MŰEGYETEM 1782

CSONKA PÁL DOCTORAL SCHOOL

GENERALIZATION OF SCHEDULING AND TIME-COST TRADE-OFF PROBLEMS IN PROJECT MANAGEMENT

PhD Doctoral Dissertation

Helga Csordás

Department of Construction Technology and Management

2014

Contents

1 Introduction

1.1 Intentions of the dissertation

The aim of this dissertation is to find new methods and developments of known project models to reduce the distance between theoretic models and reality. The results of the research are dividable in two sections. The first one is the generalization of the cost data in Time-cost Trade-off problem. The aim is to make the known problem suitable for handle more technological alternatives in construction works. The other research area is directed towards the time data of the problem, it examines the possibilities of using arbitrary calendars.

1.2 Principles

1.2.1 The features of project models

The project model is represented on AOA network. It is based on graph technique. The *graph* is the set of nodes and arcs. Let the set of arcs be A and the set of nodes N. The item number of the sets are $|N| = n$ and $|A| = m$. The data of the problem can be associated to the items of the graph. In most cases real or natural values are associated to the arcs. The features of the project network in this dissertation are follows:

1. Every activity occurs exactly once in it.
2. It has got one s source node (does not have any arc into it) and one r sink node (does not have any arc out of it). In case of every $i \neq s, r$ node there exists a $P(s,r) = \{s = x_0, x_1, ..., i, ..., x_k = r\}$ path.
3. There is not arcs with same start and end nodes.
4. The prefix of values on arcs are arbitrary, which separate them to $(A^+ \cup A^0 \cup A^- = A)$ disjoint sets.
5. It is permitted to exist $H = \{i = x_0, x_1, ..., x_k, x_0 = i\}$ loop, except the loop by one arc.
6. The $[N, A^+ \cup A^0]$ subgraph is coherent and has got one source node.

1.2.2 Types of problems

Scheduling problem In scheduling the input data are the activity times and the time distances between them. So there is one time parameter for every process even activity or connection. This is denoted by τ_{ij}, $\forall ij \in A$. The aim of the problem is to find the minimal duration time. It comes from the maximal paths - minimal potentials linear programming algorithm. The result is a potential for $\forall i \in N$ nodes, which is denoted by μ_i. The constraint is:

$$\mu_j - \mu_i \geq \tau_{ij}, \forall ij \in A \tag{1}$$

It means a minimal constraint for the time distance between the start and end nodes of the given process. In case of project modelling it can arise the claim for using maximal constraints. Typical cases are overhaul or limitation the rent time of a high-value machine. This constraint requires just an opposite relation like the one before.

$$\mu_j - \mu_i \leq \tau_{ij}, \forall ij \in A \tag{2}$$

Towards the uniformity this latter constraint must be transformed. It effects that maximal constraints appear in the model with negative time parameters and the arcs directed opposite of reading.

Time-cost Trade-off It is assumed that the processes of the project can be realized in different duration. To realize a real process in minimal time is *rush time* noted by a_{ij}. Its cost level is noted by $K_{a_{ij}}$. To realize the process in minimal cost level is *normal time* noted by b_{ij}. Its cost level is $K_{b_{ij}}$. The cost levels between these extremes are determined by linear interpolation. The gradient of the line is *cost intensity* noted by c_{ij} which can be determined from the given data. $c_{ij} = \frac{K_{a_{ij}} - K_{b_{ij}}}{b_{ij} - a_{ij}}$

The connections of these parameters are demonstrated on Fig 1.

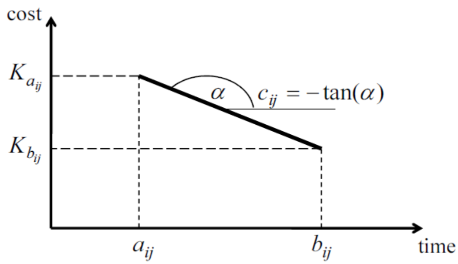

Fig. 1.: Parameters of Time-cost Trade-off problem

1.3 Applied sources

1.3.1 Engineering practice

In project scheduling there are mostly foreign examples which shows that it is possible to give exact and followable evaluation for organization. I made a survey what experiences are on Hungarian construction practice.

On the whole it can be said the largest problem is the lack of preparatory work. ... In the hurry published plans are not well-thought-out and this is the base of a fast compiled winner tender. To gain the contract it should take-on such financial and scheduling constraints which cannot include any reserves. In construction there are a lot of machine applied of which organization always has pitfalls unfortunately. In addition the most works are founded to each other. So if there is slip in any of them, it influences to the others. Nevertheless the dead line is sacrosanct. So slips must be compensated in some way. Generally it means overtime which generates extra cost. But the budget even does not have reserves, so it must cut down on other areas. Generally it causes loss of quality or technical quantity.

In scientific approach it needs to work out such flexible Time-cost Trade-off models, which follows the characteristics of projects and considers the limits in time and cost. So it should widen the theoretic Time-cost Trade-off problem with generalizations which are useful for practice.

Publication in connection with this matter:

Beruházási ütemtervek hibaforrásai, Budapest, Építőmester 2007. szeptember - október, pp.68-70.

1.3.2 Literature review

Topological order To simplify the operability of problem or accelerate the algorithm it usually arise the claim to number the nodes in topological order. It means that arcs are directed from the node lower numbered to the node higher numbered. The solution and border conditions of the problem can be found among others in papers of Ahuja et. al. [1] or Frank [12]. As they also established it is easy to see that in case of a network with loops the topological order is impossible. Unfortunately the presented typical problems may have loops in scheduling.

Let the set of arcs with positive process times be A^+, the set of arcs with negative process times be A^- and the set of residual arcs be A^0. It can be seen that loops do not exist in subgraph $[N, A^+]$. The condition of reality is that $[N, A^+ \cup A^0]$ subgraph is coherent and has one source and one sink node. These are conditions further. The arcs ij where $i < j$ are parts of topological order and create the set of A^t.

Counting path variants Path variants can be identified between two selected (x and y) items referred to both nodes or arcs From the *M adjacency matrix* it can be composed *VAR path variant matrix* by the next procedure:

$$VAR_{ij} = \begin{Bmatrix} VAR_{ij} + VAR_{ik}, \text{ if } VAR_{kj} > 0, \; i, j \neq k \\ VAR_{ij}, \text{ otherwise} \end{Bmatrix}$$

Based on this the path variants of the items are determinable:

- In case of the selected items: $var_x = var_y = VAR_{xy}$
- In case of the other items: $var_k = VAR_{xk} \cdot VAR_{ky}$ as the totality of incoming and outgoing path variants.

Features of the result:

- Because of the acyclic and topologic features in the VAR matrix not zero elements are only above the main diagonal.
- If a k item is not part of any path between the selected items, then $var_k = 0$.

Scheduling problem The problem based on a linear programming method, where a Primal and a Dual problem can be identified. The extremes of objective functions equal which is the optimum at the same time.

Primal problem Given an $[N, A]$ project network. According to a given τ_{ij} $\forall ij \in A$ system the aim is to find such $\mu \Rightarrow \mu_i$ $\forall i \in N$ potential system which fulfils the next conditions.

$$\begin{aligned} \mu_j - \mu_i &\geq \tau_{ij}, \forall ij \in A \\ \mu_s &= 0 \\ \mu_r &\rightarrow \min \end{aligned} \tag{3}$$

Dual problem Given an $[N, A]$ project network. The aim is to find such f flow system which fulfils the next conditions.

$$\begin{aligned} \sum_{is \in A} f_{is} - \sum_{sj \in A} f_{sj} &= -1 \\ \sum_{ik \in A} f_{ik} - \sum_{kj \in A} f_{kj} &= 0 \ \ \forall k \in N \setminus \{s, r\} \\ \sum_{ir \in A} f_{ir} - \sum_{rj \in A} f_{rj} &= 1 \end{aligned} \tag{4}$$

$$\sum_{ij \in A} f_{ij} \cdot \tau_{ij} \rightarrow \max$$

The definition of the project network results that $\sum_{is \in A} f_{is} = 0$, as there do not exist such arcs. The f flow system generates a unit flow from s to r, which assigns a path $P(s, r)$ where

$$\sum_{ij \in P(s,r)} \tau_{ij} \rightarrow \max$$

Loop in the network The corollary of the possibility of maximal constraints is the appearance of loops in the network. The loop rate of an arbitrary H loop is the summation of the process times in the loop, noted by ρ_H.

$$\rho_H = \sum_{ij \in H} \tau_{ij}$$

In case of negative process times it is possible that the loop rate is not positive therefore the scheduling is solvable. Loops may in connection with each other and effects to each other. Nodes are in the same Q *aggregated loop* if there is a loop which includes all the examined nodes. Nodes out of aggregated loops create W *acyclic sets*.

There are more algorithm for finding loops. Here it must to identify all the aggregated loops and nodes which are in the same aggregated loop. In 1962 Warshall [35] gave a path finding method which was utilized and modified by Vattai [34] later. At first it has to determine the sized $M = n \times n$ adjacency matrix.

$$M_{ij} = \begin{Bmatrix} 1, \text{if } ij \in A \\ 0, \text{ otherwise} \end{Bmatrix}$$

That gives V *connectivity matrix* which comes from the next procedure:

$$V_{ij} = \begin{Bmatrix} 0, \text{ if } \nexists \text{ path from } i \text{ to } j \\ 1, \text{ if } \exists\ P(i,j) \end{Bmatrix}$$

Changing process times One of the first results was published in 1972 by Klafszky [24]. Here such basic definitions were formulated like departure time, waiting time and travelling time.

Especially for project problems a model was published by Hallefjorda and Wallace [18]. According to the paper every activity in the project model has some claim of resource. The available of resource can be summarized in a work pattern which is in proportion to a basic calendar. According to this it can be assigned a progression to every activity. Different resources can have different work patterns. If more resources need to an activity, the progression will be the intersection of their work patterns.

In case of processes without resources it can be applied the basic calendar. Such process is for example the consolidation of concrete, which happens all the calendar days. So a typical basic calendar includes all the calendar days.

The τ_{ij} duration means the necessary workdays. According to the given capacity and applied technology it can be determinated the work. After defining the capacity of resource it can be determined τ_{ij} duration - which can be considered constant - as a basic value. The real or calendarized value is depending on the departure time (μ_i) of the process. This method was also used in the work of Franck et. al. [11]. The notations of them are applied in the dissertation.

In the network it is assigned a d_{ij} *calendar vector* to every ij arc, which is defined by the following unit step function:

$$d_{ij}(t) = \left\{ \begin{array}{l} 1, \text{ if the resource is available on the day } t \\ 0, \text{ otherwise} \end{array} \right\}$$

In case of any τ_{ij} process time and μ_i departure time the $\vartheta_{ij}(\mu_i)$ *calendarized process time* can be evaluated with the aid of d_{ij} calendar vector.

$$\tau_{ij} = sgn(\tau_{ij}) \sum_{t=\mu_i}^{\min\{\mu_i+\vartheta_{ij}(\mu_i)\}} d_{ij}(t) \quad (5)$$

Features of the calendarized process times are follows:

- Thanks to the definition the absolute value of calendarized process time is not lower than its basic time.

$$\begin{aligned} 0 &\leq \tau_{ij} \leq \vartheta_{ij}(\mu_i) \\ \vartheta_{ij}(\mu_i) &\leq \tau_{ij} \leq 0 \end{aligned} \quad (6)$$

- The relations correlated to each other are not change. So in case of any $\tau_{ij}^{(1)} < \tau_{ij}^{(2)}$ basic times it is true that

$$\vartheta_{ij}^{(1)}(\mu_i) < \vartheta_{ij}^{(2)}(\mu_i) \quad (7)$$

- If a process departures later, it cannot finish earlier. In other words in case of any $\varepsilon > 0$ value it is true that

$$\vartheta_{ij}(\mu_i + \varepsilon) \geq \vartheta_{ij}(\mu_i) - \varepsilon \quad (8)$$

Publication in connection with this matter:
Változó folyamatidők alkalmazása hálós modellezésben, ÉTE Építésszervezés és Építéstechnológia Konferencia 2009, Budapest

Time-cost Trade-off problem The problem is originated to the minimal cost flow problem. These flow algorithms can be founded in the work of Ahuja [1]. Solutions were published in 1969 by Klafszky [25], later in 1972 by Hajdu and Klafszky [15]. These are based on maximal flow problem but able to handle only minimal constraints. In 2004 and 2005 there were published two papers by Mál振usz [28] and [29]. Here it is possible to use maximal constraints without limitation. This later version is presented here.

Primal problem In a given $[N, A]$ digraph the aim is to find such τ and μ systems, where it is true that

$$\begin{aligned} \tau_{ij} + \mu_i - \mu_j &\leq 0 \\ -\tau_{ij} &\leq -a_{ij} \\ \tau_{ij} &\leq b_{ij} \\ \sum_{ij \in A} c_{ij} \cdot \tau_{ij} &\rightarrow \max \end{aligned} \tag{9}$$

Dual problem In a given $[N, A]$ digraph the aim is to find such f flow system, where it is true that

$$\sum_{kj \in A} f_{kj} - \sum_{ik \in A} f_{ik} = 0, \ \forall k \in N \tag{10}$$

$$\sum_{\substack{ij \in A \\ f_{ij} < c_{ij}}} (c_{ij} - f_{ij}) \cdot (b_{ij}) - \sum_{\substack{ij \in A \\ c_{ij} < f_{ij}}} (f_{ij} - c_{ij}) \cdot (a_{ij}) \rightarrow \min$$

The flow conditions are not distinguished for nodes s and r, so the flow circulates. According to this it must expand the $[N, A]$ digraph with an rs arc which should not limit the solvability of the task.

In case of maximal constraints let it be prescribe $a_{ij} \leq b_{ij} \leq 0$ and $c_{ij} = 0$ parameters.

2 Scientific results of the dissertation

2.1 Technological change in Time-cost Trade-off problem

In Time-cost Trade-off problem it must be decided the applying technology and resources forward to determine the time and cost parameters of the process. However in civil practice it is typical to exist more technological alternatives for an engineering problem. In case of more technological alternatives it must be run the known optimization algorithm for every variant. In case of more processes it must be run it for all the variation of the variants. In this case the run time of the algorithm can increase exponential. Moreover there are too many redundant counting.

At first I examined how to take in more technological alternatives in the mathematical model of the Time-cost Trade-off problem based on maximal flow algorithm.

Thesis 1. *I defined the integrated cost function and the restrictions on parameters for taking the technological change in the Time-cost Trade-off problem. I defined the term of technological change and its input data. I showed a mathematical model which is appropriate for using the known Time-cost Trade-off problem based on maximal flow algorithm.*

On the analogy of the original problem there are given $[N, A]$ and a_{ij}, b_{ij} and c_{ij} values on arcs. Let $B \in A$ be the set of arcs, where there are more - here two - a_{ij}, b_{ij} and c_{ij} parameters. So on the arcs of the set B there are more technological variants. It is evident that time parameters of them are different as this is the gain of expansion of the problem. According to this let they be the slower (l) and faster (g) technologies. Individually cost functions are check up with the one in the original problem (Fig. 1.). The relations of parameters are presented in Fig. 2. which let be constraints henceforth.

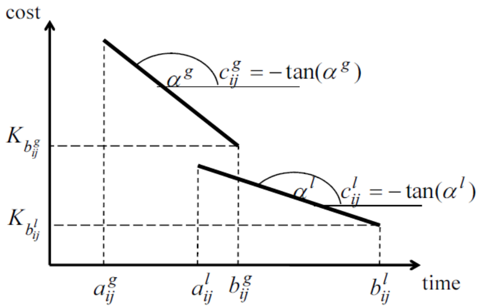

Fig. 2.: Activity with multiple parameters

$$\begin{aligned} a_{ij}^g &< a_{ij}^l; \quad b_{ij}^g < b_{ij}^l; \quad c_{ij}^g > c_{ij}^l \\ a_{ij}^l &\leq b_{ij}^g \\ K_{b_{ij}^g} &\geq K_{b_{ij}^l} + (b_{ij}^l - b_{ij}^g) \cdot c_{ij}^l \end{aligned} \tag{11}$$

In the course of integrating it needs to define the technological change which happens on $[a_{ij}^l - 1; a_{ij}^l]$ unit period. Its cost intensity can be evaluated from data of the two cost functions.

$$c_{ij}^v = \frac{\left[K_{b_{ij}^g} + \left(b_{ij}^g - (a_{ij}^l - 1)\right) \cdot c_{ij}^g\right] - \left[K_{b_{ij}^l} + \left(b_{ij}^l - a_{ij}^l\right) \cdot c_{ij}^l\right]}{a_{ij}^l - (a_{ij}^l - 1)} \tag{12}$$

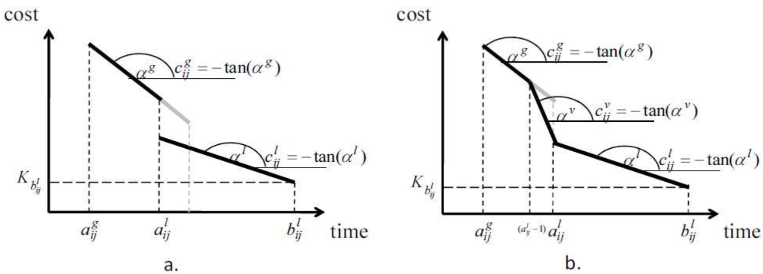

Fig. 3.: The integrated cost function

Regarding to the integrated cost function in the mathematical model there are three c_{ij} values on the arcs of the set $B \in A$ while (11) and (12) conditions are realized. Let $\forall ij \in B$ cut in pieces with two further nodes (x and y) which are inserted between the nodes i and j.and composed $ix, xy, yj \in A$ arc group. These "inner" nodes are not direct connected to the other arcs of the network. The new arcs separate the integrated cost function (Fig. 3.b.) according to the gradients and create three arcs with one c_{ij} value. It is shown in Fig 4. the separating of the integrated cost function and the costs of the normal times.

It is called *technological change* when in case of an $xy \in B$ arc the τ_{xy} value changes from 1 to 0.

$K_{b_{ix}} = 0$, $K_{b_{xy}} = 0$, $K_{b_{yj}} = K_{b^l_{ij}}$. Thus the problem is reconducted to the same data structures as in the original model. In case of maximal constraints it is needless to separate as here $c_{ij} = 0$ condition is prescribed.

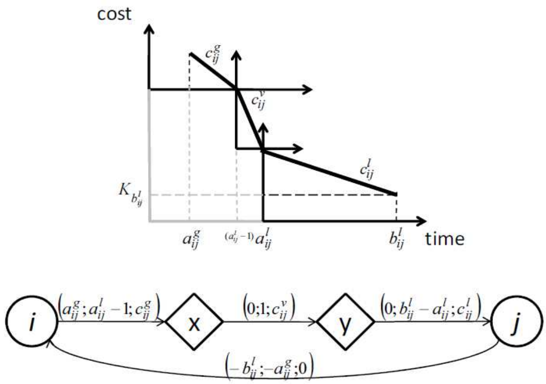

Fig. 4.: network model of technological change

The constraints and objective functions correspond with the elements of the original problem with an addition. The durations in an arc group in the set B must be changed in the proper order as according to their interpretations they are in close connection with each other.

The consequence of the method is using inconstant cost levels. Because of this different result sets are valid in iterations. This does not guarantee that the solution is in global optimum as this point can be not determined squarely in variable result sets. Therefore the only demanded feature is the equality of the primal and dual objective functions which guarantees the optimal solution in convex result sets. This is insurable by complying with the optimal criteria.

Further there are presented three methods.

At first I examined how apply the mathematical model and I worked out the flow arranging method.

2.1.1 Flow arranging method

For using the maximal flow algorithm it must to adjust the model so that changing of durations are happened in the proper order. The values of the residual network originated in cost intensities. So the solution is modification of cost intensities. Let the cost intensities on arcs in the set B be defined as follows:

$$c_{ix} = \left\{ \begin{array}{l} \max\left\{c_{ij}^{g}; c_{ij}^{v}+1\right\}, \text{ if } \tau_{xy}=1 \\ c_{ij}^{g}, \text{ otherwise} \end{array} \right\}$$

$$c_{xy} = \left\{ \begin{array}{l} c_{ij}^{v}, \text{ if } \tau_{xy}=1 \\ \min\left\{c_{ij}^{v}; c_{ij}^{g}-1\right\}, \text{ otherwise} \end{array} \right\}$$

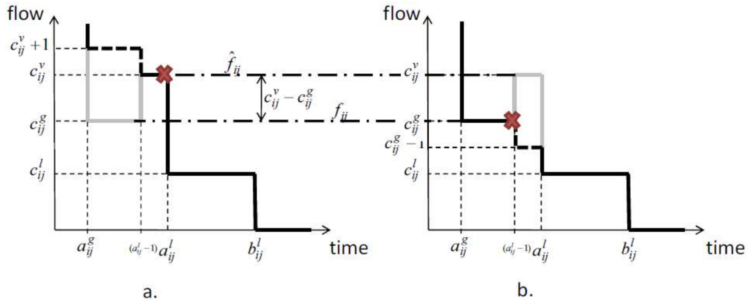

Fig. 5.: Flow levels at technological change

As it is shown in Fig. 5. in case of technological change the flow must be reduced with $c_{xy} - c_{ix}$. It means it have to find flow from r to s.

The problem only comes up if $c_{ij}^{g} < c_{ij}^{v}$. In this case after technological change on the arcs ix and xy the residual values become not allowable. Otherwise the integrated cost function in the physical model is convex in every point, so the linear programming problem can be solved trivially.

Publication in connection with this matter:
A Network Flow Algorithm For Time-Cost Trade-off With Technological Decision, 7th International Conference Organization, Technology And Management In Construction, Zadar, Croatia, 2006. (Co. Mályusz Levente) (ISBN 953-96245-6-8)

2.1.2 Flow diverting method

The mathematical model of the arcs in set B is shown in Fig. 6.

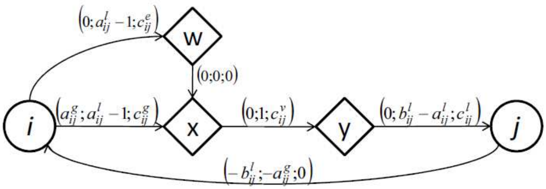

Fig. 6.: Mathematical model of flow diverting method

To avoid parallel arcs it needs to insert an additional node (w). The iw arc is inserted to divert the flow. Its cost intensity also cannot be constant.

$$c_{ij}^e = \max\{0; c_{ij}^v - c_{ij}^g + \tau_{xy}\}$$

At technological change the residual value is reduced but it is needless to find flow backward. If the arc iw is full, by a local flow arranging unit flow can be directed to the arc ix. Thus the residual value of arc with cost intensity $c_{iw} = c_{ij}^v - c_{ij}^g$ is zero which is an allowable value.

The virtual cost generated by "unnecessary" flows on the arcs belonged to the set B can be evaluated.

$$\sum_{iw \in B} (b_{iw} - \tau_{iw}) \cdot c_{iw} = \sum_{\substack{ij \in B \\ c_{ij}^v > c_{ij}^g}} \max\left\{0; \left[(a_{ij}^l - 1) - \tau_{ij}\right] \cdot (c_{ij}^v - c_{ij}^g)\right\} \tag{13}$$

Publications in connection with this matter:

Beruházások erőforrásainak optimális kiválasztása a költségtervezési feladat alapján, 12. Projektmenedzsment Fórum, Budapest, 2009.

Optimal Selection of Recourses in Projects Based on the Classical Time - Cost Trade – off , Hungary, 2010., Periodica Polytechnica Social and Management Sciences , 2009. 17/1 pp. 47-55.

2.1.3 Independent flows method

The basic of the method is an observation that the maximal flow algorithm generates allowable solutions in every iteration. So the solutions in every iteration are optimal. If every optimal solution is considered as a first optimal solution, then they can be handled independently. The residual values of the independent flows are in Table 1.

class	*opt.criteria*	μ *system*	z *system*	
I	*only* 1*st*	$b_{ij} < \mu_j - \mu_i$	$z_{ij} = 0$	$z_{ji} = 0$
II	2*nd and* 3*th*	$a_{ij} < \mu_j - \mu_i < b_{ij}$	$z_{ij} = c_{ij}$	$z_{ji} = 0$
III	*only* 2*nd*	$a_{ij} = \mu_j - \mu_i$	$z_{ij} = \infty$	$z_{ji} = 0$
IV	*only* 3*th*	$\mu_j - \mu_i = b_{ij}$	$z_{ij} = c_{ij}$	$z_{ji} = 0$
V	*not anyone*	$a_{ij} = \mu_j - \mu_i = b_{ij}$	$z_{ij} = \infty$	$z_{ji} = 0$

Table 1.: Arc classification according to optimality criteria in case of independent flows

As the earlier flows are cancelled it can be never resulted not allowable residual network. So it is needless to find flows back. Only it must be guaranteed that the technological change happens before reduction the duration of faster technology. So let the next modification be used.

$$c_{ix} = \tau_{xy} \cdot c_{xy} + c_{ix}$$

Publication in connection with this matter:
Activities With Multi-Parameters In Time-Cost Trade-Off, Hungary, 2011. Pollack Periodica, 2011. Vol. 6, No. 2, pp. 37–48.

Thesis 2. *I worked out three algorithms to built the mathematical model of the integrated cost function in the known Time-cost Trade-off problem based on maximal flow algorithm.*

- *In case of the flow arranging method I proved that after technological change of a given process there exists a new flow-system met the optimal criteria while the parameters of the given process are unchanged.*
- *In case of the flow diverting method the equilibrium consumption is performed in every iteration. The assistant items built in the model modify the cost of the scheduling, but I showed a formula which gives the exact rate of this deflection.*
- *In case of the independent flows method the solution is allowable in every iteration. I showed the events when the solution does not perform the equilibrium consumption. I proved that in case of non performing the equilibrium consumption the algorithm gets back in the set of solution performing the consumption in the next iterations.*

I compared the three algorithms. At worst the flow arranging method can have got longer run time like the original one because the continuous searching flow backward. The flow diverting method and the independent flows method eliminate this step, so they have shorter run time squarely. By increasing the number of the variants, the run time of the original solution increases exponential, while it is only polynomial in case of independent flows method.

At the flow diverting method the dummy cost is appeared and has to be solved. The independent flows method brakes away significantly from the LP problem based on maximal flow algorithm. Nevertheless the efficiency of these algorithms is much more favourable.

2.2 Calendarization

The matter of calendarization is that for processes in scheduling in case of a given necessary worktime (τ_{ij}), departure time (μ_i) and a calendar vector (d_{ij}) as the work pattern it can be determi- nated the calendarized process times ($\vartheta_{ij}(\mu_i)$) which are inconstant. The problem is defined in time T which is the maximal acceptable project duration. The aim is calendarization of Time-cost Trade-off problem of which part is the scheduling. For applying the project model in general it is need to use arbitrary calendars and maximal constraints. Because of this there may grow up such loops (H) in scheduling, of which loop rate (ρ_H) has got changeable prefix.

2.2.1 Scheduling

Let p be the *number of possible paths* $P_i(s,r), i = 1..p$.

Let it be defined L $p \times n$ sized matrix as the matrix of $P_i(s,j), j \in P_i(s,r)$ *lengths of paths.*

$$\begin{aligned} L[i,j] &\geq \sum_{\substack{xy \in P_i(s,j) \\ P_i(s,j) \in P_i(s,r)}} \vartheta_{xy}(L[i,x]) \\ i &= 1...n, j = 1...p \end{aligned} \tag{14}$$

Generally the equality is realized, but it is not surely in case of loops.

If a loop H_k is part of a path $P_i(s,r)$ then it can be given $x \in N_{H_k}$ as the firstly achieved point in the loop along the path. This is the departure time of counting round in the loop. If the loop is counted round according to (14), then getting back to the start point it is got a new (check) value for the point x. The difference of the two values on x is the loop rate. The loop rate is a feature of the loop. It is known in the "constant" scheduling problem, if the loop rate is positive, the scheduling does not have got finite solution. But in case of calendarized process times the loop rate is changeable. If the current loop rate is positive, it can be counted round the loop again from the check value which comes from the loop rate. This iteration can be continued while the check value is not larger than the start value. These potentials are the lengths of paths in the H_k loop.

Thesis 3. *I proved that in scheduling with applying arbitrary calendars and maximal constraints the shown iteration method gives the first allowable solution for the L_{ij} lengths of paths. Further I proved that the loop rate getting from this iteration method is zero.*

Let it named *critical loop* such a loop H_k where the loop rate is zero.

The *waiting time* noted by ω_{ij} for calendarized process can be defined as the different between the number of $d_{ij}(t) = 1$ calendar days between the potentials of start and end nodes and the basic time.

$$\omega_{ij} = \left(\sum_{t=\mu_i}^{\mu_j} d_{ij}(t) \right) - \tau_{ij}, \forall ij \in A$$

Let such $P_k(s,r)$ path where $\omega_{ij} = 0, \forall ij \in P_k(s,r)$ be called *calendarized critical path.*

Thesis 4. *I showed that based on the maximal path - minimal potential linear programming problem it can be determined the primal - dual pair for scheduling with applying arbitrary calendars and maximal constraints. I showed that the solution can be the follows:*

1. *There exists finite solution. In this case the solution can be two types:*
   (a) *There exists one or more $P_k(s,r)$ calendarized critical path, which accomplishes that*
   $$\mu_r = \max\{L_{kr} \mid k = 1, ..., p\} = \sum_{ij \in P_k(s,r)} \vartheta_{ij}(\mu_i)$$
   (b) *There exists one or more H_k critical loop, which effects "split" in scheduling. Here $\mu_j - \mu_i > \vartheta_{ij}(\mu_i)$, $\forall ij \in A \mid i \notin N_{H_k}, j \in N_{H_k}$. Because of this*
   $$\begin{aligned} L_{ir} &> \sum_{xy \in P_i(s,r)} \vartheta_{xy}(L_{ix}) \\ j &= 1, ..., p \end{aligned}$$
2. *There does not exist finite solution, which means the overrun of T project duration. It may come from the iteration counting in a loop, or just a path variant where $ij \in P(s,r)$, $\vartheta_{ij}(\mu_i) = \infty$.*

I showed two algorithm for the calendarized scheduling problem. The first one follows closely the algorithm given for constant process times. I named it **traditional method**. Through an example I showed the disadvantages of this solution. Namely, because of the possibility of changing the sign of the loop rate there may be numerous redundant counting in every iteration. This is eliminated with the **loop finder method**. By comparing the two algorithms the different between the necessary number of actions is considerable, which proportional with the number of set of loops and the necessary number of iterations in them.

Publications in connection with this matter:

Longest Path Problem in Networks with Loops and Time Dependent Edge Lengths, 8th International Conference Organization, Technology And Management In Construction, Umag, Croatia, 2008. (ISBN 953-96245-8-4)

Scheduling in Networks with Time Dependent Arc Lengths Based on a Loop Finder Algorithm, 2012., Croatia, Organization, Technology & Management in Construction: An International Journal, Vol. 4, No. 2, pp. 512-519. (ISSN: 1847-5450, EISSN: 1847-6228)

2.2.2 Time-cost Trade-off problem

After analyzing the scheduling problem I examined how should apply calendar in the Time-cost Trade-off problem. At first I tried to transform a logistic problem by Cai et. al. [5]. Unfortunately the solution is not proper for applying nor negative process times neither loops. I proved that with counter-examples.

Publication in connection with this matter:
Calendarization in Time-Cost Trade-off Based on a Transit Problem 10th International Conference Organization, Technology And Management In Construction, Sibenik, Croatia, 2011. (ISBN 978-953-7686-01-7)

According to these cognitions it needs to work out an absolutely new method for applying calendars without restrictions in the project model.

I worked out a heuristic method for calendarization the Time-cost Trade-off in case of applying arbitrary calendars and maximal constraints. The algorithm generates such μ and ϑ systems of which features correspond with the features of the algorithm based on maximal flow - minimal cut problem which gives optimal solution proved.

Thesis 5. *Based on the scheduling problem and the experiences from the transformation of a calendarized logistic problem I worked out a heuristic method for calendarization of Time-cost Trade-off problem in case of arbitrary calendars and maximal constraints generating loops. I defined the terms time-effectivity and cost-effectivity. I proved the number of iterations at most* $\sum_{i=1}^{m^t}(b_i - a_i)$, *where* $m^t = |A^t|$.

The worked out algorithm is different from the solution based on maximal flow algorithm even in basic steps. The algorithm based on cost-effectivity monitoring start with an all probability not optimal $\tau \Rightarrow \tau_{ij} = a_{ij}, \forall ij \in A$ system.

Let it be called *time-effectivity* and noted by $\lambda_{ij}(\tau_{ij}, \mu_i)$ the number of calendar days which need to increase the process time with one work day according to the departure time. Let it be $\tau_{ij} = \sum_{t=\mu_i}^{t_1} d_{ij}(t)$ and $\tau_{ij} + 1 = \sum_{t=\mu_i}^{t_2} d_{ij}(t)$. In this case $\lambda_{ij}(\tau_{ij}, \mu_i) = \frac{1}{t_2 - t_1}$.

The *cost-effectivity* of a given arc ij - noted by ce_{ij} - is in direct proportion to the cost intensity (c_{ij}) and the actual time effectivity ($\lambda_{ij}(\tau_{ij}, \mu_i)$) of the arc and in inverse proportion to the number of path variants along the arc (var_{ij}), so

$$ce_{ij} = \frac{c_{ij} \cdot \lambda_{ij}(\tau_{ij}, \mu_i)}{var_{ij}}$$

In case of a scheduling the cost-effectivities can be determined on all arcs. Choosing the maximal value of them causes the largest cost decreasing while the decreasing of global float is minimal.

External float is the different between the actual project duration (μ_r) and the maximal acceptable project duration (T). *Internal float* is the unapplied time on the non critical paths correlated to the length of critical paths. Because of the global approach the basic of the floats only can be the basic calendar which generally contains all calendar days.

The algorithm The flow chart of the algorithm is in Fig. 7.

Main ideas of the algorithm:

- There must exist at least one calendarized critical path in the accepted solutions. Otherwise the modifications in the previous iteration come cancelled. This means that the "split" because of critical loop is not allowable, which is a total shutdown of the project in real.
- Firstly it must be eliminated the internal floats, because it effects lower and lower cost levels to a given project duration.
- If internal floats are run out to a given project duration, the solution can be considered optimal.

The algorithm does not increase the project duration while find the scheduling with lowest cost belonged to the given project duration. This feature guarantees the optimal solution also in Time-cost Trade-off problem with constant process times. Beyond mathematical techniques I encode the algorithm in Scilab 5.3.0. For confirmation the solution I tested examples with constant process times. The results equal the ones from algorithm based on maximal flow algorithm.

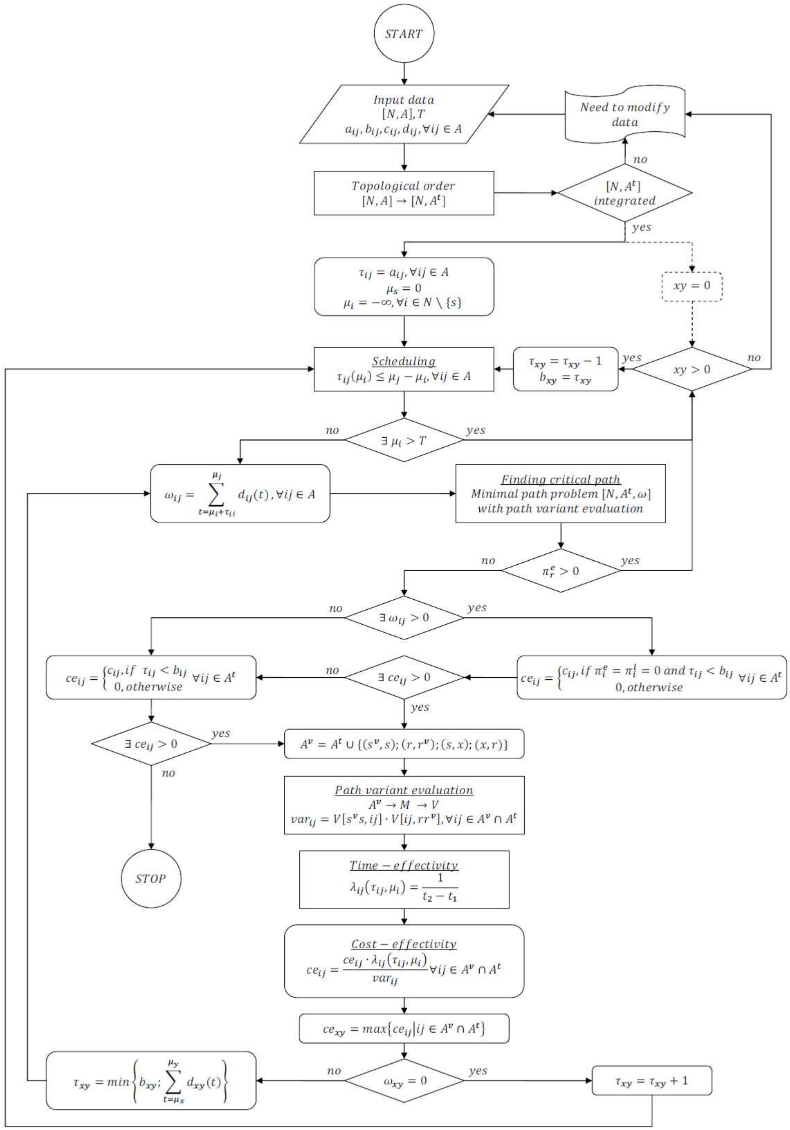

Fig. 7.: Flow chart of the algorithm Time-cost Trade-off based on cost effectivity monitoring

3 Further researching directions

Calendarization of Time-cost Trade-off problem is a great leap forward using the model in practice. Of course it is even not the mapping of the entire reality.

1. Confirmation of the parameters of the heuristic method can be achieved by further test running.

2. Other conditions can be built into the model such as many previous results and also the technological change.

3. Other researching theme can be the generalization of the worked out calendarized Time-cost Trade-off problem. It is worth to examine the operability of "split" because of the critical loop.

    - The ideal size of the allowable "split".
    - Comparison of possible methods or their applying together.

Long-term target can be utilization the worked out model in practice which can effect at least a new project management software.

References

[1] Ahuja, R.K., Magnati, T.L., Orlin, J.B., 1993., *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, NJ, pp. 164-165.

[2] Ahuja, V., Thiruvengadam, V., 2004., *Project scheduling and monitoring: current research status*, Construction Innovation, Vol. 4., pp. 19–31,

[3] Bellman, R. 1958., *On a Routing Problem*, Quarterly of Applied Mathematics, 16(1), pp. 87-90.

[4] Bérubé, J.F., Potuin, J.Y., Vaucher, J., 2006., *Time Dependent Shortest Path Through a Fixed Sequence of Nodes: Application to a Travel Planning Problem,* Computers and Operation Research, Vol. 33., pp. 1838-1856.

[5] Cai, X., Sha, D., Wong, C.K., 2007., *Time-varying network optimization*, Springer, pp. 21-24.

[6] Cheng, C.E. , Ding, Q., Lin, B.M.T., 2004., *A concise survey of scheduling with time-dependent processing times*, European Journal of Operation Research, Vol.152., pp. 1-13.

[7] Christodoulou, S., 2009., *Construction imitating ants: Resource-unconstrained scheduling with artificial ants*, Automation in Construction, Vol.18. pp. 285-293.

[8] Dean, B. C., 2004., *Algorithms for Minimum-Cost Paths in Time-Dependent Networks with Waiting Policies,* Networks, Vol. 44., Iss. 1., pp. 41 - 46.

[9] Dijkstra, E. W., 1959., *A Note on Two Problems in Connexion With Graphs*, Numerische Mathematik, Vol.1., pp. 269–271.

[10] Floyd, R.W., 1962., *Algorithm 97: Shortest path* Communications of the ACM, Vol. 5., No.6., pp. 345.

[11] Franck, B., Neumann, K., Schwindt, C., 2001., *Project scheduling with calendars* OR Spektrum, Vol 23., pp. 325-334.

[12] Frank A., 2008., *Operációkutatás*, Oktatási segédanyag, ELTE TTK Operációkutatási Tanszék (http://www.cs.elte.hu/~frank/jegyzet/opkut/ulin.2008.pdf)

[13] Fulkerson, R. D.,1961., *A network flow computation for project cost curves*, Management Science Vol. 2., No. 2. January, pp. 167-168.

[14] Hajdu M., 1993., *An algorithm for solving the cost optimization problem in precedence diagramming method*, Periodica Polytechnica ser. Civil Engineering, Vol. 37., No. 3., pp. 231-247.

[15] Hajdu M., Klafszky E., 1993., *An algorithm to solve the cost optimization problem through an activity on arrow type network (CPM/cost problem)*, Periodica Polytechnica ser. Architecture, Vol. 37, No. 1-4., pp. 27-40.

[16] Hajdu M., 1996, *PDM Time Cost Trade Off: Activities Are Splittable or Non-Spittable*, Optimization, Vol. 38., pp. 155-171.

[17] Hajdu, M., Mályusz, L., 1996., *A minimális és maximális átfutási idő probléma megoldása speciális esetben,* Közúti Közlekedés- és Mélyépítéstudományi Szemle Vol 3., pp. 133-137.

[18] Hallefjorda, Å. , W. Wallace, S., 1998, *Work patterns in project scheduling*, Annals of Operations Research Vol.82., pp. 1–81.

[19] Hamacher, H.W., Tjandra, S.A., 2002., *Earliest Arrival Flow Model with Time Dependent Capacity for Solving Evacuation Problems,* Pedestrian and Evacuation Dynamics, pp. 267-276.

[20] Jordán T., 2007., *Ütemezés*, Oktatási segédanyag, ELTE, Operációkutatási Tanszék (http://www.cs.elte.hu/~jordan/utemezes/index.html)

[21] Karp, R.M., Orlin, J.B., 1981., *Parametric shortest path algorithms with an application to cyclic staffing*, Discrete Appl. Math., Vol.3., pp. 37-45.

[22] Kelley, J.E., 1959., *Critical Path Planning and Scheduling: Mathematical Basis*, Operation Research, Vol. 9., No3.

[23] Kelley, J.E., Walker, M.R., 1959., *Critical Path Planning and Scheduling*, Proc. the Eastern Joint Computer Conference, Boston

[24] Klafszky, E., 1972., *Determination of Shortest Path in a Network with Time-Dependent Edge-Lengths*, Math. Operationsforsch. u. Statist., Vol 3., pp. 255-257.

[25] Klafszky E., 1969., *Hálózati folyamok*, Budapest

[26] Komáromi É., 2005., *Operációkutatás No.2., Lineáris Programozás*, Oktatási segédanyag, Budapesti Közgazdaságtudományi és Államigazgatási Egyetem, Operációkutatás Tanszék (http://gazdasz2.atw.hu/linearis_programozas_jegyz_nemme.pdf)

[27] Levner, E., Kats, V., 1998., *A parametric critical path problem and an application for cyclic scheduling*, Discrete Appl. Math., Vol. 87., pp. 149-158.

[28] Mályusz L., 2004., *A költségtervezési Time-cost trade-off feladat általánosítása és megoldása*, Alkalmazott Matematikai Lapok, Vol. 21., pp. 365-377.

[29] Mályusz L., 2005., *Monoton Növekedõ költségfüggvényû tevékenységek alkalmazása a költségtervezési Time-cost trade-off feladatban*, Alkalmazott Matematikai Lapok, Vol.22., pp. 199-213.

[30] Mályusz L., *Döntéstámogató módszerek*, Oktatási segédanyag, Bu- dapesti Műszaki és Gazdaságtudományi Egyetem, Építéskivitelezési Tanszék

[31] Moore, E.F., 1957., *The Shortest Path Through a Maze*, Proceedings of an International Symposium on the Theory of Switching (Cambridge, Massachusetts, 2-5 April, 1957), Harvard University Press, Cambridge, pp. 285-292.

[32] Nagy T., 2009., *Hálózati folyamok,* Oktatási segédanyag, Kelet-Magyarországi Informatika Tan- anyag Tárház, Miskolci Egyetem (http://www.uni-miskolc.hu/~matente/2012_tav/HALOZATI_FOLYAMOK.html #d5e222)

[33] Orda, A., Rom, R., 1990., *Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length,* Journal of the Association for Computing Machinery, Vol. 37., No. 3., July, pp. 607-625.

[34] Vattai Z., 1993., *Branch & Bound technika alkalmazása épitőipari sorolási feladatok megoldására*, Budapesti Műszaki Egyetem, doktori értekezés

[35] Warshall, S., 1962., *A Theorem on Boolean Matrices*, Journal of the ACM , Vol. 9. , Iss. 1., pp. 11-12.