Hamdjatou Kane
Gilbert Nkubili
Département des sciences administratives
Université du Quebec en Outaouais in Canada

Barthelemy Ateme-Nguema
Département des sciences de la gestion
Université du Quebec en Abitibi-Témiscamingue in Canada

A WAY TO DEAL WITH THE PROJECT CRASHING PROBLEM

Introduction

Completing projects as quickly as possible remains a constant preoccupation of all managers (Liberatore, Pollack-Johnson 2006). The literature is replete with affirmations that the businesses able to develop new products in the shortest time enjoy a substantial competitive edge (Swink, 2003).

Practitioners, academics and other professionals continue to search for ways, means and tools for determining a priori how much a project can be accelerated, based on inherent constraints such as budget and resources. Contemporary managerial emphasis on gaining competitive advantages obliges project managers to scope the costs of accelerating each activity of projects underway or upcoming. In the context of resource criticality, the cost of acceleration depends to a large degree on the type of resources assigned, their availability, their quantity and so on.

In this article, we propose a new approach to solving the problem of reaching an optimal compromise between duration and resources for the acceleration of projects in a context of resource criticality, in which the substitution of the resources is considered.

1. Review of the literature

In conventional project management, the mandate is carried out within predetermined constraints such as specifications, deadlines and budget. Even though the anticipated length of a project in planning is usually longer than the critical path (Boctor 2005), projects seldom follow the classical project management scheme or meet deadlines (Gerk, Qassim 2008). In order to address this deficiency, project execution time can be accelerated to compensate for potential delays. By acceleration we mean finishing a project sooner than originally planned. In order to accelerate a project, we must have all of the information relevant to its constituent activities, including the types and quantities of the resources involved, unit costs, durations, priority relationships and so on.

The usual method of accelerating a given activity within a project is to assign to it more resources or resources with expertise and/or skills greater than those initially at the disposal of the project manager. This of course involves increased costs. However, accelerating some project activities can bring a reduction of indirect costs (Dodin, Elimam 2008; Evensmo and Karlsen, 2008) in the form of salary, amortisement of equipment and infrastructure, and so on.

Some authors have proposed accelerating projects by superimposing activities (Roemer, Ahmadi 2004). This approach is used essentially in the development of new products and services. The superimposition of activities consists of carrying out in parallel (partially or wholly) activities that were organized sequentially in the project plan. Another acceleration technique involves substituting certain activities with one and/or several other activities. There is thus the possibility of accelerating a project by applying the conventional approach, superimposing activities and substituting certain activities (Gerk, Qassim 2008).

Several approaches have been described in the literature for solving the time/cost trade-off problem. However, few of these take into consideration project particularities, that is, the peculiarities of the project resources. We may cite the cut search approach proposed by Kane, Azondekon (2008) and the linear programming model proposed by Alban (2008).

However, most of these approaches are deficient and difficult to apply to large complex projects (i.e. 100 activities or more). Calculation time in particular increases considerably. There are new methods better suited to solving the acceleration problem in the case of large projects for which more than two resources are available. These require minimal time for calculation.

We recommend a method based on application of the tabu algorithm and describe in the following section the principal steps of our approach.

2. Methodology

The logical procedure usually used to solve the duration/cost trade-off problem is the simultaneous mathematical equation approach, one equation representing time and the other representing project cost. The time taken to complete the activities as a whole depends on individual activity duration and on the priority relationships between the tasks. Cost depends on the resources assigned and of course on the allowed duration of each task. This approach generally seeks to determine exact activity completion times and total project time. These methods differ primarily in terms of the optimization techniques used to choose the activities to be accelerated and calculate by how much to accelerate them. Among these, we may cite the CPM/PERT method, the linear exact and non-linear programming techniques (Alban, 2008), the cut search approach (Kane, Azondekon 2008), and algorithmic and heuristic techniques (Bolduc, Laporte et al. 2010).

However, methods based on exact mathematical programming run into difficulty as project scale (number of activities or tasks) increases and the problem becomes of the so-called NP-hard type. The solving of this type of problem requires complex optimization techniques and the tabu search is the technique that we feel provides the best potential solution for project acceleration purposes. This algorithm technique also minimizes problem-solving time.

2.1. Mathematical model

The aim of this mathematical model is to calculated the total cost of the project. We describe below the parameters, the variables and the "objective" function of the model.

Let t_i be the time (from the beginning of the project) at which activity i is to begin according to the project plan and let x_i be the number of units of time by which activity i is accelerated.

The remaining parameters are as follows:
d_i – the normally expected duration of activity i
c_i – the unit cost of accelerating activity i
u_i – the maximum number of time units by which activity i can be accelerated
n – the number of activities (1 being the first activity and n being the last)
T – the normally expected duration of the critical path of the un-accelerated project
T_m – the calculated project duration after the m^{th} iteration, $m = 1, 2, 3, \ldots, M$

$T_m = T_a$ – the optimal duration attainable by accelerating the project
$P(j)$ – the set of activities immediately preceding activity j
C_{ki} – the normal cost per unit of time of resource k assigned to activity i
a_{ki} – the cost per time unit of obtaining via resource k a one-time-unit acceleration of activity i
K_{ki} –the total number of k resources assigned to activity i
N_a – the total number of activities that can be accelerated
$\sum_{ki=1}^{K_{ki}} c_{ki} d_i$ – the normally expected cost of completing activity i
$\sum_{ki=1}^{K_{ki}} a_{ki} x_i$ – the additional cost of accelerating activity i by x_i units of time
B – the additional budget available for accelerating the project
C_{NET} – the net cost of accelerating activity i by x_i units of time
C_n – the normally expected total cost of completing the project
$C_{\max} = C_n + B$ – the total cost not to be exceeded due to project acceleration
C_{aTOT} – the net total additional cost of accelerating the project
C_{aTOTm} – the net additional cost of the project at the m^{th} iteration

Objective function:

To minimize $C_{aTOT} = \alpha_i \sum_{i=1}^{N_a} x_i \sum_{ki}^{K_{ki}} (a_{ki} - c_{ki})$ (1)

Subject to:

$\alpha_i = 1$ if i is selected for acceleration;

0 if i is not selected for acceleration (2)

$$t_j \geq t_i - x_i \qquad \forall (i, j) \in P(j) \tag{3}$$

$t_n \leq T$ is the instant of the end of the last activity of the project (4)

$$T \leq T_{m+1} \leq T_m \quad \text{avec} \quad T_M = T_a \tag{5}$$

$$0 \leq x_i \leq u_i \tag{6}$$

$$x_i \leq u_i \tag{7}$$

$$C_{aTOT} \leq B \tag{8}$$

With:

$t_1 = 0$ is the start of the first activity of the project

$t_i \geq 0 \quad \forall i$

$$C_{NET} = \sum_{ki=1}^{K_{ki}} c_{ki}\left(d_i - x_i\right) + \sum_{ki}^{K_{ki}} a_{ki}x_i = \sum_{ki}^{K_{ki}} c_{ki}d_i - \sum_{ki}^{K_{ki}} c_{ki}x_i + \sum_{ki}^{K_{ki}} a_{ki}x_i = \sum_{ki}^{K_{ki}} c_{ki}d_i + x_i \sum_{ki}^{K_{ki}} \left(a_{ki} - c_{ki}\right) \tag{9}$$

$$C_n = \sum_{i=1}^{N} \sum_{ki=1}^{K_{ki}} c_{ki}d_i \tag{10}$$

The problem amounts to minimizing the total project duration while remaining within the limits of additional budget *B*. T_m (expected project duration based on the critical path) is calculated using the CPM method. The objective function for minimizing T_m according to $T \leq T_{m+1} \leq T_m\, T$ consists of choosing an activity on the critical path and accelerating it by one unit of time.

Having established the method of calculating project duration and cost, our goal is to find a new approach to optimizing the solution to the problem of finding the best trade-off between project completion time and cost, with the aim of obtaining the greatest decrease in time at the lowest cost. It is at this stage that we use the tabu method to determine the project-accelerating option that costs the least. In the following section, we present the approach based on the tabu algorithm.

2.2 The tabu algorithm

The "tabu" algorithm is a local-search meta-heuristic that explores the neighbourhood beyond the optimum solution obtained (Xu et al. 2009). This search method uses an iterative process to shift from the current solution towards a neighbouring solution that achieves a superior goal. In order to avoid futile cycles, that is, exploration of solutions similar to those previously examined, the search generates a "tabu" list of shifts and solutions explored in previous iterations (Liu et al. 2010). In order to improve the efficiency of the iterative process, the tabu algorithm maintains a follow-up of the local information as well as of the search process itself (Bolduc et al. 2010). The other principles of search with tabu, namely aspiration, intensification and diversification, are treated in detail in works published by Glover (1989; 1990) and by Glover and Laguna (1997).

In each iteration, our tabu algorithm first explores the entire solutions space (the project plan as a whole) and thus defines the zone in which it will subsequently intensify the search for the activity to be accelerated. The activity configuration scheme, duration and the total cost of the project are then updated, based on acceleration of the activity thus identified.

The process is stopped when the conditions regarding project duration relative to predefined budgetary constraints are met or when it has been determined that no activity within the predefined zone of search can be accelerated to obtain a desirable result (Figure l).

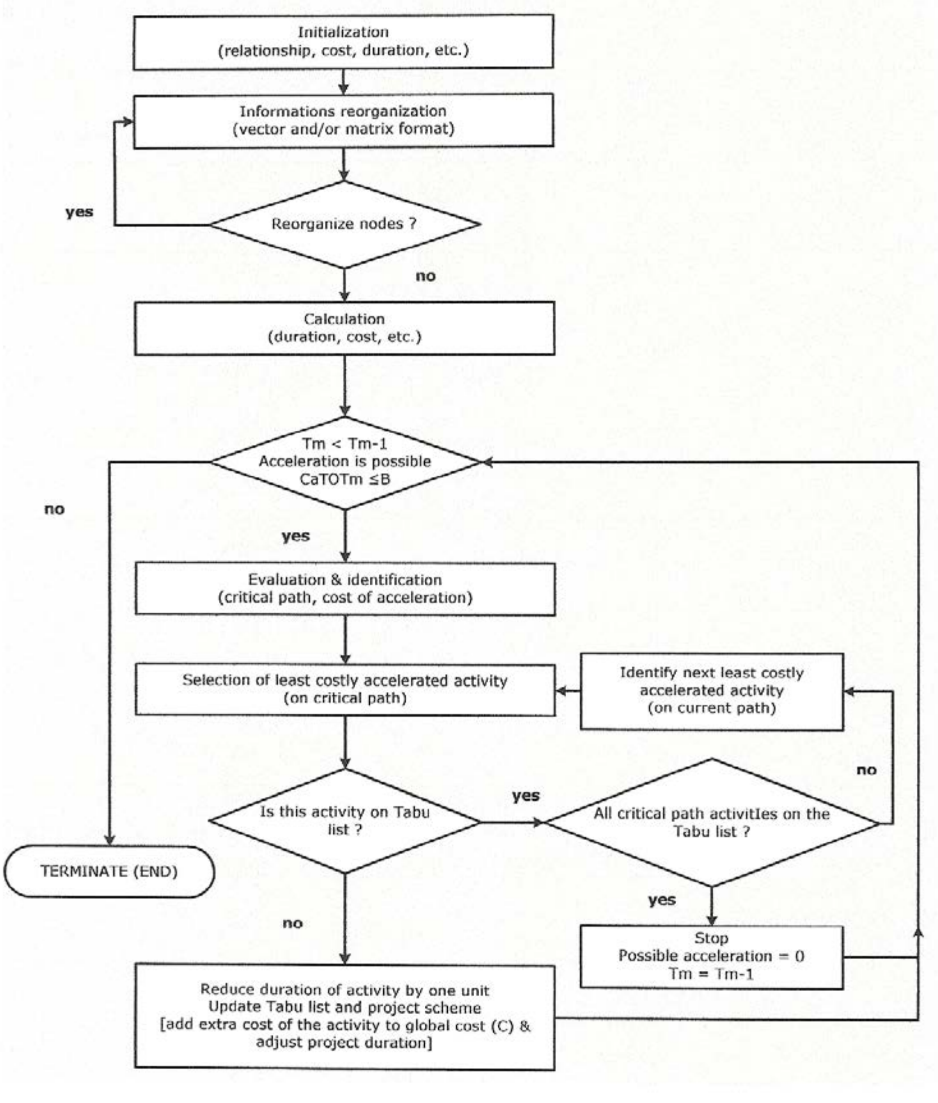

Figure 1. The project acceleration algorithm with Tabu

3. Applications

In order to validate the proposed project acceleration algorithm, we performed tests on real projects involving numerous activities (over 100). The project includes 172 activities requiring four different types of resource. Figures 2a and 2b show the project network, while Table 1 in appendix provides the time and resource-associated costs for each activity.

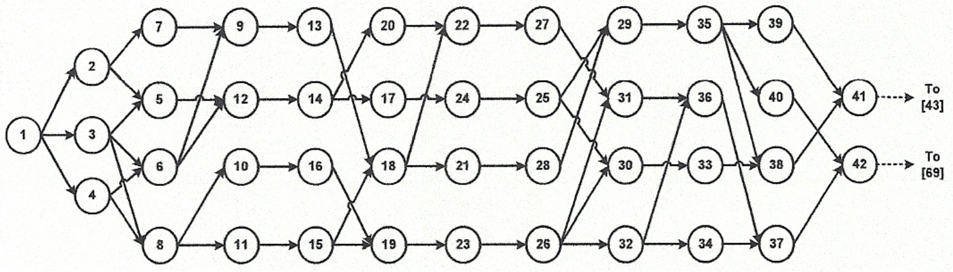

Figure 2a. The project network

Source: Doerner et al. 2008.

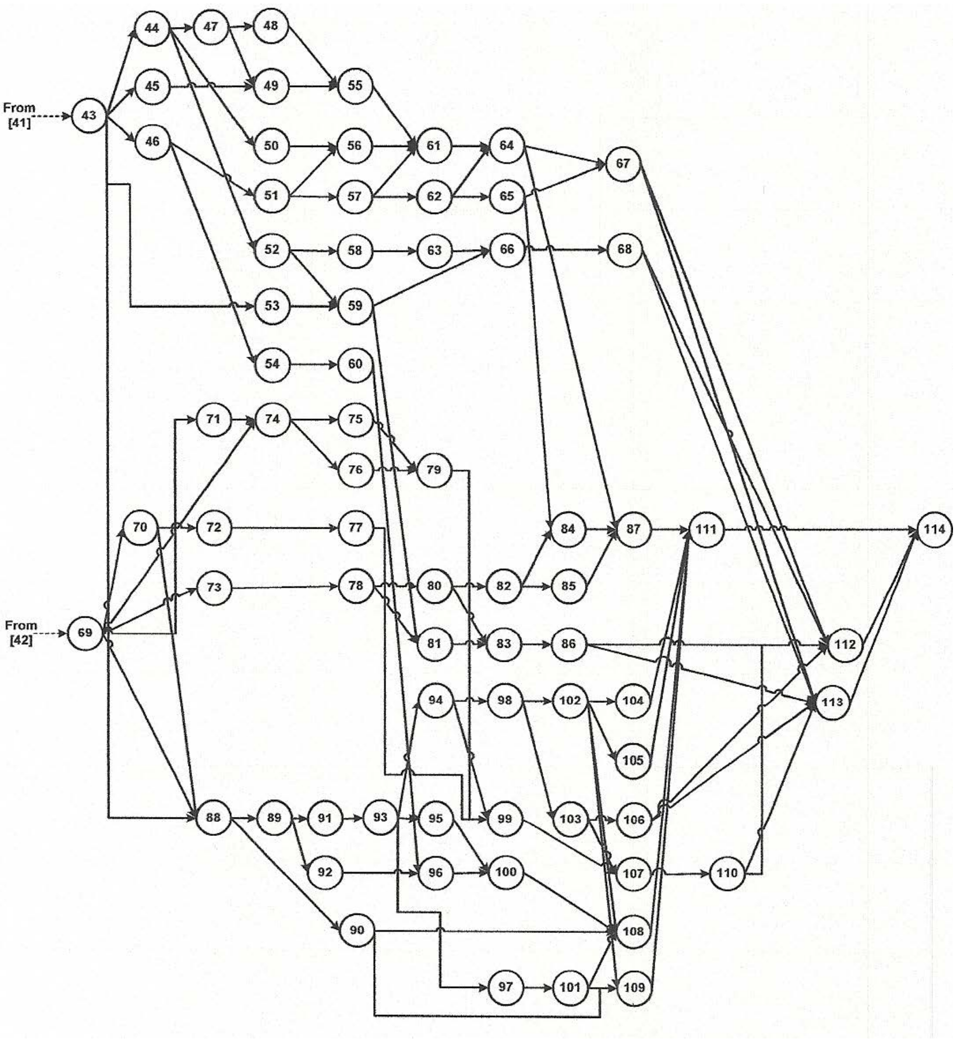

Figure 2b. The project network

Source: Doerner et al. 2008.

4. Results

We used MATLAB 6.0 to implement our algorithm. The results obtained show that the project can be accelerated with minimal increases in cost. Furthermore, the calculation time was relatively short. The expected project duration before acceleration was 83 weeks for a total cost of $1,483,600. We defined the normal unit cost of each resource arbitrarily and set the unit cost of acceleration 50% higher (see Table 1 in appendix).

If we also suppose that any activity can be accelerated without limits and that we have at our disposal a budgetary increase of up to 20% with which to accelerate the project, we can obtain a reduction in project duration from 83 to 47 weeks at an additional cost of $293,450. This is a 43.3% reduction of project time for a cost increase of 19.77%. The algorithm achieved this result by carrying out 113 iterations, which required about 3 seconds of calculation time.

These results indicate that the algorithm is effective and could be used on a daily basis by professionals to accelerate large-scale projects involving relatively large numbers of tasks.

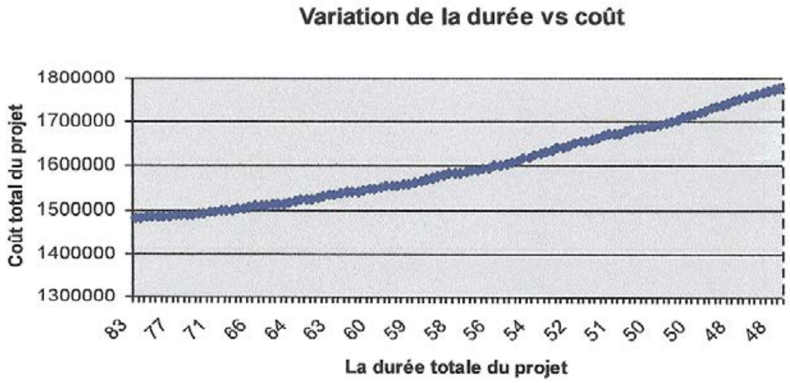

Figure 3. Cost variation VERSUS project duration

Conclusion

In this study, we have presented the results obtained using a new tool developed to solve the problem of finding the optimal trade-off between project duration and cost in the context of resource criticality. The tool we have developed applies principles of tabu search to optimize the process of identifying project activities to be accelerated.

Using our project acceleration algorithm on projects previously treated using other approaches, we demonstrated that the tool identifies the same solution or in some cases a better solution and with a shorter calculation time.

This work thus proposes a new avenue to explore with more in-depth studies for the improvement of project management.

Appendix

Table 1

Duration, acceleration and associated resource costs of activities (corresponding to arrows in the CPM/PERT diagram) for the project

Act.	Path	Pqte	d_i	u_i	C_{k1}	C_{k2}	C_{k3}	C_{k4}	a_{k1}	a_{k2}	a_{k3}	a_{k4}
1	(1,2)	---	2	1	700	700	700	700	1050	1050	1050	1050
2	(1,3)	---	3	2	900	900	900	900	1350	1350	1350	1350
3	(1,4)	---	3	2	600	600	600	600	900	900	900	900
4	(2,5)	1	2	l	600	600	600	600	900	900	900	900
5	(2,7)	1	2	1	100	100	100	100	150	150	150	150
6	(3,5)	2	2	1	700	700	700	700	1050	1050	1050	1050
7	(3,6)	2	4	3	400	400	400	400	600	600	600	600
8	(3,8)	2	5	4	100	100	100	100	150	150	150	150
9	(4,6)	3	3	2	1600	1600	1600	1600	2400	2400	2400	2400
10	(4,8)	3	2	l	3000	3000	3000	3000	4500	4500	4500	4500
11	(5,12)	4;6	6	5	700	700	700	700	1050	1050	1050	1050
12	(6,9)	7;9	3	2	900	900	900	900	1350	1350	1350	1350
13	(6,12)	7;9	2	1	600	600	600	600	900	900	900	900
14	(7,9)	5	4	3	600	600	600	600	900	900	900	900
15	(8,9)	8;10	2	1	100	100	100	100	150	150	150	150
16	(8,10)	8;10	5	4	700	700	700	700	1050	1050	1050	1050
17	(8,11)	8;10	6	5	400	400	400	400	600	600	600	600
18	(9,13)	11;14;15	4	3	100	100	100	100	150	150	150	150
19	(10,16)	16	3	2	1600	1600	1600	1600	2400	2400	2400	2400
20	(11,15)	17	4	3	3000	3000	3000	3000	4500	4500	4500	4500
21	(12,14)	11;13	2	1	700	700	700	700	1050	1050	1050	1050
22	(13,18)	18	2	1	900	900	900	900	1350	1350	1350	1350
23	(14,17)	21	3	2	600	600	600	600	900	900	900	900
24	(14,20)	21	3	2	600	600	600	600	900	900	900	900
25	(15,18)	19	2	l	100	100	100	100	150	150	150	150
26	(15,19)	20	3	2	700	700	700	700	1050	1050	1050	1050
27	(16,19)	19	2	1	400	400	400	400	600	600	600	600
28	(17,24)	23	6	5	100	100	100	100	150	150	150	150
29	(18,21)	25	5	4	1600	1600	1600	1600	2400	2400	2400	2400
30	(18,22)	22;25	2	1	3000	3000	3000	3000	4500	4500	4500	4500
31	(19,23)	26;27	3	2	700	700	700	700	1050	1050	1050	1050
32	(20,22)	24	l	0	900	900	900	900	1350	1350	1350	1350
33	(21,28)	29	3	2	600	600	600	600	900	900	900	900
34	(22,27)	30;32	3	2	600	600	600	600	900	900	900	900
35	(23,26)	31	4	3	100	100	100	100	150	150	150	150
36	(24,25)	28	3	2	700	700	700	700	1050	1050	1050	1050
37	(25,29	36	2	1	400	400	400	400	600	600	600	600
38	(25,30)	36	5	4	100	100	100	100	150	150	150	150
39	(26,30)	35	2	1	1600	1600	1600	1600	2400	2400	2400	2400
40	(26,31)	35	2	1	3000	3000	3000	3000	6000	4500	4500	4500
41	(26,32)	35	2	1	3000	3000	3000	3000	4500	4500	4500	4500
42	(27,31)	35	4	3	700	700	700	700	1050	1050	1050	1050
43	(28,29)	34	5	4	900	900	900	900	1350	1350	1350	1350
44	(29,35)	33	3	2	600	600	600	600	900	900	900	900
45	(30,33)	37;43	4	3	600	600	600	600	900	900	900	900
46	(31,36)	38;39	2	1	100	100	100	100	150	150	150	150
47	(32,34)	40;42	3	2	700	700	700	700	1050	1050	1050	1050
48	(32,36)	41	3	2	400	400	400	400	600	600	600	600
49	(33,38)	41	5	4	100	100	100	100	150	150	150	150
50	(34,37)	45	2	1	1600	1600	1600	1600	2400	2400	2400	2400

Table 1(continued)

Act.	Path	Pqte	d_i	u_i	C_{k1}	C_{k2}	C_{k3}	C_{k4}	a_{k1}	a_{k2}	a_{k3}	a_{k4}
51	(35,38)	47	4	3	3000	3000	3000	3000	4500	4500	4500	4500
52	(35,39)	44	2	1	700	700	700	700	1050	1050	1050	1050
53	(35,40)	44	2	1	200	200	200	200	300	300	300	300
54	(36,37)	44	3	2	900	900	900	900	1350	1350	1350	1350
55	(37,42)	46;48	2	1	600	600	600	600	900	900	900	900
56	(38,41)	50;54	5	4	600	600	600	600	900	900	900	900
57	(39,41)	49;51	2	1	100	100	100	100	150	150	150	150
58	(40,42)	52	3	2	700	700	700	700	1050	1050	1050	1050
59	(41,43)	53	2	1	700	700	700	700	1050	1050	1050	1050
60	(42,69)	57	2	1	400	400	400	400	600	600	600	600
61	(43,44)	55;58	4	3	200	200	200	200	300	300	300	300
62	(43,45)	59	2	1	700	700	700	700	1050	1050	1050	1050
63	(43,46)	59	3	2	900	900	900	900	1350	1350	1350	1350
64	(43,53)	59	1	0	600	600	600	600	900	900	900	900
65	(43,88)	59	3	2	600	600	600	600	900	900	900	900
66	(44,47)	59	2	1	100	100	100	100	150	150	150	150
67	(44,500	61	4	3	700	700	700	700	1050	1050	1050	1050
68	(44,52)	61	3	2	400	400	400	400	600	600	600	600
69	(45,49)	61	4	3	100	100	100	100	150	150	150	150
70	(46,51)	62	2	1	1600	1600	1600	1600	2400	2400	2400	2400
71	(46,54)	63	2	1	3000	3000	3000	3000	4500	4500	4500	4500
72	(47,48)	63	3	2	700	700	700	700	1050	1050	1050	1050
73	(47,49)	66	2	1	200	200	200	200	300	300	300	300
74	(48,55)	66	1	0	900	900	900	900	1350	1350	1350	1350
75	(49,55)	72	3	2	600	600	600	600	900	900	900	900
76	(50,56)	69;73	4	3	600	600	600	600	900	900	900	900
77	(51,56)	67	2	1	100	100	100	100	150	150	150	150
78	(51,57)	70	4	3	700	700	700	700	1050	1050	1050	1050
79	(52,58)	70	3	2	700	700	700	700	1050	1050	1050	1050
80	(52,59)	68	5	4	400	400	400	400	600	600	600	600
81	(53,59)	68	4	3	200	200	200	200	300	300	300	300
82	(54,60)	64	2	1	1600	1600	1600	1600	2400	2400	2400	2400
83	(55,61)	71	1	0	3000	3000	3000	3000	4500	4500	4500	4500
84	(56,61)	74;75	2	1	700	700	700	700	1050	1050	1050	1050
85	(57,61)	76;77	3	2	900	900	900	900	1350	1350	1350	1350
86	(57,62)	78	2	1	600	600	600	600	900	900	900	900
87	(58,63)	78	5	4	600	600	600	600	900	900	900	900
88	(59,66)	79	2	1	100	100	100	100	150	150	150	150
89	(59,81)	80;81	4	3	700	700	700	700	1050	1050	1050	1050
90	(60,96)	80;81	3	2	600	600	600	600	900	900	900	900
91	(61,64)	82	4	3	400	400	400	400	600	600	600	600
92	(62,64)	83;84;85	2	1	100	100	100	100	150	150	150	150
93	(62,65)	86	3	2	1600	1600	1600	1600	2400	2400	2400	2400
94	(63,66)	86	2	1	3000	3000	3000	3000	4500	4500	4500	4500
95	(64,67)	87	2	1	700	700	700	700	1050	1050	1050	1050
96	(64,87)	91;92	3	2	900	900	900	900	1350	1350	1350	1350
97	(65,67)	91;92	2	1	700	700	700	700	1050	1050	1050	1050
98	(65,84)	93	3	2	600	600	600	600	900	900	900	900
99	(66,68)	93	4	3	100	100	100	100	150	150	150	150
100	(67,112)	94	5	4	600	600	600	600	900	900	900	900
101	(67,113)	95;97	2	1	100	100	100	100	150	150	150	150
102	(68,112)	95;97	4	3	700	700	700	700	1050	1050	1050	1050
103	(68,113)	99	3	2	400	400	400	400	600	600	600	600
104	(69,70)	99	4	3	100	100	100	100	150	150	150	150
105	(69,71)	60	2	1	1600	1600	1600	1600	2400	2400	2400	2400
106	(69,73)	60	2	1	3000	3000	3000	3000	4500	4500	4500	4500
107	(69,74)	60	1	0	700	700	700	700	1050	1050	1050	1050
108	(69,88)	60	3	2	900	900	900	900	1350	1350	1350	1350
109	(70,72)	60	4	3	100	100	100	100	150	150	150	150

Table 1(continued)

Act.	Path	Pqte	d_i	u_i	C_{k1}	C_{k2}	C_{k3}	C_{k4}	a_{k1}	a_{k2}	a_{k3}	a_{k4}
110	(70,88)	104	2	1	600	600	600	600	900	900	900	900
111	(71,74)	104	3	2	600	600	600	600	900	900	900	900
112	(72,77)	105	2	1	100	100	100	100	150	150	150	150
113	(73,78)	109	2	1	700	700	700	700	1050	1050	1050	1050
114	(74,75)	106	4	3	400	400	400	400	600	600	600	600
115	(74,76)	107;111	2	1	100	100	100	100	150	150	150	150
116	(75,790	107;111	3	2	700	700	700	700	1050	1050	1050	1050
117	(76,79)	114	2	1	400	400	400	400	600	600	600	600
118	(77,99)	115	2	1	100	100	100	100	150	150	150	150
119	(78,80)	112	6	5	1600	1600	1600	1600	2400	2400	2400	2400
120	(78,81)	113	3	2	3000	3000	3000	3000	4500	4500	4500	4500
121	(79,99)	113	5	4	700	700	700	700	1050	1050	1050	1050
122	(80,82)	115;117	2	1	900	900	900	900	1350	1350	1350	1350
123	(80,83)	119	4	3	600	600	600	600	900	900	900	900
124	(81,83)	119	2	1	600	600	600	600	900	900	900	900
125	(82,84)	89;110	2	1	100	100	100	100	150	150	150	150
126	982,85)	122	3	2	700	700	700	700	1050	1050	1050	1050
127	(83,86)	122	2	1	400	400	400	400	600	600	600	600
128	(84,87)	123;124	4	3	100	100	100	100	I50	150	150	150
129	(85,87)	125	2	1	1600	1600	1600	1600	2400	2400	2400	2400
130	(86,112)	126	2	1	3000	3000	3000	3000	4500	4500	4500	4500
131	(86,113)	127	3	2	700	700	700	700	1050	1050	1050	1050
132	(87,111)	127	2	1	200	200	200	200	300	300	300	300
133	(88,89)	128;129	3	2	900	900	900	900	1350	1350	1350	1350
134	(88,90)	65;108;110	2	1	600	600	600	600	900	900	900	900
135	(89,91)	65;108;110	5	4	600	600	600	600	900	900	900	900
136	(89,92)	133	3	2	100	100	100	100	150	150	150	150
137	(90,108)	133	3	2	700	700	700	700	1050	1050	1050	1050
138	(90,109)	134	2	1	700	700	700	700	1050	1050	1050	1050
139	(91,93)	134	4	3	400	400	400	400	600	600	600	600
140	(92,96)	135	3	2	200	200	200	200	300	300	300	300
141	(93,94)	136	4	3	700	700	700	700	1050	1050	1050	1050
142	(93,95)	139	2	1	900	900	900	900	1350	1350	1350	1350
143	(93,97)	139	6	5	600	600	600	600	900	900	900	900
144	(94,98)	139	2	1	700	700	700	700	1050	1050	1050	1050
145	(94,99)	141	3	2	900	900	900	900	1350	1350	1350	1350
146	(95,100)	141	4	3	900	900	900	900	1350	1350	1350	1350
147	(96,100)	142	1	0	600	600	600	600	900	900	900	900
148	(97,101)	90;140	2	1	600	600	600	600	900	900	900	900
149	(98,102)	143	3	2	100	100	100	100	150	150	150	150
150	(98,103)	144	2	1	700	700	700	700	1050	1050	1050	1050
151	(99,107)	144	3	2	700	700	700	700	1050	1050	1050	1050
152	(100,108)	118;121;145	2	1	400	400	400	400	600	600	600	600
153	(101,108)	148	4	3	700	700	700	700	1050	1050	1050	1050
154	(101,109)	148	2	1	900	900	900	900	1350	1350	1350	1350
155	(102,104)	149	3	2	h00	h00	600	600	900	900	900	900
156	(102,105)	149	2	1	600	600	600	600	900	900	900	900
157	(102,108)	149	2	1	100	100	100	100	150	150	I50	150
158	(102,109)	149	5	4	700	700	700	700	1050	1050	1050	1050
159	(103,106)	150	3	2	400	400	400	400	600	600	600	600
160	(103,107)	150	5	4	100	100	100	100	150	150	150	150
161	(104,111)	155	2	1	1600	1600	1600	1600	2400	2400	2400	2400
162	(105,111)	156	4	3	3000	3000	3000	3000	4500	4500	4500	4500
163	(106,112)	159	3	2	700	700	700	700	1050	1050	1050	1050
164	(106,113)	159	4	3	200	200	200	200	300	300	300	300
165	(107,110)	151;160	2	1	900	900	900	900	1350	1350	1350	1350
166	(108,111)	137;152;153;157	2	1	600	600	600	600	900	900	900	900
167	(109,111)	138;154;158	4	3	600	600	600	600	900	900	900	900
168	(110,112)	165	3	2	100	100	100	100	150	150	150	150
169	(110,113)	165	2	1	700	700	700	700	1050	1050	1050	1050

Table 1(continued)

Act.	Path	Pqte	d_i	u_i	C_{k1}	C_{k2}	C_{k3}	C_{k4}	a_{k1}	a_{k2}	a_{k3}	a_{k4}
170	(111,114)	132;161;162;166;167	3	2	700	700	700	700	1050	1050	1050	1050
171	(112,1140)	100;102;130;163;168	4	3	400	400	400	400	600	600	600	600
172	(113,114)	101;103;131;164;169	5	4	200	200	200	200	300	300	300	300

References

Alban T. (2008), *Accélération des projets et allocation des ressources: des enjeux de la gestion multi-projet*. Mémoire de Maîtrise, Université du Québec en Outaouais.

Boctor Fayez F. (2005), *Le problème d'allocation et de nivellement des ressources*. ASAC 2005 Conference, Toronto (Ontario), Canada.

Bolduc M.C., Laporte G., Renaud J., Boctor F.F. (2010), *A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars*. "European Journal of Operational Research", 202 (1): 122-130.

Dodin B., Elimam A.A. (2008), *Integration of equipment planning and project scheduling*. "European Journal of Operational Research", 184(3): 962-980.

Evensmo J., Karlsen J. T. (2008), *Looking for the Source – Where Do Crash Costs Come From?*. "Cost Engineering", *50(7).*

Gerk J.E.V., Qassim R.Y. (2008), *Project Acceleration via Activity Crashing, Overlapping, and Substitution*. "IEEE Transactions On Engineering Management", 55(4): 590-601.

Glover F.W. (1989), *Tabu Search – Part I*. "ORSA Journal on Computing", INFORMS: Institute for Operations Research 1: 190.

Glover F.W. (1990), *Tabu Search – Part II*. "ORSA Journal on Computing", INFORMS: Institute for Operations Research 2: 4.

Glover F.W. Laguna M. (1997), *Tabu Search*. "Kluwer Academic Publishers", Boston.

Kane H., Azondekon S.H. (2008), *Un algorithme pour la résolution d'un problèrne d'accélération de projet avec des ressources multiples*. 7e Conférence Intemationale de Modélisation et Simulation – MOSIM'08 – du 31 mars au 2 avril 2008-Paris-France, Modélisation, Optimisation et Simulation des systemes: Communications, Coopération, et Coordination.

Liberatore M.J., Pollack-Johnson B. (2006), *Extending project time-cost analysis by removing precedence relationships and activitystreaming*. "International Journal of Project Management", 24(6): 529-535.

Liu J., Cheng H., Shi X., Xu J. (2010), *A Tabu Search Algorithm for Fast Restoration of Large Area Breakdown in Distribution Systems*. "Energy and Power Engineering", 1-5.

Liu S. (2003), *Fuzzy activity Times In critical Path and Project Crashing problems*. “Cybernetics and Systems: An International Journal”, 34: 161-172.

Rakotomalala H.L. (2002), *Le compromis durée/ressource en gestion de projet dans un contexte de multiplicité des ressources : Analyse et impact sur la performance de projet*. Mémoire de Maîtrise, Université du Québec en Outaouais.

Roemer T.A., Ahmadi R. (2004), *Concurrent Crashing and Overlapping in Product Development*. “Operations Research”, INFORMS: Institute for Operations Research 52: 606-622.

Swink M. (2003), *Completing projects on-time: how project acceleration affects new product development*. “Journal of Engineering and Technology Management”, 20(4): 319-344.

Xu K., Feng Z., Jun K. (2009), *A Tabu Search algorithm for scheduling jobs with controllable processing times on a single machine to meet due-dates*. “Computers & Operations Research”, *37:* 1924-1938.

Wakas S., June L.W. (2009), *Cost Reduction for the Project Completion in Shortest Possible Duration by Stretching Noncritical Activities*. “Australian Journal of Basic and Applied Sciences”, 3(4): 4526-4533.