively available periods.

Subroutine SPLIT determines the starting and terminating periods for each block of available periods.

### 3. Input

For combining two projects, the algorithm needs the following inputs:

- a. Project duration, from PERT computations.
- b. Number of resources ( $K+1$ , where  $K$  is the number of work spaces). Manpower was counted as resource number one.
- c. Number of activities, including dummy activities.
- d. Decision for unscheduled activities.
- e. Additionally, for each activity:
  - (1) Activity number
  - (2) EPST, LPFT and slack, from PERT computations.
  - (3) Work space number, duration, absolute value of the slope of the duration-resource function, and required unit resources, i.e., minimum, normal and maximum.

(4) Activity type, splittable or not. Input 1 for a splittable activity, and 0 for a non-splittable activity.

#### 4. Possible Actions

Four possible actions are allowed for scheduling an activity. They are:

P1 = Schedule at normal resource utilization,

P2 = Schedule by splitting,

P3 = Check previously scheduled set of activities, which use same work space with the activity in question and are in process during periods in the interval of [EPST,LPST], to facilitate the scheduling,

P4 = Schedule by crashing,

and the sequence is to be specified by the user. Six combinations are possible.

#### C. RECOMMENDATIONS ON USING THE PROGRAM

The following recommendations are provided as an aid to potential users.

1. First, one should check the validity of the assumptions, which are made in this study, to real situations.

2. The user must use more than one combination from those proposed sequences for scheduling, to obtain a better solution.

3. Once any activity is split, the resulting subactivities cannot be split any more.

4. Only one activity can be processed in one work space. If more than one activity can be processed simultaneously in

the same work space, a dummy work space having the same availability with the original one must be added to the resource availability set.

5. The procedure must be repeated whenever a new job is added to the activity set and/or some changes occur in the shipyard project.

6. The starting time of an activity is defined as the beginning of the period and the terminating time as the end of the period. So, for a dummy activity with zero duration, the EST is greater than LFT of that activity.

7. The residual pool is a decision device for excess or deficient resource periods.

## V. TESTING THE PROPOSED ALGORITHM

A FORTRAN-coded version of the algorithm was run on the IBM-360 system at W. R. Church Computer Center of the Naval Postgraduate School using two sets of data (data sets 1 and 2). For each data set, six runs were made using different sequences of four possible actions, to schedule the project in question.

### A. GENERAL

#### 1. Data

Data set 1 (from [9] p. 154)) describes a network which contains 11 activities and has a project duration of 15 days. The data were expanded to illustrate the algorithm. So, for each activity, a duration-resource function, a work space in which the activity is performed, and an activity type were defined. It was also assumed that the project was performed in three work spaces. Table I provides the details of this data set and Table II shows initial resource availabilities by period. The first row of this table shows manpower and the remaining rows show work space availabilities. The availabilities of the three work spaces were selected arbitrarily. For example, it was assumed that the work space number one (Table II second line) was blocked (-1) in the periods from 6 to 8 and from 12 to 14, arbitrarily, by two activities of the (shipyard) project network.

The headings of the columns in tables of this chapter are:

SLK = Slack.

WS = Work Space number.

DUR = Duration.

SLP = Slope.

RS = Amount of resources used.

NR = Normal applicable unit resources.

MR = Maximum applicable unit resources.

MNR = Minimum applicable unit resources.

TP = Activity type.

Data set 2 (from the Long Beach Naval Shipyard for the USS ROBISON, DDG 12) contains 51 activities and has a project duration of 21 days. This data set is actually a small portion of the ship project network. Expansions were also made for this data set. Furthermore, it was assumed that the project needs 12 work spaces. Table III summarizes this data set and Table IV shows resource availabilities by period.

#### B. SCHEDULING RESULTS

The six combinations of four possible actions (see Chapter IV for description of P1 through P4) are as follows:

1. P1, P2, P3, P4
2. P1, P2, P4, P3
3. P1, P3, P2, P4

Table I. INITIAL SET OF ACTIVITIES, DATA SET 1

ACTIVITY	EPST	LPFT	SLK	NS	DUR	SLP	NR	MR	MNR	TP	
1-	2	1	7	5	1	2	1	3	4	3	0
1-	4	1	2	0	2	2	1	6	7	6	0
1-	7	1	7	6	3	1	0	4	6	2	0
2-	3	3	11	5	1	4	1	3	4	2	1
3-	6	7	12	5	3	1	1	4	4	4	1
4-	5	3	8	1	3	5	2	2	3	2	1
4-	8	3	10	0	2	8	2	4	5	3	0
5-	6	8	12	1	2	4	1	2	4	2	0
6-	9	11	15	1	1	3	1	2	3	2	1
7-	8	2	10	6	1	3	1	5	6	4	0
8-	9	11	15	0	3	5	2	2	3	2	1

Table II. RESOURCE AVAILABILITIES, DATA SET 1

10	11	8	6	9	7	12	14	9	7	8	10	13	9	10
1	1	1	1	1	-1	-1	-1	1	1	1	-1	-1	-1	1
1	1	1	1	-1	-1	-1	1	1	1	1	1	-1	-1	1
-1	-1	1	1	1	1	1	-1	1	-1	-1	-1	1	1	1

Table III. INITIAL SET OF ACTIVITIES, DATA SET 2

ACTIVITY	EPST	LPFT	SLK	WS	DUR	SLP	NR	MR	MNR	TP
1- 18	1	14	9	1	5	1	1	2	1	1
1- 21	1	15	11	1	4	1	6	6	4	1
1- 23	1	20	16	3	4	1	1	2	1	0
1- 24	1	6	0	4	5	1	2	3	2	1
1- 27	1	9	3	5	6	0	1	1	1	1
1- 32	1	20	14	3	6	1	2	3	1	1
1- 33	1	14	11	5	3	0	2	2	1	0
1- 34	1	14	12	7	2	1	2	2	1	0
1- 35	1	14	12	7	2	1	2	3	2	1
1- 36	1	14	12	8	2	1	1	2	1	1
1- 37	1	14	11	9	3	0	2	2	2	0
1- 38	1	14	12	8	2	0	2	2	2	1
1- 39	1	14	12	10	2	1	2	3	2	1
1- 40	1	14	12	9	2	0	2	2	2	1
1- 41	1	14	11	11	3	1	2	3	2	1
1- 42	1	14	12	7	2	1	2	3	2	1
18- 19	6	17	9	1	3	1	4	5	4	0
19- 20	9	21	9	1	3	1	3	4	3	1
19- 908	9	20	12	12	0	0	0	0	0	0
20- 980	12	20	9	12	0	0	0	0	0	0

Table III. INITIAL SET OF ACTIVITIES, DATA SET 2 (continued)

21-	22	5	20	11	2	5	1	3	4	2	1
22-	912	10	20	11	12	0	0	0	0	0	1
23-	912	5	20	16	12	0	0	0	0	0	0
24-	25	6	9	0	1	4	1	3	4	2	1
25-	26	10	12	0	3	3	1	2	3	2	1
25-	908	10	20	11	12	0	0	0	0	0	0
26-	911	12	12	1	12	0	0	0	0	0	0
27-	28	7	14	3	4	5	1	3	5	2	1
27-	910	7	20	14	12	0	0	0	0	0	1
28-	29	12	20	3	4	6	2	2	3	1	1
29-	915	18	20	3	12	0	0	0	0	0	0
911-	30	13	16	0	2	4	1	1	2	1	1
30-	31	17	20	0	3	4	1	2	3	2	1
31-	914	20	20	1	12	0	0	0	0	0	0
32-	910	7	20	14	12	0	0	0	0	0	0
33-	32	4	14	11	12	0	0	0	0	0	0
34-	32	5	14	10	12	0	0	0	0	0	0
35-	32	3	14	12	12	0	0	0	0	0	0
36-	32	3	14	12	12	0	0	0	0	0	0
37-	32	4	14	11	12	0	0	0	0	0	0
38-	32	3	14	12	12	0	0	0	0	0	0
39-	32	3	14	12	12	0	0	0	0	0	0
40-	32	4	14	12	12	0	0	0	0	0	0
41-	32	4	14	11	12	0	0	0	0	0	0
42-	32	3	14	12	12	0	0	0	0	0	0

Table III. INITIAL SET OF ACTIVITIES, DATA SET 2 (Continued)

908- 999	10	20	11	12	0	0	0	0	0	0
910- 999	7	20	14	12	0	0	0	0	0	0
912- 999	10	20	11	12	0	0	0	0	0	0
914- 999	20	20	1	12	0	0	0	0	0	0
915- 999	18	20	3	12	0	0	0	0	0	0
998- 999	12	20	9	12	0	0	0	0	0	0

Table IV. RESOURCE AVAILABILITIES DAY PERIOD, DATA SET 2

20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
1	1	1	-1	-1	1	1	1	1	1	-1	1	1	1	1			
1	1	1	1	-1	-1												
-1	1	1	1	1	1	1	1	-1	-1	1	1	1	1	1			
1	1	1	1	1	1												
1	1	1	-1	1	1	1	1	1	1	1	-1	1	1	1			
-1	-1	1	1	1	1												
-1	-1	-1	1	1	1	1	1	1	1	1	-1	1	1	1			
-1	-1	1	1	1	1												
1	1	1	1	-1	-1	-1	1	1	1	1	1	1	1	1			
-1	1	1	1	1	1												
1	1	1	-1	1	1	1	1	1	-1	-1	1	1	1	1			
1	1	1	1	1	1												
1	1	1	-1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	-1	1											
1	1	1	-1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	-1	1	1	1	1	-1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1												
1	-1	1	-1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1												
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
1	1	1	1	1	1												

4. P1, P3, P4, P2

5. P1, P4, P2, P3

6. P1, P4, P3, P2

and all six were run for each data set.

Four measures of effectiveness (MOE) were used to compare the results. They are:

1. Average CPU time. For each sequence of actions three runs were made, in the morning, at noon, and at night, to obtain a relatively stable CPU time.

2. Number of unscheduled activities (NUA).

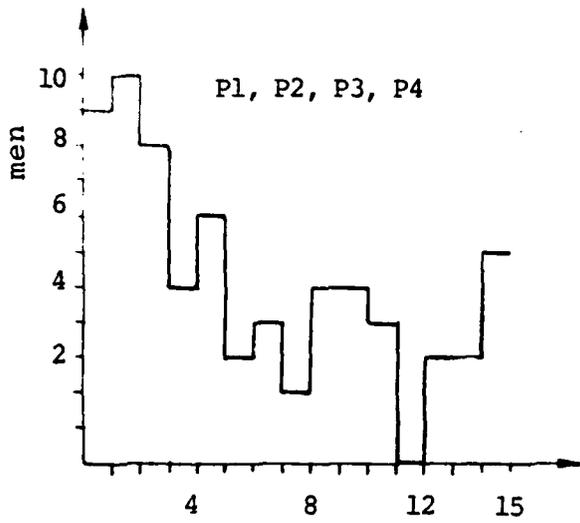
3. Sum of the squares of the resource usage deviations from the maximum resource usage for each period (SSM).

4. Sum of the squares of the resource usage deviations from the mean resource usage for each period (SSA).

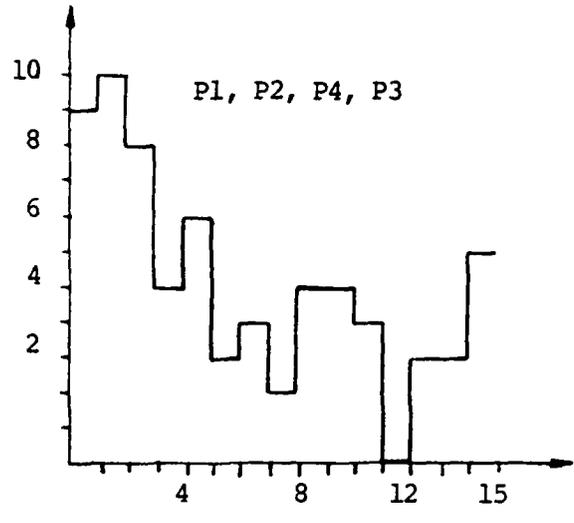
The last two MOEs, SSM and SSA, were intended to be indicators for resource usage profile and smaller numbers are preferable for these MOEs, i.e., they would be zero if the resource usage is the same for each period.

Figure 5.1 (a)-(f) for data set 1 and Figure 5.2 (a)-(f) for data set 2 show resource loading by period for each sequence of actions.

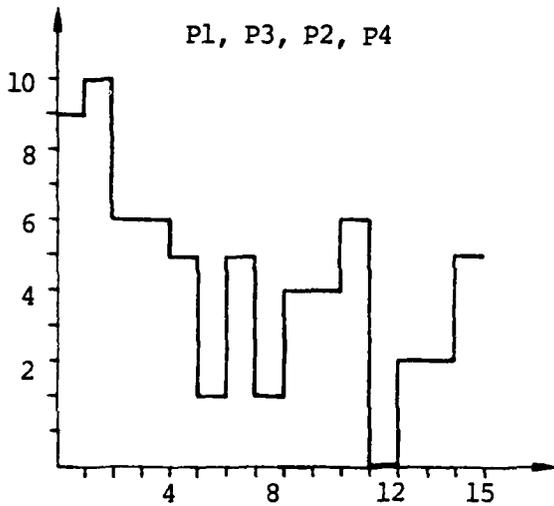
Note that the sequences of (P1, P2, P3, P4) and (P1, P2, P4, P3) result in the same usage profile for two data sets as shown in Figure 5.1(a) and (b), and 5.2(a) and (b). That is, after the sequence of normal resource utilization and splitting, the crashing or checking previously scheduled set of activities regardless of which is done, does not affect the resources requirements at all.



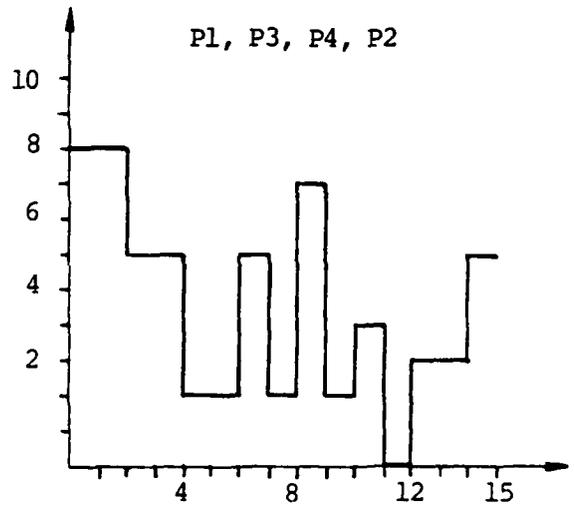
(a)



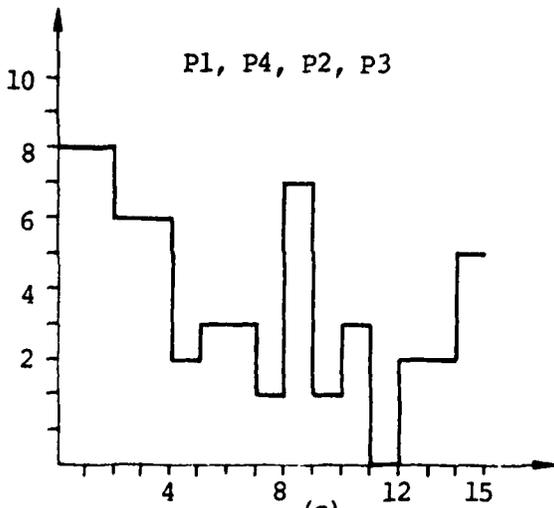
(b)



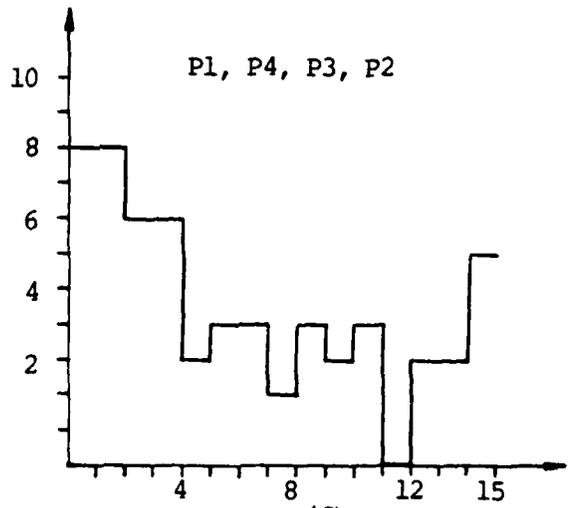
(c)



(d)

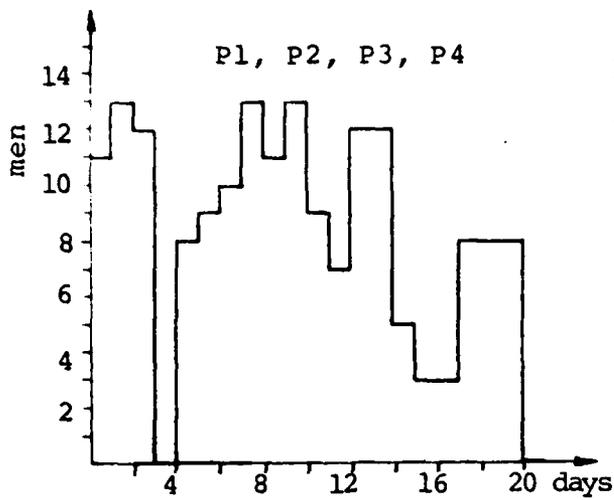


(e)

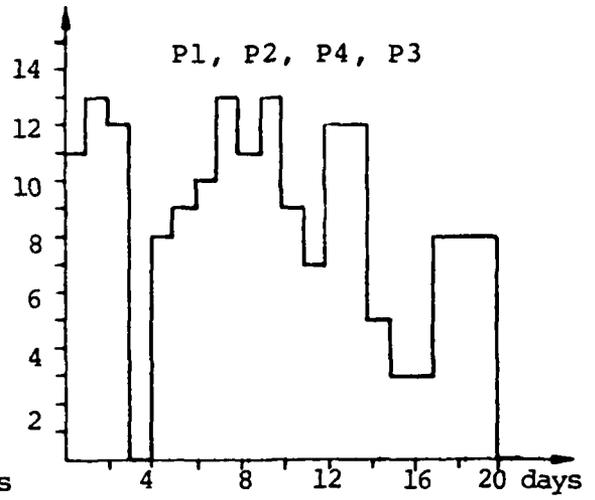


(f)

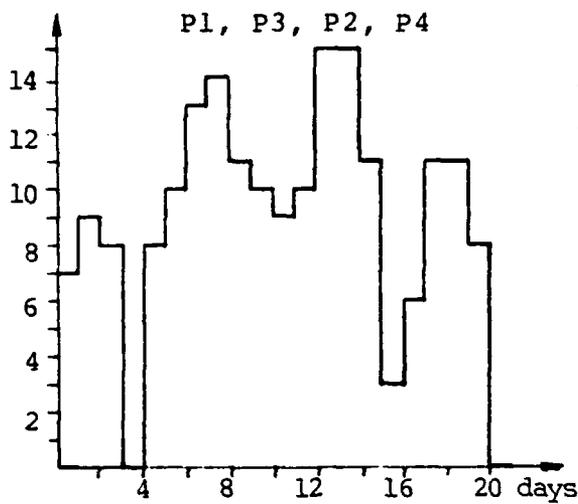
Figure 5.1



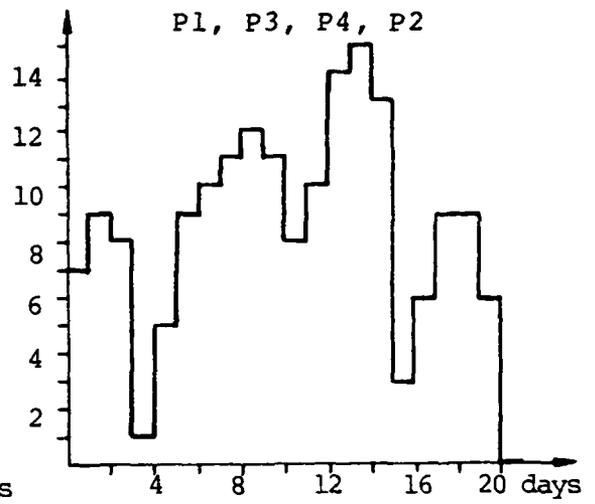
(a)



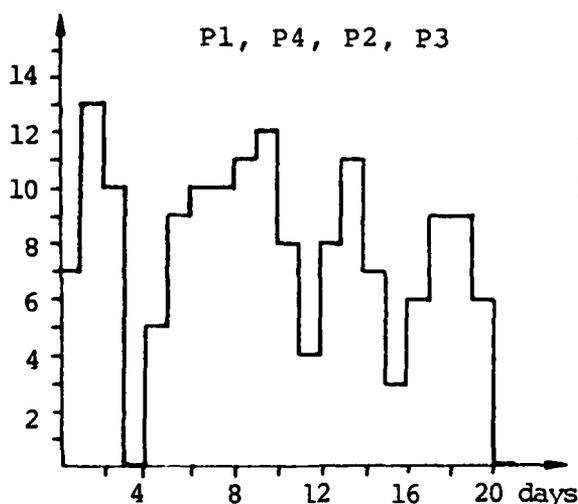
(b)



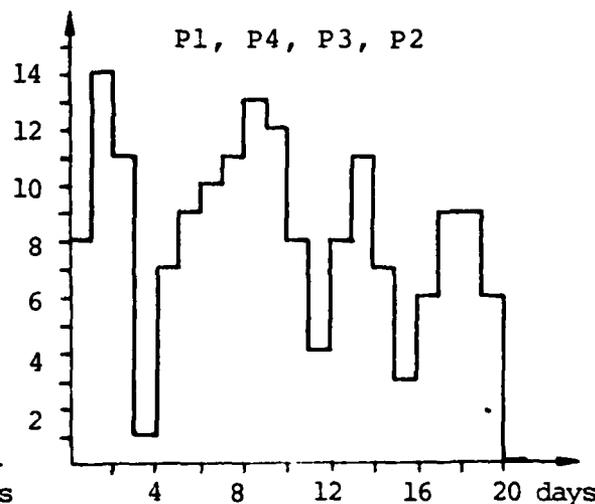
(c)



(d)



(e)



(f)

Figure 5.2

The other observation that can be made from these figures is the close relation between the uniformity of the resource usage profiles and the MOEs of SSM and SSA. That is, for smaller SSM and SSA, the resource usage profiles are more uniform (see Tables VII and VIII).

According to the values of NUA and SSA, the sequence of (P1, P4, P3, P2) appears to provide a better solution than the others. Therefore, the schedule corresponding to this sequence (sixth sequence) is provided in Tables V (a)-(b) and VI (a)-(b).

Tables V(a) and VI(a) provide the schedules for the two data sets, respectively, with the sixth sequence (P1, P4, P3, P2). The starting time of an activity is defined as the beginning of the period and the terminating time as the end of the period. Therefore, for a dummy activity with zero duration, the EST is greater than the LFT for that activity.

The activities which could not be scheduled due to work space nonavailability are shown in Tables V(b) and VI(b), respectively. They are (4,8) for data set 1, and (1,18) and (1,24) for data set 2. Therefore, with the given precedence and work space availabilities, the ship's force scheduling problem for each data set is actually not feasible.

The activity (4,8) can be scheduled by crashing it, if the work space number two would be available for periods 5 and 6. This could be accomplished by delaying the activity of (shipyard) project network, which is currently using the

Table V(a). SCHEDULED SET OF ACTIVITIES, DATA SET 1

ACTIVITY	EST	LFT	RS	WS	SLK
8-	9	13	15	3	3 0
1006-1009	15	15	3	1	0
5-	6	8	11	2	2 1
3-	6	7	7	4	3 0
2006-2009	11	11	2	1	0
2-	3	3	4	4	1 1
7-	8	9	9	6	1 1
4-	5	3	5	3	3 0
1-	2	1	2	3	1 0
1-	7	6	6	4	3 0
1-	4	1	2	6	2 0

Table V(b). UNSCHEDULED SET OF ACTIVITIES, DATA SET 1

ACTIVITY	EPST	LPFT	SLK	WS	DUR	SLP	NR	MR	MNR	TP
4	8	3	10	0	2	8	2	4	5	3 0

work space in the periods from 5 to 7, by two periods and crashing the activity (5,6). Thus, the work space number two would be scheduled as follows:

Activity (1,4) in periods 1 and 2;

Activity (4,8) in the periods from 3 to 6;

Shipyards activity in the periods from 7 to 9;

Activity (5,6) in periods 10 and 11;

and no unscheduled activities would be remained.

This example suggests the importance of continual communication and negotiation between the ship's management and the shipyard's management about unscheduled ship project activities.

Table VII for data set 1 and Table VIII for data set 2 provide a summary of the four MOEs for the six combinations of actions.

As far as the number of unscheduled activities are concerned, the sequence of (P1, P4, P3, P2) gives the least number of unscheduled activities for both data sets. Also, for this sequence, the SSA is the minimum for each data set.

Table VI(a). SCHEDULED SET OF ACTIVITIES, DATA SET 2

ACTIVITY	EST	LFT	RS	WS	SLK
19- 20	17	19	3	1	0
914- 999	20	19	0	12	0
31- 914	20	19	0	12	1
915- 999	20	19	0	12	0
29- 915	20	19	0	12	1
998- 999	12	11	0	12	0
20- 980	20	19	0	12	1
912- 999	12	11	0	12	0
908- 999	20	19	0	12	1
25- 908	20	19	0	12	1
19- 908	10	9	0	12	0
910- 999	9	8	0	12	0
32- 910	20	19	0	12	1
23- 912	7	6	0	12	0
1- 23	5	8	1	3	0
30- 31	18	20	3	3	0
28- 29	14	15	3	4	0
27- 910	20	19	0	12	1
22- 912	20	19	0	12	1
21- 22	16	20	3	2	0

Table VI(a) SCHEDULED SET OF ACTIVITIES, DATA SET 2 (Continued)

18-	19	8	10	4	1	0
911-	30	13	15	2	2	0
1001-1032	13	15	2	3	0	
1-	21	2	3	6	1	0
34-	32	15	14	0	12	0
41-	32	4	3	0	12	1
37-	32	12	11	0	12	1
33-	32	12	11	0	12	1
1-	37	1	3	2	9	0
1-	33	12	14	2	5	0
42-	32	4	3	0	12	0
40-	32	12	11	0	12	1
39-	32	12	11	0	12	1
38-	32	12	11	0	12	1
36-	32	12	11	0	12	1
35-	32	12	11	0	12	1
1-	34	5	6	2	7	0
27-	28	7	11	3	4	0
1-	42	1	2	2	7	1
1-	41	5	7	2	11	7
1-	40	4	5	2	9	8
1-	39	13	14	2	10	0
1-	38	5	6	2	8	8
1-	36	1	2	1	8	1
1-	35	7	8	2	7	6

Table VI(a) SCHEDULED SET OF ACTIVITIES, DATA SET 2 (Continued)

26- 911	12	11	0	12	1
25- 26	10	11	3	3	0
2001-2032	9	9	3	3	0
24- 25	6	7	4	1	0
1001-1027	6	9	1	5	0
2001-2027	1	4	1	5	0
3001-3032	1	3	2	3	0

Table VI(b) UNSCHEDULED SET OF ACTIVITIES, DATA SET 2

ACTIVITY	EPST	LPFT	SLK	WS	DUR	SLP	NR	MR	MNR	TP
1- 24	1	6	0	4	5	1	2	3	2	1
1- 18	1	14	9	1	5	1	1	2	1	1

Table VII. DATA SET 1

Sequence of Actions	Average CPU Time	NUA	SSM	SSA
P1, P2, P3, P4	27.04	1	646	130
P1, P2, P4, P3	26.44	1	646	130
P1, P3, P2, P4	27.44	2	607	125
P1, P3, P4, P2	26.96	2	411	112
P1, P4, P2, P3	28.35	1	362	107
P1, P4, P3, P2	27.17	1	826	92

Table VIII. DATA SET 2

Sequence of Actions	Average CPU Time	NUA	SSM	SSA
P1, P2, P3, P4	34.45	3	794	333
P1, P2, P4, P3	30.39	3	794	333
P1, P3, P2, P4	35.06	2-1/2*	877	346
P1, P3, P4, P2	32.81	2	1210	309
P1, P4, P2, P3	30.55	3	718	257
P1, P4, P3, P2	33.37	2	1007	250

\* 1/2 represents one unscheduled subactivity.

## VI. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

### A. SUMMARY AND CONCLUSIONS

1. The algorithm assumes two project networks with one common resource requirement and that one of the project networks has been already scheduled. The problem that this algorithm was designed for was the scheduling of a ship's force during a shipyard overhaul. Specifically, it combines a ship project network with a shipyard project network, which has been already scheduled, in developing a total ship overhaul project network. Two types of constraints are allowed. First one is work space on board ship which is common to both ship's and shipyard's project networks, and second one is the ship's manpower. It was assumed that resources, such as shipyard's materials, tools and devices which are needed for ship overhaul, are available whenever needed.

The algorithm consists of two phases. Each of these phases essentially corresponds to a single resource subproblem. The first phase schedules the ship project network based on the common resource, which is the work space on board ship. Scheduling starts at the end of the ship project duration based on PERT and proceeds backward until first period of project duration has been reached. The second phase computes the ship's manpower requirements by period, and then computes a "residual" manpower pool. Then, the algorithm searches for the period in which the residual resource pool

is minimum and tries to increase that minimum by delaying an eligible activity forward. In other words, it tries to level out peak demands. A final "residual" pool is computed at the end of phase II and provides information about manpower availability for non-industrial work.

2. The algorithm was tested with two sets of data and the sequence combination involving first normal resource utilization, then a crashing, then checking for available common resource from previously scheduled set of activities, and finally splitting appears to be the most promising with respect to the four MOEs.

3. Although the proposed algorithm is a heuristic solution procedure, it gives a solution range through the use of different sequences of possible actions for scheduling an activity.

4. The algorithm does not consider any variable unit resource requirements. That is, the unit resource requirements for any activity or subactivity have been assumed constant throughout the activity or subactivity duration. This deficiency may be corrected by dividing the activity, which requires variable resources, into subactivities each requiring constant unit resources.

## B. RECOMMENDATIONS

1. It is recommended that further study be made done of this algorithm to develop extensions which allow relaxation of

the assumptions of homogeneity of ship's manpower, availability of shipyard's materials and tools which are needed for ship overhaul, and variable unit resource requirements, and the inclusion of more than one uncommon and/or common resource requirement.

2. The solving of an actual ship overhaul scheduling problem is far beyond the scope of this thesis. It needs to be studied in more detail by at least a team consisting of systems analysts, systems engineers, and human factors engineers. It is an extremely complex resource-constrained allocation problem. This complexity is compounded by communication problems between ship and shipyard prior to and/or during the ship overhaul process, the many uncertainties concerning availability of resources, and the organizational difficulties between the involved commands.

3. The reason for using a "backward" pass in the algorithm to schedule the common resource, work spaces, was to insure that an activity of the ship project would be finished by its LPFT while not violating the work space requirements of the shipyard project activities. That is, the starting and terminating periods of an activity of the shipyard project were assumed given and unchangeable.

Although, the work space availabilities of the two examples (see Chapter V) were arbitrary, the manpower loading profiles (see Figures 5.1 and 5.2) and unscheduled activities obtained from those examples resemble anticipated real-world

situations. Therefore, the ship's management would need to negotiate with the shipyard's management to extend the project duration or to provide more availability of work spaces so that a more uniform manpower loading profile could be obtained and the list of unscheduled activities in the ship project network could be eliminated.

APPENDIX A  
VERBAL FLOWCHART

A. MAIN PROGRAM

- 1- 3 : Comment cards.
- 4- 7 : Variable declaration statements.
- 8- 10 : Comment cards.
- 11- 16 : Call subroutine INPUT and perform initialization.
- 17- 22 : Start backward scheduling, print current period.
- 23- 39 : Order the activities according to their LPFT.
- 40- 44 : Write the ordered set of activities.
- 45- 59 : Determine eligible activities which have LPFT  
equal to the current period.
- 60- 62 : Comment cards.
- 63- 76 : Order eligible activities according to activity  
splittability and slack.
- 77- 80 : Print ordered eligible activity set.
- 81- 87 : Remove an activity from eligible activity set  
and identify it.
- 88- 90 : Comment cards.
- 91 : Determine common resource availability.
- 92-140 : Check scheduling possibilities. That is, normal  
resource utilization, splitting, crashing or  
checking previously scheduled set of activities.
- 141-142 : Comment cards.
- 144 : Decision for unscheduled activities.

- 145-150 : Place unscheduled activity in the set of unscheduled activities.
- 151-160 : Schedule the activity at normal resource utilization.
- 161-169 : Schedule the activity by crashing.
- 170-187 : Place the scheduled activity in the set of scheduled activities.
- 188-208 : Try to schedule the activity by splitting.
- 209-221 : Schedule the last subactivity at normal resource utilization.
- 222-234 : Schedule the last subactivity by crashing.
- 235-255 : Place unscheduled subactivities in the set of activities.
- 256-263 ; Allocate the slack to the first subactivity.
- 264-274 : Start to schedule using the manpower resource.
- 275-287 : Compute excess resource availability for manpower.
- 288-298 : Locate the peak manpower usage period and try to reduce.
- 299-331 : Specify which activity is to shift and how much.
- 332-338 : Shift the activity to later on the schedule.
- 339-351 : Recompute the excess resource availability for manpower.
- 352-353 : Check for second pass.
- 354-360 : Place unscheduled activities in the activity set and try to schedule with a second pass.
- 361-374 : Output.

375-400 : Format statements.

401-402 : STOP and END.

## B. SUBROUTINES

### 1. Subroutine INPUT

1 : Subroutine INPUT.

2- 5 : Variable declaration statements.

6- 10 : Read and write the parameters.

11- 23 : Read and write activity set.

24- 35 : Read and write resource availabilities.

36- 53 : Initialize the scheduled activity set and some  
auxiliary arrays.

54- 63 : Initialize the unscheduled activity set.

64- 72 : Format statements.

73- 74 : RETURN and END.

### 2. Subroutine ORDER

1 : Subroutine ORDER

2- 5 : Variable declaration statements.

6- 23 : Bubble sorting to order on first criterion.

24- 41 : Bubble sorting to order on second criterion.

42- 43 : RETURN and END.

### 3. Subroutine WSC

1 : Subroutine WSC

2- 5 : Variable declaration statements.

6- 18 : Determine common resource availability periods.

19- 30 : Determine the blocks of available periods.

31- 37 : Determine the starting and terminating periods  
for the longest block of available periods.

38- 39 : Dummy statements.

40- 41 : RETURN and END.

4. Subroutine SPLIT

1 : Subroutine SPLIT

2- 5 : Variable declaration statements.

6- 24 : Determine starting and terminating periods of  
subactivities.

25- 26 : RETURN and END.

APPENDIX B  
DESCRIPTION OF THE DATA CARDS

1. Parameter Card: Card # 1

Columns	Explanation
1-3	Number of project periods.
4-6	Number of resources.
7-9	Number of activities.
10	Decision for unscheduled activities. For precedence revision input 0, otherwise input 1.

2. Activity Identification Cards: For the following M cards, where M is the number of activities.

Columns	Explanation
1-3	Activity starting node.
4-6	Activity terminating node.
7-9	EPST of the activity.
10-12	LPFT of the activity.
13-14	Slack for the activity.
15-17	Work space number of the activity.
18-20	Activity duration.
21-23	Slope of the time-resource function (absolute value).
24-26	Normal unit resources required by the activity.
27-29	Maximum unit resources required by the activity.

30-32 : Minimum unit resources required by the activity.

33 : Activity type characteristic. Input 1 for splittable activities; input 0 for non-splittable activities.

3. Manpower Resource Availability Cards: Each card contains information for a sequence of 15 periods. For example, three cards would be needed for a project of 31 to 45 periods in duration.

Columns	Explanation
---------	-------------

1-3	Manpower resource availability for 1 st period.
...	...
43-45	Manpower resource availability for 15 th period.

4. Common Resource Availability Cards: Each card contains information for a sequence of 15 periods. A new card for the following each 15 periods.

Columns	Explanation
---------	-------------

1-3	Common resource availability for 1 st period.
...	...
43-45	Common resource availability for 15 th period.

COMPUTER PROGRAM

COMBINING TWO PROJECT NETWORKS

```

1  C   PROGRAM TO COMBINE TWO PROJECT NETWORK
2  C   WITH ONE COMMON RESOURCE AND ONE UNCOMMON RESOURCE
3  C   AND ONE OF THE PROJECT NETWORK IS ALREADY SCHEDULED.
4  C   IMPLICIT INTEGER=2 (A-Z)
5  C   DIMENSION A(200,12),R(31,200),S(300,7),E(30,12),TEMP(12),U(50,12),
6  C   IY(200),PS(200),V(50),T(9,2),VT(300,12)
7  C   COMMON N,K,IA,DECIDE,A,R,S,E,Y,U,PS,TEMP,V,T,VT
8  C
9  C   CC PHASE ONE: SCHEDULING ON COMMON RESOURCE.
10 C
11 C   PASS=1
12 C   NSR=0
13 C   CALL INPUT
14 C   39 NSAI=NSA+1
15 C   NUA=0
16 C   ORDR=0
17 C   DO 22 DAY=1,N
18 C
19 C   CC START TO BACKWARD SCHEDULING.
20 C
21 C   IN=N-DAY+1
22 C   WRITE(6,602) IN
23 C   IF (ORDR.EQ.1) GO TO 400
24 C   IF (IA.EQ.1) GO TO 400
25 C   ORDR=1
26 C   IA=IA
27 C   JC=12
28 C   C1=0
29 C   C2=4
30 C   INDX
31 C   DO 401 I=1,IA
32 C   DO 401 J=1,JC
33 C   401 VT(I,J)=A(I,J)
34 C   402 CONTINUE
35 C   CALL ORDER (IA,JC,C1,C2,INDX)
36 C   DO 403 I=1,IA
37 C   DO 403 J=1,JC
38 C   403 A(I,J)=VT(I,J)
39 C   404 CONTINUE
40 C   IF (DAY.GE.2) GO TO 400
41 C   WRITE(6,605)
42 C   WRITE(6,604)
43 C   DO 410 I=1,IA
44 C   410 WRITE(6,200) (A(I,J),J=1,12)
45 C   400 M=0
46 C
47 C   CC DETERMINE ELIGIBLE ACTIVITY SET.
48 C
49 C   DO 2 I=1,30
50 C   JN=IA-I+1

```

```

51         IF (A(JN,4).NE.IN) GO TO 405
52         M=M+1
53         DO 1 J=1,12
54         E(1,J)=A(JN,J)
55         1 A(JN,J)=0
56         2 CONTINUE
57     405 IF (M.EQ.0) GO TO 22
58         IA=IA-M
59         IF (M.LE.1) GO TO 221
60     CC
61     CC ORDER ELIGIBLE ACTIVITY SET ACCORDING TO SPECIFIED CRITERION.
62     CC
63         IA=M
64         JC=12
65         C1=12
66         C2=5
67         INDX=2
68         DO 407 I=1,IA
69         DO 406 J=1,JC
70     406 VT(I,J)=E(I,J)
71     407 CONTINUE
72         CALL ORDER (IA,JC,C1,C2,INDX)
73         DO 409 I=1,IA
74         DO 408 J=1,JC
75     408 E(I,J)=VT(I,J)
76     409 CONTINUE
77         WRITE (6,606)
78         WRITE (6,604)
79     221 DO 223 I=1,M
80     223 WRITE (6,200) (E(I,J),J=1,12)
81         INDEX=0
82         DO 21 L=1,M
83         FC=0
84         DLB=E(L,7)-E(L,8)*(E(L,10)-E(L,11))
85         F=E(L,3)
86         KK=E(L,6)+1
87         DD=IN-F+1
88     CC
89     CC DETERMINE COMMON RESOURCE AVAILABILITY
90     CC
91     210 CALL WSC (F,IN,DD,KK,PIP,SP,TF,TL)
92         IF (PIP.EQ.0) GO TO 227
93         IF (PIP.GE.DD) GO TO 6
94         CALL SPLIT (DD,SP)
95         DO 224 I=1,SP
96         DO 225 J=1,2
97     225 T(I,J)=F+T(I,J)-1
98     224 CONTINUE
99         P2P=0
100        DO 3 I=1,SP

```

```

101      IF (P2P.GE. (T(1,2)-T(1,1)+1)) GO TO 3
102      P2P=T(1,2)-T(1,1)+1
103      TF=T(1,1)
104      TL=T(1,2)
105      3 CONTINUE
106      IF (P2P.GE.E(L,7)) GO TO 6
107      IF (P2P.GE.DLB) GO TO 7
108      IF (FC.GE.1) GO TO 216
109      DO 215 I=1,NSA
110      IF (E(L,6).NE.S(I,6)) GO TO 215
111      IF (S(I,3).GT.E(L,4)) GO TO 215
112      IF ((P2P+S(I,7)).GE.DLB) GO TO 228
113      215 CONTINUE
114      216 IF (E(L,12).EQ.1) GO TO 10
115      GO TO 4
116      CC
117      CC SCAN PREVIOUSLY SCHEDULED SET OF ACTIVITIES
118      CC TO FACILITATE THE SCHEDULING.
119      CC
120      227 DO 226 I=1,NSA
121      IF (E(L,6).NE.S(I,6)) GO TO 226
122      IF (S(I,3).GT.E(L,4)) GO TO 226
123      IF (S(I,7).GE.DLB) GO TO 228
124      226 CONTINUE
125      GO TO 4
126      228 IF (E(L,4).LE.S(I,4)) GO TO 211
127      IF ((E(L,4)-S(I,4)).GE.DLB) GO TO 213
128      GO TO 211
129      213 M1=S(I,4)+1
130      M2=M1+S(I,7)-1
131      212 S(I,7)=0
132      DO 229 I=M1,M2
133      229 A(KK,I)=1
134      FC=FC+1
135      GO TO 210
136      211 S(I,3)=S(I,3)+S(I,7)
137      S(I,4)=S(I,4)+S(I,7)
138      M2=S(I,3)-1
139      M1=M2-S(I,7)+1
140      GO TO 212
141      CC
142      CC DECISION FOR UNSCHEDULED ACTIVITY.
143      CC
144      4 IF (DECIDE.EQ.0) GO TO 32
145      WRITE (6,608) E(L,1),E(L,2)
146      NUA=NUA+1
147      DO 5 J=1,12
148      U(NUA,J)=E(L,J)
149      5 E(L,J)=0
150      GO TO 21

```

```

151 CC
152 CC SCHEDULE ACTIVITY AT NORMAL RESOURCE UTILIZATION.
153 CC
154 6 NSA=NSA+1
155 EL1=E(L,1)
156 EL2=E(L,2)
157 C=E(L,9)
158 D=E(L,7)
159 WRITE(6,610) EL1,EL2
160 GO TO 8
161 CC
162 CC SCHEDULE ACTIVITY AT CRUSHED LEVEL.
163 CC
164 7 NSA=NSA+1
165 EL1=E(L,1)
166 EL2=E(L,2)
167 C=E(L,10)
168 D=DLB
169 WRITE(6,611) E(L,1),E(L,2)
170 CC
171 CC PLACE THE SCHEDULED ACTIVITY IN THE SCHEDULED SET.
172 CC
173 8 EL6=E(L,6)+1
174 DO 9 IP=TF,TL
175 9 A(EL6,IP)=-1
176 S(NSA,1)=EL1
177 S(NSA,2)=EL2
178 S(NSA,3)=TF
179 S(NSA,4)=TF+D-1
180 S(NSA,5)=C
181 S(NSA,6)=KK-1
182 S(NSA,7)=TL-TF-D+1
183 WRITE(6,700) NSA
184 IF(INDEX.EQ.1) GO TO 15
185 DO 61 I=1,12
186 61 E(L,I)=0
187 GO TO 21
188 CC
189 CC TRY TO SCHEDULE THE ACTIVITY BY SPLITTING.
190 CC
191 10 IF(SP.LE.1) GO TO 111
192 WRITE(6,601) E(L,1),E(L,2),SP
193 WRITE(6,202)
194 WRITE(6,206)
195 DO 110 I=1,SP
196 110 WRITE(6,203) (T(I,J),J=1,2)
197 SP)=1
198 GO TO 112
199 111 SP)=0
200 112 IF(PIP.GE.E(L,7)) GO TO 11

```

```

201      IF (PIP.GE.DLB) GO TO 12
202      GO TO 4
203      11 IF (V(DD).EQ.1) GO TO 13
204      INDEX=0
205      GO TO 15
206      12 IF (V(DD).EQ.1) GO TO 14
207      INDEX=0
208      GO TO 15
209      CC
210      CC SCHEDULE FIRST SUBACTIVITY AT NORMAL RESOURCE UTILIZATION.
211      CC
212      13 NSA=NSA+1
213      EL1=E(L,1)+1000*SP1
214      EL2=E(L,2)+1000*SP1
215      TF=T(SP,1)
216      TL=T(SP,2)
217      C=E(L,9)
218      D=TL-TF+1
219      INDEX=1
220      WRITE(6,610) EL1,EL2
221      GO TO 8
222      CC
223      CC SCHEDULE FIRST SUBACTIVITY AT CRUSHED LEVEL.
224      CC
225      14 NSA=NSA+1
226      EL1=E(L,1)+1000*SP1
227      EL2=E(L,2)+1000*SP1
228      TF=T(SP,1)
229      TL=T(SP,2)
230      C=E(L,10)
231      D=TL-TF+1
232      INDEX=1
233      WRITE(6,611) EL1,EL2
234      GO TO 8
235      CC
236      CC PLACE UNSCHEDULED SUBACTIVITIES IN THE ACTIVITY SET.
237      CC
238      15 SSP=SP-INDEX
239      ORDR=0
240      IA1=IA+1
241      IA=IA+SSP
242      TSLACK=PIP-D
243      DO 20 I=IA1,IA
244      A(I,1)=E(L,1)+1000*(I-IA1+1+INDEX)
245      A(I,2)=E(L,2)+1000*(I-IA1+1+INDEX)
246      A(I,3)=T(IA-I+1,1)
247      A(I,4)=T(IA-I+1,2)
248      A(I,5)=0
249      A(I,6)=E(L,6)
250      A(I,7)=T(IA-I+1,2)-T(IA-I+1,1)+1

```

```

251         A(I,8)=E(L,8)
252         DO 19 J=9,11
253     19   A(I,J)=E(L,J)
254         A(I,12)=0
255     20   CONTINUE
256     CC
257     CC  ALLOCATE THE SLACK TO LAST SUBACTIVITY.
258     CC
259         IF (TSLACK.LE.0) TSLACK=0
260         A(IA,5)=A(IA,5)+TSLACK
261         IF (TSLACK.LE.0) GO TO 21
262     21   CONTINUE
263     22   CONTINUE
264         IF (NSA.LT.NSA1) GO TO 33
265         IF (PASS.GE.2) GO TO 23
266         DO 30 I=1,N
267     30   Y(I)=A(I,1)
268     CC
269     CC  PHASE TWO: SCHEDULING ON UNCOMMON RESOURCE.
270     CC
271         N1=0
272         WRITE(6,42)
273         DO 231 I=1,K
274     231  WRITE(6,204) (A(I,J),J=1,N)
275     CC
276     CC  COMPUTE EXCESS RESOURCE AVAILABILITY FOR UNCOMMON RESOURCE.
277     CC
278     23  DO 25 I=1,N
279         IF (PS(I).EQ.1) GO TO 25
280         DO 24 J=NSA1,NSA
281         IF (S(J,3).EQ.1) GO TO 241
282         GO TO 24
283     241  IJ=S(J,4)
284         DO 242 L=1,IJ
285     242  Y(L)=Y(L)-S(J,5)
286     24   CONTINUE
287     25   CONTINUE
288     CC
289     CC  LOCATE PEAK DEMAND PERIOD FOR UNCOMMON RESOURCE
290     CC  AND TRY TO SMOOTH OUT.
291     CC
292     26  MIN=Y(I)
293         M=1
294         DO 27 I=2,N
295         IF ((PS(I).EQ.1).OR. (Y(I).GE.MIN)) GO TO 27
296         MIN=Y(I)
297         M=I
298     27  CONTINUE
299     CC
300     CC  SPECIFY WHICH ACTIVITY IS TO SHIFT AND HOW MUCH ?

```

```

301 CC
302 DO 282 I=1,NSA
303 VT(I,1)=S(I,7)
304 282 VT(I,2)=1
305 IR=NSA
306 JC=2
307 C1=0
308 C2=1
309 INDX=1
310 CALL ORDER (IR,JC,C1,C2,INDX)
311 W=0
312 DO 283 J=1,NSA
313 IF (VT(I,1).LE.0) GO TO 283
314 IF ((S(VT(I,2),3).EQ.M1.OR.(S(VT(I,2),3).EQ.(M-VT(I,1)+1))) GO TO 3
315 150
316 GO TO 283
317 350 W=VT(I,1)
318 M1=S(VT(I,2),4)+1
319 M2=M1+W-1
320 DO 284 J=M1,M2
321 IF (PS(J).EQ.0) GO TO 284
322 IB=J-1
323 IF (IB.LT.M1) GO TO 283
324 GO TO 270
325 284 CONTINUE
326 GO TO 285
327 283 CONTINUE
328 PS(M)=1
329 N1=N1+1
330 IF (N1.GE.(N+1)) GO TO 36
331 GO TO 26
332 CC
333 CC SHIFT THE ACTIVITY BY W TO FORWARD.
334 CC
335 270 W=IB-M1+1
336 285 S(VT(I,2),3)=S(VT(I,2),3)+W
337 S(VT(I,2),4)=S(VT(I,2),4)+W
338 S(VT(I,2),7)=S(VT(I,2),7)-W
339 CC
340 CC RECOMPUTE THE EXCESS RESOURCE AVAILABILITY FOR UNCOMMON RESOURCE
341 CC
342 M1=M-W+1
343 M2=M1+W-1
344 M4=S(VT(I,2),4)
345 M3=M4-W+1
346 DO 286 J=M1,M2
347 R(S(VT(I,2),6),J)=1
348 286 Y(J)=Y(J)+S(VT(I,2),5)
349 DO 287 I=M3,M4
350 287 Y(J)=Y(J)-S(VT(I,2),5)

```

```

351      GO TO 26
352      36 PASS=PASS+1
353      IF ((NUA.EQ.0).OR.(PASS.GE.3)) GO TO 33
354      DO 38 I=1,NUA
355      DO 37 J=1,12
356      A(I,J)=U(I,J)
357      37 U(I,J)=0
358      38 CONTINUE
359      IA=NUA
360      GO TO 39
361      33 WRITE(6,702)
362      WRITE(6,701)
363      DO 34 I=1,NSA
364      34 WRITE(6,300) (S(I,J),J=1,7)
365      WRITE(6,703)
366      WRITE(6,604)
367      DO 35 I=1,NUA
368      35 WRITE(6,200) (U(I,J),J=1,12)
369      WRITE(6,42)
370      WRITE(6,204) (Y(J),J=1,N)
371      DO 198 I=2,K
372      198 WRITE(6,204) (R(I,J),J=1,N)
373      GO TO 999
374      32 WRITE(6,201)
375      42 FORMAT(//,2X,'RESOURCE AVAILIBILITIES BY PERIOD.',/)
376      200 FORMAT(2X,14,'-',14,2(2X,13),2X,12,3(2X,13),3(2X,12),2X,11,/)
377      201 FORMAT(//,2X,'REVISE PRECEDENCE RELATIONS AND/OR REMOVE SOME ACTIV
378      *ITIES')
379      202 FORMAT(/,2X,'SPLITTED ACTIVITY')
380      203 FORMAT(/,2(2X,14))
381      204 FORMAT(//,15(2X,12))
382      206 FORMAT(2X,15,2X,15)
383      300 FORMAT(/,2X,14,'-',14,2(2X,13),2X,12,2X,13,2X,12)
384      601 FORMAT(//,2X,'ACTIVITY ',14,'-',14,' IS SPLITTED TO ',11,' SUBACTI
385      IVITIES',//,2X,'EPST',2X,'LPFT')
386      602 FORMAT(/,2X,'PERIOD= ',13,/)
387      603 FORMAT(//,2X,'UNORDERED INITIAL SET OF ACTIVITIES.',/)
388      604 FORMAT(//,3X,'ACTIVITY',1X,'EPST',1X,'LPFT',1X,'SLK',3X,'WS',2X,'
389      IDUR',2X,'SLP',2X,'NR',2X,'MR',1X,'MNR',1X,'TP')
390      605 FORMAT(//,2X,'ORDERED SET OF ACTIVITIES.',/)
391      606 FORMAT(//,2X,'ORDERED SET OF ELIGIBLE ACTIVITIES.',/)
392      608 FORMAT(//,2X,'ACTIVITY ',14,'-',14,' CANNOT BE SCHEDULED')
393      610 FORMAT(//,2X,'ACTIVITY ',14,'-',14,' CAN BE SCHEDULED AT NORMAL R
394      ESOURCE UTILIZATION')
395      611 FORMAT(//,2X,'ACTIVITY',14,'-',14,' CAN BE SCHEDULED BY CRUSHING'
396      )
397      700 FORMAT(//,2X,'SCHEDULED ',12,' TH ACTIVITY',/)
398      701 FORMAT(4X,'ACTIVITY',2X,'EST',2X,'EFT',2X,'RS',2X,' WS',2X,'SLK')
399      702 FORMAT(//,2X,'SCHEDULED SET OF ACTIVITIES.',/)
400      703 FORMAT(//,2X,'UNSCHEDULED SET OF ACTIVITIES.',/)

```

401 999 STOP  
402 END

```

1      SUBROUTINE INPUT
2      IMPLICIT INTEGER*2 (A-Z)
3      DIMENSION A(200,12),R(31,200),S(300,7),E(30,12),TEMP(12),U(50,12),
4      1Y(200),PS(200),V(50),T(9,2),VT(300,12)
5      COMMON N,K,IA,DECIDE,A,R,S,E,Y,U,PS,TEMP,V,T,VT
6      CC
7      CC READ AND WRITE PARAMETERS.
8      CC
9      READ(5,100) N,K,IA,DECIDE
10     WRITE(6,500) N,K,IA,DECIDE
11     CC
12     CC READ AND WRITE ACTIVITY SET.
13     CC
14     WRITE(6,603)
15     WRITE(6,604)
16     DO 1 I=1,IA
17     READ(5,101) (A(I,J),J=1,12)
18     1 WRITE(6,200) (A(I,J),J=1,12)
19     IA1=IA+1
20     DO 3 I=IA1,200
21     DO 2 J=1,12
22     2 A(I,J)=0
23     3 CONTINUE
24     CC
25     CC READ RESOURCE AVAILABILITIES.
26     CC
27     WRITE(6,42)
28     DO 4 I=1,K
29     READ(5,102) (R(I,J),J=1,N)
30     4 WRITE(6,102) (R(I,J),J=1,N)
31     K1=K+1
32     DO 6 I=K1,31
33     DO 5 J=1,200
34     5 R(I,J)=0
35     6 CONTINUE
36     CC
37     CC INITIALIZE THE SCHEDULED SET AND SOME AUXILIARY ARRAY.
38     CC
39     DO 8 I=1,300
40     DO 7 J=1,7
41     7 S(I,J)=0
42     DO 45 J=1,12
43     45 VT(I,J)=0
44     8 CONTINUE
45     DO 17 I=1,50
46     17 V(I)=0
47     DO 12 I=1,200
48     Y(I)=0
49     12 PS(I)=0
50     DO 19 I=1,9

```

```

51      DO 18 J=1,2
52      18 T(I,J)=0
53      19 CONTINUE
54      CC
55      CC  INITIALIZE THE UNSCHEDULED SET.
56      CC
57      DO 11 J=1,12
58      TEMP(J)=0
59      DO 9 I=1,30
60      9 E(I,J)=0
61      DO 10 L=1,50
62      10 U(L,J)=0
63      11 CONTINUE
64      42 FORMAT (//,2X,'RESOURCE AVAILIBILITIES BY PERIOD.',/)
65      100 FORMAT (3I3,I1)
66      101 FORMAT (4I3,12,6I3,I1)
67      102 FORMAT (15I3)
68      200 FORMAT (2X,I4,'-',14,2(2X,I3),2X,12,3(2X,I3),3(2X,I2),2X,I1,/)
69      500 FORMAT (2X,'NUMBER OF PERIODS  =',I3,/,2X,'NUMBER OF RESOURCES =',
70      112,/,2X,'NUMBER OF ACTIVITIES=',I3,/,2X,'DECISION',12X,'=',I1,/)
71      604 FORMAT (///,3X,'ACTIVITY',1X,'EPST',1X,'LPFT',1X,'SLK',3X,'WS',2X,'
72      10UR',2X,'SLP',2X,'NR',2X,'MR',1X,'MNR',1X,'TP',/)
73      RETURN
74      END

```

```

1      SUBROUTINE ORDER (IR, JC, C1, C2, INDEX)
2      IMPLICIT INTEGER*2 (A-Z)
3      DIMENSION A(200, 12), R(31, 200), S(300, 7), E(30, 12), TEMP(12), U(50, 12),
4      IY(200), PS(200), V(50), T(9, 2), VT(300, 12)
5      COMMON N, K, IA, DECIDE, A, R, S, E, Y, U, PS, TEMP, V, T, VT
6      DO 1 I=1, JC
7          1 TEMP(I)=0
8      CC
9      CC BUBBLE SORTING ON FIRST CRITERION.
10     CC
11         L=0
12         M=IR-1
13         IF (INDEX.EQ.1) C1=C2
14         2 DO 4 I=1, M
15             IP1=I+1
16             IF (VT(I, C1) .LE. VT(IP1, C1)) GO TO 4
17             L=L+1
18             DO 3 J=1, JC
19                 TEMP(J)=VT(I, J)
20                 VT(I, J)=VT(IP1, J)
21                 VT(IP1, J)=TEMP(J)
22         3 CONTINUE
23         4 CONTINUE
24         IF ((L.EQ.0) .OR. (M.EQ.1)) GO TO 5
25         L=0
26         M=M-1
27         GO TO 2
28         5 IF (INDEX.EQ.1) GO TO 9
29     CC
30     CC BUBBLE SORTING ON SECOND CRITERION.
31     CC
32         L=0
33         M=M-1
34         6 DO 8 I=1, M
35             IP1=I+1
36             IF (VT(I, C1) .NE. VT(IP1, C1)) GO TO 8
37             IF (VT(I, C2) .LE. VT(IP1, C2)) GO TO 8
38             L=L+1
39             DO 7 J=1, JC
40                 TEMP(J)=VT(I, J)
41                 VT(I, J)=VT(IP1, J)
42                 VT(IP1, J)=TEMP(J)
43         7 CONTINUE
44         8 CONTINUE
45         IF ((L.EQ.0) .OR. (M.EQ.1)) GO TO 9
46         L=0
47         M=M-1
48         GO TO 6
49         9 RETURN
50     END

```

```

1      SUBROUTINE WSC (F,L,D,KN,PIP,SP,TF,TL)
2      IMPLICIT INTEGER*2 (A-Z)
3      DIMENSION A(200,12),R(31,200),S(300,7),E(30,12),TEMP(12),U(50,12),
4      IY(200),PS(200),V(50),T(9,2),VT(300,12)
5      COMMON N,K,IA,DECIDE,A,R,S,E,Y,U,PS,TEMP,V,T,VT
6      CC
7      CC DETERMINIG COMMON RESOURCE AVAILBILTY PERIODS.
8      CC
9      DO 1 I=1,D
10     1   V(I)=-1
11         PIP=0
12     DO 2 I=F,L
13         IF(R(KN,I).EQ.-1) GO TO 2
14         IF1=I-F+1
15         V(IF1)=1
16         PIP=PIP+1
17     2   CONTINUE
18         IF(PIP.EQ.0) GO TO 6
19     CC
20     CC ACTIVITY IS TO SPLITTED IF THE LARGEST COMMON
21     CC RESOURCE AVAILABILITY PERIOD IS NOT SUFFICIENT.
22     CC
23         SP=0
24         SS=-1
25     DO 3 I=1,D
26         IF(V(I).EQ.SS) GO TO 3
27         SS=-SS
28         IF(V(I).LT.0) GO TO 3
29         SP=SP+1
30     3   CONTINUE
31         IF(SP.GE.2) GO TO 6
32     DO 4 I=1,D
33         IF(V(I).GT.0) GO TO 5
34     4   CONTINUE
35     5   TF=F+I-1
36         TL=TF+PIP-1
37         GO TO 7
38     6   TF=0
39         TL=0
40     7   RETURN
41     END

```

```

1      SUBROUTINE SPLIT (D,SP)
2      IMPLICIT INTEGER*2 (A-Z)
3      DIMENSION A(200,12),R(31,200),S(300,7),E(30,12),TEMP(12),U(50,12),
4      IY(200),PS(200),V(50),T(9,2),VT(300,12)
5      COMMON N,K,IA,DECIDE,A,R,S,E,Y,U,PS,TEMP,V,T,VT
6      CC
7      CC DETERMINE START AND FINISH PERIODS OF SUBACTIVITIES.
8      CC
9          I1=1
10         DO 5 I=1,SP
11         DO 1 J=I1,0
12         IF (V(J).GT.0) GO TO 2
13         1  CONTINUE
14         2  T(I,1)=J
15         J1=J+1
16         DO 3 L=J1,0
17         IF (V(L).LT.0) GO TO 4
18         3  CONTINUE
19         IF (L.GE.0) GO TO 6
20         4  T(I,2)=L-1
21         I1=L+1
22         5  CONTINUE
23         GO TO 7
24         6  T(I,2)=0
25         7  RETURN
26         END

```

## LIST OF REFERENCES

1. ARINC Research Corp. Report 1634-01-1-1460 Computerized Programs for Improving Overhaul Planning, by D. M. McLaughlin, November, 1975.
2. Burgess, A. R. and Killebrew, J. B., "Variation in Activity Level on a Cyclic Arrow Diagram," Journal of Industrial Engineering, Vol. 13, No. 2, pp. 76-83, March-April 1962.
3. Burman, P. J., Precedence Networks for Project Planning and Control, McGraw Hill, 1972.
4. Davis, E. W., "Resource Allocation in Project Networks-- A Survey," Journal of Industrial Engineering, Vol. 17, No. 4, pp. 177-188, April, 1966.
5. Davis, E. W. and Heidorn, G. E., "An Algorithm for Optimal Project Scheduling Under Multiple Constraints," Management Science, Vol. 17, No. 12, pp. B803-B816, August, 1971.
6. Hollaway, C. A. and Nelson, R. T., "Alternative Formulation of the Job Shop Problem with Due Dates," Management Science, Vol. 20, No. 1, pp. 75-75, September, 1973.
7. Hollaway, C. A., Nelson, R. T., and Suraphongschai, V., "Comparison of a Multi-Pass Heuristic Decomposition Procedure with Other Resource-Constrained Project Scheduling Procedures," Management Science, Vol. 25, No. 9, pp. 862-872, September, 1979.
8. McGee, A. A., and Markarian, M. D., "Optimum Allocation of Research Engineering Manpower within a Multi-Project Organizational Structure," IEEE Transactions on Engineering Management, Vol. 9, No. 9, pp. 104-108, September, 1962.
9. Moder, J. J. and Phillips, C. R., Project Management with CPM and PERT, 2d Ed., Reinhold Publishing Co., 1970.
10. Leachman, R. C., "Multiple Resource Leveling in Construction Systems Through Variation of Activity Intensities," Paper presented at the Naval Postgraduate School, Monterey, 14 July 1980.
11. Levy, F. K., Thompson, G. L., and Wiest, J. D., "Multi-ship, Multishop, Workload Smoothing Programming," Naval Research Logistics Quarterly, Vol. 9, No. 1, pp. 37-44, March, 1962.

12. Shackelton, N. J., "Minimizing the Cost of Projects in Naval Shipyards," Ph.D. Dissertation, Naval Postgraduate School, Monterey, 1973.
13. Thesen, A., "Heuristic Scheduling of Activities Under Resource and Precedence Restrictions," Management Science, Vol. 23, No. 4, pp. 412-422, December, 1976.
14. Wiest, J. D., "A Heuristic Model for Scheduling Large Projects with Limited Resources," Management Science, Vol. 13, No. 6, pp. B359-B376, February, 1967.
15. Wiest, J. D. and Levy, F. K., A Management Guide to PERT/CPM, Prentice-Hall, 1969.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4. Lt. E. F. Roland, USN, Code 55Ro Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
5. Lt. Osman Karadağ, Turkish Navy Cayirbasi Cad. No: 257 Buyukdere, ISTANBUL TURKEY	2
6. Assoc. Prof. A. W. McMasters, Code 55Ms Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
7. Deniz Kuvvetleri Komutanlığı Eğitim Daire Başkanlığı Ankara TURKEY	2
8. Deniz Harp Okulu Komutanlığı Heybeliada, ISTANBUL TURKEY	1
9. Golcuk Tershane Komutanlığı Golcuk, KOCAELI TURKEY	1
10. Orta Doğu Teknik Üniversitesi Sistem Bilimleri Arastirma Enstitusu Bolum Baskanlığı Ankara TURKEY	1

11. Charles E. Sprenger, Code 377  
Long Beach Naval Shipyard  
Long Beach, California 90802

1

**NU  
DATE  
FILME**