Study on a Mathematical Basis of Critical Path Method and its Application

By

Yoshimi NAGAO* and Kazuhiro YOSHIKAWA*

(Received January 19, 1966)

This paper is concerned with establishing the mathematical basis of the Critical Path Method—a new tool for planning and scheduling projects.

The mathematical model upon which the Critical Path Method is based is a parametric linear program that has the objective of computing the utility of a project as a function of its duration.

In order to solve the parametric linear program, primal dual algorithm may effectively be used and be solved efficiently by network flow method.

This paper contains also the application of this algorithm to the scheduling of quaywall construction at Pier No. 8E, Port of Kobe.

1. Introduction

Until just a few years ago, there was no generally accepted formal procedure to aid in the management of projects. Each manager had his own scheme, which often involved the limited use of the bar chart—a useful tool in production management but inadequate one for the complex interrelationship associated with project management.

During the past few years growing interest has developed in the problems of managing large projects.

While it is generally realized that the fundamental characteristic of all projects is that all the activities involved must be performed in some well defined order, it appears that little has been done to make explicit use of this fact. Recently, however, two parallel efforts, which take their origins in the series-parallel relations among project activities, have been underway. One of these efforts is called the PERT system, and the other effort is called the Critical Path Method.

In this paper, we deal with a mathematical basis of the Critical Path Method. The mathematical model upon which the Critical Path Method is based is a parametric linear program that has the objective of computing

* Department of Civil Engineering.

the utility of a project as a function of its duration. For each feasible project duration, a feasible project schedule is obtained that has a maximum utility among all feasible schedules of the same project duration.

In order to solve the parametric linear program, primal dual algorithm may effectively be used and solved efficiently by network flow method.

Our objective in this paper will be to review a primal dual algorithm and a network flow algorithm for finding a feasible project schedule to the Critical Path Method.

Applying this algorithm to the scheduling of quaywall construction at Port of Kobe, it was made clear that this algorithm is a general procedure which works well along with the analysis of Critical Path Method for the scheduling of public works.

2. Formulation of the Problem

Let E be a finite partially ordered set of $n+1$ elements called events. There are two distinguished event in E, source and sink, respectively, with the property that source precedes and sink follows every event in E.

Each event is denoted by a nonnegative integer, its label. Since E is partially ordered, we may assume that the events are labeled such that if event i precedes event j then $i<j$. In particular, source is given the label 0 and sink is given the label n.

Also associated with event i is a nonnegative number, t_i, which represents the time at which the event occurs. Thus, if event i precedes event j then $t_i \leq t_j$. We will always let $t_0=0$.

An activity is an element, (i, j), of $E \times E$, such that $i<j$. Associated with each activity is a nonnegative number, y_{ij}, its duration. It is assumed that activity (i, j) must be performed sometime between the occurrences of event i and event j. Thus we must have

$$y_{ij}+t_i-t_j \leq 0. \tag{1}$$

A project, P, is a set of events and activities with the property that if event k is in P then k is either source or sink, or else there exist events i and j in P such that activities (i, k) and (k, j) are both in P.

An assignment of durations, y_{ij}, to activities and occurrence times, t_i, to events in P is called a schedule. A schedule will be denoted by $\{\boldsymbol{y}, \boldsymbol{t}\}$, where $\boldsymbol{y}$ and $\boldsymbol{t}$ are vectors whose coordinates are the y_{ij} and t_i, respectively, which define the schedule. If there are m activities in P, $\{\boldsymbol{y}, \boldsymbol{t}\}$ may be interpreted as a vector in an $(m+n+1)$ dimensional Euclidean space.

Sometimes the duration of an activity is a matter of management decision subject to certain restrictions. The simplest restrictions, and the only ones with which we will deal, are that y_{ij} be bounded above and below for each activity in P. That is, there are numbers d_{ij} and D_{ij} such that

$$0 \leq d_{ij} \leq y_{ij} \leq D_{ij} < \infty \tag{2}$$

for all (i, j) in P. We will call D_{ij} the normal duration of activity (i, j): d_{ij} will be called the crash duration.

The value of d_{ij} is an approximation to the fastest time in which an activity can be performed and is determined by the nature of the activity and the environment in which it must be performed. On the other hand, D_{ij} must usually be established by fiat. It represent a reasonable performance time under "normal" circumstances.

A schedule satisfying equations (1) and (2) with $t_0=0$ is called a feasible schedule.

The duration actually selected for each activity when forming a feasible schedule is made to depend upon its utility. For the moment we will assume that the utility of an activity is a linear function of its duration on the closed interval defined by equation (2) and has the form: $c_{ij}y_{ij}$, where $0 \leq c_{ij} < \infty$.

The utilities of the individual activities in P,

$$U(\lambda) = \sum_{(i,j)\in P} c_{ij} y_{ij}. \tag{3}$$

The duration of a schedule is $\lambda = t_n$.

It is clear that among all feasible schedules having a given duration, λ, there is at least one which has maximum utility, i.e., maximizes equation (3). Such a feasible schedule will be called optimal. We denote this values of equation (3) for this schedule by $U(\lambda)$.

Considered as a function of λ, $U(\lambda)$ will be called the project utility function.

3. The Primal Dual Algorithm

We may view the problem of maximizing equation (3) subject to equations (1) and (2) with $t_0=0$ and $t_n=\lambda$ as a parametric linear program with parameter λ. We propose to use the primal dual algorithm to solve it and proceed as follows:

Let $\{\boldsymbol{y}, \boldsymbol{t}\}$ be an optimal feasible schedule of duration λ and define the following sets of activities:

$$\left.\begin{aligned} Q_1 &= \{(i, j)|y_{ij}+t_i-t_j = 0, \ (i, j)\in P\}, \\ Q_2 &= \{(i, j)|y_{ij} = D_{ij} > d_{ij}, \ (i, j)\in P\}, \\ Q_3 &= \{(i, j)|y_{ij} = D_{ij} = d_{ij}, \ (i, j)\in P\}, \\ Q_4 &= \{(i, j)|y_{ij} = d_{ij} < D_{ij}, \ (i, j(\in P\}. \end{aligned}\right\} \tag{4}$$

The salient features of the primal dual algorithm, when specialized to the present case may be summarized in the following.

Let $\{\boldsymbol{y}, \boldsymbol{t}\}$ be an optimal feasible schedule of duration λ. It remains to develop a method for solving equations (1), (2) and (3). To do consider the restricted dual of equations (1), (2) and (3) called the restricted dual program:

Find σ_{ij}, $(i, j)\in P$ and δ_i, $0\leq i\leq n$, that minimize the linear form

$$\sum_{(i,j)\in P} c_{ij}\sigma_{ij}, \tag{5}$$

subject to

$$\left.\begin{array}{lll} 1. & p_{ij} = \sigma_{ij}+\delta_i-\delta_j \geq 0, & (i, j)\in Q_1 \\ 2. & \sigma_{ij} \geq 0, & (i, j) = Q_1\cap Q_2 \\ 3. & \sigma_{ij} = 0, & (i, j) = P-(Q_1-Q_3) \\ 4. & \sigma_{ij} \leq 0, & (i, j) = Q_1\cap Q_4 \\ 5. & \delta_0 = 0 \ \ and \ \ \delta_n = 1, & \end{array}\right\} \tag{6}$$

then the solution $\{\boldsymbol{y}^*, \boldsymbol{t}^*\}$ defined by

$$\left.\begin{array}{ll} y_{ij}^* = y_{ij}-\theta\sigma_{ij}, & (i, j)\in P \\ t_i^* = t_i-\theta\delta_i, & 0 \leq i \leq n \end{array}\right\} \tag{7}$$

is an optimal feasible schedule of duration $\lambda^*=\lambda-\theta$, where $0\leq\theta\leq\theta_0$ and

$$\theta = \min[\theta_1, \theta_2, \theta_3], \tag{8}$$

where

$$\left.\begin{aligned} \theta_1 &= \begin{cases} \min\limits_{p_{ij}<0} [(y_{ij}+t_i-t_j)/p_{ij}], \\ +\infty, \ \ if \ \ p_{ij} \geq 0 \ \ for \ \ all \ (i, j)\in P, \end{cases} \\ \theta_2 &= \begin{cases} \min\limits_{\sigma_{ij}<0} [(y_{ij}-D_{ij})/\sigma_{ij}], \\ +\infty, \ \ if \ \ \sigma_{ij} \geq 0 \ \ for \ all \ \ (i, j)\in P, \end{cases} \\ \theta_3 &= \begin{cases} \min\limits_{\sigma_{ij}>0} [(y_{ij}-d_{ij})/\sigma_{ij}], \\ +\infty, \ \ if \ \ \sigma_{ij} \leq 0 \ \ for \ all \ \ (i, j)\in P. \end{cases} \end{aligned}\right\} \tag{9}$$

However, if equation (6) is inconsistent then there are no feasible schedules of duration less than λ.

The primal dual algorithm now consists in finding an optimal feasible schedule $\{\boldsymbol{y}, \boldsymbol{t}\}$ of duration λ and then solving equations (5) and (6) to determine $\{\boldsymbol{y}^*, \boldsymbol{t}^*\}$ of duration $\lambda=\theta_0$. $\{\boldsymbol{y}^*, \boldsymbol{t}^*\}$ is called a characteristic schedule.

Using this new optimal feasible schedule the process is repeated until no feasible schedules of shorter duration can be found. At this point the algorithm terminates.

4. A Network Flow Algorithm

It remains to develop a method for solving equations (5) and (6). To do this consider the dual of equations (5) and (6), called the restricted primal problem.

In order to formulate the restricted primal problem, we may consider the following dual linear programs:

$$D|\lambda):\quad \left.\begin{array}{ll} 1. & y_{ij}+t_i-t_j \le 0, \\ 2. & y_{ij} \le D_{ij}, \\ 3. & -y_{ij} \le -d_{ij}, \\ 4. & -t_0+t_n \le \lambda, \\ & U(\lambda) = \max \sum_{(i,j)\in P} c_{ij}y_{ij}. \end{array}\right\} \tag{10}$$

$$P|\lambda):\quad \left.\begin{array}{ll} 1. & f_{ij},\ g_{ij},\ h_{ij},\ v_{ij} \ge 0, \\ 2. & \sum_{(i,j)\in P} f_{ij} = \sum_{(j,k)\in P} f_{jk}, \qquad (1 \le j \le n-1), \\ 3. & \sum_{(0,j)\in P} f_{0j} = \sum_{(i,n)\in P} f_{in} = v, \\ 4. & f_{ij}+g_{ij}-h_{ij} = c_{ij}, \\ & P(\lambda) = \min (\lambda v + \sum_{(i,j)\in P} D_{ij}g_{ij} - \sum_{(i,j)\in P} d_{ij}h_{ij}. \end{array}\right\} \tag{11}$$

From the above equations of primal problem, the equations of restricted primal problem are obtained as follows:

$$RP|\lambda):\quad \left.\begin{array}{ll} 1. & f_{ij},\ g_{ij},\ h_{ij},\ v_{ij} \ge 0, \\ 2. & \sum_{(i,j)\in P} f_{ij} = \sum_{(j,k)\in P} f_{jk}, \qquad (1 \le j \le n-1), \\ 3. & \sum_{(0,j)\in P} f_{0j} = \sum_{(i,n)\in P} f_{in} = v, \\ 4. & f_{ij}+g_{ij}-h_{ij} = c_{ij}, \\ 5. & \sum (t_j-t_i-y_{ij})f_{ij} + \sum (D_{ij}-f_{ij})g_{ij} \\ & \quad + \sum (y_{ij}-d_{ij})h_{ij} + (\lambda - t_n + t_0)v = 0, \\ & \max \sum_{(0,j)\in P} f_{0j}. \end{array}\right\} \tag{12}$$

On the other hand, in the optimal feasible schedule of duration λ, we can assume that

$$\left.\begin{array}{l} t_0 = 0, \quad t_n = \lambda, \\ y_{ij} = \min (D_{ij},\ t_j-t_i). \end{array}\right\} \tag{13}$$

Then, it is not necessary to consider the term $(\lambda - t_n + t_0)v$ in equation (12)-5.

We may use the following expressions insted of equations (12) through the above assumption.

$$RP|\lambda): \left.\begin{array}{ll} 1. & f_{ij},\ g_{ij},\ h_{ij} \geq 0, \\ 2. & \sum_{(i,j)\in P} f_{ij} = \sum_{(j,k)\in P} f_{jk}, \quad (1 \leq j \leq n-1), \\ 3. & f_{ij} + g_{ij} - h_{ij} = c_{ij}, \\ 4. & \text{(i)}\ y_{ij} + t_i - t_j < 0 \Rightarrow f_{ij} = 0, \\ & \text{(ii)}\ y_{ij} < D_{ij} \qquad \Rightarrow g_{ij} = 0, \\ & \text{(iii)}\ y_{ij} > d_{ij} \qquad \Rightarrow h_{ij} = 0, \\ & \max \sum_{(0,j)\in P} f_{0j} \end{array}\right\} \quad (14)$$

In these equations, if $g_{ij}=0$, then we have $f_{ij} \geq c_{ij}$ and if $h_{ij}=0$, then we have $f_{ij} \leq c_{ij}$.

Finally, we can obtain the following equations of restricted primal problem as an equivalent expression of equations (14):

Find f_{ij}, $(i, j)\in P$, that maximize the linear form

$$\sum_{(0,j)\in P} f_{0j}, \quad (15)$$

subject to 1. $\sum_{(i,j)\in P} f_{ij} - \sum_{(j,k)\in P} f_{jk} = 0, \quad (1 \leq j \leq n-1),$ (16)

and

$$\left.\begin{array}{lll} 2. & 0 \leq f_{ij} \leq c_{ij}, & (i, j)\in Q_1 \cap Q_2, \\ & & (e(i, j, 1) = 0,\ e(i, j, 2) > 0), \\ 3. & 0 \leq f_{ij} = c_{ij}, & (i, j)\in Q_1 - (Q_2 \cup Q_3 \cup Q_4), \\ & & (e(i, j, 1) < 0,\ e(i, j, 2) > 0), \\ 4. & f_{ij} \geq c_{ij}, & (i, j)\in Q_1 \cap Q_4, \\ & & (e(i, j, 1) < 0,\ e(i, j, 2) = 0), \\ 5. & f_{ij} = 0, & (i, j)\in P - Q_1, \\ & & (e(i, j, 1) > 0). \end{array}\right\} \quad (17)$$

Where, $e(i, j, 1)$ and $e(i, j, 2)$ defined by

$$\left.\begin{array}{l} e(i, j, 1) = t_j - (t_i + D_{ij}) \\ e(i, j, 2) = t_j - (t_i + d_{ij}) \end{array}\right\} \quad (18)$$

are called the floater.

We may interpret f_{ij} to be the amount of a homogeneous commodity being transported through a network whose nodes correspond to the events of P and whose branches correspond to the activities of P. Equations (16) are flow conservation equations. Capacity restrictions on the allowable flow in a branch are stated in constraints (17).

The problem is to maximize the flow into node n subject to the capacity restrictions. The following labeling method is used to solve the maximal flow problems.

The algorithm may be started with the zero flow. The computation then progresses by a sequence of "labelings" (Routine I below), each of which either results in a flow of higher value (Routine II below) or terminates with the conclusion that the present flow is maximal.

Given an integral flow f_{ij}, we proceed to assign labels to nodes of the network, a label having one of the forms

$$L(j) = [i \pm \varepsilon(j) | LS], \tag{19}$$

where i indicates the node from which one came to label node j and $\varepsilon(j)$ indicates the minimum, but positive, excess branch flow along the path to node j.

During Routine I, a node is considered to be in one of three stages: unlabeled, labeled and scanned or labeled and unscanned. In equation (19), L indicates the labeled and S indicates the scanned.

Initially all nodes are unlabeled.

Routine I (labeling process).

1) First the source 0 receives the label

$$L(0) = [\triangle\triangle\infty | L\triangle], \tag{20}$$

where, $\triangle$ denotes the blank. The source is now labeled and unscanned: all other nodes are unlabeled.

2) Consider any labeled node i, not yet scanned.

(i) If $(i, j) \in Q_1 \cap Q_2$ for some unlabeled node j and $f_{ij} < c_{ij}$, assign the label

$$L(j) = [i + \varepsilon(j) | L\triangle] \tag{21}$$

to node j, where

$$\varepsilon(j) = \min[\varepsilon(i), c_{ij} - f_{ij}]. \tag{22}$$

(ii) If $(i, j) \in Q_1 \cap (Q_3 \cup Q_4)$ for some unlabeled node j, assign the label

$$L(j) = [i + \varepsilon(i) | L\triangle] \tag{23}$$

to node j.

(iii) If $(i, j) \in Q_1 - (Q_2 \cup Q_3 \cup Q_4)$ for some unlabeled node j, leave node j unlabeled.

3) Consider any unlabeled node i, not yet scanned.

(i) If $(i, j) \in Q_1 \cap (Q_2 \cup Q_3)$ and $f_{ij} > 0$, assign the label

$$L(i) = [j - \varepsilon(i) | L\triangle] \tag{24}$$

to node i, where

$$\varepsilon(i) = \min [\varepsilon(j), f_{ij}] . \tag{25}$$

(ii) If $(i, j) \in Q_1 \cap Q_4$ and $f_{ij} > c_{ij}$, assign the label

$$L(i) = [j - \varepsilon(i) | L\triangle] \tag{26}$$

to node i, where

$$\varepsilon(i) = \min [\varepsilon(j), f_{ij} - c_{ij}] . \tag{27}$$

(iii) If $(i, j) \in Q_1 - (Q_2 \cup Q_3 \cup Q_4)$, leave node i unlabeled.

Use labeling rules 2) and 3) alternately where applicable until it is no longer possible to label an unlabeled node. When applying these rules, if a node is a candidate for a label in several ways, use any applicable label. When the labeling process terminates through the above procedure, source and other nodes i, j are scanned, then we replace $\triangle$ by S in scanned part of all labels.

In this way, when the labeling process terminates, if sink n is labeled, process to Routine **II**. If sink n is not labeled, the algorithm terminates, the maximum flow having been obtained.

Routine II (flow change).

4) The sink n has been labeled

$$L(n) = [k + \varepsilon(n) | L\triangle] , \tag{28}$$

replace f_{kn} by $f_{kn} + \varepsilon(n)$.

5) Now consider event k, in general, if node k is labeled

$$L(k) = [j + \varepsilon(k) | LS] , \tag{29}$$

replace f_{jk} by $f_{jk} + \varepsilon(n)$, if it is labeled

$$L(k) = [j - \varepsilon(k) | LS] , \tag{30}$$

replace f_{jk} by $f_{jk} - \varepsilon(n)$.

6) Now process in the same manner to consider node j. Eventually source will be reached. At that time Routine **II** terminates. Using the new values of f_{ij} and erasing all labels, Routine **I** is repeated.

This completes the rules for the network flow algorithm.

5. Determination of Optimal Feasible Schedule and Project Cost

If f_{ij}, $(i,j) \in P$ is an optimal flow for the restricted primal problem associated with the optimal feasible schedule $\{y, t\}$ of duration λ, then it is also a feasible flow for the restricted primal problem associated with the optimal feasible schedule $\{y^*, t^*\}$ of duration $\lambda - \theta_0$.

Let I be the set of labeled and unscanned nodes and J the set of unlabeled nodes obtained at the termination of the flow algorithm.

Further, let

$$\left.\begin{aligned} Q_5 &= \{(i, j) \mid i \in I,\ j \in J \ \ \textit{and} \ \ (i, j) \in Q_1\}, \\ Q_6 &= \{(i, j) \mid i \in J,\ j \in I \ \ \textit{and} \ \ (i, j) \in Q_1\}. \end{aligned}\right\} \tag{31}$$

Then we obtain

$$\left.\begin{aligned} &1.\ \ \textit{if} \ \ i \in I,\ j \in J,\ \ \textit{then} \\ &\quad f_{ij} = c_{ij},\ \ \textit{if} \ \ (i, j) \in Q_1 - (Q_3 \cup Q_4), \\ &2.\ \ \textit{if} \ \ i \in J,\ j \in I,\ \ \textit{then} \\ &\quad f_{ij} = c_{ij},\ \ \textit{if} \ \ (i, j) \in Q_1 - (Q_2 \cup Q_3), \\ &\quad f_{ij} = 0,\ \ \ \textit{if} \ \ (i, j) \in Q_1 \cap (Q_2 \cup Q_3), \\ &3.\ \ \textit{if} \ \ i \in I \ \ \textit{and} \ \ (i, j) \in Q_1 \cap (Q_3 \cup Q_4), \\ &\quad \textit{then} \ \ j \in I. \end{aligned}\right\} \tag{32}$$

σ_{ij} and δ_i defined by

$$\left.\begin{aligned} \sigma_{ij} &= 1, && \textit{if} \ \ (i, j) \in Q_1 - (Q_3 \cup Q_4), \\ & && (e(i, j, 1) \le 0,\ \ e(i, j, 2) > 0), \\ & && \textit{and} \ \ i \in I,\ \ j \in J, \\ \sigma_{ij} &= -1, && \textit{if} \ \ (i, j) \in Q_1 - (Q_2 \cup Q_3), \\ & && (e(i, j, 1) < 0,\ \ e(i, j, 2) \ge 0), \\ & && \textit{and} \ \ i \in J,\ \ j \in I, \\ \sigma_{ij} &= 0, && \textit{otherwise} \\ \textit{and} \ \ \delta_i &= 0,\ \ i \in I, \\ \delta_i &= 1,\ \ i \in J, \end{aligned}\right\} \tag{33}$$

constitute of feasible solution to equations (6) through the maximum flow minimum cut theorem.

Then, in order to obtain the optimal feasible schedule $\{y^*, t^*\}$ of duration $\lambda^* = \lambda - \theta_0$, we can use the following equations insted of equations (7):

$$
\left.\begin{array}{lll}
1) & 1.\ y^*_{ij} = y_{ij} - \xi, & if\ \ i \in I,\ \ j \in J, \\
 & & e(i,j,1) \leq 0\ \ and \\
 & & e(i,j,2) > 0, \\
 & 2.\ y^*_{ij} = y_{ij} + \xi, & if\ \ i \in J,\ \ j \in I, \\
 & & e(i,j,1) < 0\ \ and \\
 & & e(i,j,2) \geq 0, \\
 & 3.\ y^*_{ij} = y_{ij}, & otherwise, \\
2) & 1.\ t^*_i = t_i, & if\ \ i \in I, \\
 & 2.\ t^*_i = t_i - \xi, & if\ \ i \in J,
\end{array}\right\} \tag{34}
$$

where

$$
\left.\begin{array}{l}
\xi = \min(\xi_1, \xi_2), \\
\xi_1 = \min\limits_{Q_5} [e(i,j,k)], \\
\xi_2 = \min\limits_{Q_6} [-e(i,j,2)], \qquad (k = 1, 2).
\end{array}\right\} \tag{35}
$$

and

Finally, increasing project cost ΔP of duration $\lambda^* = \lambda - \theta_0$ is obtained as follows:

$$
\left.\begin{array}{l}
\Delta P = \xi[\sum\limits_{Q_5} f_{ij} - \sum\limits_{Q_6} f_{ij}] \\
\quad = \xi \sum\limits_{(0,j) \in P} f_{0j}.
\end{array}\right\} \tag{36}
$$

6. Application to the Scheduling of Quaywall Construction at Port of Kobe

Port of Kobe has been developed into an international port which has a large hinterland called "Hanshin Industrial Area". It is visit by many liners from all over the world.

The economic development of the world including Japan has increased the amount of foreign trade among many countries. Consequently, the number of liners or other ships coming in and out of Port of Kobe has increased much more and the size of the ships has also increased.

In order to accelerate the progress of economic development of Japan through foreign trade, we must increase and rationalize port facilities.

After the World War II, Pier No. 7, No. 8 W and Maya Wharves were constructed, but Port of Kobe met with a severe shortage of wharves due to the rapid growth of Japanese economy, and a new plan for Pier No. 8 E was developed to cope with imminent needs.

The structure adopted for quaywall of Pier No. 8 E is shown in Fig. 1 using a circular cell made of steel plate insted of sheet piles. If the steel plate cellular bulkhead type is possible to be adopted for quaywall, the construction cost will be cheaper, the process of work will be easier and execution will be more rapid than other types of quaywall.

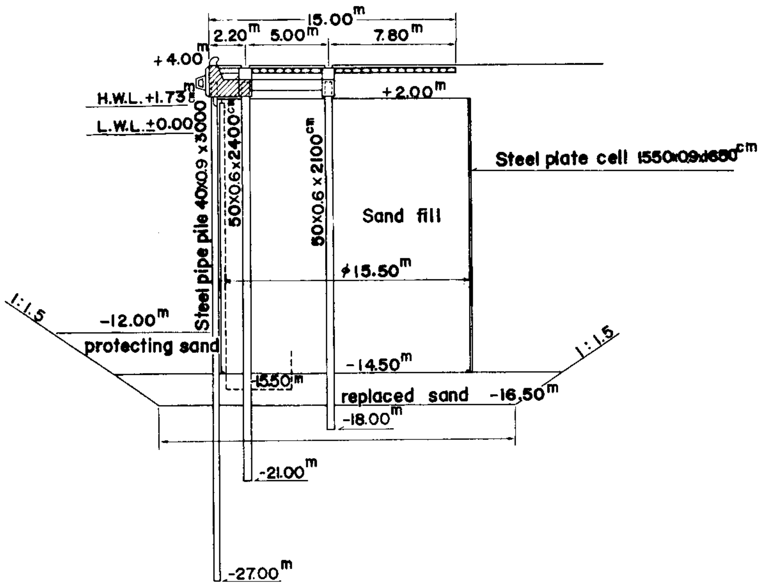


Fig. 1. Standard cross section of quaywall. (steel plate cellular quaywall).

This type of structure was already constructed for the quaywalls of Maya Wharves. As a matter of fact, it was the first test in the world, so the method of the design calculation might not be precise.

After the construction of this type of quaywall, Port of Kobe was attacked several times by strong storms, and this structure of quaywall withstood these storms. From these experiences, it is certain that there are no problems in the making of a steel plate cellular bulkhead.

It may be thought that the diameter of the cellular bulkhead for quaywall of Pier No. 8 E should become the same diameter of 15.5 m.

To resist the vertical load and horizontal force, the steel pipe piles of diameter 40 cm will be driven vertically in front of the cellular bulkhead and ones of diameter 50 cm will be driven vertically inside the cellular bulkhead, then they will be connected by steel forms.

The following is the summary of execution of this work.

Owing to the soft foundation, the sea bed will be replaced by river sand of good quality with grab type dreadger for 4 m^3 capacity. The steel plate cell will be carried and set in place by a floating crane, then filled with soil.

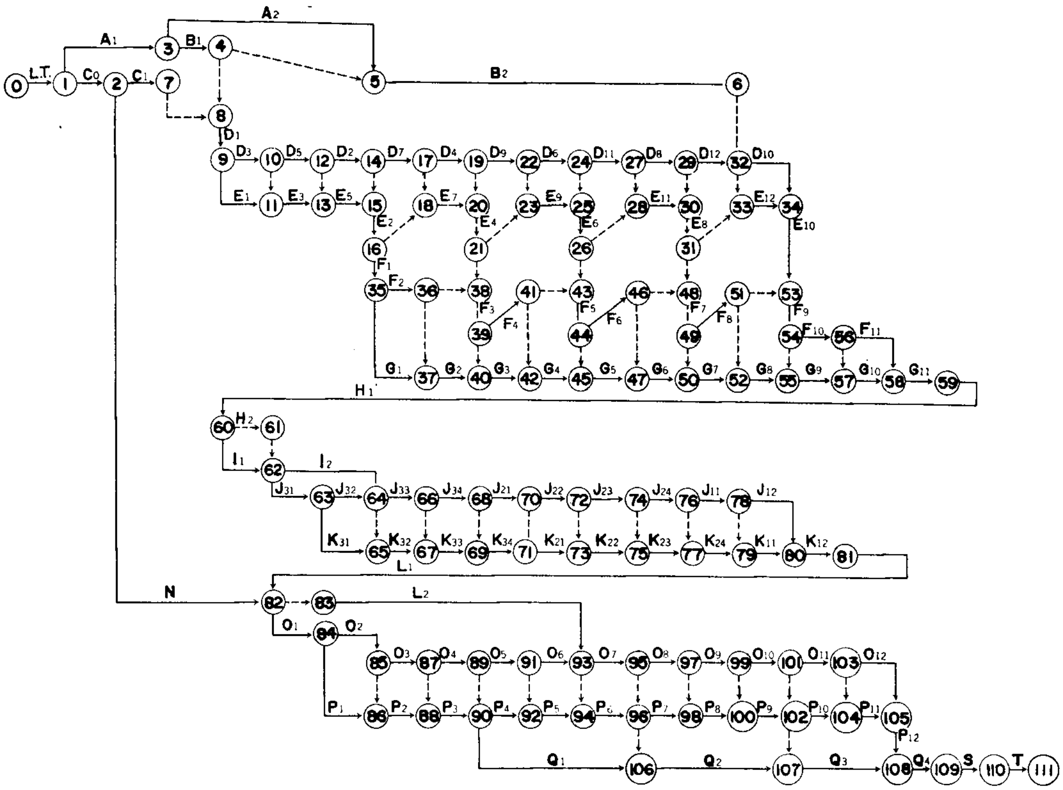
The cell will be of 9 mm thick steel plate and the stiffening steel plates of 30~35 cm in width, 9 mm thickness will be fixed at three points in radial direction and eight parts in the cylindrical direction. Total weight become 63 tons each. These will be constructed in the shipyards in Kobe.

As soil of good quality can not be obtained from the seabed, mountain soil that has less clay will be brought for the filling. We can fill 3,000 m³ of soil in celles in 3~4 days.

The driving of steel pipe piles shown in Fig. 1 will be done by the piling barge.

The sequencing relations among the activities of the quaywall construction project at Pier No. 8 E are shown in an arrow diagram of Fig. 2.

Note that in Fig. 2 there are certain activities represented by brokenline



Note :

A	Dredging Sea Bed	K	Connecting Battered Piles by Steel Forms
B	Replacing Sand	L	Concreat-placing of Retaining Block
C	Steel Plate Cell Construction	N	Precast Beams Construction
D	Carrying and Setting of Steel Plate Cell	O	Setting of Precast Beams on Piles
E	Filling Soil in Cell	P	Connecting Battered Piles and Precast Beams
F	Driving Sheet Pile of Arc Part	Q	Retaining-wall Construction between Precast Beams
G	Filling Soil in Arc Part	S	Setting of Stopper
H	Placing Sand for Protection of Cell	T	Pavement
I	Driving Sand Piles by Vibrocomposer		
J	Driving Battered Pile		

Fig. 2. Sequencing relations among the activities of the quaywall construction project.

arrows. These are "dummy" activities that only signify the sequencing of certain other activities.

Let us assume that the utility of an activity is measured in terms of its cost. Maximizing utility then means minimizing cost. The result of the project utility function computation is a project cost curve that is piecewise linear, nonincreasing, and convex where it is defined. However, this cost curve generally only reflects the direct costs involved in performing project activities. These costs include such things as labor, equipment and materials --- the direct costs of the project.

Table 1 is a summary of the information supplied to supervision for the quaywall construction problem of Fig. 2.

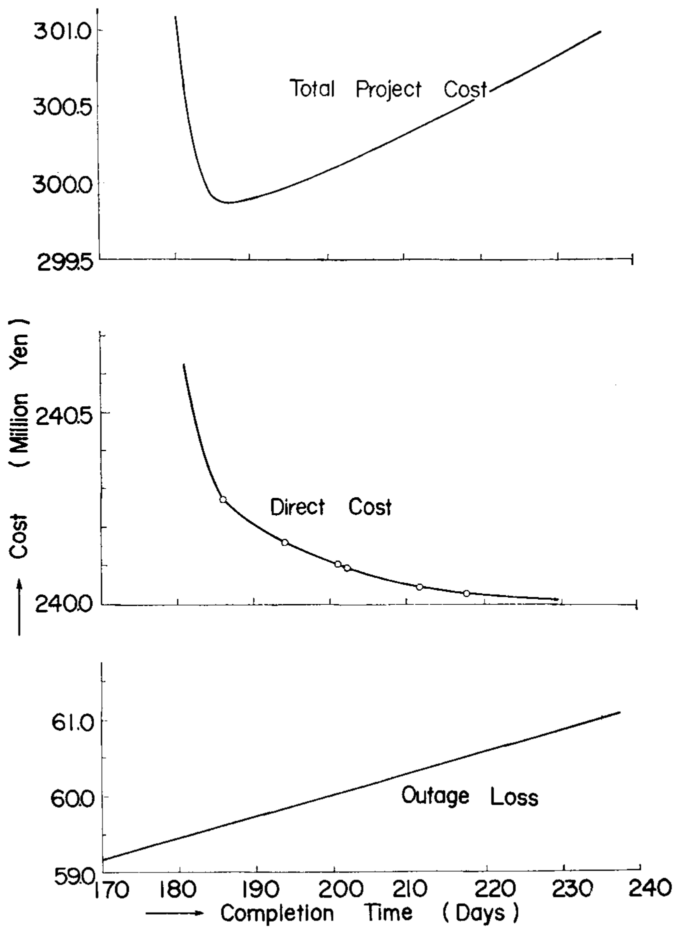


Fig. 3. Project cost curve.

Table 1. Summary of the information for the quaywall construction problem.

Activity code	Sequence i	Sequence j	Duration D_{ij}	Duration d_{ij}	Cost slope	Activity code	Sequence i	Sequence j	Duration D_{ij}	Duration d_{ij}	Cost slope
L.T.	0	1	10	10	∞	Dummy	26	43	0	0	0
C_0	1	2	21	13	414.2	Dummy	27	28	0	0	0
A_1	2	3	11	8	51.5	D_8	27	29	1	1	∞
C_1	2	7	21	13	414.2	E_{11}	28	30	4	3	1.3
N	2	82	66	48	6.8	Dummy	29	30	0	0	0
B_1	3	4	3	2	1.7	D_{12}	29	32	1	1	∞
A_2	3	5	11	8	51.5	E_8	30	31	4	3	1.3
Dummy	4	5	0	0	0	Dummy	31	33	0	0	0
Dummy	4	8	0	0	0	Dummy	31	48	0	0	0
B_2	5	6	3	2	1.7	Dummy	32	33	0	0	0
Dummy	6	32	0	0	0	D_{10}	32	34	1	1	∞
Dummy	7	8	0	0	0	E_{12}	33	34	4	3	1.3
D_1	8	9	1	1	∞	E_{10}	34	53	4	3	1.3
D_3	9	10	1	1	∞	F_2	35	36	1	1	∞
E_1	9	11	4	3	1.3	G_1	35	37	1	1	∞
Dummy	10	11	0	0	0	Dummy	36	37	0	0	0
D_5	10	12	1	1	∞	Dummy	36	38	0	0	0
E_3	11	13	4	3	1.3	G_2	37	40	1	1	∞
Dummy	12	13	0	0	0	F_3	38	39	1	1	∞
D_2	12	14	1	1	∞	Dummy	39	40	0	0	0
E_5	13	15	4	3	1.3	F_4	39	41	1	1	∞
Dummy	14	15	0	0	0	G_3	40	42	1	1	∞
D_7	14	17	1	1	∞	Dummy	41	42	0	0	0
E_2	15	16	4	3	1.3	Dummy	41	43	0	0	0
Dummy	16	18	0	0	0	G_4	42	45	1	1	∞
F_1	16	35	1	1	∞	F_5	43	44	1	1	∞
Dummy	17	18	0	0	0	Dummy	44	45	0	0	0
D_4	17	19	1	1	∞	F_6	44	46	1	1	∞
E_7	18	20	4	3	1.3	G_5	45	47	1	1	∞
Dummy	19	20	0	0	0	Dummy	46	47	0	0	0
D_9	19	22	1	1	∞	Dummy	46	48	0	0	0
E_4	20	21	4	3	1.3	G_6	47	50	1	1	∞
Dummy	21	23	0	0	0	F_7	48	49	1	1	∞
Dummy	21	38	0	0	0	Dummy	49	50	0	0	0
Dummy	22	23	0	0	0	F_8	49	51	1	1	∞
D_6	22	24	1	1	∞	G_7	50	52	1	1	∞
E_9	23	25	4	3	1.3	Dummy	51	52	0	0	0
Dummy	24	25	0	0	0	Dummy	51	53	0	0	0
D_{11}	24	27	1	1	∞	G_8	52	55	1	1	∞
E_6	25	26	4	3	1.3	F_9	53	54	1	1	∞
Dummy	26	28	0	0	0	Dummy	54	55	0	0	0

(Table 1 continued)

Activity code	Sequence		Duration		Cost slope	Activity code	Sequence		Duration		Cost slope
	i	j	D_{ij}	d_{ij}			i	j	D_{ij}	d_{ij}	
F_{10}	54	56	1	1	∞	L_2	83	93	12	8	13.8
G_9	55	57	1	1	∞	O_2	84	85	1	1	∞
Dummy	56	57	0	0	0	P_1	84	86	1	1	∞
F_{11}	56	58	1	1	∞	Dummy	85	86	0	0	0
G_{10}	57	58	1	1	∞	O_3	85	87	1	1	∞
G_{11}	58	59	1	1	∞	P_2	86	88	1	1	∞
H_1	59	60	3	2	1.7	Dummy	87	88	0	0	0
H_2	60	61	3	2	1.7	O_4	87	89	1	1	∞
I_1	60	62	11	8	2.0	P_3	88	90	1	1	∞
Dummy	61	62	0	0	0	Dummy	89	90	0	0	0
J_{31}	62	63	3	2	7.7	O_5	89	91	1	1	∞
I_2	62	64	11	8	2.0	P_4	90	92	1	1	∞
J_{32}	63	64	3	2	7.7	Q_1	90	106	5	3	4.2
K_{31}	63	65	2	1	2.0	Dummy	91	92	0	0	0
Dummy	64	65	0	0	0	O_6	91	93	1	1	∞
J_{33}	64	66	3	2	7.7	P_5	92	94	1	1	∞
K_{32}	65	67	2	1	2.0	Dummy	93	94	0	0	0
Dummy	66	67	0	0	0	O_7	93	95	1	1	∞
J_{34}	66	68	3	2	7.7	P_6	94	96	1	1	∞
K_{33}	67	69	2	1	2.0	Dummy	95	96	0	0	0
Dummy	68	69	0	0	0	O_8	95	97	1	1	∞
J_{21}	68	70	3	2	7.7	P_7	96	98	1	1	∞
K_{34}	69	71	2	1	2.0	Dummy	96	106	0	0	0
Dummy	70	71	0	0	0	Dummy	97	98	0	0	0
J_{22}	70	72	3	2	7.7	O_9	97	99	1	1	∞
K_{21}	71	73	2	1	2.0	P_8	98	100	1	1	∞
Dummy	72	73	0	0	0	Dummy	99	100	0	0	0
J_{23}	72	74	3	2	7.7	O_{10}	99	101	1	1	∞
K_{22}	73	75	2	1	2.0	P_9	100	102	1	1	∞
Dummy	74	75	0	0	0	Dummy	101	102	0	0	0
J_{24}	74	76	3	2	7.7	O_{11}	101	103	1	1	∞
K_{23}	75	77	2	1	2.0	P_{10}	102	104	1	1	∞
Dummy	76	77	0	0	0	Dummy	102	107	0	0	0
J_{11}	76	78	5	4	7.7	Dummy	103	104	0	0	0
K_{24}	77	79	2	1	2.0	O_{12}	103	105	1	1	∞
Dummy	78	79	0	0	0	P_{11}	104	105	1	1	∞
J_{12}	78	80	5	4	7.7	P_{12}	105	108	1	1	∞
K_{11}	79	80	3	2	2.0	Q_2	106	107	5	3	4.2
K_{12}	80	81	3	2	2.0	Q_3	107	108	5	3	4.2
L_1	81	82	12	8	13.8	Q_4	108	109	5	3	4.2
Dummy	82	83	0	0	0	S	109	110	9	7	1.3
O_1	82	84	1	1	∞	T	110	111	36	30	7.3

On the basis of this information, that of Table 1 and the arrow diagram of Fig. 2, we may apply the algorithm of the previous sections to obtain the direct project cost curve approximated in Fig. 3.

Clearly there are other costs that contribute to the total project cost such as overhead and distributives, and perhaps penalties for not completing the project or a portion of it by a certain time. These external costs must also be taken into account when management plans how the project should be implemented relative to over-all objectives. The major portion of the external costs usually vary only with the duration of the project. Thus, they form a cost curve that will be called the indirect cost curve of the project.

A typical question that management might ask is "How should the project be implemented so that the total project cost is minimal?". The answer to this question can be approximated by adding the direct and indirect cost curves together to form a total project cost curve and then selecting the schedule corresponding to the minimum total project cost.

Indirect cost curve and total project cost curve are also included in Fig. 3.

7. Concluding Remarks

In this paper, we deal with the mathematical basis of the Critical Path Method. The mathematical model is based on a parametric linear program that has the objective of computing the utility of a project as a function of its duration. For each feasible project duration, a feasible project schedule is obtained that has maximum utility among all feasible schedules of the same project duration.

A primal dual algorithm and a network flow algorithm have been developed for finding a feasible project schedule to the Critical Path Method.

In the application of this algorithm, calculations have been executed on an automatic digital computer FACOM 222.

In these calculations, most parts of calculating time were spent to find the minimum cut. Therefore, it is desirable to establish an another algorithm to find the minimum cut more easily. In order to develop another algorithm, a new research project is already started by our research groups. The next paper to be prepared by the authors in the near future, will deal with the new algorithm for finding a feasible project schedule to the Critical Path Method.

Acknowledgment

The authors wish to express their deepest appreciation to Prof. Dr. Tojiro Ishihara for his cordial instruction and also to express our hearty gratitude to Messrs. S. Akagi, M. Haruna of Kyoto University and K. Ikegami of FACOM Co. Ltd. for their assistance throughout this research.

References

1) J. E. Kelly, Jr. : Critical Path Planning and Scheduling: Mathematical Basis, JORSA, 7, 1959 & JORSA 9, 1961.
2) J. E. Kelly, Jr. : Parametric Programming and Primal Dual Algorithm, JORSA, 7, 1959.
3) J. E. Kelly, Jr. : Extension of the Construction Scheduling Problem, UNIVAC Application Research Center, Remington Rand UNIVAC, Philadelphia, November 18, 1958.
4) J. E. Kelly, Jr. : Critical Path Planning and Scheduling : Case Histories, JORSA, 8, 1960.
5) L. R. Ford, Jr. & D. R. Fulkerson : Maximal Flow Through a Network, Canad. J. Math., 8, 1956.
6) L. R. Ford, Jr. & D. R. Fulkerson : A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem, Canad. J. Math., 9, 1957.
7) D. R. Fulkerson : A Network Flow Computation for Project Cost Curves, Management Science, 7, 1961, Math. Division, RAND corp., March 18, 1960.
8) D. R. Fulkerson : Expected Critical Path Lengths in PERT Networks, JORSA, 10, 1962.
9) D. R. Fulkerson & M. R. Walker : Critical Path Planning and Scheduling, Mauchly Associates, Inc., Ambler, Pa., 1959.