ETM
ENGINEERING & TECHNOLOGY MANAGEMENT

Title: Comparison of Different Models in Multiple Project Scheduling and Resource Allocation

Course:
Year: 1992
Author(s): T. Daim, N. Hendarsin, S. McDermant, L. Poh and C. P. Saravanabhava

Report No: P92026

Abstract: Multiproject scheduling and resource allocation methods are reviewed. Optimization models and heuristics are compared. Earlier comparisons reported in the literature are summarized. The use of heuristics is illustrated using a 3-project example with 23, 40 and 20 activities in the projects. The "Total Float Rule" is reported to have the best overall performance except for the cases where resources are loosely constrained. The report recommends performing several heuristics simultaneously in order to hedge against adopting poor schedules.

COMPARISON OF DIFFERENT MODELS IN MULTIPLE PROJECT SCHEDULING AND RESOURCE ALLOCATION

Tugrul Daim, Niko Hendarsin, Sean MacDermant,
Leong Poh, C.P. Saravanabhava

EMP-P9226

COMPARISON OF DIFFERENT MODELS IN MULTIPLE PROJECT SCHEDULING AND RESOURCE ALLOCATION

TEAM MEMBERS:

Tugrul Daim
Niko K. Hendarsin
Sean MacDermant
Leong Poh
C.P. Saravanabhava

PREPARED FOR:

Dr. Dundar F. Kocauglu
Emgt 545, Spring 1992

PORTLAND STATE UNIVERSITY

EXECUTIVE SUMMARY

Multi-project systems are fairly common. Almost every individual or company is involved in many projects. Scheduling and allocating resources to multiple projects is more complicated than for the single-project. From the study, we know that there are two ways to treat the scheduling and resource allocation of multiple projects. One way is to treat the several projects as if they were each elements of a single large project. Another way is to consider that all projects are completely independent. Although these two ways lead to different scheduling and allocation outcomes, the conceptual basis for scheduling and allocating resources is the same. Development of an efficient and dynamic multi-project scheduling system is a vital task to project manager. And there are three important parameters to measure scheduling effectiveness. These are: schedule slippage, resource utilization and in-process inventory.

There are various types of multiproject scheduling and resource allocation techniques. However, these techniques are categorized into two methods, which are optimization techniques, generally referred to as mathematical programming and heuristic methods. Mathematical programming can be used to obtain optimal solutions to certain types of multiproject scheduling problems. These procedures determine when an activity should be scheduled in specific periods. Heuristic methods are more realistic approaches in attacking the resource-constrained multiproject scheduling problems. They essentially use simple priority rules, such as shortest task first, to determine which task should receive resources, and which task must wait. This may identify feasible solutions to the problem.

There is great value in solving a problem that is of the same type as your businesses' typical problem using optimization techniques. Benchmarking the results of heuristics as a percentage of the optimal solution may be the most valuable use of mathematical programming techniques as applied to the multi-project scheduling problem. While developing heuristic rules for models, it is important to note that the model outcome produces a good feasible solution, not an optimal solution. Two basic approaches for formulating the sequencing rules mentioned in our literature research are:

1. Davis and Patterson [3] treated multiple project as one single big project. Heuristic sequencing rules was based upon the objective of minimizing total delay time with equal penalty.
2. Kurtulus and Narula [4] treated multiple projects as aggregation individual of projects. The sequencing rules was based upon the objective of minimizing of the total delay time with unequal penalty.

In Davis and Patterson's study, the comparisons concluded that MINSLK was the best heuristic. This heuristic had previously been found to be the most effective by Mize, Fendly, and Patterson in a multi-project scheduling problem. It was concluded that network structure and resource requirements may affect which heuristic is really the most effective. Thus the best heuristic is dependent upon the characteristics of a specific project problem.

Since all the studies used different multi-project scheduling problems, no one scheduling method can be deemed universally best. It is a mistake to assume that one heuristic will always provide you with the best schedule. It is recommended that several heuristics be performed simultaneously to hedge against adopting poor schedules. This should be part of the regular scheduling process. Since most heuristics can be run in a short time, this is a minor inconvenience relative to the value it provides. Taking a little extra time developing the schedule can help you avoid putting all your eggs in one heuristic's basket. Experience with the results of different heuristics will eventually provide insight into the specific instances when each heuristic outperforms the others. Insights should be used cautiously, and it is always recommended to run two or more heuristics on any scheduling problem.

TABLE OF CONTENTS

PAGE

I. INTRODUCTION

What Is A Project?

The American Heritage Dictionary of English language defines a project as "an undertaking requiring concentrated effort." [8] In plain English, a project is defined as a job with a single objective.

A project involves a single, definable end result usually measured in terms of cost, schedule, and performance requirements over a specified life cycle. for example, constructing a house, writing an article, making a meal, etc. are projects.

What Is Project Management?

Project management is concerned with planning, scheduling, and controlling non-routine activities within certain time and resource constraints. In simple terms, project management is defined as planning, controlling and managing people.

Multi-project System

A multi-project system or rather a multiple-project system exists when an individual or an organization is simultaneously working on two or more projects. One example could be a building contractor who is simultaneously working on many different projects worldwide. An organization such as construction company, simultaneously undertaking multiple projects of different sizes at different sites, has to satisfy the following two often conflicting objectives:

1. Completion of all the projects to optimally achieve the overall objectives of the organization in terms of value, time, and cost.

2. Maintenance of an adequate level of resources to support all the projects and maximum utilization of scarce resources.

The management, while executing the projects, is interested in attaining the following goals:

1. To minimize the total duration of the projects.
2. To maximize utilization of scarce resources from the availability level throughout the project duration.
3. To minimize of the total cost of executing the projects.

Project Scheduling Problems

Based on the number of projects and the amount of resources required by the projects, problems are classified into four broad categories:

1. Single project single resource.
2. Single project multiple resource.
3. Multiple projects single resource.
4. Multiple projects multiple resources.

Multi-project Scheduling And Resource Allocation

Multi-project systems are fairly common. Almost every individual or company is involved in many projects. Scheduling and allocating resources to multiple projects is more complicated than for the single-project.

From the study, we know that there are two ways to treat the scheduling and resource allocation of multiple projects. One way is to treat the several projects as if they were each elements of a single large project. Another way is to consider that all projects are completely independent. Although these two ways lead to different scheduling and allocation outcomes, the conceptual basis for scheduling and allocating resources is the same.

When a project manager is faced with several projects, he or she needs to be reminded that each project has its own set of activities, due dates, and resource requirements. The manager also needs to acknowledge that each project has its own penalties for not meeting time, cost, and performance goals. Therefore, it is not hard to realize that the complicated multi-project problem also involves determining how to allocate resources to and setting a completion time for, a new project that is added to an existing set of ongoing projects.

Development of an efficient and dynamic multi-project scheduling system is a vital task to project manager. And there are three important parameters to measure scheduling effectiveness. These are: schedule slippage, resource utilization and in-process inventory.

Schedule Slippage

Schedule slippage is defined as the time past a project's due date when the project is completed. It is the most important criteria in project management as slippage may result in penalty costs and that reduce profit. In addition, slippage of one project may disturb the overall organization as the ripple effect causes other projects to slip as well.

Resource Utilization

Resource utilization is a measure of effectiveness as all the projects in a multi-project organization are competing for the same scarce resources. It is ideal to attain a smooth resource allocation system and thus effective resource utilization is of particular concern to industrial firms.

Work-In-Process Inventory

Work-in-process inventory is related to the amount of work waiting to be processed as there is a shortage of some resources. It is measured in term of total project delay and total resource idle time. If an industrial organization has a large investment in in-process inventory, it may indicate that there is a lack of efficiency. The remedy involves a trade-off between the cost of in-process inventory and the cost of the resources, usually capital equipment, needed to reduce the in-process inventory levels.

A good multi-project scheduling and resource allocation system is to optimize the three criteria. However, these criteria cannot be optimized at the same time. As it always happens in real world, trade-offs are involved. A firm must decide which criterion is most applicable in any given situation, and then use that criterion to evaluate its various scheduling and resource allocation options.

There are various types of multi-project scheduling and resource allocation techniques. In the next section, we look into two common approaches.

II. MAJOR APPROACHES

There are various types of multiproject scheduling and resource allocation techniques. However, these techniques are categorized into two methods, which are optimization techniques, generally referred to as mathematical programming and heuristic methods.

Mathematical programming can be used to obtain optimal solutions to certain types of multiproject scheduling problems. These procedures determine when an activity should be scheduled in specific periods. The three common objectives are there:

1. minimum total throughput time for all projects.
2. minimum total completion time for all projects.
3. minimum total lateness or lateness penalty for all projects

On the other hand, heuristic methods are used to avoid the difficulties with analytical formulation of realistic problems, especially in large, resource-constrained multiple project scheduling. Heuristic methods are more realistic approaches in attacking the resource-constrained multiproject scheduling problems. They essentially use simple priority rules, such as shortest task first, to determine which task should receive resources, and which task must wait. This may identify feasible solutions to the problem.

A major task in heuristic scheduling is to develop activity sequencing rules that support widely prevalent and realistic project situations. In other words, heuristic procedures for resource-constrained multiproject scheduling represent the only practical means for finding workable solutions to the large, complex multiproject problems normally found in everyday life.

MATHEMATICAL PROGRAMMING EXAMPLE MODEL

(IGP)

The example of mathematical programming is taken from Mohanty and Siddiq's paper [1]. An integer goal programming model is formulated here. The approach begins with the translation of management goals into explicit objective functions and various decision-making restrictions into explicit constraints. Two kinds of constraints are normally faced in the Project Scheduling problem. first, there are technological restrictions on the order in which the activities can be performed. A feasible solution is defined as that solution which can satisfy these constraints. An optimal solution will find answer to the following questions:

1. When will each activity be performed?
2. What resources will be allocated to perform that activity?

Assumptions:

The following assumptions are made to formulate the model:

1. All activity durations are integers.
2. Looping and dangling of activities are not allowed.
3. Activities once started cannot be interrupted (splitting of activities is not permitted).
4. The amount of different resources required by activity remains constant throughout its duration.
5. Amount of each resource available per period is constant and known.

Notations used:

i Project index; $i=1,\ldots,I$, I = total number of projects

j activity index; $j=1,\ldots,N_i$, N_i = number of activities in i-th project

? How about t_j?

t time index

k resource type index: k=1,...,K

K total number of resources

T_{ai} absolute due date for the i-th project

T_{ei} earliest possible date by which the i-th project could be completed

r_{ijkt} amount of resource type k required by the activity j in time period t for i-th project

R_{kt} amount of resource type k available in the time period t

S_{ij} allowable slack for activity j of the i-th project

X_{ijt} a zero-one decision variable that identifies the time frame t, in which activity j of the i-th project can be assigned. The number of variables to define a specific activity are equal to (1 + allowable slack).

X_{ijt} = 1, if the j-th activity of the i-th project is assigned

= 0, otherwise

X_{it} a zero-one decision variable identifying the completion of all the activities of a particular project i

X_{it} = 1, if all the activities of project i are completed

= 0, otherwise

Model inputs:

The initial inputs that are to be supplied to the model are as follow:

1. Number of projects I.
2. A finite set of activities N_i for the i-th project.
3. For each activity $j \in N_i$, the duration d_{ij} and the number of resource units r_{ijk} needed

to complete this activity.

4. A set P of precedence relationships indicating activity j-1 that has to be completed before activity j can be started.
5. The minimal duration of the project.
6. For each activity $j \in N_i$; EST (j) = early start time of activity j, and LST (j) = late start time of activity j.

Model constraints:

Activity completion constraints

Each activity can select only one time-frame for its occurrence, i.e.

$$\sum_{t=1}^{1+S_{ij}} X_{ijt} = 1 \quad \text{for all j}$$

The number of such constraints are equal to the number of activities of all the projects.

Precedence relationship constraints

These constraints are required, when a particular activity cannot be started until one or more other activities have been completed, e.g. for an i-th project, let j be the proceeding activity and j' be the following activity then the precedence relationship constraint can be written as:

$$\sum_{t_j} t_j X_{ijt} - \sum_{t_{j'}} t_{j'} X_{ij'} \quad t \geq 0$$

Define

The number of such constraints ~~are~~ is equal to the number of precedence-constrained activities.

Project completion constraints

The X_{it} variable for each project will be zero until all of its activities have been completed, i.e. and i-th project cannot be considered to be complete until

$$\sum_{t=1}^{1+S_{ij}} X_{ijt}=1, \text{ for all } N_i \text{ activities of the i-th project.}$$

The constraint can be written as,

$$X_{it} \leq (1/N_i) \sum_{j=1}^{N_i} \sum_{t} X_{ijt}$$

Such constraints are required for each time period and for each type of resource and can be state as stated

$$\sum_{i=1}^{I} \sum_{j=1}^{N_i} r_{ijkt}X_{ijt}+d_k^- -d_k^+ = R_{kt}$$

where d = underutilization of the k-th resources, and d = overutilization of the k-th resources.

Lateness minimization constraints

A project is considered to be late, if it is completed after certain desired due data period T_{di}, where $T_{ei} \leq T_{di} \leq T_{ai}$. Minimization of project lateness constraint can be written as

$$\sum_{i=1}^{I} \sum_{t=Td_{i+1}}^{\text{Max } T_{Ai}} (1-X_{it})+d_i^- - d_i^+ = 0$$

where d_i = deviational variable with respect to project i . Let P_{it} be some penalty associated with the lateness of the i-th project through certain time period, then as alternative constraint can be defined as the minimization of lateness penalty given as

I max T_{Ai}

$$\sum_{i=1}^{I} \sum_{t=T_{di+1}}^{max\, T_{Ai}} P_{it}(1-X_{it})+d_i^- - d_i^+ = \beta$$

where β = some value.

Goals

The different goals that can be identified are as follows:

1. Schedule only l cranes,
2. Schedule no more than m carpenters,
3. Schedule no more than n laborers,
4. Minimize lateness.

Where l, m, n correspond to the resource availability level of cranes, carpenters and laborers, respectively.

Performance measures

The following measures are defined:

1. Project slippage

   It has always been the prime duty of the project manager to avoid Project Slippage since:

   - It results in the late deliveries to clients incurring penalty costs and loss of good will.
   - May disturb the whole schedule especially in the case of multiple projects. If refers to the departure of a project past its CPM calculated finish time and is defined by Project slippage = T_s - T_o where T_s = extended

duration of the project under resource constrained situation, and T_o = duration of the project as computed by CPM.

2. Resource constrained scheduling efficiency

   This can be defined as $E = 1-(T_s-T_o)/T_o$

3. Total project delay

   For a given set of projects, total project delay is given as the sum (over all the projects) of the difference between the assigned scheduled finish time of a project and the duration of the critical path in an early start schedule. This measure gives an indication of the delays introduced as a result of limitations on resource availability and as a result of the scheduling rule employed. This can be defined as

$$\sum_{t=1}^{I} (T_{si}-T_{oi})$$

   where T_{si} = extended duration of the i-th project under resource constrained situation.

4. Weighted total delay

   For a given set of projects the weighted total delay is defined as the sum (over all the projects) of the total resources demanded by a project multiplied by the total delay of the project.

$$\sum_{i=1}^{I} \sum_{k=1}^{K} W_w R_{kt}(T_{si}-T_{oi})$$

   where, W_w = the weightage given to the different resource type.

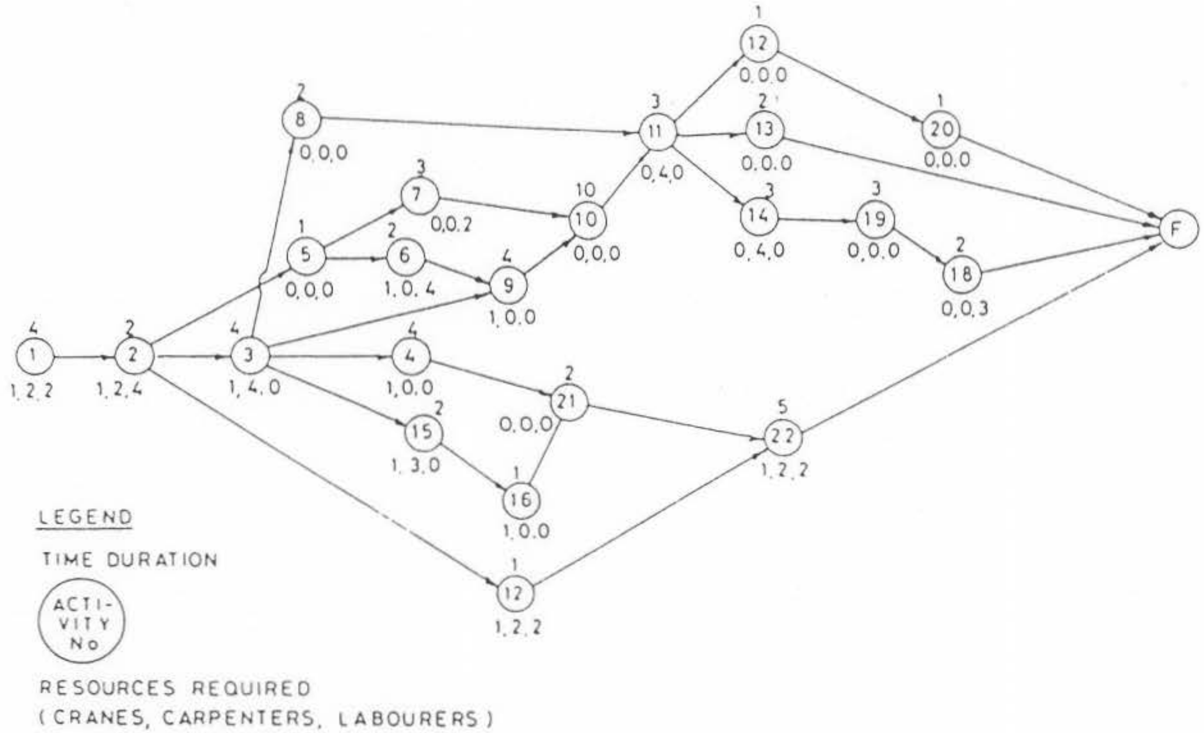
5. Total resource idle time

   The total resource idle time is the amount of time that resources are idle during a schedule span. Idle time can be measured in units of resource type-days. It is a result of the unavailability of direct project work, which in turn is a result of the scheduling method employed.

$$\sum_{k=1}^{K} \sum_{t} W_w R_{kt}$$

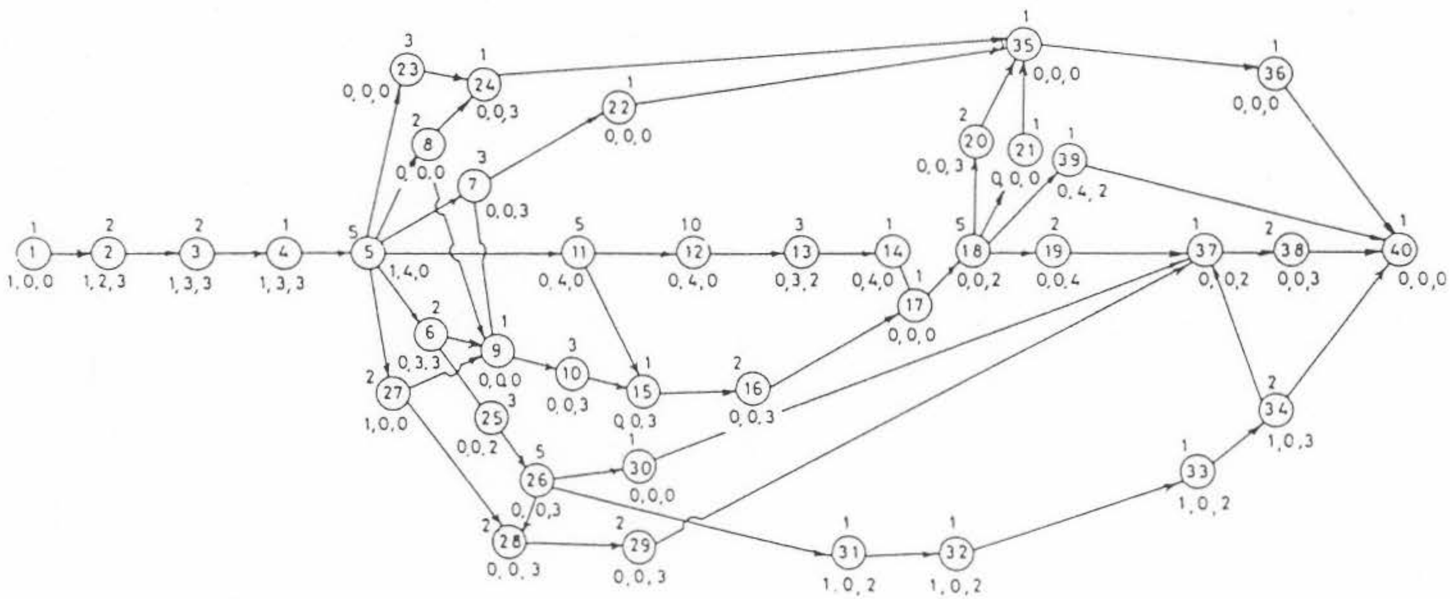
Three different projects shown in Figures. [1], [2], and [3], having 23, 40 and 20 activities respectively, are considered for analysis of the IGP model. The total number of decision variables and the deviational variables in IGP model are found to be 1749 and 409 respectively. Attempts were made to apply IGP model to single project-3 resources case, 2 projects-3 resources case and 3 projects-3 resources case. The results are presented in Tables [1], [2], and [3]. The Following remarks are considered to be worth mentioning:

1. It can be seen from the tables that the nature of the project (size, complexity and associated activities) has an impact over the solution results. For example, each project has an unique level of resources requirements and the performance measure varies from project to project. when the resource level is (1, 5, 5) for Project 2 and is reduced to (1,4,4), the project slippage increases, although resource idle time does not change much.
2. When the model was applied to 2 projects-3 resources case (Table (2)), for decrease in resource level from (1,6,6) to (1,5,5) project slippage increases drastically and idle resource time also increases. when the resource level is brought from (1,9,9) to (1,8,8), there is no significant effect on scheduling efficiency but idle resource time increases, and total project delay decreases:



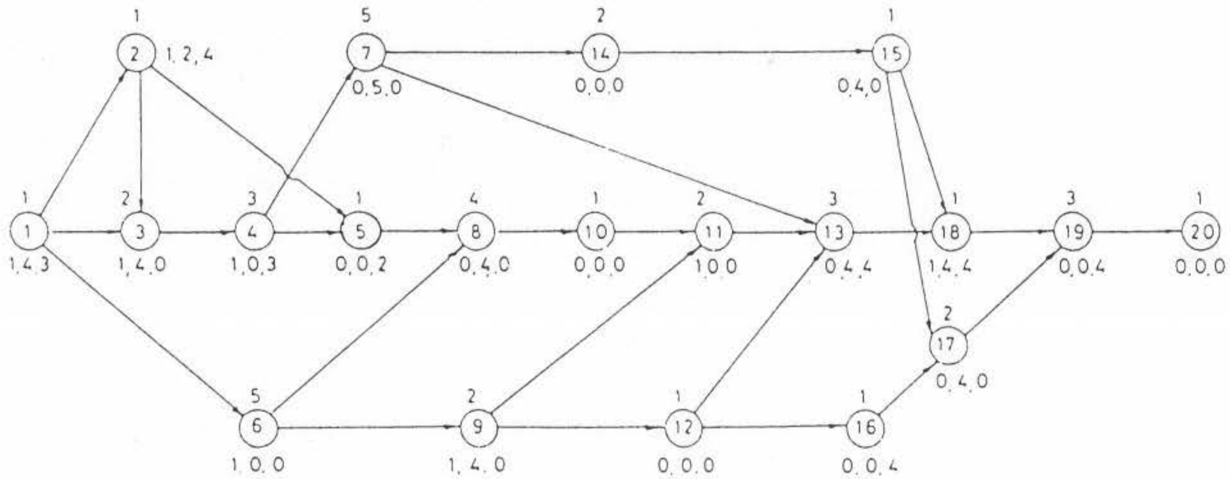
Network of Project No. 1.

Figure 1. [1]



Network of Project No. 2.

Figure 2. [1]



Network of Project No. 3.

Figure 3. [1]

Results of IGP model under 1 project-3 resources situation

Goal priority structure: P_1 schedule only l cranes, P_2 schedule no more than m carpenters, P_3 schedule no more than n labourers, and P_4 minimise total project lateness

Critical path method (time duration)	Resource constrained (time duration)	Resource level (l,m,n)	Project slippage	Total project delay	Scheduling efficiency	Weighted project delay	Idle resource time	Project
01 Jan. 88-04 Feb. 88	01 Jan. 88-06 Jan. 88	1,7,7	2	2	0.942	2.84	19.64	1
01 Jan. 88-04 Feb. 88	01 Jan. 88-06 Feb. 88	1,6,6	2	2	0.942	2.72	17.42	1
01 Jan. 88-04 Feb. 88	01 Jan. 88-10 Feb. 88	1,5,5	6	6	0.82	7.80	20.4	1
01 Jan. 88-04 Feb. 88	01 Jan. 88-10 Feb. 88	1,4,4	6	6	0.82	7.44	17.94	1
05 Jan. 88-20 Feb. 88	05 Jan. 88-20 Feb. 88	1,7,7	0	0	1.00	0.0	42.12	2
05 Jan. 88-20 Feb. 88	05 Jan. 88-21 Feb. 88	1,6,6	1	1	0.978	1.36	40.66	2
05 Jan. 88-20 Feb. 88	05 Jan. 88-23 Feb. 88	1,5,5	4	4	0.91	5.2	41.68	2
05 Jan. 88-20 Feb. 88	05 Jan. 88-27 Feb. 88	1,4,4	7	7	0.851	8.68	42.34	2
30 Jan. 88-25 Feb. 88	30 Jan. 88-26 Feb. 88	1,7,7	1	1	0.961	1.42	17.35	3
30 Jan. 88-25 Feb. 88	30 Jan. 88-28 Feb. 88	1,6,6	1	1	0.961	1.36	15.73	3
30 Jan. 88-25 Feb. 88	30 Jan. 88-26 Feb. 88	1,5,5	1	1	0.961	1.3	14.11	3
Infeasible solution produced		1,4,4						3

Table 1. [1]

Results of IGP model for 2 projects–3 resources constrained scheduling situation

Critical path method (time duration)	Resource constrained (time duration)	Resource level (l,m,n)	Project slippage	Total project delay	Scheduling efficiency	Weighted project delay	Idle resource time
01 Jan. 88–20 Feb. 88	01 Jan. 88–20 Feb. 88	2,10,10	0.0	0.0	1.0	0.0	74.08
01 Jan. 88–20 Feb. 88	01 Jan. 88–27 Feb. 88	1,09,9	2.0	19.0	0.9607	29.26	24.10
01 Jan. 88–20 Feb. 88	01 Jan. 88–21 Feb. 88	1,8,8	1.0	15.0	0.96	22.20	19.44
01 Jan. 88–20 Feb. 88	01 Jan. 88–22 Feb. 88	1,7,7	2.0	17.0	0.98	24.14	17.74
01 Jan. 88–20 Feb. 88	01 Jan. 88–25 Feb. 88	1,6,6	5.0	23.0	0.901	31.98	18.64
01 Jan. 88–20 Feb. 88	01 Jan. 88–02 Mar. 88	1,5,5	10.0	32.0	0.804	41.60	21.78

Table 2. [1].

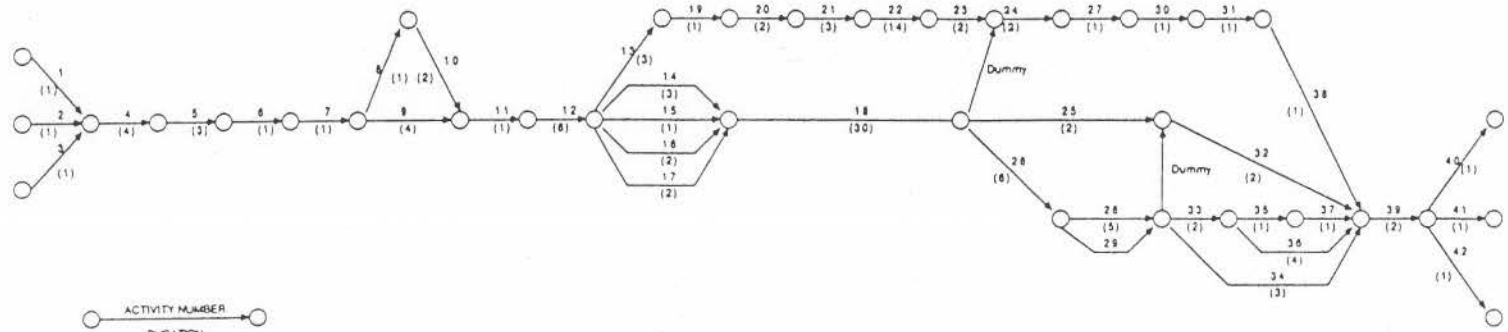
Results of integer goal programming model for 3 projects–3 resources constrained scheduling situation

Critical path method (time duration)	Resource constrained (time duration)	Resource level (l, m, n)	Project slippage	Total project delay	Scheduling efficiency	Weighted project delay	Idle resource time
01 Jan. 88–25 Feb. 88	01 Jan. 88–25 Feb. 88	3,15,15	0.0	0.0	1.0	0.0	139.89
01 Jan. 88–25 Feb. 88	01 Jan. 88–25 Feb. 88	2,14,14	0.0	2.0	1.0	5.68	80.53
01 Jan. 88–25 Feb. 88	01 Jan. 88–25 Feb. 88	2,13,13	0.0	2.0	1.0	5.56	77.17
01 Jan. 88–25 Feb. 88	01 Jan. 88–25 Feb. 88	2,12,12	0.0	3.0	1.0	8.16	73.81
01 Jan. 88–25 Feb. 88	01 Jan. 88–27 Feb. 88	2,11,11	2.0	5.0	0.96	13.3	75.77
01 Jan. 88–25 Feb. 88	01 Jan. 88–27 Feb. 88	2,10,10	2.0	5.0	0.96	13.0	72.29
01 Jan. 88–25 Feb. 88	01 Jan. 88–05 Mar. 88	1,09,09	8.0	38.0	0.857	58.52	20.05
01 Jan. 88–25 Feb. 88	01 Jan. 88–06 Mar. 88	1,08,08	9.0	45.0	0.84	66.66	17.69

Table 3. [1]

3. When the model was applied to 3 projects-3 resource case (Table(3)), scheduling efficiency almost remains constant for a number of resource level (particularly at high values), but is significantly reduced at low levels of resources. Similar findings are observable in case of project slippage also.

4. It can be seen that under various resource level the performance measures undergo drastic change and indicating a trade-off between the measures to the management. similar results can be observed for 3 projects-3 resources case (Table (3)). It can be inferred that with the change in resource level, the completion time for projects varies and thereby presenting to the management a cost-time trade off situation. Therefore, in the subsequent section, a possible extension of the model is shown to study the cost-time trade-off.



Typical network of home construction project

Figure 4. [7]

TABLE 4. List of activities and resource requirements

Act. No.	Activity Name	Duration (days)	Resource type	Required workers
1	Placement	1	Supervisor	1
2	Temp. plumbing	1	Plumber	1
3	Temp. electricity	1	Electrician	1
4	Foundation 1	4	Steeplejack	3
5	Hardening	3		
6	Foundation 2	1	Steeplejack	3
7	Rough plumbing 1	1	Plumber	1
8	Foundation 3	1	Steeplejack	3
9	Rough plumbing 2	4	Plumber	1
10	Hardening	2		
11	Scaffolding	1	Steeplejack	3
12	Framing	6	Framer	2
13	Exterior woodwork	3	Carpenter	2
14	Roofing	3	Roofer	2
15	Plumbing (gas)	1	Gas plumber	1
16	Plumbing	2	Plumbing	1
17	Wiring	2	Electrician	1
18	Interior woodwork	30	Carpenter	1
19	Sheet metal work	1	Sheet metal worker	1
20	Wire mesh	2	Plasterer	2
21	Exterior plasterer	3	Plasterer	4
22	Drying	14		
23	Exterior painting	2	Painter	1
24	Finish plasterer	2	Plasterer	3
25	Doors	2	Door worker	1
26	Interior painting	6	Painter	1
27	Gutter	1	Sheet metal worker	1
28	Tiles	5	Tile worker	2
29	Wall paper	5	all paper worker	2
30	Scaffolding	1	teeplejack	3
31	Balcony	1	Carpenter	2
32	Fusuma partition	2	Partition worker	1
33	Carpet	2	Carpet worker	1
34	Electric appliance	3	Electrician	1
35	Miscellaneous parts	1	Carpenter	2
36	Boiler	4	Plumber	1
37	Kitchen	1	Carpenter	1
38	Mortar	1	Plasterer	4
39	Cleaning	2	Cleaner	2
40	Screen	1	Carpenter	1
41	Storm windows	1	Carpenter	1
42	Tatami mat	1	Carpenter	1

division.

To aid in the analysis of Mitsui's operation a model was developed which would schedule up the 50 simultaneous projects (homes). Each project could comprise of up to 1000 activities over a 400 day planning horizon. five key resources were considered (although the model is dimensioned to consider up to 50 resources). It was desirable that the model provides for updating as additional home contracts arrived. A mechanism for calculating and presenting the free slack was provided, as well as resource profiles, to aid management in their final operational decision making.

The measures developed by Kurtulus and Davis[5] are used namely Average Resource Utilization Factor(AUF) and Average Resource Load Factor(ARLF). The ARLF defines where the peak of the total resource requirements is located in the original critical path duration. A peak located at the center of the time-only duration will have an ARLF of zero. A peak in the first half of the duration will have a negative ARLF, while a peak in the second half will have a positive ARLF. The AUF indicates the average 'tightness' of the constraints on each resource.

Discussions with Mitsui's management indicated that in general each activity, except the drying period, required only one critical resource to process. Using this assumption, the Average Resource Load Factor was calculated as a value of 0.21. This implies that the peak of average resource requirements is located in the second half of the critical path duration found by time analysis only, but is close to the center of the duration. The firm also felt that the Average Utilization Factor was in the region of 1.0. An AUF of 1.0 indicates that the company has enough resources on average to process their projects, but has insufficient resources to meet peak period demands. This condition was typical for Mitsui Home.

The following assumptions have been made in the development of the Multi-Project Model(MPM):

1. Activity times and precedence relationships are deterministic.
2. No activity splitting or pre-emption is allowed. (if splitting is possible, the activity may be considered as two or more activities.)
3. The quantity and types of resources required for each activity are known and constant.

These assumptions are generally consistent with the operational conditions found in the housing industry. While job splitting and pre-emption may occur, the update feature of the model will address these considerations.

Since resources are limited, conflicts among activities eligible to start at a given period will occur. the heuristic decision rule resolves the conflicts and allocates resources to activities. Unless sufficient resources are available for every period, a number of activities may be delayed beyond their 'unconstrained' schedule time.

Two heuristic decision rules are used for heuristic model, MPM. There are Shortest Activity from Shortest Project(SASP) and First Come First Served(FCFS).

The heuristic model, MPM, consists of the following steps.

Step 1. Calculate early start times of all activities based on time-only analysis. Set DAY = 0.

Step 2. If all the activities are scheduled, stop. Otherwise, list the activities which are eligible to start on the current DAY.

Step 3. Sort the activities on the list in increasing order of the SASP value. ties are broken by FCFS.

Step 4. Allocate the resources to activities based on the order of the sorted list, starting with the activity at the top of the list, as long as the resources are available. If there is not enough resources to schedule all the activities, postpone those activities which are not scheduled by one day.

Step 5. Update the early start time for the activities postponed.

Step 6. Set DAY = DAY + 1. Go to Step 2.

In Step 1, the early start times for all activities for all projects are calculated based on time-only analysis. The late start times, however, need not to be calculated. Therefore, only the forward pass calculations of a CPM routing are needed.

In Step 2, activities eligible to start on the current day are listed. Only those activities on the list are considered for resource allocation. Even though there may be an activity on the next day whose SASP value is much smaller than those of activities on the current scheduling day, the heuristic does not consider that smaller SASP activity until the next day. Once an activity is started(scheduled), it is not interrupted until it finishes. No 'look-ahead' feature has been provided in this model.

In step 3, activities on the current list are sorted based on the SASP value. The activity with the smaller SASP value is given the higher priority. The SASP value for each activity is given by:

$$SASP(M,N) = TMIN(M) + T(M,N)$$

where,

SASP(M,N) = the SASP value of activity N of project M,

TMIN(M) = the critical path duration of project M based on time-only analysis,

T(M,N) = the duration of the activity N of project M.

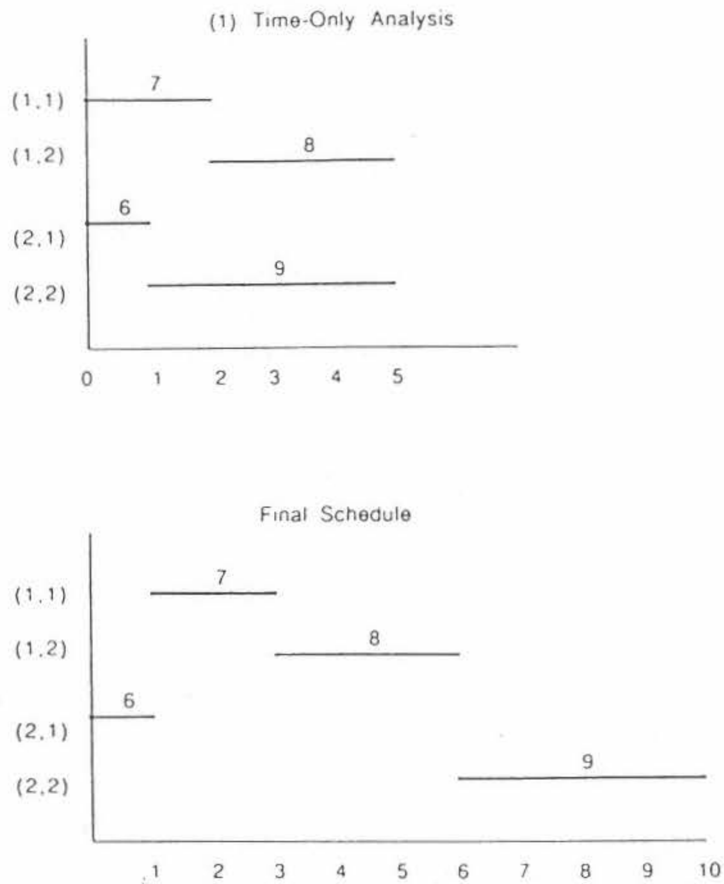


Illustration of scheduling based on the SASP rule

Figure 5. [7]

activity(2,2) is eligible to start, but the resource availability is zero. Activity(2,2) is postponed to day 3. Activity(1,1) is completed at the end of day 2. On day 3, activities(1,2) and (2,2) are eligible to start. Activity (1,2) is scheduled based on the SASP values. When activity(1,2) finishes (at the end of day 5), activity (2,2) will start.

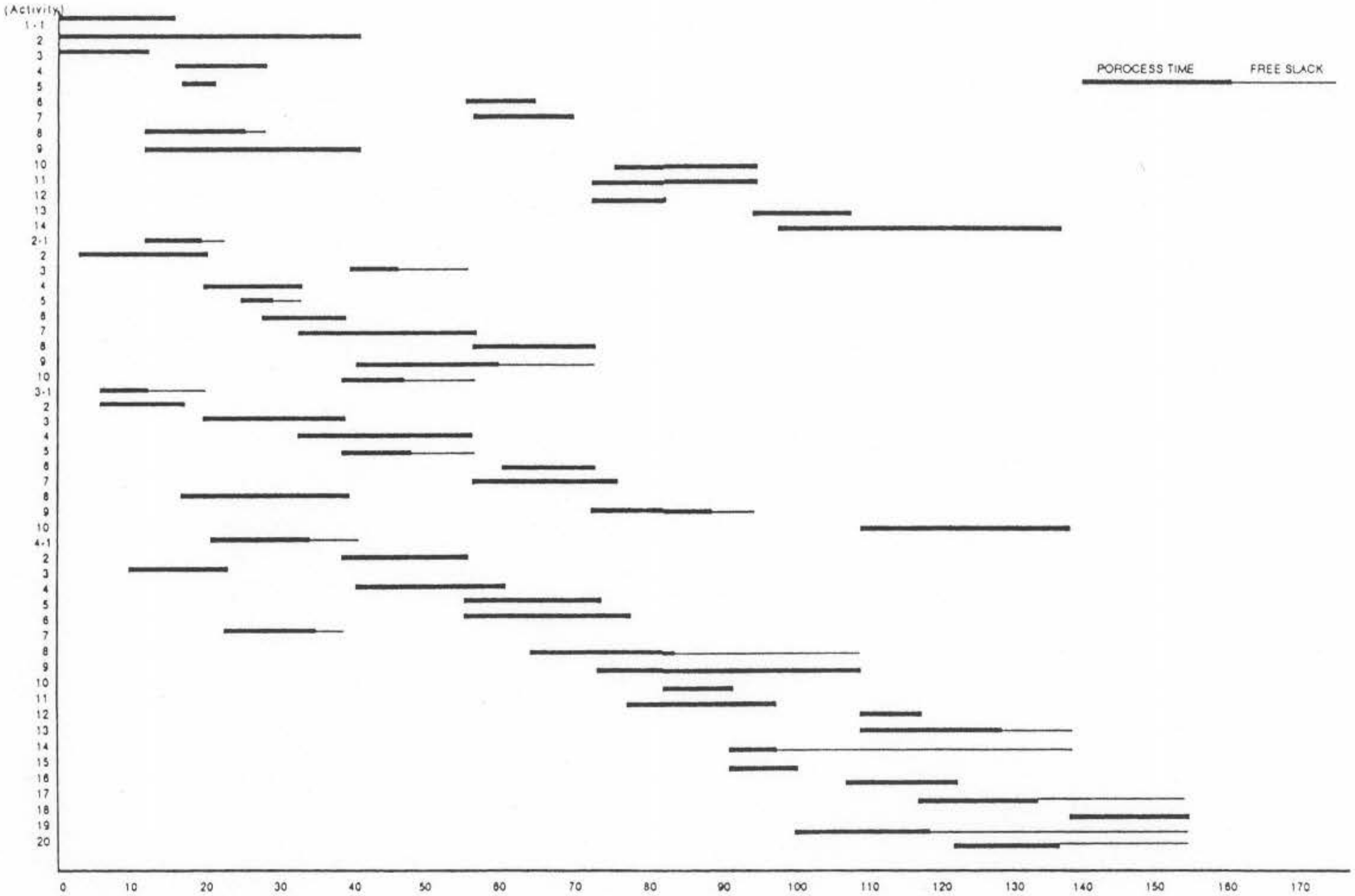
Results of the heuristics example is shown in figure 6.

III. MATHEMATICAL PROGRAMMING

Mathematical programming approaches to multi-project scheduling problems have both advantages and disadvantages to heuristic approaches. The chief advantage is that mathematical optimization techniques provide an optimal schedule with respect to the model's objective.

The most common objective for an optimization model is to minimize the completion time for the multiple projects. Linear Programming (LP) models can be formulated fairly easily for multi-project networks. If the network is relatively small, the LP solution can be obtained in a short time. However, this requires that the project activity durations be known with certainty. In the real world, this is rarely the case. Introducing probabilistic parameters to the LP increases the size of the matrix and makes it much more difficult to solve.

Another disadvantage to LP models and other mathematical optimization techniques is that the models are relatively expensive to develop and maintain. Since projects are temporary endeavors by definition, these development and maintenance costs must absorbed by the single scheduling problem. When the next group of projects comes up for scheduling, a new model



Final schedule by MPM

Figure 6. [7]

must be developed. A linear programming model with the flexibility to accept modified inputs, without having to completely rebuild the model would be desirable.

Since there are many packages available that utilize heuristic techniques that will allow this flexibility, their use should be strongly considered. These rule based packages provide a much quicker return of results. However, the value of an optimal solution should not be underestimated. Some of these software packages also provide LP solutions (or sometimes Integer Programming and Goal Programming solutions), although the run times may be longer. There is great value in solving a problem that is of the same type as your businesses' typical problem using optimization techniques.

If you then solve the same problem with the different heuristics utilized by your software, you will have a standard by which to measure their performance. Benchmarking the results of heuristics as a percentage of the optimal solution may be the most valuable use of mathematical programming techniques as applied to the multi-project scheduling problem.

This benchmarking process must be repeated as often as the structure of your problems changes appreciably. If you find yourself solving a great variety of multi-project scheduling problems that are dissimilar, you may not gain much from the benchmark analysis.

But if your problems are consistently alike in structure, and differ chiefly in the parameters used, you will likely find that one heuristic is consistently best, and further that it is relatively close to the optimal solution. In this case the trade-off of a little performance off the optimal solution for a greater turn-around of results is well worth it.

IV. HEURISTICS

Today, most organizations undertake multiple projects at different locations. The concurrent management of multi-projects within a single organizational framework involves a lot of complexity in scheduling, monitoring and allocation of available resources [1].

To satisfy the overall objective stated before, literally hundreds of different heuristic based models are in use today. The question is: Do all of the models produce good, feasible solutions?

The question can be answered by looking into the fundamentals of heuristic modeling. That is **the sequencing rules** followed to develop a model. While developing heuristic rules for models, it is important to note that the model outcome produces a good feasible solution, not an optimal solution. Later, following this section the effectiveness of heuristic sequencing rules relative to an optimum solution is discussed.

Two basic approaches for formulating the sequencing rules mentioned in our literature research are:

1. Davis and Patterson [3] treated multiple project as one single big project. Multiple projects were artificially combined with dummy start and end activities into a large project. Heuristic sequencing rules was based upon the objective of minimizing total delay time with equal penalty.
2. Kurtulus and Narula [4] treated multiple projects as aggregation individual of projects. The sequencing rules was based upon the objective of minimizing of the total delay time with unequal penalty.

Let CT_i = Resources-constrained completion time of project i.

CP_i = Unconstrained critical path of project i.

M = Total number of projects.

Under equal penalty multi-project approach the objective is to minimize total delay,

$$z = \Sigma_{i=1}^{M} \{ CT_i - CP_i \}$$

Where as under the single project approach the objective is to minimize total delay,

$$z = Max_i\, CT_i - Max_i\, CP_i$$

The two objective functions will not necessarily lead to the same value of z.

For the following section please note common denotions used:

CP, EST and LST = standard CPM analysis.

d_{ij} , EST_{ij} , and LST_{ij} = duration, early and late start time respectively of activity j of project i.

P_i and CP_i = penalty and critical path length.

r_{ij} = per period requirement of resource type i by activity j.

M = number of different resources type.

Heuristic sequencing rules considered by Davis and Patterson. [3]

1. Minimum Job Slack (MINSLK).

   Priority in resolving resource conflicts is given to the activity with minimum slack.

   $MINSLK = CP \{ LST_j - EST_j \}$

Slack is continuously updated. Any delayed activity has its slack reduced by the amount of the delay.

2. Resource Scheduling Method (RSM).

Decision priority is given to precedence of that activity which supports minimum project duration time - d_{ij}

d_{ij} = increase in project duration resulting when activity j follows activity i.

$RSM = Max \{ 0; [EFT_i - LST_j]\}$

Pairwise Comparison method is used for comparing d_{ij} at each schedule interval.

3. Minimum late Finish Time (LFT).

Priority rule is assigned to activity with least Late Finnish Time (LST) determined by CPM.

$LFT = Min \{ LST_i \}$

4. Greatest Resource Demand (GRD).

Priority rule is assigned on basis of total resource unit requirements of all types. Higher priority is given for greater resources demands (potential resource bottlenecks activity).

$GRD = d_j \Sigma_{i=1}^{M} r_{ij}$

5. Greatest Resources Utilization (GRU).

Priority is assigned to that combination of activity which results in maximum resources utilization in each scheduling interval. The other way of looking at this method is minimizing Idle resource in an interval.

$GRU = Max \{ \Sigma_{i=1}^{M} r_{ij} \}$

6. Shortest Imminent Operation (SIO).

   Shortest job first rule assigns priority on the basis of activity duration. It is effective in reducing the average total processing time for a group of jobs.

   $$SIO = Min \{ d_i \}$$

7. Most Jobs Possible (MJP).

   This rule gives priority to that combination of activity which results in the greatest number of activity being scheduled in any interval. The heuristic rule is based upon determination of the greatest no .of jobs possible made purely with regard to resource feasibility and not resource utilization.

8. Select Jobs Randomly (RAN).

   This heuristic sequence rule assigns priority among competing jobs on a purely random basis.

Heuristic sequencing rules considered by Kurtulus and Narula.[4]

1. Maximum Duration and Penalty - (DURPEN).

   Equal delay penalties are assigned to projects $= P_i$

   $$DURPEN = Min \{ P_i d_{ij} \}$$

   Priority is given to the activity with the longest operation first (LOF).

2. Maximum Penalty - (MAXPEN).

   Priority is given to the activity with maximum penalty first.

   $$MAXPEN = Max \{ P_{ij} \}$$

   If $P_{ij} = P_i$

   $$MAXPEN = MAX \{ P_i \} = FCFS$$

   The priority is given on First Come First Serve (FCFS) basis.

3. Maximum Total Duration Penalty - (MAXTOP).

   $$MAXTOP = Max \{ P_i (CP_i + d_{ij})\}$$

   $CP_i + d_{ij})$ = the total duration

   P_i = unequal penalty for i project

   If P_i is equal penalty the rule will give priority to longest activity from the longest project first.

4. Maximum Total Work Content - (MAXTWK).

   Priority assigned to activity with maximum value of work content.

   $$MAXTWK = TWK_i + d_{ij} \Sigma_{k=1}^{k} d_{ij}$$

   where,

   $$TWK_i = \Sigma_{k=1}^{k} \Sigma_{j \epsilon ASi} d_{ij} r_{ijk}$$

r_{ijk} =amount of resources k required by activity j of project i
AS_i =set of activity already schedule from project i
K = no.of resources

5. Minimum Total Duration Penalty - (MINTWK)

   Same as above, except that priority is assigned to activity with minimum value of work content.

6. Shortest Activity from the Shortest Project - (SASP1)

   Schedule the activity with minimum value of ($CP_i + d_{ij}$). Tie breaker is maximum penalty.

7. Shortest Activity from the Shortest Project - (SASP2)

   Same as above but the tie breaker is First Come First Server (FCFS).

Until now, heuristics rules were basically emphasised on CPM network analysis. Badiru [2] in his paper address heuristics using PERT networks with precedence and resource constraints. Badiru [2] discusses in his paper towards development of "standardization of performance measures for project scheduling heuristics in PERT networks." That is to develop some form of aggregate criterion for comparing rules discussed later in the following section.

PERT heuristic sequencing rules considered by Badiru [2].

1. Activity-Time (ACTIM).

   ACTIM scheduling heuristic represents the maximum time that an activity controls in the project network on any one path. It is computed for each project activity by subtracting the activity's latest start time from the critical path time as shown below:

ACTIM = CP time - Activity Latest start time.

2. Activity-Resource (ACTRES).

ACTRES is a scheduling heuristic using combination of the activity time and resources requirements.

ACTRES = Activity time * Resource Requirement

3. Time-Resource (TIMRES).

Another priority rule composed of equally weighted portions of ACTIM and ACTRES.

TIMRES = 0.5(ACTIM) + 0.5(ACTRES)

where ACTIM and ACTRES are calculated as previously discussed.

4. GENRES.

GENRES is in effect a modification of TIMRES with unequal weights assigned to ACTIM and ACTRES. The procedure for GENRES is a computer search technique whereby various weights of w (between 0 and 1) are used in the computational expression shown below.

GENRES = w (ACTIM) + (1-w) (ACTRES)

5. Resource Over Time (ROT).

The scheduling heuristic is simply the resource requirement divided by activity time as shown below:

$$\text{ROT} = \frac{\text{Resource requirement}}{\text{Activity time}}$$

The resource requirement in the above expression can be replaced by scaled sum of resource requirements in the case of multiple resource types.

6. Composite Allocation Factor (CAF).

CAF was developed by Badiru [2]. For each activity *i*, CAF is computed as a weighted and scaled sum of two components.

a) Resource Allocation Factor (RAF) a measure of the expected resource consumption per unit time.

$$\text{Scaled RAF}_i = \frac{\text{Unscaled RAF}_i}{\text{Max}\{\text{RAF}_i\}} * 100$$

b) Stochastic Activity Duration Factor (SAF) is defined as follows,

$$\text{SAF}_i = \mu_i + \frac{\sigma_i}{\mu_i}$$

where,

μ_i expected duration for activity i,

σ_i standard deviation of duration for activity i, and

σ_i / μ_i coefficient of variation of the duration of activity *i*.

Therefore, now CAF is defined as,

$$\text{CAF}_i = (w)\ \text{RAF}_i + (1+w)\ \text{SAF}_i$$

where w is a weighing factor between 0 and 1.

It is on the basis of the magnitudes of CAF that an activity will be allocated resources and a time slot in the project schedule . An activity that lasts longer, consumes more resources, and varies more in duration will have a larger magnitude of CAF. Such an activity is given priority during the scheduling process.

The issue of uncertainty in activity duration has been extensively addressed here.

V. STUDIES ON COMPARISON OF SCHEDULING RULES

As discussed in the previous section, many heuristic scheduling rules have been identified during the literature search. These rules have been studied by researchers in the scope of determining the best rule for different cases of multiple project scheduling. Major studies as discussed in the previous section, have been done by Kurtulus, Narula, Davis, Patterson, and Badiru. In this section these studies will be presented and analyzed. Their performance measures will be studied and analysis of the results will be presented.

Davis and Patterson, in their 1975 paper, studied eight different heuristic project scheduling rules in general. The heuristics tested were chosen because they had been found effective in eleven earlier studies, only two of which had compared the heuristic solutions to an optimal solution. They based the study on single projects under conditions of multiple limited resource requirements.

Kurtulus and Davis also presented a study which evaluated the performance of different heuristics for multiple project scheduling rules. In their study, they assumed no pre-emption: once an activity was started its progress was not interrupted. In addition, they assumed the amount of resource required by a specific activity was constant, as was the amount of a resource available per period. Kurtulus and Davis studied the three heuristics most favored in previous research as well as six new heuristics which they proposed. The purpose was to compare the performance of the nine total heuristics in achieving the objective of minimizing the total project delay.

Kurtulus and Narula expanded the research on multiple project scheduling by analyzing application of heuristic solution procedures to resource constrained, multi-project scheduling problem under equal and unequal penalties. They categorized the performance of ten scheduling rules with respect to four project summary measures, namely, resource-constrainedness, location of the peak requirements, and problem size. It was shown that the choice of a scheduling rule can be based upon the resource-constrainedness, and problem size. Their results were based on scheduling, in detail, over 3000 multi-project scheduling problems, each containing three projects and from 24 to 66 activities.

Badiru contributed to the research on multiple project scheduling by presenting a study on standardization of performance measures for project scheduling heuristics. He proposed three quantitative measures related to the project duration: rule efficiency ratio, sum of rule efficiency ratios and percentage deviation from minimum.

Performance Measures and Problem Kinds.

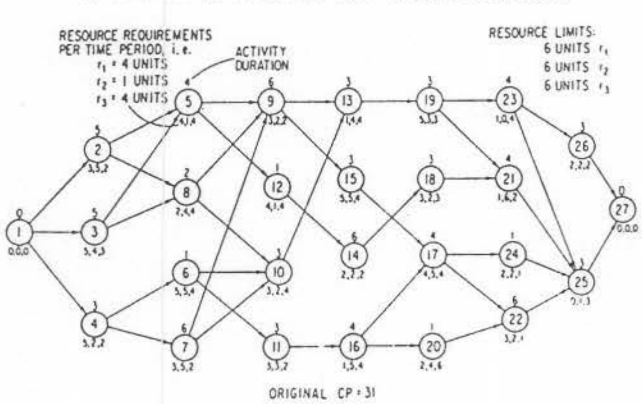
In this sub section the procedures to measure the performance of heuristic rules that were used in the study will be presented and discussed. The problem kinds and network types justifying the applied measures will also be presented.

In the research of Davis and Patterson, heuristic rules were measured relative to the optimum schedule duration for a group of test problems. The experiment consisted of successively solving eighty-three different problems and comparing the result of each against the optimal solution.

The test problems used in the experiment involved fifty seven different computer generated networks. These networks arbitrarily limited in size to between twenty and twenty seven activities to guarantee ease of optimum solution. Within this constraint, individual networks were generated under a procedure which produced problems with such characteristics as network structure and resource requirements similar to those encountered by the authors in practice. For example the ratio of number of arcs to number of activities in the sample varied between 1.0 and 3.0. Activity durations varied between one and nine time units. Each activity had fixed multiple-unit requirements of up to three different resource types, with a maximum of three types per project. All types were subject to fixed resource availabilities which were constant over the project duration. The ratio of required work units to available work units, measured over the critical path duration, varied between 0.58 and 1.50. While only fifty seven different networks were used, some were assigned different combinations of resource availabilities, giving a total of eighty three different problems. A test problem example is shown in Figure(7), along with the results.

The heuristics selected were employed within the framework of a parallel approach in which activity priority is determined during scheduling rather than before .

COMPARISON OF HEURISTIC AND OPTIMUM SOLUTIONS



SCHEDULE RESULTS	
OPTIMUM DURATION = 64	RSM = 74
LFT = 67	RAN = 74
GRU = 68	MJP = 76
SIO = 71	GRD = 80
MINSLK = 74	

Figure (7) Example test problem

The research of Kurtulus and Davis breaks away from others by providing a categorization process based on two powerful project summary measures. The first measure identifies the location of the peak of total resource requirements and the second measure identifies the rate of utilization of each resource type. The performance of the rules are classified according to values of these two measures.

The first measure is termed the Average Resource Load Factor (ARLF), and it identifies whether the peak of the total resource requirements is located in the first or second half of the project's critical path duration. The value of ARLF for a collection of projects is imply the average of the individual project values.

The second summary measure termed the Average Utilization Factor (AUF) indicates the average tightness of the constraints on each required resource. Utilization Factor(UF) is calculated for each problem. UF for each resource is defined as the ratio of the total amount required to the amount available in the time period. When UF is equal to or less than 1, there will be zero project delay. Also if two problems have the same UF values for each resource in each time period and if the resources are constraining then the same resulting delay should be obtained for each problem when the same sequencing rule is applied.

The test projects studied in this research had a range of 0.6 to 1.6 for the AUF factor. The range of values for the ARLF was then generated by using selected values of project summary measures which were studied by Davis, Mize, Pascoe, and Patterson in their previous studies. These summary measures include measures as the number of activities, number of parallel paths, activity time, and resource requirement distributions, and complexity. As a result 3 categories of project shapes were studied (Figure 8).

Since the time and resource requirements of the activities were compatible, when all projects were chosen from the first category, a minimum value for ARLF was determined as -3.0 and when they were chosen from the second category, a maximum value of ARLF of +3.0 was determined. This range was arbitrarily divided into 7 equal intervals, the midpoints of which were set at -3, -2, -1, 0, 1, 2, 3 . Since some initial tests suggested that the project delay was much more sensitive to changes in the value of the AUF than ARLF, the range of AUF was divided into eleven equal intervals with midpoints 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6 . Among the 77 project sets generated by repeating 11 levels of AUF for each level of ARLF, the total number of activities per problem varied between 34 and 63 activities.

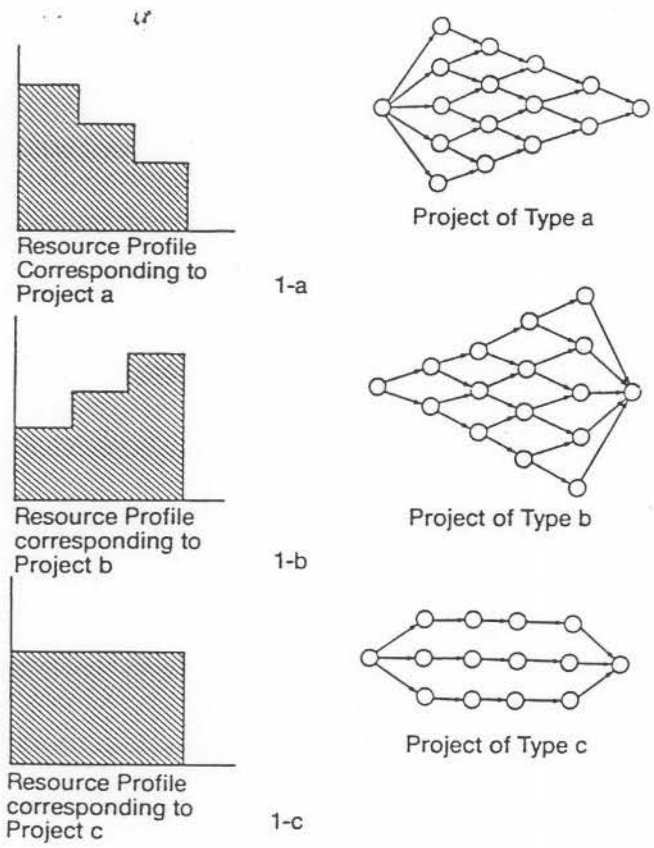


Figure (8) Categories of Project Shapes

In short their experimental design depended upon the following factors. If we define y(ijl) denote the delay obtained when rule i is applied with performance measures at levels j and l respectively, then $y(ijl) = u + Rule(i) + ARLF(j) + AUF(l) + Error$.

They ranked the performance of the rules using this function.

Kurtulus and Narula used network complexity as their first measure. This measure for the single project problem was introduced by Pascoe. It has been hypothesized that complexity affects the performance of the scheduling rule and hence must be an integral part of the experimental design.

The second summary measure they used in their study measures the effect of peak resource requirements for each resource on the performance of a scheduling rule. This measure has two components : Maximum Load Factor (MLF) for each resource and Average Load Factor (ALF) which is calculated by dividing the MLF by the total number of activities in the problem. MLF was sensitive to the number of activities in the problem. ALF, however should have considerable appeal for management. It shows both the magnitude and location of the peak requirements of the most critical resources. In practice, firms have adopted different scheduling strategies according to the location and magnitude of the peak requirements of their critical resources, and ALF was expected to affect the performance of the scheduling rules and project delay.

The third summary measure, termed the average utilization factor (AUF), indicates the average tightness of the constraints on each resource. This measure was used in the research of Kurtulus and Narula as well. The fourth summary measure used in the study is the penalty function. The penalty function in its most general form is a function of a daily overhead figure assigned by the firm and a daily amount negotiated by a customer. Levels of penalties change according to the type of assignment of priority. In some cases the longest, in some the shortest, in some the project requiring the greatest amount of resources is assigned the highest priority.

Problem size in terms of number of activities is the fifth summary measure used in the study. The network shapes studied by Kurtulus and Davis were used in this research to generate the networks. By changing the levels of the measures within determined ranges the authors generated 3168 problems. In this study the project delay which was defined by Kurtulus and Davis before was redefined as: y = Overall mean + Rule + Penalty + ALF + AUF + Complexity + Size + Error

Badiru as mentioned before addressed the need for the standardization of performance measures for comparing scheduling heuristics in his paper. He proposed three quantitative measures related to the project duration. The first one was rule efficiency ratio which is the ratio of the minimum project duration observed during the comparative analysis to the project duration obtained from the rule under consideration. The second one was sum of rule efficiency ratios, which is the sum of the ratios computed for each test problem under each scheduling heuristic. The third measure was percent deviation from minimum, which measures the deviation of the project duration produced by a given heuristic from the overall minimum observed. He used 30 test problems which was a mixture of large and small projects.

VI. RESULTS

In Davis and Patterson's study each of the eighty-three test problems was solved optimally via bounded enumeration and then with the eight heuristic sequencing rules described. The optimal duration was taken as a base of comparison for the heuristically obtained durations. The comparisons concluded that MINSLK was the best heuristic. This heuristic had previously been found to be the most effective by Mize, Fendly, and Patterson in a multi-project scheduling problem. It was concluded that network structure and resource requirements may affect which heuristic is really the most effective. Thus the best heuristic is dependent upon the characteristics of a specific project problem. For example, it was found that heuristic performance was generally poorer on those problems in which there was a high ratio of average resource requirements per activity to the amount available, and that the MINSLK rule seemed to be more affected by the presence of this factor than were any of the other rules.

Heuristic Scheduling Rules versus Summary of Results; A:Average % increase above optimum, N:No of problems for which an optimum duration was found.

	MINSLK	RSM	LFT	RAN	GRU	GRD	SIO	MJP
A	5.6	6.8	6.7	11.4	13.1	13.1	15.3	16.1
N	24	12	17	4	2	11	1	2

Kurtulus and Davis, as a result of their two different measures classified the heuristic rules accordingly. According to their results, if the important variable to management is the occurrence and the timing of peak of the resource requirements, then the choice of rule should be based on the following table:

LEVEL	RULE
-2.5<ARLF<-3.5	SASP better than MINSLK, MAXSLK, and MAXTWK
-1.5<ARLF<-2.5	SASP better than MAXTWK
-0.5<ARLF<-1.5	MAXTWK better than SASP
-0.5<ARLF<0.5	No best Rule, MAXTWK, SASP, MINSLK, MAXSLK, SOF
0.5<ARLF<1.5	SASP better than FCFS and MAXTWK
1.5<ARLF<2.5	MOF better than MINSLK and MAXTWK
2.5<ARLF<3.5	SASP better than MINSLK

On the other hand, if management is primarily concerned with the tightness of this resource, the choice will be based on the following table:

AUF	RANK	(a) Based on Delay	(b) Based on Rank	(c) Based on Number First
0.6	1	MINSLK	MINSLK	MINSLK
	2	FCFS	FCFS	MAXSLK
0.7	1	MINSLK	MINSLK	MINSLK
	2	SASP	SASP	SASP
0.8	1	SASP	SASP	MINSLK
	2	MINSLK	MINSLK	SASP
0.9	1	SASP	SASP	SASP
	2	MAXTWK	MINSLK	MINSLK
1.0	1	MAXTWK	SASP	SASP
	2	SASP	MAXTWK	MAXTWK
1.1	1	SASP	SASP	SASP
	2	MAXTWK	MAXTWK	MAXTWK

If both the location of the peak and the tightness of the resources are important then Kurtulus and Davis recommended the choice to be based on the following table:

	AUF Range	
ARLF Range	0.6 to 0.8	0.9 to 1.6
-3.5 to -2.5	MINSLK	SASP
-2.5 to -1.5	MAXTWK	SASP
-1.5 to -0.5	SASP	MAXTWK
-0.5 to 0.5	MINSLK	SASP,SOF,MAXTWK
0.5 to 1.5	SASP	SASP
1.5 to 2.5	MINSLK	MOF,SASP
2.5 to 3.5	MINSLK	SASP

This Table Suggested "One Best Rule" for Various ARLF and AUF Categories.

The results of Kurtulus and Narula enhanced the previous studies at a very high level. The number of the problems they studied helped to generalize their results. The overall performance of the rules are summarized below. The results obtained under equal penalties confirm the previous finding of Kurtulus and Davis and to an extent of Davis and Patterson.

RULE versus Number of times each rule was ranked first

(Unequal Penalties) (Equal Penalties)

MAXPEN	717		SASP1	210
MAXTOP	469		SASP2	210
SASP2	459		MINSLK	125
SASP1	445		MAXTWK	102
MINSLK	422		RCRS	64
MAXTWK	420		SOF	59
DURPEN	241		LOF	41
SOF	147		MINTWK	34
MINTWK	76		LALP	25
SLKPEN	71		MAXSLK	16

Performance of the rules for small and large problems is summarized below.

UNEQUAL PENALTIES				EQUAL PENALTIES			
SMALL		LARGE		SMALL		LARGE	
MAXPEN	380	MAXPEN	337	MAXTWK	88	SASP1,2	133
MAXTWK	324	SASP2	312	SASP1	77	MINSLK	69
MAXTOP	248	SASP1	278	SASP2	74	SOF	42
MINSLK	196	MINSLK	226	MINSLK	56	MINTWK	23
SASP1	167	MAXTOP	221	FCFS	45	FCFS	19
SASP2	147	DURPEN	131	LOF	29	MAXTWK	14

As seen above MAXTWK rule performed high for small problems. It is because the activities with high resource requirements are responsible for creating bottlenecks at the work centers and such bottlenecks are strongly correlated with project delays in small projects compared to large projects. Hence scheduling these activities as soon as resources become available should minimize project delays in small problems. The findings for large problems confirmed the results of previous studies of Kurtulus and Davis, and Davis and Patterson.

The research also concluded that MINSLK is the best when the resources are constrained but not tight. This performance of MINSLK can be attributed to the possibility that the constrained and unconstrained critical paths of the projects were identical over most of the time periods during which resources were not tight. Classification of results for different levels of AUF are summarized below.

	EQUAL PENALTIES		UNEQUAL PENALTIES	
	Constrained (0.6<AUF<0.8)	Tight (0.9<AUF<1.6)	Constrained (0.6<AUF<0.8)	Tight (0.9<AUF<1.6)
Small	MINSLK	MAXTWK	MINSLK	MAXPEN
Large	MINSLK	SASP	MINSLK	MAXPEN

Badiru in his research defined many heuristic rules with different abbreviations. Since these abbreviations were explained in the previous section, only rankings will be presented in this section.

The rankings with respect to each of the performance measures is shown below:

Rank	Sums of Efficiency Ratios	Deviation from Min Project Duration	PERT Resource Constrained Duration
1	CAF	SPD	ACTRES
2	ROT	ROT	ROT
3	MLCT	HNIS	HNIS
4	ACTIM	CAF	MTS
5	MLST	ACTRES	MLCT
6	GENRES-2	MTS	LPD
7	TIMRES	TIMRES	MLST
8	MTS	LPD	SPD
9	HNIS	GENRES-2	ACTIM
10	ACTRES	ACTIM	TIMRES
11	GENRES-1	GENRES-1	GENRES-1
12	SPD	MLST	GENRES-2
13	LPD	MLCT	CAF

It has been known that the performance of a scheduling approach can sometimes deteriorate with the increase in project sizes. Further study showed that the first part of the ranking with respect to the efficiency ratios were same for the 8 large projects out of the 30 test problems. Therefore it was easily concluded that the rules' ranks are controlled more by the values for the large projects. The second ranking shows changes for individual problems. Some rules appear to be the worst and some worst, no inbetween. The third ranking may change according to CPU differences. For example CAF needs many computations so at higher CPU characteristics this rule might be ranked higher.

VII. ANALYSIS

Since all the studies used different multi-project scheduling problems, no one scheduling method can be deemed universally best. Heuristic performance is difficult to evaluate on a global scale. Any attempt to find a panacea among heuristics will result in frustration and failure. To hope that what works best for the construction industry also works best for service industries, hi-tech manufacturers, and the entertainment industry is to wish against long odds. However, any business is only concerned with effectively scheduling its own projects, and though there is no way to say for certain at this point which method to use for a given industry, it is possible to provide a systematic approach to evaluating the various methods as they apply to a specific businesses' projects. It is a mistake to assume that one heuristic will always provide you with the best schedule.

When evaluating the heuristics, it can be helpful to solve the problem using optimization techniques first. With many of the software packages available, networks can be solved to get the optimal solution. These solutions generally take longer to run, but in the initial heuristic evaluation stage it is worth it to have an optimal solution to use as a benchmark. Then the heuristics can be graded as a percentage of the optimal solution. For example, if the objective is to minimize slippage, and the optimal duration of the project is 100 days, then a heuristic scheduling the project for a 110 day duration, would have a grade of 110% of the optimal duration. This information is valuable in that it provides instant feedback as to how close to optimal each heuristic is on your multi-scheduling problem. Where a mathematical optimization can be performed without heavy investment in model development, it is foolhardy to ignore the results of such a model just because the model takes a little longer to run. This benchmarking should be done periodically, or at least as often as the structure of your typical problem changes significantly. If your typical problem is relatively stable in terms of structure, then the development of a mathematical optimization model to solve your multi-project scheduling problems is a project worth evaluating.

For a quicker heuristic evaluation, several of the summary measures used by Kurtulus & Davis and later by Kurtulus & Narula can be calculated for the problem most typical to your business. Then you can make a best guess for the most applicable heuristic by choosing the best performer identified by Kurtulus, Davis, and Narula on problems with similar summary measures. It is worthy to note that Davis & Patterson did not identify their problem set at all, so general conclusions drawn from their work should not be assumed to extend to specific problems. In selecting heuristics, it is important to evaluate performance based on how they work for you. In the study by Davis & Patterson, several heuristics that had been recommended by previous studies proved to perform worse than random scheduling.

It is recommended that several heuristics be performed simultaneously to hedge against adopting poor schedules. This should be part of the regular scheduling process. Since most heuristics can be run in a short time, this is a minor inconvenience relative to the value it provides. Taking a little extra time developing the schedule can help you avoid putting all your eggs in one heuristic's basket. Experience with the results of different heuristics will eventually provide insight into the specific instances when each heuristic outperforms the others. These insights should be used cautiously, and it is always recommended to run two or more heuristics on any scheduling problem.

If the multi-project scheduling problems you face are fundamentally alike in structure and resource constraintedness, a heuristic evaluation process should be undertaken to determine which heuristic best fit your businesses typical problem. It will likely turn out that 3 or 4 heuristics consistently perform better than the others. If so, these are the ones that should be run simultaneously during each project scheduling session.

The objective of the project schedule should be clearly defined and accepted by all involved in the scheduling process. Often a minimum duration objective is accepted because it is assumed that the shortest completion time will minimize costs and maximize profits. This is not always a valid assumption. When resource requirements are not smooth, the costs to meet varied staffing levels may go unconsidered. If these costs (cost to hire employees and cost to fire employees) can be defined, any schedule can then be evaluated based on cost. Much of the output from various rule based scheduling packages can be easily loaded into spreadsheets which can automate the cost calculations. Evaluations based on cost provide the level playing field to objectively determine which methods should be used for scheduling. The evaluations based on cost are most consistent with the goals of most businesses, and provide the clearest results.

VIII. EXPERIMENTATION

After reviewing the literature on multiple project scheduling, synthesizing many comparisons done by many researchers and analyzing their results, we have decided to support our conclusions with some experimental results.

We have chosen a multi-project problem from the literature that we have gathered from the literature search. This problem is explained in the "Major approaches" section. It has three projects having 23, 40 and 20 activities. There are three kinds of resources used (figure 1,2, and 3).

The authors solved this problem by integer goal programming. We tried two heuristic rules: 1.Selecting the activity with less total float, 2.Selecting the activity with the shortest duration.

Both heuristics were studied by the other researchers that has been mentioned in this study. However some different abbreviations were used. But the first one showed high performance at almost every ranking, so it was expected to give better results than the second one. Also the first one was found to be effected much when constraints are tight. To test this resources were decreased gradually for different runs.

Texim project software was used to simulate this scheduling problem. The resource levels were changed for each run of each heuristic. The durations and the critical project for each run are presented below.

TOTAL FLOAT						SHORTEST DURATION		
Resources	Start	End	Slippage	CP		End	Slippage	CP
6,6,1	1/1/92	**4/16/92**	**48**	**1**		**5/14/92**	**69**	**2**
8,8,1	1/1/92	**4/6/92**	**38**	**3**		**4/24/92**	**49**	**2**
10,10,2	1/1/92	**3/3/92**	**4**	**1**		**3/24/92**	**18**	**2**
12,12,2	1/1/92	**2/28/92**	**0**	**2&3**		**3/6/92**	**0**	**2**
14,14,2	1/1/92	**2/28/92**	**0**	**2**		**3/6/92**	**0**	**2**
15,15,3	1/1/92	**2/28/92**	**0**	**2**		**3/10/92**	**4**	**2**

TEXIM Results

As it can be seen from the above results, Total Float performs better than Shortest Duration at every occasion. The slippage does not increase when resources are constrained as much as it does in Shortest Duration. The experimentation results were graphed in figure 9 & 10. These graphs give us a good idea about the perfect slippage. When resources are moderate, rate of increase of slippage with respect to resource level changes, is the same for both rules. But when resources are very limited or more than moderate amount, total float has a less rate of increase for project slippage.

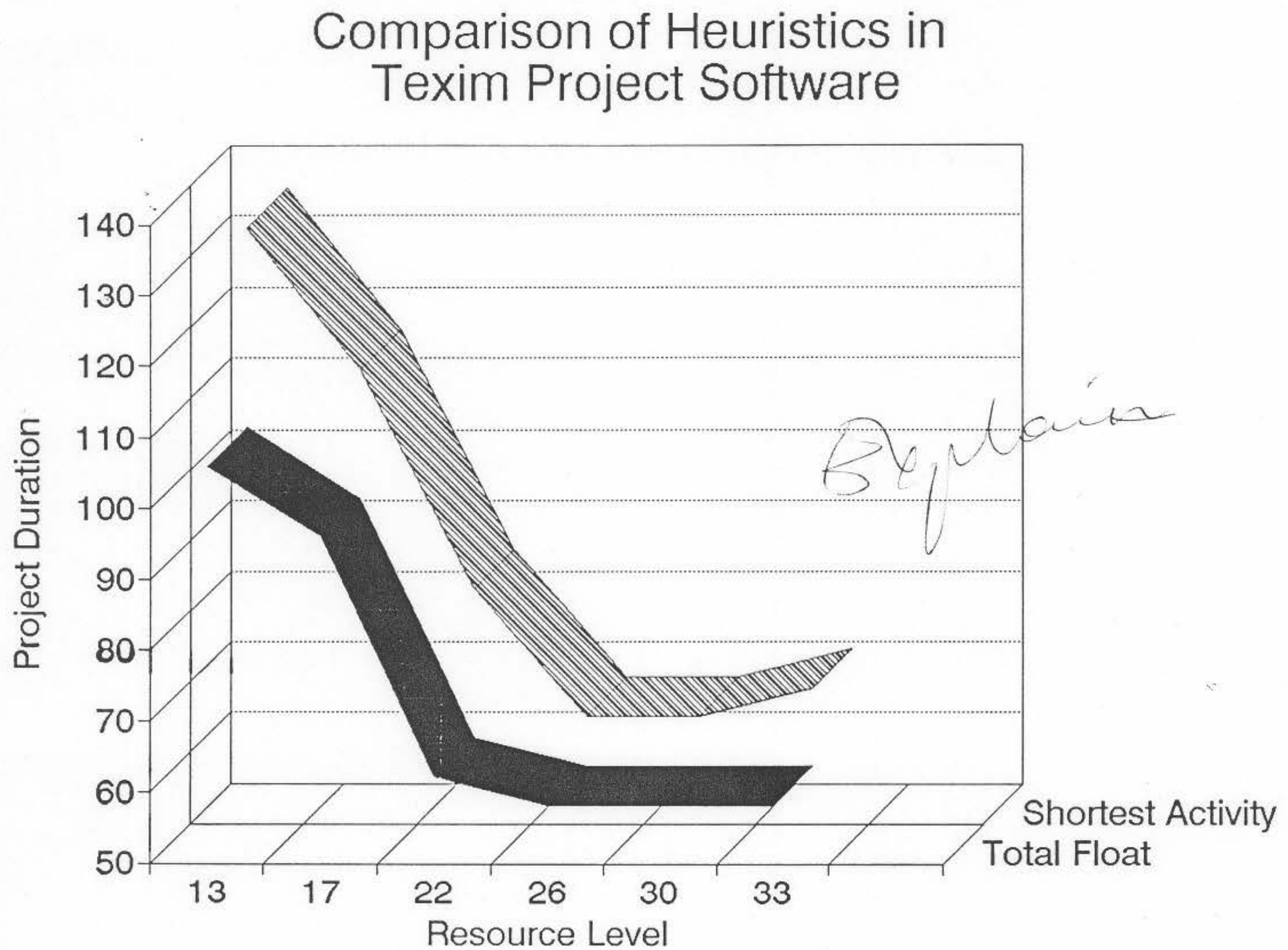

Comparison of Heuristics in
Texim Project Software
Project Duration
140
130
120
110
100
90
80
70
60
50
13
17
22
26
30
33
Resource Level
Shortest Activity
Total Float
Explain

Figure 9.

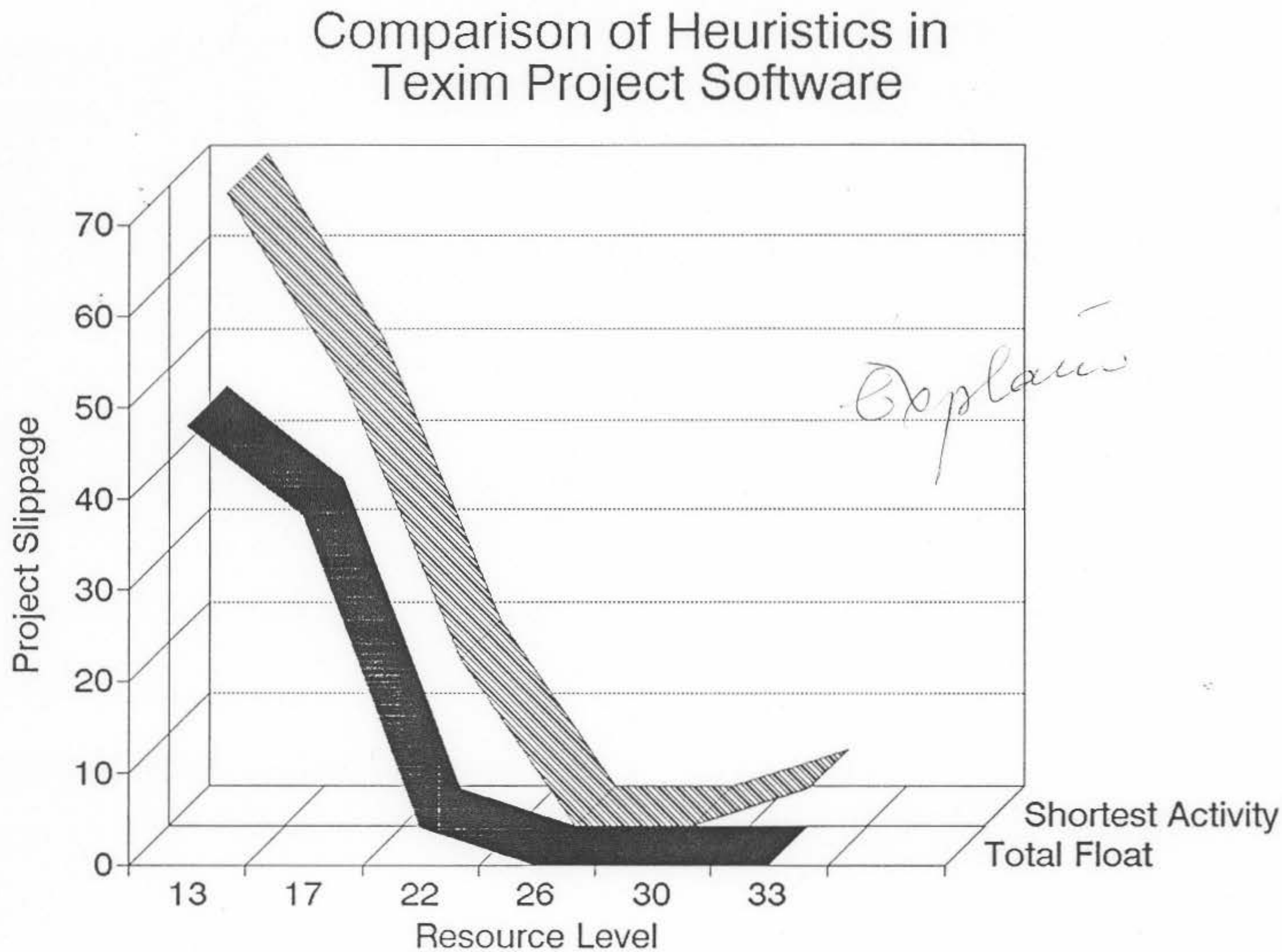

Comparison of Heuristics in
Texim Project Software
Project Slippage
70
60
50
40
30
20
10
0
13
17
22
26
30
33
Resource Level
Shortest Activity
Total Float

Figure 10.

IX. CONCLUSIONS AND RECOMMENDATIONS

* Since all the studies used different multi-project scheduling problems, no one scheduling method can be deemed universally best.

* If one heuristic stands above the rest in overall performance, it is the MINSLK heuristic (also called Total Float rule). However, there are circumstances where the MINSLK rule performs poorly, including the case where resources are loosely constrained. It is generally ill advised to follow one heuristic's schedule blindly, without regard to other possibilities.

* It is recommended that several heuristics be performed simultaneously to hedge against adopting poor schedules. Since most heuristics can be run in a short time, this is a minor inconvenience relative to the value it provides. Taking a little extra time developing the schedule can help you avoid putting all your eggs in one heuristic's basket.

* If the multi-project scheduling problems you face are fundamentally alike in structure and resource constrainedness, a heuristic evaluation process should be undertaken to determine which heuristic best fits your businesses' typical problem. It will likely turn out that 3 or 4 heuristics consistently perform better than the others. If so, these are the ones that should be run simultaneously during project scheduling.

* When evaluating the heuristics, it is very helpful to solve the problem using optimization techniques first. Then the heuristics can be graded as a percentage of the optimal solution. For example, if the objective is to minimize slippage, and the optimal duration of the project is 100 days, then the heuristic that schedules the project for a 110 day duration, would have a grade of 110% of the optimal duration. This information is valuable in that it provides instant feedback as to how close to optimal each heuristic is on your multi-scheduling problem. This benchmarking should be done periodically, or at least as often as the structure of your typical problem changes significantly.

* For a quick heuristic evaluation, several of the summary measures used by Kurtulus & Davis and later by Kurtulus & Narula can be calculated for the problem most typical to your business. Then you can make a best guess for the most applicable heuristic by choosing the best performer identified by Kurtulus, Davis, and Narula on problems with similar summary measures. It is worthy to note that Davis & Patterson did not identify their problem set at all, so general conclusions drawn from their work should not be assumed to extend to specific problems.

* It is a mistake to conclude that any one scheduling heuristic will always provide you with the best schedule.

* The objective of the project schedule should be clearly defined and accepted by all involved in the scheduling process. Often a minimum duration objective is accepted

because it is assumed that the shortest completion time will minimize costs and maximize profits. This is not always a valid assumption. When resource requirements are not smooth, the costs to meet varied staffing levels may go unconsidered. If these costs (cost to hire employees and cost to fire employees) can be defined, any schedule can then be evaluated

based on cost. Evaluations based on cost provide the level playing field to objectively determine which methods should be used for scheduling. The evaluations based on cost are most consistent with the goals of most businesses, and provide the clearest results.

X. REFERENCES

1. Mohanty R.P., Siddiq M.K., "Multiple Projects - Multiple Resource Constrained Scheduling: A Multiobjective Analysis", Engineering Costs and Production Economics, 18, 1989, pp 83-92

2. Badiru A.B., " Towards the Standardization of Performance Measures for Project Scheduling Heuristics", IEEE Transactions on Engineering Management, Vol 35, No 2, May 1988, pp 82-89

3. Davis E.W., Patterson J.H., " A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling", Management Science, Vol 21, No 8, April 1975, pp 944-955

4. Kurtulus I.S., Narula S.C., " Multi-Project Scheduling: Analysis of Project Performance", IIE Transactions, Vol 17, No 1, 1985, pp 58-66

5. Kurtulus I.S., Davis E.W., " Multi-Project Scheduling: Categorization of Heuristic Rules Performance", Management Science, Vol 28, No 2, February 1982, pp 161-172

6. Deckro R.F., Winkofsky E.P., Hebert J.E., Gagnon R., " A Decomposition Approach to Multi-project Scheduling", European Journal of Operational Research, 51, 1991, pp 110-118

7. Deckro R.F., Tsubakitani S., " A Heuristic for Multi-project Scheduling with Limited Resources in the Housing Industry", European Journal of Operational Research, 49, 1990, pp 80-91

8. The American Heritage Dictionary of the English Language, Dell Publishing Co., New York, p. 563