PERIODICA POLYTECHNICA SER. CIVIL ENG. VOL. 37, NO. 3, PP. 231–247 (1993)

AN ALGORITHM FOR SOLVING THE COST OPTIMIZATION PROBLEM IN PRECEDENCE DIAGRAMMING METHOD[1]

Miklós HAJDU

Faculty of Civil Engineering,
Department of Building Management and Organization
Technical University of Budapest
H–1521 Budapest, Hungary

Received: June 2, 1993

Abstract

In this paper an extension is given to the original CPM problem first solved by KELLEY and WALKER (1959) later by FULKERSON (1961) to the Precedence Diagram network (PDM). The following precedence relationships are allowed between activities: Start-to-Start-t (SSt), Finish-to-Start-t (FSt), Start-to-Finish-t, (SFt), Finish-to-Finish-t (FFt). The solution of the problem is based on network flows theory.

Keywords: network technique, precedence diagramming, time cost trade-offs.

1. Introduction

In this paper we give an algorithm to the problem to minimize the project cost in precedence diagrammimg network plan.

The introduction and solution of the CPM/cost problem first appeared in KELLEY and WALKER's paper. (KELLEY & WALKER, 1959). The result of KELLEY's work is an algorithm based on the primal dual algorithm of linear programming. The solution of the problem by flow algorithm (KELLEY, 1961) can be found in FULKERSON's paper (FULKERSON, 1961).

The earlier versions of precedence diagramming appeared in the work of ROY (ROY, 1958) and FONDAHL (FONDAHL, 1961). Their concepts gained further notice by J. D. CRAIG in the users manual of IBM 1440 Project Control System (IBM 1964).

When we developed our method we used KELLEY (KELLEY , 1961) and FULKERSON's (FULKERSON, 1961) results.

In our paper we assume that the reader is at home in network flows the element and the basics problems of network theory such as digraph, maximum flow minimum cut problem, shortest and longest path through

[1]This research is supported by OTKA F4112

a network, etc. We have no possibilities to discuss these in this present paper.

2. The Model and the Solution of the Problem

An $[N, \mathcal{A}]$ directed graph is given. The graph can only have one source and sink, and there must be a path from source (s) to sink (t) through all $i \in N$. There can be more arrows between any two nodes, loops are not allowed. The nodes represent activities. The duration of the i^{th} activity let be τ_i. The activities are carried out continuously in time, splitting is not allowed. The arrows serve to describe the technological and organizational relations between activities.

Between any two activities the following precedence relations are allowed: Start-to-Start-z_{ij} (SSz_{ij}); Finish-to-Start-z_{ij} (FSz_{ij}); Finish-to-Finish-z_{ij} (FFz_{ij}); Start-to-Finish-z_{ij} (SFz_{ij}).

These relations give the minimal allowable distance between the beginning (finishing) of i activity and the beginning (finishing) of j activity.

SSz_{ij}– means that at least z_{ij} or more lag time has to be between the beginning of i and the beginning of j activity.

FSz_{ij}– means that at least z_{ij} or more lag time has to be between the finishing of i and the beginning of j activity.

SFz_{ij}– means that at least z_{ij} or more lag time has to be between the beginning of i and the finishing of j activity.

FFz_{ij}– means that at least z_{ij} or more lag time has to be between the finishing of i and the finishing of j activity.

We call the above mentioned precedence relations as minimal relations as z_{ij} stands for the minimal distance between the distinguished points of activities. In network planning the so-called maximal precedence relations are also used but in that case z_{ij} stands for the maximal allowable distance between the distinguished points of activities. We have mentioned these types of precedence relations only for the sake of completeness as we extend the cost optimization problem to the network where the use of maximal relations is not allowed.

There are given to all activities a lower and higher time bound, the crash and the normal duration. Their notations a_i and b_i ($a_i \leq b_i$). Let be given to the normal duration of all activities a Cn_i normal cost, which shows the cost of the activity if its accomplishing duration is b_i. Moreover there is given a c_i cost factor to all i activities, which shows how much the associated cost increases when the duration of an activity is decreased by one day. Knowing all this we can determine the so-called crash cost corresponding to crash duration:

$$Cr_i = Cn_i + c_i(b_i - a_i).$$

Further restriction to c_i is that it should not be negative that is:

$$Cr_i \geq Cn_i \quad \forall\ (i) \in N.$$

All this can be shown on *Fig. 1*.

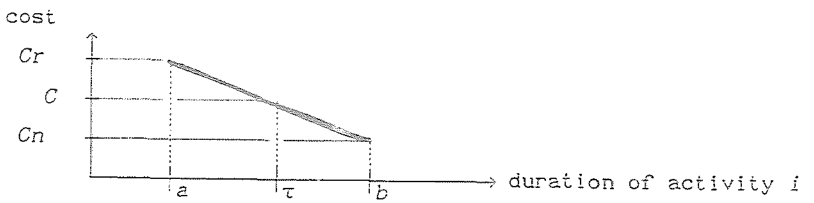


Fig. 1.

The actual activity duration is τ_i. It must be between the crash and normal duration

$$a_i \leq \tau_i \leq b_i, \qquad \forall(i) \in N. \tag{1}$$

Let's denote the beginning of an i activity π_{iS} and the finishing of it by π_{iF}. As splitting of activities is not allowed π_{iS} determines π_{iF} and vice versa. The following conditions can be noted between activities

$$\pi_{jS} - \pi_{iS} \geq z_{ij} \qquad \forall(i,j) \in \mathcal{A} \text{ and SS}z_{ij}; \tag{2a}$$
$$\pi_{jF} - \pi_{iF} \geq z_{ij} \qquad \forall(i,j) \in \mathcal{A} \text{ and FF}z_{ij}; \tag{2b}$$
$$\pi_{jF} - \pi_{iS} \geq z_{ij} \qquad \forall(i,j) \in \mathcal{A} \text{ and SF}z_{ij}; \tag{2c}$$
$$\pi_{jS} - \pi_{iF} \geq z_{ij} \qquad \forall(i,j) \in \mathcal{A} \text{ and FS}z_{ij}. \tag{2d}$$

Eliminating the π_F values by using of $\pi_F = \pi_S + \tau$ equality and introducing the notations $\tau^F_{i_{ij}}$ and $\tau^F_{j_{ij}}$ the conditions (2a-d) will change as follows:

$$\pi_j - \pi_i + \tau^F_j - \tau^F_i \geq z_{ij} \qquad \forall(i,j) \in \mathcal{A} \tag{2}$$

where π_i, π_j are the beginning of the activities, and

$$\tau^F_{j_{ij}} = \begin{cases} \tau_j & \text{If the } (i,j) \text{ relation runs into the finish of } j^{th} \text{ activity which means SF or FF relation.} \\ 0 & \text{If the } (i,j) \text{ relation runs into the beginning of } j \text{ activity which means SS or FS relation.} \end{cases}$$

$$\tau^F_{i;j} = \begin{cases} \tau_i & \text{If the } (i,j) \text{ relation runs from the end of} \\ & i^{th} \text{ activity which means a FS or FF relation.} \\ 0 & \text{If the } (i,j) \text{ relation runs from the beginning of} \\ & i^{th} \text{ activity which means SS or SF relation.} \end{cases}$$

DEFINITION: We call a given precedence relation with respect to a given activity a finish type relation if the relation runs from or arrives at the end of the activity. In case of running in the SF and FF relations will be of finish type, in case of running out relations the FS and FF relations will be finish type with respect to the given activity.

Knowing the definition $\tau^F_{i;j}$ can be determined in the following way, too. If the relation is a finish type one with respect to the activity then $\tau^F_{i;j} = \tau_i$ otherwise zero.

Let the beginning of the start activity be zero and the finishing of the last activity be p.

$$\pi_s = 0, \tag{3}$$

$$\pi_t + \tau_t = p. \tag{4}$$

On a network where τ is not fixed but corresponding to (1) can move between an upper and a lower time bound, several project durations can be available. Evidently, the p project duration we want to achieve has to be between the accomplishable minimal and maximal project durations.

$$p_{min} \leq p \leq p_{max}. \tag{5}$$

Changing the duration of activities even the same p duration time can be achieved in sevaral different ways. It is evident that in case the cost slope of the activities different is from the total cost and will correspond to the same duration time. We are searching for the cheapest possible solution corresponding to a given duration time. We can establish the following objective function with the help of *Fig. 1*.

$$\min(\sum_N \{Cn_i + (b_i - \tau_i)c_i\}) \tag{6a}$$

As Cn_i and $(b_i c_i)$ are constant the above objective function is equivalent to the following:

$$\max\Big(\sum_N (c_i \tau_i)\Big). \tag{6}$$

Summarizing the point mentioned above, the mathematical model of the problem is the following:

$$\begin{array}{llllll} a_i & \leq & \tau_i & \leq b_i & \forall\ (i) \in N & (1) \\ \pi_j - \pi_i + \tau_j^F - \tau_i^F & & & \geq z_{ij} & \forall\ (i,j) \in \mathcal{A} & (2) \\ \pi_s & & & = 0 & & (3) \\ \pi_t & + & \tau_t & = p & & (4) \\ p_{min} & \leq & p & \leq p_{max} & & (5) \\ \max & \left(\sum_N (c_i \tau_i)\right) & & & & (6)=(1^*) \end{array}$$

Before we establish the dual problem to this question, we shall introduce some new notions.

DEFINITION: We call a φ_{ij} flow going through an (i,j) arrow a finish type with respect to i node and denote it φ_{ij}^F if the (i,j) precedence relation runs out of the finish of activity i, that is the relation is a finish type relation (FSz_{ij} or FFz_{ij}) with respect to i.

DEFINITION: We call a φ_{ij} flow going through an (i,j) arrow a finish type with respect to i node and denote it φ_{ji}^F if the (i,j) precedence relation runs into the finish of node i, that is the relation is a finish type relation (SFz_{ji} or FFz_{ji}) with respect to i.

Let's add to the network an FS(−p) precedence relation running out of t into s activity. We denote the set of arrows thus increased with $\mathcal{A}^*$. This surplus of arrows does not change the primal problem.

Now we can establish the following dual problem according to the primal problem.

We are to find in the network a φ_{ij} flow, the value of which is Θ and which minimizes the following objective function:

$$\min \{p\Theta - \sum_{A} z_{ij}\varphi_{ij} + \sum_{\substack{c<F \\ \mathcal{A}^*}} [c_i - F_i]]b_i - \sum_{\substack{c<F \\ \mathcal{A}^*}} [c_i - F_i]]a_i,$$

where $F_i := \sum_j \varphi_{ij}^F - \sum_j \varphi_{ji}^F \quad \forall i \in N$ and $(i,j) \in \mathcal{A}^*$.

Thus F_i is the sum of the finish type flows running out from activity i decreased by the sum of the finish type flows running into node i.

The following lemma points out the strict connection between the two problem.

LEMMA : *If there exists a π and τ policy which satisfies the primal, and a φ_{ij} flow for the dual, then* $(1^*) \leq (2^*)$, *that is*

$$\sum_N c_i\tau_i \leq p\Theta - \sum_A z_{ij}\varphi_{ij} + \sum_{\substack{c>F \\ A^*}}[c_i - F_i]b_i - \sum_{\substack{c<F \\ A^*}}[c_i - F_i]a_i$$

and equality holds if and only if the following are satisfied:

if	$\pi_j - \pi_i + \tau_i^F - \tau_i^F > z_{ij}$	then	$\varphi_{ij} = 0$	(1°)
if	$b_i > \tau_i$	then	$c_i \leq F_i$	(2°)
if	$a_i < \tau_i$	then	$c_i \geq F_i.$	(3°)

PROOF:

$$\sum_N c_i\tau_i \leq p\Theta - \sum_A z_{ij}\varphi_{ij} + \sum_{\substack{c>F \\ A^*}}[c_i - F_i]]b_i - \sum_{\substack{c<F \\ A^*}}[c_i - F_i]]a_i$$

If we replace b_i and a_i by τ_i, the value of the dual objective function will definitely decrease. If we can prove that is greater than or equal to the primal objective function, the original statement is proved.

$$\sum_N c_i\tau_i \leq p\Theta - \sum_A z_{ij}\varphi_{ij} + \sum_{\substack{c>F \\ A^*}}[c_i - F_i]\tau_i - \sum_{\substack{c<F \\ A^*}}[F_i - c_i]\tau_i =$$

$$= p\Theta - \sum_A z_{ij}\varphi_{ij} + \sum_A [c_i - F_i]\tau_i =$$

$$= p\Theta - \sum_A z_{ij}\varphi_{ij} + \sum_N c_i\tau_i - \sum_i \left[\sum_{\substack{j \\ A^*}} \varphi_{ij}^F\tau_i - \sum_{\substack{j \\ A^*}} \varphi_{ij}^F\tau_i\right]$$

Subtracting the value of $\sum c_i\tau_i$ from both sides then

$$\Theta p \geq \sum_A z_{ij}\varphi_{ij} + \sum_i \left[\sum_j \varphi_{ji}^F\tau_i - \sum_j \varphi_{ij}^F\tau_i\right]$$

$$\Theta p \geq \sum_A z_{ij}\varphi_{ij} + \sum_A \varphi_{ij}^F\tau_i - \sum_A \varphi_{ji}^F\tau_i + \Theta\tau_t$$

If we replace z_{ij} by $\pi_j - \pi_i + \tau_{j_{ij}}^F - \tau_{i_{ij}}^F$ which is greater than or equal to it, the value of the dual objective function will definitely decrease.

$$\Theta p \geq \sum_A (\pi_j - \pi_i)\varphi_{ij} + \sum_A (\tau_{j_{ij}}^F - \tau_{i_{ij}}^F)\varphi_{ij} + \sum_A \varphi_{ij}^F\tau_i - \sum \varphi_{ij}^F\tau_i + \Theta\tau_t,$$

$$\Theta p \geq \sum_A (\pi_j - \pi_i)\varphi_{ij} + \Theta\tau_t =$$

$$= \sum_A \pi_t\varphi_{it} + \sum_A \pi_s\varphi_{sj} + \Theta\tau_t =$$

$$= (p - \tau_t)\Theta + \Theta\tau_t$$

$$\Theta p \geq p\Theta.$$

With this the inequality of the lemma is proved, and equilibrium exists if and only if suffecting

$$\begin{array}{lll} c_i > F_i & \text{then} & b_i = \tau_i; \\ c_i < F_i & \text{then} & a_i = \tau_i; \\ \varphi_{ji} > 0 & \text{then} & z_{ij} = \pi_j - \pi_i + \tau^F_{j_{ij}} - \tau^F_{i_{ij}}. \end{array}$$

Reversing this,

$$\begin{array}{llll} \pi_j - \pi_i + \tau^F_{j_{ij}} - \tau^F_{i_{ij}} > z_{ij} & \text{then} & \varphi_{ij} = 0 & (1^\circ) \\ b_i > \tau_i & \text{then} & c_i \leq F_i & (2^\circ) \\ a_i < \tau_i & \text{then} & c_i \geq F_i. & (3^\circ) \end{array}$$

These are exactly the equilibrium conditions stated in the lemma. An important consequence of this theorem will be presented as follows:

THEOREM: (weak form of equilibrium)
If there exists a π_i and τ_i policy which satisfies the primal problem to a given p project duration and a φ_{ij} flow on the network and also the value of the primal objective function (1^*) is equal to (2^*) the objective function of the dual problem, which means (1^*)=(2^*), then the solutions of both problems are optimal.

PROOF: (in an indirect way)
Let's denote the value of the primal objective function (1^*) by P and that of the dual by D.

Let's assume that $P = D$, but there exists a P^* solution where $P^* > P$. In this case $P^* > P = D$, but this is a contradiction according to the lemma.

Let's assume that $P = D$, but there exists a better dual solution D^*, where $D^* < D$. In this case $D^* < D = P$ but this is a contradiction according to the lemma. In this way the theorem is proved.

THEOREM (duality theorem)
According to a given p duration time ($p_{min} \leq p \leq p_b$) there is a π, τ and φ policy, where the values of the objective functions are equal, that is optimal. (p_b is the project duration calculated with the normal duration of the activities.)

The proof of the theorem is constructive, that is it gives the algorithm, too.

PROOF:
As starting trivial solution let π_i system be the time policy derived from $\tau_i = b_i$ values and let $\varphi_{ij} = 0$ on all arrows.

This is an optimal solution as the values of the primal and dual objective functions are equal, $\sum_A c_i b_i$. The project duration corresponding to $\tau_i = b_i$ is $p = \pi_t + \tau_t$. We denote this project duration p_b as we have calculated it from the normal duration of activities.

If we know a π, τ, and Φ optimal system corresponding to any p then we can give a p^*, π^*, τ^* and Φ^*, which satisfies the lemma that is also optimal and $p^* < p$.

This statement says that if there exists an optimal solution to a p project duration, we can move on to an optimal solution which corresponds to a smaller project duration. As we know the optimal solution corresponding to a p_b project duration we can give the optimal solution to all project durations, where $p < p_b$.

We comprehend this through a two-step construction.

In the first step we increase φ flow to φ^* so that the duality conditions formulated in the lemma continue to be true, that is the solution is still optimal.

In the second step we decrease p project duration in a way that the duality conditions remain fulfilled, that is the solution is still optimal.

First step:
Let's examine on a certain arrow or on a node which duality conditions can be accomplished in what kind of combination? The '+' indicates that the duality condition is accomplished, the '−' indicates that it is not.

Depending on which conditions are accomplied on a certain arrow or node we can get some information on the φ_{ij} flow passing through on arrows, and some information on F_i passing through on nodes. These tell us what the values of φ_{ij} and F_i should be in case the conditions are satisfied. This information can be found in the fourth column. We have classed the arrows and the nodes in the fifth column $(\mathcal{A}1 - \mathcal{A}2), (N1 - N4)$ considering the φ_{ij} and F_i values passing through them.

Equilibrium conditions 1°	2°	3°	Flow information	Classing of arr.	κ_{ij}	κ_{ji}
+	not defined on the arrows		$\varphi_{ij} = 0$	$\mathcal{A}1$	0	0
−			$\varphi_{ij} \geq 0$	$\mathcal{A}2$	∞	φ_{ij}

1°	2°	3°	F information	Classing of nodes	$\kappa_{i_S i_F}$	$\kappa_{i_F i_S}$
defined	+	–	$F_i \geq c_i$	N1	∞	$F_i - c_i$
only	–	+	$F_i \leq c_i$	N2	$c_i - F_i$	∞
on the	+	+	$F_i = c_i$	N3	0	0
arrows	–	–	F_i no limit	N4	∞	∞

In the first step the flow has to be increased so that the solution remains optimal that is the duality conditions are accomplished to all arrows. If we increase the flows in a way that the arrows remain in the same arrow classes and the activities remain in the same node classes, the solution will be still optimal.

With the help of data on flows we can give the value how much the flow on a certain arrow can be increased or decreased.The sixth and seventh columns show the capacities which have come about as a result of this kind of argument. In the nodes the F_i values have to be chanced so that the equilibrium conditions of the lemma remain valid. The suitable modification of the F_i values can be assured by the following technique.

We cut each node into two and transform it into an arrow. One new node represents the beginning of the activity the other the finishing of it. We connect these two nodes in both directions by an arrow.

Let (i_S, i_F) arrow point from the beginning of the activity to the end, and (i_F, i_S) arrow vice versa. Let the end type relations corresponding to i activity run into/from i_F node and the rest of the relations into/from i_F point. In this case of flows running into i_F, not taking in consideration the flows going through (i_F, i_S) and (i_S, i_F) arrows.

On the network thus transformed we can assure by the correct choice of the capacities of arrows (i_S, i_F) and (i_S, i_F) that the equilibrium conditions in the lemma corresponding to the values of F_i remain. The correct capacities on (i_F, i_S) and (i_F, i_S) arrows are shown in the sixth and seventh columns of the table above. The reader can check this easily himself.

Searching for maximal flow on the transformed network with the above given capacities we get Ψ_{ij} flow. Adding this to the original φ_{ij} flow, and getting back to the original network the new flow on the arrows will be,

$$\varphi^*_{ij} = \varphi_{ij} + \Psi_{ij} \quad \forall (ij) \in \mathcal{A}^*$$

During this step the equilibrium conditions of the lemma will be valid, because we have chosen the capacities so that the arrows and the activities remain in the same class.

With this step the aim of which was to increase the flow by keeping the duality conditions, we have come to the second step.

Second step:
On the transformed network we are to find a $\pi_i^*, \tau_{ij}^*, p^* < p$ system corresponding to the increased φ_{ij}^* flow where the duality conditions are accomplished that is the solution remains optimal. (On the transformed network π_{is} denotes the early start of the activity and π_{iF} the early end, that is $\pi_{iF} - \pi_{iS} = \tau_i$.)

In the first step we were looking for a maximal flow. In this case there exists an (S,T) cut in the transformed network the arrows of which are saturated. In the cut there can be $\mathcal{A}1$ type arrows, and N2, N3 type activities. In the cut backwards there can be $\mathcal{A}1$, $\mathcal{A}2$ type arrows and N1, N3 type nodes.

To determine the new π potentials we give a δ value.

The determination of δ value come about in the following way:

$$\delta := \min\{\delta_{A1}, \delta_{N2}, \delta_{N3}, \delta_{A1^\circ}, \delta_{A2^\circ}, \delta_{N1^\circ}\delta_{N3^\circ}\},$$

where

in case of arrows in the cut:

$$\begin{array}{llllll} \delta_{A1} & := \min\ \{\pi_j - \pi_i + \tau_j - \tau_i - z_{ij} & : & \forall\ (i,j) \in \mathcal{A}1 & (i,j) & \in (\mathrm{S,T})\} \\ \delta_{\mathrm{N2}} & := \min\ \{\tau_i - a_i & : & \forall\ (i) \in \mathrm{N2} & (i_S, i_B) & \in (\mathrm{S,T})\} \\ \delta_{\mathrm{N3}} & := \min\ \{\tau_i - a_i & : & \forall\ (i) \in \mathrm{N3} & (i_S, i_F) & \in (\mathrm{S,T})\}. \end{array}$$

In case of arrows going backwards in the cut:

$$\begin{array}{llllll} \delta_{A1^\circ} & > 0 & & \forall\ (i,j) \in \mathcal{A}1 & (i,j) & \in (\mathrm{S,T}) \\ \delta_{A2^\circ} & > 0 & & \forall\ (i,j) \in \mathcal{A}2 & (i,j) & \in (\mathrm{S,T}) \\ \delta_{\mathrm{N1}^\circ} & = \min\{b_i - \tau_i & : & \forall\ (i) \in \mathrm{N1} & (i_S, i_F) & \in (\mathrm{S,T})\} \\ \delta_{\mathrm{N3}^\circ} & = \min\{b_i - \tau_i & : & \forall\ (i) \in \mathrm{N3} & (i_S, i_F) & \in (\mathrm{S,T})\}. \end{array}$$

The δ value thus produced is definitely larger then zero. Let the new π_i^* potential system on the transformed network be determined according to the following:

$$\pi_i^* = \begin{cases} \pi_i & \text{if} \quad i \in S, \\ \pi_i - \delta & \text{if} \quad i \in T. \end{cases}$$

Thus p will become $p^* = p - \delta$ and an i activity duration can be determined from the following formula:

$$\tau_i^* = \pi_{i_F}^* - \pi_{i_S}^* \qquad \forall \quad i \in N.$$

The δ value had been constructed so that the duality conditions on the arrows continue to be accomplished. Arrows and activities transformed into arrows where both nodes fall into S or T set of points, the arrows and nodes remain in the same class.

In the case of relations and activities in the cut the following change takes place corresponding to the flow.

In the case of $\mathcal{A}1$ arrow type decreasing of π_j by δ_{A1} value the arrow will become $\mathcal{A}2$ type.

In the case of N2 activity type decreasing of π_{iF} by δ_{N3} value the N2 type will become N1 type, with value smaller than this it becomes N3 type.(Suppose that the value of δ_{N2} comes from this node).

In the case of N3 activity type decreasing of π_{iF} by δ_{N3} the N3 type node will become N1 type, with value smaller than this it remains in its own class. (Suppose that the value of δ_{N2} comes from this node).

In the case of relations and activities going backwards in the cut the following change takes place corresponding to the flow.

The $\mathcal{A}1$ type arrow remains $\mathcal{A}11$ type when the π_j potential is changed by any $\delta_{A1^*} > 0$ value.

The $\mathcal{A}2$ type arrow becomes $\mathcal{A}1$ type arrow when changed the potential by any $\delta_{A2^*} > 0$ value.This means that the so far critical relation with respect to the activity will no longer be critical.

Thus δ_{A2^*} can be greater than zero only if there is a critical path leading to activity j. As there is only flow along critical paths (where 1° is satisfied) and (j, i) relation got into the cut in a way that $F_i - c_i$ value flew backwards on it, this assumes that there are critical paths leading to j node which go through nodes other than node i. This automatism assures that by taking δ value optional large the relation still satisfies the duality conditions.

In the case of N1 type nodes if we decrease the π_{iS} value by δ_{N1^*} the activity becomes N2 type, if we decrease it by a smaller value it will become N3 type. (Suppose that the value of δ_{N1^*} comes from this node).

In the case of N3 type nodes if we decrease the π_{iS} value by δ_{N3^*} the activity becomes N2 type, if we decrease it by a smaller value it will remain in its own class. (Suppose that the value of δ_{N3^*} comes from this node).

These changes in types correspond to the flows and by chosing the smallest of theese we assure that all the changes satisfy the duality conditions.

Thus we completed step two.

After all this we have to go back to step one and increase the flow again and than in the second step p project duration can be decreased. These steps must be alternately repeated until we reach the project duration wanted or the flow becomes infinitely great. In this case there exists an $s \to t$ path along which all the capacities are infinite. This means that arrows on this path determine the beginning and the end of the activities, so this is the longest, so the critical path. The nodes on this path can be of classes N1, N4, or N2. The duration of the activity in class N1, which is a_i, can not be further decreased. In the case of activities in class N4 the normal and crash duration are equal $a_i = b_i$ so this activity can not be decreased either. In the case of activities in class N2 the duration of the activity is b_i so it can be decreased, but with this the project duration would increase, because the activity is reversed critical. The length of the critical path can not be further decreased in case of infinite flows. The importance of this reflection, that the algorithm comes to an end if the flow is infinitely great, is quite great because in the case of Precedence Diagramming Method p_{min} cannot be calculated from the crash times, which means that $p_{min} \neq p_a$. It can happen that the two values are the same but the opposite can happen, too.

This algorithm gives us the p_{min} project duration on the network two. This was also an unsolved problem so far.

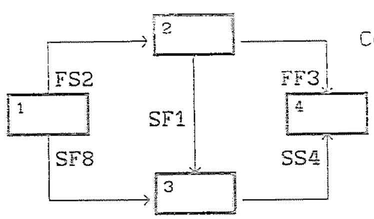
Thus the theorem is constructively proved.

An important consequence of the theorem is presented hereby.

THEOREM: (strong equilibrium)
If there exists an optimal (P) primal solution and an optimal dual solution (D) then their values are equal.

PROOF: According the duality theorem there exists a maximal primal solution (P^*) and a minimal dual solution (D^*) which are optimal that is $(P^*) = (D^*)$. As (P) and (D) are also optimal so $(P) = (P)$ and $(D^*) = (D)$, but then $(P) = (D)$.

3. A Sample Problem to Demonstrate the Algorithm



Codes of act.	1	2	3	4
b_i	4	5	6	10
a_i	1	3	2	4
c_i	3	3	2	1

Fig. 2.

On the given diagram (*Fig. 2*) we can see a network. In the table the normal and crash time values are given. There are also given the cost factors (c_i) corresponding to the activities. The relations between the activities are shown on the arrows. The small digits in the nodes indicate the codes of the activities.

We want to find the optimal solution corresponding to p_{min}.

Step 0.

The calculation of p_b project duration corresponding to the normal activity duration.

It is enough to calculate the earliest start of the activities.

The π_i values stand for the earliest start of the activities. The earliest finish of activities comes from $\pi_{iF} = \pi_i + \tau_i$.

Activity	1	2	3	4
π_i	0	6	2	6

$$p_b = \pi_{tF} = \pi_t + \tau_t = 6 + 10 = 16.$$

To start with let's take the time values $\tau_i = b_i$ and π policy calculated from them and $\varphi_{ij} = 0$ flow. This is the optimal solution corresponding to p_b project duration as the values of the primal and dual objective functions are the same.

Step 1 (increasing the flows)

Table 1

						precedence relations						
						from	(i)	1	1	2	2	3
activity code		1	2	3	4	to	(j)	2	3	3	4	4
classing of nodes	N1-N4	2	2	2	2	classing of arrows	$A1 - A2$	2	2	1	2	1
old F_i values		0	0	0	0	old flow	φ_{ij}	0	0	0	0	0
capacity	$\kappa_{i_S i_F}$	3	3	2	1	capacity	κ_{ij}	∞	∞	0	∞	0
	$\kappa_{i_F i_S}$	∞	∞	∞	∞		κ_{ji}	0	0	0	0	0
						max flow	Ψ_{ij}	0	1	0	0	1
$F_i^* = F_i + \Psi_{ij}$		0	0	-1	1	$\varphi_{ij}^* = \varphi_{ij} + \Psi_{ij}$		0	1	0	0	1

Step 2 (decreasing π_i, τ_i, p).

Table 2

arrows in the cut	from / to				
	from	2	4_S		
	to	4	4_F		
δ values		2	6		
δ min		2			
code of activity		1	2	3	4
earliest start		0	6	2	6
earliest finish		4	11	8	14
act.dur.$\tau_i = \pi_{i_F} - \pi_{i_S}$		4	5	6	8

The new project duration $p = 14$.

The costs have increased by 2 units that is by 1 money unit per time unit.

Step 1^* (increasing the flows)

Table 3

						precedence relations						
						from	(i)	1	1	2	2	3
activity code		1	2	3	4	to	(j)	2	3	3	4	4
classing of nodes	N1-N4	2	2	2	3	classing of arrows	$\mathcal{A}1 - \mathcal{A}2$	2	2	1	2	2
old F_i values		0	0	-1	1	old flow	φ_{ij}	0	1	0	0	1
capacity	$\kappa_{i_S i_F}$	3	3	3	0	capacity	κ_{ij}	∞	∞	0	∞	∞
	$\kappa_{i_F i_S}$	∞	∞	∞	0		κ_{ji}	0	1	0	0	1
						max flow	Ψ_{ij}	3	0	0	3	0
$F_i^* = F_i + \Psi_{ij}$		3	3	-1	1	$\varphi_{ij}^* = \varphi_{ij} + \Psi_{ij}$		3	1	0	3	1

*Step 2** (decreasing π_i, τ_i, p)

Table 4

arrows in the cut	from / to	1_S / 1_F	4_S / 4_F		
δ values		3	4		
δ min		3			
code of activity		1	2	3	4
earliest start		0	3	2	6
earliest finish		1	8	8	11
act.dur.$\tau_i = \pi_{i_F} - \pi_{i_S}$		1	5	6	5

The new project duration $p = 11$.

The costs have increased by 12 money units that is 4 cost unit per time unit.

*Step 1*** (increasing the flows)

Table 5

						precedence relations						
						from	(i)	1	1	2	2	3
activity code		1	2	3	4	to	(j)	2	3	3	4	4
classing of nodes	N1-N4	1	2	2	3	classing of arrows	$A1 - A2$	2	2	1	2	2
old F_i values		3	3	-1	1	old flows	φ_{ij}	3	1	0	3	1
capacity	$\kappa_{i_S i_F}$	∞	0	3	0	capacity	κ_{ij}	∞	∞	0	∞	0
	$\kappa_{i_F i_S}$	0	∞	∞	0		κ_{ji}	3	1	0	3	1
						max flow		0	0	0	0	0
new F_i flow F_i^*		3	3	-1	1	new flow φ_{ij}^*		3	1	0	3	1

*Step 2*** (decreasing π_i, τ_i, p)

Table 6

arrows in the cut	from / to	2_S / 2_F	4_S / 4_F		
δ values		2	1		
δ min			1		
code of activity		1	2	3	4
earliest start		0	3	2	6
earliest finish		1	7	8	10
act.dur.$\tau_i = \pi_{i_F} - \pi_{i_S}$		1	4	6	4

The new project duration $p = 10$.
The costs have increased by 4 units, that is 4 cost units per time unit compared to the previous project duration.

*Step 1**** (increasing of flow)

Table 7

activity code		1	2	3	4	precedence relations from (i) / to (j)		1 / 2	1 / 3	2 / 3	2 / 4	3 / 4
classing of nodes	N1-N4	1	3	2	1	classing of arrows	$\mathcal{A}1 - \mathcal{A}2$	2	2	1	2	2
old F_i values		3	3	-1	1	old flows	φ_{ij}	3	1	0	3	1
capacity	$\kappa_{i_S i_F}$	∞	0	3	∞	capacity	κ_{ij}	∞	∞	0	∞	0
	$\kappa_{i_F i_S}$	0	0	∞	0		κ_{ji}	3	1	0	3	1
						max flow	Ψ_{ij}	0	∞	0	0	∞
$F_i^* = F_i + \Psi_{ij}$						$\varphi_{ij}^* = \varphi_{ij} + \Psi_{ij}$		3	∞	0	3	∞

There exists a $P(s_S \rightarrow t_F)$ path leading from the beginning of the start node into the finishing of the terminal node, along which the flow can be increased by an infinitely great value. This means that on the given network we cannot achieve a project duration smaller then $p = 10$. This project duration is smaller in fact than the project duration calculated from the crash times, the value of which $p_a = 14$ time units.
Thus we have solved the problem.

To end our paper we must mention that the maximal available project duration is not equal to the project duration calculated from the normal times. In this present problem if we consider the third activity with its crash time and all the rest with their normal time we shall get the maximal project duration. The value of this $pmax = 20$. As the basis if the algorithm is that it oves from a trivial optimal solution to an another optimal solution corresponds to a smaller project duration, and we only now the optimal solution corresponding to p_b, we can not give the optimal solution corresponding to a greater or $pmax$ project duration.

This has only theoretical importance in fact, as the solution with the smallest cost belongs to the time policy calculated from the normal times.

References

KELLEY, J. E. – WALKER, M. R. (1959) :Critical Path Planning and Scheduling *Proc. of Eastern Joint Computer Conference*, Boston 1959.

KELLEY, J. E. (1961) : Critical Path Planning and Scheduling: Mathematical Basis. 3Op. Res. 9. (1961), p. 296320,

FULKERSON, D. R. (1961): A Network Flow Computation for Project Cost Curves. *Management Sci.* (1961) p. 167178.

ROY, B. (1959): Theorie des graphes. Contribution de la theorie des graphes' l'etude de certains problemes lineaires. Comptes Rendus des Seances de l'Academie des Sciences, seance du Avril, 1959, S. 24372439.

FONDAHL, J.W. (1961) :A Non Computer Approach to the Critical Path Method for the Construction Industry, 1st Edition 1961, 2nd Edition 1962 Department of Civil Engineering, Stantford University,

IBM (1964):Programmbeschreibung fr das IBM 1440 Project Control System (PCS).