Open access • Journal Article • DOI:10.1057/JORS.1972.48

Resource-constrained Project Scheduling — The State of the Art — Source link

Willy Herroelen

Institutions: Katholieke Universiteit Leuven

Published on: 01 Sep 1972 - Journal of the Operational Research Society (Palgrave Macmillan UK)

Topics: Project management and Schedule (project management)

KATHOLIEKE UNIVERSITEIT LEUVEN

Faculteit der Economische en Toegepaste Economische Wetenschappen

RESOURCE-CONSTRAINED PROJECT SCHEDULING - THE STATE OF THE ART [*]

by

Willy S. HERROELEN

Onderzoeksrapport Nr. 7201

DEPARTEMENT TOEGEPASTE ECONOMIE

Centrum voor Bedrijfseconomie

[*] To appear in Operational Research Quarterly

Dekenstraat 2
B - 3000 Leuven (België)

Tel. 016/286.54

RESOURCE-CONSTRAINED PROJECT SCHEDULING - THE STATE OF THE ART

Willy S. Herroelen [*]

ABSTRACT

It is the purpose of this article to review the various solutions that have been proposed for the resource-constrained project scheduling problem. An attempt is made to give the state of the art to date, as well as to point out potential future courses of development. Emphasis is placed on the basic approach involved in each technique rather than on the computational steps required to obtain a solution. In principle the review is limited to solution techniques described in the open literature ;other important work will be treated also, as far as it is known to the author.

---

[*] Willy S. Herroelen is assistant at the Department of Applied Economics, Leuven University. The author wishes to thank Professor F. Van Winckel, Professor R. Vandenborre and Professor J. Van der Eecken for their helpful comments and suggestions on an earlier version of this paper.

INTRODUCTION

The planning and control of large projects is a difficult and utmost important problem for modern enterprise that many network planning techniques have tried to handle. Practical application of these techniques leads however to many difficulties. During the planning phaze of a project, project management must indeed solve a lot of problems concerning technical realization as well as the time, cost and resource aspects involved.

Common network planning techniques, such as PERT (Program Evaluation and Review Technique) and CPM (Critical Path Method), essentially concern themselves with the time aspect only. These methods aim to minimize project duration, assuming that the various resources required for project realization are available. In practice however, project realization requires the use of various resources, whose often limited availability directly influences planning objectives, time estimations, scheduling and progress control. When activities require resources for their execution (e.g. manpower, materials, equipment, capital, etc.) that are only available in a limited amount, bottle-necks may appear, e.g. activities cannot be started on time due to the unavailability of resources, activities requiring the same resource that is only available one unit at the time must be delayed, etc.

The various resource problems that may appear during project scheduling can be divided in three classes : time/cost tradeoff, resource leveling and resource allocation.

Time/cost tradeoff problems may appear when there are no constraints imposed on the availability of resources. The problem then consists of reducing project completion time by adding additional resources to certain activities, so that the execution of these activities may be accelerated. When this is the case, there are many different ways in which activity durations may be selected so that project completion times of the resulting schedules are all equal. However, each schedule may yield a different value of total project direct cost. It would therefore be desirable to have some method for determining the least costly schedule for any given project duration. Several such methods have been developed, each of which hinge upon various assumptions about the form of the activity direct cost-duration relationship. This class of problems has been extensively reviewed elsewhere[1,2] and will not be treated in this article.

The time/cost tradeoff procedures just mentioned implicitly assume that the addition of resources causes no conflicts. It is possible however that one aims, given a total project completion time, to level the various resource requirements over time. This resource leveling problem occurs when sufficient resources are available for the completion of the project, but one tries to keep the resource usage as much as possible to a constant rate. This problem, for which many optimization and heuristic procedures are available[3-7], lies also outside the scope of this article.

When total resource usage is restricted to a given limit, the objective may be to allocate the various resources to the activities in such a way that the project duration is minimized. It is this type of problem that forms the subject of this survey.

In writing the review an effort was made to throw special light on new approaches that were not yet available at the time previous surveys were written.[1,2,8] In what follows, the various contributions will be categorized by the method used. In principle the discussion will be limited to solution techniques described in the open literature; other important work will be treated also, as far as it is known to the author.

Solution techniques for the resource-constrained project scheduling problem have been offered by the application of linear and integer programming, dynamic programming, implicit enumeration, bounded enumeration (branch-and-bound) and heuristic programming.

INTEGER PROGRAMMING

An integer programming approach is offered by Gonguet.[9] The time period for the project is divided into a number (w) of intervals. The decision variables ($X_{L,t}$) assume the values 1 or 0 depending on whether or not a particular activity (L) is scheduled to start during a particular time interval (t). The number of decision variables is therefore the number of possible start times for all activities. Technological constraints ensure that the start time for dependent activities is later than the completion time of prior activities. The number of technological constraints is the number of dependent activities. Resource constraints are based on the need to ensure that the demand for each resource during each time interval does not exceed its availability. Thus, for each interval, the number of resource constraints is the number of resources required.

The objective function is that of minimizing the start time of the last activity in the project (one can be easily created), the number of possible start times being dependent on the number of intervals in the scheduling period.

Though the formulation of the resource-constrained project scheduling problem as an integer programming problem poses no serious difficulties, Gonguet points out that practical application of his model is very limited due to the large number of variables and constraints. For a project made up of 16 jobs requiring a single resource, the formulation as an integer linear programming problem requires, for a scheduling period of 32 time units, already 239 variables and 63 constraints. Gonguet's procedure can thus be denoted as computationally prohibitive.

In fact, until recently, it was generally accepted that integer programming, because of the large number of variables and constraints and due to the fact that no integer linear programming codes had lead to uniformly good results, should be discarded as a computational approach for the resource-constrained project scheduling problem.

It must be noted however that for each type of problem alternative formulations exist. An efficient formulation will always depend upon the clever choice and definition of variables. This constitutes a challenging problem of design.

Pritsker, et. al.[10] have proposed a formulation in which (integer) 0-1 variables denote for certain periods (depending on job arrival time, due dates, precedence constraints, etc.) whether or not an activity is completed in these periods. The authors also refer to a similar "0-1 formulation"[11] in which the variables indicate whether or not a job has been completed prior to some selected periods. The formulation can accomodate a wide range of real-world situations including multiple resource constraints, due dates, job splitting, resource substitutability, and concurrency and nonconcurrency of job performance requirements. The algorithm may use one of three possible objective functions : minimizing the time by which all projects are completed (minimizing makespan) and minimizing total lateness or lateness penalty for all projects. For a sample problem of 8 jobs and requirements of three resource types, the formulation requires 37 contraints and 33 variables. This problem was solved on the RAND's IBM 7044 computer in 2.3 seconds. These are the only results reported by the authors, who conclude that their research coupled with the immense research on zero-one programming codes should yield practical procedures for obtaining optimal solutions to certain types of scheduling problems.

Franke[12] simplifies a model proposed by Riester and Schwinn[13], that requires two types of 0-1 variables. The first type is used to denote whether or not an activity is scheduled on a certain moment in time, while the second type denotes whether or not the project is finished on or prior to a certain moment in time. Franke also presents two alternative formulations which, unlike the Riester and Schwinn formulation, do not require the division of the scheduling period in fixed time intervals. The 0-1 variables hereby denote whether or not an activity must be finished before the start of another activity. The author does not report any computational results but concludes that the major difficulty in using these models, is the lack of a practical algorithm for solving integer programming problems.

Prabhakar[14] has offered a 0-1 integer linear programming formulation that differs from the others just mentioned, in that the entire project planning period is not necessarily considered for resource allocation at one time. Furthermore the problem formulation takes into account some unusual features, peculiar to the construction industry. According to the author, there are certain aspects associated with the problem formulation and solution that make the approach both technically and computationnally feasible. These are (1) breaking down of a large problem into smaller subproblems which are easily solved on the computer; (2) ease with which complex restrictions arising from socio-economic considerations and industry practices can be incorporated into the model ; (3) use of a fast and efficient computer code to solve 0-1 integer linear programming problems and the special structure of the problems which enables one to reach a solution fairly close to the optimum within a few seconds on a large scale digital computer.

Activities and resource requirements must satisfy the following criteria : (1) a particular activity can call for resources which vary in amount from day to day ; (2) amounts of resources utilized should not exceed the available capacities ; (3) use of resources should be continuous so that the overall resource profile should look like a bell shaped curve ; (4) a weekly pattern of resource utilization should be built with possible increasing demands for a resource to happen early in the week ; (5) quantities of certain resources should be in multiples of two ; (6) an activity may sometimes need to be broken up into two or more parts (each part of an activity can however be considered as an activity in itself so that the requirement of continuity of any single activity is always met) ; (7) certain activities in parallel streams can occur together and certain others cannot, that is to say certain activities can be started only after completion of some other activities; (8) duration of an activity depends on the amount of reseources used : a shorter duration can be obtained by increasing the resources and vice versa.

The solution process runs as follows. First, the maximum time period over which any meaningful resource allocation can be made is determined. This period could cover the entire project life or part of it depending on whether the project is of short duration or extending over several months. This period is then subdivided into smaller planning cycles of one or two weeks. The unit of time considered is one day. The allocation problem is solved iteratively by making a series of passes over the time period under consideration. Each of these passes consists in solving a set of subproblems generated over the given time period. During the first pass, there could be as many subproblems as the number of intervals, but during subsequent passes it is seldom necessary to solve all the subproblems. At the beginning of each new pass the algorithm internally examines the resource profile charts generated at the end of the previous pass for any uneven of undesired pattern. New subproblems are then set up only at those places where it is required to smooth out a resource profile or to achieve a desired pattern. There are some definite criteria to decide where and when to generate a new subproblem during each pass.

The zero-one integer programming algorithm used was a Union Carbide Corporation program developed by Tang[15] and based on the original Balas' enumerative technique.[16] Some of the major modifications are (1) the addition of a backtrack procedure which greatly reduces the storage capacity needed, (2) inclusion of cost considerations in Balas' first test which improves the efficiency, and (3) ranking of the variables also to improve efficiency. The algorithm was programmed in FORTRAN IV (Level H) for the IBM 360/65 multiprogramming system. Prabhakar reports that several test problems ranging from 40 variables and 60 restrictions to 200 variables and 12 restrictions were solved in a few seconds.

DYNAMIC PROGRAMMING

No practical dynamic programming approach for the resource-constrained project scheduling problem is known to the author. Carruthers and Battersby[8] give a dynamic programming formulation for the disjunctive problem. This is a special kind of the resource-constrained project scheduling problem in which two or more activities may compete for the same resource that is available only in one unit. The authors conclude that the amount of information to be stored if one is to use the dynamic programming method will be considerable but dimensionality could be reduced perhaps by the use of Lagrange multipliers.

The only dynamic programming approach for the resource leveling problem mentioned above, that has been carried out to some depth is reported by Petrović[6,7]. The method uses in its functional equation the sum of the squared deviations from mean resource requirements. For that reason, the extension of the method to the resource-constrained project scheduling problem seems not trivial. Petrović reports no computational experience. His papers however are very interesting from a conceptual point of view.

BALAS' IMPLICIT ENUMERATION APPROACH

Balas[17] defines the common critical path problem in terms of a search for the start times (t_i) of all activities (i = 1, 2, ..., n) such that the start time of the last activity (t_n) is as early as possible, subject to all procedural constraints. He adds resource constraints to ensure that the demand on each resource at each point in time is not greater than the amount available.

Balas proves this problem to be analogous to a machine sequencing problem defined as the search for an optimal sequence of the performance of m items (or lots of items) on q machines, where each item must be performed on a given sequence of machines, i.e. (1) a given operation on a given item must be performed on a given machine (or a set of identical machines) ; (2) the operations to be performed on a given item must be ordered by a set of sequence relationships of the type $t_j - t_i \geqslant d_{ij}$; (3) there exists freedom of choice as to the sequence of operations that must be performed on a given machine ; (4) a sequence is sought that minimizes t_n , the total time required for the performance of all items.

Balas shows that these two equivalent problems can be reduced to the problem of finding an optimal selection of arcs in a disjunctive graph with stability conditions. For this reduced problem, the author has developed elsewhere an implicit enumeration algorithm.[18] This algorithm solves the problem by generation of a sequence of PERT-networks satisfying certain stability conditions. Computational experience with the method is not given by the author.

BOUNDED ENUMERATION TECHNIQUES

The application of the bounded enumeration approach to resource-constrained project scheduling problems, was introduced by Mueller-Merbach in 1967.[19-21] The algorithm can accomodate multiple resources per project, but only one resource type per job is allowed in requirements and availabilities of only one unit. The method starts from a feasible solution obtained by heuristic techniques. When this solution can be recognized as optimal, the algorithm stops. The optimality of a solution is guaranteed when project duration equals either the length of the critical path obtained without consideration of resource constraints, or the sum of the durations of the activities requiring a resource and the unused time created before the first and after the last activity requiring the same resource. When it is impossible to recognize the solution as optimal, the enumeration process is started and among the resource requiring activities with computed start dates, the first one is chosen. If it can be shown that the particular activity sequence built up so far cannot lead to a better solution than the one already obtained, further branching along this route is dropped and the algorithm traces back to the last scheduled resource requiring activity. In case a solution improvement is obtained, the optimality test is applied and the enumeration process continues. The algorithm was programmed in FORTRAN for the IBM 7040. The author concludes that computation time is very high (for a 30 node-50 activities problem, with 20 activities and a single resource, computation time was about 35 minutes). Furthermore the computation time cannot be estimated in advance.

In August 1967, Johnson[22,23] presented another such scheme called BETINA (Bounded Enumeration Technique in Network Analysis). As was the case for the Mueller-Merbach paper, this one is also limited to the simplest case of the constrained-resource problem involving only one resource type per job, no job interruptions, and constant resource demand

and availability for every time period of project duration. The fundamental idea of both algorithms lies in the efficient exploration of the decision tree of the problem, terminating the search along a particular route when a minimum bound for the solution currently being examined exceeds a feasible solution already obtained.

Johnson's method calls for the following framework. First, a partial schedule (PS_k) is defined which represents the state of the decision process at a particular time (t_k) when a decision must be made, i.e. partial schedule PS_k divides the set of tasks into four mutually exclusive and collectively exhaustive subsets. Those tasks that have been scheduled and completed are termed the complete set (C_k). Tasks still in progress at time t_k make up an active set (A_k). The set of tasks not yet scheduled, but with all of their logical predecessors complete, is called the decision set (D_k) since only tasks from this set can be scheduled at time t_k. The remaining unscheduled tasks are termed the remaining set (R_k). Defining z_o to be the duration of an existing feasible solution and w_k to be a minimum bound on all complete schedules containing partial schedule PS_k, the process can be summarized as follows.

Given partial schedule PS_k, heuristically choose an attractive feasible subset F_k^* (none, one or more activities) out of the decision set (branching), construct partial schedule PS_{k+1}, and test whether the minimum bound, w_{k+1}, is greater than or equal to the existing best solution, z_o (bounding). If $w_{k+1} \geq z_o$, further search along this route cannot lead to an optimal solution and therefore return to consider other feasible subsets F_k^s. On the other hand, if $w_{k+1} < z_o$, develop all F_{k+1}^s and continue. Of course, if partial schedule PS_{k+1} is a complete schedule, then $z_o = t_{k+1}$, and the process returns to the last unbounded partial schedule. The algorithm terminates when there are no more unbounded, unexplored feasible subsets. Besides the use of minimum bounds, the algorithm also relies on some partial solution dominance tests which, according to Johnson decrease significantly the computation time required to find an optimal solution.

The BETINA-program was written in FORTRAN IV and run on the IBM 360/65 at the M.I.T. Computation Center. Johnson states that computation times tend to rise rapidly with the size of the problem. He concludes that computational difficulty is, for all practical purposes, an unpredictable function of the detailed problem structure and hence, that his technique is an unreliable optimization procedure for most real project scheduling problems.

Davis[24,25] developed the MARK I program which permits the determination of optimal (minimum duration) solutions for more general cases. Problems involving several resource types per job are solved as quickly as single-resource problems. In addition, job interruptions may be allowed, and the job requirements of each resource type may vary over the duration of each job, subject to integer restrictions. Davis uses a formulation of the resource-constrained project scheduling problem similar to the formulation of the assembly line balancing problem given by Gutjahr and Nemhauser.[26] Following this approach the original problem (where jobs are broken up in tasks with unit duration) is reduced to one of finding the shortest path between the start and end nodes of a finite directed graph. Subject graph must be created, taking into account certain rules, by means of the generation of feasible subsets, i.e. sets, which, if they contain a certain task, also contain all predecessors of this task. In this new network, the A-network, nodes are represented by subsets of the total set of tasks (defined in a special way) while the length of the arcs denote one schedule day. The author then proves that, because of the correspondance between the arcs of the network and scheduling days in the original problem, the path that leads to the final node and contains the smallest number of arcs, represents the minimum-duration schedule. The large number of possible arc connections that can be generated connecting the feasible subsets is reduced by reducing the number of feasible subsets through an ingenious application of bounding techniques.

In this way the MARK I program was able to find the optimal solution (starting from a heuristic starting solution) for 48 out of 65 artificially created networks each consisting of about 20 activities and requirements of three resource types. For the remaining 17 problems no optimal solution could be obtained because some program constraints were violated. In those cases however an approximate solution was found which was at least one day shorter than the critical solution. The mean computation time on the IBM 7094 amounts to 56 seconds, but the variance is considerable. Because the available storage capacity was sometimes exceeded before a final solution could be obtained, Davis remarks that computer memory storage and not computation time was the primary operating constraint on the IBM 7094. He also reports that work has begun on a MARK II program version of the algorithm for the IBM 360 computer.[9,25]

A bounded enumeration approach to the resource-constrained project scheduling problem also is offered by Schrage.[27] He considers both the nonpreemptive and the preemptive case. The nonpreemptive case deals with nonpreemptive or no-lotsplitting constraints which state that once a required resource r(i) has been assigned to an activity i, r(i) must remain assigned to activity i for a length of time t(i) representing the activity duration. When preemption is allowed, it is sufficient that the resource r(i) has been assigned to activity i for intervals of time totaling t(i).

Schrage gives an efficient enumeration procedure for generating all active schedules. A schedule is denoted "not active" if there exists some activity that can be started earlier without changing the start times of any other activities and without violating the precedence, resource and preemption or nonpreemption constraints. Based on this enumerative scheme he develops a branch-and-bound method for implicitly enumerating all schedules and determining the optimum. The algorithm was programmed for the IBM 7094 computer and tested on a series of problems (also job shop and flow shop problems). Computational times seem to be acceptable.

Tabourier[28] developed a particular branch-and-bound method (TADSEP) for the disjunctive problem (already defined above). The method was developed in the context of a synchronization problem of traffic lights, where each of the tasks (T_i) to be scheduled refers to the fact that light i must be green during a period (d_i) and the disjunctive constraints result from the fact that certain pairs of lights are not allowed to be green simultaneously. Computational experience with the experimental algorithm (programmed in FORTRAN for the CDC 6600 computer) is only given for a problem consisting of 10 tasks and 20 disjunctive constraints. A total of 241 optimal solutions was found in 7 seconds.

HEURISTIC PROGRAMMING

Many heuristic programming approaches have been proposed for the resource-constrained project scheduling problem. The approaches that have been extensively reviewed elsewhere[1,2], will only be briefly discussed for completeness sake.

One of the first heuristic methods was reported by Kelley.[29] The author suggests ranking the activities by technological sequence and within that order, by some intuitive measure of the task's importance, such as "slack" computed by a critical path procedure. He then suggests serial and parallel methods for finding the minimum-length schedule subject to stated resource constraints. Recognizing that different rankings gave different project durations, the author recommended repeating the process with various rankings.

Moder and Phillips[30,31] suggest using the "latest start date" from CPM calculations as a ranking device. This seems to be a more dynamic measure of a task's importance, since it is equivalent to ranking by remaining slack.

Verhines[32] on the other hand, proposes that if two activities compete for the same resource in such a way that only one of these activities can be started, the one with the longest remaining series of activities should be given priority.

A very promising heuristic approach is proposed by Fehler.[33] The method first schedules the eligible activities that require no scarce resources. An activity is denoted as eligible when all its predecessors are already scheduled so that its earliest start time can be computed. Next, the eligible resource requiring activities are considered. When more than one eligible resource requiring activity is available, a sequence of two such activities is built. For this sequence the soonest possible project finishing date is computed. Parallel activities are not considered. The sequence is then reversed and again the soonest possible project finishing date is computed. The first activity of the sequence leading to the highest project duration is then dropped. With the "winning" activity and the following eligible resource requiring activity two new sequences are built, etc. until one single activity remains. This activity is then scheduled. The method then schedules the new eligible activities requiring no resources and considers the new eligible resource requiring activities for scheduling. This process continues until all project activities are scheduled. Sufficient computational experience with this variation-enumeration method is not available so far.

Riester and Schwinn[13] propose parallel procedures for various types of resource problems, where the activities eligible to start on each time moment are ranked according to a set of three priority rules. The allocation procedure may be connected to a guided Monte-Carlo simulation which enables the probability intervals, required for the simulation, to be altered after each computerrun, so that an idea about the efficiency of the priority ranking is obtained and continuously better solutions may be found up to a certain level. Room is also left for a right-shift procedure as proposed by Wiest.[34-36] The procedures are illustrated on a practical problem in the coal mining sector.

Pascoe[37] has analyzed the results obtained with many heuristic procedures, concluding that they heavily depend on the logic of the project itself, the procedure of allocation and the sorting routine employed. Similar conclusions have been obtained by Bosman and Oosterhoff[38] and Herroelen.[39]

Mueller-Merbach[21] has made an elaborate test of ten heuristic sorting routines comparing them with his own bounded enumeration approach. The best results were obtained with a version of the smallest soonest start time routine.

All the approaches mentioned so far, are directed towards the problem of resource allocation in a single project. Attention however, has also been given to the multiproject problem.

Mc Gee and Markarian[40] offer a method suitable for dealing with the multiproject problem. Their model heavily relies on a time/cost tradeoff formulation of the CPM type mentioned earlier. The authors offer many flow charts of the algorithm but no computer program. Computational results are not given.

Wiest[34-36] has developed the SPAR-1 program (Scheduling Program for Allocation of Resources), that is able to handle single or multiple projects, fixed or variable crew sizes and constant or variable shop limits over the scheduling period. The basic program is based on three heuristics. The first one allocates resources, period by period, to jobs listed in order of their early start times. The second heuristic is used in choosing amongst alternative candidates : when several jobs compete for the same resources, preference is given to the jobs with the least total slack. The last of the three heuristics allows for the rescheduling of noncritical jobs, whenever possible, in order to free resources for scheduling critical jobs where no slack time is available. The basic program just described is modified by a number of additional scheduling heuristics or subroutines generally designed to increase the use of available resources and/or to decrease the length of the schedule. A package of search rroutines is also included in the program. SPAR-1 was programmed in FORTRAN both for the IBM 7094 and the Control Data G-20, and may be dimensioned to handle a project with 1200 single-resource jobs, 500 nodes and 12 shops over a time span of 300 days. Multiple projects may be scheduled within the same total job constraint. Computation times seem reasonable (about 5 to 10 minutes).

One of the most elaborate methods, capable of handling multiple projects, is the RAMPS-technique.[41,42] Despite the fact that the computational steps involved in RAMPS (Resource Allocation and Multi-Project Scheduling) are nowhere fully described in detail, it is known[43] that the method works by dividing the overall problem in various sub-problems which correspond to time periods (days, months, etc.). For each period the program evaluates possible scheduling combinations by delaying or interrupting tasks. From these combinations the method only accepts those compatible with the resource constraints. The selection of combinations is subsequently judged in function of (1) a priority combination of objectives (such as start and finish each task as soon as possible, give priority to critical activities, maximize tasks simultaneously in process, etc.) ; (2) the corresponding cost ; (3) the indicated total delay for each project ; and (4) the relative importance of the various projects under consideration. In order to accomplish all this, three sets of input data are required for each activity namely the amount of resource required, time required and cost of splitting an activity once it has begun. In addition the program requires project information such as start date, desired completion date and dollar-penalty rate for delay of completion or alternatively a project priority rating. The system will accept up to 700 activities and 60 different resources. The output consists of a work schedule for each project including costs and resources and a summary of resources used each time-period, classified by resource type. In this sense RAMPS seems to be the most universally applicable heuristic computer program for resource allocation that is presently available.

Combe[9] describes a program called ASTRA-DISK using an activity sorting based on project priority, latest start time, earliest start time and duration. Frère, Peperstraete and Roba[9] present a multi-project planning program that may take 300 different resources and 5000 tasks into account where each task may be allocated a maximum of eight resources. The program uses priority heuristics for the allocation process such as criticality, etc. The program is written in FORTRAN IV and comprises 5000 instructions. At present it is available on an IBM 360/50 with 256 000 central storage bytes. Performance times are very variable according to the number of constraining resources and the number of listings required. For a 1500 task scheduling problem with 100 constraining resources, computation time amounts to 30 minutes.

Oshima[9] discusses the NHK-SMART program that allocates so-called key resources first and then attempts to allocate the other ones without changing key resource assignments. In handling the resources, consideration is given to the schedule time and the slack time of activities. Computational experience is not given.

Fendley[44] has developed a complete multi-project scheduling system making use of the minimum-slack-first priority rule for sequencing individual jobs such that total costs at least approach a minimum. The method sets realistic due dates by analyzing the resource load on the facility to determine the amount of slippage that must occur to perform all projects with fixed resources.

Finally, it should be added that several computer manufacturers offer commercial program packages based on a heuristic approach[45,46] (comparisons of some of these programs are available[47,48]).

CONCLUSIONS

This summary should make evident that the conclusion of Laue[2], who states that "it is notable that no practical analytical methods are offered for the solutions of problems with constraints on resources", must somewhat be modified. This article indeed reveals that besides various heuristic procedures, also a wide variety of analytical optimizing techniques are available.

In regard to the heuristic approaches it can be said that some of the techniques will obviously be more powerful than others. However, the relative advantages of the various heuristic methods are difficult to assess because, at the moment, no exhaustive objective comparison of efficiency seems available in the open literature.

Recent interest is focused again on optimization methods, possibly due to the development of larger and faster computers and new advances in mathematical programming. Among these methods bounded enumeration and integer programming seem most promising. Although for large projects these methods must encounter serious difficulties due to cumbersome formulation and excessive memory requirements and computation times, optimism remains justified if one realizes that such approaches heavily rely on problem design. For a given type of scheduling environment, alternative formulations do often exist. As it was emphasized before, an efficient formulation depends upon a judicious choice of definition for the variables. It thus seems possible that alternative integer programming formulations may lead to better results. As for the branch-and-bound approach it seems reasonable to accept that alternative formulations and construction of stronger bound values as well as powerful partial solution dominance criteria, may approve upon the methods now developed.

It is quite surprising that, up till now, the dynamic programming approach has been given so little attention. Although the computational difficulties inherent to this approach are apparent, efforts together with computational experience in this area are strongly welcome.

As an overall conclusion one can therefore say that future courses of development in resource-constrained project scheduling problems should be directed towards the search for alternative formulations for the integer programming approach ; an effort to devise more powerful bounds and dominance criteria for use in a branch-and-bound technique, and a serious effort to involve dynamic programming in this problem. Furthermore some exhaustive evaluation and objective ranking of the available heuristic solutions would be a useful contribution.

REFERENCES

[1]E. W. Davis (1966) Resource allocation in project network models - A survey. J. ind. Engng 17, 177.

[2]H. J. Laue (1968) Efficient methods for the allocation of resources in project networks. Unternehmensforschung 12, 133.

[3]A. R. Burgess and J. B. Killebrew (1962) Variation in activity level on a cyclic arrow diagram.
J. ind. Engng 13, 76.

[4]L. Dewitte (1964) Manpower leveling in PERT networks.
Data Proc. Sci. Engng 29.

[5]F. K. Levy, G.L. Thompson and J.D. Wiest (1962) Multi-ship, multi-shop, workload smoothing program.
Nav. Res. Logist. Q. 9, 37.

[6]R. Petrović (1968) Optimization of resource allocation in project planning. Ops. Res. 16, 559.

[7]R. Petrović (1969) On optimization of resource leveling in project plans. Chapter in Project Planning by Network Analysis : Proceedings of the Second International Congress of Internet (Lombaers (ed.)). North-Holland Publishing Company, Amsterdam, 268.

[8]J. A. Carruthers and A. Battersby (1966) Advances in critical path methods. Opl. Res. Q. 17, 359.

[9]Chapter in Project Planning by Network Analysis : Proceedings of the Second International Congress of Internet (Lombaers (ed.)). North-Holland Publishing Company, Amsterdam.

[10] A. A. B. Pritsker, L. J. Watters and P. M. Wolfe (1969) Multiproject scheduling with limited resources : a zero-one programming approach. Mgmt Sci. 16, 93.

[11] A. A. B. Pritsker and L. J. Watters (1968) A zero-one programming approach to scheduling with limited resources. RAND Corporation. Paper RM-5561-PR.

[12] G. Franke (1971) Zur Ablaufplanung in Netzwerken (Scheduling in Networks). Zeitschrift für Betriebswirtschaftliche Forschung 23, 106.

[13] W. F. Riester and R. Schwinn (1970) Projektplanungsmodelle (Project Planning Models). Physica-Verlag, Würzburg-Wien.

[14] T. Prabhakar (1970) Resource allocation/leveling in engineering/ construction projects. Paper presented at the 37th Meeting of ORSA, April 20-22, 1970.

[15] Y. P. Tang (1968) A study on the selection of project portfolio and the related zero-one integer linear programs. R & D Report. Union Carbide Corporation, South Charleston, W. Va., March 11, 1968.

[16] E. Balas (1965) An additive algorithm for solving linear programs with zero-one variables. Ops Res. 13 , 517.

[17] E. Balas (1971) Project Scheduling with resource constraints. Chapter in Applications of Mathematical Programming (Beale (ed.)). The English Universities Press, London.

[18] E. Balas (1968) Machine sequencing via disjunctive graphs : an implicit enumeration algorithm. Management Science Research Report N° 125, Carnegie-Mellon University, Pittsburgh.

[19] H. Mueller-Merbach (1967) Erfahrungen mit Methoden der Betriebsmitteleinsatzplanung in CPM-Netzplänen (Experience with resource allocation methods in CPM-networks). Proceedings of Internet I, Dunod, Paris, 461.

[20] H. Mueller-Merbach (1967) Ein Verfahren zur Planung des optimalen Betriebsmitteleinsatzes bei der Terminierung von Grossprojekten (An optimal method for resource allocation in large projects). Zeitschrift für wirtschaftliche Fertigung 62, 83, 135.

[21] H. Mueller-Merbach (1970) Optimale Reihenfolgen (Optimal Sequences). Springer-Verlag, Berlin.

[22] T. J. R. Johnson (1967) A branch and bound algorithm for the resource constrained project scheduling problem. M. I. T., August 1967.

[23] T. J. R. Johnson (1967) An algorithm for the resource constrained project scheduling problem. Ph. D. Thesis, Sloan School of Management, M.I.T.

[24] E. W. Davis (1968) An exact algorithm for a multiple constrained-resource project scheduling problem. Yale University, Department of Administrative Science.

[25] E. W. Davis and G.E. Heidorn (1971) An algorithm for optimal project scheduling under multiple resource constraints. Mgmt Sci. 17, B-803.

[26] A. L. Gutjahr and G. L. Nemhauser (1964) An algorithm for the line balancing problem. Mgmt Sci. 11, 308.

[27] L. Schrage (1970) Solving resource-constrained network problems by implicit enumeration - Nonpreemptive case. Ops. Res. 18, 263.

[28] Y. Tabourier (1969) Problèmes d'ordonnancement à constraintes purement disjonctives (Scheduling with disjunctive constraints). Rev. fr. Inf. Rech. opér. 3, 51.

[29] J. E. Kelley (1963) The critical path method : resources planning and scheduling. Chapter 21 of Industrial Scheduling (Muth & Thompson (eds.)). Prentice-Hall Inc., Englewood Cliffs.

[30] J. J. Moder and C. R. Phillips (1964) Project Management with CPM and PERT. Reinhold Publishing Corporation, Chapman & Hall, Ltd., London.

[31] J. J. Moder and C.R. Phillips (1970) Project Management with CPM and PERT. Second Edition. Van Nostrand Reinhold Company, New York.

[32] D. R. Verhines (1963) Optimum scheduling with limited resources. Chem. Engng. Progr. 59, 65.

[33] D. Fehler (1969) Die Variationen-Enumeration. Ein Näherungsverfahren zur Planung des optimalen Betriebsmitteleinsatzes bei der Terminierung von Projekten (Variation-enumeration. A heuristic method for resource allocation in project scheduling). Elektronische Datenverarbeitung 11, 479.

[34] J. D. Wiest (1963) The scheduling of large projects with limited resources. Ph. D. Dissertation. Carnegie Institute of Technology.

[35] J. D. Wiest (1967) A heuristic model for scheduling large projects with limited resources. Mgmt Sci. 13, B359.

[36] J. D. Wiest (1964) Some properties of schedules for large projects with limited resources. Ops Res. 12, 395.

[37] T. L. Pascoe (1966) Allocation of resources CPM. Rev. fr. Rech. opér. 38, 31.

[38] A. Bosman and W. Oosterhoff (1968) Allocation of production-factors in project-planning. Paper presented at the European Meeting IMS, TIMS, ES, IASPS, Amsterdam, September 2-7, 1868.

39 W. S. Herroelen (1972) Methodologische benadering van de heuristische programmering en praktische toepassing op een projektplanningsprobleem (Methodological approach of heuristic programming and practical application on a resource allocation problem). Ph. D. Thesis, Leuven University. In preparation.

40 A. A. Mc Gee and M. D. Markarian (1961) Optimum allocation of research/engineering manpower within a multi-project organizational structure. Paper IBM 61-907-171. IBM Federal Systems Division, Oswego, New York.

41 S. Lambourn (1963) RAMPS - A new tool in planning and control. Comp. J. 5, 300.

42 J. Moshman, J. Johnson and M. Larsen (1963) RAMPS - A technique for resource allocation and multi-project scheduling. AFIPS Conference Proceedings 23, 17.

43 J. F. Boss (1966) Prise en considération des contraintes pesant sur la disponibilité des moyens dans les méthodes de chemin critique (Considering contraints on resource availability in critical path methods). Rev. fr. Rech. opér. 38, 3.

44 L. G. Fendley (1968) Toward the development of a complete multiproject scheduling system. J. ind. Engng 19, 505.

45 IBM World Trade Corporation (1966) IBM Systems/360, Project Management System (360A-CP-04X). Application Description.

46 IBM World Trade Corporation (1969) IBM System/360, Resource Allocation (Real/360) for Project Control System/360 (360-CP-08X). Application Description Manual.

47 Computer Programme Users Report N° 5 (1968) A comparison of commercially available programmes for network analysis. Department of Civil Engineering, Loughborough University of Technology.

[48] M. H. Chambers (1968) Survey of CPA programs. Appendix 3 in _Critical Path Analysis in Practice_ (Thornley (ed.)), Tavistock Publication, London.