A PROGRAMMED ALGORITHM FOR RESOURCE SCHEDULING IN CONSTRUCTION

طريقة مبرمجة لاعداد البرامج الزمنية لمشروعات التشييد في حالة عدم توافر عناصر العمل

By Adel I. Eldosouky

Assistant Professor, Faculty of Engineering, Mansoura University

الملخص :

يعتبر هذا البحث خطوة على طريق استعمال الطرق الدقيقة لاعداد البرامج الزمنية لمشروعات التشييد في حالة عدم توافر عناصر العمل حيث ان برامج الحاسب الالى المتوافرة حاليا تستخدم طرق تقريبية لاعداد هذه البرامج الزمنية . وقد تم تطوير طريقة " دراسة الحلول الممكنة كل على حدة بالطريقة الضمنية " ، والتى أعدت خصيصا لحل هذه المسألة ، وذلك لتناسب متطلبات مشروعات التشييد من حيث امكانية تداخل توقيتات نشاطات المشروع ، وضرورة بدء أو انتهاء بعض الانشطة في مواعيد محددة . وقد تم اعداد برنامج للحاسب الالى لاستخدام الطريقة المطورة حيث تم تقليل وقت التشغيل اللازم للحصول على البرنامج الزمنى المثالى بتوصيف حد أدنى لفترة المشروع ، وبترتيب نشاطات المشروع طبقا لوقت بدءها المتأخر المحدد بطريقة المسار الحرج ،وكذلك بترتيب عناصر العمل طبقا لنسبة مجموع استخدام الانشطة لها الى متوسط توافر هذه العناصر . كما قدم البحث تطبيقا على مشروع واقعي مكون من ٨٠ نشاط ويستخدم ١٠ عناصر عمل مهمة وذو قيود على توقيتات بعض أنشطته ، حيث ثبت انه باستخدام أجهزة الحاسب الالى الحديثة ذات السرعات العالية تكون الطريقة المطورة مناسبة للاستخدام مع مشروعات التشييد .

ABSTRACT

If the critical path determined completion time of the construction project is constrained by the lack of resources, which are needed to carry out some of the activities, then the analysis required for scheduling the activities is much more involved. There are two major categories of procedures for solving this problem. The first involves the use of some heuristic rules in determining priorities among activities competing for available resources. The second involves some form of mathematical programming to produce the optimal schedule. Heuristic scheduling rules, computer programmed to give good feasible schedules have been the basis for all practical systems to date. In the same time, the optimal solution of the problem has not been paid enough attention in the construction management literature. The increasing availability of more powerful computer systems should see more emphasis on the development of procedures to give the optimal solution for large complex projects which occur in construction practice. In this paper a computer program is developed for implementing an implicit enumeration algorithm to give the optimal solution of the problem for construction projects. The algorithm has been chosen to satisfy large projects and its efficiency has been established by specifying a lower bound on the length of the schedule and by sorting project activities and resources in a way such that the computation time is greatly reduced. Experience in applying the algorithm to an actual project is reported.

1. INTRODUCTION

When determining the project completion time using CPM, one implicitly assumes that the project is only constrained by the logical interrelationships depicted by the network. If the resources requirements for the various activities cannot be secured whenever required, then the activities may exceed their total float and this will delay the entire project. The construction manager's problem, therefore, becomes one of allocating available resources to the activities in the manner that will result in the least delay of the project completion.

This problem of resource constrained project scheduling is solved by two distinctly different approaches. The first category includes heuristic approaches which are usually used for the purpose of finding a feasible solution. Heuristic models are based on a process of decision making according to a set of priority rules that are set based on activity characteristics. Different models can be formulated by using different combinations of priority rules. Studies have indicated that the use of least float as the first priority rule usually leads to a good solution. Most available software packages (such as Project Cost Model and Primavera Project Planner), which are man-machine interactive procedures, rely on heuristic rules for the development of their resource scheduling systems.

The second category consists of procedures designed to produce the optimal solution. Early attempts of this class formulated the problem as an integer programming problem. The various formulations differ in the number of variables and constraints needed to define the mathematical programme for a given project. Because of the inability of computer codes to efficiently store and solve large networks, early attempts were unsuccessful. Specialized algorithms for solving this problem were then developed.

There are three superior specialized optimization approaches for solving the resource scheduling problem. These are the bounded enumeration of activity completion times given by Davis and Heidorn [1], the branch and bound solution approach introduced by Stinson et al [2], and the implicit enumeration of activity completion times developed by Talbot and Patterson [3]. A study of these exact approaches was made by Patterson [4] who declared that the branch and bound procedure produces solution in the minimum amount of computation time whereas the implicit enumeration procedure requires far less computer storage than do the other approaches. However, computer programs for implementing these techniques were not provided. On the other hand, the techniques were not tested with large construction networks.

In this paper, the implicit enumeration procedure is adopted as the efficient use of computer storage permits solution of large construction projects. A computer program for implementing this algorithm is developed. Both the imposed start and finish dates and the overlaps of the activities are taken into consideration. It is assumed that an activity cannot be interrupted once begun and resources are demanded by an activity in constant amounts throughout the duration of the activity. A significant decrease in the amount of computation time required to solve the problem has been achieved by specifying a lower bound on the length of the schedule, by sorting project activities according to their CPM late start timing and by sorting project resources according to their usage relative to their availability. The programmed algorithm is then used to schedule the activities of a real-life construction project and the computation results are given.

2. DESCRIPTION OF THE IMPLICIT ENUMERATION ALGORITHM

In this section the implicit enumeration algorithm for resource scheduling is described. The algorithm is divided into two parts. The first part deals with the setup for the computation and the second part deals with the actual procedure.

Setup

Activities in the project must be numbered (hence, considered for scheduling) such that if activity i is a predecessor to activity j, then $i < j$. The project horizon PH is set equal to the summation of the duration of all activities. Project completion time T_n (where n is the last activity in the project) is initially set equal to PH. The late finish times LF of the activities are determined by a CPM analysis. The upper bounds on the activities finish times U are initially set as (1) and they will be updated as improved solutions are obtained.

$$U_j = LF_j + (PH - LF_n), \qquad j=1,2,\dots,n \tag{1}$$

Procedure

The algorithm begins with an attempt to find the earliest feasible assignments for activities 1,2,3 and so on, in order. When activity n has been assigned a finish time SF_n a shorter duration schedule has been found. It is $R = T_n - (SF_n - 1)$ periods shorter than the existing best solution T_n. The improved solution is stored and the upper bounds U are reduced by R units. Then the assignment procedure begins again with activity 1.

If an activity cannot be assigned a resource and precedence feasible completion time below the calculated upper bound for the activity, then the procedure backtracks to the next lower activity number. If the lower numbered activity can be assigned a finish time in a period which would not lengthen the duration of the existing schedule, an attempt is again made to determine a feasible completion period for the original activity for which a resource feasible finish period could not be determined. If a more attractive completion time for the next lower numbered activity cannot be found, then backtracking proceeds to the second lower numbered activity in the network; etc. Optimality is assured within this algorithm when the attempt is made to backtrack past the first activity in the network.

3. REDUCING THE ALGORITHM COMPUTATIONAL TIME

The general approach to the problem is to start with an heuristic solution and to improve upon it systematically until the optimum is found. If the initial heuristic solution greatly exceeds the optimal solution (due to PH being set equal to the sum of the activity duration), then a fair amount of time may be expended in generating improved but nonoptimal solutions. On the other hand, the procedure consists of a systematic evaluation of all possible activity finish times for each activity in the project in order to verify optimality. This step of the algorithm also consumes a considerable amount of computational time. However, the following improvements are introduced in order to reduce the algorithm computational time: (1) specifying a lower bound on the length of the schedule (2) sorting project activities according to their late start timing LS and (3) sorting project resources according to their requirements and availabilities. Details of these accelerators are given next.

Specifying a Lower Bound on the Length of the Schedule

The optimum solution of the resource scheduling problem is achieved when an attempt is made to backtrack below activity 1, i.e. when all possible activity assignments have been evaluated. However, another criterion to achieve optimality is to find a solution which equals a proven lower bound on the length of the schedule. When an improved solution with a reduced project completion time SF_n is found, it replaces the existing solution. If SF_n equals a lower bound on the solution, then SF contains the optimal schedule. Using this lower bound criterion will reduce computational time in many cases.

The lower bound on the length of a resource constrained schedule, as given by Willis [5], is the maximum of the time-based or the resource-based bounds. The former is the critical path length of a project; CP, which is equal to the technological earliest completion time. The latter is given by the maximum resource total usage for a particular resource k ($\sum_{j=1}^{n} r_{jk}\, d_j$) divided by quantity of that resource available on a per period basis; RAQ_k . The symbols d_j and r_{jk} stand for duration of activity j and usage of resource k by activity j respectively. Of course, the latter estimate (which must be carried out for all key resources K) gives the equivalent resource time periods required to complete a project. The lower bound is, therefore, the smallest integer which is greater than or equal to the value of LB given by Eq. 2. Obviously, a schedule of length less than this lower bound is not feasible.

$$LB = \max \{ CP, \max_k (\frac{1}{RAQ_k} \sum_{j=1}^{n} r_{jk}\, d_j) \} \tag{2}$$

Sorting Project Activities

The selection of the activity numbering rule is quite important. It determines the order in which activities are considered for scheduling, which in turn significantly affects the efficiency of the algorithm. The chosen rule, however, should give a better heuristic solution to the problem than do any other rule.

Talbot [6] studied 100 ten-activity problems to evaluate eight different heuristic rules used as activity-numbering rules. In order to reduce computational time, it is recommended to use the rule which results in maximum percentage of problems for which the heuristic solution equals the optimum solution and on which the rule is fastest to the optimum. These conditions can be satisfied by using the minimum LS rule.

Accordingly, the minimum LS priority rule will be used by the present research to specify the order in which the activities will be considered for scheduling. Ties will be broken by lowest activity number.

Sorting Project Resources

The objective of sorting project resources is to identify resource infeasibility as early as possible. This can evidently reduce computational time when the number of resources exceeds two as is the case with construction projects.

Talbot [6] suggested that project resources are sorted such that the resources having the maximum frequency of highest per-period requirement relative to average resource availability has the smallest numerical label. Specifically, an index (3) is calculated for each resource:

$$\text{Index}_k = d_j \quad \text{for the k which maximizes } \phi$$
$$\text{Otherwise, Index}_k = 0. \qquad \ldots\ldots\ldots(3)$$

where ϕ is the ratio of r_{jk} to average RAQ_k calculated over the activity critical path determined total float period. These indices are summed over all activities, then the resources are sorted in decreasing order of summed indices.

4. MODIFYING THE ALGORITHM TO SATISFY ACTIVITIES IMPOSED DATES

The implicit enumeration algorithm, as developed by Talbot and Patterson [3], assumes no imposed dates on the timings of the project activities. In many construction projects, certain activities must be started and/or completed by specified dates, for example, construction of an earth dam embankment which must be performed in a dry season. The algorithm must, therefore, be modified to suit such cases.

In case of an activity i imposed start date AST_i the earliest start of the activity will be the latter of the earliest start calculated according to the network logic and the earliest start as specified. This constraint will simply be satisfied while calculating the activity current start time CS_i. Thus:

$$CS_i = \max \left(AST_i \ , \ \max \left((CF_l - OV_{li}) \mid l \in P_i \right) \right) \qquad (4)$$

where CF_l is the current finish time of activity l, OV_{li} is the overlap (finish to start relationship) between activities i and l and P_i is the set of immediate predecessors of activity i (it is assumed that for the first activity in the project P_1 = null).

The more complex case is the imposed finish date AFT. The latest finish of the activity will be the earlier of the calculated latest finish and the latest finish as specified. If the imposed latest finish is earlier than the earliest finish as calculated, the latest finish is ignored and the finish date is assumed to be the earliest finish date. To accommodate this situation, the upper bound on the activity finish time U is set equal to the activity latest finish time and it will not be updated as improved solutions are obtained. If the activity cannot be assigned a resource and precedence feasible completion time less than or equal to this upper bound, then backtracking occurs. The algorithm will try to reassign previous activities to satisfy this constraint. Failure to do this will mean that the imposed finish date of the activity cannot be satisfied with the specified level of resources.

However, it is better to delay the consideration of an activity imposed finish time until an improved solution is obtained (this is because PH is set equal to the summation of the activity durations which means that the initial values of U are very large.) Of course, this good solution will greatly reduce the U values which in turn will save computational time once backtracking occurs.

5. DESCRIPTION OF THE COMPUTER PROGRAM FOR IMPLEMENTING THE MODIFIED ALGORITHM

In this section of the paper, description of the computer programs developed for implementing the modified implicit enumeration algorithm is introduced. Two programs were developed to provide the required information: program P1 for analysis of construction networks and program P2 for analysis of construction resources. The first will provide activities critical path determined LS and LF timings. The second is used to establish a lower bound on the length of the schedule according to Eq. 2, to sort project activities according to their CPM late start timing and to sort key resources according to Eq. 3. Programs P1 and P2 are written in FORTRAN and are described in Appendix I and II respectively.

Another subprogram was written to implement the scheduling procedure described in section 2 together with the modifications mentioned in section 4. Details of this subprogram are given in Fig. 1 which is a flow diagram for the modified enumeration algorithm. Resource restrictions are maintained through the use of two arrays: a resource total demand matrix TD and a resource availability matrix RAQ. While TD is calculated every time an activity is considered for assignment, RAQ is calculated once at the start of the calculations. To assign activity i a precedence and resource feasible timings such that CS_i equals k_1 and CF_i equals k_2. TD is calculated for resources $k = 1, 2, \ldots, K$ through the period $t = k_1, \ldots, k_2$. Resource usage r_{jk} for those activities which have been considered for scheduling (those with $L_j = 1$) and are alive during this period are then summed up. Of course, the assignment is feasible if $(RAQ - TD)_{kt} \geqslant 0$, $k = 1, \ldots, K$, $t = k_1, \ldots, k_2$.

The new subprogram is now linked to program P2 and thus the latter becomes a resource analysis / resource scheduling program. The computer RAM requirements for program P2, to handle 100 activities, 15 key resources and 260 time periods is 0.123 MB.

6. PRACTICAL APPLICATION

Test problems found in resource scheduling literature contain at most 27 activities and 3 different resource types. Of course, heavy construction projects contain more activities and use more resources. The modified algorithm is used now to schedule the activities of an earth dam project which was studied by Thompson [7].

The project consists of 80 activities and uses 10 key resources. Table 1 gives a list of the project activities, activities predecessors and overlaps and resources usage by the activities. The production levels of the key resources, given in Table 2, are assumed to be constant over contract duration. The project start date is 5. 7.76. The activities imposed start and finish dates (due to the physical constraints of the construction of the earth dam) are given in Table 3.

The project was scheduled by program P2 using an IBM AT 386 computer. The scheduled completion date equals the early start completion date. This means that optimality of the schedule was verified using the lower bound criterion given by Eq. 2. However, about one-half minute of CPU time was required to schedule this project. The resulting schedule is given in Table 4 where all imposed dates have been satisfied. A sample of the resource usage and production reports is shown in Fig. 2.

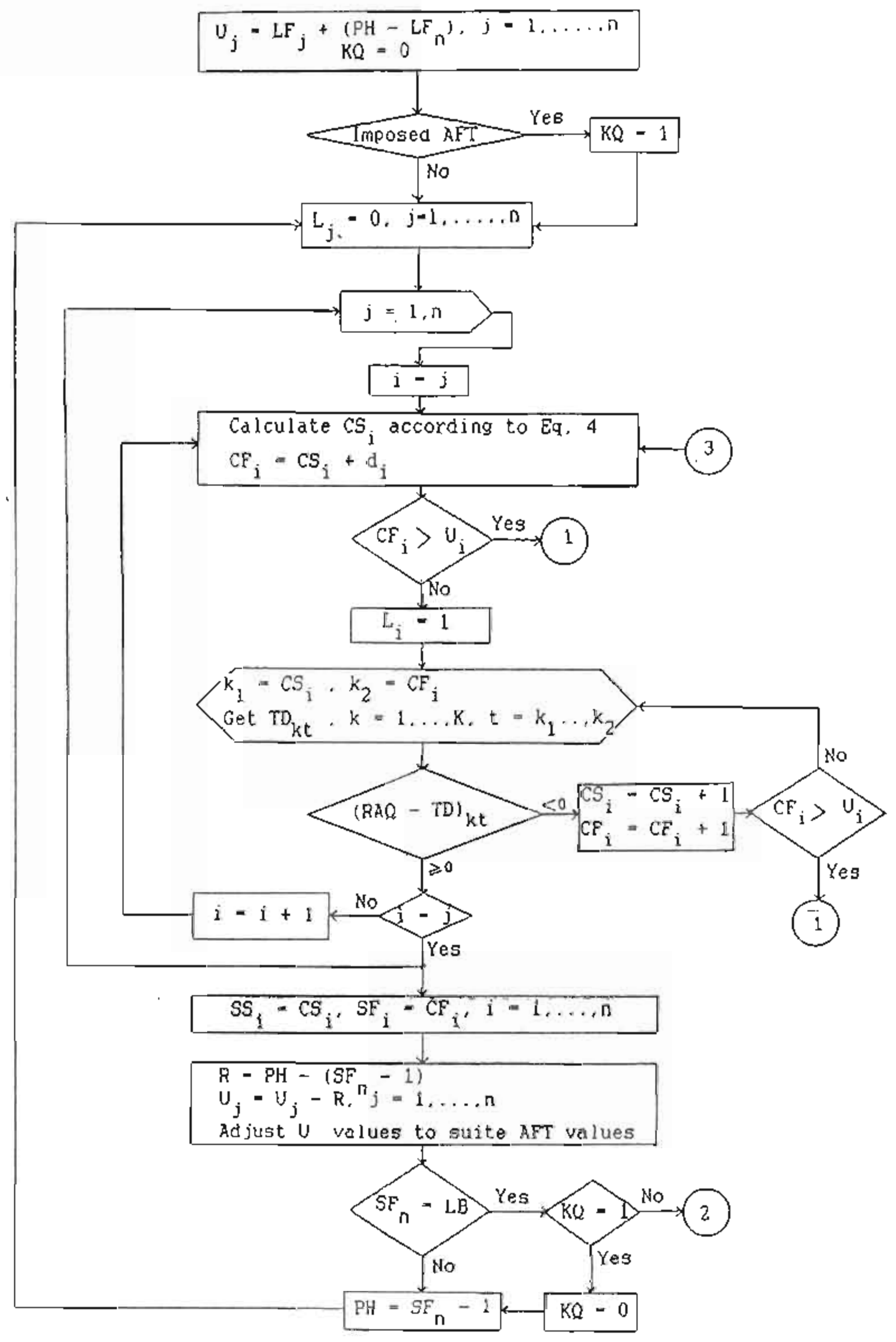
$U_j = LF_j + (PH - LF_n)$, $j = 1, \ldots, n$
KQ = 0
Imposed AFT
Yes
KQ = 1
No
$L_j = 0$, $j = 1, \ldots, n$
j = 1,n
i = j
Calculate CS_i according to Eq. 4
$CF_i = CS_i + d_i$
3
$CF_i > U_i$
Yes
1
No
$L_i = 1$
$k_1 = CS_i$, $k_2 = CF_i$
Get TD_{kt}, $k = 1, \ldots, K$, $t = k_1 \ldots k_2$
No
$(RAQ - TD)_{kt}$
<0
$CS_i = CS_i + 1$
$CF_i = CF_i + 1$
$CF_i > U_i$
Yes
$\bar{1}$
≥0
No
i = i + 1
i = j
Yes
$SS_i = CS_i$, $SF_i = CF_i$, $i = 1, \ldots, n$
$R = PH - (SF_n - 1)$
$U_j = U_j - R$, $j = 1, \ldots, n$
Adjust U values to suite AFT values
$SF_n = LB$
Yes
KQ = 1
No
2
Yes
No
$PH = SF_n - 1$
KQ = 0

Figure 1. Flow Diagram of the Modified Enumeration Algorithm

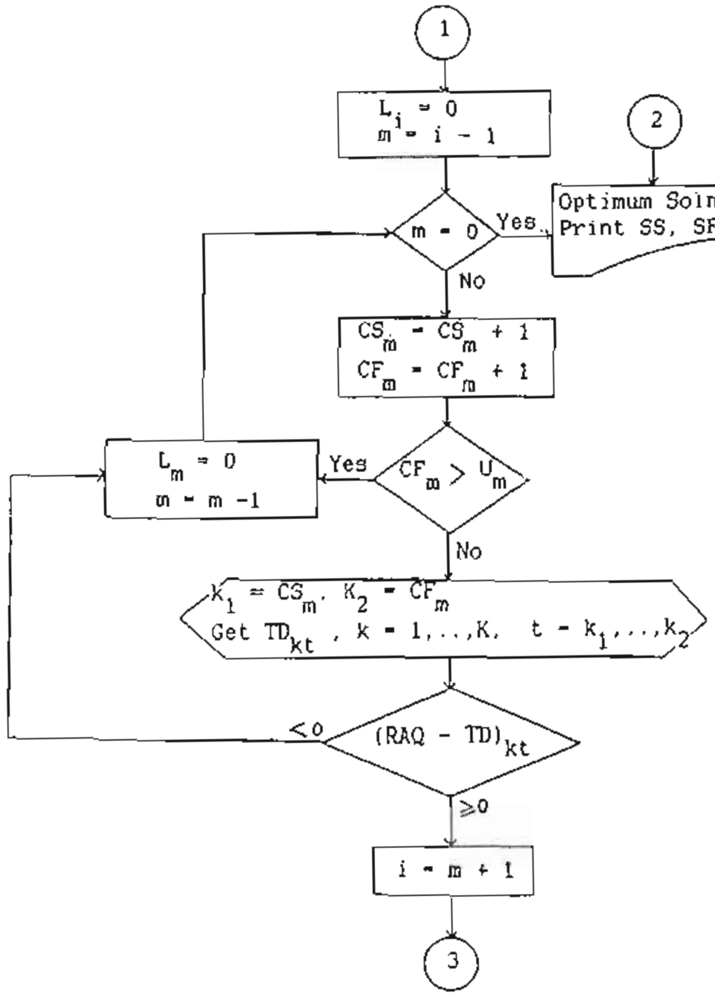

Figure 1. (Continued) Flow Diagram of the Modified Enumeration Algorithm

To test the effect of backtracking on the solution time, the production level of "Drilling & Grouting Plant" had been reduced by one unit and the program was rerun. This case consumed about 12 minutes of CPU time to verify that there is no feasible schedule with this level of production. In fact, with an AT 486 computer, this solution time will evidently be reduced.

Table 1. Data of the Example Project.

NO.	ACTIVITIES	DURATION (wks)	OVERLAPS WITH PREDECESSORS	USAGE OF RESOURCES									
				AIR (cfm)	BULLDOZERS	TRUCKS	GROUT PLT.	EXCAVATORS	DUMPERS	TUNNEL PLT.	CONC. PUMP	DRILLS	GRADERS
10	PRELIMINARY WORK	20		0	0	0	0	0	0	0	0	0	0
20	ESTABLISH SITE	12	9 10	500	1	0	0	0	1	0	0	0	0
30	SITE SERVICES	8	4 10	0	0	0	0	0	0	0	6	0	0
40	OPEN UP WEIR QUARRY	8	10 20	850	1	0	0	1	1	0	0	0	0
50	OPEN UP WATERFALL QUARRY	8	10 20	850	1	0	0	1	1	0	0	0	0
60	ERECT CONCRETE PLANT	6	8 20	0	1	0	0	0	0	0	0	0	0
70	HAUL ROAD 1	8	6 40	250	2	4	0	2	0	0	0	5	1
80	HAUL ROAD 2	12	4 70	250	2	4	0	2	0	0	0	5	1
90	HAUL ROAD 3	16	6 40	250	2	4	0	2	0	0	0	5	1
93	OPEN UP BORROWPITS	6	0 20	0	0	4	0	1	0	0	0	0	0
96	CONSTRUCT TRIAL EMBANKMENT	12	0 93	0	1	4	0	1	0	0	0	0	1
98	RIVER DIVERSION	0	0 202 0 204 0 210	0	0	0	0	0	0	0	0	0	0
99	CLEAR SITE ON COMPLETION	10	0 205 0 206 0 380 0 390 0 395 0 707 0 802	1700	4	4	0	3	0	0	0	0	0
101	CLEAR RESERVIOR AREA	66	0 20	0	1	0	0	0	0	0	0	0	0
102	CONSTRUCT ACCESS ROAD	20	0 40	250	2	4	0	2	0	0	0	5	1
103	CLEAR DAM AREA	32	0 20	0	2	0	0	0	0	0	0	0	0
201	DRIVE DIVERSION TUNNEL EAST	20	0 20 0 30 0 70	1200	0	0	0	0	4	1	0	5	0
202	CONCRETE DIVERSION TUNNEL EAST	24	0 50 0 60 0 201	250	0	0	0	0	4	0	2	0	0
203	DRIVE DIVERSION TUNNEL WEST	10	0 702	1200	0	0	0	0	4	1	0	5	0
204	CONCRETE DIVERSION TUNNEL WEST	15	0 703	250	0	0	0	0	4	0	2	0	0
205	ERECT OUTLET PIPEWORK	12	0 705 0 706 0 0	250	0	0	0	0	0	0	0	0	0
206	TUNNEL OUTLET	4	0 706	0	0	0	0	0	0	0	0	0	0
207	EXC. VALVE TOWER SHAFT & BASE	10	0 20 0 30 8 80	600	0	2	0	0	0	0	0	5	0
208	EXC. FOREBAY	5	0 20 0 30 8 80	400	0	0	0	1	2	0	0	0	0
209	FABRICATE FOREBAY SHUTTERS	6	0 20 0 30	0	0	0	0	0	0	0	0	4	0
210	CONSTRUCT FOREBAY	12	0 50 0 60 2 208 0 209	250	0	0	0	0	1	0	0	0	0
301	CONSTRUCT U.S. COFFERDAM	5	0 40 0 98	0	2	4	0	4	0	0	0	0	1
302	STILLING BASIN COFFERDAM	2	0 301	0	1	4	0	1	0	0	0	0	1
311	EXC. DAM FOUNDATION CENTRE	10	0 103 0 301	0	0	4	0	1	0	0	0	0	0
314	EXC. CUT-OFF TRENCH CENTRE	10	8 311	600	0	4	0	3	0	0	0	0	0
315	CONSTRUCT CUT-OFF WALL CENTRE	8	6 314	600	0	0	0	0	1	0	0	0	0
320	EXC. DAM FOUNDATION N. TO 2040	10	0 311	0	0	3	0	1	0	0	0	0	0
321	EXC. DAM FOUNDATION N. TO 2071	4	0 320 0 102	0	0	3	0	1	0	0	0	0	0
322	EXC. CUT-OFF TRENCH N. TO 2071	5	5 321	400	0	3	0	2	0	0	0	0	0
324	EXC. CUT-OFF TRENCH N. TO 2040	4	8 320	400	0	3	0	2	0	0	0	0	0
325	CONST. CUT-OFF WALL N. TO 2040	4	3 324	600	0	0	0	0	1	0	0	0	0
327	CONST. CUT-OFF WALL N. TO 2071	8	2 322	600	0	0	0	0	1	0	0	0	0
330	EXC. DAM FOUNDATION S. TO 2040	10	0 311	0	0	3	0	1	0	0	0	0	0
331	EXC. DAM FOUNDATION S. TO 2071	4	0 330	0	0	3	0	1	0	0	0	0	0
332	EXC. CUT-OFF TRENCH S. TO 2071	6	3 331	400	0	3	0	2	0	0	0	0	0
334	EXC. CUT-OFF TRENCH S. TO 2040	10	8 330	600	0	3	0	2	0	0	0	0	0
335	CONST. CUT-OFF WALL S. TO 2040	14	8 334	600	0	0	0	0	0	0	0	0	0
337	CONST. CUT-OFF WALL S. TO 2071	20	5 332	600	0	0	0	0	1	0	0	0	0
340	CONSTRUCT EMBANKMENT TO 2025	9	7 918 0 96 0 405	0	3	10	0	5	0	0	0	0	2
350	CONSTRUCT EMBANKMENT TO 2040	13	0 340 0 407 4 926 4 936	0	3	10	0	5	0	0	0	0	2
360	CONSTRUCT EMBANKMENT TO 2050	9	0 350 0 408 9 928 9 938	0	3	10	0	5	0	0	0	0	2
370	COMPLETE EMBANKMENT	13	0 360	0	3	10	0	5	0	0	0	0	2
380	PLACE RIP-RAP	14	13 370	0	0	2	0	2	0	0	0	0	0
390	TOPSOIL EMBANKMENT	16	13 370	0	1	0	0	0	4	0	0	0	0
395	CONSTRUCT CREST ROAD	12	0 370	250	1	0	0	0	0	0	0	0	1
401	FABRICATE SPILLWAY SHUTTERS 1	6	0 20 0 30	0	0	0	0	0	0	0	0	0	0
402	FABRICATE SPILLWAY SHUTTERS 2	6	0 20 0 30	0	0	0	0	0	0	0	0	0	0
403	FABRICATE SPILLWAY SHUTTERS 3	20	0 20 0 30	0	0	0	0	0	0	0	0	0	0
404	EXC. SPILLWAY BELLMOUTH BASE	4	0 301	600	0	2	0	1	0	0	0	0	0
405	CONST. SPILLWAY BASE TO 2025	14	0 401 0 502 2 946	400	0	0	0	0	3	0	0	0	0
406	CONST. SPILLWAY BASE TO 2033	4	0 405	400	0	0	0	0	3	0	0	0	0
407	CONST. SPILLWAY TOWER TO 2044	4	0 402 0 406	400	0	0	0	0	3	0	0	0	0
408	CONST. SPILLWAY TO 2056	12	0 403 0 407	400	0	0	0	0	2	0	0	0	0
409	COMPLETE SPILLWAY BELLMOUTH	20	0 408	400	0	0	0	0	2	0	0	0	0
501	DRIVE SPILLWAY TUNNEL EAST	25	0 20 0 30 10 90	2400	0	0	0	0	4	2	0	7	0
502	DRIVE SPILLWAY TUNNEL WEST	5	0 404	2400	0	0	0	0	4	2	0	7	0
503	CONCRETE SPILLWAY TUNNEL	30	0 50 0 60 0 501 0 502	250	0	0	0	0	4	0	2	0	0
601	FABR. STILLING BASIN SHUTTERS	12	0 20 0 30	0	0	0	0	0	0	0	0	0	0
602	EXC. STILLING BASIN	8	0 302	600	0	4	0	3	0	0	0	0	0
603	CONSTRUCT STILLING BASIN	24	0 601 4 602	400	0	0	0	0	4	0	0	0	0
701	FABRICATE VALVE TOWER SHUTTER	10	0 20 0 30	0	0	0	0	0	0	0	0	0	0
702	CONCRETE VALVE TOWER SHAFT	3	0 50 0 60 0 207	250	0	0	0	0	2	0	0	0	0
703	CONCRETE VALVE TOWER BASE	5	0 203 0 701	250	0	0	0	0	1	0	0	0	0
704	CONSTRUCT VALVE TOWER	20	0 703	250	0	0	0	0	1	0	0	0	0
705	ERECT VALVE TOWER PIPEWORK	3	0 801	250	0	0	0	0	0	0	0	0	0
706	PLUS DIVERSION TUNNEL	6	0 101 0 409 0 503 0 603 0 370	250	0	0	0	0	1	0	0	0	0
707	LINING OF SCOUR AREA	4	4 706	0	0	0	0	0	0	0	0	0	0
801	CONSTRUCT VALVE TOWER BRIDGE	12	0 202 0 704	250	0	0	0	0	1	0	0	0	0
802	CONSTRUCT SPILLWAY BRIDGE	16	0 370 0 409	250	0	0	0	0	1	0	0	0	0
918	GROUT CUT-OFF CENTRE	28	8 315	250	0	0	4	0	2	0	0	0	0
926	GROUT CUT-OFF N. TO 2040	22	4 325	250	0	0	2	0	1	0	0	0	0
928	GROUT CUT-OFF N. TO 2090	15	6 327	250	0	0	3	0	1	0	0	0	0
936	GROUT CUT-OFF S. TO 2040	17	14 335	250	0	0	4	0	0	0	0	0	0
938	GROUT CUT-OFF S. TO 2090	17	10 337	250	0	0	3	0	0	0	0	0	0
946	GROUT BELLMOUTH BASE	3	2 404	250	0	0	2	0	0	0	0	0	0

Table 2 Resources Production levels of the Example Project

Resource	Production Level
Compressed Air	3450 cfm
Bulldozers	5 No.
Dump Trucks	10 No.
Drilling & Grouting Plant	10 No.
Excavators & Loaders	5 No.
Dumpers	8 No.
Tunnel Plant	2 No.
Concrete Pumps	4 No.
Rock Drills	15 No.
Graders	2 No.

Table 3 Activities Imposed Start and Finish Dates

No.	Activity	AST	AFT
208	Excavate Forebay	27.12.76	6. 2.77
98	River Diversion	2. 1.78	2. 1.78
340	Construct Embankment 1	11. 9.78	12.11.78
350	Construct Embankment 2	8. 1.79	8. 4.79
360	Construct Embankment 3	3. 9.79	4.11.79
370	Construct Embankment 4	31.12.79	30. 3.80

7. SUMMARY AND CONCLUSION

A step has been taken to implement optimal solution approaches for solving the resource scheduling problem in construction. The implicit enumeration procedure has been modified to suit the requirements of construction projects. Three improvements have been used to reduce the algorithm computational time. A computer program has been developed to carry the modified algorithm into effect.

Application to a real-life construction project has been demonstrated. The algorithm guarantees the optimal schedule and its computer RAM requirement is very reasonable. The solution time using today's powerful computers makes this algorithm a promising approach to construction management.

Table 4. Optimum Schedule of the Example Project.

CHANIA DAM (BASIC MODEL)

START DATE: 5. 7.76 TOTAL DURATION = 223 WEEKS FINISH DATE: 12.10.80

NO	ACTIVITY	DURATION	SCHEDULED WEEKS SS	SF	SCHEDULED DATES SS	SF
10	PRELIMINARY WORK	20	1	20	5. 7.76	21.11.76
20	ESTABLISH SITE	12	12	23	20. 9.76	12.12.76
30	SITE SERVICES	8	17	24	25.10.76	19.12.76
40	OPEN UP WEIR QUARRY	8	14	21	4.10.76	28.11.76
50	OPEN UP WATERFALL QUARRY	8	14	21	4.10.76	28.11.76
60	ERECT CONCRETE PLANT	6	24	29	13.12.76	23. 1.77
70	HAUL ROAD 1	8	16	23	18.10.76	12.12.76
80	HAUL ROAD 2	12	22	33	29.11.76	20. 2.77
90	HAUL ROAD 3	16	42	57	18. 4.77	7. 8.77
95	OPEN UP BORROWPITS	6	24	29	13.12.76	23. 1.77
96	CONSTRUCT TRIAL EMBANKMENT	12	30	41	24. 1.77	17. 4.77
98	RIVER DIVERSION	0	79	79	2. 1.78	2. 1.78
99	CLEAR SITE ON COMPLETION	10	214	223	4. 8.80	12.10.80
101	CLEAR RESERVIOR AREA	66	34	99	21. 2.77	28. 5.78
102	CONSTRUCT ACCESS ROAD	20	56	75	25. 7.77	11.12.77
103	CLEAR DAM AREA	32	24	55	13.12.76	24. 7.77
201	DRIVE DIVERSION TUNNEL EAST	20	25	44	20.12.76	8. 5.77
202	CONCRETE DIVERSION TUNNEL EAST	24	45	68	9. 5.77	23.10.77
203	DRIVE DIVERSION TUNNEL WEST	10	39	48	28. 3.77	5. 6.77
204	CONCRETE DIVERSION TUNNEL WEST	15	61	75	29. 8.77	11.12.77
205	ERECT OUTLET PIPEWORK	12	202	213	12. 5.80	3. 8.80
206	TUNNEL OUTLET	4	202	205	12. 5.80	8. 6.80
207	EXC. VALVE TOWER SHAFT & BASE	10	26	35	27.12.76	6. 3.77
208	EXC. FOREBAY	6	26	31	27.12.76	6. 2.77
209	FABRICATE FOREBAY SHUTTERS	6	25	30	20.12.76	30. 1.77
210	CONSTRUCT FOREBAY	12	49	60	6. 6.77	28. 8.77
301	CONSTRUCT U.S. COFFERDAM	6	79	84	2. 1.78	12. 2.78
302	STILLING BASIN COFFERDAM	2	85	86	13. 2.78	26. 2.78
311	EXC. DAM FOUNDATION CENTRE	10	85	94	13. 2.78	23. 4.78
314	EXC. CUT-OFF TRENCH CENTRE	10	87	96	27. 2.78	7. 5.78
315	CONSTRUCT CUT-OFF WALL CENTRE	8	94	101	17. 4.78	11. 6.78
320	EXC. DAM FOUNDATION N. TO 2040	10	95	104	24. 4.78	2. 7.78
321	EXC. DAM FOUNDATION N. TO 2071	4	109	112	31. 7.78	27. 8.78
322	EXC. CUT-OFF TRENCH N. TO 2071	3	111	113	14. 8.78	3. 9.78
324	EXC. CUT-OFF TRENCH N. TO 2040	4	97	100	8. 5.78	4. 6.78
325	CONST. CUT-OFF WALL N. TO 2040	4	98	101	15. 5.78	11. 6.78
327	CONST. CUT-OFF WALL N. TO 2071	8	140	147	5. 3.79	29. 4.79
330	EXC. DAM FOUNDATION S. TO 2040	10	95	104	24. 4.78	2. 7.78
331	EXC. DAM FOUNDATION S. TO 2071	4	105	108	3. 7.78	30. 7.78
332	EXC. CUT-OFF TRENCH S. TO 2071	6	106	111	10. 7.78	20. 8.78
334	EXC. CUT-OFF TRENCH S. TO 2040	10	101	110	5. 6.78	13. 8.78
335	CONST. CUT-OFF WALL S. TO 2040	14	103	116	19. 6.78	24. 9.78
337	CONST. CUT-OFF WALL S. TO 2071	20	120	139	16.10.78	4. 3.79
340	CONSTRUCT EMBANKMENT TO 2025	9	115	123	11. 9.78	12.11.78
350	CONSTRUCT EMBANKMENT TO 2040	13	132	144	8. 1.79	8. 4.79
360	CONSTRUCT EMBANKMENT TO 2050	9	166	174	3. 9.79	4.11.79
370	COMPLETE EMBANKMENT	13	183	195	31.12.79	30. 3.80
380	PLACE RIP-RAP	14	196	209	31. 3.80	6. 7.80
390	TOPSOIL EMBANKMENT	16	183	198	31.12.79	20. 4.80
395	CONSTRUCT CREST ROAD	12	196	207	31. 3.80	22. 6.80
401	FABRICATE SPILLWAY SHUTTERS 1	6	25	30	20.12.76	30. 1.77
402	FABRICATE SPILLWAY SHUTTERS 2	6	25	30	20.12.76	30. 1.77
403	FABRICATE SPILLWAY SHUTTERS 3	20	25	44	20.12.76	8. 5.77
404	EXC. SPILLWAY BELLMOUTH BASE	4	85	88	13. 2.78	12. 3.78
405	CONST. SPILLWAY BASE TO 2025	14	94	107	17. 4.78	23. 7.78
406	CONST. SPILLWAY BASE TO 2035	4	108	111	24. 7.78	20. 8.78
407	CONST. SPILLWAY TOWER TO 2044	4	112	115	21. 8.78	17. 9.78
408	CONST. SPILLWAY TO 2056	12	122	133	30.10.78	21. 1.79
409	COMPLETE SPILLWAY BELLMOUTH	20	172	191	15.10.79	2. 3.80
501	DRIVE SPILLWAY TUNNEL EAST	25	117	141	25. 9.78	18. 3.79
502	DRIVE SPILLWAY TUNNEL WEST	5	89	93	13. 3.78	16. 4.78
503	CONCRETE SPILLWAY TUNNEL	30	142	171	19. 3.79	14.10.79
601	FABR. STILLING BASIN SHUTTERS	12	25	36	20.12.76	13. 3.77
602	EXC. STILLING BASIN	8	145	152	9. 4.79	3. 6.79
603	CONSTRUCT STILLING BASIN	24	149	172	7. 5.79	21.10.79
701	FABRICATE VALVE TOWER SHUTTER	10	25	34	20.12.76	27. 2.77
702	CONCRETE VALVE TOWER SHAFT	3	36	38	7. 3.77	27. 3.77
703	CONCRETE VALVE TOWER BASE	8	49	56	6. 6.77	31. 7.77
704	CONSTRUCT VALVE TOWER	20	69	88	24.10.77	12. 3.78
705	ERECT VALVE TOWER PIPEWORK	8	106	113	10. 7.78	3. 9.78
706	PLUG DIVERSION TUNNEL	6	196	201	31. 3.80	11. 5.80
707	LINING OF SCOUR AREA	4	202	205	12. 5.80	8. 6.80
801	CONSTRUCT VALVE TOWER BRIDGE	12	94	105	17. 4.78	9. 7.78
802	CONSTRUCT SPILLWAY BRIDGE	16	196	211	31. 3.80	20. 7.80
918	GROUT CUT-OFF CENTRE	28	94	121	17. 4.78	29.10.78
926	GROUT CUT-OFF N. TO 2040	22	98	119	15. 5.78	15.10.78
928	GROUT CUT-OFF N. TO 2090	16	142	157	19. 3.79	8. 7.79
936	GROUT CUT-OFF S. TO 2040	17	103	119	19. 6.78	15.10.78
938	GROUT CUT-OFF S. TO 2090	17	134	150	22. 1.79	20. 5.79
946	GROUT BELLMOUTH BASE	6	87	91	27. 2.78	23. 4.78

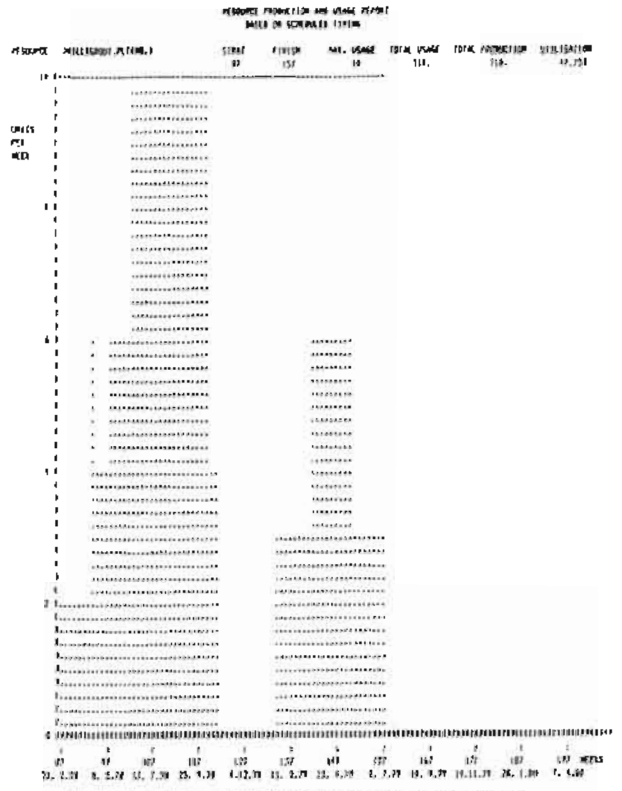

Figure 2. Histogram of Resource "Drilling & Grouting Plant"

REFERENCES

1. Davis, E.W. and Heidorn, G.E., " An Algorithm for Optimal Project Scheduling under Multiple Resource Constraints," Management Science, 17, 12 (August 1971), B803 - B816.
2. Stinson, J.P., Davis, E.W. and Khumawala, B.M., " Multiple Resource Constrained Scheduling Using Branch and Bound," AIIE Trans., 10, 3 (September 1978), 252 - 259.
3. Talbot, F.B. and Patterson, J.H., "An Efficient Integer Programming Algorithm With Network Cuts for Solving Resource Constrained Scheduling Problems," Management Science, 24, 11 (July 1978), 1163 - 1174.
4. Patterson, J.H., "A Comparison of Exact Approaches for Solving the Multiple Constrained Resource Project Scheduling Problem," Management Science, 30, 7 (July 1984), 854 - 867.
5. Willis, E.M., "Scheduling Construction Projects," John Wiley & Sons, 1986.
6. Talbot, F.B., "Resource Constrained Project Scheduling With Time-Resource Tradeoffs: The Nonpreemptive Case," Management Science, 28, 10 (October 1982), 1197 - 1210.
7. Thompson, P.A., "Construction of Chania Dam : Method Statement and Estimate," Unpublished Report, UMIST, UK, 1976.

Appendix I - Description of Program P1

The network analysis program, named P1, establishes a complete activity schedule. In order to perform this function, it calls the following ten subroutines:

1. SUBROUTINE ADATA which reads in activity data through four read statements. In the project line, the user provides project title, project start day, month and year, number of activities and method of specifying overlaps between the activities. In the activity lines, the user gives activities number, title, duration and imposed start and finish dates. In the precedence logic lines, the user supplies activity number and its preceding activities with overlaps in between. At the end, the user specifies information required to control volume of printing.
2. SUBROUTINE PDATE which formulates the project calendar that can be used to get calendar date scheduling.
3. SUBROUTINE LOGIC which checks network logic. It detects network logical errors such as existence of more than one start or finish activity. It also checks overlap values which must not be greater than a preceding activity duration.
4. SUBROUTINE ASTEP which assigns a sequence step to each activity in the project. The sequence step is the earliest logical position in the network that an activity can occupy while maintaining its proper dependencies. This information is helpful in network drawing and it is used to guarantee that the set of activities are not in a loop.
5. SUBROUTINE FPASS which determines activities early start and early finish times through forward pass of network calculations.
6. SUBROUTINE BPASS which determines activities late finish and late start times through backward pass of network calculations.
7. SUBROUTINE SLACK which calculates activities total, free and back floats.
8. SUBROUTINE SORT is used to sort activities according to their activity numbers.
9. SUBROUTINE AOUT which prints project calendar, activities successors and activities sequence step (if required.)
10. SUBROUTINE NRES which prints a report containing information about activities timings with or without calendar dates, activities floats and project completion date.

Note: The program will create a file containing all necessary information about project activities that will be required by a resource analysis program.

Appendix II - Description of Program P2

The resource analysis program; named P2, forecasts period by period resource usage. In order to perform this function, it reads the data file containing necessary information about project activities (which was created by program P1) and then calls the following subroutines:

1. SUBROUTINE RDATA which reads in resource data through five read statements: In the start line, the user provides total number of resources and an index that determines timing for resource analysis (early start, late start, or schedule start.) In the resource lines, the user supplies resource number, resource title and resource type (key or secondary resource.) In the resource usage lines, the user gives activity number, reference numbers of the resources used by the activity and number of units of the resources used by the activity. In the resource information lines, the user specifies the resource reference number, number of times of resource production change and the corresponding production levels. At the end the user chooses resources to be included in the output report which will be in the form of tables or histograms.
2. SUBROUTINE RANL which forecasts total demand and usage of the resources and their start and finish times according to the specified timing.
3. SUBROUTINE ROUT which prints out resource production and usage reports in the specified form.