Journal of the
Operations Research
Society of Japan

VOLUME 11 *December 1968* NUMBER 2

CRITICAL PATH ANALYSIS FOR A PROJECT WITH DIVISIBLE ACTIVITIES

TAKASHI KOBAYASHI

Faculty of Engineering, University of Tokyo, Tokyo

(Received February 26, 1968)

Abstract

This paper presents an algorithm for determining the allocation that minimizes the critical path length in the project network with divisible activities. It is a scheme such that the dual problem to the original is solved by the simplex method using multipliers together with the usual critical path algorithm which is applied for finding the variable with the most negative relative cost factor. It is more efficient when the network has much more paths than arcs corresponding to the locations where divisible activities can be allocated.

1. Introduction

An activity in a project is called divisible if it can be divided up and done at some different locations in the project network. Jewell [3] and the author [4] have shown algorithms for determining the allocation that minimizes the critical path length in the project with a single divisible activity. This paper presents an algorithm in the project with one or more divisible activities. It is based on the simplex method using multipliers [1] and the critical path algorithm through a network without divisible activities. A similar algorithm is developed for a machine loading problem [2].

2. Mathematical Preparations

We shall introduce the following notations.

A_0: the subset of the (numbers of) arcs that represent the locations of the undivisible activities.

$A_k (k=1, 2, \cdots, K)$: the subset of the arcs that represent the locations where the k-th divisible activity can be allocated.

We shall assume that $A_j \cap A_k = \phi (j \neq k)$ and that $A_1 \cup A_2 \cup \cdots \cup A_K = \{\mu_1, \mu_2, \cdots, \mu_M\}$.

$T_\mu (\mu \in A_0)$: the completion time for the activity corresponding to arc μ.

$U_k (k=1, 2, \cdots, K)$: the total completion time needed for the k-th divisible activity.

$\Pi_1, \Pi_2, \cdots, \Pi_J$: all paths from the node which represents the start of the project to the node which represents the termination through the project network.

Then the problem can be formulated as the following linear programming P.

Problem P: Minimize t,

subject to

(1) $$t + \sum_{m=1}^{M} a_{mj} t_{\mu_m} \geqq C_j \qquad (j=1, 2, \cdots, J),$$

(2) $$\sum_{m=1}^{M} a_{mj} t_{\mu_m} \geqq C_j \qquad (j=J+1, J+2, \cdots, J+K),$$

(3) $$t_{\mu_m} \geqq 0 \qquad (m=1, 2, \cdots, M),$$

where

(4) $$a_{mj} = \begin{cases} -1 & \text{if } 1 \leqq j \leqq J \text{ and } \mu_m \in \Pi_j, \\ 1 & \text{if } J+1 \leqq j \leqq J+K \text{ and } \mu_m \in A_j, \\ 0 & \text{otherwise}, \end{cases}$$

(5) $$C_j = \begin{cases} \sum_{\mu \in \Pi_j \cap A_0} T_\mu & \text{if } 1 \leqq j \leqq J, \\ U_{j-J} & \text{if } J+1 \leqq j \leqq J+K. \end{cases}$$

We shall condiser the dual problem D to Problem P.

Problem D: Maximize $\left(\sum_{j=1}^{J+K} C_j x_j\right)$,

subject to

(6) $$\sum_{j=1}^{J} x_j = 1,$$

(7) $$\sum_{j=1}^{J+K} a_{mj} x_j \leqq 0 \qquad (m=1, 2, \cdots, M),$$

(8) $$x_j \geqq 0 \qquad (j=1, 2, \cdots, J+K).$$

Problem D can be reduced to Problem D' which involves only equality restraints by introducing slack variables y's.

Problem D': Minimize $\left(z = -\sum_{j=1}^{J+K} C_j x_j\right)$,

subject to

(9) $$\sum_{j=1}^{J} x_j = 1,$$

(10) $$\sum_{j=1}^{J+K} a_{mj} x_j + y_m = 0 \qquad (m=1, 2, \cdots, M),$$

(11) $$x_j \geqq 0 \qquad (j=1, 2, \cdots, J+K),$$

(12) $$y_m \geqq 0 \qquad (m=1, 2, \cdots, M).$$

The system of equalities in Problem D' is shown in Table 1.1. It is not canonical form[1], but the constant factor is one in the first equation

1) A system of M equations of N variables $x_1, x_2, \cdots, x_N (N \geqq M)$, is said to be in canonical form if there exists an ordered subset $(x_{j_1}, x_{j_2}, \cdots. x_{j_M})$ of M variables such that for each m, x_{j_m} has a unit coefficient in the m-th equation and has zero coefficients elsewhere.

B.V.	x_1	··	x_j	··	x_J	x_{J+1}	··	x_{J+k}	··	x_{J+K}	y_1	··	y_m	··	y_M	$-z$	1
(1)	1	··	1	··	1	0	··	0	··	0	0	··	0	··	0	0	1
(y_1)	a_{11}	··	a_{1j}	··	a_{1J}	$a_{1,J+1}$	··	$a_{1,J+k}$	··	$a_{1,J+K}$	1	··	0	··	0	0	0
⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮	⋮
(y_m)	a_{m1}	··	a_{mj}	··	a_{mJ}	$a_{m,J+1}$	··	$a_{m,J+k}$	··	$a_{m,J+K}$	0	··	1	··	0	0	0
⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮	⋮
(y_M)	a_{M1}	··	a_{Mj}	··	a_{MJ}	$a_{M,J+1}$	··	$a_{M,J+k}$	··	$a_{M,J+K}$	0	··	0	··	1	0	0
($-z$)	$-C_1$	··	$-C_j$	··	$-C_J$	$-C_{J+1}$	··	$-C_{J+k}$	··	$-C_{J+K}$	0	··	0	··	0	1	0

Table 1. 1.

B.V.	x_1	··	x_j	··	x_J	x_{J+1}	··	x_{J+k}	··	x_{J+K}	y_1	··	y_m	··	y_M	$-z$	1
x_1	1	··	1	··	1	0	··	0	··	0	0	··	0	··	0	0	1
y_1	0	··	$a_{1j}-a_{11}$	··	$a_{1J}-a_{11}$	$a_{1,J+1}$	··	$a_{1,J+k}$	··	$a_{1,J+K}$	1	··	0	··	0	0	$-a_{11}$
⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮	⋮
y_m	0	··	$a_{mj}-a_{m1}$	··	$a_{mJ}-a_{m1}$	$a_{m,J+1}$	··	$a_{m,J+k}$	··	$a_{m,J+K}$	0	··	1	··	0	0	$-a_{m1}$
⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮	⋮
y_M	0	··	$a_{Mj}-a_{M1}$	··	$a_{MJ}-a_{M1}$	$a_{M,J+1}$	··	$a_{M,J+k}$	··	$a_{M,J+K}$	0	··	0	··	1	0	$-a_{M1}$
$-z$	0	··	$-C_j+C_1$	··	$-C_J+C_1$	$-C_{J+1}$	··	$-C_{J+k}$	··	$-C_{J+K}$	0	··	0	··	0	1	C_1

Table 1. 2.

B.V.	x_1	··	x_j	··	x_J	x_{J+1}	··	x_{J+k}	··	x_{J+K}	y_1	··	y_m	··	y_M	··	$-z$	1
*	$a_{01}^{(i)}$	··	$a_{0j}^{(i)}$	··	$a_{0J}^{(i)}$	$a_{0,J+1}^{(i)}$	··	$a_{0,J+k}^{(i)}$	··	$a_{0,J+K}^{(i)}$	$b_{01}^{(i)}$	··	$b_{0m}^{(i)}$	··	$b_{0M}^{(i)}$	··	0	$b_{00}^{(i)}$
⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮		⋮		⋮		⋮	⋮
*	$a_{m1}^{(i)}$	··	$a_{mj}^{(i)}$	··	$a_{mJ}^{(i)}$	$a_{m,J+1}^{(i)}$	··	$a_{m,J+k}^{(i)}$	··	$a_{m,J+K}^{(i)}$	$b_{m1}^{(i)}$	··	$b_{mm}^{(i)}$	··	$b_{mM}^{(i)}$	··	0	$b_{m0}^{(i)}$
⋮	⋮		⋮		⋮	⋮		⋮		⋮	⋮		⋮		⋮		⋮	⋮
*	$a_{M1}^{(i)}$	··	$a_{Mj}^{(i)}$	··	$a_{MJ}^{(i)}$	$a_{M,J+1}^{(i)}$	··	$a_{M,J+k}^{(i)}$	··	$a_{M,J+K}^{(i)}$	$b_{M1}^{(i)}$	··	$b_{Mm}^{(i)}$	··	$b_{MM}^{(i)}$	··	0	$b_{M0}^{(i)}$
$-z$	$c_1^{(i)}$	··	$c_j^{(i)}$	··	$c_J^{(i)}$	$c_{J+1}^{(i)}$	··	$c_{J+k}^{(i)}$	··	$c_{J+K}^{(i)}$	$d_1^{(i)}$	··	$d_m^{(i)}$	··	$d_M^{(i)}$	··	1	$d_0^{(i)}$

Table 1. 3.

Table 1. Simplex Tableaus for Problem D'

or zero elsewhere, so we may consider the constant factor as if one of the basic variables, and Table 1.1. may be used as a simplex tableau for cycle 0. It can be easily reduced to canonical form by pivoting on some variable x_j $(1 \leqq j \leqq J)$. For example, by pivoting on x_1, Table 1.2. is obtained. It is a standard simplex tableau, so the simplex method can be carried out as usual after cycle 1. Here we shall assume that Table 1.3. is obtained at cycle i. Then

$$B^{(i)} \equiv \begin{pmatrix} b_{00}^{(i)} & b_{01}^{(i)} & \cdots & b_{0M}^{(i)} \\ b_{10}^{(i)} & b_{11}^{(i)} & \cdots & b_{1M}^{(i)} \\ \vdots & \vdots & & \vdots \\ b_{M0}^{(i)} & b_{M1}^{(i)} & \cdots & b_{MM}^{(i)} \end{pmatrix}$$

is the inverse of the basis for cycle i and

$$d^{*(i)} \equiv (d_0^{(i)}, d_1^{(i)}, \cdots, d_M^{(i)})$$

is the simplex multiplier vector. Therefore, we have for $j=1, 2, \cdots, J$,

$$\text{(13)} \qquad \begin{aligned} c_j^{(i)} &= d_0^{(i)} + \sum_{m=1}^{M} a_{mj}\, d_m^{(i)} - C_j \\ &= d_0^{(i)} - \{ \sum_{\mu_m \in \Pi_j} d_m^{(i)} + \sum_{\mu \in \Pi_j \cap A_0} T_\mu \}, \end{aligned}$$

$$\text{(14)} \qquad \begin{aligned} a_j^{(i)} &= b_0^{(i)} + \sum_{m=1}^{M} a_{mj}\, b_m^{(i)} \\ &= b_0^{(i)} - \sum_{\mu_m \in \Pi_j} b_m^{(i)}, \end{aligned}$$

and for $j=J+1, J+2, \cdots, J+K$,

$$\text{(15)} \qquad \begin{aligned} c_j^{(i)} &= \sum_{m=1}^{M} a_{mj}\, d_m^{(i)} - C_j \\ &= \sum_{\mu_m \in A_{j-J}} d_m^{(i)} - U_{j-J}, \end{aligned}$$

$$\text{(16)} \qquad \begin{aligned} a_j^{(i)} &= \sum_{m=1}^{M} a_{mj}\, b_m^{(i)} \\ &= \sum_{\mu_m \in \Pi_{j-J}} b_m^{(i)}, \end{aligned}$$

where

$$a_j^{(i)}=\begin{pmatrix} a_{0j}^{(i)} \\ a_{1j}^{(i)} \\ \vdots \\ a_{Mj}^{(i)} \end{pmatrix} \quad \text{and} \quad b_m^{(i)}=\begin{pmatrix} b_{0m}^{(i)} \\ b_{1m}^{(i)} \\ \vdots \\ b_{Mm}^{(i)} \end{pmatrix}.$$

From (13), to find s such that $c_s^{(i)}=\min_{1\leq j\leq J} c_j^{(i)}$ is equivalent to seeking a critical path through the network where the length of are μ is given by T_μ if $\mu\in A_0$ or $d_m^{(i)}$ if $\mu=\mu_m(m=1, 2, \cdots, M)$. Consequently, if the critical path length $p^{(i)}$ is greater than $d_0^{(i)}$, then we may take x_s, the variable corresponding to the critical path Π_s, into the basic set in the next cycle. Or, if (i) $p^{(i)}\leqq d_0^{(i)}$, (ii) $d_m^{(i)}\geqq 0$ for $m=1, 2, \cdots, M$, and (iii) $\sum_{\mu_m\in A_k} d_m^{(i)}\geqq U_k$ for $k=1, 2, \cdots, K$, an optimal solution to D' is obtained from Table 1.3., and also an optimal solution to P is given by putting $t=d_0^{(i)}$, $t_{\mu_m}=d_m^{(i)}(m=1, 2, \cdots, M)$.

Here, if we apply the simplex method using multipliers together with an algorithm for seeking a critical path through the network without divisible activities, we need not list all paths and their coefficient vectors, but it is enough to take up only paths that become critical at some cycle. The algorithm is shown in detail in the next section.

3. Algorithm

Let $b_m^{*(i)}=(b_{m0}^{(i)}, b_{m1}^{(i)}, \cdots, b_{mM}^{(i)})$.

The algorithm is as follows:

Step 1. Let $B^{(0)}=I$ and $d^{*(0)}=0$.

Step 2. Choose a path Π as you like, and let

$$u_0=1,$$

$$u_m=\begin{cases} -1 & \text{if } \mu_m\in\Pi, \\ 0 & \text{otherwise}, \end{cases}$$
$$(m=1, 2, \cdots, M)$$

$$v=-\sum_{\mu\in\Pi\cap A_0} T_\mu ,$$

$$r=0, \quad \text{and} \quad i=1 .$$

Step 3. Let

$$b_r^{*(i)}=\frac{1}{u_r}b_r^{*(i-1)},$$

$$b_m^{*(i)}=b_m^{*(i-1)}-\frac{u_m}{u_r}b_r^{*(i-1)}(m\neq r),$$

$$d^{*(i)}=d^{*(i-1)}-\frac{v}{u_r}b_r^{*(i-1)}.$$

Step 4. If there exists s such that $d_s^{(i)}<0$, put $v=d_s^{(i)}$, $u=b_s^{(i)}$, and go to Step 9.

Step 5. If there exists s such that $\sum_{\mu_m\in A_s} d_m^{(i)}<U_s$, put $v=\sum_{\mu_m\in A_s} d_m-U_s$ and $u=\sum_{\mu_m\in A_s} b_m$, and go to Step 9.

Step 6. Seek a critical path Π through the network where the length of arc μ is equal to T_μ if $\mu\in A_0$ or $d_m^{(i)}$ if $\mu=\mu_m(m=1, 2, \cdots, M)$.

Step 7. Let p be the length of Π. If $v=d_0^{(i)}-p\geqq 0$, terminate.

Step 8. Put $u=b_0^{(i)}-\sum_{\mu_m\in\Pi} b_m^{(i)}$.

Step 9. Find r such that $u_r>0$ and that

$$\frac{1}{u_r}b_r^{*(i)}=\underset{m(u_m>0)}{\text{lexico-min}}\frac{1}{u_m}b_m^{*(i)},$$

where lexico-min means the minimum by the lexicographic rule[2].

Step 10. Add 1 to i, and go back to Step 3.

At termination, $(t, t_{\mu_1}, t_{\mu_2}, \cdots, t_{\mu_m})=d^{*(i)}$ is an optimal solution to P. Since

2) A vector $\boldsymbol{a}$ is said to be minimum in a set of vectors by the lexicographic rule if for any vector $\boldsymbol{b}$ in it, $\boldsymbol{a}-\boldsymbol{b}$ is equal to zero vector or has non-zero elements the first of which is negative.

we need not obtain an optimal solution to D', any set of basic variables of any cycle is not recorded. Note that there exists an m such that $u_m>0$ at Step 9. For, let us assume that $u_m \leq 0$ for all m. Then, the set of z values has no lower bound, and so, Problem P has no feasible solution. On the other hand, by letting

$$t_{\mu_m} = U_k \quad \text{for all} \quad \mu_m \in A_k \ (k=1, 2, \cdots, K), \quad \text{and}$$

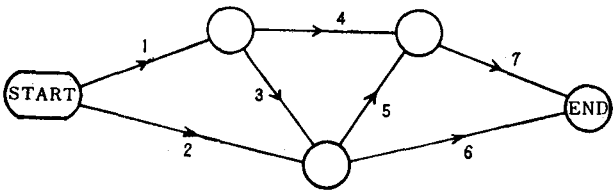
$$t = N \times \max \{\max_{\mu \in A_0} T_\mu, \ \max_k U_k\},$$

where N is the number of arcs, a feasible solution to Problem P is obtained, which is contradiction. Hence, there exists an m such that $u_m>0$ at Step 9.

4. Illustative Example

Let us apply the algorithm for a project network with two divisible activities as shown in Fig. 1.

By letting $\Pi=\{1, 4, 7\}$ at first, and carrying out the steps of the algorithm as in Table 2., an optimal solution to P such that $(t, t_1, t_2, t_5, t_6)=(12, 0, 3, 5, 9)$, is obtained.

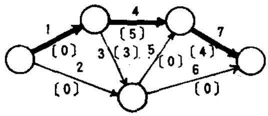
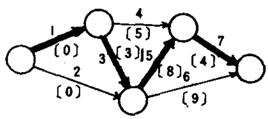


$A_0=\{3, 4, 7\}$, $T_3=3$, $T_4=5$, $T_5=4$,

$A_1=\{\mu_1=1, \mu_2=6\}$, $U_1=9$,

$A_2=\{\mu_3=2, \mu_4=5\}$, $U_2=8$.

Fig. 1. Example of Project Network

Cycle i	$B^{(i)}$					$a^{(i)}$	$u=B^{(i)}a^{(i)}$	Caluculation for v. ($A_1=\{\mu_1=1,\ \mu_2=6\}$, $A_2=\{\mu_3=2,\ \mu_4=5\}$)
	$d^{*(i)}$					—	v	
	1						①	
		1					-1	
			1					
0				1				
					1			
						—	-9	$\Pi=\{1,4,7\}$, $p=9$, $v=-9$
	1							
	1	1				1	1	
			1			1	①	$v=\sum_{\mu_m\in A_1} d_m^{(1)}-U_1=-9$
1				1				
					1			
	9					—	-9	
	1							
	1	1	-1					
			1					
2				1		1	1	$v=\sum_{\mu_m\in A_2} d_m^{(2)}-U_2=-8$
					1	1	①	
	9		9			—	-8	
	1					1	1	
	1	1	-1			-1		
			1					
3				1	-1		①	
					1	-1	-1	$\Pi=\{1,3,5,7\}$, $p=15$,
	9		9		8	—	-6	$v=-6$

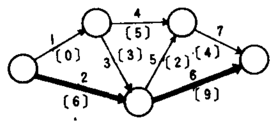
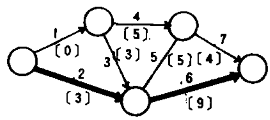
Cycle i	$B^{(i)}$					$a^{(i)}$	$u = B^{(i)}a^{(i)}$	Caluculation for v. ($A_1=\{\mu_1=1,\ \mu_2=6\}$, $A_2=\{\mu_3=2,\ \mu_4=5\}$)
	$d^{*(i)}$					—	v	
4	1			−1	1	1	②	
	1	1	−1				2	
			1			−1	−1	
				1	−1	−1	−1	
				1			−1	
	9		9	6	2	—	−6	$\Pi=\{2, 6\}$, $p=15$, $v=-6$
5	0.5			−0.5	0.5			
		1	−1	1	−1			
	0.5		1	−0.5	0.5			
	0.5			0.5	−0.5			
	0.5			0.5	0.5			
	12		9	3	.5			$\Pi=\{2, 6\}$, $p=12$, $v=0$
Notes	The zero elements are omitted. The encircled elements indicate the positions of pivot.							The bracketed numbers indicate arc lengths. The bold lines indicate the critical paths.

Table 1. Tableaus for computations.

5. Remarks

This algorithm is more available when the project network has much more paths than arcs corresponding to locations of divisible activities. It is better that a path Π selected at Step 2 is such that is found as fast as possible or is critical through the network where the length of arc μ is equal to T_μ if $\mu \in A_0$ or 0 if $\mu \notin A_0$ as in the example. In the latter case, it hardly occurs that there exists m such that $d_m^{(i)} < 0$ at some cycle i. It is easy to revise the algorithm so as to be applicable to the case when a_{mj} is equal to E_j, not necessarily one, if $J+1 \leq j \leq J+K$ and $\mu_m \in A_{j-J}$.

References

[1] Dantzig, G., *Linear Programming and Extensions*, Princeton University Press, 1963.

[2] Glassey, C.R., "An Algorithm for a Machine Loading Problem," *J. Ind. Eng.* 18 (1967), 584–588.

[3] Jewell, W.S., "Divisible Activities in Critical Path Analysis," *Opns. Res.*, 13 (1965), 747–760.

[4] Kobayashi, T., "Note on the Critical Path Analysis for a Project with a Divisible Activity," *JORSJ* 9 (1967), 137–140.