5th International/11th Construction Specialty Conference
5e International/11e Conférence spécialisée sur la construction

Vancouver, British Columbia
June 8 to June 10, 2015 / *8 juin au 10 juin 2015*

ONE RELATION TO RULE THEM ALL: THE POINT-TO-POINT PRECEDENCE RELATION THAT SUBSTITUTES THE EXISTING ONES

Miklos Hajdu[1,2,3]
[1] Ybl Miklos Faculty of Architecture and Civil Engineering, Szent István University, Hungary
[2] Faculty of Architecture, Budapest University of Technology and Economics, Hungary
[3] miklos.hajdu63@gmail.com

Abstract: Precedence Diagram Method (PDM) has gained the widest acceptance in the scheduling practice in the last decades due to its modeling flexibility over other existing techniques, and to the relative simplicity of its mathematical background. The four basic precedence relationships have been serving planners for more than half a century. However, even this model is not flexible enough; proper modeling of overlapped activities seems to be a never-ending debate. Different practical and theoretical solutions have been proposed during the years for better modeling overlapped activities. The most promising among them is the development of a new type of relation that can connect any two arbitrary points of the related activities. These relations can be called point-to-point relations. Different authors in various ways have proposed similar solutions. To the best of our knowledge, the literature on the mathematical model of PDM using this new relation is lacking. Main results of the paper are: 1) standardized discussion of the different approaches to point-to-point relations; 2) proper mathematical model of PDM with point-to-point relations; 3) introduction of the algorithm that can handle point-to-point relations with both minimal and maximal lag to define the earliest and latest feasible time policy.

1 INTRODUCTION AND LITERATURE REVIEW

Widely used network techniques are more than half a century old. The results of Fondahl, (Fondahl 1961), Roy (Roy 1959), (Roy 1960), IBM (IBM 1964) and many others have led to the present form of the Precedence Diagram Method, the prevailing network technique of our times. PDM has hardly changed during the decades in spite of the critiques it has received about its modeling capabilities. Proper modeling of overlapping activities seems to be a never-ending debate when traditional precedence relations are used, (Douglas et al. 2006) because traditional endpoint relations are simply not suitable for describing this kind of logic. Different solutions have been proposed using the traditional precedence relations; but fragmentation of activities and developments based on this idea (Tarek & Menesi 2010) seem to be the best theoretical solution despite the arising practical problems, namely the multiplication of the number of activities and precedence relations. Probably the fragmentation technique has given the idea of connecting the inner points of the activities, which will be discussed in this paper. These point-to-point relations connecting the internal points of the activities seem to be theoretically more suitable for modeling overlapping activities – especially if continuous activities are assumed - as multiple relations are allowed between the activities. To the best of our knowledge four partly parallel works regarding point-to-point relations can be found: Kim (Kim 2010, 2012) calls his new relations bee-line relations and the graphical representation Bee-line Diagram (BDM), while Francis and Miresco (Francis & Mireco 2000, 2002) call their new relations temporal functions and they call their graphical representation method

chronographic approach. Plotnick (Plotnick 2004) calls his method Relationship Diagramming method (RDM) using the term of 'event' for the internal points. Ponce de Leon (Ponce de Leon 2010) uses the term Graphical Diagramming Method (GDM) and connected internal points are called embedded nodes. Despite the differences in terminology and definitions, e.g. bee-line and RDM relation does not allow a lag between the connected inner points, maximal lags are defined only in the work of Francis and Miresco, the concept behind all these works is the same. All these improvements regarding the relationships can be seen as a new type of precedence relation that can substitute all traditional precedence relations, as it will be shown later. The goal of this paper is twofold: firstly, to remedy a common shortcoming of these works (authors have failed to present the mathematical model); secondly, to show that traditional precedence relationships can be derived from the general point-to-point relations discussed in this paper. A proper mathematical model and the algorithm will be introduced using standardized technical terminology. The new point-to-point relation can be seen as a generalization of traditional precedence relations. It will be shown that the existing traditional precedence relations are special cases of the point-to-point relations; they connect the endpoints of the activities instead of the internal points.

2 THE MATHEMATICAL MODEL

2.1 Notations

Let a directed acyclic graph be given with one start (*s*) and one finish node (*f*). Let **N** = {1,2,…i…j…n} stand for the set of nodes also called activities. **A** will define the set of arcs, also called precedence relations. The 'super' relation defined later can have minimal or maximal lags, therefore $\mathbf{A}_{min}$ and $\mathbf{A}_{max}$ subsets are introduced for differentiating relations with minimal and maximal lags. In the algorithm, relations with maximal lags will be transformed into relations with minimal lags. In this case **A*** denotes the set of relations . An activity *i* is defined by its start and finish points P^i_s, P^i_F, shortly S^i or F^i, or by any of the two aforementioned points and its duration d^i. Additional points of the activities can also be defined. P^i_k stands for the k^{th} point of activity *i*. The relative place of the k^{th} point of activity *i* is defined by the time span (t^i_k) from the start point of activity *i* (P^i_S). A relation can be defined between any l^{th} internal point of activity *j* and any k^{th} internal point of activity *i* by defining the time that must elapse between the two points ($z^{i\,j}_{k\,l}$). Therefore this 'super' precedence relation can be defined either by the points and the lag as (P_{ik};P^j_l, $z^{i\,j}_{k\,l}$) or by the relative positions of the points and the lag (t^i_k; t^j_l, $z^{i\,j}_{k\,l}$). Explanation can be seen in Fig. 1. Lags can be defined using production volumes as well.

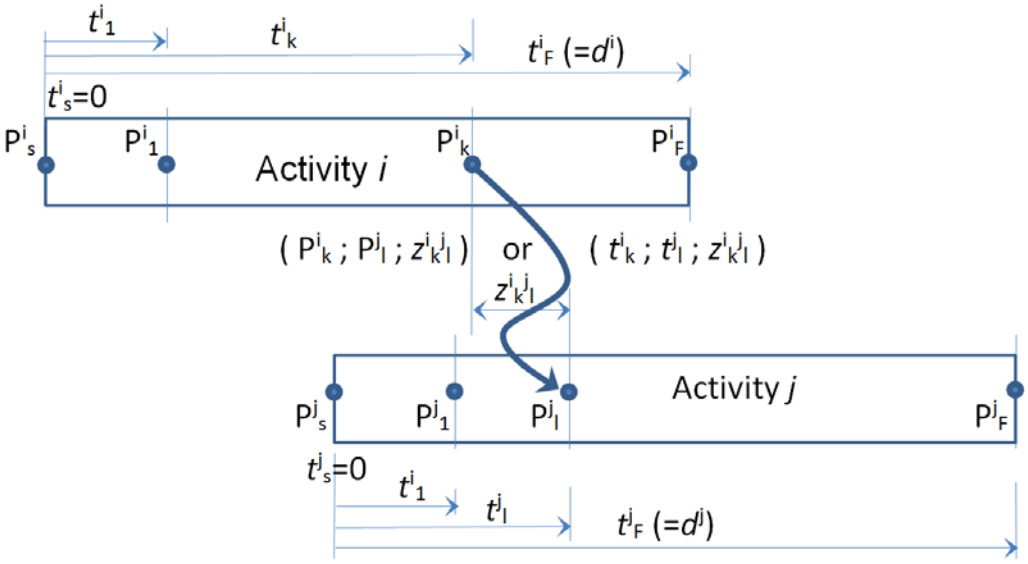

Figure 1: Explanation of notations and the the 'super' precedence relation. (t^i_k ; t^j_l; $z^{i\,j}_{k\,l}$)

Let the time when a point is accomplished be called point or event time and denoted by T^i_k. This way T^i_S stands for the start, and T^i_F stands for the finish of activity *i*. Table 1 shows how the traditional relations and bee-line relations can be derived from the new 'super' relation. Transformation of temporal functions (Francis & Miresco 2002) and RDM relations is obvious so this is not presented here. If instead of time, the names of the points of the activities are used for describing the relation, then even the same notation can be used. (e.g. instead of SS100 days (S; S; 100 days) can be used.) Based on the above notations the following model can be defined.

Table 1: 'Super' relation can be used instead of traditional and bee-line precedence relations

	Known precedence relations	Equivalent point-to-point relation
PDM	Start-to-Start z^{ij}	(0 ; 0; $z^i_S{}^j_S$) or (S; S; $z^i_S{}^j_S$)
	Finish-to-Start z^{ij}	(d^i ; 0; $z^i_F{}^j_S$) or (F; S; $z^i_F{}^j_S$)
	Finish-to-Finish z^{ij}	(d^i ; d^j; $z^i_F{}^j_S$) or (F; F; $z^i_F{}^j_F$)
	Start-to-Finish z^{ij}	(0 ; d^j; $z^i_S{}^j_S$) or (S; F; $z^i_S{}^j_F$)
BDM	(t^i_k, t^j_l)	(t^i_k; t^j_l ; 0)

2.2 The model

The first two conditions tell that all precedence relations must be satisfied. [1] describes the precedence relations with minimal lags, while [2] describes the precedence relations with maximal lags.

[1] $T^j_l - T^i_k \geq z^i_k{}^j_l \qquad \forall\ (P^i_k;P^j_l)\ \in \mathbf{A}_{min}$

[2] $T^j_l - T^i_k \leq z^i_k{}^j_l \qquad \forall\ (P^i_k;P^j_l)\ \in \mathbf{A}_{max}$

By definition $T^i_k = T^i_S + t^i_k$ and $T^j_l = T^j_S + t^j_l$ therefore [1] and [2] can be modified as:

[1*] $T^j_S - T^i_S \geq z^i_k{}^j_l - t^j_l + t^i_k \qquad \forall\ (P^i_k;P^j_l)\ \in \mathbf{A}_{min}$

[2*] $T^j_S - T^i_S \leq z^i_k{}^j_l - t^j_l + t^i_k \qquad \forall\ (P^i_k;P^j_l)\ \in \mathbf{A}_{max}$

The finish of the activities can be calculated according to [3]. Activities are assumed to be continuous [4]. Let's set the start of the project to zero. [5]

[3] $T^i_S + d^i = T^i_F \qquad \forall\ i\ \in \mathbf{N}$

[4] $T^i_k - t^i_k = T^i_S \qquad \forall\ i\ \in \mathbf{N}$ and $\exists\ P^i_k$ (P^i_k exist)

[5] $T^s_S = 0$

The T policy that satisfies [1*], [2*], [3], [4], and [5] is called a feasible time policy. An infinite number of feasible time policies exist, but the objective of the model is to find that/those time policy/policies where the project duration is the minimum, that is

[6] $T^f_F - T^s_S \rightarrow \min$ that is $T^f_F - 0 \rightarrow \min$ that is $T^f_F \rightarrow \min$

This model is an LP model, so any LP solver can be used for solving it, furthermore, based on the simplistic structure of this LP problem different efficient primal dual algorithms can be developed. The solution shown below is based on the modification of the simplistic and widely used CPM/PDM time analysis. This approach is probably the easiest to digest for planning engineers.

3 ALGORITHMS

3.1 Point-to-point relations with minimal lag

The goal of the algorithm is to find the optimal time policy, the earliest and the latest out of the existing – sometimes millions of – optimal time policies. The earliest optimal time policy is denoted by E^i_S and E^i_F. The latest optimal policy is denoted by L^i_S and L^i_F. The applied algorithm has two phases. The result of the first phase is the earliest optimal time policy, while the result of the second phase is the latest optimal time policy.

Let's suppose for the sake of simplicity that only relations with minimal lags are allowed in the network. In this case, the earliest start and finish of an activity *j* can only be calculated, if the earliest start dates for all its predecessors are known. As all precedence relations must be satisfied, the early start of a given *j* can be defined by the maximum of the shifts caused by the preceding relations of activity *j*, that is:

[7] $E^j_S = \max \{ E^i_S + t^i_k + z^{ij}_{kl} - t^j_l \quad \forall (P^i_k;P^j_l) \in \mathbf{A}_{min} \}$

To start, an activity with known predecessors has to be found. In the beginning, only the start activity satisfies this condition: all of its predecessors are known because it does not have any. After these introductory thoughts, the steps of the first phase, that is the steps aiming to find the earliest time policy, can be summarized as follows:

Step 1

Let $E^i_S = -\infty$ and $E^i_F = -\infty \ \forall i \in \mathbf{N}$; Let $E^S_S = 0$ Let g:=1

Step 2

REPEAT

g:=g+1

Choose an activity *j* from the unknowns ($E^i_S = -\infty$) with known predecessors only.

IF there is no such activity then GO TO Step 3

$E^j_S = \max \{ E^i_S + t^i_k + z^{ij}_{kl} - t^j_l \quad \forall (P^i_k;P^j_l) \in \mathbf{A}_{min} \}$; $E^j_F = E^j_S + d^j$

UNTIL *g=n*

Step 3

IF g<n

THEN

STOP (There is a loop in the network.)

ELSE

$p = E^f_F$ (Project duration is the same as the early finish of the finish activity.)

During the backward pass, the latest optimal time policy will be defined. It is completed by working from the terminal activity to the initial activity in reverse direction of the arrows. It is based on the observation that the late activity times of an activity can only be calculated, if these dates are known for all of its successors. As all successor relations must be satisfied, the late start of a given *i* can be defined by the maximum of the shifts caused by the succeeding relations of activity *i*, that is:

[8] $L^i_S = \min \{ L^j_S + t^j_k - z^{ij}_{kl} - t^i_l \quad \forall (P^i_k;P^j_l) \in \mathbf{A}_{min} \}$

The rules of the backward pass can be summarized as follows:

Step 1

Let $L^i_S = \infty$ and $L^i_F = \infty \ \forall i \in \mathbf{N}$; Let $L^{i=F}_S = p - d^f$; Let g:=1

Step 2

REPEAT

g:=g+1

Choose an activity *i* from the unknowns ($L^i_S = \infty$) with known successors only.

$L^i_S = \min \{ L^j_S + t^j_k - z^{ij}_{kl} - t^i_l \quad \forall (P^i_k;P^j_l) \in \mathbf{A}_{min} \}$; $L^i_F = L^i_S + d^i$

UNTIL *g=n*

(Note: Loop detection is not necessary during the backward pass.)

3.2 Point-to-point relations with mixed lags

Calculations with mixed lags require more computational steps. Maximal relations have to be transformed into minimal relations first. Comparing conditions [1] and [2], it can be seen that the difference between a relation with minimal or maximal lags lies in the direction of the operand. Transforming a relation with maximal lag into a relation with minimal lag requires a simple multiplication by -1.

[2] $T^j_l - T^i_k \le z^{i\,j}_{k\,l}$ $\quad \forall\ (P^i_k; P^j_l) \in \mathbf{A}_{max}$ / *(-1)

[9] $T^i_k - T^j_l \le -z^{j\,i}_{l\,k}$ $\quad \forall\ (P^i_k; P^j_l) \in \mathbf{A}_{max}$

This is nothing else but a relation from *j* to *i* with a negative minimal lag (see Fig. 2).

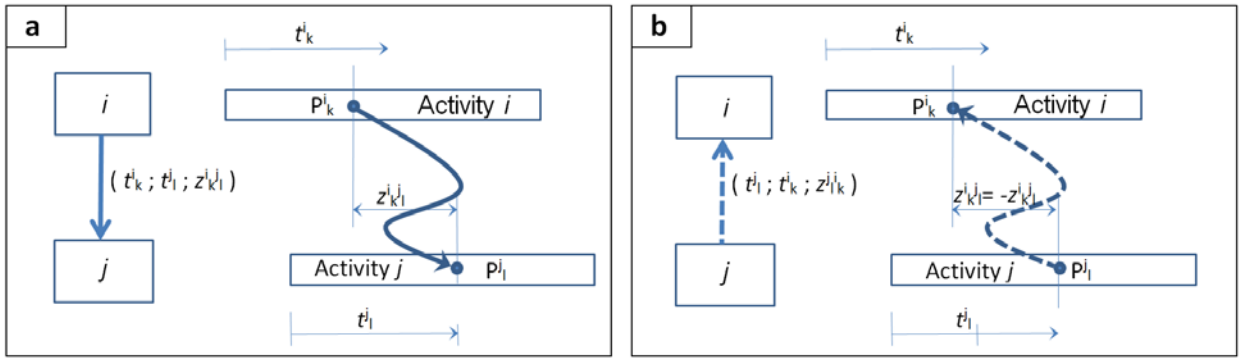

Figure 2: Point-to-point relation with maximal lag a) and its minimal equivalent b)

Traditional precedence relations with maximal lags, and their transformed equivalent minimal lags can be found in Table 2.

Table 2: Traditional precedence relations and their equivalent minimal versions.

Traditional precedence relations with maximal lag	Equivalent point-to-point relation with maximal lag	Transformed equivalent precedence relations with minimal lag*	Transformed equivalent point-to-point relation with minimal lag*
maxSSz	max(0; 0; z)	SS-z	(0; 0; -z)
maxFSz	max(d^i; 0; z)	SF-z	(0, d^j; -z)
maxFFz	max(d^i; d^j; z)	FF-z	(d^i; d^j; -z)
maxSFz	max(0, d^j; z)	FS-z	(d^i; 0; -z)

* Transformed equivalents go in the opposite direction

However, converting relations with maximal lags into their equivalent minimal relations can result in so-called transformation loops. In Fig. 3 it can be seen that there is a loop between B and C.

The simple time analysis presented in 3.1 cannot be used in case of loops. One can easily check it by selecting the activities. E.g. during the forward pass activity A has to be chosen first. After that none of the activities can be selected, B has two predecessors but only A is known and C is not; C has two predecessors but only A is known and B is not, so the algorithm will stop here.
In case of loops, different algorithms exist to find the longest path. We will use the modified version of the algorithm developed by Bellman (Bellman, 1958) and Ford (1956) for finding the shortest path between any two points of a cyclic graph. The algorithm is based on the idea that during the forward pass all activities are calculated using the dates of their predecessors even if those have not taken their final dates yet. When all activities have been calculated this way, it has to be checked whether there have been changes in activity dates or not. If the answer is yes, then the entire calculation must be repeated again and again, until we come to the results we had in the previous iteration. In every iteration, at least

one activity takes its final value, so after maximum *n* iterations we get the results. Usually much fewer iterations are necessary. If the n^{th} iteration still brings changes, then the network cannot be solved due to the maximal relations that probably impose logic on the network, which contradicts the minimal or other maximal lags. E.g. imagine that *B* can start minimum five days after the finish of A (F;S; 5) point-to-point relation but another relation describes that B should start maximum 4 days after the finish of *A* (F;S;max4).

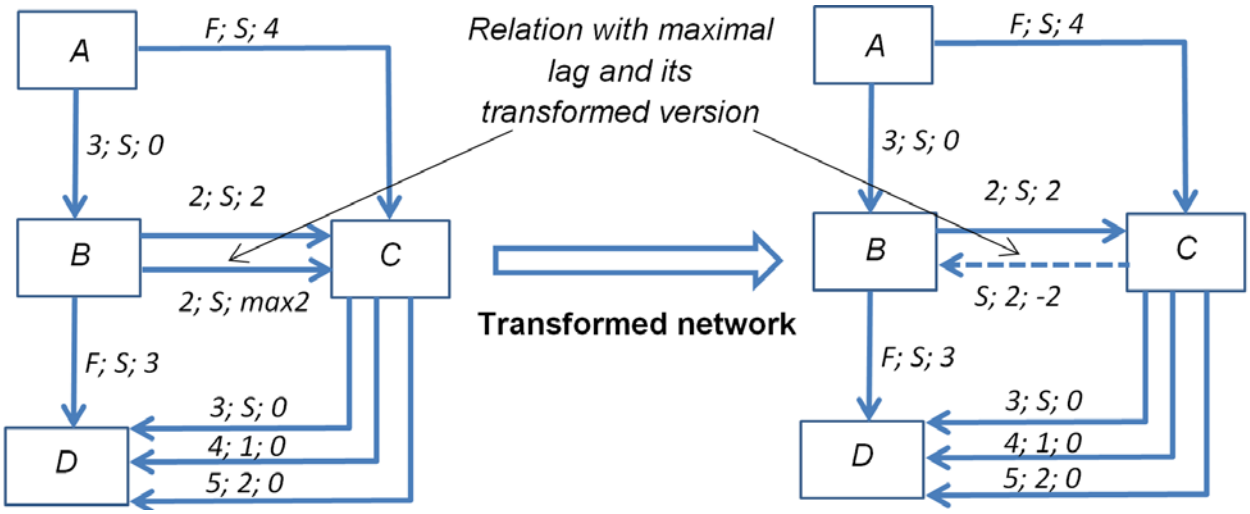

Figure 3: Transformation of relations with maximal lag into their minimal equivalent can result in loops.

This definite contradiction cannot be solved. In this case, the value of the transformation loop will be positive, and the result for the project duration will increase by at least this value in every iteration even after the n^{th} iteration. The steps below summarize the forward pass:

Step 1

Let $E^i_S = -\infty$ and $OLD\_E^i_S = -\infty$ $\forall i \in \mathbf{N}$; Let $E^{j=S}_S=0$ and $OLD\_E^{j=S}_S=0$; Let *h*=0; Let *No_of_Iter*=0

Step 2

REPEAT

There_were_changes:=FALSE; No_of_Iter:= No_of_Iter+1

REPEAT

h:=h+1

Select any *j* activity that has not been selected in this iteration yet

$E^j_S = \max \{ OLD\_E^j_s; (E^i_S + t^i_k + z^{i\ j}_{k\ l} - t^j_l \quad \forall (P^i_k;P^j_l) \in \mathbf{A^*}) \}$; $E^j_F=E^j_S+d^j$

IF $E^j_S > OLD\_E^j_S$ THEN *There_were_changes:=TRUE*

UNTIL *h=n*

Let $OLD\_E^i_S=E^i_s$ $\forall i \in \mathbf{N}$

UNTIL *No_of_iter>n* or *There_were_changes:=FALSE*

Step 3

IF No_of_Iter>*n* THEN There is no solution. (There is a loop with positive value.)

IF *There_were_changes:=FALSE* THEN we arrived to a feasible optimal time policy (All activity dates remained unchanged after two iterations.)

The rules of backward pass can be summarized as follows:

Step 1

Let $L^i_S = -\infty$ and $OLD\_L^i_S = -\infty$ $\forall i \in \mathbf{N}$; Let $L^{j=S}_S=E^{j=s}_S$ and $OLD\_L^{j=S}_S= L^{j=S}_S$; Let *h=n*;

Step 2

REPEAT

There_were_changes:=FALSE;

REPEAT

Select any *i* activity that has not been selected in this iteration yet
L^i_S = min { OLD_L^i_S; ($L^j_S + t^j_k - z^{ij}_{kl} - t^i_l$ ∀ $(P^j_k;P^j_l)$ ∈ **A***) }; $L^i_F=L^i_S+d^i$
IF L^i_S> OLD_L^i_S THEN *There_were_changes:=TRUE*
h:=h-1
UNTIL *h=0*
Let OLD_L^i_S=L^i_s ∀ *i* ∈ **N**
UNTIL *There_were_changes:=FALSE*

Notes to the algorithm:

- Loop detection was done during the forward pass, so there is no need for that during the backward pass.
- Any order of activities can be used during the algorithm, which can largely modify the number of iterations. Here we used the ascending order of activities during the forward pass and the descending order during the backward pass. In the optimal case, the first iteration presents the results and the second iteration will validate this, in the pessimistic case, *n+1* iterations are necessary.
- Following the optimal order of the forward pass will be the worst during the backward pass, and vice versa.

4 SAMPLE PROJECT

4.1 Sample project: only minimal lags are allowed

A small sample project is shown in Fig. 4 a) consisting only of relations with minimal lags. Results can be seen in Figure 4b), calculations can be tracked below.

Forward pass:

Only activity A can be selected. E^A_S=0; $E^A_F=E^A_S+d^A$=0+6=6

Only activity *B* can be selected. E^B_S = {($E^A_S + t^A_k + z^{AB}_{60} - t^B_0$) }={(0+3+0-0)}=3; $E^B_F=E^B_S+d^B$=3+6=9

Only activity C can be selected. E^C_S=max{($E^A_S+t^A_6+z^{AC}_{60}-t^C_0$);($E^B_S+t^B_2+z^{BC}_{20}-t^C_0$)}={(0+6+4-0);(3+2+2-0)}= 10; $E^C_F=E^C_S+d^C$=15

Only activity D can be selected. E^D_S=max{($E^B_S+t^B_6+z^{BD}_{60}-t^D_0$);($E^C_S+t^C_3+z^{CD}_{30}-t^D_0$);($E^C_S+t^C_4+z^{CD}_{41}-t^D_1$); ($E^C_S+t^C_5+z^{CD}_{52}-t^D_2$)}={(3+6+3-0);(10+3+0–0);(10+4+0–1);(10+5+0–2)}={12;13;13;13}=13; $E^D_F=E^D_S+d^D$=17

Early dates for all activities have been calculated, the forward pass is finished. (See Fig. 4)

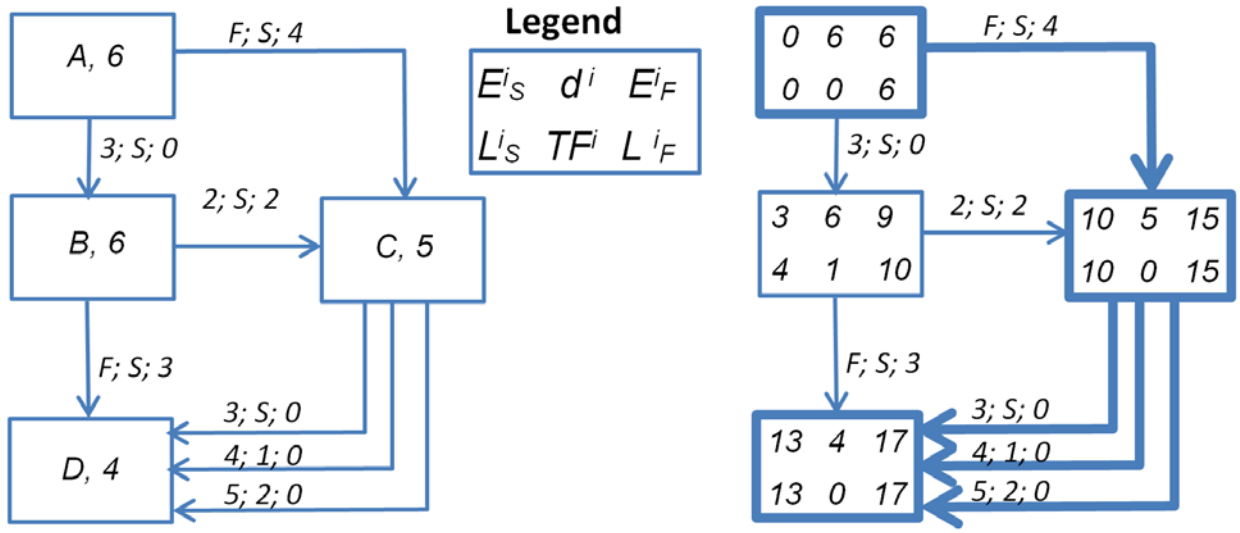

Figure 4a) Sample project with minimal lags.

Figure 4b) Results of the calculations.

Backward pass:

Only activity *D* can be selected. $L^D_F=17$; $L^D_S=L^D_F-d^D=17-4=13$

Only activity C can be selected. $L^C_S=\min\{(L^D_S+t^D_2-z^C_5{}^D_2-t^C_5);(L^D_S+t^D_1-z^C_4{}^D_1-t^C_4);(L^D_S+t^D_0-z^C_3{}^D_0-t^C_3)\}=\min\{(13+2-0-5);(13+1-0-4);(13+0-0-3)\}=\min\{(10;10;10)\}=10$ $L^C_F=L^C_S+d^C=10+5=15$

Only activity *B* can be selected. $L^B_S=\min\{(L^D_S+t^D_0-z^C_6{}^D_0-t^C_6);(L^C_S+t^C_0-z^B_2{}^C_0-t^B_2)=\min\{(13+0-3-6);(10+0-2-2)\}=\min\{(4;6)\}=4$ $L^B_F=L^B_S+d^B=4+6=10$

Only activity *A* can be selected. $L^A_S=\min\{(L^B_S+t^B_0-z^A_3{}^B_0-t^A_3);(L^C_S+t^C_0-z^A_6{}^C_0-t^A_6)=\min\{(4+0-0-3);(10+0-4-6)\}=\min\{(1;0)\}=0$ $L^A_F=L^A_S+d^A=0+6=6$

Late dates for all activities have been calculated, the backward pass is finished. (See Fig. 4)

4.2 Sample project: both minimal and maximal lags are allowed

A small sample project is shown on Fig. 5a) consisting of relations with minimal and maximal lags. The network with the transformed maximal relation and with the results is shown in Fig. 5b). Due to the transformation loop, the iterative algorithm has to be used.

Any order of the activities can be used. Here we use the A;B;D;C order for the forward pass. The calculations can be followed in Figure 6. Boxes of those activities that have been changed during the iteration are filled with grey.

Iteration #1

$E^A_S=\max\{OLD\_E^A_s\}=0$; $E^A_F=E^A_S+d^A=0+6=6$

$E^B_S=\max\{OLD\_E^B_S;(E^A_S+t^A_3+z^A_3{}^B_0-t^B_0);(E^C_S+t^C_0+z^C_0{}^B_2-t^B_2)\}=\{-\infty;(0+3+0-0\};(-\infty+0-2-2)\}=3$
$E^B_F=E^B_S+d^B=9$

$E^D_S=\max\{OLD\_E^D_S;(E^B_S+t^B_6+z^B_6{}^D_0-t^D_0);(E^C_S+t^C_3+z^C_3{}^D_0-t^D_0);(E^C_S+t^C_4+z^C_4{}^D_1-t^D_1)(E^C_S+t^C_5+z^C_5{}^D_2-t^D_2)\}=\{-\infty;(3+6+3-0\};(-\infty+3+0-0);(-\infty+4+0-1);(-\infty+5+0-2)\}=12$ $E^D_F=E^D_S+d^D=16$

$E^C_S=\max\{OLD\_E^C_S;(E^B_S+t^B_2+z^B_2{}^C_0-t^C_0);(E^A_S+t^A_6+z^A_6{}^C_0-t^C_0)=\{-\infty;(3+2+2-0\};(0+6+4-0)\}=10$
$E^C_F=E^C_S+d^C=15$

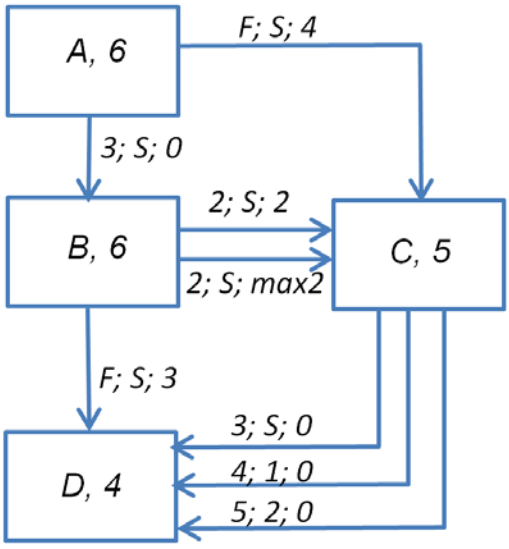

Figure 5a) Sample project with mixed lags.

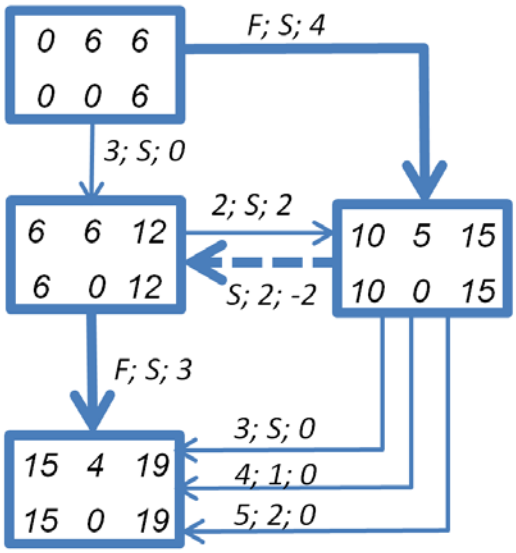

Figure 5b) Transformed network with the results

Iteration #2

E^A_S= max { $OLD\_E^A_s$ }=0; $E^A_F=E^A_S+d^A$=0+6=6

E^B_S=max{$OLD\_E^B_S$;($E^A_S+t^A_3+z^A_3{}^B_0-t^B_0$);($E^C_S+t^C_0+z^C_0{}^B_2-t^B_2$)}={3;(0+3+0-0};(10+0-2-2)}=6
$E^B_F=E^B_S+d^B$=12

E^D_S=max{$OLD\_E^D_S$;($E^B_S+t^B_6+z^B_6{}^D_0-t^D_0$);($E^C_S+t^C_3+z^C_3{}^D_0-t^D_0$);($E^C_S+t^C_4+z^C_4{}^D_1-t^D_1$)($E^C_S+t^C_5+z^C_5{}^D_2-t^D_2$)}={12; (6+6+3-0}; (10+3+0-0); (10+4+0-1); (10+5+0-2)}=15 $E^D_F=E^D_S+d^D$=19

E^C_S=max{$OLD\_E^C_S$;($E^B_S+t^B_2+z^B_2{}^C_0-t^C_0$);($E^A_S+t^A_6+z^A_6{}^C_0-t^C_0$)={10;(6+2+2-0};(0+6+4-0)}=10
$E^C_F=E^C_S+d^C$=15

Iteration #3

E^A_S= max { $OLD\_E^A_s$ }=0; $E^A_F=E^A_S+d^A$=0+6=6

E^B_S=max{$OLD\_E^B_S$;($E^A_S+t^A_3+z^A_3{}^B_0-t^B_0$);($E^C_S+t^C_0+z^C_0{}^B_2-t^B_2$)}={3;(0+3+0-0};(10+0-2-2)}=6
$E^B_F=E^B_S+d^B$=12

E^D_S=max{$OLD\_E^D_S$;($E^B_S+t^B_6+z^B_6{}^D_0-t^D_0$);($E^C_S+t^C_3+z^C_3{}^D_0-t^D_0$);($E^C_S+t^C_4+z^C_4{}^D_1-t^D_1$)($E^C_S+t^C_5+z^C_5{}^D_2-t^D_2$)}={12; (6+6+3-0}; (10+3+0-0); (10+4+0-1); (10+5+0-2)}=15 $E^D_F=E^D_S+d^D$=19

E^C_S=max{$OLD\_E^C_S$;($E^B_S+t^B_2+z^B_2{}^C_0-t^C_0$);($E^A_S+t^A_6+z^A_6{}^C_0-t^C_0$)={10;(6+2+2-0};(0+6+4-0)}=10
$E^C_F=E^C_S+d^C$=15

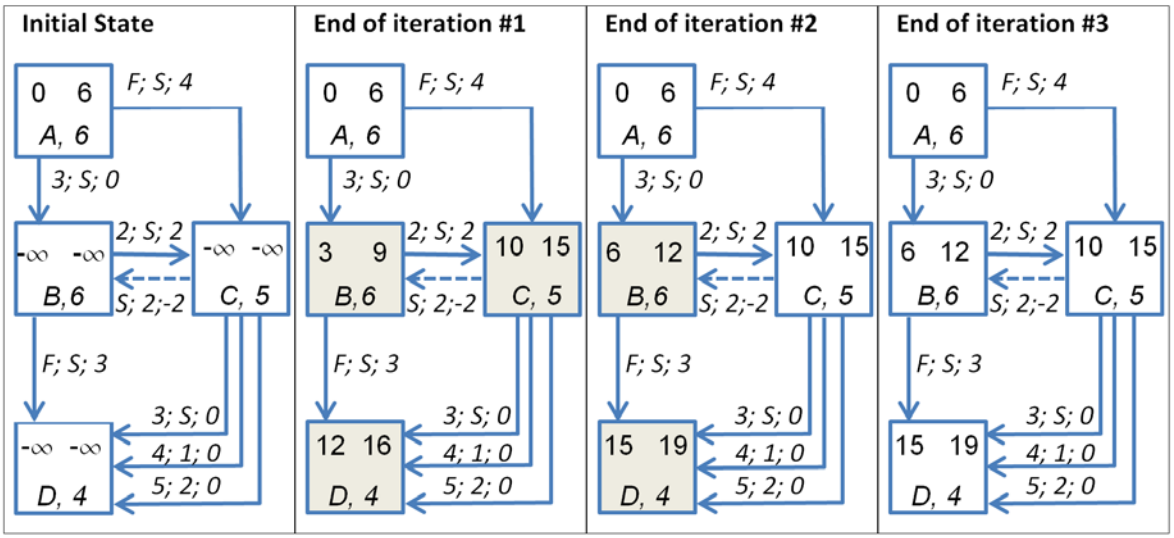

Figure 6: Forward pass computation.

No activity dates have changed in the course of iteration #3; therefore the forward pass is finished. For the backward pass the D; B; C; A sequence is selected. Calculations can be followed below.

Iteration #1

L^D_S= min{$OLD\_L^D_S$; (nil) }=15; $L^D_F=L^D_S+d^D$=19

L^B_S=min{$OLD\_L^B_S$;($L^D_S+t^D_0-z^B_6{}^D_0-t^B_6$);($L^C_S+t^C_0-z^B_2{}^C_0-t^B_2$)}=min{∞;(15+0-3-6);(∞+0-2-2)}=6
$L^B_F=L^B_S+t^B$=12

L^C_S=min{$OLD\_L^C_S$;($L^B_S+t^B_2-z^C_0{}^B_2-t^C_0$);($L^D_S+t^D_0-z^B_3{}^D_0-t^B_3$); ($L^D_S+t^D_1-z^B_4{}^D_2-t^B_4$) ($L^D_S+t^D_2-z^B_5{}^D_2-t^B_5$)}=min{∞;(6+2-(-2)-0);(15+0-0-3);(15+1-0-4);(15+2-0-5)}=10 $L^C_F=L^C_S+t^C$=15

L^A_S=min{$OLD\_L^A_S$;($L^C_S+t^C_0-z^A_6{}^C_0-t^A_6$);($L^B_S+t^B_0-z^A_3{}^B_0-t^A_3$)}=min{∞;(10+0-4-6);(6+0-0-3)}=0 $L^A_F=L^A_S+t^A$=6

Iteration #2

L^D_S= min{OLD_L^D_S; (nil) }=15; L^D_F=L^D_S+d^D=19

L^B_S=min{OLD_L^B_S;(L^D_S+t^D_0-$z^B_6{}^D_0$-t^B_6);(L^C_S+t^C_0-$z^B_2{}^C_0$-t^B_2)}=min{6;(15+0-3-6);(10+0-2-2)}=6
L^B_F=L^B_S+t^B=12

L^C_S=min{OLD_L^C_S;(L^B_S+t^B_2-$z^C_0{}^B_2$-t^C_0);(L^D_S+t^D_0-$z^B_3{}^D_0$-t^B_3); (L^D_S+t^D_1-$z^B_4{}^D_2$-t^B_4) (L^D_S+t^D_2-$z^B_5{}^D_2$-t^B_5)}=min{10;(6+2-(-2)-0);(15+0-0-3);(15+1-0-4);(15+2-0-5)}=10 L^C_F=L^C_S+t^C=15

L^A_S=min{OLD_L^A_S;(L^C_S+t^C_0-$z^A_6{}^C_0$-t^A_6);(L^B_S+t^B_0-$z^A_3{}^B_0$-t^A_3)}=min{0;(10+0-4-6);(6+0-0-3)}=0 L^A_F=L^A_S+t^A=6

Activity dates have not changed during the iteration. Calculations are finished. Results of the backward pass (and the forward's as well) can be seen in Fig. 5b.

5 DISCUSSIONS AND FURTHER RESEARCH

A new 'super' precedence relationship based on the results of Francis & Mireco and Kim has been discussed in the paper. It has been shown that the traditional precedence relations (SS, SF, FS and FF with either minimal or maximal lags) could be derived from the new relation. It has also been shown that different results of parallel works e.g. bee-line relations can also be derived from this new 'super' relation; therefore claiming these results as a new technique is a wrong approach. This new relation forms a connection between two arbitrary points of two activities; therefore the name point-to-point relation fully describes the nature of this new relation. Following this logic, the traditional precedence relations, which could be called endpoint relations, form a subset of point-to-point relations as they connect the endpoints of the activities. Point-to-point relations affect the very fundaments of network techniques, therefore all definitions, generalizations, problems based on the 'old' precedence relations must be checked and modified accordingly, if necessary, including the definitions and calculations of floats, the definition of the critical path, the classification of critical activities (Weist 1981) (Hajdu 1996) (Walls & Lino 2001), the algorithms for resource optimization etc. To our best knowledge, this work has not been done yet.

References

Bellman, R.E. 1958 On a Routing Problem. *Quarterly of Applied Mathematics* **16** 1959. 87-90

Douglas, E.E.; Calvery T. T. McDonald, D.F.; Winter, R.M. 2006. The Great Negative Lag Debate. 2006 *AACE International Transactions* PS 02.01.- PS 02.07

Fondahl, J.W 1961. A non-computer approach to the critical path method for the construction industry. *Technical Report #9* The Construction Institute, Department of Civil Engineering, Stanford University, Stanford California

Ford, L.R. 1956. Network Flow Theory. *The Rand Corporation* 1956.

Francis, A., Miresco, E.T. 2000. Decision Support for Project Management Using a Chronographic Approach. *Proceedings of the 2nd International Conference on Decision Making in Urban and Civil Engineering,* 2000 Lyon, France, 845-856.

Francis, A., Miresco, E.T. 2002. Decision Support for Project Management Using a Chronographic Approach. *Journal of Decision Systems, Special issue JDS-DM in UCE: Decision Making in Urban and Civil Engineering*, **11**(3-4): 383-404.

Hajdu, M. 1996. Network Scheduling Techniques For Construction Project Management. *Kluwer Academic Publishers*, ISBN 0-7923-4309-3

Hegazy, T & Menesi, W. 2010 Critical Path Segments Scheduling Technique. *Journal of Construction Engineering and Management*, **136**(10): 1078-1085

IBM 1964. Users Manual for IBM 1440 Project Control System (PCS). 1964

Kim, S. 2010. Advanced Networking Technique *Kimoondang,* South Korea 2010

Kim, S. 2012. CPM Schedule Summarizing Function of the Beeline Diagramming Method. *Journal of Asian Architecture and Building Engineering*, **11**(2) November 2012; 367-374

Plotnick FL, 2004 Introduction to Modified Sequence Logic, *Conference Proceedings, PMICOS (first annual) Conference*, April 25, 2004, Montreal, Canada

Ponce de Leon, G. 2008 Graphical Planning method. *PMICOS Annual Conference, Chicago, IL, 2008*

Roy, G.B., 1959 Théorie des Graphes: Contribution de la théorie des graphes á l1 étude de certains problémes linéaries, *Comptes rendus des Séances de l1 Acedémie des Sciences*. séence du Avril 1959, s2437-2449, 1959

Roy, G.B. 1960. Contribution de la théorie des graphes à l'étude de certains problems d'ordonnancement", *Comptes rendus de la 2ème conférence internationale sur la recherché opérationnelle*, Aix-en-Provence, English Universities Press, Londres 171-185.

Valls, V and Lino, 2001. Criticality Analysis in Activity-on-Node Networks with Minimal Time Lags. *Annals of Operations Research*, **102**(1-4) 17-37

Wiest, J.D. 1981 Precedence diagramming method: Some unusual characteristics and their implications for project managers, *Journal of Operations management*, Volume 1/3 Feb. 1981. 121-130