European Journal of Operational Research 49 (1990) 3–13
North-Holland

Some efficient multi-heuristic procedures for resource-constrained project scheduling

Fayez F. Boctor
Faculté des Sciences de l'Administration, Université Laval, Québec, Canada

Abstract: Various heuristic procedures have been proposed to solve the well known NP-hard, resource-constrained project scheduling problem. Most of these procedures are parallel heuristics. In this paper, some multi-heuristic procedures employing parallel rules as well as serial rules are suggested. These multi-heuristics were compared to 13 single heuristic procedures. This comparison is based on a set of 36 small problems (5 to 20 activities) and 30 large scale projects (38 to 111 activities). The main performance measure used to assess the efficiency of each heuristic was the project completion time expressed as a percentage of either the optimal duration (for small problems) or the critical path length (for large projects).

As an example, a three-heuristic procedure yielded the optimum schedule in 27 of the 36 small problems (75%) and produced the shortest duration schedule for 56 of the 66 test problems (85%).

Keywords: Project scheduling, resource allocation, heuristics

1. Introduction

Since the pioneering work of Kelley (1963) and Wiest (1963) the resource-constrained project scheduling problem has occupied a great number of researchers. Over the past 25 years there have been more than 80 publications and theses that have addressed the different versions of this challenging problem. Some survey papers and monographs that describe the 'state of the art' are available (Elmaghraby, 1977; Davis, 1973, 1974; Herroelen, 1972) but not completely up-to-date.

The general resource-constrained project scheduling problem can be stated as follows: "Given a set of interrelated activities (precedence relations) where each activity can be performed in one of several modes (ways) and each mode is characterized by a known duration and given resource requirements, when should each activity begin and which resource–duration mode should be adopted so as to optimize some managerial objective? Obviously, the solution has to respect the precedence relations and resource limits."

Different versions of this problem are studied in the literature. These versions can be divided into categories according to the number of simultaneously scheduled projects (*single* or *multiple*), the nature of the optimized objective function, the nature of the employed resources, and what will be called the preemption condition.

Both in the single- and multi-project cases, the most widely used objective has been the minimization of *project duration* (Brand et al., 1964; Wiest, 1967; Pritsker et al., 1969; Fisher, 1970; Davis et al., 1971; Patterson et al., 1974; Talbot et al., 1978). The other frequently reported objectives are the minimization of the *total project cost* (Slowinski, 1980, 1981; Talbot, 1982) and the maximization of the *project net present value* (Doersch et al., 1977). In addition, in the multi-project case we may optimize the tardiness penality or any other function of individual project completion dates.

Resources may be available in limited quantities but *renewable* from period to period. Labor hours are a good example of this resource cate-

Received November 1989

gory. If the total amount of a resource is limited over the project life, the resource is called *nonrenewable*. An example of such a resource is money. If both the total amount over the project life and the per period availability are limited, the resource is said to be *doubly constrained*. In most of the published research it is assumed that resources are renewable. Doubly constrained resources are treated in Slowinski (1980, 1981), Weglarz (1981) and Talbot (1982).

Two alternative preemption assumptions may be made. If activities, once started, cannot be interrupted then we have a *nonpreemptive* case. Otherwise, it is *preemptive*. Most of the published research addressas the nonpreemptive case. The preemptive case was studied in Schrage (1972), Weglarz et al. (1977) and Slowinski (1980, 1981).

Note that most of the papers mentioned do not treat the general problem—where each activity could be performed in one of several ways—but address the restricted version where each activity has only *one* duration–resource option. Only a few recent works attempt to solve the general problem where activities are characterized by a continuous duration–resource function (Weglarz, 1976; Slowinski et al., 1978; Weglarz, 1980) or by a discrete duration–resource function (Slowinski, 1980, 1981, Talbot, 1982). Hereafter, we consider the usual version of the problem where resources are renewable, activities can not be interrupted, there is only one execution mode for each activity and the managerial objective is to minimize the project delay.

However, all these problem varieties are NP-hard. Thus, since projects often consist of hundreds or thousands of activities, heuristics are the only computationally-feasible solution methods. Unfortunately, the performance of heuristics depends on problem characteristics and it is quite difficult to predict, beforehand, the most efficient heuristic for a given problem. Davis (1975) and Patterson (1976) attempted to determine problem parameters which may determine the efficiency of some heuristic sequencing rules. Patterson suggested using these parameters to preselect one or several heuristics. However, we do not have any idea of the probability that the first heuristic selected produces the best solution or about the number of heuristics we have to use in order to obtain the best schedule.

Table 1
Tested heuristic rules

Sequencing rules tested within a 'parallel approach' [a]

MINSLK	Activity having the smallest slack in scheduled first
RSM	Resource scheduling method (Brand et al., 1964). At time t calculate for each pair of schedulable activities i and j the index $x_{ij} = \max(0,\ t + \text{duration of } i\text{-th late start time of } j)$. Schedule first the activity having the smallest x_{ij}
MINLFT	Give precedence to the activity having the smallest late finish time (as given by the critical path analysis)
RAN	Select an activity randomly
GRD	Give priority to the activity which requires the greatest number of resource units of all types
SA	Choose first the shortest activity
SRD	Smallest resource demand (the inverse of GRD)
LA	Choose first the longest activity

Sequencing rules tested within a 'serial approach'

In all cases, activities are sorted in the ascending order of a first key and in case of tie, we use a second sorting key. Hereafter we give the first and second keys of each tested rule

$J-I$	(1) Final node number	(2) Initial node number (ascending order)
$J-D$	(1) Final node number	(2) Duration (ascending order)
$J-R$	(1) Final node number	(2) Total resource requirements (descending order)
$J-S$	(1) Final node number	(2) Total slack (ascending order)
$I-J$	(1) Initial node number	(2) Final node number (ascending order)

[a] Ties were broken by choosing activity having the lowest number.

[b] To obtain node numbers we have to draw the 'activity-on-arrow' diagram.

The approach proposed hereafter is different from Patterson's approach. We suggest, whatever the problem characteristics, using a preselected combination of heuristic rules and retaining the best solution. To identify the most efficient combinations of rules we proceeded as follows. First, based on 66 projects of different degrees of complexity, we evaluated the performance of 13 sequencing rules. Then, based on these results, the best combinations of 2, 3, 4,... rules were determined. Obviously, the same approach can be used to determine the best heuristic combinations for the other versions of the resource-constrained project scheduling problem.

In this paper it will be shown that, if the objective is to minimize the project duration, a combination of three heuristics has a relatively high probability of giving the best and even the optimum solution. By the best solution we mean the best among the 13 solutions given by the 13 tested heuristics. Evidently, a combination of 4 or 5 heuristics can lead to better results. The relation between the probability of getting the best solution and the number of heuristic rules to be used is also analyzed.

2. Test problems and sequencing rules

The set of test problems involves two subsets. The first subset is composed of 36 small problems containing 5 to 20 activities and using up to 3 resource types. These problems were derived by varying resource availabilities in 15 problems found in the literature. The 30 real-life projects included in the second subset are composed of 38 to 111 activities. The main charcteristics of these problems are given in Appendix 1.

The performance of 13 selected sequencing heuristic rules was studied. Eight of these rules were employed within a parallel approach (priority indices are updated each time an activity is scheduled) while the other five rules were employed within a serial approach (priority indices are determined once before starting the scheduling operation). For more details about parallel and serial approaches see Kelley (1963). Table 1 exhibits the sequencing rules that are used in this study.

3. Results

Each of the 66 problems was solved with each of the 13 selected heuristics. The results obtained are given in Appendix 2. Let us first examine the performance of each individual rule; in the next section we suggest some efficient combinations of rules.

In Table 2, we give some performance measures to asses the efficiency of each of the 13 heuristics. The table contains two types of measures: those based on the average percentage increase in project duration (the first 4 lines), and those based on the number of times each heuristic proeduced the optimum or the shortest duration (the last 4 lines). The heuristics in the table are arranged in ascending order of the first measure. According to all of the used measures, we may conclude that there exist three performance classes:

(i) the most efficient rules (MINSLK, MINLFT and RSM),

(ii) the low performance class (SRD and SA), and

(iii) the average performers (all the others).

Based on these measures we can also see that the MINSLK rule is the most efficient one. However, the probability that this rule gives the shortest duration is only 65.15% ($\frac{43}{66}$) with a 99 percent confidence interval from 51.5% to 78.8%, which is not really satisfactory. As will be shown in the next section, higher probabilities can be obtained if we use a combination of rules.

It may be important to notice that, in general, the rules that were tested performed better with small problems than with large projects. The only heuristic which gave comparable percentages of best solutions in the two cases is the MINSLK heuristic.

Finally, we observe that serial heuristics are much faster than parallel ones. On average, a parallel heuristic, except the RSM heuristic, requires about 5 times the CPU time required by a serial heuristic. The RSM heuristic needs 3 times the CPU time of other parallel heuristics.

4. Efficient rule combinations

To get good schedules it has been suggested by some researchers that more than one heuristic be used and that the best of the solutions obtained be

Table 2
Some performance measures

Rule	MINSLK	MINLFT	RSM	GRD	$J-I$	$J-R$	RAN	$I-J$	LA	$J-S$	$J-D$	SA	SRD
Average percentage increase above the optimum: Small problems	6.05	6.94	8.20	8.42	9.13	9.13	9.74	10.86	11.32	11.51	15.43	17.16	19.87
Average percentage increase above the critical path length: Small problems	30.03	31.37	33.37	33.21	34.15	34.19	34.73	36.14	36.52	36.96	41.50	43.51	46.75
Average percentage increase above the critical path length: Large projects	13.91	16.13	15.67	19.84	20.35	19.72	16.51	18.64	19.45	20.51	20.79	19.11	21.36
Average percentage increase above the critical path length: All problems	22.70	24.45	25.33	27.12	27.88	27.61	26.45	28.19	28.76	29.48	32.09	32.42	35.21
Number of times optimum duration was produced: Small problems	22	22	18	17	15	15	16	13	18	11	12	8	6
Number of times the shortest duration was produced: Small problems	23	23	19	18	16	16	17	13	18	12	13	10	9
Number of times the shortest duration was produced: Large projects	20	11	14	4	6	7	6	4	9	5	3	2	1
Number of times the shortest duration was produced: All problems	43	34	33	22	22	23	23	17	27	17	16	12	10

Table 3
Efficient heuristic combinations

Number of heuristics	Heuristic combination	Number of times the best solution was obtained	Probability of getting the best solution	99% confidence interval	Average percentage increase above the critical path length	Standard deviation
2	MINSLK, MINLFT	51	77.3	65.3– 89.3	0.2145	0.0392
	MINSLK, $J-R$	50	75.8	63.5– 88.0	0.2128	0.0431
	MINSKL, GRD	49	74.2	61.7– 86.8	0.2132	0.0391
	MINSLK, RSM	49	74.2	61.7– 86.8	0.2185	0.0440
3	MINSLK, RSM, MINLFT	56	84.8	74.6– 95.1	0.2067	0.0404
	MINSLK, MINLFT, $J-R$	55	83.3	72.7– 94.0	0.2047	0.0399
	MINSLK, MINLFT, GRD	55	83.3	72.7– 94.0	0.2075	0.0382
	MINSLK, RSM, GRD	55	83.3	72.7– 94.0	0.2055	0.0402
	MINSLK, GRD, $J-R$	55	83.3	72.7– 94.0	0.2029	0.0398
4	MINSLK, RSM, MINLFT, GRD	60	90.0	82.7– 99.1	0.2005	0.0392
	MINSLK, RSM, MINLFT, SA	59	89.4	80.6– 98.2	0.2000	0.0388
	MINSLK, RSM, MINLFT, SRD	59	89.4	80.6– 98.2	0.2008	0.0392
	MINSLK, RSM, GRD, SA	59	89.4	80.6– 98.2	0.1985	0.0387
	MINSLK, MINLFT, GRD, $J-R$	59	89.4	80.6– 98.2	0.1985	0.0388
5	MINSLK, RSM, MINLFT, GRD, SA	63	95.5	89.5–100	0.1939	0.0376
	MINSLK, RSM, MINLFT, GRD, SRD	63	95.5	89.5–100	0.1947	0.0380
	MINSLK, MINLFT, GRD, SRD, $J-R$	62	93.9	87.1–100	0.1927	0.0375
	MINSLK, MINLFT, GRD, SA, $J-R$	62	93.9	87.1–100	0.1927	0.0375
	MINSLK, RSM, MINLFT, GRD, $J-R$	62	93.9	87.1–100	0.1976	0.0390

retained. However, it is difficult to take advantage of this suggestion if we do not know how many heuristics to use and which ones. It is the purpose of this section to suggest some efficient combinations of heuristics.

Table 3 gives the most efficient combinations of those heuristics tested. As shown, using a two-heuristic combination, the number of times the best solution can be obtained may be as high as 51 (77%). This number increases to 56, 60 and 63 if we use, respectively, a 3- ,4- or 5-heuristic combination. In many cases, it may be sufficient to use a combination of three heuristics. If this is the case, Table 3 gives five such combinations. It is useful to notice that some of these combinations (those including the RSM parallel heuristic) consume more computer processing time than others (especially those including the $J-R$ serial heuristic). Table 3 indicates also that the MINSLK rule is common in all efficient combinations. On the other hand, some heuristics such as RAN, LA, $J-I$, $J-D$, $J-S$ and $I-J$ are not included in any of the efficient combinations indicated in Table 3.

Table 4
Rules to be used within the suggested scheme and their order

Order	Rule	Cumulative probability of obtaining the best solution (%)	99% confidence interval (%)
1	MINSLK	65.1	51.5– 78.8
2	MINLFT	77.3	65.3– 89.3
3	RSM	84.8	74.6– 95.1
4	GRD	90.9	82.7– 99.1
5	SA (or SRD)	95.5	89.5–100
6	$J-I$	97	92.1–100
7	$J-R$	100	

Further, if we examine the results given in Appendix 2, we can see that to obtain the best solution for all of the 36 small problems we have to use any one of the following four 6-heuristic combinations: MINSLK, RSM, either MINLFT or $I-J$, GRD, either SA or SRD, and $J-R$. With respect to the set of large projects, we need to use a combination of only 5 heuristics: MINSLK, RSM, MINLFT, GRD and $J-I$. For the whole set of 66 problems we have to use one of following two 7-heuristic combinations: MINSLK, RSM, MINLFT, GRD, either SA or SRD, J–I and J–R.

A good strategy for solving any resource-constrained project scheduling problem is to try first one of the most efficient heuristics. If the results are not satisfactory, choose one of those heuristics which make an efficient combination with the first one and use it to construct another schedule. If results are still unsatisfactory choose a third heuristic, and so on. To be sure of obtaining the highest cumulative probability of getting the best solution we can use the sequence of heuristics suggested in Table 4. However, as the RSM heuristic requires more computational time than others, we may prefer to use as a last resort in this case one of the best sequences: MINSLK, MINLFT, J–R, GRD, SA, J–I and RSM. The corresponding cumulative probabilities of obtaining the best solution are: 65.1, 77.3, 83.3, 89.4, 93.9, 95.5 and 100%.

5. Summary and conclusions

This paper is confined to the usual version of the resource-constrained project scheduling problem. Namely, the problem addressed here is that in which resources are renewable, activities cannot be interrupted, there is only one execution mode for each activity, and the managerial objective is to minimize the project delay. To obtain a good solution, it is suggested that more than one heuristic be used to solve the problem and that the best solution be retained.

This paper also shows that a combination of three heuristics have a high probability of giving the best and even the optimum solution. These probabilities were estimated to be as high as 85% (with 99% confidence interval from 75% to 95%) for the best solution among the 13 methods tried and 75% (with 99% confidence interval from 58% to 92%) for the optimum solution. A scheme for selecting the heuristics to be used with any problem was also suggested.

Given these results a natural question is now whether they can be extended to problems of larger size. It may be very expensive to design and undertake a statistical experiment to answer this question. A better research issue may be that of identifying the most efficient heuristic combinations to solve the other versions of the resource-constrained project scheduling problem. Presently, we are exploring this research area and it will be the subject of a forthcoming paper.

References

Boctor, F.F., (1976), "The Resource-constrained project scheduling problem", M. Sc. thesis, Cairo University.

Brand, J.D., Meyer, W.L., and Shaffer, L.R. (1964), "The resource scheduling problem in construction", Civil engineering studies report no. 5, Department of Civil Engineering, University of Illinois, Urbana, IL.

Burman, P.J. (1972), *Precedence Network for Project Planning and Control*, McGraw-Hill, New York.

Davis, E.W. (1969), "An exact algorithm for multiple constrained resource project scheduling problem", Ph.D. thesis, Yale University.

Davis, E.W. (1973), "Project scheduling under resource constraints: Historical review and categorization of procedures", *AIIE Transactions* 5/4, 297–313.

Davis, E.W. (1974), "Networks: Resource allocation", *Industrial Engineering* 6/4, 22–32.

Davis, E.W. (1975), "Project network summary measures constrained – Resource scheduling", *AIIE Transactions* 7/2, 132–142.

Davis, E.W., and Heidron, G.E. (1971), "An algorithm for optimal project scheduling under multiple resource constraints", *Management Science*, 17/12, B803–B816.

Doersch, R.H., and Patterson, J.H. (1977), "Scheduling a project to maximize its present value: A zero-one programming approach", *Management Science* 23/8, 882–889.

Elmaghraby, S.E. (1977), *Activity Networks: Project Planning and Control by Network Models*, Wiley, New York.

Fisher, M.L. (1970), "Optimal solution of resource constrained network scheduling problem", Technical report no. 56, Operations Research Center, Massachusetts Institute of Technology.

Herroelen, W.S. (1972), "Resource constrained project scheduling – The state of the art", *Operations Research Quarterly* 23/3, 261–275.

Huber, W.D. (1971), "Computational experience with zero-one programming for multiproject scheduling", M.B.A., Pensylvania State University,.

Kelly, J.E., Jr. (1963), "The critical path method: Resource planning and scheduling", Ch. 21 in *Industrial Scheduling*, J.F. Muth and G.L. Thompson (eds.), Prentice-Hall, Englewood Cliffs, NJ, 347–365.

Martino, R.L. (1968), *Resource Management*, McGraw-Hill, New York.

Moodies, C.L., and Mandville, D.E. (1966), "Project resource balancing by assembly line balancing techniques", *Industrial Engineering* 17/7, 377–383.

Patterson, J.H. (1976), "Project scheduling: The effect of problem structure on heuristic performance", *Naval Research Logistics Quarterly* 23/1, 95–123.

Patterson, J.H., and Huber, W.D. (1974), "A horizon-varying, zero-one approach to project scheduling", *Management Science* 20/6, 990–998.

Petrić, J. (1970), *Mrežno Planiranje I Upravljanje, Informator*, Zagreb.

Pritsker, A.B., Watters, L.J., and Wolfe, P.M. (1969), "Multi-project scheduling with limited resources: A zero-one programming approach", *Management Science* 16/1, 93–109.

Schrage, L. (1972), "Solving resource-constrained network problems by implicit enumeration, preemptive case", *Operations Research* 20/3, 668–677.

Slowinski, R., and Weglarz, J. (1978), "Solving the general project scheduling problem with multiple constrained resources by mathematical programming", Proceedings 8th IFIP Conference on Optimization Techniques.

Slowinski, R. (1980), "Two approaches to problems of resources allocation among project activities: A comparative study", *Journal of the Operations Research Society* 31/8, 711–723.

Slowinski, R. (1981), "Multiobjective network scheduling with efficient use of renewable and non-renewable resources", *European Journal of Operational Research* 7/3, 265–273.

Talbot, B., and Patterson, J.H. (1978), "An efficient integer programming algorithm with network cuts for solving resource – Constrained sequencing problems", *Management Science* 24/11, 1163–1174.

Talbot, F.B. (1982), "Resource constrained project scheduling with time–Resource trade offs: The non-preemptive case", *Management Science* 28/10, 1197–1210.

Weglarz, J. (1976), "Time-optimal control of resource allocation in a control of operations framework", *IEEE Transactions, Systems, Man and Cybernetics* SMC-6/11, 783–788.

Weglarz, J. (1980), "Control in resource allocation systems", *Foundation Control Engineering* 5/3, 159–180.

Weglarz, J. (1981), "Project scheduling with continuously divisible double constrained resources", *Management Science* 27/9, 1040–1053.

Weglarz, J., Blazewicz, J., Cellary, W., and Slowinski, R. (1977), "Algorithm 520 – An automatic revised simplex method for constrained resources network scheduling", *ACM Transactions Mathematical Software* 3/3, 295–300.

Wiest, J.D. (1963), "The scheduling of large projects with limited resources", Ph.D. dissertation, Carnegie Institute of Technology.

Wiest, J.D. (1967), "A heuristic model for scheduling large projects with limited resources", *Management Science* 13/6, B359–B377.

Wiest, J.D., and Levy, F.K. (1972), *A Management Guide to PERT/CPM*, Prentice-Hall, Englewood Cliffs, NJ.

Appendix 1

Problem	Source	Number of activities [a]	Critical path length	Resource limits 1	Resource limits 2	Resource limits 3	Optimum duration [b]
Small problems data							
1	Pritsker et al.	11	7	8	5	4	7
2	(1969) p.100			6	4	4	9
3				5	3	2	13
4	Davis (1969) p.200	30	6	12	10	12	8
5				11	12	12	8
6				12	11	10	8
7				12	12	10	8
8				12	11	11	8
9	Martino (1968) project 736	16	19	8			19
10				7			26
11				6			28
12	Davis et al. (1971) p.B-806	5	6	5	5	3	7
13				4	4	3	7
14	Hubber (1971) p.25	17	7	3	3	3	8
15				2	2	2	12
16	Hubber (1971) p.29	29	5	18			5
17				13			6
18				12			7
19				11			7
20	Brand et al. (1964) p.6	13	18	2	1	2	19
21	Davis (1974) p.28 [c]	11	18	6	7	6	20
22			13	6	7	6	15
23	Fisher (1970) p.13	7	5	1	2		6
24				1	1		6
25	Moodies et al. (1966) p.378	7	8	5			8
26				4			11
27	Hubber (1971) p.49	15	8	9			8
28	Wiest et al. (1972) p.99	10	6	11			6
29				10			7
30				9			8
31	Wiest et a. (1972) p.110	10	10	11			12
32				12			12
33	Burmann (1972) p.186	11	15	7			19
34				8			18
35				9			15
36	Martino (1968) p.8	12	9	2			9
Large problems data							
37	Petrić (1970) p.194	87	124	35	27	21	
38				31	24	28	
39				27	21	24	
40				22	19	20	
41	Petrić (1970) p.176	103	36	57	64	59	
42				51	57	52	
43				46	51	48	

Appendix 1 (continued)

Problem	Source	Number of activities [a]	Critical path length	Resource limits			Optimum duration [b]
				1	2	3	
44	Petrić (1970) p.261	111	129	30	33	30	
45	(modified duration)			27	29	26	
46				24	26	24	
47	Petrić (1970) p.287	46	37	23	25	23	
48				17	22	18	
49				16	19	15	
50				15	16	13	
51	Petrić (1970) p.309	82	104	75	69	96	
52				69	59	79	
53	Petrić (1970) p.315	44	69	43	54	48	
54				41	52	46	
55				36	44	40	
56				34	42	37	
57				32	40	35	
58	Petrić (1970) p.323	87	106	90	74	70	
59				84	68	66	
60				79	64	63	
61	Brand (1964) p.48	38	78	2	2	6	
62				2	1	5	
63	Boctor (1976) p.167	76	96	90	100	66	
64				80	100	56	
65				72	100	48	
66	Boctor (1976) p.168	108	73	50	100	100	

[a] Including dummy activities.

[b] As given in the literature.

[c] Problem 22 is identical to 21 except that all durations are decreased by one.

Appendix 2

Problem	Critical path length	Optimum duration	Parallel heuristics								Serial heuristics				
			MINSLK	RSM	MINLFT	RAN	GRD	SA	SRD	LA	$J-I$	$J-D$	$J-R$	$J-S$	$I-J$
Obtained durations: Small problems															
1	7	7	9	7	9	9	9	10	10	9	8	10	7	7	9
2	7	9	9	9	9	9	9	12	15	9	9	11	9	10	9
3	7	13	13	16	15	15	15	15	15	13	15	15	15	15	15
4	6	8	8	8	8	8	9	8	11	8	8	8	8	8	8
5	6	8	8	8	8	8	9	8	10	8	8	8	8	8	9
6	6	8	8	8	8	9	9	8	9	8	8	8	8	8	8
7	6	8	8	8	8	8	9	8	9	8	8	8	8	8	8
8	6	8	8	8	8	8	9	8	9	8	8	8	8	8	8
9	19	19	21	21	22	22	25	27	29	27	22	22	19	24	22
10	19	26	28	28	31	31	28	27	31	33	28	29	26	30	29
11	19	28	30	33	37	37	29	33	32	39	33	33	33	36	33
12	6	7	7	8	7	7	7	9	7	7	7	7	8	8	7
13	6	7	7	8	7	7	7	9	7	7	7	7	8	8	7
14	7	8	8	8	8	8	9	9	9	9	8	8	8	8	9
15	7	12	12	12	12	12	13	12	12	13	12	12	12	12	13
16	5	5	5	5	5	5	6	6	6	5	5	8	6	6	5
17	5	6	7	7	7	7	6	8	7	7	7	9	7	7	7
18	5	7	7	8	7	7	7	10	8	7	7	10	8	8	8
19	5	7	7	8	7	7	7	10	9	7	9	10	8	8	8
20	18	19	22	19	23	23	21	23	23	21	20	22	20	20	22
21	18	20	25	26	25	25	26	23	23	26	25	25	25	25	25
22	13	15	17	19	17	18	19	16	16	19	18	18	18	18	18
23	5	6	6	6	6	7	6	8	8	6	7	7	7	7	7
24	5	6	6	6	6	7	6	8	8	6	7	7	7	7	7
25	8	8	8	9	8	9	8	9	14	8	9	9	9	9	9
26	8	11	13	13	11	13	11	13	14	13	13	13	13	13	13
27	8	8	9	9	9	10	8	10	11	8	10	10	10	10	11
28	6	6	7	7	7	7	7	8	7	9	7	7	7	7	7
29	6	7	7	7	7	7	7	8	7	9	7	7	7	7	7
30	6	8	9	9	8	9	9	9	9	10	9	9	9	9	8
31	10	12	14	14	14	14	14	12	12	14	14	14	14	14	14
32	10	12	14	14	12	12	12	12	12	14	12	12	12	12	12
33	15	19	19	19	19	20	19	20	21	20	20	20	20	20	20
34	15	18	18	18	19	19	18	19	19	19	20	20	20	20	20
35	15	15	15	15	15	15	15	16	16	15	15	15	15	15	15
36	9	9	9	9	10	10	9	10	10	9	10	11	9	11	10

Obtained durations: Large projects

37	124	129	127	127	127	139	133	135	141	129	133	127	132	131
38	124	135	135	137	135	142	143	143	145	134	136	140	136	136
39	124	141	138	141	142	150	144	147	149	139	143	139	142	141
40	124	160	156	152	158	177	155	171	170	158	155	156	160	157
41	36	38	41	41	40	40	41	42	45	47	47	47	47	42
42	36	42	44	47	46	49	48	49	50	50	50	50	50	46
43	36	47	47	48	49	50	50	51	55	55	55	55	55	48
44	129	132	135	134	135	134	137	143	139	135	135	135	135	139
45	129	137	137	136	138	137	144	141	143	145	145	145	145	142
46	129	140	144	146	146	145	142	148	142	167	167	167	167	161
47	37	37	37	38	38	41	37	40	39	39	39	39	39	40
48	37	46	46	42	42	49	46	46	48	45	45	45	45	45
49	37	61	63	65	64	63	64	67	61	61	61	61	61	63
50	37	68	74	70	70	67	71	76	72	73	73	73	73	69
51	104	104	104	109	109	109	109	109	104	119	119	119	119	119
52	104	109	109	114	114	109	111	111	109	124	126	124	126	124
53	69	70	70	75	75	79	79	79	73	80	80	80	80	80
54	69	80	80	82	82	85	82	80	80	82	82	82	82	82
55	69	80	81	82	82	86	82	82	80	82	82	82	82	82
56	69	82	81	88	88	100	85	88	90	89	89	82	89	82
57	69	82	92	84	84	87	87	92	90	93	90	87	90	93
58	106	106	106	106	106	106	111	111	106	106	106	106	106	106
59	106	106	106	108	106	108	121	116	106	108	108	106	107	106
60	106	106	106	106	108	108	127	124	106	108	109	108	108	108
61	78	84	82	84	84	84	85	85	84	82	85	82	82	82
62	78	91	91	91	91	91	93	93	91	91	95	91	91	91
63	96	97	100	97	99	99	99	100	99	98	98	98	98	99
64	96	98	100	98	101	101	102	101	100	101	101	101	101	102
65	96	99	101	99	104	104	102	105	104	102	102	102	102	106
66	73	82	82	81	81	83	81	82	82	81	81	81	81	82