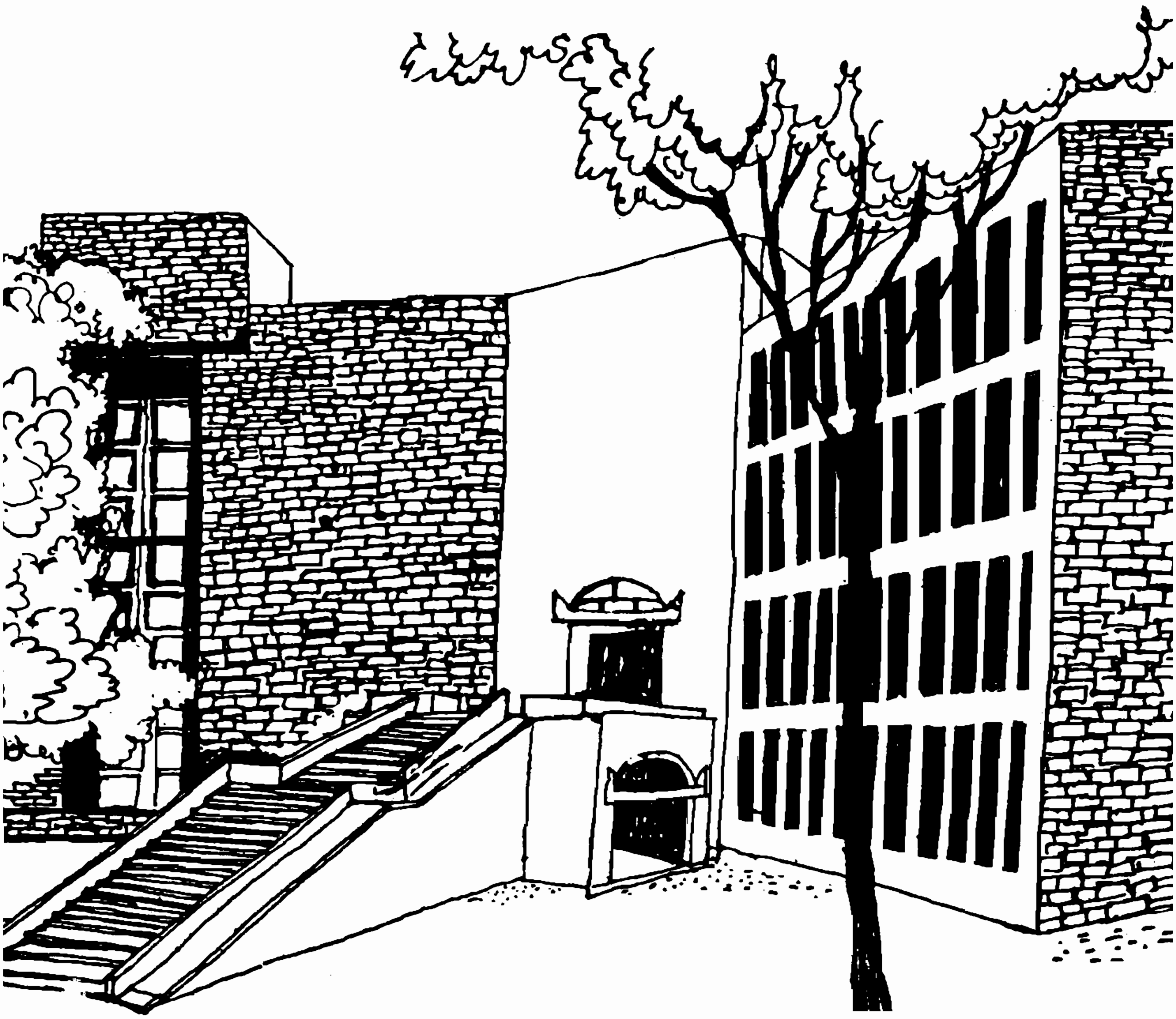
विद्याविनियोगाद्विकासः
IIM
AHMEDABAD

# Working Paper



GANTT CHART COLOURING

By

V. Venkata Rao

&

C. Syam Prasad



W P No. 955
August 1991



INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380 015
INDIA

# Gantt Chart Colouring

V. Venkata Rao
C. Syam Prasad

Indian Institute of Management
Ahmedabad 380015
1991

# Contents

# Gantt Chart Colouring

V. Venkata Rao and C. Syam Prasad

Abstract

This paper presents three rules for colouring of a Gantt chart that displays the machine and job schedules of a job shop. These rules have been used on the job schedules of a real job shop and the performance of the proposed rules has been measured with respect to four different data sets of the above job shop. Of the three rules the rule based on the jobs using the critical machines performed most satisfactorily according to the measure of performance used. This measure of performance is based on the principle of proper colouring according to which operations of the same job should receive the same colour, while neighbouring operations of a machine should receive different colours. The paper also proposes the concept of degree of neighbourliness which may form the basis for new heuristics for colouring.

## 1. INTRODUCTION

This paper deals with the problem of colouring a Gantt chart in interactive scheduling systems for job shops. The paper proposes three colouring rules and studies their performance on the schedules generated for a real job shop that manufactures a certain category of spare parts for automobiles. Some of the major concepts of this paper are drawn from those introduced in our earlier papers on interactive Gantt charting[2,3].

Job shop scheduling, in general, has been an active field for research for long. The recent focus of research in this field is on interactive systems, particularly on interactive Gantt charting [5]. The features of scheduling, unscheduling, and perturbing in an interactive Gantt charting system have already been discussed in [2]. The same paper has also introduced the scheme of proper colouring of a Gantt chart and has shown that the problem of properly colouring a Gantt chart with minimum number of colours is equivalent to the problem of graph colouring, which is an NP-complete problem.

A Gantt chart is said to be properly coloured if all the operations of a job receive the same colour while neighbouring operations belonging to different jobs on the same machine receive different colours. In practice, the colour pallette available in a system has a fixed number of colours, and, as mentioned before, it is very difficult to determine whether the colours available are adequate to properly colour the chart under consideration. Hence, in actual packages we have to necessarily resort to heuristic colouring rules which may violate the proper colouring principle. Therefore, it is essential to develop and test colouring rules which are simple to implement, which are quick in execution, and which do not violate significantly the principle of proper colouring.

In section 2 of this paper, we describe the relevant details of the job shop whose data has been used in this study. In section 3, we present the details of the software used for generating the schedules of the above job shop, and for assigning colours to the operation blocks of the corresponding Gantt charts. In section 4, we describe three different colouring rules and give the details of the data sets to which these rules have been applied. This section also discusses how the performance of a colouring rule is measured. Tables I and II present our results on the performance of the three rules on four different data sets of the job shop mentioned above. In section 5, we interpret the results of our experiments. Section 6 proposes the concept of degree of neighbourliness which can be used to devise improved heuristics. The last section, section 7, presents our concluding remarks along with some directions for future research.

## 2. DESCRIPTION OF THE JOB SHOP USED IN THE STUDY

The various colouring rules have been studied using the detailed job schedules developed for the job shop described in [1]. Each job has about five to twenty operations, each operation requiring a different machine centre. The machine routing varies from job to job. An operation on a machine once begun should be processed to its completion before another operation can be undertaken for processing on the same machine. There are about 1200 different products, and 65 different machine centres. The machine centres are classified into two categories, reporting and non-reporting. A reporting machine can not process more than one job at a time, where as a non-reporting machine does not have such capacity restriction. Because of their criticality, the reporting centres are in general the focus of attention of the schedulers. The detailed scheduling is done once a month. The number of jobs considered for monthly scheduling varies from 30 to 100. Some of these jobs are partly processed, where as others are not. Some jobs have their due dates within the planning horizon, where as the remaining have their due dates falling beyond the planning horizon.

## 3. DETAILS OF THE SOFTWARE

An interactive job scheduling system has been implemented for this job shop[1]. This system runs on PC/AT and has features for entering and retrieving machine and job details, for generating a detailed schedule of the jobs on the machines, and for exhibiting the output in various ways. Of particular interest to us in this paper is the procedure used by the package to develop machine and job schedules. For scheduling a given set of jobs, the user first specifies a certain job attribute using which the jobs are to be sorted. A job attribute could be its processing time, due date, number of operations, revenue to be earned etc. The sorting order (ascending/descending) is also specified by the user. In addition, the user gives a second attribute which is to be used to resolve ties in sorting. Then the system considers the jobs according to their sorted order and schedules them one after the other. Once a job is considered for scheduling, each of its operations is scheduled in sequence on its machine at the earliest possible time. Only after scheduling all the operations of a job, the next job in the sorted sequence is taken up for scheduling. The scheduling procedure involves many other details which are not relevant for this paper.

A separate program, GANTT, has been developed to convert a schedule produced by the JSS package into a Gantt chart format, and to display the Gantt Chart on the VDU. The GANTT program colours the operation bars of the chart according to a user specified colouring rule. The major inputs to the GANTT program are a schedule, the sorted sequence of jobs used in producing the schedule and a colouring rule.

The Gantt chart allocates one horizontal line to show the schedule of one machine. The time axis runs from left to right. This format, while quite suitable for depicting the schedule of reporting machines, poses some problems for showing the schedule of non-reporting machines, because non-reporting machines can process many jobs in parallel. If, during a time interval, more than one job is scheduled on a non-reporting machine, the GANTT package shows only the operation bar corresponding to the operation which finishes last. In case the user wants to see all the jobs completely during such a time interval on the machine under consideration, he/she has to bring on to the screen a separate window, which shows all the operations scheduled only on the machine under consideration, the overlapping operations coming one on top of the other.

## 4. EXPERIMENTS ON GANTT CHART COLOURING

Experiments are conducted by using 3 different colouring rules on schedules generated by using 4 different scheduling rules. Each of these 12 combinations have been used on 4 different job sets. For each experiment involving a combination of a job set, a scheduling rule and a colouring rule, the number of times the proper colouring property is violated is recorded. As the user is interested not necessarily in the schedule of all machines, the above measure is computed in two different ways: 1. by taking into account the schedules of only the reporting machines, 2. by taking into account the schedules of all the machines. These two sets of results are reported separately in tables I and II, each table consisting of 48 combinations.

### 4.1. Scheduling Rules

The four scheduling rules used differ from one another in the order in which the jobs are sorted before they are considered for scheduling by the JSS package.

i. RPT: The sorting order is shortest processing time job first; ties are broken using the smallest due date job first.

ii. DDT: The smallest due date job first; ties are broken using the shortest processing time job first.

iii. REV: Highest revenue job first; ties are broken using the smallest due date job first.

iv. NOP: Job with minimum number of operations first; ties are broken using the smallest due date job first.

### 4.2. Colouring Rules

The experiments are conducted using a PC/AT with a CGA monitor. The number of colours available for colouring an operation bar is 7. As per the proper colouring principle all the operations of the same job have to receive the same colour. All the following three rules ensure this property; while, only the last two have the explicit objective of minimising the possibility that two adjacent jobs on a machine are coloured alike.

i. Rule H: The colour code assigned to a job is obtained by the following hashing function of the job code.

colour code = (job code MOD 7) + 1

ii. Rule S: The colour codes are assigned to the jobs in the same order in which they were considered for scheduling by the JSS package. Thus the rule for colouring is

colour code = (position of job in the sorted sequence MOD 7) + 1

iii. Rule C: The colour codes are assigned using the same sorted sequence as above; but unlike rule S, this rule makes two passes through the sequence to assign colours to the jobs. In the first pass, only the jobs which are scheduled on the most critical machine are coloured in sequence and in the second pass the remaining jobs are coloured. The most critical machine is that which has the largest number of operations scheduled on it. This rule has been devised for two reasons:

1. Most of the violations of the proper colouring principle can be assumed to occur on the critical machine. Hence the colouring performance can be improved by treating separately the jobs using the critical machine.

2. In general the critical machine based scheduling rules are reported to yield better performance.

**Example:**

We discuss below how colours would be assigned to the Gantt chart of Figure 1 by colouring rules S and C.

Figure 1A shows a Gantt chart displaying 3 machines and 4 jobs. Assume that the colour codes are assigned according to Rule H.

Assume, further, that the scheduling rule sorts the jobs in the order (1,2,3,4), and that the colour codes available on the colour monitor are 3.

Instead of Rule H, if Rule S is used for colouring, then the colours assigned to the four jobs (1,2,3,4) would be (1,2,3,1) respectively (Figure 1B).

In the above figure, the number of operations scheduled on M3 is 4, which is larger than those on either of the two machines. Hence M3 is the critical machine. The four operations scheduled on M3 belong to jobs 3,2, and 1. If Rule C is used for colouring then the jobs that would be coloured first are those on the critical machine. These jobs would be assigned colours in the sequence in which they occur on the critical machine. Therefore job 3 would be assigned colour code 1, job 2 colour code 2 and job 1 would be assigned colour code 3. After the jobs on the critical machine are coloured, the remaining jobs could be coloured in the sequence in which they occur in the sorted list. In this example, the only remaining job is 4. Hence it would be colour code 1 (Figure 1C).

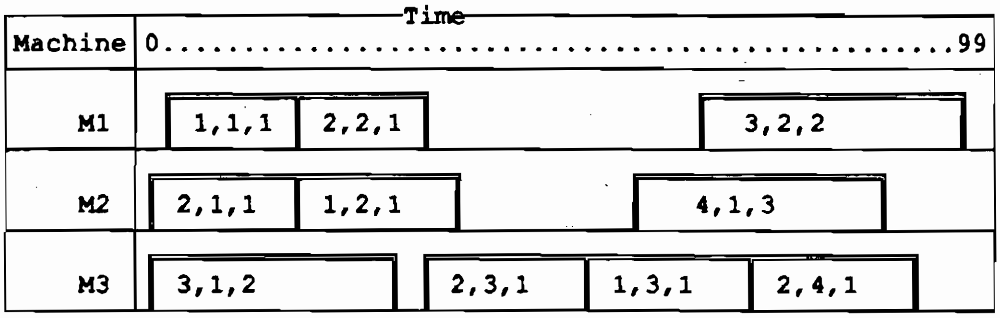



Figure 1A: A Gantt Chart displaying 3 machines and 4 jobs

Notes:

1. Legend for operation bar: Job#, Operation#,Color Code

2. No. of times proper colouring is violated = 4
   Total number of operation bars = 10
   Measure of performance = (4/10) * 100 = 40%

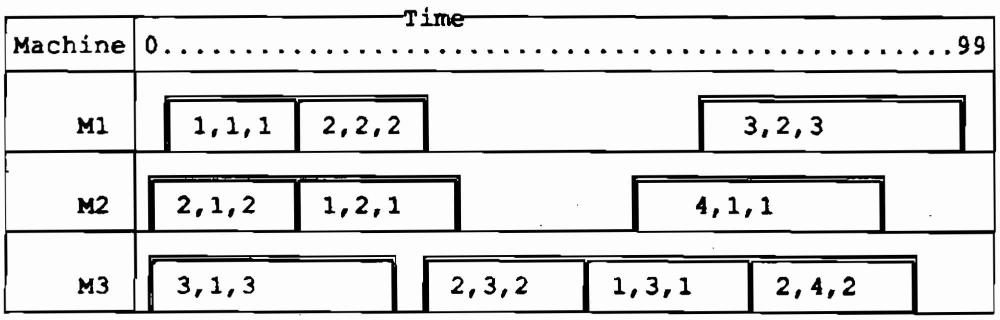



Figure 1B: Colour codes produced by colouring Rule S

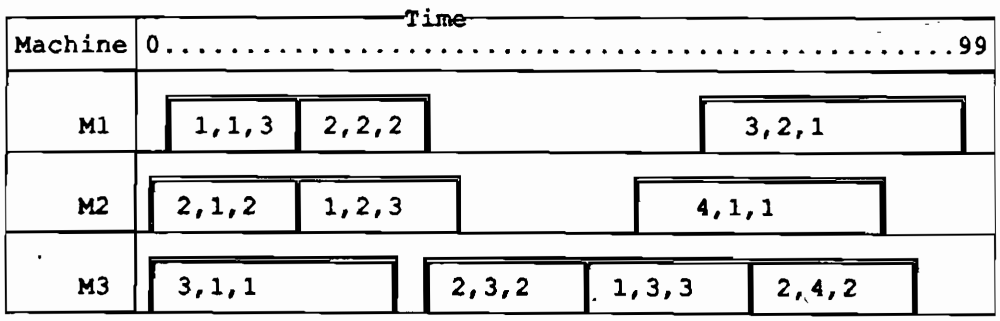



Figure 1C: Colour codes produced by colouring Rule C

### 4.3. Job Sets

Four different job sets pertaining to the data of the job shop mentioned above have been considered. These sets apply to job data of different months. The sets differ from one another specifically in the number of jobs and the total number of operations[Table III].

### 4.4. Measure of Performance

Whenever two operations occur on a machine such that:

i. the operations belong to two different jobs,

ii. the time at which the second operation begins is equal or ( in the case of a non-reporting machine) is less than the time at which the first operation ends, and

iii. the colours assigned to the two operations are the same

then we consider this occurrence as an instance of improper colouring. For each combination of a job set, a scheduling rule, and a colouring rule the value of n, where n = the total number of times proper colouring is violated is recorded and is considered to be the measure of performance of the colouring rule. This measure is expressed as a percentage of total number of operations scheduled in the job set under consideration. Thus the higher the measure the worse the performance of the colouring rule.

For each job set, the three rules are ranked, with rank 1 denoting the rule with the best performance and rank 3 denoting that with the worst performance. A rule is judged to have performed better than another rule in a job set, if it has obtained better performance value than the other in three or four scheduling rules.

## 5. ANALYSIS OF RESULTS

i. The scheduling rules do not seem to influence the effectiveness of a colouring rule significantly. For example, consider the ranks obtained by colouring rule C in the four job sets(Table I). It has obtained the best rank under each scheduling rule at least once. This is not surprising because the scheduling rules and the colouring rules differ in their objectives.

ii. Among the three colouring rules, rule H has performed consistently worse than the other two rules S and C. This behaviour is also quite consistent with what we expect, because H is totally dependent on a hashing function which is completely oblivious to the principle of proper colouring.

iii. Colouring rule C has performed better than rule S; in Table I, among the four job sets considered, rule C obtained the first rank thrice where as rule S obtained the first rank only once. In Table II rule C gets the first rank thrice and rule S twice. By and large there is a good correlation between the ranks obtained by the three rules between the two tables. This shows that the ranking of a rule does not change much whether we consider all the machines or only the reporting ones.

iv. As the number of jobs or operations in a job set increase we may expect the performance of a colouring rule to become poorer. This however has not happened in our experiments. For example, the measure of performance reported for job set 4 is not necessarily worse than the measure of performance for job set 3, even though the job set 4 has roughly three times as many jobs as job set 3. This is due to the fact that our measure of performance is a percentage (or, equivalently a ratio). The fact that the ratio has not increased as expected merely shows that the number of times the proper colouring is violated has increased less than proportionately with the increase in the total number of operations. An analysis of the absolute number of violations of proper colouring shows that this number indeed worsens with an increase in the number of operations or jobs.

v. In general the performance figures in Table II are worse than those in Table I. This is due to the fact that the schedule of Table I is a subset of the corresponding schedule of Table II. Therefore, the denominator in the performance indicator in both the schedules remains the same, where as the numerator of the schedule of Table II is greater than or equal to that of Table I.

vi. For operational purposes, all the rules have performed quite well if we consider only the reporting machines.

## 6. HEURISTIC USING DEGREE OF NEIGHBOURLINESS: SCOPE FOR FURTHER RESEARCH

The main idea behind rule S is that adjacency of two jobs in the sorted list is likely to lead to adjacency of these two jobs in the actual schedule also, and hence adjacent jobs in the sorted list should not be coloured alike. This assumption is not strictly correct because of two reasons:

i. Two adjacent jobs in the sorted list may not be neighbours in the Gantt chart if the two jobs have little commonality in their machines.

ii. Even if two jobs are adjacent on a Gantt chart their operation bars may be separated by a time gap.

Because of the above reasons, a better way of colouring would be to take into account the degree of neighbourliness of each pair of jobs on the Gantt chart before assigning colours to them. That is, colours should be assigned not before but after the schedule is actually developed. If a colour is to be repeated, it is better to assign it to two jobs with a low degree of neighbourliness rather than to those with a high degree of neighbourliness.

This naturally brings to light the question of how to define degree of neighbourliness of a pair of jobs. A reasonable measure for degree of neighbourliness of two jobs on a given chart is the total number of times the operations of the two jobs occur as neighbours when we consider the schedules of all the machines.

A colouring rule based on the above ideas is under implementation and empirical results on the performance of this rule are still not available.

## 7. CONCLUDING REMARKS

The results of this paper show that simple heuristic rules for colouring of a Gantt chart are quite effective in dealing with job schedules arising in practice. The rule based on degree of neighbourliness is still under development and is expected to give better performance than the rules already presented in the paper. This rule however may fare poorly in its execution time. For the data sets experimented with, the three rules H, S, and C consumed negligible amounts of execution time. For job shops with larger number of jobs the execution time however may be more. It is to be noted that as all the experiments of this study have been performed on the jobs of a specific job shop, all the conclusions strictly apply only to job shops similar to the one considered. We could have done the experiments by generating the data randomly;

but in that case we do not know whether the conclusions would apply to any real job shop at all.

Recently some new search techniques such as the simulated annealing technique have been used for solving the graph colouring problem [4]. It is worth while to use these techniques on the problem of Gantt chart colouring and to analyze their performance.

In reality the effectiveness of any colouring scheme depends on how exactly the user interacts with a scheduling system and with which parts of the schedule he/she deals with most frequently; about the exact nature of such interaction, we have very little information at present.

**Table I: Performance of Colouring Rules (Reporting Machines)**

| Rule | Job Set-1 | | | Job Set-2 | | | Job Set-3 | | | Job Set-4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | S | C | H | S | C | H | S | C | H | S | C |
| RPT | 1.39 | 0.69 | 0.00 | 2.37 | 0.47 | 0.95 | 3.91 | 0.00 | 0.78 | 2.43 | 0.27 | 1.35 |
| DDT | 1.04 | 1.04 | 0.00 | 3.79 | 1.90 | 1.42 | 3.91 | 0.00 | 3.13 | 1.08 | 3.24 | 0.81 |
| REV | 1.74 | 1.39 | 0.35 | 5.21 | 4.76 | 1.42 | 2.34 | 0.78 | 0.00 | 1.89 | 3.78 | 2.43 |
| NOP | 1.39 | 0.69 | 1.39 | 2.37 | 1.90 | 0.95 | 3.91 | 0.00 | 0.00 | 2.43 | 1.62 | 1.35 |
| Rank | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 1 | 2 | 3 | 2 | 1 |

**Table II: Performance of Colouring Rules (All Machines)**

| Rule | Job Set-1 | | | Job Set-2 | | | Job Set-3 | | | Job Set-4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H | S | C | H | S | C | H | S | C | H | S | C |
| RPT | 3.82 | 3.47 | 1.39 | 6.16 | 1.42 | 4.27 | 6.25 | 0.78 | 3.91 | 6.22 | 3.24 | 3.78 |
| DDT | 3.82 | 3.82 | 2.43 | 7.11 | 8.06 | 2.37 | 6.25 | 1.56 | 6.25 | 7.30 | 4.05 | 3.24 |
| REV | 4.86 | 4.51 | 2.08 | 9.48 | 1.42 | 3.79 | 7.03 | 3.13 | 0.78 | 7.57 | 4.59 | 4.05 |
| NOP | 4.17 | 2.43 | 3.82 | 5.21 | 2.84 | 1.42 | 10.16 | 1.56 | 1.56 | 4.32 | 4.32 | 3.78 |
| Rank | 3 | 2 | 1 | 2 | 1 | 1 | 3 | 1 | 2 | 3 | 2 | 1 |

Note: The numerical entries in Tables I and II are percentages. The rows represent the scheduling rules and the columns the colouring rules.

**Table III: Job set Details**

| Job Set | Number of Jobs | Number of Operations |
|---|---|---|
| 1 | 41 | 288 |
| 2 | 37 | 211 |
| 3 | 26 | 128 |
| 4 | 66 | 370 |

## REFERENCES

[1]. Venkata Rao V., Barua S. K., Madhavan T., and Syam Prasad C.(1990), An Interactive Job Scheduling System for an Engineering Company, National Seminar on Computer Aided Management, IIM, Ahmedabad, Feb 24, 1990.

[2]. Kaujalgi V. B. et al.(ed.), Computer and Information Sciences - Current Trends in Applications, Tata McGrawHill Publishing Co. Ltd, New Delhi, pp 271-285, 1989.

[3]. Venkata Rao V., and Syam Prasad C., Interactive Gantt Charting: Features, Data Structures and Algorithms, Working Paper 728, IIM, Ahmedabad, Feb 1988.

[4]. Hertz, A. and Werra, D.de, Using Tabu Search Techniques for Graph Colouring, Computing, Vol 39, pp345-351, 1987.

[5]. Jones Christopher V., The Three Dimensional Gantt Chart, Operations Research, Vol 36, No.6, pp 891-903, Nov-Dec 1988.

[6]. Christophides, N. (1975), Graph Theory: An Algorithmic Approach, Academic Press, New York, 1975.