

Przegląd Naukowy – Inżynieria i Kształtowanie Środowiska nr 2 (48), 2010: 60–74  
(Prz. Nauk. Inż. Kszt. Środ. 2 (48), 2010)  
Scientific Review – Engineering and Environmental Sciences No 2 (48), 2010: 60–74  
(Sci. Rev. Eng. Env. Sci. 2 (48), 2010)

**Mieczysław POŁOŃSKI**

Katedra Geoinżynierii SGGW w Warszawie  
Department of Geotechnical Engineering WULS – SGGW

## **Przykład obliczenia terminów i zapasów czasu w harmonogramach jednopunktowych z różnymi typami relacji między zadaniami**

### **How to calculate dates and slack time in PDM diagram with different relationship types between tasks**

**Słowa kluczowe:** sieci jednopunktowe, relacje, terminy zadań, ścieżka krytyczna

**Key words:** Precedence Diagramming Method (PDM), relationship, activity dates, critical paths

#### **Wprowadzenie**

Podstawy teoretyczne budowy i obliczania harmonogramów sieciowych znane są już od szeregu lat (Marcinkowski 2008, Połoński 2001). Jednak przez długi czas stosowano harmonogramy oparte na budowie sieci zależności na podstawie tak zwanej notacji dwupunktowej. Definiuje ona, że zadanie odwzorowywane jest w postaci strzałki, leżącej między dwoma zdarzeniami określającymi fakt jego rozpoczęcia i zakończenia. Zdarzenie kończące daną czynność (tzw. zdarzenie następujące) jest jednocześnie zdarzeniem rozpoczynającym (poprzedzającym) kolejne zadanie. Oznacza to,

że kolejne zadanie nie może się rozpocząć bez całkowitego zakończenia jego poprzedników. Na podstawie tak przyjętych założeń opracowano powszechnie znany i łatwy w zastosowaniu (nawet przy obliczaniu ręcznym) algorytm wyznaczania terminów i zapasów czasu w harmonogramie. Zdecydowana większość publikacji, podawanych algorytmów obliczeniowych i przykładów dotyczy tej właśnie notacji. Jednak od pewnego czasu znacznie większą popularność zdobyły sieci jednopunktowe. Mają one również szerokie zastosowanie w popularnych programach komputerowych do harmonogramowania realizacji przedsięwzięć inżynierskich. W literaturze brak jest przykładów ilustrujących sposób wyznaczania terminów zadań i zapasów czasu w tego typu sieciach zależności, szczególnie gdy zawierają różne typy relacji między zadaniami.

## Możliwości modyfikacji budowy sieci jednopunktowych i ich wpływ na wyniki obliczeń

W jednopunktowych sieciach zależności zadania są przedstawione w postaci prostokątnych bloków, a strzałki między nimi obrazują powiązania technologiczne i organizacyjne, występujące między zadaniami (Murray 2007). Z chwilą opracowania tej właśnie notacji zdefiniowano również nowe typy relacji między zadaniami. W sieciach jednopunktowych wprowadzono trzy dodatkowe typy relacji: relację RR (rozpocznij – rozpocznij), ZZ (zakończ – zakończ) oraz budzącą wiele kontrowersji i praktycznie nieużywaną relację RZ (rozpocznij – zakończ). Dodatkowo na każdej relacji (łącznie ze standardową ZR, czyli zakończ – rozpocznij) wprowadzono możliwość deklarowania tzw. zwłoki (dodatniej lub ujemnej), czyli dodatkowego czasu, o jaki zostanie opóźnione (zwłoka dodatnia) lub przyspieszone (zwłoka ujemna) wykonanie następnika. Ważne jest, w jaki sposób te nowe typy relacji zostały zdefiniowane. Otóż relacja typu RR oznacza, że następnik dopóty nie może się rozpocząć, dopóki nie rozpocznie się poprzednik (czyli może się rozpocząć później). Relacja ZZ oznacza natomiast, że następnik nie może się zakończyć dopóty, dopóki nie zakończył się poprzednik (czyli również może się zakończyć później). Należy zauważyć, że w obu typach relacji następnik jest podporządkowany poprzednikowi.

Harmonogramy jednopunktowe wzbogacono również o kilka dodatkowych opcji obliczeniowych, takich jak: możliwość występowania kilku zadań początkowych i/lub końcowych w sieci zależności, moż-

liwość deklarowania terminów dyrektywnych dla zadań, możliwość przypisywania różnych kalendarzy do poszczególnych zadań, możliwość planowania zadania w terminach najpóźniejszych, a nie najwcześniejszych. Wprowadzenie zupełnie nowych typów relacji i dodatkowych opcji oczywiście musiało zmienić sposób obliczania terminów i zapasów czasu zadań w tak skonstruowanej sieci zależności. Jednak w literaturze, a nawet opisach programów nadal stosuje się definicje i sposób interpretacji kluczowych pojęć, jakimi są zapasy czasu i ścieżka krytyczna, wywodzące się z sieci dwupunktowych, bez uwzględnienia wprowadzonych rozszerzeń. Niestety prowadzi to czasami do nieporozumień.

Ponieważ równocześnie z wprowadzeniem sieci jednopunktowych dużą popularność zdobyły oparte na nich programy obliczeniowe do budowy i analizy harmonogramów, w związku z tym aktualnie obliczenia najczęściej wykonuje się na podstawie jednego z tych programów. Jednak jak wynika z doświadczeń autora i spostrzeżeń menedżerów projektów (Korman i Daniels 2003), wyniki tych obliczeń oraz ich interpretacja nie zawsze są czytelne, a czasami przynoszą zaskakujące spostrzeżenia. Najczęściej można spotkać się z następującymi przypadkami:

- ścieżka krytyczna nie przebiega od czynności początkowej do końcowej sieci,
- skrócenie zadania krytycznego nie prowadzi do skrócenia całego przedsięwzięcia,
- wyznaczenie jako zadania krytycznego zadania, którego wydłużenie nie opóźnia całego przedsięwzięcia,

- uznanie jako zadania krytycznego zadania, którego skrócenie może prowadzić do wydłużenia całego przedsięwzięcia.

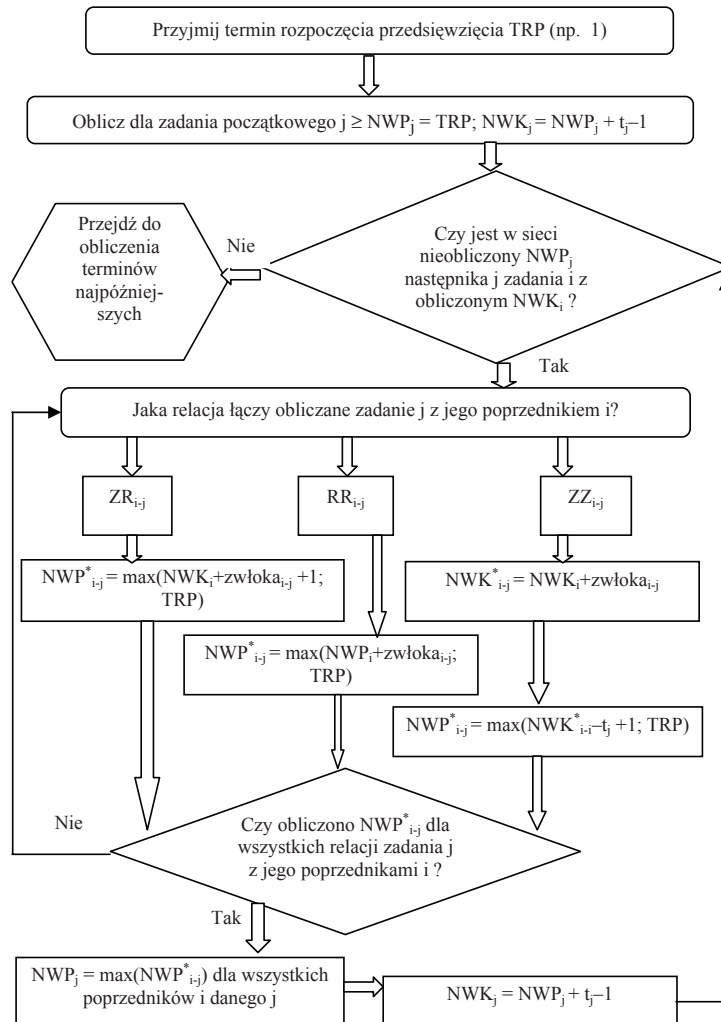
Występowanie tego rodzaju przypadków prowadzi z jednej strony do znacznych trudności w poprawnej interpretacji wyników obliczeń jednopunktowych harmonogramów sieciowych. Z drugiej strony próba przeanalizowania wyników obliczeń konkretnego harmonogramu czy obliczenie prostego przykładu z wykonaniem analiz ręcznie napotyka na znaczne trudności z powodu braku odpowiednich algorytmów obliczeniowych. Pomimo znacznej popularności tego rodzaju harmonogramów nie spotyka się w literaturze bardziej zaawansowanych przykładów obliczeniowych czy precyzyjnych algorytmów obliczeniowych. Nie ułatwiają tego zadania również twórcy programów komputerowych, którzy w dokumentacjach nie publikują konkretnych algorytmów obliczeniowych, a jedynie zadowolają się podawaniem ogólnych, opisowych definicji, często zresztą niezbyt precyzyjnych, a nawet wprowadzających w błąd (Połoński 2008). W dalszej części artykułu zostaną zaprezentowane opracowane przez autora algorytmy (i przykład), które pozwolą obliczyć najwcześniejsze możliwe i najpóźniejsze dopuszczalne terminy zadań oraz zapasy czasu w dowolnym, jednopunktowym harmonogramie sieciowym.

### **Algorytm wyznaczania terminów i zapasów czasu zadań w sieci jednopunktowej**

Poniżej przedstawiono w postaci schematów blokowych algorytm, pozwalający obliczyć terminy najwcześniejsze

możliwe (rys. 1) i najpóźniejsze dopuszczalne (rys. 2) oraz zapas całkowity i swobodny zadań (rys. 3) w jednopunktowej sieci zależności przy dowolnych typach relacji. Nie jest to jedyny możliwy wariant tego algorytmu, jest jednak zgodny z najbardziej popularnym na polskim rynku programem MS Project. Został on opracowany na podstawie doświadczeń autora, gdyż w opisie programu nie ma ścisłego sposobu obliczania tych wartości. W prezentowanych poniżej schematach używano następujących oznaczeń:

$i, j$  – numer zadania w sieci zależności,  
 $t_i$  – czas zadania  $i$ ,  
 $zwłoka_{i-j}$  – dodatnia lub ujemna zwłoka na relacji  $i-j$ ,  
 TRP – termin rozpoczęcia przedsięwzięcia przyjmowany z założenia,  
 TZP – termin zakończenia przedsięwzięcia przyjmowany z założenia jako  $\max(NWK_j)$  wszystkich zadań w sieci,  
 $NWP_j$  – najwcześniejszy początek zadania  $j$ ,  
 $NWP_{i-j}^*$  – najwcześniejszy początek zadania  $j$ , wynikający wyłącznie z relacji tego zadania z jego poprzednikiem  $i$ ,  
 $NWK_i$  – najwcześniejszy koniec zadania  $i$ ,  
 $NWK_{i-j}^*$  – najwcześniejszy koniec zadania  $j$ , wynikający wyłącznie z relacji tego zadania z jego poprzednikiem  $i$ ,  
 $NPP_{i-j}^*$  – najpóźniejszy początek zadania  $i$ , wynikający wyłącznie z relacji tego zadania z jego następnikami  $j$ ,  
 $NPK_{i-j}^*$  – najpóźniejszy koniec zadania  $i$ , wynikający wyłącznie z relacji tego zadania z jego następnikami  $j$ ,  
 $ZC_i$  – zapas całkowity zadania  $i$ ,  
 $ZS_i$  – zapas swobodny zadania  $i$ ,  
 $ZS_{i-j}^*$  – zapas swobodny zadania  $i$ , wynikający wyłącznie z relacji tego zadania z jego następnikiem  $j$ .



RYSUNEK 1. Algorytm wyznaczania terminów najwcześniejszych możliwych zadań w sieci jednopunktowej dla różnych typów relacji między zadaniami  
 FIGURE 1. Algorithm for determining earliest-start date in PDM diagram with different types of relationships between tasks

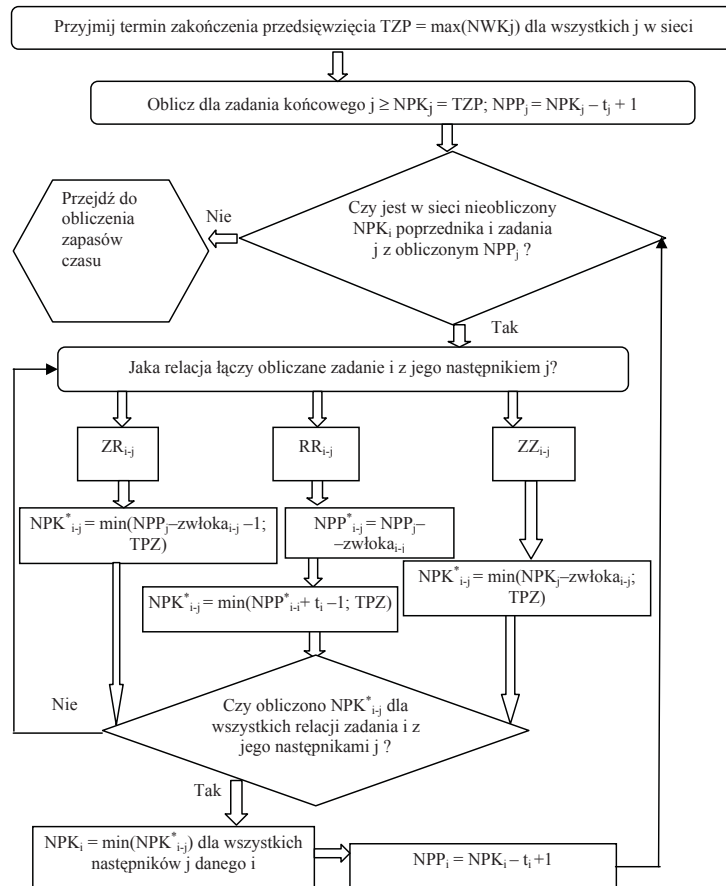
### Przykład obliczeniowy

Poniżej (tab. 1) przedstawiono krótki przykład obliczeniowy, ilustrujący sposób prowadzenia obliczeń według za-

mieszczonych algorytmów. Sieć zależności liczy 17 zadań. Obliczenia wykonano w kolejnych dniach roboczych, zakładając termin rozpoczęcia robót równy 1. W programie MS Project przyjęto ka-

TABELA 1. Przykład. Dane i wyniki obliczeń  
 TABLE 1. Example. Data and computational outcomes

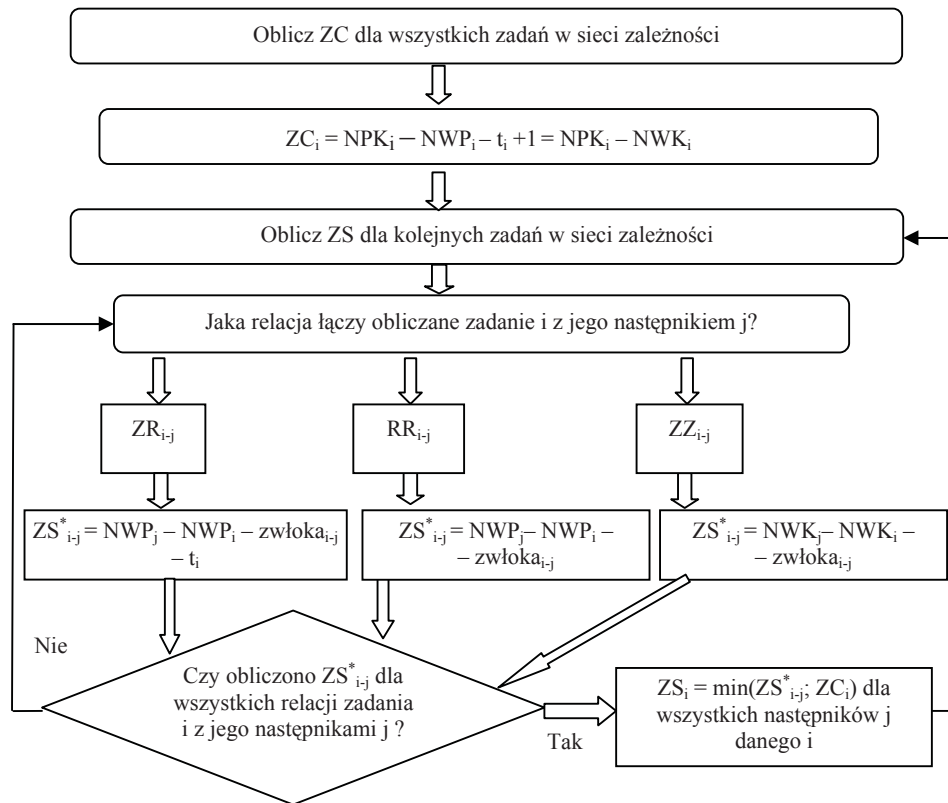
Lp. No	Nazwa Name	Czas Time	Poprzedniki id zadania, relacja i zwłoka Predecessor Id, Relationship and Lag	NWPi	NWKi	NPPi	NPKi	ZCi	ZSi
1	A	3 dn		1	3	1	3	0 dn	0 dn
2	B	3 dn	1ZZ-1 dzień	1	3	4	6	3 dn	3 dn
3	C	4 dn	1ZR+1 dzień	5	8	5	8	0 dn	0 dn
4	D	3 dn	3RR+1 dzień; 1 RR-2 dn; 2RR	6	8	6	8	0 dn	0 dn
5	E	2 dn	3ZZ-1 dzień; 4ZZ+1 dzień	8	9	15	16	7 dn	0 dn
6	F	1 dzień	4RR; 7ZR-2 dn	9	9	13	13	4 dn	0 dn
7	G	4 dn	4ZZ+2 dn; 2	7	10	7	10	0 dn	0 dn
8	H	2 dn	5ZR-1 dzień; 6ZZ+1 dzień	9	10	16	17	7 dn	6 dn
9	I	5 dn	6; 8ZZ+1 dzień; 10RR+3 dn	13	17	14	18	1 dzień	0 dn
10	J	3 dn	7ZZ+2 dn	10	12	10	12	0 dn	0 dn
11	K	4 dn	10RR+2 dn	12	15	12	15	0 dn	0 dn
12	L	2 dn	10	13	14	18	19	5 dn	4 dn
13	M	4 dn	12ZZ+1 dzień; 14RR; 9ZZ	16	19	17	20	1 dzień	0 dn
14	N	2 dn	8RR; 9ZR-2 dn	16	17	17	18	1 dzień	0 dn
15	O	3 dn	14RR+1 dzień; 13RR-1 dzień	17	19	20	22	3 dn	2 dn
16	P	6 dn	12RR; 11ZR+3 dn	19	24	19	24	0 dn	0 dn
17	R	3 dn	16RR-1 dzień; 13ZR+1 dzień; 15ZR-1 dzień	21	23	22	24	1 dzień	1 dzień



RYSUNEK 2. Algorytm wyznaczania terminów najpóźniejszych dopuszczalnych zadań w sieci jedno-punktowej dla różnych typów relacji między zadaniami  
 FIGURE 2. Algorithm for determining latest-finish date in PDM diagram with different types of relationships between tasks

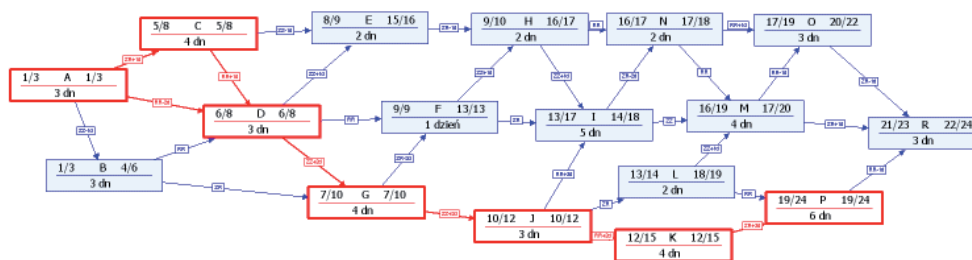
lendarz bez dni wolnych, a jako datę rozpoczęcia 1.03.2010. Budowa sieci zależności oraz relacje między zadaniami zostały tak dobrane, aby możliwie pełnie ilustrowały różne przypadki obliczeniowe. Równocześnie prezentowane są kopie ekranów z programu MS Project 2007 (rys. 4 i 5), pozwalające porównać wyniki obliczeń ręcznych z wynikami obliczeń w pro-

gramie komputerowym. Na rysunku 4 zamieszczono sieć zależności, a dla każdego zadania podano: w lewym górnym rogu termin najwcześniejszego możliwego rozpoczęcia i zakończenia zadania, w prawym górnym termin najpóźniejszego dopuszczalnego rozpoczęcia i zakończenia zadania (rozdzielone nazwą zadania), a na dole czas jego trwania. Wyniki obliczeń ręcznych zamieszczono w tabelach 2, 3 i 4.



RYSUNEK 3. Algorytm wyznaczania zapasu całkowitego i swobodnego zadań w sieci jednopunktowej dla różnych typów relacji między zadaniami.

FIGURE 3. Algorithm for determining total float and free float in PDM diagram with different types of relationships between tasks



RYSUNEK 4. Kopia ekranu z programu MS Project 2007, przedstawiająca budowę sieci zależności oraz wyniki obliczeń przykładowego obliczeniowego

FIGURE 4. Print screen of MS Project 2007 showing creation of relationship network and outcomes of computational example

TABELA 2. Obliczenie najwcześniejszych możliwych terminów zadań  
 TABLE 2. Calculating earliest-start date of tasks

Lp. No of j	Nazwa Name j	Czas Time j	Poprzednik Predecessor i	Typ relacji Relation- ship i-j	Zwłoka Lag i-j	NWK* <sub>i-j</sub>	NWP* <sub>i-j</sub>	NWP <sub>j</sub>	NWK <sub>j</sub>
1	2	3	4	5	6	7	8	9	10
1	A	3 dn						1	1+3-1=3
2	B	3 dn	1	ZZ	-1	3-1=2	max(2-3+1;1)=1	1	1+3-1=3
3	C	4 dn	1	ZR	1		3+1+1=5	5	5+4-1=8
4	D	3 dn	1	RR	-2		max(1-2;1)=1	6	6+3-1=8
			2	RR	0	1+0=1			
			3	RR	1	5+1=6			
5	E	2 dn	3	ZZ	-1	8-1=7	7-2+1=6	8	8+2-1=9
			4	ZZ	1	8+1=9	9-2+1=8		
7	G	4 dn	2	ZR	0		3+0+1=4	7	7+4-1=10
			4	ZZ	2	8+2=10	10-4+1=7		
6	F	1 dzień	4	RR	0		6+0=6	9	9+1-1=9
			7	ZR	-2	10-2+1=9			
8	H	2 dn	5	ZR	-1		9-1+1=9	9	9+2-1=10
			6	ZZ	1	9+1=10	10-2+1=9		
10	J	3 dn	7	ZZ	2	10+2=12	12-3+1=10	10	10+3-1=12
			6	ZR	0		9+0+1=10		
9	I	5 dn	8	ZZ	1	10+1=11	11-5+1=7	13	13+5-1=17
			10	RR	3	10+3=13			
11	K	4 dn	10	RR	2		10+2=12	12	12+4-1=15
12	L	2 dn	10	ZR	0		12+0+1=13	13	13+2-1=14



cd. tab. 2 / Table 2, cont.

1	2	3	4	5	6	7	8	9	10
14	N	2 dn	8	RR	0		9+0=9	16	16+2-1=17
			9	ZR	-2	17-2+1=16			
13	M	4 dn	9	ZZ	0	17+0=17	17-4+1=14	16	16+4-1=19
			12	ZZ	1	14+1=15	15-4+1=12		
			14	RR	0		16+0=16		
15	O	3 dn	13	RR	-1		16-1=15	17	17+3-1=19
			14	RR	1		16+1=17		
16	P	6 dn	11	ZR	3		15+3+1=19	19	19+6-1=24
			12	RR	0		13+0=13		
17	R	3 dn	13	ZR	1		19+1+1=21	21	21+3-1=23
			15	ZR	-1		19-1+1=19		
			16	RR	-1		19-1=18		

TABELA 3. Obliczenie najpóźniejszych dopuszczalnych terminów zadań  
TABLE 3. Calculating latest-finish date of tasks

Lp. No of j	Nazwa Name j	Czas Time j	Poprzednik Predecessor i	Typ relacji Relation- ship i-j	Zwłoka Lag i-j	NPP* <sub>ij</sub>	NPK* <sub>ij</sub>	NPK <sub>i</sub>	NPP <sub>i</sub>
17	R	3 dn						24	24-3+1=22
16	P	6 dn	17	RR	-1	22-(-1)=23	min(23+6-1,24)=24	24	24-6+1=19
15	O	3 dn	17	ZR	-1		22-(-1)-1=22	22	22-3+1=20
13	M	4 dn	15	RR	-1	20-(-1)=21	min(21+4-1,24)=24	20	20-4+1=17
			17	ZR	1		22-1=20		

14	N	2 dn	13	RR	0	17-0=17	17+2-1=18	18	18-2+1=17
			15	RR	1	20-1=19	19+2-1=20		
12	L	2 dn	13	ZZ	1		20-1=19	19	19-2+1=18
			16	RR	0	19-0=19	19+2-1=20		
11	K	4 dn	16	ZR	3		19-3-1=15	15	15-4+1=12
			13	ZZ	0		20-0=20		
9	I	5 dn	14	ZR	-2		17-(-2)-1=18	18	18-5+1=14
			9	RR	3	14-3=11	11+3-1=13		
10	J	3 dn	11	RR	2	12-2=10	10+3-1=12	12	12-3+1=10
			12	ZR	0		18-0-1=17		
8	H	2 dn	9	ZZ	1		18-1=17	17	17-2+1=16
			14	RR	0	17-0=17	17+2-1=18		
6	F	1 dzień	8	ZZ	1		17-1=16	13	13-1+1=13
			9	ZR	0		14-0-1=13		
7	G	4 dn	6	ZR	-2		13-(-2)-1=14	10	10-4+1=7
			10	ZZ	2		12-2=10		
5	E	2 dn	8	ZR	-1		16-(-1)-1=16	16	16-2+1=15
			5	ZZ	1		16-1=15		
4	D	3 dn	6	RR	0	13-0=13	13+3-1=15	8	8-3+1=6
			7	ZZ	2		10-2=8		
3	C	4 dn	4	RR	1	6-1=5	5+4-1=8	8	8-4+1=5
			5	ZZ	-1		16-(-1)=17		
2	B	3 dn	4	RR	0	6-0=6	6+3-1=8	6	6-3+1=4
			7	ZR	0		7-0-1=6		
1	A	3 dn	2	ZZ	-1		6-(-1)=7	3	3-3+1=1
			3	ZR	1		5-1-1=3		
			4	RR	-2	6-(-2)=8	8+3-1=10		

TABELA 4. Obliczenie zapasu całkowitego i swobodnego zadań  
 TABLE 4. Calculating total float and free float of tasks

Lp. No of i	Nazwa Name i	Czas Time i	Następnik Successor j	Typ relacji Relation- ship i-j	Zwłoka Lag i-j	ZC <sub>i</sub>	ZS* <sub>i,j</sub>	ZS <sub>i</sub>
1	A	3 dn	2	ZZ	-1	3-3=0	3-3-(-1)=1	0
			3	ZR	1		5-1-1-3=0	
			4	RR	-2		6-1-(-2)=7	
2	B	3 dn	4	RR	0	6-3=3	6-1-0=5	3
			7	ZR	0		7-1-0-3=3	
3	C	4 dn	4	RR	1	8-8=0	6-5-1=0	0
			5	ZZ	-1		9-8-(-1)=2	
4	D	3 dn	5	ZZ	1	8-8=0	9-8-1=0	0
			6	RR	0		9-6-0=3	
			7	ZZ	2		10-8-2=0	
5	E	2 dn	8	ZR	-1	16-9=7	9-8-(-1)-2=0	0
6	F	1 dzień	8	ZZ	1	13-9=4	10-9-1=0	0
			9	ZR	0		13-9-0-1=3	
7	G	4 dn	6	ZR	-2	10-10=0	9-7-(-2)-4=0	0
			10	ZZ	2		12-10-2=0	
8	H	2 dn	9	ZZ	1	17-10=7	17-10-1=6	6
			14	RR	0		16-9-0=7	
9	I	5 dn	13	ZZ	0	18-17=1	19-17-0=2	0
			14	ZR	-2		16-13-(-2)-5=0	
10	J	3 dn	9	RR	3	12-12=0	13-10-3=0	0
			11	RR	2		12-10-2=0	
			12	ZR	0		13-10-0-3=0	
11	K	4 dn	16	ZR	3	15-15=0	19-12-3-4=0	0
12	L	2 dn	13	ZZ	1	19-14=5	19-14-1=4	4
			16	RR	0		19-13-0=6	
13	M	4 dn	15	RR	-1	20-19=1	17-16-(-1)=2	0
			17	ZR	1		21-16-1-4=0	
14	N	2 dn	13	RR	0	18-17=1	16-16-0=0	0
			15	RR	1		17-16-1=0	
15	O	3 dn	17	ZR	-1	22-19=3	21-17-(-1)-3=2	2
16	P	6 dn	17	RR	-1	24-24=0	21-19-(-1)=3	0
17	R	3 dn				24-23=1	ZS* <sub>i,j</sub> =ZC <sub>i</sub> =1	1

## Podsumowanie i wnioski

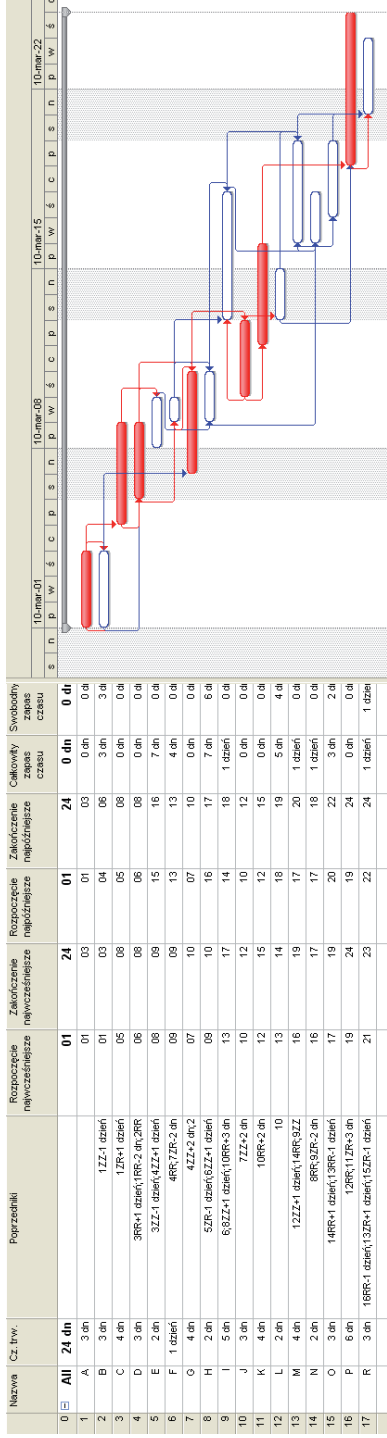
Śledząc sposób prowadzenia obliczeń, warto zwrócić uwagę na następujące ich aspekty:

1. W prezentowanym przykładzie występuje tylko jedno zadanie początkowe i jedno zadanie końcowe. Istnieje możliwość prowadzenia obliczeń bez zachowania tych warunków, jednak prowadzi to do dodatkowych komplikacji (np. czy dla wszystkich zadań początkowych przyjąć ten sam termin rozpoczęcia). W takich sytuacjach autor jest zwolennikiem wprowadzania czynności pozornych (np. o nazwie Start i/lub Koniec) z czasem równym zero, łączącym standardową relacją zakończ – rozpocznij wszystkie dotychczasowe zadania początkowe i/lub końcowe z nowo utworzonymi zadaniami pozornymi.
2. Podczas obliczania najwcześniejszych możliwych terminów przy wyznaczaniu terminu następnika połączonego relacją zakończ – zakończ należy najpierw policzyć NWK tego zadania, wynikające z analizowanej relacji, a dopiero w dalszej kolejności jego NWP i ewentualnie porównać je z pozostałymi NWP, wynikającymi z reszty relacji obliczanego zadania z jego poprzednikami.
3. Podczas obliczania najpóźniejszych dopuszczalnych terminów przy wyznaczaniu terminu poprzednika połączonego relacją rozpocznij – rozpocznij należy najpierw policzyć NPP tego zadania, wynikające z analizowanej relacji, a dopiero w dalszej kolejności jego NPK i ewentualnie porównać je z pozostałymi NPK, wynikającymi z reszty relacji

obliczanego zadania z jego następnikami.

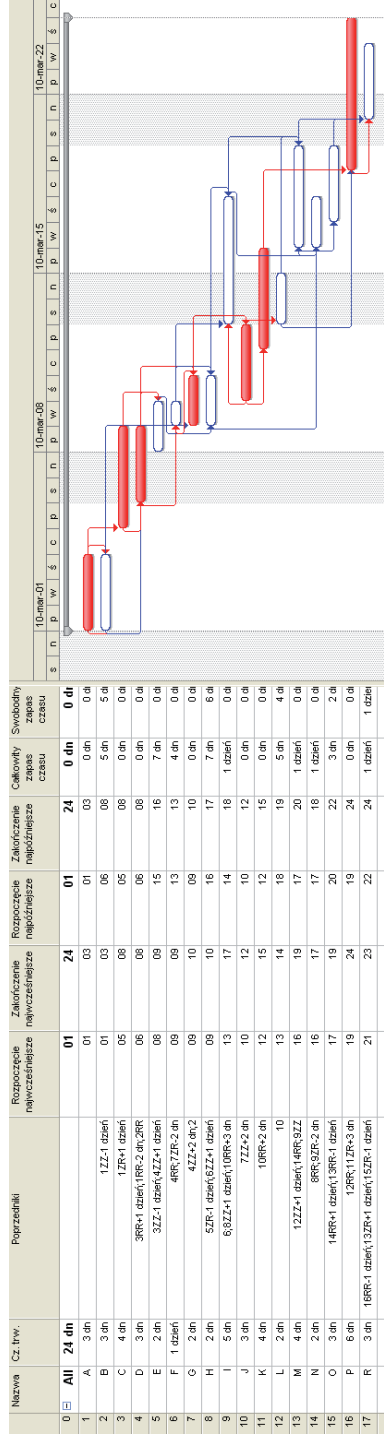
Analizując uzyskane wyniki obliczeń, należy zauważyć że:

1. Jedyne końcowe zadanie w sieci nie jest zadaniem krytycznym, co oznacza, że ścieżka krytyczna nie musi przebiegać od zadania początkowego do końcowego. W tym przypadku ostatnie krytyczne zadanie P z zadaniem końcowym harmonogramu R połączone jest relacją rozpocznij – rozpocznij, a ponieważ czas trwania zadania P jest dłuższy od zadania R, to właśnie zadanie P wyznacza termin zakończenia całego harmonogramu.
2. Skrócenie krytycznego zadania G nie prowadzi do skrócenia całego przedsięwzięcia (rys. 5 i 6). Należy zauważyć, że zadanie to połączone jest z krytycznym następnikiem i poprzednikiem relacją zakończ – zakończ. Skrócenie zadania G nie zmienia terminu jego zakończenia i w konsekwencji terminów realizacji jego następnika i całego harmonogramu.
3. Skrócenie krytycznego zadania J z trzech dni do na przykład jednego zamiast skrócić, wydłuża całe przedsięwzięcie z 24 do 26 dni (rys. 5 i 7), co pokazuje, jak nieumiejętne stosowanie rozszerzonych typów relacji w konstrukcji jednopunktowych sieci zależności może prowadzić do całkiem niezamierzonych efektów. W tym konkretnym wypadku zadanie J połączone jest z poprzednikiem relacją zakończ – zakończ, co według podanego algorytmu powoduje, że skrócenie tego zadania przy niezmiennym terminie zakończenia



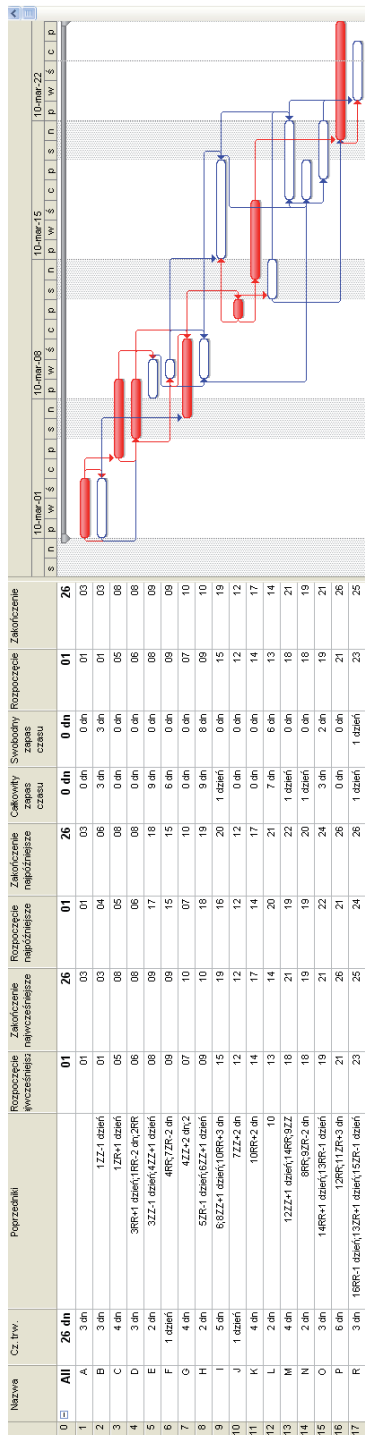
RY-SUNEK 5. Kopia ekranu z programu MS Project 2007 z danymi i wynikami obliczeń przykładowego obliczeniowego

FIGURE 5. Print screen of MS Project 2007 showing data and outcomes of computational example



RY-SUNEK 6. Skróceniu krytycznego zadania G nie prowadzi do skrócenia całego harmonogramu

FIGURE 6. Shortening a critical G task does not shorten the whole schedule



RYСУNEK 7. Skróceniu krytycznego zadania J prowadzi do wydłużenia całego harmonogramu  
 FIGURE 7. Shortening a critical J task does lengthen the whole schedule

opóźnia jego rozpoczęcie. Z kolei następnik zadania J połączony jest z tym zadaniem relacją rozpoczynij – rozpoczniij, a więc opóźnienie rozpoczęcia poprzednika opóźnia jego rozpoczęcie, a ponieważ jest to zadanie krytyczne, więc zakończenie całego harmonogramu.

Na podstawie przeprowadzonych obliczeń można wyciągnąć następujące wnioski:

- stosowanie rozszerzonych typów relacji w sieciach jednopunktowych z jednej strony uelastycznia budowany harmonogram, jednak z drugiej prowadzi do znacznie trudniejszej interpretacji wyników obliczeń,
- skrócenie zadań, leżących na ścieżce krytycznej w sieci jednopunktowej z rozszerzonymi typami relacji między zadaniami, nie musi prowadzić do skrócenia całego harmonogramu, a w skrajnych przypadkach może nawet powodować jego opóźnienie,
- aktualnie stosowany sposób wyznaczania zapasu całkowitego czasu w sieciach jednopunktowych nie uprawnia do identycznej interpretacji tego parametru jak w sieciach dwupunktowych.

Celem artykułu było przedstawienie sposobu obliczania terminów i zapasów czasu w sieciach jednopunktowych z różnymi typami relacji między zadaniami. Opracowane algorytmy oraz zamieszczony przykład obliczeniowy pozwalają w prosty sposób przeanalizować sposób wyznaczania tych wartości. Znajomość stosowanych rozwiązań pozwoli użytkownikom programów komputerowych nie tylko poprawnie interpretować uzyskiwane wyniki obliczeń, lecz również lepiej konstruować sieci zależności, bę-

dającą podstawą planowania i zarządzania realizacją obiektów inżynierskich.

## Literatura

- KORMAN R., DANIELS S.H. 2003: Critics Can't Find the Logic in Many of Today's CPM Schedules. *ENR: Engineering News-Record* 5, 26 250, 20.
- MARCINKOWSKI R. 2008: Teoretyczne podstawy harmonogramowania realizacji zadań budowlanych. Problemy przygotowania i realizacji inwestycji budowlanych. Warsztaty dla inżynierów budownictwa, Puławy 22–24 października 2008 r., Warszawa.
- MURRAY B.W. 2007: *Faster Construction Projects With CPM Scheduling*. McGraw-Hill Professional Publishing, New York.
- POŁOŃSKI M. 2001: *Harmonogramy sieciowe w robotach inżynierskich*. Wydawnictwo SGGW, Warszawa.
- POŁOŃSKI M. 2008: Stosowanie rozszerzonych typów relacji w harmonogramach przedsięwzięć budowlanych. Problemy przygotowania i realizacji inwestycji budowlanych. Warsztaty dla inżynierów budownictwa, Puławy 22–24 października 2008 r., Warszawa.

## Summary

**How to calculate dates and slack time in PDM diagram with different relationship types between tasks.** This article aims to present method of calculating dates and slack time in PDM diagrams with different types of relationships between tasks. Algorithms that have been worked out by the Author as well as enclosed computational example enable to analyze a process of determining these values. Knowledge of adopted algorithms facilitates accurate interpretation of calculation outcomes and improved construction of relationship networks for users of computer programmes. These two crafts are ultimately fundamental for planning and realization management of engineering objects.

### Author's address:

Mieczysław Połoński  
Szkoła Główna Gospodarstwa Wiejskiego  
ul. Nowoursynowska 159, 02-776 Warszawa  
e-mail: mieczyslaw\_polonski@sggw.pl