PERIODICA POLYTECHNICA SER. ARCHITECTURE VOL. 37, NOS. 1-4, PP. 27-40 (1993)

AN ALGORITHM TO SOLVE THE COST OPTIMIZATION PROBLEM THROUGH AN ACTIVITY ON ARROW TYPE NETWORK (CPM/COST PROBLEM)[1]

Miklós HAJDU and Emil KLAFSZKY

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

Received: Oct. 12, 1992

Abstract

In this paper we offer a new algorithm to the problem of cost optimization solved first by KELLEY and WALKER and later by FULKERSON. The problem is called 'Critical Path Method' CPM/cost in short after KELLEY and WALKER. We have developed our method on the basis of KELLEY and WALKER's work. Our method is simpler in our opininion and is easier to program.

Keywords: network technique, cost optimization, CPM/cost problem.

1. Introduction

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

When we developed our method we used KELLEY (1961) and FULKERSON's (1961) result. While in FULKERSON's algorithm the arrows of the network were doubled, and the maximum flow problem was repeated on this extended network, we have accomplished by using KELLEY's (1961) ideas that no arrows have to be added to the network. This increases the speed of the algorithm both in theory and practice. More than this we believe that this algorithm is simpler and easier to make its computer program than any of the kind used earlier.

In our paper we assume that the reader is at home in network flow theory such as longest path through a network, etc. We have no possibilities to discuss these in this present paper.

[1]This research is supported by OTKA F4112

2. The Model and the Solution of the Problem

Let be given a directed graph. It can have only one source and sink. Let source be s and sink be t. In the graph there cannot be a loop and more than one arrow between two nodes.

Let the set of the nodes be N, and that of the arrows A. To denote the graph the symbol $[N, \mathcal{A}]$ is used. In this network the arrows of the graph represent activities, the nodes stand for events. The event denoted by i in the network means: all the activities running into i must be finished by the time i takes place and all the activities running out of i can only start when i has taken place. If we denote the duration of an activity with τ and the occurrence place of an event with π, we immediately come to the following condition

$$\pi_j - \pi_i \geq \tau_{ij} \qquad \forall(i,j) \in \mathcal{A}. \tag{1}$$

Let be given to all τ_{ij} activity durations a lower and a higher time bound, which determine an interval in which τ_{ij} has to be. The low bound is called crash time, the upper bound is called normal time. Let's denote the crash time and normal time corresponding to (i,j) arrow a_{ij} and b_{ij}. So the next condition is:

$$a_{ij} \geq \tau_{ij} \geq b_{ij} \qquad \forall(i,j) \in \mathcal{A}. \tag{2}$$

Keeping condition (1) we can give several π_i systems on the graph to a given τ_{ij} system which satisfy the condition. The π_i system where π_i is the smallest possible for all i, but satisfies condition (1) is called minimum time policy. The π_t value corresponding to this system is denoted by p and is called the project duration of the network assuming that $\pi_s = 0$

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

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

If we regard a minimum π_i system corresponding to a given τ_{ij} system there exists a path from s to t along which condition (1) is accomplished with equilibrium. This path is called critical path. Let be given to all (i,j) normal duration a Cn_{ij} normal cost and a c_{ij} cost factor which show how much the associating cost increment is when the activity duration is increased by one day. Knowing this we can determine the so-called crash cost (Cr_{ij}) corresponding to the crash time

$$Cr_{ij} = Cn_{ij} + c_{ij}(b_{ij} - a_{ij}).$$

Further restriction to c_{ij} is that it should not be negative, that is:

$$Cr_{ij} \geq Cn_{ij} \qquad \forall(i,j) \in \mathcal{A}.$$

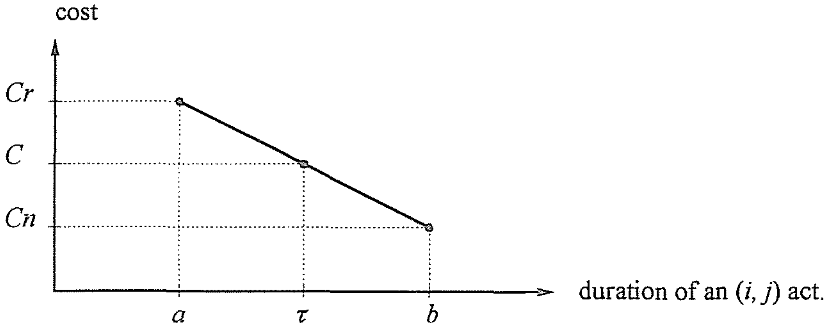


Fig. 1.

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

On a network where τ_{ij} is not determined but can move according to (2) between a lower and upper bound several kinds of project duration are available. On a network calculated with normal durations $\{b_{ij}\}$ π_t is the maximum project duration possible, it is denoted by $p_{\max}$. On a network calculated with crash durations π_t is the minimum project duration possible, let it be denoted by $p_{\min}$. In changing τ_{ij} values any project duration is available where

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

Changing the activity durations the same p project duration is available in several ways. It is evident that in case the cost slope of the activities differs different total cost will correspond to the same project duration. We are searching for the cheapest possible solution corresponding to a given project duration. We can establish the following objective function with the help of *Fig. 1*

$$\min(\sum_A \{Cn_{ij} + (b_{ij} - \tau_{ij})c_{ij}\}.$$

As Cn_{ij} and $(b_{ij}c_{ij})$ are constant, the above objective function is equivalent to the following:

$$\max(\sum_A c_{ij}\tau_{ij}). \tag{6}$$

Summarizing the points mentioned above the problem is the following:

We are to find the π_i and τ_{ij} system which satisfies the following conditions

$$
\begin{array}{lll}
\pi_j - \pi_i & \geq \tau_{ij} & \forall \quad (i,j) \in \mathcal{A} \\
a_{ij} \leq \tau_{ij} & \leq b_{ij} & \forall \quad (i,j) \in \mathcal{A} \\
\pi_s & = 0 & \\
\pi_t & = p & \\
p_{\min} \leq p & \leq p_{\max} &
\end{array}
$$

and maximizes the following objective function

$$\max\Big(\sum_A (c_{ij}\tau_{ij})\Big).$$

We call this the primal problem of the CPM/cost problem. The dual of this problem can be described as follows.

Let φ_{ij} be the flow from s point to t point the total value of which is Θ. We are to find on digraph $[N, \mathcal{A}]$ the φ_{ij} flow to which

$$\Theta p + \sum_{\substack{c_{ij} > \varphi_{ij} \\ (i,j)\in A}} (c_{ij} - \varphi_{ij}) b_{ij} - \sum_{\substack{\varphi_{ij} > c_{ij} \\ (i,j)\in A}} (\varphi_{ij} - c_{ij}) a_{ij} \longrightarrow \text{is minimal.}$$

We transform the first element of the graph of the dual problem so that the later established theorems can be easier to prove. Using the $\pi_t = p$ and $\pi_s = 0$ relations:

$$\Theta p = (\pi_s(-\Theta)) + (\pi_2 0) + \cdots + (\pi_j 0) + \cdots + (\pi_t \Theta) =$$

$$\sum_{j\in N} \{\pi_j [\sum_{\substack{i \\ (i,j)\in A}} \varphi_{ij} - \sum_{\substack{i \\ (i,j)\in A}} \varphi_{ji}]\} = \sum_{i\in N} \{\pi_j \sum_{\substack{i \\ (i,j)\in A}} \varphi_{ij}\} - \sum_{i\in N} \{\pi_i \sum_{\substack{j \\ (j,i)\in A}} \varphi_{ij}\} =$$

$$\sum_{j\in N} \{\pi_j \sum_i \sum_j \varphi_{ij}\} - \sum_{i\in N} \{\pi_i \sum_i \sum_j \varphi_{ij}\} = \sum_A (\pi_j - \pi_i)\varphi_{ij}.$$

The objective function is the following after transformation

$$\sum_A (\pi_j - \pi_i)\varphi_{ij} + \sum_{\substack{c_{ij} > \varphi_{ij} \\ (i,j)\in A}} (c_{ij} - \varphi_{ij}) b_{ij} - \sum_{\substack{\varphi_{ij} > c_{ij} \\ (i,j)\in A}} (\varphi_{ij} - c_{ij}) a_{ij} \longrightarrow \min.$$

There is a strict connection between the primal and dual problems that we established in the lemma hereby.

LEMMA: If there exist a τ and π policy corresponding to the primal problem and a φ flow, then

$$\sum_A (c_{ij}\tau_{ij}) \leq \sum_A (\pi_j - \pi_i)\varphi_{ij} + \sum_{\substack{c_{ij} > \varphi_{ij} \\ (i,j)\in A}} (c_{ij} - \varphi_{ij}) b_{ij} - \sum_{\substack{\varphi_{ij} > c_{ij} \\ (i,j)\in A}} (\varphi_{ij} - c_{ij}) a_{ij}$$

and equilibrium exists if and only if

$$\begin{array}{llll} \pi_j - \pi_i > \tau_{ij} & \text{then} & \varphi_{ij} = 0 & 1^\circ \\ b_{ij} > \tau_{ij} & \text{then} & \varphi_{ij} \geq c_{ij} & 2^\circ \\ a_{ij} < \tau_{ij} & \text{then} & \varphi_{ij} \leq c_{ij} & 3^\circ \end{array}$$

PROOF: On an arrow where $\varphi_{ij} < c_{ij}$

$$c_{ij}\tau_{ij} \leq (\pi_j - \pi_i)\varphi_{ij} + (c_{ij} - \varphi_{ij})b_{ij}$$

according to the lemma.

If we replace $(\pi_j - \pi_i)$ and b_{ij} by τ_{ij} which is smaller or equal to them the inequality will become equality so the original statement will be true. Equality exists if and only if

$$\begin{array}{llll} \varphi_{ij} > 0 & \text{then} & \pi_j - \pi_i = \tau_{ij} & 1^{\circ\circ} \\ c_{ij} > \varphi_{ij} & \text{then} & b_{ij} = \tau_{ij} & 2^{\circ\circ} \end{array}$$

On an arrow where $\varphi_{ij} < c_{ij}$

$$c_{ij}\tau_{ij} \leq (\pi_j - \pi_i)\varphi_{ij} - (\varphi_{ij} - c_{ij})a_{ij}$$

according to the lemma. If we replace $(\pi_j - \pi_i)$ by τ_{ij} which is smaller or equal to it, and a_{ij} by τ_{ij} which is greater or equal to it (this will mean subtracting a greater number than we had in the original subtraction) the inequality will become equality so the original statement will be true.

Equality holds if and only if

$$\begin{array}{llll} \varphi_{ij} > 0 & \text{then} & \pi_j - \pi_i = \tau_{ij} & 1^{\circ\circ} \\ c_{ij} < \varphi_{ij} & \text{then} & a_{ij} = \tau_{ij} & 3^{\circ\circ} \end{array}$$

Summarizing the above mentioned equality in the lemma it exists if and only if

$$\begin{array}{llll} \varphi_{ij} > 0 & \text{then} & \pi_j - \pi_i = \tau_{ij} & 1^{\circ\circ} \\ c_{ij} > \varphi_{ij} & \text{then} & b_{ij} = \tau_{ij} & 2^{\circ\circ} \\ c_{ij} < \varphi_{ij} & \text{then} & a_{ij} = \tau_{ij} & 3^{\circ\circ} \end{array}$$

Reversing this

$$\begin{array}{llll} \pi_j - \pi_i > \tau_{ij} & \text{then} & \varphi_{ij} = 0 & 1^\circ \\ b_{ij} > \tau_{ij} & \text{then} & \varphi_{ij} \geq c_{ij} & 2^\circ \\ a_{ij} < \tau_{ij} & \text{then} & \varphi_{ij} \leq c_{ij} & 3^\circ \end{array}$$

This is exactly what is stated in the lemma.

There is an important consequence to the lemma which will be presented as follows:

THEOREM: (weak form of equilibrium) If there exists a π_i and τ_{ij} policy which satisfies the primal problem with an optional p project duration and a φ_{ij} flow on the network and also the value of the primal objective function (P) is equal to (D) the objective function of the dual problem which means $P = D$, then the solutions of both problems are optimal.

PROOF: (in an indirect way) 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 project duration $(p_{\min} \leq p \leq p_{\max})$ there is a π, τ, and φ policy where the values of the objective functions are equal, that is optimal.

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

PROOF: The solution corresponding to the $p = p_{\max}$ project duration comes about trivially. Let π_i system be the time policy derived from $\tau_{ij} = b_{ij}$ values. In this case $\pi_t = p_{\max}$. Let φ_{ij} be zero on all arrows. This is an optimal solution as the values of the primal and dual objective functions are equal $\sum_A c_{ij} b_{ij}$. If there is an optimal solution for any p then we can give a p^*, π^*, τ^* and ϕ^* that satisfy the lemma that is it 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 to the maximum project duration we can give the optimal solution to all p project duration in the $p_{\min}, p_{\max}$ interval. 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 which duality conditions can be accomplished in what kind of combination. The '+' indicates that the given duality condition is accomplished, the '−'indicates that it is not.

Out of the eight possible combinations only (+++) and (++−) don't figure as these combinations cannot come about on one arrow. If all three conditions were accomplished on any arrow, it would mean, the τ_{ij} duration came between the normal time and the crash time and $\pi_j - \pi_i > \tau_{ij}$, that is the activity has slack time. In this case, however, increasing the duration of the activity by one day, we would get a better solution to the same p, consequently an arrow to which all three conditions can be accomplished cannot correspond to any optimal solution. The same argument excludes (++−) combination, too.

Depending on which conditions are accomplished on a certain arrow we can get some information on the flows passing through. This information tells us what the flows should be in case the conditions are accomplished. We have placed this information in the fourth column. In the fifth column we have classed the arrows in five groups ($\mathcal{A}1 - \mathcal{A}5$) according to which equilibrium condition is accomplished.

Equilibrium condition			Information	Group of	κ_{ij}	κ_{ji}
1°	2°	3°	on flows	act.		
+	−	+	$\varphi_{ij} = 0$	$\mathcal{A}1$	0	0
+	−	−	$\varphi_{ij} = 0$	$\mathcal{A}1$	0	0
−	+	+	$\varphi_{ij} = c_{ij}$	$\mathcal{A}2$	0	0
−	+	−	$\varphi_{ij} \geq c_{ij}$	$\mathcal{A}3$	∞	$\varphi_{ij} - c_{ij}$
−	−	+	$\varphi_{ij} \leq c_{ij}$	$\mathcal{A}4$	$c_{ij} - \varphi_{ij}$	φ_{ij}
−	−	−	$\varphi_{ij} \geq 0$	$\mathcal{A}5$	∞	φ_{ij}

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, the solution will still be optimal. With the help of data from flows information we can tell how much the flow can be increased or decreased on a certain arrow. The sixth and seventh columns show the capacities which came about as a result of this kind of argument. If we want to find maximal flow on a network with the above given capacities, we will get ψ_{ij} flow. Adding this to the original φ_{ij} flow, the new flow value on the arrows will be

$$\varphi^*_{ij} = \varphi_{ij} + \psi_{ij} \qquad \forall (i,j) \in \mathcal{A}.$$

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:

We try to find a π_i^*, τ_{ij}^* system, where $p^* < p$ to the increased φ_{ij}^* flow where the duality conditions are accomplished that is the solution remains optimal. We mean that we change τ and π values in a way that the arrow will stay in the arrow class corresponding to the flow on the arrow. The classing of certain arrows will inevitably change.

In the first step we were looking for a maximal flow. In this case there exists an (S,T) cut the arrows of which are saturated.

In the cut $\mathcal{A}1, \mathcal{A}2$ and $\mathcal{A}4$ type arrows can occur. In the cut all arrow types can occur backwards.

On the (i,j) arrows of the cut we decrease π_j potentials by a well chosen δ value.

On the (i,j) arrows going backwards in the cut we decrease the potential of i event by δ value.

The determination of δ comes about in the following way.

When the arrow is in the cut

In the case of $\mathcal{A}1$ type arrows

$$\delta_1 := \min\{\pi_j - \pi_i - b_{ij} : (i,j) \in A1,\ (i,j) \in (S,T)\}.$$

In this case the arrow will become $\mathcal{A}4$, $\mathcal{A}5$ type. In case of decreasing π_j with a $\delta < \delta_1$ value the arrow will remain $\mathcal{A}1$ type.

In the case of $\mathcal{A}2$ type arrows

$$\delta_2 := \min\{\pi_j - \pi_i - a_{ij} : (i,j) \in A2,\ (i,j) \in (S,T)\}.$$

In this case the arrow will become $\mathcal{A}3$ type. In case of decreasing π_j with $\delta < \delta_2$ value the arrow will remain $\mathcal{A}2$ type.

In the case of $\mathcal{A}4$ type arrows

$$\delta_4 := \min\{\pi_j - \pi_i - a_{ij} : (i,j) \in A4,\ (i,j) \in (S,T)\}.$$

In this case the arrow will become $\mathcal{A}3$ type. In case of decreasing π_j with $\delta < \delta_4$ value the arrow will remain $\mathcal{A}2$ type.

When the arrow goes backwards in the cut

In the case of $\mathcal{A}1$ type arrows δ_{1^c} can be of any value, the arrow remains $\mathcal{A}1$ type.

In the case of $\mathcal{A}2$ type arrows

$$\delta_{2^*} := \min\{b_j - \pi_j + \pi_i : (i,j) \in A2,\ (i,j) \in (T,S)\}.$$

In this case the arrows will become $\mathcal{A}4$ type. In case we decrease π_i in a smaller degree the arrow will remain $\mathcal{A}2$ type. A larger decrease cannot be allowed, because in that case the arrow will not take the time values according to the flow which means the solution will not be optimal.

In the case of $\mathcal{A}3$ type arrows

$$\delta_{3^*} := \min\{b_j - \pi_j + \pi_i : (i,j) \in A3,\ (i,j) \in (T,S)\}.$$

In this case the arrows will become $\mathcal{A}4$ type. In case we decrease π_i in a smaller degree the arrow will remain $\mathcal{A}2$ type. A larger decrease cannot be allowed, because in that case the arrow will not take the time values according to the flow which means the solution will not be optimal.

In the case of $\mathcal{A}4$ type arrows δ_{4^*} can be of any value, the arrow will be $\mathcal{A}1$ type.

In the case of $\mathcal{A}5$ type arrows δ_{5^*} can be of any value, the arrow will be $\mathcal{A}1$ type.

Knowing these data δ can be determined in the following way

$$\delta := \min\{\delta_1, \delta_2, \delta_4, \delta_{2^*}, \delta_{3*}\}.$$

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

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

On arrows where both intersections are in S set of nodes, or both intersections are in T set of nodes duality conditions are automatically accomplished. On arrows which are in the cut or backwards in the cut they are accomplished because on determinated δ we had the aim to continue accomplishing these conditions.

Thus p will become $p^* = p - \delta$ and an (i,j) duration can be determinated from the following formula:

$$\tau_{ij}^* = \min\{\pi_j^* - \pi_i^*; b_{ij}\} \quad \forall (i,j) \in \mathcal{A}.$$

The $\mathcal{A}4$ and $\mathcal{A}5$ type arrows which go backwards in cuts will become $\mathcal{A}1$ type arrow in case $\delta = 1$. This means that (i,j) activity which had been critical so far, that is π_j was determined by the activity, will not continue to be critical. If there is no critical path to j event δ can only be chosen to be zero at $\mathcal{A}4$ and $\mathcal{A}5$ backward going arrows in cut. In this case there is no guarantee that the algorithm will not come to a stop somewhere before it reaches $p_{\min}$ duration time in that p cannot be decreased further. The value of δ on these arrows can only be chosen optionally large if the automatism of the algorithm guarantees that there is another critical path to j.

This automatism can easily be proved. As $(i, j) \in \mathcal{A}$ arrow goes backwards in the cut, that is the backward (j, i) arrow is saturated (φ_{ij} flow passes through it) some flows (at least φ_{ij}) have to flow to j node. As flows can only be on critical arrows, critical path can go to j from other nodes not only from i that is the algorithm automatically accomplishes the expectation that an $\mathcal{A}4$ and $\mathcal{A}5$ type arrow can only go backwards in a cut if there is a critical path leading to j from another node.

Thus we have completed step two.

After all this we can go back to step one and increase the flow and then in the second step p project duration can be decreased. These steps must be alternately repeated until we reach the project duration wanted. It is advisable to calculate $p_{\max}$ project duration as well as $p_{\min}$ project duration when starting solving the problem. Thus we can check whether the project duration we give is accomplishable at all.

Thus the theorem is constructively proved.

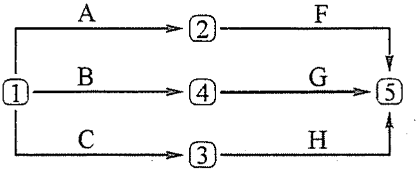
An important consequence of the theorem is presented hereby

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

PROOF: According the duality theorem there exist 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

On the given diagram we can see a network. In the table the normal and crash time values are given. There are also given the cost factors (c_{ij}) corresponding to the activities



	A	B	C	D	E	F	G	H
b_{ij}	4	7	3	5	2	10	7	2
a_{ij}	3	5	2	3	2	8	5	1
c_{ij}	10	12	5	8	0	6	14	10

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

Step 0 (The determination of $p_{\min}$ and $p_{\max}$).

Arrow	A	B	C	D	E	F	G	H
τ_{ij}	4	7	3	5	2	10	7	2

Node	1	2	3	4	5
π_i	0	4	3	9	16

$$p_{\max} = \pi_t = 16$$

Arrow	A	B	C	D	E	F	G	H
π_{ij}	3	5	2	3	2	8	5	1

Node	1	2	3	4	5
π_i	0	3	2	6	11

$$p_{\min} = \pi_t = 11$$

To start with take $\tau_{ij} = b_{ij}$ time values and the event times calculated from there and the $\varphi_{ij} = 0$ flow. This is the optimal solution corresponding to $p_{\max}$ project duration.

Step 1 (increasing the flows)

arrow	A	B	C	D	E	F	G	H
arrow classes $\mathcal{A}1 - \mathcal{A}5$	4	1	4	4	1	1	4	1
capacities κ_{ij}	10	0	5	8	0	0	14	0
κ_{ji}	0	0	0	0	0	0	0	0
flows ψ_{ij}	8	0	0	8	0	0	8	0
new flows $\varphi_{ij} + \psi_{ij}$	8	0	0	8	0	0	8	8

Step 2 (decreasing π_i, τ_{ij}, p)

arrows in cut	B	E	H	F	D
δ values	2	4	11	2	2
δ min	2				

nodes	1	2	3	4	5
old π values	0	4	3	9	16
new π values	0	4	3	7	14

arrow	A	B	C	D	E	F	G	H
new duration	4	7	3	3	2	10	7	2

The new project duration $p = 14$.

The costs have increased by 16 units that is by 8 cost units per time units.

Step 1* (increasing the flow)

arrow	A	B	C	D	E	F	G	H
arrow classes $\mathcal{A}1 - \mathcal{A}5$	4	4	4	3	1	4	4	1
capacities κ_{ij}	2	12	5	∞	0	6	6	0
κ_{ji}	8	0	0	0	0	0	8	0
flows ψ_{ij}	2	6	0	0	0	2	6	0
new flows $\varphi_{ij} + \psi_{ij}$	10	6	0	8	0	2	14	0

Step 2* (decreasing π_i, τ_{ij}, p)

arrows in cut	A	G	H	D*
δ values	1	2	9	2
δ min	1			

nodes	1	2	3	4	5
old π values	0	4	3	7	14
new π values	0	3	3	7	13

arrow	A	B	C	D	E	F	G	H
new duration	3	7	3	4	2	10	6	2

*Arrow D goes backwards in the cut.

The new project duration is $p = 13$. The costs have increased by 16 units which means 16 units per time unit.

Step 1** (increasing the flows)

arrow	A	B	C	D	E	F	G	H
arrow classes $\mathcal{A}1 - \mathcal{A}5$	3	4	4	2	1	4	2	1
capacities κ_{ij}	∞	6	5	0	0	4	0	0
κ_{ji}	0	6	0	0	0	2	0	0
flows ψ_{ij}	4	0	0	0	0	4	0	0
new flows $\varphi_{ij} + \psi_{ij}$	14	6	0	8	0	6	14	0

Step 2** (decreasing π_i, τ_{ij}, p)

arrows in cut	F	G	H
δ values	2	1	6
δ min	1		

nodes	1	2	3	4	5
old π values	0	3	3	7	13
new π values	0	3	3	7	12

arrow	A	B	C	D	E	F	G	H
new duration	3	7	3	4	2	9	5	2

The new project duration is $p = 12$.

The costs have increased by 20 units per time unit.

Step 1* (increasing the flows)

arrow	A	B	C	D	E	F	G	H
arrow classes $\mathcal{A}1 - \mathcal{A}5$	3	4	4	2	1	2	3	1
capacities κ_{ij}	∞	6	5	0	0	0	∞	0
κ_{ji}	4	6	0	0	0	0	0	0
flows ψ_{ij}	0	6	0	0	0	0	6	0
new flows $\varphi_{ij} + \psi_{ij}$	14	12	0	8	0	6	20	0

Step 2* (decreasing π_i, τ_{ij}, p)

arrows in cut	F	D	B	E	H
δ values	1	1	2	2	7
δ min	1				

nodes	1	2	3	4	5
old π values	0	3	3	7	12
new π values	0	3	3	6	11

arrow	A	B	C	D	E	F	G	H
new activity time	3	6	3	3	2	8	5	2

The new duration time $p = 11$.

The costs have increased by 26 units that is 26 money units per time unit.

As $p = p_{\min} = 11$, we have solved the problem.

4. Summary

In this paper we have given a new method to the time-cost trade off problem. Our solution is based on the theory of network flows and makes use of the duality theorem of linear programming. In theory our solution is faster than the one published by FULKERSON (1961) as it does not double the number of arrows as Fulkerson does.

Our algorithm is based on repeated execution of the maximal flow problem. In the course of the solution the flow problem has to be repeated as many times as the number of breaks are in the cost curve between the maximal project duration and the project duration given by us. As there are more and more new and better algorithms being developed in connection with maximal flow problem, the speed of the CPM/cost algorithm depends basically on the speed of the algorithm used to solve the maximal flow problem. In solving the time-cost trade off problem any existing maximal flow algorithm can be used. If somebody wants to make a quick program to the problem, it is advisable to examine their speed in practice as their theoretical speed may not come through due to the special structure of the network.

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

KELLEY, J. E. (1961): Critical Path Planning and Scheduling: Mathematical Basis. *Op. Res.* Vol. 9. pp. 296–320.

FULKERSON, D. R. (1961): A Network Flow Computation for Project Cost Curves. *Management Sci.* Vol. 7 pp. 167-178.

FORD, L. R. – FULKERSON, D. R. (1956): Maximal Flow Through A Network. *Canadian J. Math.* Vol. 8. pp. 399-404.