Pergamon

International Journal of Project Management Vol. 17, No. 6, pp. 383–391, 1999
© 1999 Elsevier Science Ltd and IPMA. All rights reserved
Printed in Great Britain
0263-7863/99 $20.00 + 0.00

PII: S0263-7863(98)00052-0

Dynamic Priority–Dynamic Programming Scheduling Method (DP)^{2}SM: a dynamic approach to resource constraint project scheduling

Homayoun Khamooshi
Nene University College, Northampton (UCN), UK

The purpose of this paper is twofold: first to introduce and evaluate a dynamic priority scheduling model developed in this research for solving the resource constraint project scheduling problem and second, to introduce an improvement made upon the first model by cross-breeding Dynamic Programming with the Dynamic Priority Scheduling Method (DPSM). The second model, called the Dynamic Priority Dynamic Programming Scheduling Method [(DP)^{2}SM], aims at optimising the staged resource allocation decisions in DPSM. DPSM divides a project into phases (cycles), the length of which depends on the duration of the project and the period of clock cycle selected. The scheduling process starts by allocating resources to the first phase/cycle using a variety of policies, then the best schedule is selected based on an objective function. The process continues till all the activities are scheduled. In DPSM the interaction between phases is ignored while the decisions of each phase or cycle will affect all the remaining phases. Using (DP)^{2}SM it may be possible to improve the quality of a schedule and reduce the duration of a project by optimising the overall project schedule. © 1999 Elsevier Science Ltd and IPMA. All rights reserved

Keywords: project management, resource, schedule, quality, flexibility, dynamic programming

Introduction

Project Management (PM), a relatively new branch of management science, and also a new profession, was developed during the last 30 years. Network analysis was the only technique (among all the techniques for project management) widely known but even that little used. After the development of systems theory and systems management, project management has matured and developed into a set of tools and techniques, methods, strategies and disciplines that, when effectively and efficiently applied, contributes a lot to the achievement of goals and objectives.

Nowadays project management is a science and applying it has become a profession. Project management for professionals has six aspects:

- know-how: a set of skills for effecting certain procedures.
- know-what: practical understanding and direct immediate access to facts, findings and methods.
- know-whom: knowing who to turn to for referral, advice, help, assistance.
- know-why: justification for procedures.
- know-when: strategic timing, practical wisdom.
- know-where: understanding of work environment.

There are many factors affecting those aspects of project management namely the life cycle of a project, the organisation of a project, environmental factors and cultures, tools and techniques etc. This paper presents a critical evaluation of existing tools and techniques for project scheduling and a new approach is developed for resource constraint project scheduling which may help to respond to some of the problems with existing approaches.

Project scheduling in network based PM

One of the most important features of network based project management (PM) is that of management of resources, especially the broadly known case of project scheduling under resource constraints. Project planning and scheduling based on resource allocation has received increasing attention since the early 1960's and the result of this is the growing rate of publications in the literature and widespread commercial packages for project management, most of which have the ability to do resource allocation.

Davis[3] has categorised the approaches for resource allocation in network-based project scheduling into two distinct groups, namely heuristic approaches and optimal procedures. From the very first appearance of the PERT/CPM model, heuristic resource allocation procedures were introduced as the optimal procedures are not able to cope with complexities of the real world. Davis[3] believes that "Probably the first paper reporting the use of such procedure was given by Kelley[5]". Then the trend of introducing different procedures was started. The widely known technique RAMPS (Resource Allocation and Multi-Project Scheduling) was developed by CEIR Inc. and Du Pont Company in 1960. Another algorithm developed by Wiest[8] known as Scheduling Program for Allocating Resources (SPAR) and later on SPARTAN has been quite successful. Dar-el and Tur[2] developed an algorithm for scheduling a multi-resource project, where the objective was that of minimising the project cost for any project duration within a specified range.

Holloway *et al.* in their extensive research[4] considered the evaluation of heuristic procedures for solving resource constrained project scheduling problems. Although the procedure proposed by these researchers is an obvious improvement in project scheduling, the average figure of a 0.5% delay above optimal is not a reliable one and the fact that the efficiency of this method is a function of the number of resources and the complexity of the project, and hence the network could not be denied.

The research work on constrained resource scheduling for project planning and scheduling continued. In 1982 a paper published by Talbot[7] reported on research introducing procedures for formulating and solving a general class of nonpreemptive resource constrained project scheduling problems. In his research Talbot assumes that the duration of each job is a function of the number and types of resources committed to it. The approach is broad, so that evaluation of numerous time or resource-based objective functions is possible. Minimisation of project duration and project cost are among them. He claimed the procedure provided cost-effective optimal solutions for very small problems and good heuristic solutions for larger ones. The integer programming approach of Talbot, for solving a large class of nonpreemptive resource-constrained project scheduling problems in which job performance time is a function of resource allocation, is more theoretical than practical, and the few test problems which were used to indicate that the methodology can provide optimal solutions for small problems and good feasible solutions to larger ones are not proof of practicality. Since then many researchers have developed different heuristic procedures for project scheduling but none has really helped project managers to overcome the few basic questions they might ask.

Shortcomings of existing heuristic scheduling rules in theory and practice

The importance of resource constrained project scheduling problems will in future increase as the limitation on resources will be tighter. In spite of widespread use of this model in planning and scheduling small, medium and large size projects, there is still a large gap between the published research work done so far on the topic and the practical requirements of a project manager. It cannot be denied that the model is used as it is the only available technique in planning, scheduling and management of complex projects.

The basic idea behind heuristic-based resource allocation problem is ranking the activities based on one or more priority rule(s), then applying a simulating process (simulating passage of time) to schedule the activities; the selection of a priority policy has always been questioned. The priority rules introduced so far are too many and if one considers multi-priority rules, the number of permutations is very large. Research on constrained resource network-based scheduling problems is still under way and yet it has not been possible to classify resource constrained projects such that a suitable heuristic may be selected objectively.[9] Thus a procedure, a method or even a guide-line to make a firm, reasonable decision on a schedule to be selected and implemented is a matter of considerable importance. Unfortunately the project completion time is the only measure of performance used in most applications. Therefore a quantitative and objective approach toward assessment and appraisal of different schedules is a requirement. Such a procedure should reveal which schedule would be the most suitable and promising one. In order to formulate the approach toward selecting better solutions to the problem, a number of points should be considered:

- reliability of devised approaches, that is the probability of producing the best result if one decides to use a specific priority rule,
- practicality of repetitive runs of analysis, that is searching for the best schedule by using all the available priority rules,
- performance evaluation of existing heuristic methods,
- stability of the schedules, does the priority rule to be used change if there is a change in initial conditions?

Something which is inherent in heuristic-based resource allocation problems is the unpredictability of the result. It should be admitted that the outcome of a policy or priority rule is a function of many factors. These make the study of the situation so complicated that the research carried out so far covers only some of the factors or parameters. Therefore, accepting minimum total float (e.g.) as the best priority policy and applying it may sometimes produce surprisingly poor results. In general it could be said that there is still conflicting evidence as to which sequencing priority rule produces the best result even for a given problem condition or category of projects. Pascoe[6] in his research found minimum late finish time and minimum late start time heuristics about equal in superiority while Crouston[1], also testing multi-resource single projects, found minimum late start time clearly superior. This conflicting dilemma continued and reports of the research conducted show the fact that there exists no general agreement upon ranking of the heuristic rules applied to different models of resource allocation problems. The conflicting conclusions reported by these researchers may probably be explained in terms of the experimental design of statistical analysis carried out and underlying assumptions, that is the parameters considered in their evaluation could be different, the types and characteristics of the networks used as test problems certainly were different and moreover development of new heuristics has made the analysis more varied and complicated.

The important point is that most recent researchers have emphasised project duration as the determining factor while considering project delay/duration as the sole determining factor of preference of one schedule to the others does not seem reasonable or well justified. So one can conclude that:

- There exists no general agreement among researchers/practitioners concerning the best priority rule.
- There is not a significant relationship between priorities and characteristics of networks.
- If one wants to assess the performance of a priority rule/policy, what are the measures which could be used in evaluation of different policies?
- If we all agree upon and decide on a priority rule to be used as the best performing policy how reliable is that policy? In other words, is there a high probability of producing good schedules under a variety of conditions?

Though there are many good commercial packages that provide resource constrained scheduling facilities, sometimes the way they schedule a project is a commercial secret and the user usually is not aware of the complicated computational schemes inside the program. Moreover, most present commercial packages often fail to give a good heuristic solution. To find a better schedule the proposed procedure is to use a method of trial and error; this makes the evaluation process more complicated and subjective than objective. In practice large organisations use a tailor-made package or a commercial software package usually with a fixed predetermined scheduling policy. If this is true, which is the author's belief and experience, one should look for or devise a heuristic which incorporates some generally accepted common sense objectives. It is neither practical nor economical to apply all possible priority rules and then decide on the schedule, while the only objective in scheduling is minimising the project completion time.

Dynamic Priority Scheduling Method (DPSM)

When a project is to be scheduled the recommended procedure of most researchers[3] is to schedule the project by different priority rules and choose the one with a minimum project delay (percentage increase with respect to original critical path length), then during project life cycle or monitoring, schedule again and choose the best one. If, for example, in the planning phase "minimum total float" is selected as the best priority rule, then during monitoring "minimum late start" might be the preferred scheduling policy. So it could be concluded that there is no tendency or preference toward keeping the policy uniform throughout the project life cycle. Let us look at the problem from another point of view. Consider an engineering project from the very beginning to the end; it could be divided into few subnets(smaller networks) each including hundreds of activities. *Figure 1* shows an engineering project divided into three subnets.

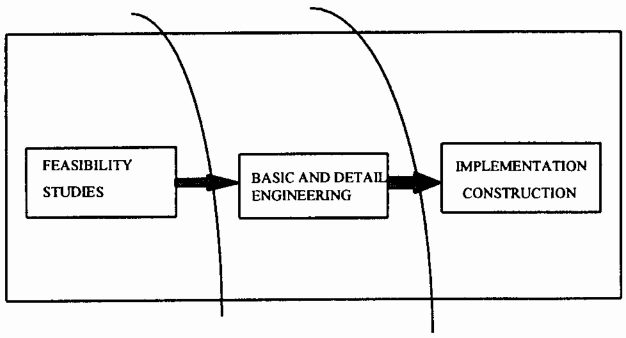


Figure 1 Project breakdown into phases/subnets

If the project in *Figure 1* is to be scheduled and a heuristic priority rule is to be applied all the way through the project, we might be sub-optimising the problem, while better solutions could be possible. The reason for that lies behind the fact that not necessarily the same scheduling policy would be the best for all subnets. If this logic is acceptable then why should we not divide a project into parts/phases and decide on each part applying selected priority rules? This clue led the author to a new approach to heuristic-based scheduling problem called the "Dynamic Priority Scheduling Method" or in short DPSM. In DPSM the project will be divided into sections; this could be a division based on different subnets if they run serially. that is, one follows the other, or a more general approach is to use a time section or so-called clock cycle to divide a project into parts, sections or phases. For each section or piece a search for the best schedule is carried out looking for the desired objective function (not necessarily the minimum duration). The procedure starts with the first section then continues to the second section and so on till all the activities are scheduled. A flow chart of DPSM computer program is shown in *Figure 2*. The objective function could be defined as desired. In this study four different options were considered:

1. Maximum number of activities scheduled per clock cycle
2. Maximum resource utilisation per clock cycle
3. Minimum fluctuation per clock cycle
4. Multi objective scheduling

When the criterion is to schedule as many activities as possible in a clock cycle then the procedure ignores the characteristics of the activity such as duration, resource requirements, criticality etc. In fact, these features come into the equation via the priority rules used for allocating resources to the activities. The selection of the activities is a function of priority rules used. If the priority rule used in a cycle is minimum duration, then the short activities are selected first. Hence, using five different priority rules in DPSM does not take into account the nature and characteristics of the activities. If the criterion is to maximise resource utilisation the model does not take into account the

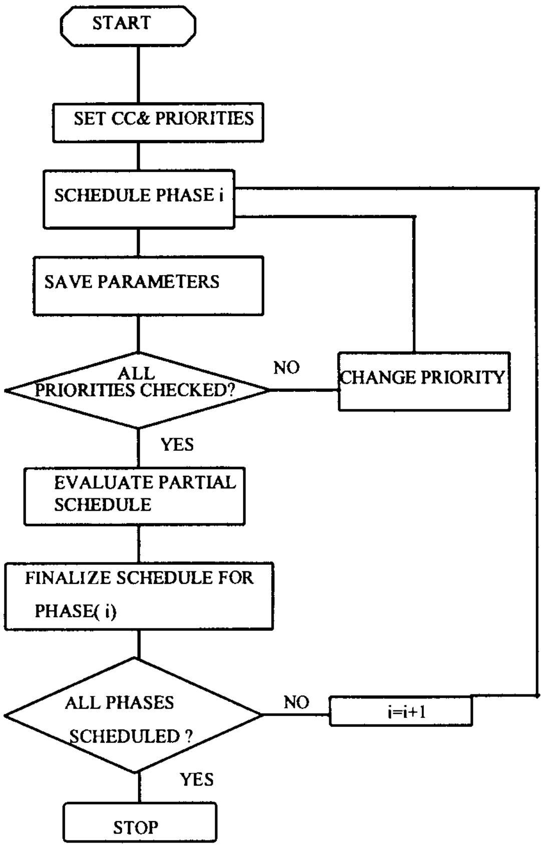


Figure 2 DPSM a simplified flowchart

importance of different resources and does not judge between various resources. The same applies to the the third and fourth objective function as well. However, one way forward is to give a kind of weighting to components of objective functions (options). The proposed model does not incorporate this feature.

Considering the case of unconstrained resource project scheduling, the objective is not to minimise duration but to produce a good quality schedule incorporating better use of resources, better distribution of float (flexibility), etc. DPSM could be used to search for and fulfil such objectives.

Dynamic Priority Dynamic Programming Scheduling Method $(DP)^2SM$

DPSM divides a project into phases (cycles) which depend on the duration of the project (length of critical path) and the clock cycle selected. The scheduling process starts by allocating resources to the first section or cycle using a variety of policies, then the best schedule is selected based on an objective function. The process continues over the second phase assuming the first phase decisions are made. The same algorithm applies to the third, fourth and the rest of the cycles till all the project is covered. The point to remember is that in DPSM the interaction between phases is ignored; in other words it does not aim at overall optimisation of the system. If the priority rule selected for the first phase is "late finish", the setting or schedule produced might be different from a schedule produced by "total float" if this was used as the priority policy. Thus, the succeeding activities are affected by the decisions made at the first phase or section (clock cycle). In general the decisions of each phase or cycle will affect all the remaining phases. It may be possible to improve the quality and duration of a schedule by trying to optimise the overall project schedule. Having that in mind, it seems that one can find a similarity between scheduling a project through phases using a number of priorities and the classical dynamic programming approach to stagewise decision making problems.

Dynamic programming is an approach that permits decomposing one large stagewise mathematical model into a number of smaller problems. Once all the smaller problems have been solved, we are left with one optimal solution to the large problem. So the technique can be applied to a decision problem that is multistage in nature. Though in project scheduling there is no evidence of stagewise decision making one cannot deny that resource allocation decisions made at each phase will affect the following decisions. If we consider each decision as a stage, we will end up with numerous stages. In DPSM the project is divided into phases (stages) which incorporate a number of resource allocation decisions within each cycle. In this proposed approach the stages are assumed to be the same as clock cycles/phases in DPSM which could exist naturally or be generated artificially. The most important feature of the dynamic programming approach is that the smaller problems or stages are not considered to be independent and the impact of decisions made in each stage on the remaining stages is taken into consideration. Let us consider the DPSM model and see how we can apply dynamic programming technique to improve the solutions (schedules) produced. If we use a clock cycle to divide the project into three sections (equivalent to three stages in dynamic programming) and make use of three different priority rules, the schematic presentation of the model will look like *Figure 3*.

CC_{ij} shows the clock cycle i with priority policy j. It is important to note that, contrary to the classical dynamic programming approach for solving large stagewise mathematical models, in applying dynamic programming to project scheduling the solutions produced for each stage are not based on an optimal approach, but partial schedules are produced by heuristic scheduling policies and as such the overall solution should not be expected to be an optimal one. However, by developing a dynamic programming model for project scheduling the schedules may be improved.

For demonstrating this model further, a case of three priority rules is used in the pictorial presentation of *Figure 4* to show the compatibility of the problem with dynamic programming model. Also a clock cycle is selected so that the project is divided into three stages as the smaller the number of phases the easier to schedule the project.

In developing a dynamic programming model for resource constrained scheduling problems there may be some complications in terms of the number of priority rules to be used and also the duration of the clock cycles. The conclusions drawn from the author's research make it clear that there is no point in incorporating a large number of priorities; including a maximum of five good priority rules should suffice. *Figure 5* shows a simplified flowchart for $(DP)^2SM$.

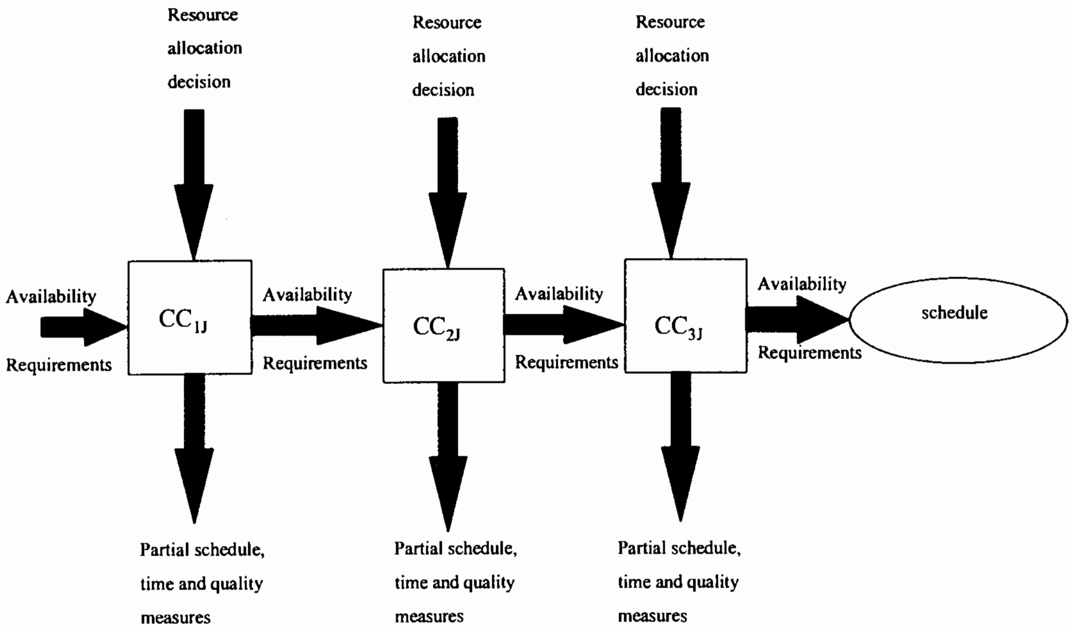


Figure 3 DPSM analogy with dynamic programming stages

Experimental design and data collection

The general policy of the experiment was to study the practical requirements of project planning and scheduling. Therefore, although artificial networks are also used in the experiments the intention has been to use large projects with multi-resource requirements. In this research a comparative study of performance of different heuristic rules is carried out while the measure of performance or efficiency of a scheduling rule has not only been project duration or project delay, but also other measures are introduced (see Appendix A for detailed discussion on performance measures). Also it is important to note that the objective of the comparative study was not to compare traditional priority rules but to study the performance of newly developed models. Data collection was done in three phases: the collection of data through generation of artificial networks; selection of a number of test problems through the survey of literature; and collection of some real-life projects. In all, 30 test problems were collected using a program generating networks. A number of networks used in the experiment were collected through literature surveys. In this section the problems and projects presented by different authors and researchers in the books and papers were collected. These problems were about 30 in number. Quite a large number of real life projects were collected through the "applied project planning and scheduling courses" conducted by the author in a variety of industries. So these projects are the ones which could be counted on much more than the artificial and theoretical networks. The number of real-life projects was 40. The number of activities in a project varied from 10 to 300 and the number of resources from 1 to 15.

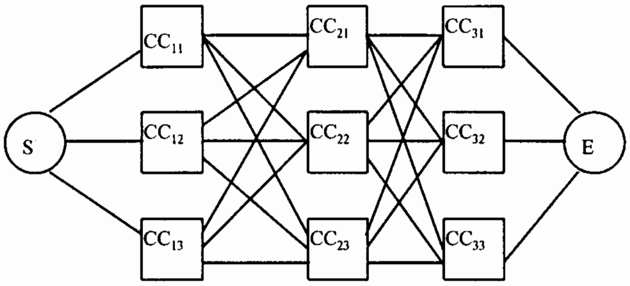


Figure 4 (DP)^{2}SM, a case of 3 priorities with 3 clock cycles analogue to travel salesman problem

Selecting the priority rules for the experiment was a difficult task. Diversity of point of views of different researches made it more complicated to decide upon the priority rules to be used in the experiment. Therefore two approaches were made. Firstly, a survey of literature and secondly, a study of project management packages available to the author (on the market). Then the best of all were selected. The priority rules selected to be used in the experiment were among the best reported in the literature. The priorities used include: (1) Total Float; (2) Early Start, (3) Late Finish; (4) Dependent Activities; (5) Critical Resource Requirement; (6) Total Resource Demand; (7) Late Start; (8) Remaining Duration; and (9) Maximum Resource Demand.

Summary of experiments and analysis of results

The newly developed heuristic approach (DPSM) was applied to 100 resource-constrained project scheduling cases through a number of experiments and the following results were obtained:

1. The clock cycle was set to a single time unit: that is, all the priorities in DPSM were tested against the objective function for each and every time unit. Though DPSM produced reliable solutions it did not perform very well. It appears that a short clock cycle generates too many changes of priority policy over the project duration, leading to longer project completion time.
2. The length of the clock cycle was increased to include the cases of 10, 25, 50 and 75% of the original critical path length of the project. As a result the performance of DPSM improved and its options

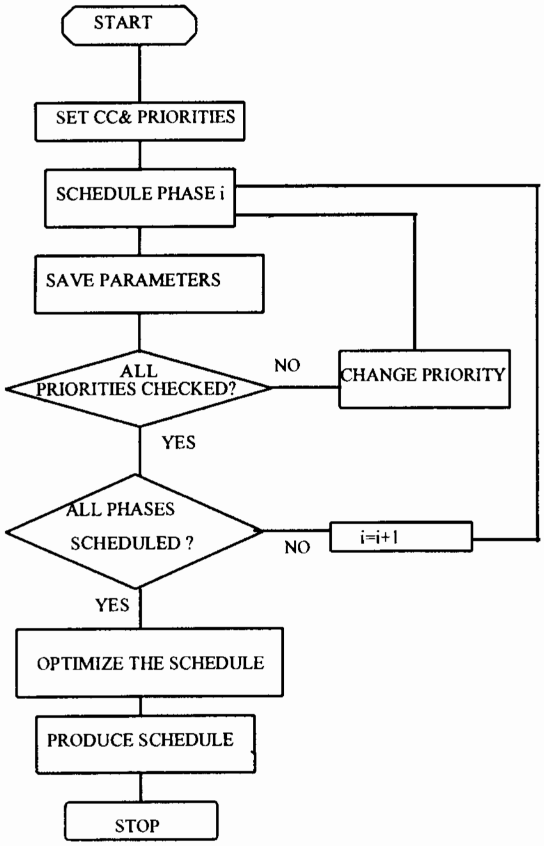


Figure 5 $(DP)^2SM$ a simplified flowchart

almost totally outperformed the other priority rules (for more detail of results see Appendix B).

3. In order to test the effect of increasing the number of priority rules to be used in DPSM the model was modified to incorporate nine priority rules (DPSM10) in contrast to five priority rules (DPSM5) used originally. The results showed that although some improvement may be gained, it is not significant and is not worth the price to be paid for it. Comparing the results of scheduling the same 100 projects by the two versions of DPSM (DPSM5 and DPSM10) showed that the number of priorities used in DPSM should not be increased.
4. In order to test how DPSM deals with larger complex projects, that is, projects with larger numbers of activity and more parallel paths, it was applied to a set of 40 practical projects. The results showed that the performance of DPSM was improved: verifying that DPSM has a much better chance of finding a good combination of activities in longer clock cycles and as a result gives better selection of priorities over the project duration.
5. Regarding the DPSM options, four different objective functions were tested. These include maximum number of activities scheduled per clock cycle, maximum resource utilisation per clock cycle, minimum fluctuation per clock cycle and a multi-objective function incorporating the first three objectives. DPSM option 2, that is "maximum resource utilisation per clock cycle", produced better solutions, though DPSM option one was shadowing it. It is to be mentioned that the last two options (option 3 and 4) were not aiming at minimising project duration so it was not expected that these options would produce very good schedules in terms of time performance measures.
6. Considering the clock cycle, analysis of the results of the experiments showed that the recommended clock cycle should be more than 50% of critical path length, preferably less than 75% of original critical path length of the project (2 to 3 clock cycles). The shorter cycles create lots of entropy or change of priority rules over the project duration which may end up with a longer project completion time.
7. Considering time as the most important factor, DPSM produced the minimum project completion time 51% of the time which is 14% more than the best traditional priority rule. It also produced the first or second best solution 79% of the time which is 15% better than the performance of the best single priority rule. On the other hand, DPSM is capable of producing unique best solutions.

Though this research did not incorporate the use of $(DP)^2SM$ in the experiments conducted, it goes without saying that by optimising DPSM schedules the best possible combination of priority rules will be implemented, thus the schedule is bound to improve, or at least remain the same.

Conclusion and further research

The diversity of heuristic-based resource-constrained project-scheduling approaches and the large number of priority rules used in the literature, as well as the significant difference between duration and characteristics of the schedules produced, led the way to the evolution of an approach called the Dynamic Priority Scheduling Method: DPSM and $(DP)^2SM$. They not only incorporate all well performing priority rules but also there is a higher chance of producing better schedules than any other priority rule.

Using DPSM one may be more certain that the schedule produced for a project is better in terms of time and other characteristics. The approaches available for selecting an appropriate policy for scheduling a project are neither practical nor reliable. DPSM is based on the notion that a project can be partitioned into a number of clock cycles and there is no obligation to use a single policy all the way through all the cycles over project duration.

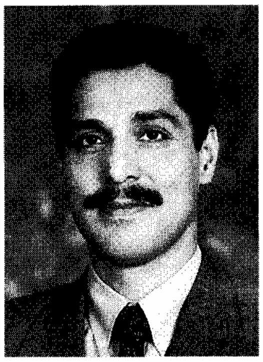
DPSM can produce good schedules without the need for checking any other priority policy. If one has the time and resources to spare on a search approach for a better schedule (shorter duration, better indices), it is recommended that DPSM is used with a variety of clock cycles instead of going for traditional priority policies. The schedules produced by DPSM are generally better, or at least as good as the best which can be produced by a variety of scheduling policies. In spite of being simple, attractive and to some extent promising, $(DP)^2SM$ needs more elaboration, consideration and analysis before any final conclusions can be reached. An important point to mention is that with the increasing speed, capacity and capability of microcomputers, there is no doubt that computer time and memory is not going to be a barrier to applying DPSM and $(DP)^2SM$ to real life projects.

There are a number of questions which could be raised demanding further research and more work which this paper has not attended to (e.g. if the criterion is to schedule as many activities in a clock cycle as possible what does this imply about the relative duration, importance etc. of the activity? If the criterion is to maximise resource utilisation how should the system judge between various resources?). The same queries could be raised when using minimum fluctuation as the objective function. Further research is needed to incorporate the above issues within the Dynamic Priority–Dynamic Programming Scheduling Method to resolve some of the practical problems associated with the model.

Further recommended readings: A Battersby (1978) *Network Analysis for Planning and Scheduling*, Macmillan. New York; Edward W Davis (1975) Project network summary measures—constrained-resource scheduling. *AIIE Transactions*, Vol. 7, No. 2, June, pp. 132–142; SE Elmaghraby (1977) *Activity Network: Project Planning and Control by Network Models*. John Wiley, Chichester, pp. 168–226; MC Grool *et al.* (1986) *Project Management in Progress—Tools and Strategies for the 90s*. Elsevier; FL Harrison (1981) *Advanced Project Management*. Gower, London, pp. 192–202; IBM Corporation (1988) Project Analysis and Control System (PROJACS) Program Reference Manual, IBM; IS Kurtulus, SC Narula (1985) *Multi-Project scheduling: analysis of project performance*. IIE Transactions, Vol. 17, No. 1. March.

References

1. Crouston, Wallace BS, Decision CPM: network reduction and solution. *Operational Research Quarterly*, 1970, **4**, 26–36.
2. Dar-el, EM and Tur, Yigal, A multi-resource project scheduling algorithm. *AIIE Transactions*, 1977, **9**(1), 44–52.
3. Davis, EW, Project scheduling under resource constraints-historical review and categorisation of procedures. *AIIE*, 1973, **Dec**, 297–311.
4. Holloway, Nelson and Surphongschai, Comparison of a multi-pass heuristic decomposition procedures with other resource-constraint project scheduling procedures. *Management Science*, 1979, **25**(9), 862–872.
5. Kelly JE, Critical Path Planning and Scheduling. Eastern Joint Computer Conference, Boston, Dec, 1959.
6. Pascoe TL, An Experimental Comparison of Heuristic Methods for Allocating Resources. Ph.D. Thesis, Eng. Dept. Cambridge University, 1965.
7. Talbot, F Brian, Resource-constrained project scheduling with time-resource trade offs: the nonpreemptive case. *Management Science*, 1982, **28**(10), 1197–1210.
8. Wiest JD, Levy FK, A Management Guide to PERT/CPM. 2nd edn, Prentice Hall Englewood Cliffs, NJ, 1997.
9. Willis, RJ, Critical path analysis and resource constrained project scheduling—theory and practice. *EJOR*, 1985, **21**, 149–155.
10. Haralakis, G, Objective evaluation of nonstorable resource-scheduling profiles. *PM*, 1985, **3**(4), 199–207.



Dr Homayoun Khamooshi graduated in Mechanical Engineering from the Abadan Institute of Technology, Iran in 1977. He was then awarded a scholarship and received his Masters degree in Industrial Engineering and Management from Asian Institute of Technology, Thailand in 1979. He was awarded a Ph.D. in Management Science from Lanacaster University, Lanacaster, UK in 1994. Dr Khamooshi is currently a senior lecturer at Nene University College Northampton, UK. Prior to joining UCN in 1993 he spent more than a decade as a project engineer, consultant and OR/MS lecturer within oil industry in the Middle East. His research interests are project management, PRAM, system dynamics and simulation.

Appendix A

Measures of performance for comparing schedules

The objective evaluation of different schedules irrespective of scheduling rule or policy is a practical requirement. So far most researchers have used the project duration as the only determining parameter in assessing a scheduling policy while other measures such as flexibility and stability of a schedule could be developed to ease and enhance the selection process.

Table 1 Performance of DPSM compared with other heuristics

PR/RULE	PCTOP	TFIOS	FFFRS	TFFRS	UNIQUE	FIRST	SECOND
P1	109.46	2.56	2.61	3.02	0	27	24
P2	114.56	2.58	3.16	3.45	0	23	22
P3	108.53	2.53	3.11	3.29	0	37	27
P4	113.43	2.67	3.05	3.65	0	28	10
P5	115.05	2.66	3.25	3.71	0	21	11
P6	112.44	2.52	3.19	3.61	2	26	16
P7	109.46	2.56	2.61	3.02	0	27	24
P8	116.66	2.66	3.58	4.27	3	16	15
P9	118.07	2.69	2.99	5.41	0	13	15
D1	109.37	2.64	2.67	3.17	0	33	23
D2	111.32	2.61	2.66	3.13	3	27	17
D3	109.47	2.56	2.61	3.02	0	27	24
D4	109.66	2.59	2.92	3.26	1	31	21
D5	108.47	2.51	2.89	3.34	8	35	24
D6	108.54	2.53	3.03	3.22	11	37	19
D7	109.29	2.54	2.62	3.01	2	33	21
D8	109.47	2.54	3.01	3.31	3	34	24
D9	108.74	2.52	2.93	3.32	2	34	24
D10	109.15	2.46	2.88	3.21	5	35	19
D11	108.86	2.52	2.61	2.97	4	39	17
D12	108.95	2.58	3.09	3.26	2	36	24
D13	107.73	2.49	2.66	3.25	6	50	29
D14	107.25	2.49	2.64	3.16	7	51	26
D15	108.81	2.52	2.65	3.01	1	42	18
D16	109.02	2.53	2.98	3.21	1	37	27
DOP	104.93	2.41	2.69	2.89	0	95	5

The term stability has been utilized in different context and varied meaning by few researchers. Willis[9] believes that a schedule should be stable in the sense that enough float should be available to the activities, so that if a delay for any reason occurs, the change does not propagate through the network and consequently changes the whole structure of a network.

The author proposes the term "flexibility" for the distribution of float and stability as stable sequence/order of activities within a schedule. The point is that if the schedule is flexible: that is, distribution of float is optimum, to some extent stability of schedule is guaranteed, since small changes will be buried with float consumption and the change does not propagate throughout the network, and only the disturbed activity(ies) will be rescheduled. It might be preferable to have a schedule which is more flexible and poor in resource utilization than having an absolutely tight and sensitive schedule which would be reshuffled by minor changes.

Now let us look at the measures which could be used in comparing schedules. Most of the measures used in this study are intoduced by the author while some are the traditional measures related to project duration. Let us define fluctuation by the following formula:

$$\mathrm{FL}_{ij+1} = \mathrm{abs}\left(\frac{\mathrm{Ru}_j - \mathrm{Ru}_{j+1}}{\mathrm{Ra}_j}\right) \tag{1}$$

where $\mathrm{FL}_{i,j+1}$: fluctuation at period $j + 1$ for resource i; Ru_j: resource i utilization at period j; Ra_j: resource i availability at period j.

Now the total fluctuation of resource i could be defined as the sum of all fluctuations from periods 1 to $n - 1$, that is

$$\mathrm{TFL}(i) = \sum_{j=0}^{n-1} \mathrm{FL}_{ij+1} = \mathrm{FL}_{i,1} + \mathrm{FL}_{i,2} + \mathrm{FL}_{i,3} + \ldots + \mathrm{FL}_{i,n-1} \tag{2}$$

Let us define the fluctuation index of resource i as:

$$\mathrm{FI}(i) = \frac{\mathrm{TFL}(i)}{\mathrm{step}} \tag{3}$$

where 'step' is the number of falls and rises or ups and downs in a profile (resource loading chart). That is if the utilization of a resource in period j is different from that of period $j + 1$, step is incremented by one unit. Total fluctuation index is defined as the average of fluctuation indices of different resources utilized in a project. That is:

$$\mathrm{TFI} = \frac{\sum_{i=1}^{\mathrm{NR}} \mathrm{FI}(i)}{\mathrm{NR}} = \frac{\mathrm{FI}(1) + \mathrm{FI}(2) + \mathrm{FI}(3) + \ldots + \mathrm{FI}(\mathrm{NR})}{\mathrm{NR}} \tag{4}$$

where NR is the number of resources utilized in a project. This parameter (Total Fluctuation Index) is a measure of efficient use of resources in a multi-resource project scheduling case.

Another measure of quality of a schedule, which can be used by planners as a yardstick in deciding upon selecting or rejecting a schedule or as a measure of performance of priority rules, is flexibility. In his paper, Harhalakis[10], proposed a ratio which is called total float efficiency or remaining total float ratio and formulated as follows:

$$\mathrm{E}_{\mathrm{tf}} = \frac{\sum_{1}^{p} \mathrm{TFo}_i - \sum_{1}^{p} \mathrm{TFc}_i}{\sum_{1}^{p} \mathrm{TFo}_i} \tag{5}$$

where E_{tf}: remaining total float ratio; TFo_i: total float of activity i prior to resource scheduling; TFc_i: total float of activity i used during resource allocation; p: total number of network activities.

It is true that the above mentioned ratio is a measure of flexibility, but there are some important drawbacks to it. The definition given in *Equation (5)* is not valid for cases when the activities are delayed beyond the point where all available floats are used in scheduling and the results should be treated with some reservations. Therefore some modification is needed to generalize *Equation (5)*; on the other hand would it not be better if we used free float instead of total float? Free float seems to be a more appropriate parameter as it is defined as the time period(s) by which an activity could be delayed without changing the starting point of dependent activities. Therefore, a new measure of performance of a scheduling policy is introduced and named 'remaining free float ratio' or 'free float scheduling efficiency' and formulated as follows:

$$\mathrm{E}_{\mathrm{ff}} = \frac{\sum_{1}^{p} \mathrm{FFo}_i - \sum_{1}^{p} \mathrm{FFc}_i}{\sum_{1}^{p} \mathrm{FFo}_i} \tag{6}$$

E_{ff} ranges between 0 and 1 for some cases and is not valid otherwise as was described for E_{tf}. A value of sum of $\mathrm{FFc}_i = 0$ will result in a value of 1 for E_{ff}, which indicates a 100% flexibility compared with the initial schedule (maximum or unlimited resource availability) or assuming that the delay of none of the activities caused by limited resources is beyond their respective free float. A value of the sum of $\mathrm{E}_{\mathrm{ff}} = 0$ indicates that all free floats of activities have been utilised during scheduling and any further delay will inevitably cause some changes to the scheduling time of dependent activities. The fact is that these two measures E_{tf} and E_{ff} are rough approximations to the flexibility of a schedule. Assuming a large value for float of an activity in a network and small values for the rest of the activities, if free float of activities with small values are consumed during scheduling, still one may get a good value for E_{ff} and similarly for E_{tf}. It is true that the flexibility of an activity is also a function of duration of that activity; that is a large duration activity with a very small float value is not considered to be very flexible. Let us define:

$$\mathrm{FLEXI}(i)_{\mathrm{tf}} = \frac{\mathrm{TF}(i)}{\mathrm{DR}(i)} \tag{7}$$

$$\mathrm{FLEXI}(i)_{\mathrm{ff}} = \frac{\mathrm{FF}(i)}{\mathrm{DR}(i)} \tag{8}$$

where $\mathrm{TF}(i)$: total float of activity i; $\mathrm{FF}(i)$: free float of activity i; $\mathrm{DR}(i)$: duration of activity i; $\mathrm{FLEXI}(i)_{\mathrm{tf}}$: flexibility index of activity i based on tf (total float); $\mathrm{FLEXI}(i)_{\mathrm{ff}}$: flexibility index of activity i based on ff (free float).

The third and fourth measures of flexibility are defined as:

$$\mathrm{TFFR} = \frac{\sum_{1}^{P} \mathrm{FLEXI}(i)_{\mathrm{tfs}}}{\sum_{1}^{P} \mathrm{FLEXI}(i)_{\mathrm{tfo}}} = \frac{\mathrm{FLEXI}(1)_{\mathrm{tfs}} + \mathrm{FLEXI}(2)_{\mathrm{tfs}} + \mathrm{FLEXI}(3)_{\mathrm{tfs}} + \ldots + \mathrm{FLEXI}(P)_{\mathrm{tfs}}}{\mathrm{FLEXI}(1)_{\mathrm{tfo}} + \mathrm{FLEXI}(2)_{\mathrm{tfo}} + \mathrm{FLEXI}(3)_{\mathrm{tfo}} + \ldots + \mathrm{FLEXI}(P)_{\mathrm{tfo}}} \tag{9}$$

$$\mathrm{FFFR} = \frac{\sum_{1}^{P} \mathrm{FLEXI}(i)_{\mathrm{ffs}}}{\sum_{1}^{P} \mathrm{FLEXI}(i)_{\mathrm{ffo}}} = \frac{\mathrm{FLEXI}(1)_{\mathrm{ffs}} + \mathrm{FLEXI}(2)_{\mathrm{ffs}} + \mathrm{FLEXI}(3)_{\mathrm{ffs}} + \ldots + \mathrm{FLEXI}(P)_{\mathrm{ffs}}}{\mathrm{FLEXI}(1)_{\mathrm{ffo}} + \mathrm{FLEXI}(2)_{\mathrm{ffo}} + \mathrm{FLEXI}(3)_{\mathrm{ffo}} + \ldots + \mathrm{FLEXI}(P)_{\mathrm{ffo}}} \tag{10}$$

where TFFR: total float flexibility ratio; FFFR: free float flexibility ratio; $\mathrm{FLEXI}(i)_{\mathrm{tfs}}$: flexibility index of activity i after scheduling based on total float; $\mathrm{FLEXI}(i)_{\mathrm{tfo}}$: flexibility index of activity i before scheduling based on 'TF'; $\mathrm{FLEXI}(i)_{\mathrm{ffs}}$: flexibility index of activity i after scheduling based on free float; $\mathrm{FLEXI}(i)_{\mathrm{ffo}}$: flexibility index of activity i before scheduling based on 'FF'.

It can be noted that values of all the above ratios remain within 0 and 1 if and only if the primary or the original floats are not fully utilised during resource allocation. Otherwise the values might be more than 1; that is, limitation on resources which leads to delay might bring more flexibility to the system (if resources could be provided). Generally the larger the value of flexibility index the better the schedule. In a constrained resource scheduling problem this may be obtained by lengthening the project duration which is a high-cost strategy and is not really favorable. The definition of TFFR (*Equation (9)*) shows that the value of flexibility is dependent upon the original distribution of float as well as the resource allocation schedule. To summarize, the four measures of flexibility, namely E_{tf}, E_{ff}, TFFR and FFFR could be used to evaluate the performance of different scheduling rules (policies). Obviously the schedule with the highest value of any of the four measures would be preferred (other things being equal).

The appraisal of performance of scheduling priority rules (heuristics) in this study is based on two aspects: time and quality factors. The time factor indices (measures) used are: (1) PCTOP: Project Completion Time Of Project; (2) FIRST: No. of times the minimum

PCT was produced by the rule, (3) SECOND: No. of times the second best solutions was produced, (4) UNIQUE: No. of times the unique best solution was produced. The quality factor indexes are: (5) TFIOS: Total Fluctuation Index of Schedule, (6) FFFRS: Free Float Flexibility Ratio of Schedule, and (7) TFFRS: Total Float flexibility Ratio of Schedule.

Appendix B

Analysis of results: case of DPSM applied to 100 projects

The results were obtained by scheduling each one of 100 projects using 16 different priority rules/policies. These priority rules/policies are defined below:

1. P1 to P9: nine of the best traditional priority rules;
2. *D1*: DPSM option 1 using a clock cycle equal to 10% of original critical path length, e.g. if the duration of the project is 200 days, the clock cycle is set at 20 days and the objective is to maximise number of activities scheduled within 20 day cycles.
3. *D2*: DPSM option 2 using a clock cycle equal to 10% of original critical path length, e.g. if the duration of the project is 200 days, the clock cycle is set at 20 days and the objective is to maximise resource utilisation within 20 day cycles.
4. *D3*: DPSM option 3 using a clock cycle equal to 10% of original critical path length, e.g. if the duration of the project is 200 days, the clock cycle is set at 20 days and the objective is to minimise resources fluctuations within 20 day cycles.
5. *D4*: DPSM option 4 using a clock cycle equal to 10% of original critical path length, e.g. if the duration of the project is 200 days, the clock cycle is set at 20 days and the objective is to combine the 3 first options and find the best results within 20 day cycles.
6. *D5–D8*: same as D1 to D4 but the clock cycle is set at 25% of the original critical path length.
7. *D9–D12*: same as D1 to D4 but the clock cycle is set at 50% of the original critical path length.
8. *D13–D16*: same as D1 to D4 but the clock cycle is set at 75% of the original critical path length.

For each run (priority rule/policy) the comparing factors (measures) namely PCTOP, FIRST, SECOND, UNIQUE, TFIOS, FFFRS, and TFFRS were recorded. With the exception of FIRST, SECOND and UNIQUE the rest of the measures were averaged over the 100 projects for each priority rule/policy, the result of which is summarised below, e.g. the first line shows that using P1 (minimum total float) as the priority rule in allocating resources to the activities of project, on average for the 100 projects the project completion time (PCTOP) is 109.46 days (time units), this priority never produced unique best schedule, 27 times was ranked first and 24 times second. The last line of data marked as DOP (DPSM Optimal results) indicates the average of the measures based on the best results produced by DPSM through a search technique. The results of the experiment is shown in *Table 1*.