

# Optimalizace – metody síťové analýzy

Bohumil Gajda

---

Bakalářská práce  
2007



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2006/2007

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bohumil GAJDA**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
  
Téma práce: **Optimalizace – metody síťové analýzy**

Zásady pro vypracování:

Cílem práce bude seznámit se s metodami používanými při síťové analýze, popsat vybranou z nich a prezentovat její užití na vybraných problémech

Práce se bude skládat z bodů

1. Vypracování přehledu problematiky optimalizace
2. Popis vybrané metody síťové analýzy
3. Zhodnotit a provést závěr

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

**Maňas, Miroslav: Teorie her a optimální rozhodování, SNTL Praha, 1974**  
**Zuchovickij, S., I., Radčiková, I., A.: Matematické metody síťové analýzy, SNTL Praha 1973**

Vedoucí bakalářské práce: **RNDr. Ing. Miloš Krčmář**  
Ústav aplikované informatiky

Datum zadání bakalářské práce: **13. února 2007**

Termín odevzdání bakalářské práce: **24. května 2007**

Ve Zlíně dne 13. února 2007



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Tématem této bakalářské práce jsou metody optimalizace a popis metod síťové analýzy. Celé dílo je rozděleno do čtyř částí.

První dvě jsou součástí teoretické části. Jedna stručně popisuje metody optimalizace a jejich principy v členění dle jejich klasifikace. Druhá je teoretickou přípravou pro práci se síťovým grafem a stanovuje a popisuje používané pojmy.

Navazující analytická část se zabývá jednotlivými metodami analýzy a optimalizace síťového grafu. Popisuje je z hlediska jejich pohledu na výslednou optimalizovanou veličinu a popisuje používané postupy.

V poslední, projektové části je vybranou metodou kritické cesty analyzován jednoduchý síťový graf a jsou prakticky ukázány jednotlivé postupy analýzy.

Závěrem konstatuji, že popsání metody mohou být účinné jen v případě přesně a kompletně popsání projektů.

*Klíčová slova:* Optimalizace, Metody síťové analýzy.

## **Abstract**

The topic of this thesis are methods of optimization and description of methods of network analysis.

The whole thesis is divided into four parts.

First two parts are concerned with the theoretical aspects of the network analysis. One of them briefly describes the methods of optimization and its principles in order of their classifications. The second one lays down the theoretical ground work for the work with network charts and determines and expands the notions used.

The subsequent analytical part deals with the particular methods of analysis and of optimization of the network charts. It describes them by taking into account their view of the resultant optimized value and presents the procedures used.

In the final project part is by the selected method of critical path analyzed simple network chart and are partially demonstrated the existing procedures of analysis.

My conclusion is that the methods described can be effective only in case of precisely and of completely described projects.

*Keywords:* Optimization, Methods of network analysis

## OBSAH

ÚVOD.....	8
<b>I</b> <b>TEORETICKÁ ČÁST</b> .....	<b>9</b>
<b>1</b> <b>OPTIMALIZACE</b> .....	<b>10</b>
1.1    KLASIFIKACE ÚLOH OPTIMALIZACE .....	10
1.2    KLASIFIKACE METOD OPTIMALIZACE .....	11
1.2.1    Analytické metody .....	11
1.2.2    Iterační metody .....	13
1.2.3    Speciální metody .....	17
1.2.4    Teorie her a optimálního rozhodování.....	18
<b>2</b> <b>STRUČNÝ ÚVOD DO GRAFŮ</b> .....	<b>19</b>
2.1    PRAVIDLA SESTAVENÍ SÍŤOVÉHO GRAFU .....	19
2.2    ČÍSLOVÁNÍ UZLŮ .....	21
2.3    FORDŮV ALGORITMUS (PRO ČÍSLOVÁNÍ UZLŮ V ROZSÁHLÝCH SÍŤÍCH).....	23
2.4    VYHLEDÁVÁNÍ CYKLŮ .....	23
<b>II</b> <b>ANALYTICKÁ ČÁST</b> .....	<b>25</b>
<b>3</b> <b>KRITICKÁ CESTA</b> .....	<b>26</b>
3.1    TRVÁNÍ PROJEKTU A KRITICKÁ CESTA.....	26
3.1.1    Nejdříve možné začátky činností .....	26
3.1.2    Určení kritické cesty .....	27
3.2    NEJPOZDĚJI PŘÍPUSTNÉ TERMÍNY UZLŮ .....	27
3.3    PODMÍNKY KRITičNOSTI UZLU A ČINNOSTÍ.....	28
3.4    ČASOVÉ REZERVY.....	28
3.5    SUBKRITICKÉ ČINNOSTI .....	29
3.6    METODY REALIZACE ALGORITMU VÝPOČTU $T_j^{(0)}$ A $T_j^{(1)}$ .....	30
3.6.1    Maticová metoda .....	30
3.6.2    Výpočet v síťovém grafu .....	31
3.6.3    Výpočet v tabulce .....	33
3.7    LINEÁRNÍ DIAGRAM PROJEKTU .....	34
<b>4</b> <b>ROZVRŽENÍ ZDROJŮ</b> .....	<b>35</b>
4.1    OPTIMALIZACE ROZDĚLENÍ ZDROJŮ VZHLEDEM K ČASU .....	35
4.1.1    Formulace úlohy při konstantních intenzitách .....	35
4.1.2    Vyrovnání nároků na zdroje.....	36
4.1.3    Formulace úlohy při proměnných intenzitách.....	37
4.1.4    Definice minimálního zdržení dokončení projektu.....	38

4.2	OPTIMÁLNÍ VYROVNÁNÍ NÁROKŮ NA ZDROJ PŘI ZADANÉM TERMÍNU .....	39
4.2.1	Formulace úlohy .....	39
4.2.2	Minimalizace střední kvadratické odchylky .....	39
4.2.3	Minimalizace maximálního nároku na zdroj .....	40
<b>5</b>	<b>MINIMALIZACE NÁKLADŮ .....</b>	<b>42</b>
5.1	MINIMALIZACE NÁKLADŮ NA PROJEKT PŘI JEHO KONSTANTNÍM TRVÁNÍ ...	42
5.1.1	Optimální plán bez rezerv .....	42
5.1.2	Algoritmus sestavení optimálního plánu bez rezerv .....	43
5.1.3	Optimální plán při existenci rezerv .....	45
5.2	PARAMETRICKÁ ÚLOHA MINIMALIZACE NÁKLADŮ NA PROJEKT .....	45
5.2.1	Matematický model úlohy .....	45
5.3	KELLEYOVA METODA .....	46
5.3.1	Struktura optimálního plánu .....	46
5.3.2	Kelleyova věta .....	46
5.3.3	Přechod k úloze o maximálním toku .....	48
5.3.4	Algoritmus řešení parametrické úlohy .....	49
5.3.5	Speciální algoritmus pro řešení úlohy o maximálním toku .....	50
5.4	NĚKTERÉ APLIKACE KELLEYOVY METODY .....	52
5.4.1	Vyhledání optimálního plánu vzhledem k nákladům .....	52
5.4.2	Optimální plán vzhledem k času.....	52
<b>6</b>	<b>MAXIMÁLNÍ TOKY V SÍTI .....</b>	<b>53</b>
6.1	PROSTÁ ÚLOHA O MAXIMÁLNÍM TOKU.....	53
6.1.1	Základní pojmy a formulace problému .....	53
6.2	ZOBECNĚNÁ ÚLOHA O MAXIMÁLNÍM TOKU .....	54
6.2.1	Formulace úlohy.....	54
6.2.2	Algoritmus .....	54
6.3	POUŽITÍ V SÍŤOVÉ ANALÝZE .....	55
6.3.1	Duální úloha k úloze o maximálním toku .....	55
<b>III</b>	<b>PROJEKTOVÁ ČÁST .....</b>	<b>57</b>
<b>7</b>	<b>PŘÍKLAD .....</b>	<b>58</b>
7.1	ZADÁNÍ.....	58
7.1.1	Graf .....	58
7.1.2	Převod do tabulky .....	58
7.2	TRVÁNÍ PROJEKTU A KRITICKÁ CESTA.....	60
7.2.1	Vyhledání nejdříve možných začátků činností a trvání projektu.....	60
7.2.2	Vyhledání kritické cesty a volných rezerv .....	60
7.3	NALEZENÍ NEJPOZDĚJI PŘÍPUSTNÝCH TERMÍNŮ.....	61

---

7.4	ZJIŠTĚNÍ PODMÍNEK KRITIČNOSTI UZLŮ A ČINNOSTÍ .....	61
7.5	ČASOVÉ REZERVY.....	63
7.6	SUBKRITICKÉ ČINNOSTI .....	63
	<b>ZÁVĚR .....</b>	<b>65</b>
	<b>CONCLUSION .....</b>	<b>66</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>67</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>69</b>
	<b>SEZNAM TABULEK.....</b>	<b>70</b>

## ÚVOD

Cílem této bakalářské práce je seznámit se s metodami používanými při síťové analýze, popsat je a na vybrané metodě kritické cesty prezentovat její užití.

### Členění bakalářské práce

V první části teorie se seznámíme se základními pojmy *optimalizace*. Popíšeme typy úloh a seznámíme se s metodami optimalizace.

V druhé části teorie si popíšeme některé principy a postupy při tvorbě *síťového grafu* a ukážeme si některé metody přípravy grafů pro další práci. Ukážeme možnosti transformace grafů – jak vyhledávat cykly, díky kterým by další výpočty byly nesmyslné, popíšeme si možnosti agregace a převodu na lineární diagram projektu (ten je přehlednější zejména pro menší projekty).

V teoretické části se budeme věnovat jednotlivým *metodám analýzy síťového grafu* a popíšeme si jejich principy.

Nejdříve se budeme věnovat optimálnímu rozvržení zdrojů vzhledem k času a vzhledem k nárokům na zdroje při zadaném termínu. Doposud neexistuje způsob určení přesného řešení této úlohy. Pomocí uvedených algoritmů se budeme snažit nalézt řešení přibližné.

V další kapitole si zformulujeme prostou a obecnou úlohu o maximálním toku v síti a popíšeme algoritmy jejího řešení a použitelnost v síťové analýze.

V páté kapitole probereme různé varianty minimalizace nákladů na projekt. Parametrickou úlohu ukazující závislost minimálních nákladů projektu na délce jeho trvání. Tyto úlohy jsou vloženy do obecného schématu teorie konvexního a lineárního programování a obecně mohou být řešeny simplexovou metodou s některými jejími modifikacemi a obecnou metodou konvexního programování. Ukážeme si i Kelleyovu metodu, převádějící tyto úlohy na známou úlohu o maximálním toku v síti, která je řešena jednoduchým algoritmem.

Na připraveném grafu si podrobně probereme metody vyhledání kritické posloupnosti činnosti (kritické cesty), tj. řetězce činnosti, který určuje minimální dobu dokončení celého projektu (kritické doby) a metody nalezení ostatních parametrů síťového grafu.



# I. TEORETICKÁ ČÁST

## 1 OPTIMALIZACE

Při základním pohledu na úlohy optimalizace je můžeme rozdělit na dvě základní skupiny. „Optimalizaci“, zabývající se statickými systémy a „Optimální řízení“, řešící optimalizační úlohy dynamických systémů.

Výsledkem optimalizace *statického systému* je nalezení extrémů (minima a maxima) účelové funkce:

$$y(x_1, x_2, x_3) = x_1^2 + 2x_2^2 + 0,5x_3^3$$
$$y(x) = 5x^4 + 3x^3 + 2x^2 + x + 1$$

U *dynamických systémů* je výsledkem optimalizace (optimálního řízení) funkce  $f(t)$ :

$$y'(t) = 2y(t) + 2u(t)$$
$$y_{km} = 3y_k + 7u_k$$

maximálně vystihující popis chování dynamického systému.

Účelovou funkcí optimalizace je reálná funkce reálných proměnných – hledaných extrémů, například  $f = \bar{c} \cdot \bar{x}$ , kde  $c$  je vektor optimalizované proměnné a  $x$  je vektor procesů [3].

### 1.1 KLASIFIKACE ÚLOH OPTIMALIZACE

Úlohy optimalizace můžeme rozdělit podle těchto kritérií:

#### 1. Volný extrém

- Jednorozměrný
- Vícerozměrný

#### 2. Klasický vázaný extrém

- Omezení typu rovnost

#### 3. Neklasický vázaný extrém

- Omezení typu nerovnost

#### 4. Antagonistické hry a optimální rozhodování

## 1.2 KLASIFIKACE METOD OPTIMALIZACE

### 1.2.1 Analytické metody

#### Jednorozměrné případy – derivace

Nechť je  $f(x)$  funkce reálné proměnné.

1. **Ostré lokální maximum** v bodě  $x_0$ , pokud má v daném bodě první i druhou derivaci a platí, že  $f'(x_0) = 0 \wedge f''(x_0) < 0$
2. **Ostré lokální minimum** v bodě  $x_0$ , pokud má v daném bodě první i druhou derivaci a platí, že  $f'(x_0) = 0 \wedge f''(x_0) > 0$

V případě, že jsou obě derivace nulové – necht' existuje přirozené číslo  $n$  tak, že  $f^{(n)}(x_0) \neq 0$  a pro všechna  $k < n$  je  $f^{(k)}(x_0) = 0$ . Pak platí:

1. je-li  $n$  sudé a  $f^{(n)}(x_0) < 0 \Rightarrow$  má funkce  $f$  v bodě  $x_0$  **Ostré lokální maximum**
2. je-li  $n$  sudé a  $f^{(n)}(x_0) > 0 \Rightarrow$  má funkce  $f$  v bodě  $x_0$  **Ostré lokální minimum**
3. je-li  $n$  liché a  $\Rightarrow$  má funkce  $f$  v bodě  $x_0$  **Inflexní bod** (bod ve kterém se funkce mění z konvexní v konkávní, nebo naopak).

Pokud je funkce  $f$  spojitá na  $J$  a  $f'(x_0) = 0$  a druhá derivace v  $x_0$  neexistuje, pak platí<sup>1)</sup>:

1. je-li  $f'(x) < 0$  v levém okolí a  $f'(x) > 0$  v pravém okolí  $x_0 \Rightarrow$  v bodě  $x_0$  má funkce ostré lokální minimum
2. je-li  $f'(x) > 0$  v levém okolí a  $f'(x) < 0$  v pravém okolí  $x_0 \Rightarrow$  v bodě  $x_0$  má funkce ostré lokální maximum [3]

#### Vícerozměrné případy – gradienty

Máme spojitou funkci reálné proměnné:  $f(x_1, \dots, x_n)$ .

Zobecnění **1. derivace**:  $\text{grad} f = \nabla f = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$

Zobecnění **2. derivace**: Hessova matice  $K = \nabla^2 f$

$$K = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \ddots & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

<sup>1)</sup>Tato věta platí i když v  $x_0$  neexistuje ani 1. derivace.

**Zobecnění inflexního bodu – sedlový bod**

Stacionární bod =  $\nabla f(x_0) = 0$

*Sylvestrovo kritérium:*

1. Účelová funkce má minimum, pokud je Hessova matice *Pozitivně definitní* (všechny hlavní subdeterminanty kladné).
2. Účelová funkce má maximum, pokud je Hessova matice *Negativně definitní* (první subdeterminant záporný a u ostatních se u střídají znaménka) [3].

Příklady:

$$\begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix} = ND \quad \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} = PD \quad \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} = \text{ani}ND, \text{ani}PD$$

**Omezení typu „=“.**

Metoda Lagrangeova multiplikátoru – klasický vázaný extrém  $f(x_1, \dots, x_n)$  při podmínkách  $g_j(x_1, \dots, x_m) = 0$ ;  $j = 1, \dots, m < n$ .

Metoda Lagrangeových multiplikátorů:

$$\phi(x, \lambda) = f(x) + \sum_{j=1}^m \lambda_j g_j(x)$$

Nechť  $f, g_1, \dots, g_m$  mají spojitě první parciální derivace a necht' jsou funkce  $\nabla g_j(x)$  lineárně nezávislé. Pak platí – je-li  $x_0$  extrém  $f$  při omezení  $g_1(x) = 0 \Rightarrow$  existuje  $\lambda_0 = (\lambda_0^1, \dots, \lambda_0^m)$  tak, že platí  $\nabla_x \phi(x_0, \lambda_0) = \nabla_\lambda \phi(x_0, \lambda_0) = 0$

1. Cílem je určit stacionární bod pro  $\forall x$  a pro  $\forall \lambda$

$$\frac{\partial \phi}{\partial x_i} = 0 \quad \text{a} \quad \frac{\partial \phi}{\partial \lambda_j} = 0$$

2. Minimum nebo maximum určíme netriviálně – pomocí totálních diferenciálů. Často je jednodušší porovnat funkční hodnoty.
3. Úloha má řešení, i když volný extrém neexistuje!
4. Dalším způsobem řešení je dosazovat z omezení. [3]

**Omezení typu „ $\leq$ “.**

Kuhn-Tuckerova věta (o sedlovém bodě) – neklasický vázaný extrém

$\max f(x_1, \dots, x_n)$  při omezeních  $g_j(x_1, \dots, x_n) \geq 0$ ;  $j = 1, \dots, m$ ;  $i = 1, \dots, n$ ;  $x_i \geq 0$

Lagrangeova funkce:  $\phi(x, \lambda) = f(x) + \sum_{j=1}^m \lambda_j g_j(x)$  + Kuhn-Tuckerova věta

Jestliže existuje  $x_0 \geq 0$  a  $\lambda_0 \geq 0$  tak, že pro  $\forall x \geq 0 \forall \lambda \geq 0$  platí:

$$\phi(x, \lambda_0) \leq \phi(x_0, \lambda_0) \leq \phi(x_0, \lambda)$$

potom  $x_0$  je optimálním řešením úlohy neklasického vázaného extrému.

Věta nemá konstruktivní charakter a pro výpočet se nehodí. Pro diferencovatelné funkce  $f, g_1$  se výpočet redukuje na tzv. Kuhn-Tuckerovy lokální podmínky:

Jestliže existují první parciální derivace  $f$  a  $g_j$  (všechny), pak nalezení extrému je ekvivalentní podmínkám:

$$\begin{aligned} \left(\frac{\partial \phi}{\partial x_i}\right)_{(x_0, \lambda_0)} &\leq 0 & \left(\frac{\partial \phi}{\partial \lambda_j}\right)_{(x_0, \lambda_0)} &\geq 0 \\ x_{0i} \left(\frac{\partial \phi}{\partial x_i}\right)_{(x_0, \lambda_0)} &= 0 & \lambda_{0j} \left(\frac{\partial \phi}{\partial \lambda_j}\right)_{(x_0, \lambda_0)} &= 0 \\ & i = 1, \dots, n & j = 1, \dots, m \end{aligned}$$

Ani v tomto případě není výpočet triviální [3].

### 1.2.2 Iterační metody

#### Komparativní (porovnávání)

princip:  $x_{k+1} = x_k + \Delta_k$   $x_0$  = počáteční podmínky, tedy konstrukce posloupnosti  $\{x_k\}_{k=0}^{\infty}$  s podmínkou

$$\lim_{k \rightarrow \infty} x_k = x_{opt}$$

#### 1. Jednorozměrové:

*Fibonacciho metoda a metoda Zlatého řezu*

Fibonacciho posloupnost:  $F_i = F_{i-1} + F_{i-2}; \quad F_0 = F_1 = 1$

tedy  $F_i \equiv \{1; 1; 2; 3; 5; 8; 13; 21; 34; \dots\}$

zlatý řez:  $\lim_{i \rightarrow \infty} \frac{F_i}{F_{i+1}} = 0,618$

po  $N$  krocích se  $[a; b]$  redukuje na délku:

$d_N = 2 \frac{F_1}{F_{N+1}} |b - a|$  pro Fibonacciho metodu

$d_N = (0,618)^N |b - a|$  pro metodu zlatého řezu

*Powellova metoda*

Využívá aproximační účelové funkce kvadratickou závislostí:  $y = a_0 + a_1x + a_2x^2$

koeficienty  $a_0, a_1, a_2$  se spočítají z trojice  $\{x_i, y_i = f(x_i)\}_{i=1}^3$  podle vztahů:

$$\begin{aligned} \downarrow \quad a_2 &= \frac{(y_3 - y_1)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_1)}{(x_2 - x_1)(x_3^2 - x_1^2) - (x_3 - x_1)(x_2^2 - x_1^2)} \\ \downarrow \quad a_1 &= \frac{y_2 - y_1 - a_2(x_2^2 - x_1^2)}{x_2 - x_1} \end{aligned} \quad \text{)}(A)$$

$\downarrow \quad a_0 = y_1 - a_1x_1 - a_2x_1^2$

podm. extrémů:  $y' = 0 \Rightarrow 2a_2x_e = -a_1; \quad x_e = -\frac{a_1}{2a_2}$  } (B) a dosadí se za  $a_1, a_2$ .

Bodem  $x_2$  se nahradí  $x_1$  a postupuje se dál.

## 2. Mnohorozměrné:

*Mapování kriteriální plochy*

- Nevýhody: obrovské množství výpočtu a nezaručené konvergence
- Výhoda: jednoduchost a algoritmizovatelnost

*Box-Wilsonova metoda*

- Nevýhody:  $(1 + 2^N)$  vyčíslení  $f(x)$
- Výhody: algoritmizovatelnost, modifikovatelnost

*Simplex je nejmenší konvexní polyedr v daném prostoru (rovnostranný trojúhelník v  $R^2$ , pravidelný čtyřstěn v  $R^3, \dots$ )*

*Metoda pravidelného simplexu*

Metoda spočívá v minimalizaci počtu vyčíslení účelové funkce  $(1 + 2^N) \rightarrow N + 1$

- Nevýhoda: určení extrému, kdy ukončit iteraci – počtem iterací, rozdílem funkčních hodnot.

*Metoda flexibilního simplexu (Nelder-Mead)*

Modifikace konstantnosti simplexů pomocí pravidelného simplexu pomocí dalšího výpočtu  $\Rightarrow$  nejdokonalejší komparativní metoda.

*Metoda cyklické záměny parametrů (Gauss-Seidl)*

Jedná se o postupné „řezy“ podle jednotlivých proměnných, ve kterých se provádí jednorozměrné optimalizace [3]

**Gradientní**

## Princip:

$$x_{k+1} = x_k + \lambda_k \text{grad } f(x_k)$$

$\lambda_k < 0$  – postup k minimu funkce

$\lambda_k > 0$  – je postup k maximu funkce

$\lambda_k = \text{konstantní} \Rightarrow$  gradientní metody s krátkým krokem

$\lambda_k = \text{proměnné} \Rightarrow$  gradientní metody s dlouhým krokem

## Výhoda:

nekumulují numerické chyby

## 1. Jednorozměrné:

*Regula falsi**Newtonova metoda*

v okolí  $x_k$  se účelová funkce rozvine do Taylorova rozvoje:

$$p(x) \approx f(x_k) + f'(x_k)(x - x_k) + f''(x_k) \frac{(x - x_k)^2}{2}$$

a nový iterační krok se hledá z podmínky  $p'(x) = 0 = f'(x_k) + f''(x_k)(x - x_k) = 0$   
a pro další  $x_{k+1}$  se volí hodnota  $x$ , tedy

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \Leftarrow \text{Newtonova metoda}$$

2. Mnohorozměrné bez omezení:

*S krátkým krokem*

*S dlouhým krokem*

Optimalizace dlouhého kroku:

$$\lambda_k = \max_{\lambda} \left\{ \frac{d}{d\lambda} p(\lambda) = 0 \right\}$$

kde  $p(\lambda) = f(x_k + \lambda \text{grad } f(x_k))$

*Kvazinevtonské*

vychází z aproximace účelové funkce  $f(x)$  kvadratickou formou:

$$f(x) \approx -(x - x^*)^T C (x - x^*) \equiv c^T x - x^T \tilde{C} x$$

pro maximum, kde  $x^*$  je bodem optima,  $c^T$  je číselný vektor,  $C$  a  $\tilde{C}$  jsou pozit. def. mat. (obvykle  $C \equiv \tilde{C}$ )

*Metoda konjugovaných gradientů*

předpoklad:  $f(x) \approx c^T x - x^T \tilde{C} x$ , pak  $\text{grad } f(x) = c - 2\tilde{C}x = -2C(x - x^*)$

Vektory  $d^{(1)}, \dots, d^{(N)}$  jsou konjugované k poz. def.  $\tilde{C}$  ( $\dim N \times N$ ), jestliže platí:  $d^{(i)T} \tilde{C} d^{(j)} = 0$  pro  $i \neq j$ .

O konjugovaných směrech platí věta:

Nechť  $d^{(1)}, \dots, d^{(N)}$  jsou konjugované směrem k  $C$ . Pak posloupnost  $x_{k+1} = s_k + \xi_k d^{(k)}$ , kde

$$\xi_k = \frac{[\text{grad } f(x_k)]^T d^{(k)}}{2d^{(k)T} C d^{(k)}}$$

konverguje k bodu extrému funkce  $f(x)$ .

– DFP (Davidov+Fletcher+Powell)

jednoduchý princip –  $x_{(i+1)} = x_i + \xi_i S_i \text{grad } f(x_i)$ , kde  $S_i$  je aproximace Hessovy matice

– PARTAN (parallel-tangent)

Metoda *paralelních tečen* (tangent)

Principem je vytvoření 2 paralelních směrů, z kterých se vybere jistý průměr.

– Algoritmus *CONGRA* (conjug-gradient)

využívá metodu konjugovaných gradientů

- 1. Nevýhoda: je potřeba vypočítat 2. derivaci

- 2. Nevýhoda: špatně konverguje pro protáhlé ekvidistantní křivky

3. Mnohorozměrné s omezením (typu  $\leq$ ):Metoda *projekce gradientu*Formulace:  $\max f(x_1, \dots, x_N)$  při lineárním omezení:  $Ax \leq b$  a podmínce nezápornosti  $x \geq 0$ <sup>1)</sup> [3]Princip: sestavení projekční matice  $Q$  z omezení tvaru:

$$Ax \leq b; -x \leq 0 \Rightarrow Q = \begin{bmatrix} A \\ -I \end{bmatrix}$$

Projekce:  $P = [I - Q^T(QQ^T)^{-1}Q]$  a nový směr  $q = (QQ^T)^{-1} \text{grad } f(x)$ **Metody náhodného vyhledávání**

princip:

$$x_{k+1} = x_k + \Delta x_k$$

$$\Delta x_k = a_k X[k, \Delta f_k, x_k] \xi_k$$

$$\Delta f_k = f(x_k) - f(x_{k-1})$$

kde  $X$  je funkce vyhodnocení (úspěšnost, neúspěšnost),  $\xi$  je rovnoměrně rozložená náhodná veličina na intervalu  $[-1; 1]$ *Klasifikace*

- *Jednoduché* (prosté, bez učení, s konstantní strategií)
- *Adaptační* (s učením, adaptací pravděpodobnosti, ...)
- *S prvky umělé inteligence* (genetické, evoluční, ...)

*Hledání bez vyhodnocení*

$$x_{k+1} = x_k + a_k \xi_k \quad \sum_{i=0}^{\infty} a_i = \infty \quad \sum_{i=0}^{\infty} a_i^2 < \infty$$

například:

$$a_k = \frac{1}{k}; \frac{c}{k}; \frac{c}{k+a}$$

*Hledání s návratem:*

$$\max: \Delta x = \frac{1}{2} a_k [1 + \text{sgn} \Delta f(x_{k-1})] \xi_k - \frac{1}{2} \Delta x_{k-1} [1 - \text{sgn} \Delta f(x_k)]$$

$$\max: \Delta x = \frac{1}{2} a_k [1 - \text{sgn} \Delta f(x_{k-1})] \xi_k - \frac{1}{2} \Delta x_{k-1} [1 + \text{sgn} \Delta f(x_k)]$$

*Hledání s lineárním přepočtem:*

$$\Delta x_k = \frac{1}{2} a_k [1 - \text{sgn} \Delta f(x_{k-1})] \xi_k - \Delta x_{k-1} [1 + \text{sgn} \Delta f(x_k)]$$

---

<sup>1)</sup>Jedná se o typicky ekonomicky formulovanou úlohu, analogickou s lineárním programováním



*Hledání s predikací:*

$$\Delta x_k = \frac{1}{2} \Delta x_k [1 - \operatorname{sgn} \Delta f(x_k)] \xi_i + \frac{1}{2} \Delta a_i [1 + \operatorname{sgn} \Delta f(x_k)] \xi_k$$

*Hledání s potrestáním náhodností:*

$$\Delta x_k = \frac{1}{2} [1 - \operatorname{sgn} \Delta f(x_k)] \Delta_{k-1} + \frac{1}{2} [1 + \operatorname{sgn} \Delta f(x_k)] a_i \operatorname{sgn} \Delta f(x_k) \operatorname{sgn} \Delta x_{k-1} \xi_k$$

### 1.2.3 Speciální metody

#### Lineární programování

Speciální úloha neklasického vázaného extrému – základní klasifikace:

$$f(x_1, \dots, x_n) = \sum_{i=1}^n c_i x_i = c^T x$$

vazební podmínky:

$$a_{11}x_1 + \dots + a_{1n}x_n \leq b_1$$

$$a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m$$

tedy  $Ax \leq b$

+ podmínka nezápornosti:  $x_i \geq 0$

Ekonomická interpretace - používá se k řešení ekonomických úloh – maximalizace zisku/minimalizace nákladů, časová optimalizace, ..., výrobní program se „převeď“ na matematickou úlohu, kde jednotlivé vlivy jsou nahrazeny vektory proměnných a vektory procesů  $f = \bar{c} \cdot \bar{x}$ .

#### Dynamické programování

- síťová forma – dopravní úloha
- tabulková forma – separovatelná účelová funkce

Bellmanův princip optimality:

Optimální trajektorie z libovolného bodu T do bodu Z nezáleží na trajektorii, která představuje cestu z bodu A do bodu T.

Poznámky:

- Řešení má vždy dva chody – přímý a zpětný
- Řešení nemusí být jednoznačné
- Při postupu se nikdy nesčítají a neporovnávají více jak 2 čísla

Analytický tvar dynamického programování – princip dělení zdrojů:

$$\operatorname{extr} f(x_1, \dots, x_N) = \sum_{i=1}^N f_i(x_i) \quad \text{separovatelná funkce – při omezení: } \sum_{i=1}^N x_i \leq b; \quad x_i \geq 0 [3]$$

## 1.2.4 Teorie her a optimálního rozhodování

### Klasifikace úloh teorie her

- *Nekonfliktní* rozhodování (jeden účastník)
- *Konfliktní* rozhodování (2 a více účastníků)
  - *Neinteligentní hráči*
    - Rozhodování při riziku
    - Rozhodování při neurčitosti
  - *2 inteligentní hráči*
    - Antagonistický konflikt
    - Neantagonistický konflikt ( $\Rightarrow$  Kooperativní, nebo nekooperativní – viz N inteligentních hráčů)
  - *N inteligentních hráčů*
    - Nekooperativní
    - Kooperativní
      - Přenosná výhra
      - Nepřenosná výhra

### Maticové hry

Matematický model konečného antagonistického konfliktu (ve kterém pro oba hráče existuje pouze konečný počet přípustných rozhodnutí).

### Diferenciální hry

Řeší hry v normálním tvaru, v nichž výplatní funkce závisí na strategiích hráčů prostřednictvím řešení soustavy diferenciálních rovnic.

## 2 STRUČNÝ ÚVOD DO GRAFŮ

V této části si ujasníme základní pojmy a postupy při tvorbě síťového grafu a zásady zobrazování jeho uzlů a činností.

*Grafem* rozumíme množinu bodů – uzlů  $\{P_0, P_1, \dots, P_n\}$  a množinu orientovaných hran  $\{(P_i, P_j)\}$ , spojujících některé dvojice těchto bodů. Hrana  $(P_i, P_j)$  má přitom počátek v uzlu  $P_i$  a konec v uzlu  $P_j$  (ve schématu ji značíme jako orientovanou úsečku) [1].

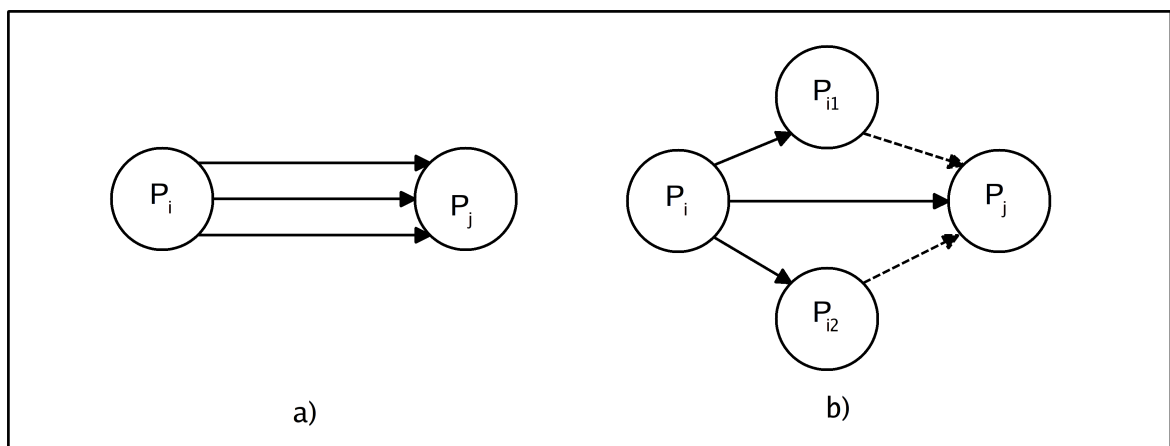
Graf nazýváme *symetrický*, když s danou hranou  $(P_i, P_j)$  obsahuje i symetrickou hranu  $(P_j, P_i)$  [1]. Pokud obsahuje jen jednu ze dvojice symetrických hran je graf *asymetrický*. V naší další práci budeme používat jen asymetrický graf.

Hrany, které mají počátek v  $P_i$  nazýváme výstupní z uzlu  $P_i$  a hrany, které mají konec v  $P_j$  nazýváme vstupní do uzlu  $P_j$ .

Pokud má graf jen jeden uzel  $P_0$ , který nemá vstupní hrany a pouze jeden uzel  $P_m$ , který nemá výstupní hrany a každé jeho hraně je přiřazeno číslo, nazýváme jej *sítí*. Posloupnost hran, ve které konec předchodí je začátkem další nazýváme *cesta*. Čísla připsaná hranám nazýváme délkami. Součet délek hran posloupnosti nazýváme *délkou cesty*.

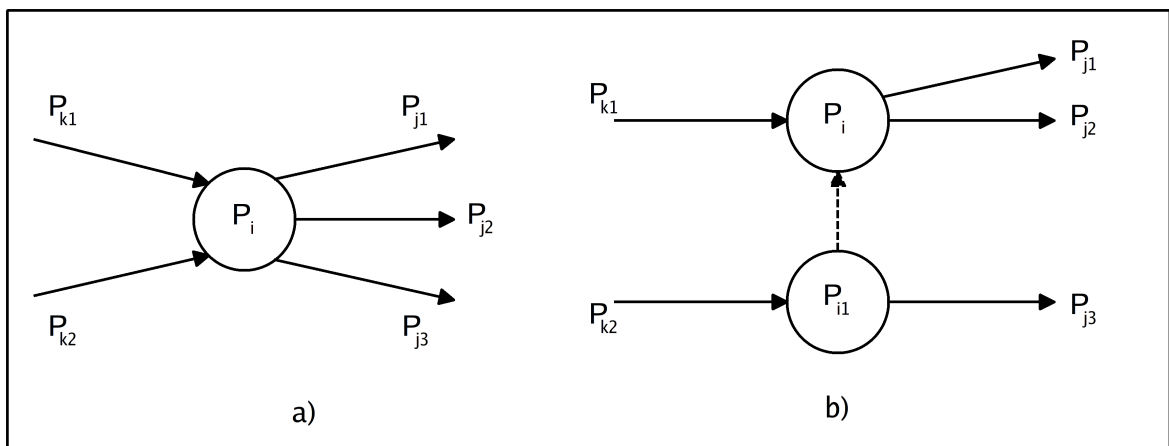
### 2.1 PRAVIDLA SESTAVENÍ SÍŤOVÉHO GRAFU

Pokud z jednoho uzlu vystupuje  $k$  činností a tytéž vstupují do jednoho uzlu, zavedeme pomocných  $(k - 1)$  uzlů a s konečným bodem je propojíme pomocí fiktivních činností o nulové délce (znázorňujeme čárkovaně) – viz obr. 1.



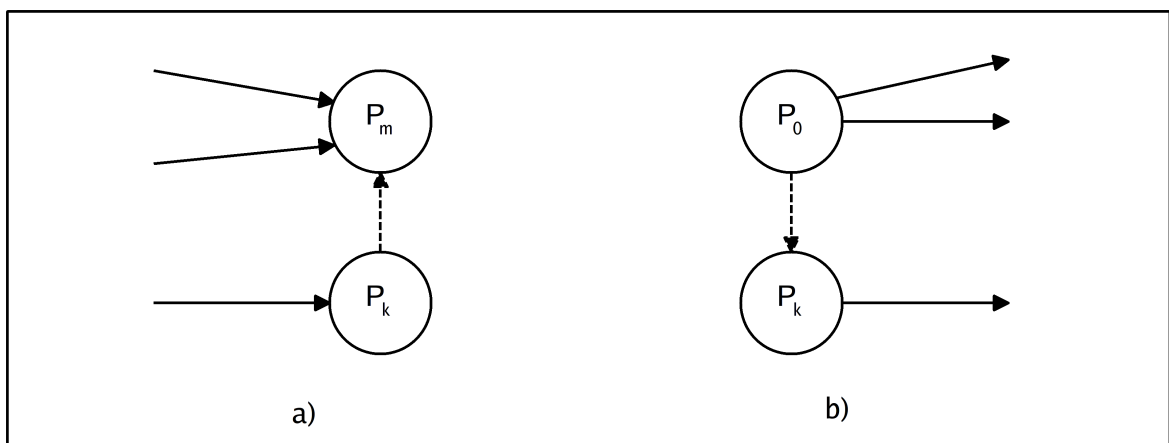
Obr. 1. Zavedení pomocných fiktivních činností s nulovou délkou trvání k oddělení činností vystupujících z jednoho uzlu a vstupujících společně opět do jednoho uzlu.

Pokud do uzlu  $P_i$  vstupuje více než jedna činnost a vystupuje z něj více než jedna činnost, ale pro provádění některých činností není nutné ukončení všech vstupních činností, zavedeme dodatečné uzly a fiktivní činnosti k oddělení těchto činností – viz obr. 2.



Obr. 2. Zavedení pomocných fiktivních činností s nulovou délkou trvání a uzlů k oddělení činností, vstupujících a vystupujících do/z jednoho uzlu, které na sebe nutně nenavazují.

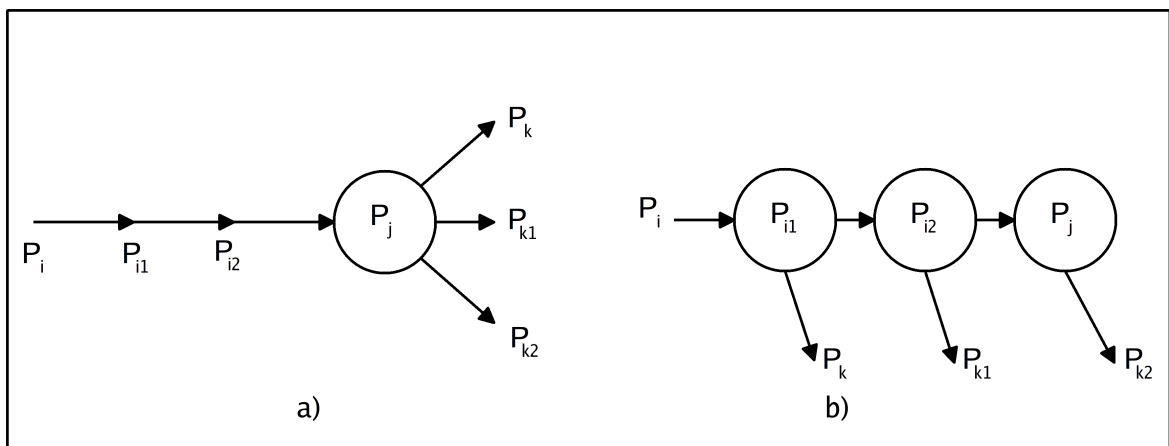
Pokud události vyjádřené uzlem  $P_k$  nepředchází žádná činnost, navážeme ji na počáteční uzel  $P_0$  fiktivní činností s nulovou délkou trvání. Pokud po události  $P_k$  nenásleduje žádná činnost, navážeme ji podobně na konečný uzel  $P_m$  – viz obr. 3.



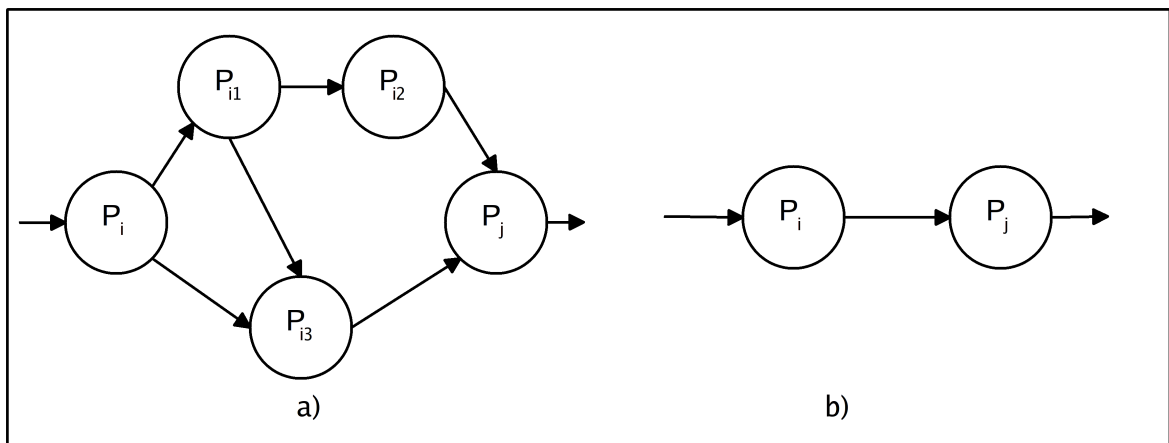
Obr. 3. Zavedení pomocných fiktivních činností s nulovou délkou trvání pro činnosti, které nemají a) předcházející, b) následující činnost.

Všechny činnosti v síťovém grafu musí být jednoduché. Pokud ale činnosti například  $(P_j, P_k)$ ,  $(P_j, P_k1)$  mohou začít po dokončení dílčích činností  $(P_i, P_j)$ , zavedeme dodatečné uzly  $P_{i1}$ ,  $P_{i2}$ , čímž rozdělíme složenou činnost a činnosti  $(P_j, P_k)$ , a  $(P_j, P_k1)$  připojíme v bodech, kdy může začít jejich činnost  $\Rightarrow P_{i1}, P_k, P_{i2}, P_k1$  – viz obr. 4.

Naopak, můžeme li několik činností, nebo část projektu, provést nezávisle na zbytku projektu, sestavujeme *agregovanou* síť, kde dílčí – nezávislou část projektu nahradíme jedinou činností o délce trvání dílčí části projektu. Tento způsob je používán především v rozsáhlých sítích a umožňuje rozdělit graf na několik dílčích síťových grafů a plánovat po částech – viz obr.5.



Obr. 4. Zavedení pomocných fiktivních činností s nulovou délkou trvání pro činnosti, které nemají a) předcházející, b) následující činnost.



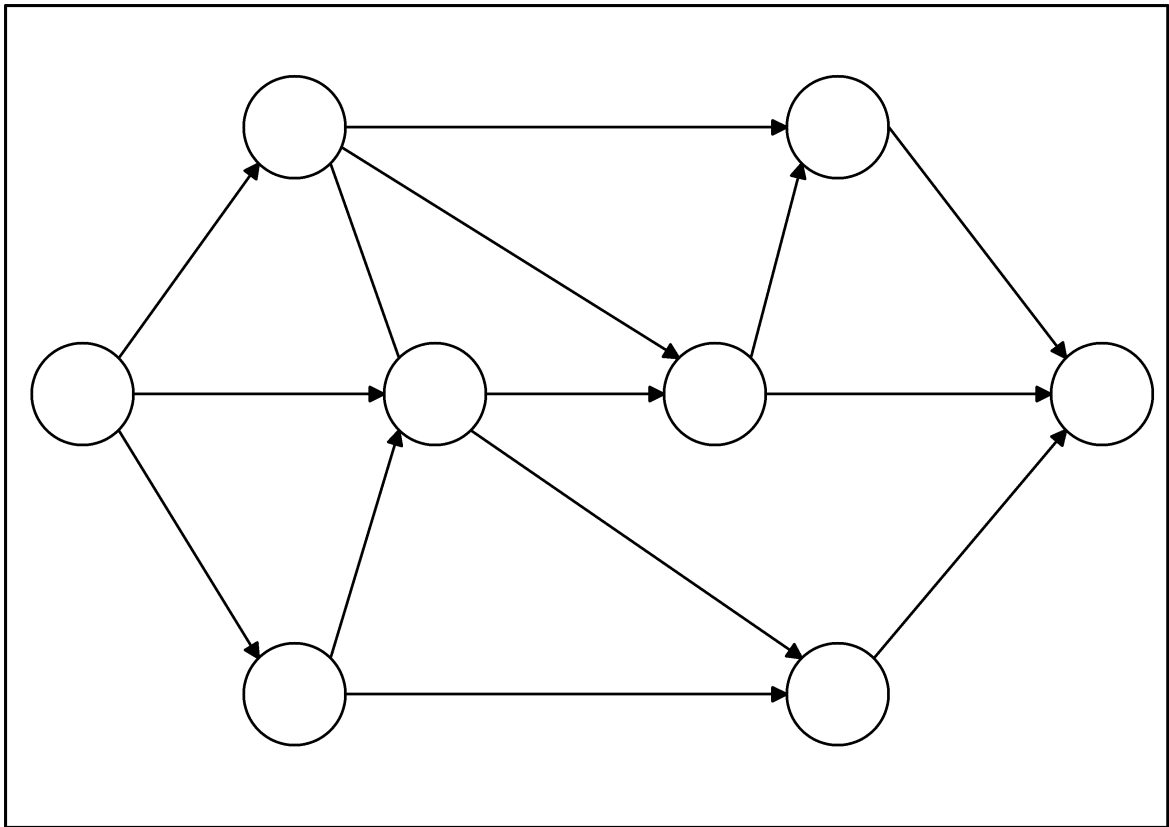
Obr. 5. Příklad sestavení agregované sítě – nahrazení dílčí části grafu (nezávislé na ostatních činnostech) jednou činností.

## 2.2 ČÍSLOVÁNÍ UZLŮ

Cílem přečíslování je, aby pro libovolnou činnost  $(P_i, P_j)$  byla splněna podmínka nerovnosti:  $i < j$ , což usnadní další práci s grafem.

Pro jednodušší síťové grafy provedeme přečíslování pomocí *metody přeškrťování hran*. Začneme tím, že si určíme uzel  $P_0$ , to je takový, který nemá žádné vstupní hrany. Přeškrtneme všechny hrany, vystupující z uzlu  $P_0$  a uzel, který zůstane po tomto kroku bez vstupních hran označíme jako uzel  $P_1$ . V dalších krocích opakujeme přeškrťování výstupních hran z právě očíslovaného uzlu a určujeme další následující, dokud nedojdeme k uzlu  $P_m$  bez dalších výstupních hran. Takto stavěný graf se však v praxi příliš nevyskytuje.

Nejběžnější případy grafů jsou silně rozvětveny, proto postupujeme pomocí kombinace *přeškrťování hran a určování řádů* jednotlivých uzlů. Ukázkový graf – viz obr. 6. Začneme jako u předchozího případu určením uzlu  $P_0$  bez vstupních hran – toto bude uzel *nultého řádu*. Všechny uzly, které po přeškrťování hran vycházejících z uzlu  $P_0$  nemají žádné vstupní hrany

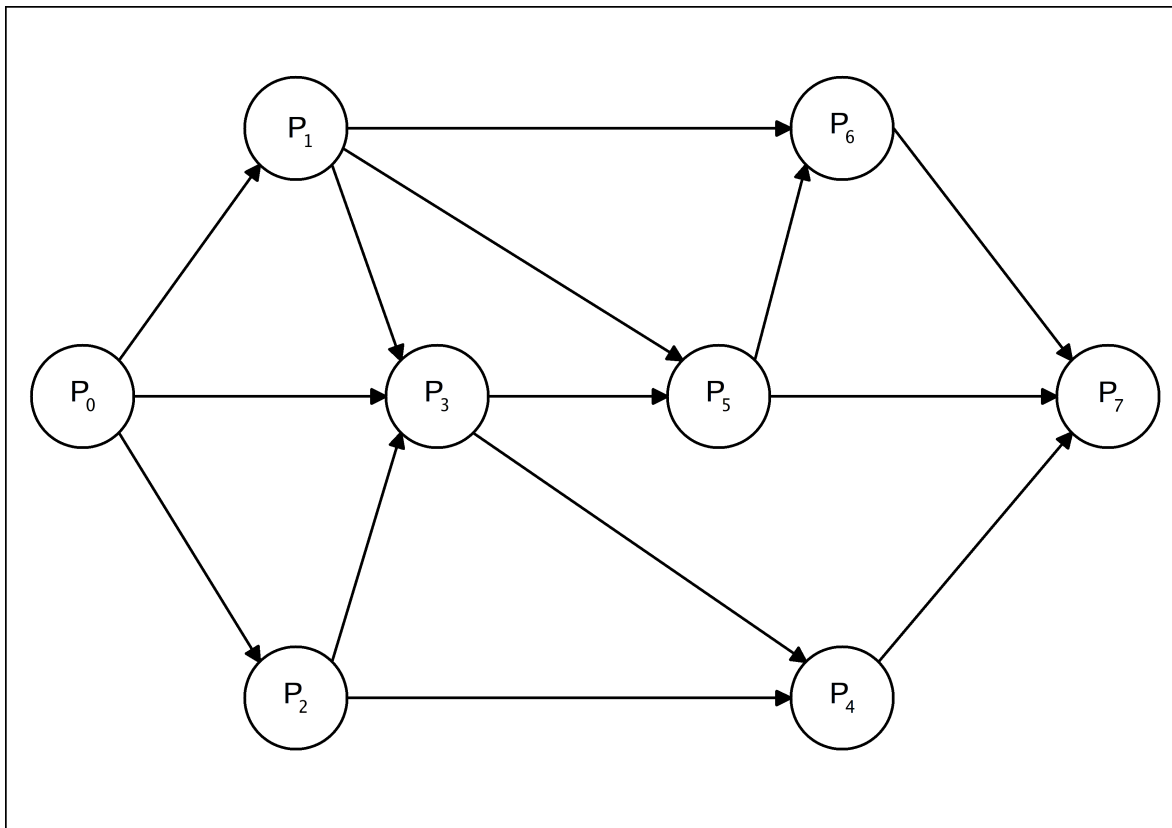


Obr. 6. Ukázka síťového grafu před přečíslováním.

nazveme uzly *prvního řádu*. V dalším kroku opakujeme přeškrtnutí hran vystupujících z uzlů prvního řádu a nalezneme uzly *druhého řádu*. Pro kontrolu – vidíme, že maximální počet hran cesty z uzlu  $P_0$  do uzlů prvního řádu je 1, do uzlů druhého řádu jsou to 2 hrany. Obecně můžeme říct, že maximální počet hran cesty spojující uzel  $i$ -tého řádu s uzlem  $P_0$  (uzlem nultého řádu) roven 1.

Očíslování uzlů nyní provedeme následujícím způsobem:

Jedinému uzlu  $P_0$  nultého řádu přiřadíme číslo 0. Uzlům prvního řádu přiřadíme čísla  $1, 2, \dots, k_1$ , uzlům druhého řádu čísla  $k_1 + 1, k_1 + 2, \dots, k_1 + k_2$  a tak pokračujeme do vyčerpání všech uzlů. Protože uzly stejného řádu nejsou navzájem spojeny a uzly nižšího řádu mají menší index, pak v očíslované síti pro libovolnou hranu  $(P_i, P_j)$  vždy platí  $i < j$  [1]. Přečíslovaný graf z obr. 6 – viz obr. 7. Postup z kapitoly o číslování uzlů může odhalit i možný výskyt cyklů – bohužel však jen z důvodu jeho nevhodnosti pro síťové grafy s nimi. Po několika krocích se totiž dostaneme na „úroveň“ vyšší, než je skutečně možný počet úrovní v grafu. V takovém případě je nutno buď použít algoritmus podporující grafy s cykly, nebo ošetříme graf tak, aby k zacyklování nedocházelo.



Obr. 7. Ukázka přecíslovaného síťového grafu.

### 2.3 FORDŮV ALGORITMUS (PRO ČÍSLOVÁNÍ UZLŮ V ROZSÁHLÝCH SÍTÍCH)

#### Počáteční krok

Každému uzlu  $P_j$  přiřadíme číslo  $\lambda_j^{(0)} = 0$  a každé hraně  $(P_i, P_j)$  přiřadíme číslo  $y_{ij} = 1$ , které zůstává stejné během celého algoritmu.

#### Obecný $q$ -tý krok

Probírají se všechny uzly ve zvoleném pořadí (např.  $P_0, P_1, \dots, P_n$  a čísla  $\lambda_j^{(q-1)}$ , vypočtená v předchozím kroku, se nahrazují novými čísly  $\lambda_j^{(q)} \geq \lambda_j^{(q-1)}$  podle vzorce:

$$\lambda_j^{(q)} = \max_i \{ \lambda_i^{(p)} + y_{ij} \} \quad (j = 1, \dots, n; \lambda_0^{(q)} = 0),$$

kde  $p$  je rovno  $q$  nebo  $q - 1$  v závislosti na tom, probíral-li se uzel  $P_i$  při  $q$ -tém kroku, nebo se ještě neprobíral. Při každém kroku je možno probírat všechny uzly v libovolném pořadí.

Obecný krok se opakuje do té doby, až v  $q$ -tém kroku zůstanou beze změny všechna  $\lambda_j^{(q-1)}$ . Potom čísla:

$$\lambda_j^* = \lambda_j^q = \lambda_j^{(q-1)} \quad (j = 0, 1, \dots, n),$$

jsou řády  $j$ -tých uzlů sítě.

### 2.4 VYHLEDÁVÁNÍ CYKLŮ

Obzvláště v rozsáhlých sítích se může vyskytnout zacyklení některých činností. To způsobuje problémy při výpočtech. Už u základního algoritmu pro přecíslování může dojít

k zacyklení a znemožnění graf přecíslovat. Pro vyhledávání cyklů používáme následující algoritmus:

*Přípravný krok* – Každému uzlu  $P_j$  přiřadíme dvě čísla  $\xi_j$  a  $\lambda_j$  která se v průběhu algoritmu mění. Každé činnosti  $(P_i, P_j)$  přiřadíme její trvání  $t_{ij}$ .  $\xi_j$  se rovná počtu neprobraných hran, které vstupují do  $P_j$ . Číslo  $\lambda_j$  je nalezené maximální trvání cesty z  $P_0$  do  $P_j$  (na začátku všechna  $\lambda_j = 0$ ). Následují střídavě kroky:

*1. krok:*

Začínáme od koncového uzlu  $P_m$  ( $\xi_m \neq 0$ ) a snažíme se sestavit cestu do některého uzlu  $P_1$ , pro který  $\xi_j = 0$  (bereme hrany v obráceném směru). Probíráme postupně činnosti vstupující do  $P_m$ , označíme si vybranou činnost v grafu (například  $(P_{i_1}, P_m) \sim$ ). Od  $\xi_m$  odečteme 1 a označíme  $P_m$  symbolem „\*“. Přejdeme k uzlu  $P_{i_1}$ . Pokud  $\xi_{j_1} = 0$ , přejdeme k *2. kroku*. Pokud ne, vyhledáme další neprobranou činnost  $(P_{j_1}, P_{j_2})$  vstupující do  $P_{i_1}$ , označíme ji symbolem  $\sim$ , od  $\xi_{j_1}$  odečteme 1 a označíme  $P_{j_1}$  symbolem „\*“. Přejdeme k dalšímu uzlu  $(P_{j_2}) \dots$

Nakonec buď přijdeme k uzlu  $P_i$  pro který  $\xi_i = 0$  a přejdeme ke *2. kroku*, nebo narazíme na uzel, který jsme již zpracovávali, takže nalezneme cyklus.

*2. krok:*

Postupujeme nyní z  $P_i$ , pro které  $\xi_i = 0$ , do  $P_m$  po označených stranách (v tomto kroku po směru). Z uzlu  $P_i$  postoupíme po hraně  $(P_i, P_{j_k})$ , vypočteme podle vzorce  $\lambda'_{j_k} = \max\{\lambda_{j_k}, \lambda_i + t_{ij_k}\}$  a nahradíme  $\lambda'_{j_k}$  číslem  $\lambda_{j_k}$ . Pokud  $\xi_{j_k} = 0$ , pak  $\lambda'_{j_k} = \lambda_{j_k}^* = T_{j_k}^{(0)}$  a smažeme označení všech hran vstupujících do  $P_{j_k}$  i označení  $P_{j_k}$ . Potom přejdeme dál po navazující hraně  $(P_{j_k}, P_{j_{k-1}}) \dots$ . Pokud  $\xi_{j_k} \neq 0$  pokračujeme od  $P_{j_k}$  po dalších neprobraných cestách proti směru hran – to znamená podle *1. kroku*. Uzel  $P_{j_k}$  považujeme za výchozí, ale označení hran a uzlů necháváme.

*Algoritmus je ukončen* (pokud se nenašel cyklus) pokud všechna  $\xi_j = 0$  a všechna  $\lambda_j = \lambda_j^* = T_j^{(0)}$ . Pokud se najde cyklus, nemá význam počítat  $\lambda'_j$ . V tomto případě algoritmus končí pokud všechna  $\xi = 0$  a slouží jen k určení cyklů.



## **II. ANALYTICKÁ ČÁST**

### 3 KRITICKÁ CESTA

#### 3.1 TRVÁNÍ PROJEKTU A KRITICKÁ CESTA

Jak jsme si řekli, v síťovém grafu máme celý projekt znázorněn jednotlivými uzly spojenými hranami jednotlivých činností. Označení délky hran vyjadřují jejich dobu trvání. *Trváním cesty* budeme rozumět součet trvání všech činností tvořících tuto cestu.

*Termínem uzlu*  $T_j$  rozumíme čas ukončení všech činností vstupujících do tohoto uzlu [1].

*Trváním projektu* nazveme nejdříve možný termín uzlu  $P_n$  (koncový uzel) a označíme  $T_n^{(0)}$ . Jinak řečeno trvání projektu je *minimální počet časových jednotek, nutných k vyplnění komplexu všech činností* [1].

Doba trvání projektu je tedy rovna maximální délce cesty z  $P_0$  do  $P_n$ . Takovou cestu budeme nazývat *kritickou cestou*, nebo také *kritickou posloupností*.

##### 3.1.1 Nejdříve možné začátky činností

Pro určování nejdříve možných začátků uzlu vyjdeme ze základního algoritmu, který jsme použili pro pro určení řádů uzlů před přečíslováním síťového grafu.

*Základní krok:*

Každému uzlu  $P_j$  připíšeme číslo  $\lambda_j = 0$ , ale místo čísla  $y_{ij}$  přiřadíme číslo  $t_{ij}$  – trvání činnosti.

*Obecný krok:*

Projdeme celou přečíslovanou síť od  $P_0$  do  $P_n$  a vypočteme nové hodnoty čísel  $\lambda_j$  podle vzorce:

$$\lambda'_j = \max_i \{\lambda'_i + t_{ij}\} \quad (j = 1, 2, \dots, n) \quad \lambda'_0 = \lambda_0 = 0 \quad (1)$$

a zapíšeme ke každému vypočtenému  $\lambda'_j$  čísla  $i_1, i_2, \dots, i_v$  těch událostí, pro které podle vzorce (1) bylo dosaženo maxima. Toto označení bude výchozí pro hledání kritické cesty.

Přesvědčíme se, že se po druhém kroku nezmění ani jedno z čísel  $\lambda'_j = \lambda_j^{(1)}$  z prvního kroku.

Síť procházíme v pořadí  $P_0, P_1, \dots, P_i, \dots, P_j, \dots, P_n$ , proto počítáme číslo  $\lambda'_j$  až po dokončení výpočtů všech  $\lambda'_i$ , potřebných pro výpočet  $\lambda'_j$  ( $i < j$ ). Čísla  $t_{ij}$  jsou stejná při všech krocích – viz (1). Číslo  $\lambda_j^{(2)}$  může být větší než  $\lambda_j^{(1)}$  pouze v případě, že  $\lambda_i^{(2)} > \lambda_i^{(1)}$  pro některé  $i < j$ , ale  $\lambda_0^{(2)} = \lambda_0^{(1)} = 0$  a také  $\lambda_1^{(2)} = \lambda_1^{(1)}$ ,  $\lambda_2^{(2)} = \lambda_2^{(1)}$  a tak dále.

Každé číslo  $\lambda'_j$  ( $j = 1, 2, \dots, n$ ), získané na základě jediného kroku, určuje maximální trvání cesty z  $P_0$  do  $P_j$ , což je rovno nejdříve možnému termínu  $T_j^{(0)}$  uzlu  $P_j$ . Konečné číslo  $\lambda'_n = T_n^{(0)}$  určuje trvání projektu.

### 3.1.2 Určení kritické cesty

Všechny hrany ležící na *kritické cestě* musí vyhovovat podmínce<sup>1)</sup>:

$$T_j^{(0)} - T_i^{(0)} - t_{ij} = 0 \quad (2)$$

Pokud podmínka (2) platí pro každou hranu cesty  $(P_0, P_{i_1}, P_{i_2}, \dots, P_{i_k}, P_n)$ , pak pro trvání cesty dostáváme:

$$t_{0i_1} + t_{i_1i_2} + \dots + t_{i_k n} = T_{i_1}^{(0)} - T_0^{(0)} + T_{i_2}^{(0)} - T_{i_1}^{(0)} + \dots + T_n^{(0)} - T_{i_k}^{(0)} = T_n^{(0)} - T_0^{(0)} = T_n^{(0)}$$

(vzhledem k tomu, že  $T_0^{(0)} = \lambda_0 = 0$ ).

Splnění podmínky (2) je zcela postačující k určení, zda činnost  $(P_i, P_j)$  leží na kritické cestě [1].

Při vyhledávání kritické cesty začínáme u hran vstupujících do posledního uzlu  $P_n$ . Z nich vybereme ty, které vyhovují podmínce (2). Pak pokračujeme uzly, které jsme v tomto kroku vybrali a v následujících uzlech opět hledáme činnosti – hrany vyhovující podmínce (2). Pokračujeme tak dlouho, dokud nedojdeme k počátečnímu uzlu  $(P_0)$ , který nemá vstupní hrany.

Postup od uzlu  $P_n$  zaručuje, že vždy můžeme vybrat hranu, která bude vyhovovat podmínce (2), vzhledem k definici nejdříve možného termínu:

$$T_{i_1}^{(0)} = \max_i \{\lambda'_i + t_{ii_1}\} = \lambda'_{i_2} + t_{i_2i_1} = T_{i_2}^{(0)} + t_{i_2i_1}.$$

Pokud bychom nepostupovali od uzlu  $P_n$ , mohla by nastat situace, kdy z uzlu  $P_{i_0}$  nevystupuje žádná hrana splňující podmínku (2).

Rozdíl  $T_j^{(0)} - t_i^{(0)} - t_{ij}$  nazveme *volnou rezervou činnosti*  $(P_i, P_j)$ . Pro činnosti, které neleží na kritické cestě, bude platit  $T_j^{(0)} - t_i^{(0)} - t_{ij} \geq 0$ .

### 3.2 NEJPOZDĚJI PŘÍPUSTNÉ TERMÍNY UZLŮ

Výpočet nejdříve možných termínů uzlů jsme si již ukázali. Začátek některých činností může však být posunut, aniž by došlo k prodloužení kritické cesty  $T_n^{(0)}$  (nebo jiné zadané doby  $T_n$  dokončení projektu). To je možné, pokud maximální trvání cesty z  $P_j$  do  $P_n$  je menší než  $T_n - T_j^{(0)}$ .

Definujme *nejpozději přípustný termín*  $T_j^{(1)}$  uzlu  $P_j$  tak, že skončení všech činností, vstupujících do  $P_j$  v termínu  $T_j^{(1)}$ , nenarušuje skončení všech činností v termínu  $T_n$ , jinak řečeno  $T_j^{(1)}$  je nejpozději přístupný konec všech činností vstupujících do  $P_j$ . Je zřejmé, že  $T_j^{(1)}$  je rovno rozdílu mezi  $T_n$  a maximálním trváním cesty z  $P_j$  do  $P_n$ .

Pro získání nejpozději přípustných termínů uzlu  $T_j^{(1)}$  můžeme použít algoritmus pro určení nejdříve možných začátků činností.

<sup>1)</sup>Jak vyplývá z kapitoly (3.1.1), obecně pro libovolnou hranu  $(P_i, P_j)$  platí  $T_j^{(0)} - T_i^{(0)} \geq t_{ij}$

Nejdříve vypočteme maximální trvání cest z  $P_1$  do  $P_n$  ( $j = n-1, n-2, \dots, 0$ ), to znamená přidělíme každému uzlu v přečíslované síti číslo  $\mu'_j = 0$  a projdeme potom síť od  $P_n$  do  $P_0$  a vypočteme nová  $\mu'_j$  na základě vzorce:

$$\mu'_j = \max_k \{\mu'_k + t_{jk}\} \quad (\mu'_n = \mu_n = 0, j = n-1, n-2, \dots, 0) \quad (3)$$

Nejpozději nutný termín  $T_j^{(1)}$  dostaneme na základě vzorce:

$$T_j^{(1)} = T_n - \mu'_j \quad (j = n, n-1, \dots, 0) \quad (4)$$

### 3.3 PODMÍNKY KRITičNOSTI UZLU A ČINNOSTÍ

*Kritický uzel* – Uzel  $P_j$  leží na kritické cestě, když a jen když je splněno [1]:

$$T_j^{(1)} = T_j^{(0)}. \quad (5)$$

Podle definice  $T_j^{(1)} = T_n(0) - \mu_j^*$  a podle 5 platí:

$$T_n^{(0)} = T_j^{(0)} + \mu_j^*. \quad (6)$$

$T_j^{(0)}$  je maximální trvání cesty z  $P_0$  do  $P_j$  a  $\mu_j^*$  je maximální trvání cesty z  $P_j$  do  $P_n$   $\Rightarrow T_j^{(0)} + \mu_j^*$  je nejdelší cesta z  $P_0$  do  $P_n$  procházející uzlem  $P_j$ . Pokud  $P_j$  leží na kritické cestě,  $T_j^{(0)} + \mu_j^* = T_n^{(0)}$  – dle vztahu (5). Pokud je toto dokázáno, cesta procházející uzlem je *kritická*.

Činnost  $(P_i, P_j)$  leží na kritické cestě, tehdy a jen tehdy, když

$$T_j^{(1)} - T_i^{(0)} - t_{ij} = 0. \quad (7)$$

Vzhledem ke vztahu  $T_j^{(1)} = T_n^{(0)} - \mu_j^*$ , můžeme napsat podmínku (7) ve tvaru

$$T_j^{(1)} - T_n^{(0)} - t_{ij} = T_n^{(0)} - (T_i^{(0)} + t_{ij} + \mu_j^*) = 0 \quad (8)$$

Kde součet  $T_i^{(0)} + t_{ij} + \mu_j^*$  vyjadřuje maximální trvání cesty z  $P_0$  do  $P_n$  procházející činností  $(P_i, P_j)$  ( $T_i^{(0)}$  je maximální trvání cesty z  $P_0$  do  $P_i$  a  $\mu_j^*$  je maximální trvání cesty z  $P_j$  do  $P_n$ ).

Činnost  $(P_i, P_j)$  pro kterou platí podmínka 7, to znamená pro kterou je nejdelší možná doba  $T_j^{(1)} - T_i^{(0)}$  je rovná jejímu trvání  $t_{ij}$  nazýváme *kritickou činností*.

Všechny činnosti na kritické cestě jsou kritické a naopak všechny kritické činnosti leží na kritické cestě [1]. Po získání všech kritických činností můžeme určit všechny kritické cesty, pokud vyjdeme z uzlu  $P_0$  a vybereme-li z každého uvažovaného uzlu kritickou činnost.

Činnost která nesplňuje podmínku (7) nazýváme *nekritickou činností*.

### 3.4 ČASOVÉ REZERVY

*Celková rezerva* činnosti je maximální počet časových jednotek, který máme k dispozici pro provedení činnosti  $(P_i, P_j)$  (pokud to dovolí charakter činnosti), aniž se prodlouží celkový čas trvání projektu  $T_n$ . To mimo jiné znamená, o kolik můžeme zpozdít začátek činnosti  $(P_i, P_j)$  proti nejdříve možnému termínu  $t_i^{(0)}$  aniž se prodlouží trvání  $T_n$  projektu [1]. Je vyjádřena:  $T_j^{(1)} = T_i^{(0)} - t_{ij}$ .

Mimo to je celková rezerva činnosti rovna rozdílu mezi termínem dokončení projektu a maximálním trváním cesty, která prochází danou činností [1].

U kritických činností je celková rezerva rovna nule.

Prodloužením trvání nekritické činnosti na účet celé celkové rezervy se vytvoří nová kritická cesta. „Posunutá“ činnost leží na nové kritické cestě.

Zpoždění začátku nekritické činnosti  $(P_i, P_j)$  vzhledem k  $T_i^{(0)}$  o celou rezervu způsobí, že všechny činnosti vystupující z uzlu  $P_j$  musí začínat v nejpozději přípustném termínu  $T_j^1$  tohoto uzlu.

Volná rezerva  $T_j^{(0)} - T_i^{(0)} - t_{ij}$  činnosti  $(P_i, P_j)$  je maximální přípustné prodloužení trvání této činnosti (nebo zpoždění jejího začátku vzhledem k termínu  $T_i^{(0)}$ ), které nezpůsobí, že by všechny činnosti  $(P_j, P_l)$  vystupující z  $P_j$  nemohly začít v nejdříve možném termínu  $T_j^{(0)}$ .

Prodloužení trvání činnosti  $(P_i, P_j)$  o část volné rezervy může zmenšit celkovou rezervu činností  $(P_k, P_i)$  vstupujících do uzlu  $P_i$ , ale nemá vliv na rezervy činností vystupujících z uzlu  $P_j$ . U uzlů, které leží na kritické cestě je celková rezerva činností končících v těchto uzlech rovna volné rezervě.

Nezávislé rezervy činností značí maximální počet časových jednotek, o které můžeme prodloužit činnost  $(P_i, P_j)$ , nebo zpozdít její počátek tak, aby všechny činnosti  $(P_k, P_i)$  končily v nejpozději přípustném termínu  $T_i^{(1)}$  a všechny činnosti  $(P_j, P_l)$  vystupující z  $P_j$  začínaly v nejdříve možném termínu  $T_j^{(0)}$ . To znamená, že využívání nezávislé rezervy nemá vliv na rezervy ostatních činností a nezávislá rezerva činnosti vycházející z uzlu na kritické cestě je rovna volné rezervě.

### 3.5 SUBKRITICKÉ ČINNOSTI

Pokud máme vyčíslenou celkovou rezervu, můžeme určit *subkritické činnosti*, tedy všechny takové, které leží na cestách, jejichž trvání  $L$  se liší od kritické nejvýše o danou hodnotu odchylky  $\delta$ .

$$T_{kr} - \delta \leq L \leq T_{kr}$$

Subkritické cesty je možno vyhledat, jestliže při výpočtu  $T_i^{(0)}$  a  $\mu^*$  vedle nalezeného čísla zapíšeme čísla těch uzlů, kterým odpovídají příslušná  $T_i^{(0)}$  nebo  $\mu_i^*$ . Tímto způsobem však najdeme jen maximální subkritické cesty procházející danou činností.

Jestliže je nutné vyhledat všechny subkritické cesty, musíme zřejmě přikročit k probrání všech cest, sestavených pouze ze subkritických činností.

Zvláštního významu nabývá úloha o subkritických činnostech v momentě, kdy termín ukončení projektu  $T$  je menší než  $T_{kr}$ . V takovém případě je nutné přistoupit kromě zkracování kritických činností i ke zkracování činností subkritických – musí být splněna nerovnost  $T < L \leq T_{kr}$ .

Při tomto pohledu na cesty můžeme narazit na dvě se stejnou celkovou rezervou, z nichž každá může být rozdělena do úseků různé délky o různé napjatosti splnění těchto činností, nebo můžeme narazit na činnosti s různými celkovými rezervami, ale provádějí se se stejnou

napjatostí. Zavádíme proto nový parametr *koeficient napjatosti*  $k_{ij}$ , vyjadřující právě napjatost splnění každé činnosti  $(P_i, P_j)$ .

Při určování koeficientu  $k_{ij}$  činnosti  $(P_i, P_j)$  vyhledáme zleva i zprava uzly jedné a téže kritické cesty nejbližší k této činnosti a utvoříme poměr trvání  $L'(i, j)$  – úseku maximální cesty, který prochází činností  $(P_i, P_j)$  mezi nalezenými uzly, trvání  $T'_{kr}(i, j)$  úseku kritické cesty mezi těmito událostmi. V grafu s více kritickými cestami může být samozřejmě i více těchto poměrů, proto na koeficient napjatosti  $k_{ij}$  považujeme ten největší. Tedy – koeficientem napjatosti se nazývá maximální poměr trvání nekryjících se úseků cesty maximálního trvání a kritické cesty, uzavřených mezi stejnými uzly, které patří oběma cestám:

$$\frac{L'(i, j)}{T'_{kr}(i, j)}$$

Pokud jsou na cestě maximálního trvání  $L(i, j)$  procházející činností  $(P_i, P_j)$ , jsou dva uzly stejné kritické cesty (různé od  $P_0$  a  $P_n$ ), potom zleva od první z nich a zprava od druhé se cesta maximálního trvání kryje s kritickou cestou. Označíme trvání kryjících se částí obou cest  $T''_{kr}(i, j)$ , potom:

$$\begin{aligned} \frac{L'(i, j)}{T'_{kr}(i, j)} &= \frac{L'(i, j) + T''_{kr}(i, j) + [T''_{kr}(i, j) + T'_{kr}(i, j)] + T'_{kr}(i, j)}{T'_{kr}(i, j)} = \\ &= \frac{L(i, j) + T_{kr} + T'_{kr}(i, j)}{T'_{kr}(i, j)} = 1 - \frac{T_{kr} - L(i, j)}{T'_{kr}(i, j)} = 1 - \frac{R_n(i, j)}{T'_{kr}(i, j)}, \end{aligned}$$

kde  $R_n(i, j)$  je celková rezerva činnosti  $(P_i, P_j)$ . Proto je také možné definovat  $k_{ij}$  jako největší z čísel  $1 - \frac{R_n(i, j)}{T'_{kr}(i, j)}$ . Odtud také odvodíme, že pro kritické činnosti je  $k_{ij} = 1$  a pro subkritické činnosti se k jedničce blíží.

### 3.6 METODY REALIZACE ALGORITMU VÝPOČTU $T_j^{(0)}$ A $T_j^{(1)}$

#### 3.6.1 Maticová metoda

Při výpočtech touto metodou nejdříve sestavíme pro přečíslovaný graf následující tabulku (pokud máme graf dobře přečíslovaný, budou v této tabulce čísla  $t_{ij}$  rozložena pouze nad diagonálou):

$P_i/P_j$	$P_0$	$P_1$	...	$P_i$	...	$P_j$	...	$P_n$
$P_0$		$t_{01}$		$t_{0i}$		$t_{0j}$		$t_{0n}$
$P_1$				$t_{1i}$		$t_{1j}$		$t_{1n}$
$\vdots$								
$P_i$		$t_{i1}$				$t_{ij}$		$t_{in}$
$\vdots$								
$P_j$		$t_{j1}$		$t_{ji}$				$t_{jn}$
$\vdots$								
$P_n$		$t_{n1}$		$t_{ni}$		$t_{nj}$		

Dále připojíme zprava sloupce  $\lambda^{(0)}, \lambda^{(1)}, \dots$  a zdola řádky  $\mu^{(0)}, \mu^{(1)}, \dots$ . Počet připojených sloupců/řádků je roven počtu kroků nutných pro ukončení  $\lambda_j^*(\mu_j^*)$

V prvním kroku vyplníme sloupec  $\lambda^{(0)}$  nulami. Potom vyplníme sloupec  $\lambda^{(1)}$ . Pro získání čísla  $\lambda_j^{(1)}$  sčítáme čísla  $t_{ij}$  zapsaná ve sloupci  $P_j$ , s odpovídajícími (dříve vypočtenými) čísly  $\lambda_i^{(p)}$  (kde  $p = 1$ , jestliže  $\lambda_i^{(1)}$  již bylo vypočteno a  $p = 0$  pokud ne). Nejvyšší součet se zapíše jako  $\lambda_j^{(1)}$  do sloupce  $\lambda^{(1)}$ . Pokračujeme se sloupcem  $\lambda^{(2)}$  (nyní je  $p = 1$ , nebo  $p = 2$ ). Pokračujeme, dokud nedojdeme ke sloupci  $\lambda^*$  (tedy k takovému, který bude shodný s předchozím).

Řádky  $\mu^{(0)}, \mu^{(1)}, \dots$  se vyplňují obdobně. V řádku  $\mu^{(0)}$  položíme  $\mu_j^{(0)} = 0$  pro všechna  $j$ . Čísla v řádku  $\mu^{(1)}$  dostaneme jako součet čísel  $t_{jk}$  zapsaná v řádku  $P_j$  s odpovídajícími vypočtenými čísly  $\mu_k^{(p)}$  (opět  $p = 1$  pro vypočítané a  $p = 0$  pro dosud nespočítané. Výsledkem je opět nejvyšší součet. Vyplňování opět ukončí nalezení řádku  $\mu^*$  (první totožný s předchozím).

Pokud je  $T_{kr}$  termínem dokončení projektu, jsou hodnoty  $T_{kr}, T_j^{(0)}$  a  $T_j^{(1)}$  určeny vztahy:

$$T_{kr} = \max_j \lambda_j^* = \max_j \mu_j^*; \quad T_j^{(0)} = \lambda_j^*; \quad T_j^{(1)} = T_{kr} - \mu_j^*; \quad (j = 0, j, \dots, n).$$

Pokud nemáme síť očíslovanou uspořádaně, nezáleží na pořadí výpočtů, pokud je očíslovaná uspořádaně, můžeme obdržet všechna čísla  $\lambda_0^*, \lambda_1^*, \dots, \lambda_n^*$  ( $\mu_0^*, \mu_1^*, \dots, \mu_n^*$ ) v jednom kroku.

Pokud si vedle vypočtených  $\lambda_j^{(q)}(\mu_j^{(q)})$  zapisujeme v závorkách čísla  $i(k)$  uzlů  $P_i(P_k)$ , pro které byla dosažena  $\lambda_j^{(q)}(\mu_j^{(q)})$ , tak po výpočtu  $\lambda^*$  a  $\mu^*$ , můžeme od výchozího uzlu  $P_s(P_0)$  pro který

$$T_s^{(0)} = \max_j T_j^{(0)} = T_{kr}$$

snadno určit celou kritickou cestu [1].

Při výpočtech maticovou metodou je vhodné použít (při známém termínu  $T_n$  ukončení projektu) vzorec

$$T_n^{(1)} = T_n; \quad T_j^{(1)} = \min_k \{T_k^{(1)} - t_{jk}\} \quad (j = n-1, n-2, \dots, 0)$$

Potom postupujeme tak, že v tabulce dole přepíšeme řádek ( $T^{(1)}$ ), který se postupně zaplňuje zprava od  $T_n^{(1)} = T_n, T_{n-1}^{(1)}, \dots, T_j^{(1)}, \dots, T_0^{(1)}$ . Číslo  $T_j^{(1)}$  spočítáme tak, že ze sloupce  $P_j$  odečteme od odpovídajících čísel řádku  $T^{(1)}$  a nejmenší ze získaných rozdílů je roven  $T_j^{(1)}$  toto číslo doplníme do políčka v řádce  $T^{(1)}$  a sloupci  $P_j$ .

Tento algoritmus se dá přizpůsobit i pro neuspořádaně očíslovanou síť.

### 3.6.2 Výpočet v síťovém grafu

V menším počtu činností můžeme výpočty provádět přímo v grafu. Pro tento typ výpočtu si znázornění každého uzlu rozdělíme na čtvrtiny. Vrchní část je označena pro číslo uzlu, levá pro vypočtené  $T_j^{(0)}$ , dolní pro číslo  $i_1, \dots, i_v$  uzlů pro které bylo dosaženo hodnoty zapsané vlevo a vpravo zapisujeme zjištěné hodnoty  $T_j^{(1)}$ .

U přečíslované sítě postupujeme podle pořadí uzlů a pro každý uzel  $P_j$  vzhledem k činnostem vstupujícím do tohoto uzlu vypočteme  $T_j^{(0)}$  podle vzorce:

$$T_j^{(0)} = \max_i \{T_i^{(0)} + t_{ij}\}, \quad T_0^{(0)} = 0 \quad (j = 1, \dots, n)$$

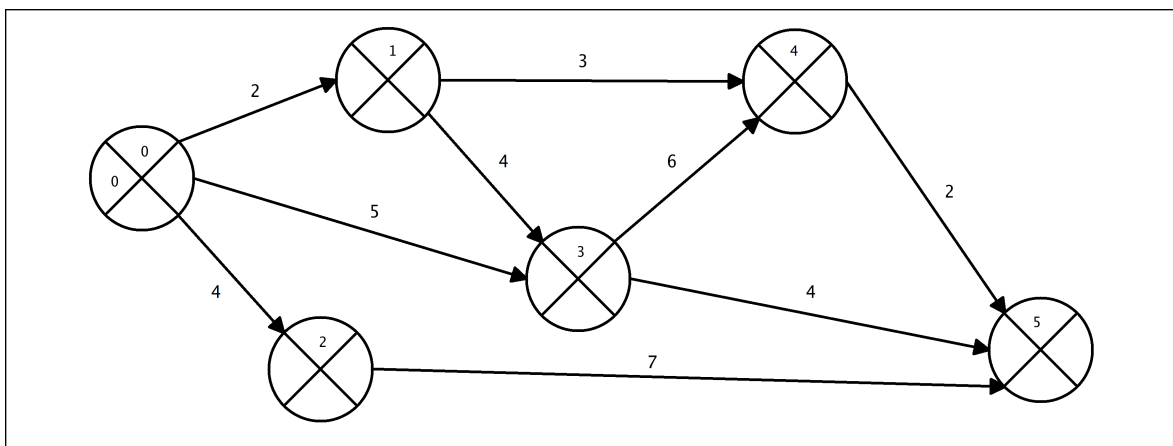
a výsledky napíšeme do levých částí. Zároveň do dolní části zapisujeme čísla  $i_1, \dots, i_v$ , pro které bylo dosaženo maxima.

Po dokončení výpočtů položíme  $T_n^{(1)} = T_n^{(0)}$  a určíme všechna  $T_j^{(1)}$  vzhledem k činnostem vystupujícím z uzlu  $P_j$  pomocí vzorce:

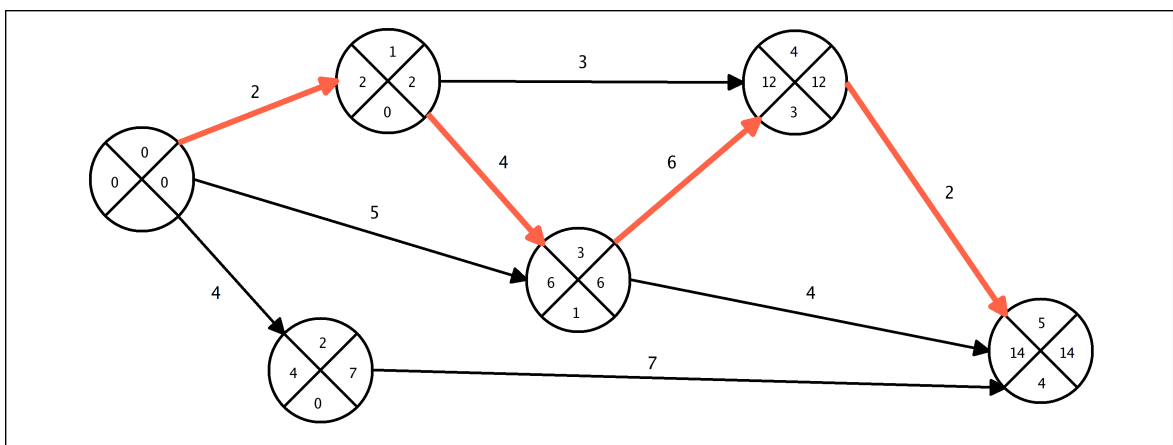
$$T_j^{(1)} = \min_k \{T_k^{(1)} - t_{jk}\}, \quad T_n^{(1)} = T_n^{(0)} \quad (j = n - 1, n - 2, \dots, 0).$$

Obdobný výpočet můžeme provést i v nepřečíslovaném grafu. V tom případě vypočítáme současně řády uzlů  $\lambda_j^* = T_j^{(0)}$  a  $\mu_j^*$ . Síť projdeme několikrát a při získání nových hodnot staré hodnoty smažeme. Po výpočtu  $T_j^{(0)}$  a  $\mu_j^*$  smažeme  $\mu_j^*$  v pravých částech a zapíšeme čísla  $T_j^{(1)} = T_{kr} - \mu_j^*$ .

Při postupu podle metody přeškrtavání hran současně v jednom kroku očíslovujeme události, vypočteme  $T_j^{(0)}$  a vyplníme dolní části. Poté vycházíme ze získaných čísel uzlů a sestupně vypočteme  $T_j^{(1)}$  pro termín  $T_{kr}$  ukončení projektu.



Obr. 8. Ukázka grafu připraveného pro výpočet (čísla uzlů v horní části).



Obr. 9. Graf s vypočtenými hodnotami a kritickou cestou (zvýrazněná).



### 3.6.3 Výpočet v tabulce

Projekt můžeme mít zadán také seznamem činností – většinou v tabulce kde na řádku u uzlu je určen koncový uzel a délka trvání, nebo je pro každý uzel dán seznam hodnot  $\lambda^{(0)}, \lambda^{(1)} \dots$  v takovém případě použijeme pro výpočet tabulku 1, kam přepíšeme hodnoty.

Uzel	Nulový krok		První krok		Druhý krok		...	
	$\lambda^{(0)}$	$\mu^{(0)}$	$\lambda^{(1)}$	$\mu^{(1)}$	$\lambda^{(2)}$	$\mu^{(2)}$	...	...
$P_0$								
$P_1$								
$\vdots$								
$P_i$								
$\vdots$								
$P_j$								
$\vdots$								
$P_n$								

Tab. 1. Ukázka tabulky pro výpočty hodnot

Postupujeme následujícím způsobem:

Nejdříve sloupce  $\lambda^{(0)}$  a  $\mu^{(0)}$  vyplníme nulami. Potom projdeme pro vyplnění sloupců  $\lambda^{(1)}$  a  $\mu^{(1)}$  seznam činností (například podle jejich zápisu). Pro činnost  $(P_i, P_j)$  vezmeme v řádku  $P_i$  nepřeskrtnuté číslo  $\lambda_i$  (sloupec  $\lambda^{(0)}$  nebo  $\lambda^{(1)}$  a přičteme k němu číslo  $t_{ij} \Rightarrow$  vypočteme číslo  $\lambda'_j = \lambda_i + t_{ij}$ . Potom zkontrolujeme řádek  $P_j$  a najdeme tam nepřeskrtnuté číslo  $\lambda_j$ . Pokud je  $\lambda'_j \geq \lambda_j$ , pak píšeme v políčku  $(P_j, \lambda^{(1)})$  číslo  $\lambda'_j$  a  $\lambda_j$  škrtneme. V opačném případě nic neměníme.

Po opravení  $\lambda_j$  provedeme opravení  $\mu_i$ . Přičteme číslo  $t_{ij}$  k nepřeskrtnutému číslu, které je napsáno v řádku  $P_j$  a sloupec  $\mu^{(1)}$  (pokud tam žádné není, tak v sloupci  $\mu^{(0)}$ ), tedy určíme  $\mu'_i = \mu_j + t_{ij}$ . Vrátime se k řádku  $P_i$  a jestliže se ukáže, že vypočtené  $\mu'_i$  není menší, než tam zapsané a nepřeskrtnuté číslo  $\mu_i$ , zapíšeme vypočítané  $\mu'_i$  do políčka  $(P_i, \mu^{(1)})$  a číslo  $\mu_i$  přeškrtneme. Pokud je  $\mu_i$  menší, než nepřeskrtnuté  $\mu_i$ , ke změně v řádku  $P_i$  nedochází.

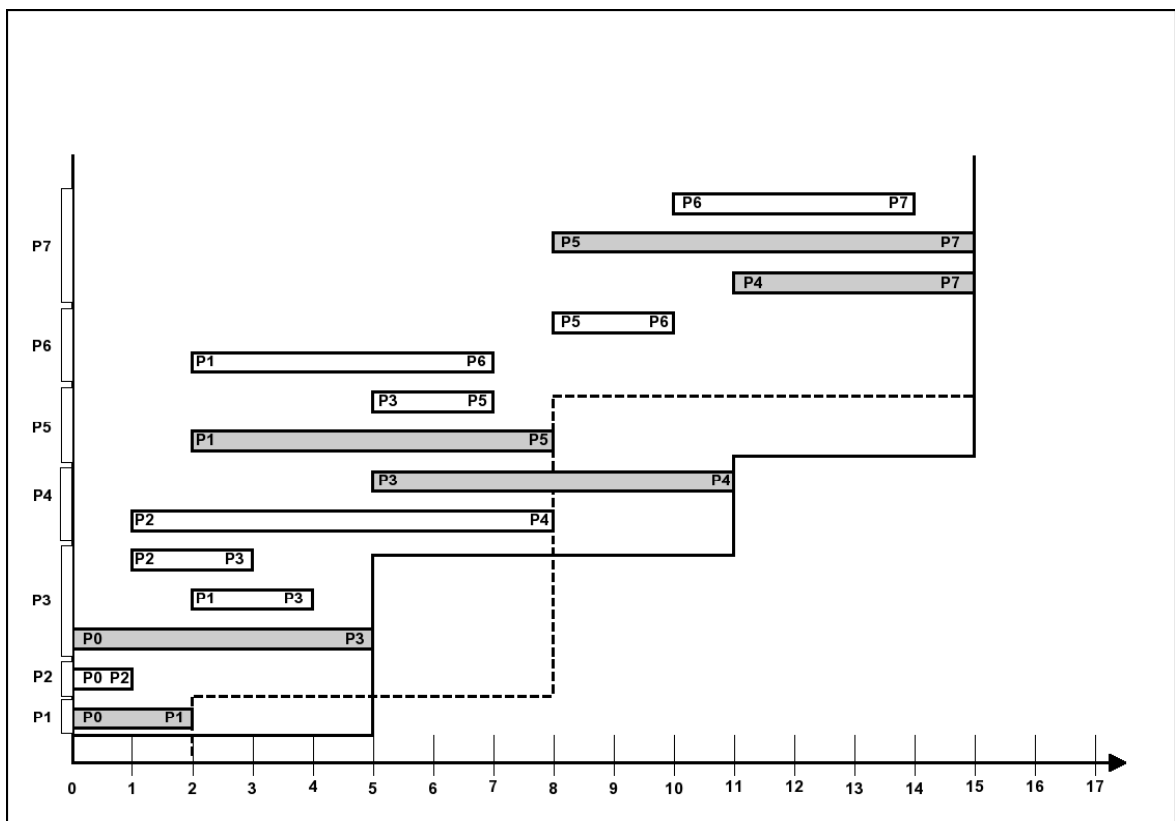
Po prvním kroku zůstanou nevyplněna políčka  $(P_0, \lambda^{(1)})$  a  $(P_s, \mu^{(1)})$ . Prázdné políčko  $(P_s, \mu^{(1)})$  znamená, že z něj nevystupují žádné činnosti. Zapíšeme do těchto políček nuly a přeškrtneme nuly v  $(P_0, \lambda^{(0)})$  a  $(P_s, \mu^{(0)})$ .

V druhém kroku procházíme obdobně celý soubor činností a vyplníme  $\lambda^{(2)}$  a  $\mu^{(2)}$ .

Výpočet je dokončen, pokud se v právě provedeném kroku nezmění žádné  $\lambda_j$  a  $\mu_j$ . Získané  $\{\lambda_j, \mu_j\}$  budou hledanými  $\{\lambda_j^*, \mu_j^*\}$  [1].

### 3.7 LINEÁRNÍ DIAGRAM PROJEKTU

Další ze způsobů jak znázornit projekt a jeho činnosti. Na vodorovné ose vyneseme časové jednotky. Jednotlivé činnosti jsou znázorněny nad sebou jako proužky začínající a končící ve stanoveném čase. Uzly  $P_i$  a  $P_j$  zapisujeme na začátek a konec činnosti. Vynášíme je ve skupinách podle růstu indexu  $j$  – koncového uzlu činnosti ( $P_i, P_j$ ). Ukázka takového diagramu je na obrázku 10. Zvýrazněnými pásky jsou znázorněny kritické činnosti, a kritické cesty (v této ukázce 2) jsou znázorněny čarami spojujícími pravé strany pásků.



Obr. 10. Ukázka lineárního projektu diagramu.

*Kritická doba projektu* je zde na první pohled viditelná a odpovídá konci pásku, který leží nejvíce vpravo.

*Kritická cesta* se určuje od činností, které končí v kritické době a směrem doleva na ně navazujícími činnostmi (končící na vertikále společné s činností od které postupujeme).

*Volná rezerva*  $T_j^{(0)} - T_i^{(0)} - t_{ij}$  činnosti ( $P_i, P_j$ ) je určena jako maximální délka úseku, o který můžeme posunout pásek vpravo, aniž bychom změnili začátky  $T_j^{(0)}$  činností, které vystupují z uzlu  $P_j$ .

## 4 ROZVRŽENÍ ZDROJŮ

Dosud jsme uvažovali o parametrech síťového grafu jen jako o časových jednotkách bez dalších omezení. V praxi ale musíme brát v úvahu omezení *zdrojů* (pracovní síla, zařízení, použitá technologie) pro dané činnosti.

Z toho pro nás vyplývá úloha optimalizovat rozvržení zdrojů tak, aby doba provedení byla minimální. Stejně tak musíme rozvrhnout *nároky* na jednotlivé zdroje.

Úplné řešení těchto úloh se zatím nepodařilo najít. Používáme proto heuristické algoritmy, které umožňují najít přibližná řešení.

### 4.1 OPTIMALIZACE ROZDĚLENÍ ZDROJŮ VZHLEDEM K ČASU

#### 4.1.1 Formulace úlohy při konstantních intenzitách

Při této formulaci uvažujeme projekt prováděný různými zdroji s daným dosažitelným limitem za časovou jednotku  $A_1(t), \dots, A_s(t)$  (například denní limit). Každá činnost se provádí pouze jedním z těchto zdrojů a je známá *konstantní intenzita nároku*  $r_{ij}^{(k)}$  zdroje při činnosti  $(P_i, P_j)$  (tj. množství  $k$ -tého zdroje potřebného pro danou činnost v časové jednotce). Trvání činnosti  $t_{ij}$  je rovněž známo [1].

Cílem úlohy je zajistit rozdělení činností tak, aby při daných omezeních bylo zajištěno dokončení projektu v co nejkratším čase.

*Objem činnosti*  $(P_i, P_j)$  je veličina  $w_{ij}^{(k)} = r_{ij}^{(k)}$ . Celkový nárok  $k$ -tého zdroje je roven:

$$\sum_{(P_i, P_j)} w_{ij}^{(k)} = \sum_{(P_i, P_j)} r_{ij}^{(k)}.$$

Bereme v úvahu jen činnosti, které potřebují  $k$ -tý zdroj. Pokud budeme předpokládat, že  $A_k(t)$  ( $k = 1, \dots, s$ ) je konstantní ( $A_k(t) = A_k$ ), potom dolní hranici dokončení termínu  $T$  definujeme jako

$$\max_k \left\{ \frac{1}{A_k} \sum_{(P_i, P_j)} w_{ij}^{(k)} \right\}$$

$T$  nemůže být menší, než  $T_{kr}$  (pro nelimitované zdroje), proto dostáváme

$$T \geq \max \left\{ T_{kr}, \max_k \left\{ \frac{1}{A_k} \sum_{(P_i, P_j)} w_{ij}^{(k)} \right\} \right\}$$

Následující algoritmus slouží k poměrně snadnému nalezení přibližného řešení uvažované úlohy (pro zjednodušení stanovíme, že všechny činnosti potřebují jeden a tentýž zdroj  $\Rightarrow s = 1$ ).

*První krok*

1) Sestavíme lineární diagram (viz kapitolu 3.7), určíme termín  $T_n = T_{kr}$  dokončení projektu, kritické činnosti a celkové rezervy. Promítneme všechny činnosti na časovou osu, činnost ležící nejvíce vlevo označíme  $\tau_0 = T_0 = 0$ , za ní následující pak  $\tau_1$ .

2) Vyšetříme všechny činnosti prováděné podle plánu v intervalu  $[\tau_0, \tau_1]$ . Očíslujeme

je vzestupně podle růstu jejich celkových rezerv (v případě rovnosti rezerv podle klesající intenzity)  $\Rightarrow$  kritické činnosti (s nulovou rezervou) budou mít nejnižší čísla.

3) Provedeme součet použitých zdrojů (jejich intenzit). Pokud nepřekročí limit zdroje, ponecháváme činnosti beze změny. V případě překročení stanoveného limitu posuneme právě probíranou činnost doprava do bodu  $\tau_1$  a pokračujeme další neprobranou činností.

#### *Obecný krok*

$k$ -tý krok předpokládá, že po  $k$  krocích činnosti nad intervalem  $[\tau_0, \tau_k]$  nepřekračují limit zdroje a můžeme přikročit ke kroku  $k + 1$ .

1) Nyní budeme uvažovat nad intervalem  $[\tau_k, T_n]$ . Umístíme každou činnost  $(P_i, P_j)$  nezačínající před bodem  $\tau_k$  tak, aby se začátek shodoval s nejdříve možným termínem  $T_i$  uzlu  $P_i$ . Pro zbývající část projektu nyní opakujeme všechny kroky z Prvního kroku odstavce 1). Na časové ose nyní budeme pracovat v intervalu  $[\tau_k, \tau_{k+1}]$ .

2) Očíslování činností nad intervalem  $[\tau_k, \tau_{k+1}]$  provedeme podle možností přípustných pro danou činnost:

- a) Pokud činnosti nepřipouštějí přerušení během jejich provádění, číslujeme nejdříve činností začínající před začátkem intervalu  $\tau_k$  a pokračují po něm. U těchto činností přepočítáme rozdíl mezi novou celkovou rezervou a úsekem od začátku činnosti po  $\tau_{k+1}$  a očíslováme v pořadí růstu těchto rozdílů. U stejných rozdílů číslujeme podle klesající intenzity. Ostatní činnosti očíslováme podle odstavce prvního kroku odstavce 2).
- b) Pokud činnosti mohou být přerušeny v průběhu provádění, postupujeme podle prvního kroku odstavce 2) s tím, že u činnosti začínající před  $\tau_k$  a nad sledovaným intervalem pokračuje, úsek vpravo od  $\tau_k$  považujeme za samostatnou činnost.

3) Postupujeme shodně s prvním krokem odstavce 3), přitom začátky činností, které začaly před  $\tau_k$  a nelze je přerušit posuneme do bodu  $\tau_{k+1}$ .

Po vyšetření všech činností je algoritmus dokončen a výsledný termín se obvykle velmi blíží k hledanému minimu.

### **4.1.2 Vyrovnání nároků na zdroje**

Vyrovnání nároků na zdroje může zkrátit termín  $T_n$  dokončení projektu, který byl vy počítán pomocí algoritmu v kapitole 4.1.2. U konečného lineárního diagramu získaného tímto algoritmem procházíme činnosti zprava a snažíme se vyrovnat nároky na zdroj podle následujícího pravidla:

- a) Pokud činnosti *nemohou* být přerušeny, vybereme interval  $[\tau_{m-1}, \tau_m = T_n]$ , kde  $\tau_{m-1}$  je projekce nejbližší k  $T_n$ . Pokud na tomto intervalu množství zdroje nepřevyšuje limit, vyšetříme interval  $[\tau_{m-2}, \tau_{m-1}]$  a činnostem, které mohou být posunuty vpravo, očíslováme podle snižující se intenzity. Potom probíráme činnosti ve vzestupném pořadí

a v případě, že součet vypočítaného nároku na zdroj intervalu  $[\tau_{m-1}, \tau_m]$  a nároku na probíranou činnost nepřesáhne limit, posuneme konec činnosti do bodu  $\tau_m$ . Tím pádem se mohou posunout činnosti na intervalu  $[\tau_{m-2}, \tau_{m-1}]$ , proto promítáme znovu začátky a konce všech činností na interval  $[T_0, \tau_{m-1}]$  a vypočteme využívání zdroje v intervalu  $[\tau'_{m-2}, \tau_{m-1}]$ .  $\tau'_{m-2}$  je nejbližší projekce k  $\tau_{m-1}$  zleva (nemusí být shodné s  $\tau_{m-2}$ . Pokud k posunu nedošlo, nemusíme provádět nové výpočty a  $\tau'_{m-2}, \tau_{m-2}$ .

Takto pokračujeme až do probrání všech činností. Je možné, že ke zkrácení dojde posunutím začátku první činnosti doprava od bodu  $T_0 = 0$ . Pak podle charakteru limitu na zdroj dojde buď ke zkrácení termínu, nebo jen k možnosti začít později, při zachovaném termínu dokončení.

- b) V případě, že činnosti *mohou* být přerušeny, je vyrovnávání nároků na zdroj efektivnější.

Postupujeme opět zprava od intervalu  $[\tau_{m-1}, \tau_m = T_n]$  očíslováním činností vzestupně s klesajícím nárokem. Pokud není překročen limit, postupně posunujeme jednotlivé činnosti, nebo jejich části (to je hlavní důvod proč může dojít k lepší optimalizaci). Po posunu opět promítneme nové začátky a konce činností a pokračujeme na dalším intervalu až do vyšetření celého diagramu.

#### 4.1.3 Formulace úlohy při proměnných intenzitách

Pro názornost budeme předpokládat, že činnosti jsou prováděny jedním zdrojem. Pro každou činnost je znám její objem  $w_{ij}$  v jednotkách zdroje. Intenzita  $t_{ij}$  provádění činnosti je definována funkcí  $A(t)$  (limit zdroje v čase  $t$ ) a omezením shora číslem  $\beta_{ij}$ ; ( $r_{ij} \leq \beta_{ij}$ ). Pokud budeme uvažovat, že délka trvání činnosti je nepřímo úměrná intenzitě, dostaneme

$$t_{ij} = \frac{w_{ij1}}{r_{ij1}} + \frac{w_{ij2}}{r_{ij2}} + \dots$$

Nejkratší trvání tedy bude mít činnost prováděná s maximální intenzitou  $\beta_{ij}$ , takže

$$t_{ij} \geq \frac{w_{ij}}{\beta_{ij}}.$$

U úloh s proměnnou intenzitou dochází k mírné změně v definici rezervy. Pokud máme dáno trvání  $t_{ij}$  všech činností  $(P_i, P_j)$ , pak pro každý uzel  $P_j$  definujeme trvání  $\mu_j^*$  maximální cesty z  $P_j$  do  $P_n$  a pro každou činnost  $(P_i, P_j)$  číslo  $\gamma_{ij} = \mu_j^* + t_{ij}$  (trvání maximální cesty ze začátku činnosti do  $P_n$ ). Tedy:

$$\gamma_{ij} = T_n - t_j^{(1)} + t_{ij} = T_n - (T_j^{(1)} - t_{ij}),$$

takže číslo  $\gamma_{ij}$  vyjadřuje rozdíl mezi trváním projektu a nejpozději přípustným začátkem činnosti  $(P_i, P_j)$ .

Soubor činností, které síťový graf umožňuje provádět v okamžiku  $t$  současně nazýváme *frontou činností*  $F(t)$ . *Maximální frontou*  $\phi(t)$  je soubor všech činností, které umožňuje síťový graf provádět v daném okamžiku  $t$  [1].

Časovou rezervu činnosti  $(P_i, P_j)$  fronty  $F(t)$  definujeme:  $t_{ij}^{(1)}$  je trvání části činnosti do okamžiku  $t$ ,  $\tau_{ij}(t) = \gamma_{ij} - t_{ij}^{(1)}$  je trvání maximální cesty ze začátku zbývajících částí činnosti do  $P_n$  a

$$L(t) = \max_{(P_i, P_j) \in F(t)} \tau_{ij}(t)$$

pro maximální z cest, které spojují začátky zbývajících částí činností fronty  $F(t)$  s  $P_n$ . Časovou rezervou  $\Delta\tau_{ij}(F(t), t)$  činnosti  $(P_i, P_j)$  fronty  $F(t)$  v okamžiku  $t$  rozumíme rozdíl  $\Delta\tau_{ij}(F(t), t) = L(t) - \tau_{ij}(t)$ .

#### 4.1.4 Definice minimálního zdržení dokončení projektu

U této úlohy budeme uvažovat o *zachovaných zdrojích*, tedy takových, jejichž zbytek v daném okamžiku nepropadá, ale přidá se k limitu zdroje v okamžiku následujícím.

Máme  $s$  zdrojů a pro každý z nich určenu neklesající funkci *dodávek*  $\psi_h(t)$ ; ( $h = 1, \dots, s$ ). Ta určuje množství  $h$ -tého zdroje dodaného k termínu  $t$ . Předpokládáme dále, že činnost  $(P_i, P_j)$  je prováděna jedním  $h$ -tým zdrojem s konstantní intenzitou  $r_{ij}^{(h)}$ , tedy trvání činnosti se nemění. Při výběru libovolných termínů síťového grafu (např. nejdříve možné začátky  $T_i(0)$  činností), můžeme sestrojít diagramy denního nároku  $R_h(t)$ ; ( $h = 1, \dots, s$ ) na každý zdroj a ke dni  $t$ , nazývané *integrální diagramy nároků na zdroje* pro každou činnost – grafy funkcí:

$$\int_0^t R_h(\tau) d\tau. \quad (9)$$

$R_h(t)$  jsou nezáporné a stupňovité, proto jsou funkce (9) neklesající a po částech lineární (dle stupňů funkce  $R_h(t)$ ) [1].

Celkové množství  $h$ -tého zdroje nutného pro dokončení projektu (vzhledem ke všem činnostem nárokujejícím tento zdroj) je

$$\sum_{P_i, P_j} r_{ij}^{(h)} t_{ij}$$

Pro dokončení projektu je tak důležité v každém čase  $t$  splnění nerovnosti při daných dodávkách

$$\psi_h(t) \geq \sum_{P_i, P_j} r_{ij}^{(h)} t_{ij} \quad (h = 1, \dots, s)$$

pokud pro jediný časový okamžik  $t$  a jediný zdroj nebude tato podmínka splněna, projekt nemůže být realizován těmito dodávkami.

*Zdrojově přípustný* termín  $T \geq T_{kr}$  projektu existuje v případě, že pro rozvržení činností v čase  $t \in [0, T]$  platí:

$$\psi_h(t) \geq \int_0^t R_h(\tau) d\tau$$

to znamená, že v žádném okamžiku dodávky převyšují požadavky. Cílem je nalezení minimálního zdrojově přípustného termínu  $T$ , tedy co nejmenšího zpoždění proti  $T_{kr}$ .

## 4.2 OPTIMÁLNÍ VYROVNÁNÍ NÁROKŮ NA ZDROJ PŘI ZADANÉM TERMÍNU

### 4.2.1 Formulace úlohy

Pokud máme zadán termín ukončení projektu, je třeba rozdělit zdroje tak, aby nárok na zdroj byl v jistém smyslu optimální – v podstatě jde o opačnou úlohu ke kapitole 4.1, kdy jsme k definovaným zdrojům (s konstantní nebo proměnnou intenzitou) hledali minimální termín dokončení projektu. Opět budeme předpokládat, že je zadáno trvání  $t_{ij}$  každé činnosti  $(P_i, P_j)$  a její konstantní intenzita  $r_{ij}$  nároku na zdroj [1].

Nejdříve si definujeme pojem optimálních nároků na zdroj. Kvůli zjednodušení budeme předpokládat nárokování jednoho druhu zdroje pro všechny činnosti. Průměrný nárok na zdroj spočítáme jako součet všech činností s jedním trváním a dělíme tuto sumu počtem dní, během nichž má být projekt dokončen. Dostaneme tak průměrný denní nárok na zdroj:

$$R_s = \frac{1}{T} \sum_{(P_i, P_j)} r_{ij} t_{ij}$$

Jako míru nerovnoměrnosti nároku na zdroj můžeme při daném plánu přijmout střední kvadratickou (směrodatnou) odchylku nároku  $R(t)$  potřebného v čase  $t$  od jeho průměrného nároku  $r_s$ , tj. hodnotu

$$\frac{1}{T} \int_0^T [R(t) - R_s]^2 dt = \frac{1}{T} \int_0^T R^2(t) dt - \frac{2R_s}{T} \frac{1}{T} \int_0^T R(t) dt + R_s^2 = \frac{1}{T} \int_0^T R^2(t) dt - R_s^2. \quad (10)$$

Potom plán splněný v daném termínu  $T$  a minimalizující (10) budeme nazývat *optimálním* [1].

Minimalizace (10) zřejmě vede k minimalizaci

$$\frac{1}{T} \int_0^T R^2(t) dt.$$

Pokud budeme mírou nerovnoměrnosti nároků na zdroj rozumět největší absolutní hodnotu odchylky potřebného zdroje  $R(t)$  od jeho průměrného denního nároku  $R_s$  – hodnotu

$$\max_{t \in [0, T]} |R(t) - R_s|, \quad (11)$$

optimálním plánem bude dosažení minimální hodnoty (11).

A za třetí, pokud za jako míru nároku na zdroj uvažujeme největší denní nárok, to znamená hodnotu

$$\max_{t \in [0, T]} R(t), \quad (12)$$

pak jako optimální je třeba brát plán, který minimalizuje (12).

Přístupy k řešení úloh optimalizace potřeby zdrojů jsou různé, všechny se však pouze přibližují k optimálnímu řešení.

### 4.2.2 Minimalizace střední kvadratické odchylky

Algoritmus pro cyklické procesy – v důsledku cykličnosti dává zde lepší výsledky, než v procesech, které jsme uvažovali. Za optimální považujeme plán, který minimalizuje (10).

Předpoklad – protože během dne každá činnost potřebuje konstantní množství zdroje a

množství činností se nemění, je funkce  $R(t)$  konstantní během dne a stupňovitá. Minimalizovaná funkce má tvar

$$R_1^2 + R_2^2 + \dots + R_T^2, \quad (13)$$

kde  $R_i$  je sumarizovaný nárok na zdroj v  $i$ -tý den [1].

*Přípravný krok* sestavíme lineární diagram projektu a všechny činnosti  $(P_i, P_j)$  umístíme do nejdříve možných začátků  $T_i^{(0)}$ .

*Obecný krok* Vyšetřujeme činnosti zprava od nejvyšší. Za přípustné posunutí vpravo považujeme jen to, které je dovoleno časovou rezervou a nevede ke zvětšení funkce (13) Pokud posunujeme činnost, předpokládáme, že činnosti již vyšetřené v daném kroku jsou pevně usazeny. Proto časová rezerva uvažované činnosti je určena počátky již umístěných činností<sup>1)</sup>.

Začneme od první činnosti, která leží nahoře v diagramu a má časovou rezervu (např. činnost  $(P_k, P_h)$ , začínající v  $i$ -tý den a končící v  $j$ -tý den). Pokud ji posuneme o jednotku vpravo,  $R_i$  se zmenší o  $r_{kh}$  a  $R_{j+1}$  se o  $r_{kh}$  zvětší. Funkce (13) se změní o hodnotu

$$[(R_{j+1} + r_{kh})^2 - R_{j+1}^2] - [R_i^2 - (R_i - r_{kh})^2] \quad (14)$$

Pokud je rozdíl (14) záporný, znamená to, že se funkce (13) zmenší a posunutí činnosti provedeme. Jelikož výraz (14) je roven  $2r_{kh}[R_{j+1} - (R_i - r_{kh})]$ , stačí pro zjištění přípustnosti posunutí činnosti  $(P_k, P_h)$  o jednotku vpravo porovnat celkový zdroj, který je potřebný v  $(j + 1)$ -ní den, s celkovým zdrojem potřebným v  $i$ -tý den, přičemž nepočítáme zdroj pro uvažovanou činnost  $(P_k, P_h)$ . Jestliže první součet není větší než druhý, pak je posun činnosti  $(P_k, P_h)$  o jednotku vpravo přípustný a provede se [1].

Dále vyšetřujeme stále stejnou činnost  $(P_k, P_h)$  na možnost posunutí o další jednotku vpravo. Pokud je to přípustné, provedeme posun a pokračujeme v postupu, dokud má činnost časovou rezervu.

Pokud posun o jednotku není přípustný, zkoumáme možnost posunutí činnosti o dvě jednotky (opět pokud to dovolí časová rezerva). Určíme proto rozdíl  $R_{j+2} - (R_{i+1} - r_{kh})$ ; ( $i \neq j$ ) a je-li záporný, určíme součet  $[R_{j+1} - (R_i - r_{kh})] + [R_{j+2} - (R_{i+1} - r_{kh})]$ .

Je-li nekladný, provedeme posun činnosti  $(P_k, P_h)$  o dvě jednotky, v opačném případě pokračujeme o tři jednotky. . . dokud není vyčerpána časová rezerva.

Po umístění činnosti  $(P_k, P_h)$  přejdeme k další činnosti umožňující posun vpravo. Po probrání všech činností provádíme obdobně vyšetřování shora dolů. Algoritmus je dokončen v okamžiku, kdy právě provedený krok nezměnil ani jedno umístění činnosti.

### 4.2.3 Minimalizace maximálního nároku na zdroj

Další algoritmus nám dává přibližné řešení úlohy optimalizace nároků na zdroje v případě, kdy za optimální považujeme plán minimalizující (12).

<sup>1)</sup>Pokud má být projekt proveden v termínu  $T_n = T_{kr}$ , kritické činnosti se nemohou posunovat.



*Přípravný krok*

Sestavíme lineární diagram projektu, ve kterém všechny činnosti umístíme do nejdříve možných počátků. Dále určíme pro každou činnost celkovou rezervu při zadaném termínu  $T$  dokončení. Vypočteme denní nárok na zdroj pro celý projekt a určíme maximální nárok na zdroj.

*Obecný krok*

Položíme úroveň zdroje o jednotku níže, než je maximální nárok získaný v předchozím kroku a zkusíme při těchto ohraničených zdrojích rozmístit projekt v zadaném termínu  $T$ .

Promítneme začátky a konce všech činností na interval  $[0, T]$ . Probíráme všechny získané intervaly zleva až do intervalu  $[\tau_k, \tau_{k+1}]$ , v němž je nárok na zdroj větší než předepsaná úroveň. Vybereme činnosti, které nad daným intervalem umožňují posun vpravo, aniž by došlo k prodloužení termínu  $T$ .

a) - Pokud takové činnosti neexistují, obnovíme rozložení z minulého kroku, které je tím pádem konečné.

b) - Pokud takové činnosti existují, vybereme z nich tu s nejmenší intenzitou a její začátek odsuneme k okamžiku  $\tau_{k+1}$ .

Provedeme posun, nutný u některých činností vpravo, způsobený posunem dané činnosti. Tento posun je minimální – do nových nejdříve možných počátků. Začátky a konce činností ležících nad zbývajícím intervalem  $[\tau_k, T]$  znovu promítneme na tento interval a hledáme další činnosti připouštějící posun vpravo. Pokračujeme dokud není algoritmus ukončen na základě možnosti a), nebo neumístíme celý projekt do zadaného limitu zdroje. V tom případě opakujeme obecný krok, dokud nenastane varianta a).

## 5 MINIMALIZACE NÁKLADŮ

V souvislosti s minimalizací nákladů budeme řešit především dva typy úloh. Jak minimalizovat náklady celého projektu při zadaném termínu dokončení a jak minimalizovat termín dokončení projektu při daných nákladech.

### 5.1 MINIMALIZACE NÁKLADŮ NA PROJEKT PŘI JEHO KONSTANTNÍM TRVÁNÍ

#### 5.1.1 Optimální plán bez rezerv

Nechť jsou pro daný projekt s předepsaným trváním  $t_{ij} = d_{ij}$  činností  $(P_i, P_j)$  vypočteny nejdříve možné a nejpozději přípustné termíny  $T_j^{(0)}$  a  $T_j^{(1)}$  uzlu  $P_j$ . Dále jsou určeny kritické a nekritické činnosti. Nechť máme závislost hodnoty  $c_{ij}$  – přímých nákladů na trvání  $t_{ij}$  každé činnosti  $(P_i, P_j)$  dānu vztahem

$$c_{ij} = -a_{ij}t_{ij} + b_{ij} \quad (a_{ij} \geq 0, b_{ij} \geq 0) \text{ – lineární varianta,}$$

nebo

$$c_{ij} = \frac{a_{ij}}{t_{ji}} \quad (a_{ij} > 0) \text{ – konvexní varianta}$$

Jak bylo vyloženo v kapitole (3), mají časové rezervy pouze nekritické činnosti. Předpokládáme zmenšení nákladů tím, že prodloužíme trvání činnosti na jejich úkor.

Vzniká úloha: *Pro nalezený kritický termín využít rezervy nekritických činností tak, aby byl získán optimální plán, tj. plán minimalizující náklady celého komplexu činností [1].*

Pro matematickou formulaci objasníme některé zvláštnosti:

Každé trvání činnosti má dosáhnout největší možné hodnoty  $\Rightarrow$  každá činnost by se měla stát kritickou. Považujeme-li termíny  $T_j$  uzlu  $P_j$  ( $j = 0, 1, \dots, n$ ) za neznámé, pak trvání každé činnosti  $(P_i, P_j)$  musí být rovna rozdílu  $T_j - T_i$ .

Vyšetřujeme úlohu při předem nalezeném kritickém termínu a kritických cestách, takže termíny činností na kritických cestách máme pevně dané.

Množinu uzlů ležících na kritických cestách označíme  $K$  (zřejmě  $P_0 \in K$  a  $P_n \in K$ ) a množinu činností  $(P_i, P_j)$ , u kterých alespoň jeden z konců nepatří do množiny  $K$  označíme  $R$ .

Nyní můž být uvedená úloha formulována takto:

najít minimum funkce

$$z = \sum_{(P_i, P_j) \in R} [-a_{ij}(T_j - T_i) + b_{ij}] \quad (a_{ij} \geq 0, b_{ij} \geq 0) \quad \text{(lineární varianta)} \quad (15)$$

nebo funkce

$$z = \sum_{(P_i, P_j) \in R} \frac{a_{ij}}{T_j - T_i} \quad (a_{ij} > 0) \quad \text{(konvexní varianta)} \quad (16)$$

při omezeních

$$\left. \begin{array}{l} T_j - T_i \geq d_{ij} \quad \text{pro všechny činnosti } (P_i, P_j) \in R \\ T_j = T_j^{(0)} = T_j^{(1)} \quad \text{pro všechny činnosti } P_j \in K \end{array} \right\} \quad (17)$$

a všech  $d_{ij}$  kladných.

V této úloze je ovšem optimální plán minimální vzhledem k nákladům pouze mezi plány zakončenými ve vypočteném kritickém termínu  $T_{kr}$ . Pokud bude termín dokončení  $T$  stanoven jako  $T > T_{kr}$ , je možno prodlužovat i délky kritických činností a náklady optimálního plánu budou menší než při  $T = T_{kr}$ .

Pomocí této úvahy zdůvodníme následující formulaci, která zobecňuje úlohy (15) – (17)

Nechť máme pro každou činnost  $(P_i, P_j)$  dáno minimální trvání  $d_{ij}$  a necht' je dán termín  $T$  dokončení celého projektu. Je třeba určit termíny  $T_j$  uzlů  $P_j (j = 1, 2, \dots, n)$  tak, aby mezi všemi plány provedenými v termínu  $T$  měl hledaný plán nejmenší náklady, jinými slovy je třeba najít minimum funkce:

$$z = \sum_{(P_i, P_j)} F - a_{ij}G(T_j - T_i) + b_{ij} \quad (a_{ij} \geq 0, b_{ij} > 0) \quad (\text{lineární varianta}), \quad (18)$$

nebo funkce

$$z = \sum_{(P_i, P_j)} \frac{a_{ij}}{T_j - T_i} \quad (a_{ij} > 0) \quad (\text{konvexní varianta}) \quad (19)$$

při omezeních

$$T_j - T_i \geq d_{ij} \quad \text{pro všechna } (P_i, P_j), \quad T_0 = 0, \quad T_n \leq T. \quad (20)$$

Protože funkce (18) i omezení (20) jsou lineární, je tato úloha obvyklou úlohou lineárního programování, kterou je možno řešit například simplexovou metodou.

Funkce (19) jako součet konvexních funkcí je konvexní a úloha je úloha konvexního programování. Tímto způsobem je možno řešit i obecnější úlohu, kdy je závislost nákladů z provedení všech činností na vektoru  $\mathbf{t} = \{t_{ij}\}$  jejich trvání dána ve tvaru

$$z = \sum_{(P_i, P_j)} f_{ij}(t_{ij})$$

kde funkce  $f_{ij}(t_{ij})$  jsou konvexní [1].

### 5.1.2 Algoritmus sestavení optimálního plánu bez rezerv

Algoritmus sestavila I. A. RADČIKOVÁ (spoluautorka [1]). Není tak těžkopádný jako algoritmus simplexové metody. Účelovou funkci (18) zapíšeme ve tvaru

$$z = \sum_{(P_i, P_j)} [-a_{ij}(T_j - T_i) + b_{ij}] = \sum_{(P_i, P_j)} b_{ij} - \sum_{i=0}^n a_i T_i$$

a místo úlohy minimalizovat (18) budeme řešit ekvivalentní úlohu maximalizace funkce

$$z = \sum_{i=0}^n a_i T_i \quad (21)$$

při omezeních

$$\left. \begin{aligned} T_i = T_j \leq -d_{ij} \quad \text{pro všechna } (P_i, P_j), \\ T_n - T_0 \leq T, \\ T_0 = 0. \end{aligned} \right\} \quad (22)$$

Duální k úloze (21) – (22) je úloha minimalizace funkce

$$W = T x_{n0} - \sum_{(P_i, P_j)} d_{ij} x_{ij} \quad (23)$$

při omezeních

$$\left. \begin{aligned} \sum_j x_{ij} - \sum_k x_{ki} &= a_i \quad (i = 1, \dots, n) \\ x_{ij} &\geq 0 \quad \text{pro všechna } (P_i, P_j) \\ x_{n0} &\geq 0. \end{aligned} \right\} \quad (24)$$

Pokud každé  $x_{ij}$  budeme chápat jako množství látky protékající hranou  $(P_i, P_j)$  za jednotku času, pak je součet  $\sum_j x_{ij}$  roven množství látky vytékající z uzlu  $P_i$  a součet  $\sum_k x_{ki}$  roven množství látky vtékající do uzlu  $P_i$ .

Uzel  $P_i$  je *zřídlem*, když v (24) platí  $a_i > 0$  a odtokem pro  $a_i < 0$ . Veličina  $a_i$  je intenzita odpovídajícího zřídla nebo odtoku. Pro  $a_i = 0$  můžeme vrchol  $P_i$  považovat buď za zřídlo, nebo odtok nulové intenzity.

Takto je možné rovnice (24) pokládat za rovnice toku v síti s množinou zřídla a odtoků, která odpovídá síťovému grafu, doplněnému hranou  $(P_n, P_0)$ .

Řešení  $\{x_{ij}\}$  soustavy (24) nazveme *přípustný tok* v síti [1].

Získali jsme přípustný tok. Sečteme-li všechny rovnice (24) s ohledem na to, že  $\sum_{i=0}^n a_i = 0$ , dostaneme

$$\sum_j x_{0j} - x_{n0} = - \sum_{i=1}^n a_i = a_0.$$

Tuto rovnici můžeme chápat jako rovnici toku uzlem  $P_0$ , jehož intenzita se rovná  $a_0$ .

Doplněním do soustavy (24) dostáváme soustavu

$$\sum_j x_{ij} - \sum_k x_{ki} = a_i \quad (i = 0, 1, \dots, n) \quad (25)$$

přičemž

$$\sum_{a_i > 0} a_i + \sum_{a_i < 0} a_i = 0. \quad (26)$$

Rovnice (26) vyjadřuje, že celková intenzita zřídla je rovna celkové intenzitě odtoků. Je nutnou podmínkou pro existenci přípustného toku v síti.

Duální úloha spočívá ve vyhledání toku v síti s množinou zřídla a odtoků, minimalizující lineární formu (23).

Algoritmus se opírá o druhou větu duality viz [1]. Podle ní jsou přípustná řešení  $\{T_j\}$  a  $\{x_{ij}\}$  obou duálních úloh tehdy a jen tehdy, když vyhovují podmínkám:

$$(T_j - T_i - D_{ij})x_{ij} = 0 \quad \text{pro všechna } (P_i, P_j). \quad (27)$$

Z této podmínky plyne

$$x_{ij} \begin{cases} = 0 & \text{pro } T_j - T_i - d_{ij} > 0, \\ \geq 0 & \text{pro } T_j - T_i - d_{ij} = 0. \end{cases}$$

Pomocí algoritmu se vybírají přípustné hodnoty  $T_i$  vyhovující podmínce (22), tedy tok je možno pouštět hranami  $(P_i, P_j)$ , pro které platí  $T_j - T_i - d_{ij} = 0$ . Tyto hrany budeme nazývat *přípustné* [1].

### 5.1.3 Optimální plán při existenci rezerv

V kapitole (5.1.1) jsme neuvažovali omezení činností  $t_{ij}$  shora. Často je však délka trvání omezena:  $d_{ij} \leq t_{ij} \leq D_{ij}$ , kde  $d_{ij}$  je minimální trvání a  $D_{ij}$  je normální trvání této činnosti.

Pokud je dán termín  $T$  dokončení projektu, pak vzniká minimalizovat funkci

$$z = \sum_{P_i, P_j} (-a_{ij}t_{ij} + b_{ij}) \quad (a_{ij} \geq 0, b_{ij} > 0) \quad (\text{lineární varianta}), \quad (28)$$

nebo funkce

$$z = \sum_{(P_i, P_j)} \frac{a_{ij}}{t_{ij}} \quad (a > 0) \quad (\text{konvexní varianta}) \quad (29)$$

při omezeních

$$\left. \begin{aligned} T_j - T_i - t_{ij} &\geq 0 \quad (\text{pro všechna}) \quad (P_i, P_j), \\ d_{ij} &\leq t_{ij} \leq D_{ij}, \\ T_0 &= 0, \quad T_n \leq T. \end{aligned} \right\} \quad (30)$$

Neznámé jsou v této úloze termíny uzlů a trvání činností. V optimálním plánu  $t_{ij}$  dosahují maximálně přípustných hodnot, takže  $t_{ij} = \min\{D_{ij}, T_j - T_i\}$ .

Na rozdíl od úloh (5.1.1) mohou mít některé činnosti časovou rezervu i v optimálním plánu (nevyužitelnou). Bude se jednat o činnosti  $(P_i, P_j)$  pro které  $t_{ij} = D_{ij} < T_j - T_i$ .

První varianta je opět úlohou lineárního a druhá konvexního programování.

Soustavu hodnot  $\{t_{ij}, T_j\}$  vyhovující podmínkám (30) budeme nazývat *přípustným plánem* a přípustný plán, minimalizující funkci  $z$  se nazývá *optimální plán* úlohy.

V důsledku zadaného termínu  $T$  je třeba zkoumat řešitelnost úlohy při tomto zadání. Položíme všechny činnosti  $t_{ij} = d_{ij}$  a vypočteme kritický termín  $m$  celého komplexu činností. Pokud  $T \geq m$ , pak má úloha řešení, jinak není soustava omezení (30) konzistentní a úloha řešení nemá.

## 5.2 PARAMETRICKÁ ÚLOHA MINIMALIZACE NÁKLADŮ NA PROJEKT

### 5.2.1 Matematický model úlohy

Jak bylo řečeno (5.1.3), trvání každé činnosti je dáno v mezích  $0 \leq d_{ij} \leq t_{ij} \leq D_{ij}$ .

Pokud  $t_{ij} = d_{ij}$ , bude odpovídající *kritický termín*  $m$  nejkratším termínem, ve kterém je možno dokončit projekt. Položíme-li  $t_{ij} = D_{ij}$ , pak kritický termín  $M$  bude nejpozdějším kritickým termínem k dokončení projektu. Analogicky nazveme  $m$  *minimální* a  $M$  *maximální omezení termínu*  $T_n - m \leq T_n \leq M$ .

Pro libovolnou hodnotu  $T_n \in [m, M]$  bude existovat vlatní optimální plán vzhledem k ceně. Parametrická úloha spočívá v určení optimálního plánu pro každý přípustný termín projektu, tj. ve vyhledání minima funkce pro každé  $\lambda$  z intervalu  $[m, M]$ .

$$z = \sum_{(P_i, P_j)} (-a_{ij}t_{ij} + b_{ij}) \quad (31)$$

při omezeních

$$T_j - T_i - t_{ij} \geq 0 \quad \text{pro všechna} \quad (P_i, P_j) \quad (32)$$

$$d_{ij} \leq t_{ij} \leq D_{ij} \quad \text{pro všechna} \quad (P_i, P_j) \quad (33)$$

$$T_0 = 0, \quad T_n = \lambda. \quad (34)$$

Dostali jsme tak duální úlohu (31) – (34) lineárního parametrického programování, kterou nazveme *parametrickou úlohou minimalizace ceny projektu*.

### 5.3 KELLEYOVA METODA

Použití metody lineárního parametrického programování pro řešení parametrické úlohy je příliš složité. Proto si vyložíme *Kelleyovu metodu* převodu parametrické úlohy na úlohu o maximálním toku, řešenou jednoduchým algoritmem.

#### 5.3.1 Struktura optimálního plánu

Z předchozích úloh vyplývá, že trvání  $t_{ij}$  libovolné činnosti  $(P_i, P_j)$  je definováno vzorcem:  $t_{ij} = \min\{D_{ij}, T_j - T_i\}$ , takže v optimálním plánu trvání  $t_{ij}$  každé činnosti  $(P_i, P_j)$  může nabývat pouze hodnot: 1)  $t_{ij} = D_{ij} = T_j - T_i$ , tedy činnost má normální trvání a nemá časovou rezervu.

2)  $t_{ij} = T_j - T_i < D_{ij}$ , to znamená, že trvání činnosti je menší než normální, ale nemá časovou rezervu.

3)  $t_{ij} = D_{ij} < T_j - T_i$ , činnost má normální trvání a má časovou rezervu, která nemůže být využita.

*Struktura optimálního plánu* vznikne rozkladem všech činností na množiny:

1. Dle časových rezerv:

Množina  $E_1$  všech činností  $(P_i, P_j)$ , které vyhovují podmínce 1), nebo 2) (bez rezerv).

Množina  $E_2$  všech činností  $(P_i, P_j)$ , které vyhovují podmínce 3) (s rezervami).

2. Dle vztahu mezi hodnotami délek trvání činností v optimálním plánu:

Množina  $Q_1$  všech činností  $(P_i, P_j)$ , pro které platí  $t_{ij} = D_{ij} > d_{ij}$ .

Množina  $Q_2$  všech činností  $(P_i, P_j)$ , pro které platí  $t_{ij} = D_{ij} = d_{ij}$ .

Množina  $Q_3$  všech činností  $(P_i, P_j)$ , pro které platí  $t_{ij} = d_{ij} < D_{ij}$ .

Množina  $Q_4$  všech činností  $(P_i, P_j)$ , pro které platí  $d_{ij} = t_{ij} < D_{ij}$ .

#### 5.3.2 Kelleyova věta

Nechť je pro některou hodnotu  $T_n = \lambda$  vyřešena úloha (31) – (34) a  $\{t_{ij}^{\sim}, T_j^{\sim}\}$  je získaný optimální plán, takže

$$z_{\min} = \sum_{(P_i, P_j)} (-a_{ij} t_{ij}^{\sim} + b_{ij}) = C.$$

Formulujeme úlohu – vyjdeme z daného optimálního plánu  $T_n^{\sim} = \lambda$  a sestrojíme nový optimální plán

$$\{t'_{ij}, T'_j\} = \{t_{ij}, T_j^{\sim}\} - \Theta\{\xi_{ij}, \eta_{ij}\} = \{t_{ij} - \Theta\xi_{ij}, T_j^{\sim} - \Theta\eta_j\} \quad (35)$$

trvání  $T_n = \lambda - \Theta$ , a určit horní mez  $\Theta_0$  růstu nezáporného parametru  $\Theta$  [1]. Hledáme tedy směr  $\{\xi_{ij}, \eta_j\}$  úsečky s počátkem v bodě  $\{t_{ij}, T_j^{\sim}\}$ , v jejímž každém bodě nabývá funkce  $z$  minima při omezeních (32) – (34), a v určení veličiny  $\Theta_0$  přípustného kroku podél této úsečky.  $\Theta_0$  je délka odpovídajícího intervalu lineárnosti po částech lineární funkce:

$$\min z = f(\lambda).$$

Řešení – pro funkci  $z$  podél úsečky (35) máme

$$z = \sum_{(P_i, P_j)} (-a_{ij}t'_{ij} + b_{ij}) = \sum_{(P_i, P_j)} (-a_{ij}t_{ij} + b_{ij}) + \Theta \sum_{(P_i, P_j)} a_{ij}\xi_{ij} = C + \Theta \sum_{(P_i, P_j)} a_{ij}\xi_{ij}$$

a pokud najdeme směr úsečky  $\{\xi_{ij}, \eta_{ij}\}$ , je úloha převedena na minimalizaci funkce

$$\sum_{(P_i, P_j)} a_{ij}\xi_{ij}$$

Najdeme nyní omezení pro proměnné  $\xi_{ij}, \eta_j$  tak, aby byl hledaný plán přípustný [1].

Je-li  $T_j^{\sim} - T_i^{\sim} - t_{ij} = 0$ , pak

$$\sigma_{ij} = \xi_{ij} + \eta_i - \eta_j \geq 0 \quad \text{pro všechna } (P_i, P_j) \in E_1.$$

Je-li  $T_j^{\sim} - T_i^{\sim} - t_{ij} > 0$ , může být  $\sigma_{ij}$  jak kladné, tak záporné číslo.

Z (33) máme

$$d_{ij} \leq t_{ij} - \Theta\xi_{ij} \leq D_{ij}. \quad (36)$$

Z levé nerovnosti (vzhledem k nezápornosti  $\Theta$  vyplývá: jestliže  $t_{ij} = d_{ij}$ , pak

$$\xi_{ij} \leq 0 \quad \text{pro všechna } (P_i, P_j) \in E_1 \cap Q_3. \quad (37)$$

Z pravé nerovnosti vyplývá: jestliže  $t_{ij} = D_{ij}$  [ $(P_i, P_j) \in Q_1 \cup Q_2$ ], pak  $\xi_{ij} = 0$ .

Vzhledem ke struktuře [1, obr. 58] optimálního plánu  $\{t_{ij}, T_j^{\sim}\}$  dostáváme

$$\left. \begin{array}{l} \xi_{ij} \geq 0 \quad \text{pro všechna } (P_i, P_j) \in E_1 \cap Q_1, \\ \xi_{ij} = 0 \quad \text{pro všechna } (P_i, P_j) \in (E_1 \cap Q_2) \cup E_2. \end{array} \right\} \quad (38)$$

Z podmínky (34) dostaneme omezení pro  $\eta_j$ :

$$\eta_0 = 0, \quad \eta_m = 1$$

Jestliže v řešení úlohy (42) – (45) existují  $\sigma_{ij} < 0$  (to jen v případě kdy  $(P_i, P_j) \in E_2$ ), dostáváme

$$\Theta \leq \frac{T_j^{\sim} - T_j^{\sim} - t_{ij}}{-\sigma_{ij}} \quad \text{pro } \sigma_{ij} < 0.$$

V takovém případě je veličina  $\Theta$  ohraničena shora číslem:

$$\Theta' = \min_{\sigma_{ij} < 0} \frac{T_j^{\sim} - T_j^{\sim} - t_{ij}}{-\sigma_{ij}} > 0. \quad (39)$$

Z (36) vidíme, e pro  $\xi_{ij} < 0$  je levá nerovnost vždy splnění, ke splnění pravé pak musí platit:

$$\Theta \leq \frac{t_{ij} - D_{ij}}{\xi_{ij}},$$

tedy při  $\xi_{ij} > 0$  dostáváme další omezení shora pro  $\Theta$ :

$$\Theta'' = \min_{\xi_{ij} < 0} \frac{t_{ij}^{\sim} - D_{ij}}{\xi_{ij}} > 0 \quad (40)$$

Je-li  $\sigma_{ij} > 0$ , je pravá nerovnost (36) vždy splněna, pro splnění levé nerovnosti je nutno, aby:

$$\Theta \leq \frac{t_{ij}^{\sim} - d_{ij}}{\xi_{ij}},$$

takže při  $\xi_{ij} > 0$  dostáváme pro  $\Theta$  poslední omezení shora číslem:

$$\Theta''' = \min_{\xi_{ij} > 0} \frac{t_{ij}^{\sim} - d_{ij}}{\xi_{ij}} > 0 \quad (41)$$

$\Theta_0$  je pak určeno vzorcem

$$\Theta_0 = \min\{\Theta', \Theta'', \Theta'''\}$$

*Kelleyova věta:*

Nechť je  $\{t_{ij}^{\sim}, T_{ij}^{\sim}\}$  optimální plán úlohy (31) – (34) trvání  $T_n^{\sim} = \lambda$  a necht' je určena jeho struktura, tj. určeny množiny  $E_i, Q_j$  ( $i = 1, 2; j = 2, 3, 4$ ). Pak je pro vyhledání optimálního plánu

$$\{t'_{ij} = t_{ij}^{\sim} - \Theta \xi_{ij}, T'_j = T_j^{\sim} - \Theta \eta_j\}$$

trvání  $T_n' = \lambda - \Theta < \lambda$  třeba řešit následující úlohu lineárního programování: najít minimum funkce:

$$\sum_{(P_i, P_j)} a_{ij} \xi_{ij} \quad (42)$$

při omezeních

$$\sigma_{ij} = \xi_{ij} + \eta_i - \eta_j \geq 0 \quad \text{pro všechna } (P_i, P_j) \in E_1, \quad (43)$$

$$\xi_{ij} \begin{cases} \geq 0 & \text{pro všechna } (P_i, P_j) \in E_1 \cap Q_1, \\ = 0 & \text{pro všechna } (P_i, P_j) \in E_2 \cup (E_1 \cap Q_2), \\ \leq 0 & \text{pro všechna } (P_i, P_j) \in E_1 \cap Q_3, \end{cases} \quad (44)$$

$$\eta_0 = 0, \quad \eta_n = 1. \quad (45)$$

Je-li úloha (42) – (45) řešitelná a  $\{\xi_{ij}^*, \eta_j^*\}$  je její řešení, pak existuje číslo určené tímto řešením

$$\Theta_0 = \min\{\Theta', \Theta'', \Theta'''\} > 0 \quad (46)$$

takové, že pro každé  $\Theta$  z intervalu  $[0, \Theta_0]$  je plán  $\{t_{ij}^{\sim} - \Theta \xi_{ij}^*, T_j^{\sim} - \Theta \eta_j^*\}$  optimální pro úlohu (31) – (34) trvání  $T_n = \lambda - \Theta < \lambda$ . Jestliže úloha (42) – (45) není řešitelná, pak neexistuje optimální plán úlohy (31) – (34) s menším trváním než  $\lambda$  [1].

### 5.3.3 Přechod k úloze o maximálním toku

Podle pravidla o duálně sdružených úlohách [1] zformulujeme podmínky primární úlohy, vzhledem ke které je úloha (42) – (45) duální.

Neznámé v primární úloze označíme  $x_{ij}$ , jejich počet je roven počtu všech činností  $(P_i, P_j) \in E_1$ .

Jako koeficienty účelové funkce slouží absolutní členy omezení (43), mezi nimiž jsou



různé od nuly a rovny jedné ty, pro které  $j = n$ . Tedy účelová funkce pro primární úlohu má tvar

$$\sum_{(P_i, P_j) \in E_1} x_{in}. \quad (47)$$

Určení znamének omezení neznámých v primární úloze z (43)

$$x_{ij} \equiv 0 \text{ pro } (P_i, P_j) \in E_2 \quad (48)$$

protože na  $\sigma_{ij}$  se nekladou omezení

$$x_{ij} \geq 0 \text{ pro } (P_i, P_j) \in E_1 \quad (49)$$

Pro určení ostatních omezení primární úlohy uvažujeme matici omezení (43) – obdélníková z jednotkové matice koeficientů při  $\xi_{ij}$ , ke které je zprava připsána obdélníková matice koeficientů při  $\eta_i$  a  $\eta_j$ . Každý řádek je složen z nul a ne víc než dvou jedniček (první kladná, druhá záporná). Matice transponovaná je také složena z jednotkové s připsanou obdélníkovou maticí (řádky se skládají z 0 a  $\pm 1$ ).

Vezmeme-li za koeficienty prvky jednotkové matice a za absolutní členy koeficienty účelové funkce (42) a uvažujeme-li znaménka omezení (44), dostaneme následující omezení primární úlohy:

$$x_{ij} \begin{cases} \leq a_{ij} & \text{pro } (P_i, P_j) \in E_1 \cap Q_1, \\ \geq a_{ij} & \text{pro } (P_i, P_j) \in E_1 \cap Q_3, \\ = a_{ij} & \text{pro } (P_i, P_j) \in E_1 \cap Q_4. \end{cases} \quad (50)$$

Vezmeme-li nyní za koeficienty prvky obdélníkové matice a za absolutní členy nulové koeficienty při  $\eta_j$  v účelové funkci (42) a uvážíme-li, že neexistují omezení neznámých  $\eta_j$  ( $j = 1, 2, \dots, n-1, \eta_j$  – volné proměnné), dostáváme ještě následující omezení – rovnici primární úlohy:

$$\sum_i X_{ij} - \sum_k x_{jk} = 0 \quad (j = 1, 2, \dots, n-1) \quad (51)$$

kde se první součet týká všech činností vstupujících do  $P_j$  a druhý všech činností vystupujících z  $P_j$  [1].

Primární úlohou k duální úloze (42)–(45) je úloha maximalizace formy (47) při omezeních (48)–(51), tj. úloha o maximálním toku – kapitola (6.1). Algoritmy pro její řešení jsou těžkopádné podobně jako simplexové metody.

### 5.3.4 Algoritmus řešení parametrické úlohy

Přípravný krok

Položíme všechna  $t_{ij} = D_{ij}, T_0 = 0$  a vypočteme všechna  $T_j = \max_i \{T_i + D_{ij}\} = T_j^{(0)}$  ( $T_n = M$ ). Určíme časové rezervy pro všechny činnosti a strukturu získaného optimálního plánu.

Obecný krok

1. Podle optimálního plánu (31)–(34) pro  $T_n = \lambda \leq M$  a podle struktury sestavíme úlohu (47)–(51) o maximálním toku a řešíme ji podle kapitoly (5.3.5).

2. Podle (53) (strana 51) určíme  $\xi_{ij}$  a  $\eta_j$  a podle nich horní hranici  $\Theta_0$  a změny parametru  $\Theta$  podle (45).
3. Podle (35) najdeme nový optimální plán parametrické úlohy trvání  $T_n = \lambda - \Theta$ , kde  $0 \leq \Theta \leq \Theta_0$ . Položíme  $\Theta = \Theta_0$  a určíme strukturu získaného optimálního plánu. Pokud je třeba, opakujeme obecný krok. Algoritmus je ukončen, pokud v  $k$ -tém kroku je lineární forma úlohy neomezena a soustava omezení duální úlohy není konzistentní  $\Rightarrow$  neexistuje plán kratší, než je plán v  $(k - 1)$ -ním kroku.

### 5.3.5 Speciální algoritmus pro řešení úlohy o maximálním toku

V přípravném kroku položíme všechna  $t_{ij} = D_{ij}$ , díky čemuž budou činnosti zařazeny pouze do množin  $Q_1$  a  $Q_2$ . Proměnné  $x_{ij}$  pak musí vyhovovat omezením:

$$\left. \begin{array}{l} 0 \leq x_{ij} \leq a_{ij} \quad \text{pro } (P_i, P_j) \in E_1 \cap Q_1 \\ x_{ij} \geq 0 \quad \text{pro } (P_i, P_j) \in E_1 \cap Q_2 \\ x_{ij} = 0 \quad \text{pro } (P_i, P_j) \in E_2 \end{array} \right\} \quad (52)$$

Položíme-li všechna  $x_{ij} = 0$  obdržíme přípustný tok. V dalších krocích se trvání  $T_n$  plánu zkracuje, tedy mění se trvání  $t_{ij}$  některých činností. Ty už nyní mohou patřit do libovolné množiny  $Q$ . Pro úlohu o maximálním toku mohou existovat libovolná omezení pro  $x_{ij}$ , takže tok, pro který všechna  $x_{ij} = 0$ , již není přípustný.

Vzhledem k tomu, že řešení získané v  $k$ -tém kroku je přípustné i pro  $(k + 1)$ -ní krok, jsou v algoritmu dva přípravné kroky. První se použije jen při prvním kroku, druhý se používá při řešení úlohy vzniklé v dalších krocích.

*První přípravný krok:*

Omezení (52) zapíšeme ve tvaru tabulky. Tu vyplníme podle pravidla: je-li  $(P_i, P_j) \in E_1 \cap Q_1$ , do políčka  $(i, j)$  zapíšeme číslo  $a_{ij}$ , je-li  $(P_i, P_j) \in E_1 \cap Q_2$ , do políčka  $(i, j)$  zapíšeme  $\infty$ . Do políčka  $(j, i)$  zapíšeme v obou případech 0.

Pokud  $(P_i, P_j) \in E_2$ , nebo uzly  $P_i$  a  $P_j$  nejsou spojeny hranou, zůstane políčko prázdné.

Nyní přichází na řadu *obecný krok*

Podle tabulky vyhledáme cestu z  $P_0$  do  $P_n$  a procházíme ji za předpokladu, že  $a_{ij} > 0$ . Pokud ne, hledáme jinou cestu. Pokud neexistuje cesta z  $P_0$  do  $P_n$ , je proces ukončen.

Pokud takovou cestu nalezneme, mezi čísly  $a_{ij}^-$  najdeme  $\min\{a_{ij}\} = h$ , – propustnost sestrojené cesty.

Odečteme  $h$  ode všech  $a_{ij}^-$  a přičteme ke všem  $a_{ij}^+$ . Získáme novou tabulku a vrátíme se ke kroku vyhledávání cesty z  $P_0$  do  $P_n$ .

Je-li operace zakončena tím, že neexistuje cesta z  $P_0$  do  $P_n$ , je proces ukončen.

Množinu vrcholů sítě, které podle poslední tabulky ještě můžeme spojit nějakou cestou s  $P_0$  označíme  $U$  a množinu ostatních uzlů označíme  $V$ . Hrany, jejichž konce náležejí do různých množin, nazveme *řez*  $(U, V)$  [1].

*Druhý přípravný krok:*

Po  $k$ -tém kroku máme tabulku odpovídající úlohy o maximálním toku.

V  $(k + 1)$ -ním kroku do ní zaneseme některé změny související se změnou struktury optimálního plánu.

Jestliže v novém  $k$ -tém optimálním plánu patří činnost  $(P_i, P_j)$  do stejné množiny  $E$  a do stejné třídy  $Q$  jako v kroku  $(k - 1)$ -ním, potom políčka  $(i, j)$  a  $(j, i)$  zůstávají beze změny.

Pokud nepatří do stejné množiny a třídy změní se podle následujících kritérií:

- 1) pokud  $(P_i, P_j) \in E_1 \cap Q_3 \Rightarrow (i, j) = \infty; (j, i) = 0;$
- 2) pokud  $(P_i, P_j) \in E_1 \cap Q_4 \Rightarrow (i, j) = 0; (j, i) = 0;$
- 3) pokud  $(P_i, P_j) \in E_1 \cap Q_2 \Rightarrow (i, j) = \infty; (j, i) = 0;$
- 4) pokud  $(P_i, P_j) \in E_1 \cap Q_1;$ 
  - a) pro  $P_i \in U, P_j \in v \Rightarrow (i, j) = a_{ij}; (j, i) = 0;$
  - b) pro  $P_i \in V, P_j \in U \Rightarrow (i, j) = 0; (j, i) = a_{ij};$
- 5) pokud  $(P_i, P_j) \in E_2 \cap Q_2 \Rightarrow$  políčka  $(i, j)$  a  $(j, i)$  nevyplňujeme.

Nyní můžeme opět pokračovat obecným krokem.

Je možné ukázat, že řešení  $\{\xi_{ij}^*, \eta_j^*\}$  duální úlohy jsou určena takto [1]:

$$\left. \begin{array}{l} \xi_{ij}^* = \left\{ \begin{array}{ll} 1 & \text{pro } (P_i, P_j) \in E_1 \cap (Q_1 \cup Q_4), P_i \in U, P_j \in V, \\ -1 & \text{pro } (P_i, P_j) \in E_1 \cap (Q_3 \cup Q_4), P_i \in U, P_j \in V, \\ 0 & \text{v ostatních případech;} \end{array} \right. \\ \eta_j^* = \left\{ \begin{array}{ll} 0 & \text{pro } P_j \in U, \\ 1 & \text{pro } P_j \in V. \end{array} \right. \end{array} \right\} \quad (53)$$

## 5.4 NĚKTERÉ APLIKACE KELLEYOVY METODY

### 5.4.1 Vyhledání optimálního plánu vzhledem k nákladům

Algoritmus popsaný v kapitole 5.3.5 je použitelný i pro úlohy minimalizace nákladů projektu při jeho pevném trvání. V takovém případě je algoritmus ukončen po dosažení daného trvání  $T_n = T$ .

### 5.4.2 Optimální plán vzhledem k času

Tato úloha je v jistém smyslu opačná k předcházejícím – máme zde pevně dané náklady a musíme stanovit optimální (minimální) termín dokončení projektu.

Matematická formulace úlohy zde znamená: nalézt minimum funkce

$$z = T_n \quad (54)$$

při omezeních

$$\left. \begin{array}{l} T_j - T_i - T_{ij} \geq 0 \quad \text{pro všechna } (P_i, P_j), \\ 0 \leq d_{ij} \leq t_{ij} \leq D_{ij} \quad \text{pro všechna } (P_i, P_j), \end{array} \right\} \quad (55)$$

$$T_0 = 0;$$

$$\sum_{(P_i, P_j)} (-a_{ij}t_{ij} + b_{ij}) \leq C \quad \text{lineární varianta} \quad (56)$$

nebo

$$\sum_{(P_i, P_j)} \frac{a_{ij}}{t_{ij}} \leq C \quad \text{konvexní varianta} \quad (57)$$

Jestliže položíme  $t_{ij} = D_{ij}$  a najdeme odpovídající náklady na projekt  $C_M$ , je pro  $C < C_M$  je úloha neřešitelná. Položíme-li  $T_n = m$  ( $m$  je pevné omezení trvání projektu) a najdeme odpovídající minimální náklady  $C_m$  na projekt, bude pro  $C \geq C_m$  zřejmě  $z = m = T_n$  minimální vypočtené trvání projektu [1].

Obě varianty – lineární i konvexní lze řešit pomocí Kelleyova algoritmu.

## 6 MAXIMÁLNÍ TOKY V SÍTI

### 6.1 PROSTÁ ÚLOHA O MAXIMÁLNÍM TOKU

#### 6.1.1 Základní pojmy a formulace problému

Je dána síť  $G$  tvořená  $n + 1$  uzly  $P_0, P_1, \dots, P_n$  a hranami  $(P_i, P_j)$ . Síť je symetrický graf, to znamená, že do něj patří jak hrana  $(P_i, P_j)$ , tak i hrana  $(P_j, P_i)$  k ní symetrická.

Po cestách  $\mu(P_0, P_{i_1}, P_{i_2}, \dots, P_{i_k}, P_n)$  sestavených z hran  $(P_0, P_{i_1}), (P_{i_1}, P_{i_2}), \dots, (P_{i_k}, P_n)$  a netvořících smyčky je poslán materiál (kapalina, plyn, náklad) z uzlu  $P_0$  do uzlu  $P_n$ . Každé uspořádané dvojici uzlů  $P_i, P_j$  je přiřazeno nezáporné číslo  $c_{ij}$ , které se nazývá *propustnost hrany*  $(P_i, P_j)$  a určuje maximální množství látky, kterou může propustit hrana  $(P_i, P_j)$  za jednotku času, přičemž  $c_{i0} = c_{nj} = 0$  [1].

*Tokem*  $x_{ij}$  v hraně  $(P_i, P_j)$  ( $i, j = 0, 1, \dots, n; i \neq j$ ) se nazývá množství látky, které projde v této hraně za jednotku času [1]. *Tokem v síti* nebo prostě *tokem* budeme nazývat množinu  $\{x_{ij}\}$  toků po všech hranách sítě. Dále budeme předpokládat, že toky vyhovují následujícím omezením:

$$0 \leq x_{ij} \leq c_{ij} \quad (i, j = 0, 1, \dots, n; i \neq j) \quad (58)$$

$$\sum_{i=0}^{n-1} x_{ik} - \sum_{j=1}^n x_{kj} = 0 \quad (k = 1, \dots, n-1). \quad (59)$$

Ta znamenají, že tok v každé hraně je nezáporný a nepřevyšuje její propustnost (58) a že množství látky vstupující do uzlu (mimo uzly  $P_0$  a  $P_n$ ) je rovno množství látky z něj vystupující (59).

Tok vyhovující těmto omezením budeme nazývat *přípustným tokem*.

Z omezení (59) vyplývá, že množství látky vytékající z  $P_0$  je rovno celkovému množství látky přitékající do  $P_n$ , tj.

$$\sum_{j=1}^n x_{0j} = \sum_{i=0}^{n-1} x_{in} = z \quad (60)$$

Úloha o maximálním toku v síti spočívá v nalezení takového řešení  $x_{ij}^*$  ( $i, j = 0, 1, \dots, n; i \neq j$ ) soustavy (58) – (59), které maximalizuje lineární formu (60). Toto řešení  $\{x_{ij}^*\}$  se nazývá *maximální tok v síti* [1].

Dále si zavedeme některé doplňující pojmy. Množinu všech uzlů sítě  $G$  rozložíme na dvě disjunktní podmnožiny  $U$  a  $V$ , přičemž  $P_0 \in U$  a  $P_n \in V$ . *Řezem*  $(U, V)$  sítě  $G$  nazveme množinu všech hran, jejichž konce patří různým podmnožinám.

Každému řezu  $(U, V)$  přiřadíme číslo  $C(U, V)$  – propustnost řezu, která je rovna součtu propustností všech hran řezu začínajících v  $U$  a končících ve  $V$ :

$$C(U, V) = \sum_{P_i \in U; P_j \in V} c_{ij}$$

Libovolná cesta z  $P_0$  do  $P_n$  obsahuje alespoň jednu hranu řezu  $(U, V)$ , která začíná v  $U$  a končí ve  $V$ . Vzhledem k tomu, že propustnost cesty není větší než propustnost každé

její hrany, nemůže být ani velikost libovolného toku z  $P_0$  do  $P_n$ , která je součtem všech cest z  $P_0$  do  $P_n$  větší než propustnost libovolného řezu  $(U, V) \Rightarrow z \leq C(U, V)$ .

Z toho tedy plyne, že pokud se podaří vytvořit takový tok  $\{x_{ij}^*\}$ , že jeho velikost  $z^*$  bude rovna propustnosti některého řezu  $(U^*, V^*)$ , tedy  $z^* = C(U^*, V^*)$ , je tento tok maximální a  $(U^*, V^*)$  je řez s minimální propustností.

## 6.2 ZOBECNĚNÁ ÚLOHA O MAXIMÁLNÍM TOKU

### 6.2.1 Formulace úlohy

Zobecněnou úlohou o maximálním toku v síti  $G$ , znázorněné asymetrickým grafem, při oboustranných nenulových omezeních toku v každé hraně  $(P_i, P_j)$ , tedy úlohu vyhledání maxima funkce

$$z = \sum_{i=0}^{n-1} x_{in} = \sum_{j=1}^n x_{0j} \quad (61)$$

při omezeních

$$0 \leq b_{ij} \leq x_{ij} \leq c_{ij} \quad (i, j = 0, 1, \dots, n, 1 \neq 1) \quad (62)$$

a

$$\sum_{i=0}^{n-1} x_{ik} - \sum_{j=1}^n x_{kj} = 0 \quad (k = 1, \dots, n = 1), \quad (63)$$

dále budeme předpokládat, že  $c_{ij} = b_{ij} = 0$  pro  $(P_i, P_j) \in G [1]$ .

Pokud všechna  $b_{ij} = 0$ , pak úlohy (61) – (63) přejdou do úloh (58) – (60) uvažovaných na asymetrickém grafu. Můžeme předpokládat, že pokud  $(P_i, P_j) \in G$ , pak také  $(P_j, P_i) \in G$ , ale  $c_{ij} = 0$ .

Rozdíl mezi úlohami (61) – (63) a (58) – (60) je v tom, že pro první vždy existuje známý přípustný tok  $x_{ij} = 0$ ;  $(i, j = 0, 1, \dots, n, i \neq j)$ , ale pro druhou (tedy vyhovující omezením (62) – (63) při existenci alespoň jednoho  $b_{ij} > 0$ ) nemusí existovat přípustný tok (úloha nemá řešení). Pokud má řešení, vzniká otázka, jak najít přípustný tok.

### 6.2.2 Algoritmus

1. Sestavíme tabulku; je-li  $c_{ij} >$ , do políčka  $(P_j, P_i)$  píšeme  $c_{ji} = 0$ , pokud je  $c_{ij} = c_{ji} = 0$ , nepíšeme nic.
2. Tabulku doplníme o řádky a sloupce  $P_{-1}$  a  $P_{n+1}$ . Vyplňujeme podle pravidel - do políčka  $(P_i, P_j)$   $(i, j = 0, 1, \dots, n; i \neq j)$  místo čísla  $c_{ij} > 0$  píšeme rozdíl  $c_{ij} - b_{ij}$  a v symetrických políčkách napíšeme nuly. Do políčka  $(P_{-1}, P_j)$  napíšeme součet  $\sum_{i=0}^{n-1} b_{ij}$  a do každého políčka  $(P_i, P_{n+1})$  napíšeme odpovídající součet  $\sum_{j=1}^n b_{ij}$ . Symetrická políčka vyplníme nulami. Nakonec do políčka  $(P_n, P_0)$  napíšeme  $\infty$  a jestliže je políčko  $(P_0, P_n)$  prázdné, napíšeme do něj nulu.
3. V pomocné tabulce máme vyjádřeny propustnosti sítě. Vyhledáme maximální tok.

4. Z výsledné tabulky odmažeme přidané sloupce i řádky. Opravíme políčka  $(P_0, P_n)$  a  $(P_n, P_0)$ : pokud bylo  $c_{0n} > 0$  zapíšeme do  $(P_0, P_n)$  rozdíl  $c_{0n} - b_{0n}$  a do políčka  $(P_n, P_0)$  nulu. Pokud je  $c_{0n} = 0$ , to znamená  $(P_0, P_n) \in G$ , políčka  $(P_0, P_n)$  a  $(P_n, P_0)$  budou prázdná.
5. Znovu použijeme algoritmus pro vyhledání maximálního toku pro tabulku z kroku 4 až do získání konečné tabulky.
6. Vypočítáme maximální tok tak, že odečteme ode všech prvků  $c_{ij}$  výchozí tabulky odpovídající prvky konečné tabulky.

### 6.3 POUŽITÍ V SÍŤOVÉ ANALÝZE

#### 6.3.1 Duální úloha k úloze o maximálním toku

Připomeňme si, že každý krok algoritmu pro řešení parametrické úlohy minimalizace nákladů na projekt má tyto body:

1. Konstrukci zobecněné úlohy o maximálním toku a její řešení
2. Vyhledání hodnot proměnných odpovídající duální úlohy
3. Konstrukci nového optimálního plánu a určení jeho struktury [1].

Nechť byl v  $k$ -tém kroku získán optimální plán parametrické úlohy a určena jeho struktura (viz kapitola 5.3.1), potom podmínky odpovídající dvojici duálních úloh řešených v  $k+1$ -ním kroku budou vypadat – [1, dodatek, §2 odstavec 1].

Máme optimální řešení  $\{x_{ij}^*\}$  a optimální řešení  $\{\xi_{ij}^*, \eta_j^*\}$  duální úlohy. Podle kritéria optimálnosti (druhá věta duality) jsou přípustná řešení vzájemně duálních úloh optimální tehdy a jen tehdy, když vyhovují doplňujícím podmínkám [1]. Zapišeme tedy podmínky ve tvaru:

$$x_{ij}^*(\xi_{ij}^* + \eta_i^* - \eta_j^*) = 0 \quad \text{pro} \quad (P_i, P_j) \in E_1 \cup E_2 \quad (64)$$

a

$$(x_{ij}^* - a_{ij})\xi_{ij}^* = 0 \quad \text{pro} \quad (P_i, P_j) \in E_1 \cup E_2. \quad (65)$$

Vyhledané  $\{x_{ij}^*, \eta_j^*\}$  musí být přípustné (vyhovovat podmínkám duality) a musí splňovat podmínky (64) – (65).

Uvažujeme nyní hrany  $(P_i, P_j) \in E_1$ . Při řešení úlohy o maximálním toku určíme i minimální řez  $(U, V)$ . Pro všechny hrany  $(P_i, P_j) \in E_1 \setminus (U, V)$  je možno položit:

$$\xi_{ij}^* = 0 \quad (66)$$

$$\xi_{ij}^* + \eta_i^* - \eta_j^* = 0. \quad (67)$$

Pro všechna  $\eta_j^*$  máme

$$\eta_j^* = \begin{cases} 0, & P_j \in U, \\ 1, & P_j \in V. \end{cases} \quad (68)$$

Hodnoty  $\xi_{ij}^*$  pro hrany  $(P_i, P_j) \in E_1 \cap (U, V)$  jsou určeny vzorci (53).

Hledané optimální řešení  $\{\xi_{ij}^*, \eta_j^*\}$  duální úlohy vypočteme podle vzorců

$$\xi_{ij}^* = \begin{cases} 1 & \text{pro } (P_i, P_j) \in (E_1 \cap Q_1) \cup (E_1 \cap Q_4), P_i \in U, P_j \in V, \\ -1 & \text{pro } (P_i, P_j) \in (E_1 \cap Q_3) \cup (E_j \cap Q_4), P_j \in U, P_i \in V, \\ 0 & \text{v ostatních případech,} \end{cases} \quad (69)$$

$$\eta_0^* = \begin{cases} 0 & \text{pro } P_j \in U, \\ 1 & \text{pro } P_j \in V. \end{cases} \quad (70)$$

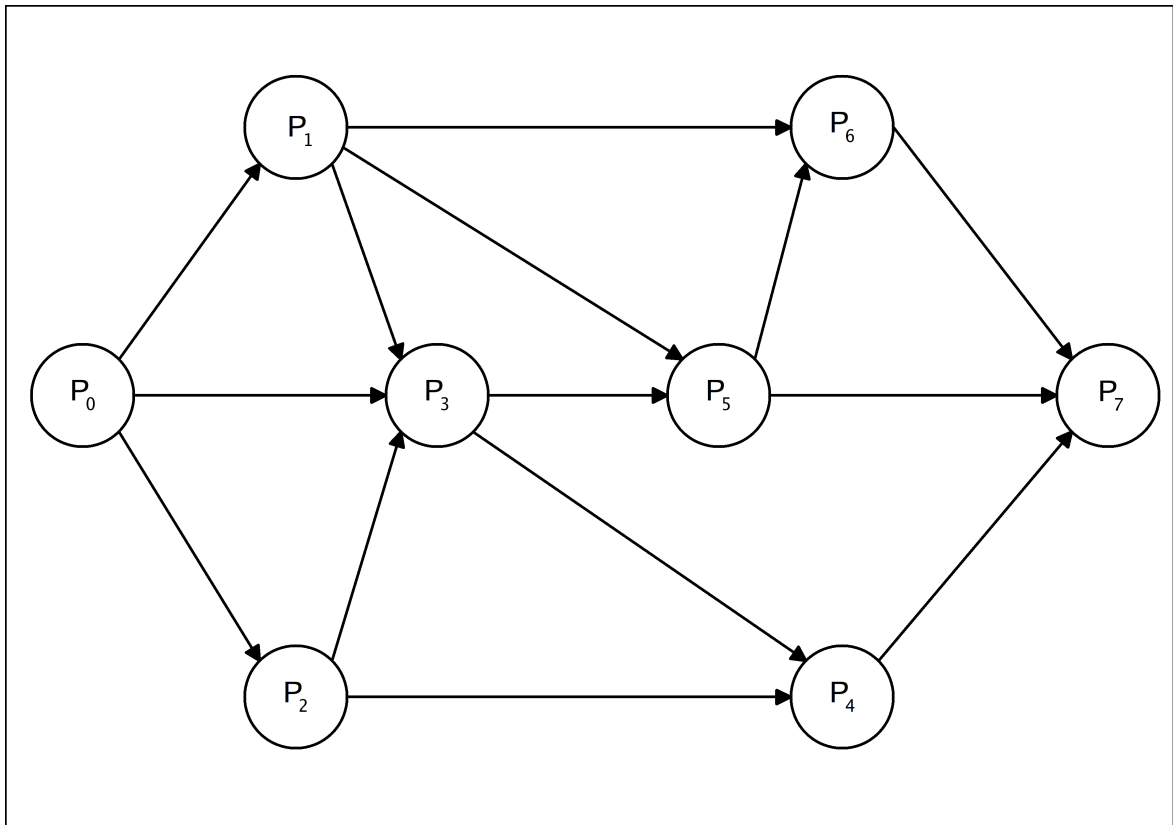


## **III. PROJEKTOVÁ ČÁST**

## 7 PŘÍKLAD

### 7.1 ZADÁNÍ

Vzhledem k tomu že, jak bylo ukázáno v teoretické části, úlohy optimalizace v síťovém grafu jsou řešeny pomocí opakovaného procházení tohoto grafu, pro praktickou ukázkou vybereme čistě matematický model malého rozsahu. Použijeme graf se sedmi uzly (jeho přečíslovaná verze byla použita jako ukáзка v kapitole (2.2) obrázek (7)).



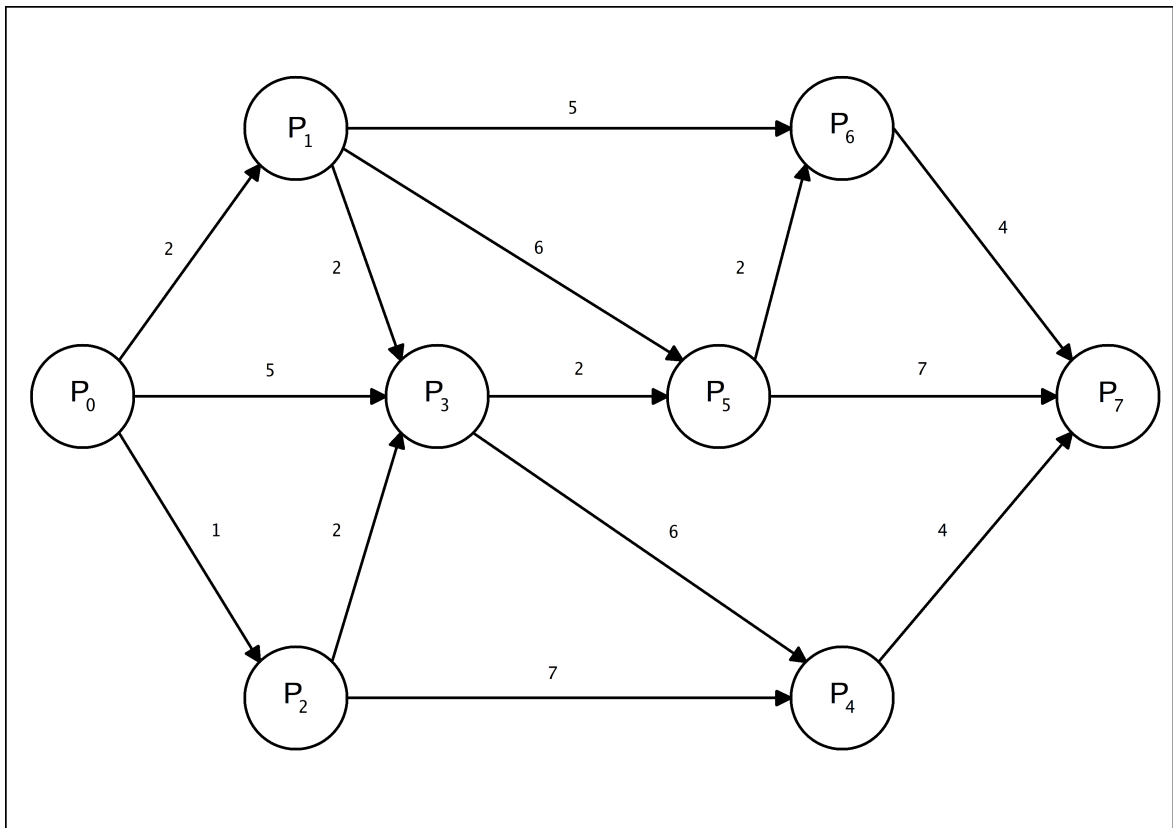
Obr. 11. Přečíslovaný graf pro praktickou ukázkou.

#### 7.1.1 Graf

Nyní doplníme do grafu k jednotlivým činnostem  $(P_i, P_j)$  doby jejich trvání  $t_{ij}$  – zapisujeme je k hranám – viz obrázek (12). Takto doplněný graf už obsahuje všechny údaje potřebné pro výpočet Trvání projektu, určení kritické cesty, vyhledání nejdříve možných i nejpozději přípustných termínů. . .

#### 7.1.2 Převod do tabulky

Pro některé výpočty je přehlednějším způsobem zápisu projektu tabulka obsahující seznam činností (je použit už z přečíslovaného grafu, čísla v kroužku u hrany zde označují číslo činnosti), následná činnost a trvání.



Obr. 12. Graf se zapsanými dobami trvání jednotlivých činností.

A takto vypadá převedená do tabulky, kde ke každé činnosti jsou přiřazena čísla následných činností a doba trvání dané činnosti.

Tab. 2. Zápis grafu z obrázku (12) do tabulky

Činnost	Následující činnost	Trvání
1	2,3	2
2	8	3
3	6,7	4
4	6,7	5
5	9	4
6	8	6
7	—	4
8	—	2
9	—	7

## 7.2 TRVÁNÍ PROJEKTU A KRITICKÁ CESTA

### 7.2.1 Vyhledání nejdříve možných začátků činností a trvání projektu

Vycházíme z algoritmu pro přečíslování uzlů, pomocí něj jsme určili i řády jednotlivých uzlů v přečíslovaném grafu takto:

$P_0$  – uzel nultého řádu

$P_1, P_2$  – uzly prvního řádu

$P_3$  – uzel druhého řádu

$P_4, P_5$  – uzly třetího řádu

$P_6$  – uzel čtvrtého řádu

$P_7$  – uzel pátého řádu

Je zde vidět dodržení pravidla pro přečíslovaný graf – pro každou činnost mezi dvěma uzly  $(P_i, P_j)$  platí  $i < j$ .

Dále budeme postupovat podle algoritmu z kapitoly 3.1.1, strana 26.

V prvním kroku přiřadíme každému uzlu  $P_j$  číslo  $\lambda_1 = 0$ .

Dále procházíme síť od uzlu  $P_0$  a podle vzorce (1) vypočteme všechna  $\lambda'_j = T_j^{(0)}$ , tedy nejdříve možné začátky jednotlivých uzlů:

$$\lambda'_0 = \lambda_0 = 0 = T_0,$$

$$\lambda'_1 = \lambda'_0 + t_{01} = 2 = T_1^{(0)} \quad (0),$$

$$\lambda'_2 = \lambda'_0 + t_{02} = 1 = T_2^{(0)} \quad (0),$$

$$\lambda'_3 = \max\{\lambda'_0 + t_{03}, \lambda'_1 + t_{13}, \lambda'_2 + t_{23}\} = \max\{5, 4, 3\} = 5 = T_3^{(0)} \quad (0)$$

$$\lambda'_4 = \max\{\lambda'_2 + t_{24}, \lambda'_3 + t_{34}\} = \max\{8, 11\} = 11 = T_4^{(0)} \quad (3),$$

$$\lambda'_5 = \max\{\lambda'_3 + t_{35}, \lambda'_1 + t_{15}\} = \max\{7, 8\} = 8 = T_5^{(0)} \quad (1),$$

$$\lambda'_6 = \max\{\lambda'_5 + t_{56}, \lambda'_1 + t_{16}\} = \max\{10, 7\} = 10 = T_6^{(0)} \quad (5),$$

$$\lambda'_7 = \max\{\lambda'_4 + t_{47}, \lambda'_5 + t_{57}, \lambda'_6 + t_{67}\} = \max\{15, 15, 14\} = 15 = T_7^{(0)} \quad (4, 5).$$

Trvání projektu  $T_7^{(0)}$  (nejdříve možný termín konečného uzlu) je tedy roven hodnotě 15.

### 7.2.2 Vyhledání kritické cesty a volných rezerv

Pro přečíslovanou síť (obr. 11) vytvoříme tabulku (Tab.3, strana 61).

Z tabulky vidíme, že činnosti  $(P_1, P_3)$ ,  $(P_2, P_3)$ ,  $(P_2, P_4)$ ,  $(P_3, P_5)$ ,  $(P_1, P_6)$  a  $(P_6, P_7)$  mají volné rezervy. U ostatních činností je rezerva rovna 0.

Nyní budeme procházet síť od uzlu  $P_7$ . Do něj vstupují dvě činnosti splňující podmínku (2) –  $(P_4, P_7)$  a  $(P_5, P_7)$ . Postupujeme tedy dále k událostem  $P_4$  a  $P_5$ . Do  $P_4$  vstupuje činnosti  $(P_3, P_4)$  bez volné rezervy a do uzlu  $P_5$  je to činnost  $(P_1, P_5)$ . Dále postupujeme k uzlu  $P_3$ , do nějž vstupuje činnost  $(P_0, P_3)$  s volnou rezervou rovnou 0 a k uzlu  $P_1$ , do nějž vstupuje činnost  $(P_0, P_1)$  s nulovou volnou rezervou. Kritické cesty jsou tedy dvě a to:  $(P_0, P_1, P_5, P_7)$  a  $(P_0, P_3, P_4, P_7)$ . Nyní je můžeme zaznačit do síťového grafu – obrázek (13).

Tab. 3. Tabulka pro určení kritické posloupnosti činností a volných rezerv (dva díly tabulky pod sebou)

$(P_i, P_j)$	$(P_0, P_1)$	$(P_0, P_2)$	$(P_0, P_3)$	$(P_1, P_3)$	$(P_2, P_3)$	$(P_2, P_4)$	$(P_3, P_4)$
$T_j^{(0)} - T_i^{(0)}$	2	1	5	3	4	10	6
$t_{ij}$	2	1	5	2	2	7	6
$T_j^{(0)} - T_i^{(0)} - t_{ij}$	0	0	0	1	2	3	0

$(P_i, P_j)$	$(P_1, P_5)$	$(P_3, P_5)$	$(P_1, P_6)$	$(P_5, P_6)$	$(P_4, P_7)$	$(P_5, P_7)$	$(P_6, P_7)$
$T_j^{(0)} - T_i^{(0)}$	6	3	8	2	4	7	5
$t_{ij}$	6	2	5	2	4	7	4
$T_j^{(0)} - T_i^{(0)} - t_{ij}$	0	1	3	0	0	0	1

### 7.3 NALEZENÍ NEJPOZDĚJI PŘÍPUSTNÝCH TERMÍNŮ

Máme určeny základní parametry síťového grafu – celkovou dobu trvání a kritickou cestu. Nyní určíme nejpozději přípustné termíny uzlů  $T_j^{(1)}$ , podle pravidel (3) a (4) ze strany (28).

V našem ukázkovém grafu tedy  $T_{kr} = T_7$ . Podle algoritmu nejdříve definujeme všechna  $\mu_j = 0$  (tedy  $\mu_0 = \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = \mu_7 = 0$ ).

V dalším kroku postupujeme grafem od  $P_7$  směrem dolů a vypočítáme všechna  $\mu'_1$ :

$$\mu'_7 = \mu_7 = 0,$$

$$\mu'_6 = \mu'_7 = t_{67} = 4,$$

$$\mu'_5 = \max\{\mu'_6 + t_{56}, \mu'_7 + t_{57}\} = \max\{6, 7\} = 7,$$

$$\mu'_4 = \mu'_7 = +t_{47} = 4,$$

$$\mu'_3 = \max\{\mu'_5 + t_{35}, \mu'_4 + t_{34}\} = \max\{9, 10\} = 10,$$

$$\mu'_2 = \max\{\mu'_4 + t_{24}, \mu'_3 + t_{23}\} = \max\{11, 12\} = 12,$$

$$\mu'_1 = \max\{\mu'_6 + t_{16}, \mu'_5 + t_{15}, \mu'_3 + t_{13}\} = \max\{9, 13, 12\} = 13,$$

$$\mu'_0 = \max\{\mu'_3 t_{03}, \mu'_2 + t_{02}, \mu'_1 + t_{01}\} = \max\{15, 13, 15\} = 15.$$

Podle vzorce (4) tedy určíme nejpozději nutné termíny  $T_j^{(1)}$ :

$$T_7^{(1)} = T_7 = 15, \quad T_3^{(1)} = 15 - 10 = 5$$

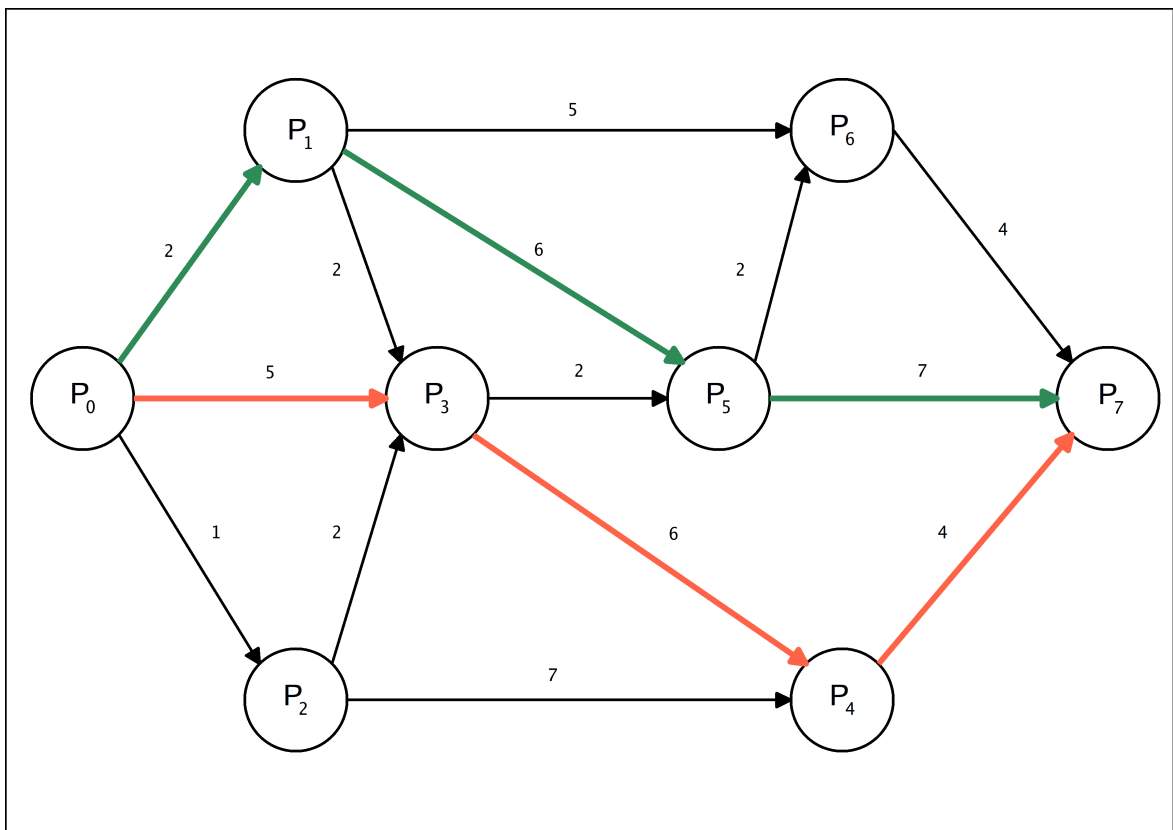
$$T_6^{(1)} = 15 - 4 = 11 \quad T_2^{(1)} = 15 - 12 = 3$$

$$T_5^{(1)} = 15 - 7 = 8 \quad T_1^{(1)} = 15 - 13 = 2$$

$$T_4^{(1)} = 15 - 4 = 11 \quad T_0^{(1)} = 15 - 15 = 0$$

### 7.4 ZJIŠTĚNÍ PODMÍNEK KRITICHNOSTI UZLŮ A ČINNOSTÍ

Jak bylo řečeno v kapitole 3.3, uzel leží na kritické cestě, když  $T_j^{(1)} = T_j^{(0)}$ .



Obr. 13. Síťový graf se zaznačenými kritickými cestami.

Porovnáme dříve vypočítané hodnoty:

nejdříve možný termín	nejpozději přípustný termín	výsledek porovnání	uzel
$T_0^{(1)} = 0$	$T_0^{(0)} = 0$	=	$P_0$
$T_1^{(1)} = 2$	$T_1^{(0)} = 2$	=	$P_1$
$T_2^{(1)} = 3$	$T_2^{(0)} = 1$	$\neq$	$P_2$
$T_3^{(1)} = 5$	$T_3^{(0)} = 5$	=	$P_3$
$T_4^{(1)} = 11$	$T_4^{(0)} = 11$	=	$P_4$
$T_5^{(1)} = 8$	$T_5^{(0)} = 8$	=	$P_5$
$T_6^{(1)} = 11$	$T_6^{(0)} = 10$	$\neq$	$P_6$
$T_7^{(1)} = 15$	$T_7^{(0)} = 15$	=	$P_7$

## 7.5 ČASOVÉ REZERVY

Celkové rezervy činnosti  $(P_i, P_j)$  vyjadřují maximální počet časových jednotek, který máme k dispozici, aniž se prodlouží trvání  $T_n$ , v našem případě  $T_7$  projektu. Spočítáme je podle vzorce:  $T_j^{(1)} - T_i^{(0)} - t_{ij}$ .

Volné rezervy činnosti  $(P_i, P_j)$  vyjadřují maximálně přípustné prodloužení trvání činnosti (nebo zpoždění začátku proti  $T_i^{(0)}$ ), které nenaruší možnost, aby všechny činnosti vystupující z  $P_j$  začínaly v termínu  $T_j^{(0)}$ . Určíme je podle vztahu:  $T_j^{(0)} - T_i^{(0)} - t_{ij}$ .

A nakonec určíme nezávislou rezervu činnosti  $(P_i, P_j)$ , která vyjadřuje maximální počet časových jednotek prodloužení, nebo zpoždění začátku činnosti, aby byla splněna podmínka, že všechny činnosti vstupující do uzlu  $P_i$  končí nejpozději v  $T_i^{(1)}$  a všechny činnosti vystupující z  $P_j$  začínají v nejdříve možném termínu  $T_1^{(0)}$ . Je vyjádřena vztahem:  $\max\{0, T_j^{(1)} - T_i^{(0)} - t_{ij}\}$ .

Celkový přehled rezerv činností je v tabulce (4).

Tab. 4. Přehled časových rezerv pro jednotlivé činnosti.

činnost $(P_i, P_j)$	$T_j^{(1)}$	$T_j^{(0)}$	$t_{ij}$	celková rezerva $T_j^{(1)} - T_i^{(0)} - t_{ij}$	volná rezerva $T_j^{(0)} - T_i^{(0)} - t_{ij}$	nezávislá rezerva $\max\{0, T_j^{(1)} - T_i^{(0)} - t_{ij}\}$
$(P_0, P_1)$	2	0	2	0	0	0
$(P_0, P_2)$	3	0	1	2	0	0
$(P_0, P_3)$	5	0	5	0	0	0
$(P_1, P_3)$	5	2	2	1	1	1
$(P_2, P_3)$	5	1	2	2	2	0
$(P_2, P_4)$	11	1	7	3	3	1
$(P_3, P_4)$	11	5	6	0	0	0
$(P_1, P_5)$	8	2	6	0	0	0
$(P_3, P_5)$	8	5	2	1	1	1
$(P_1, P_6)$	11	2	5	4	3	3
$(P_5, P_6)$	11	8	2	1	0	0
$(P_4, P_7)$	15	11	4	0	0	0
$(P_5, P_7)$	15	8	7	0	0	0
$(P_6, P_7)$	15	10	4	1	1	0

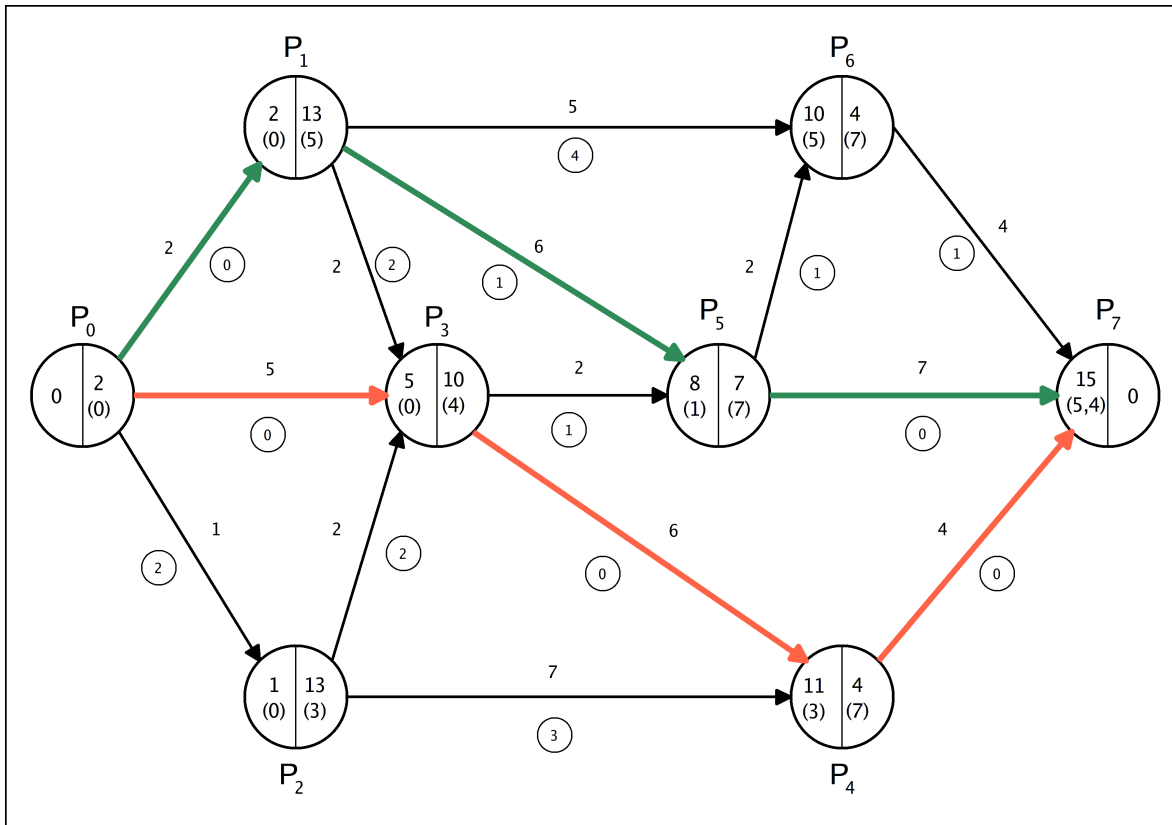
## 7.6 SUBKRITICKÉ ČINNOSTI

Jak bylo řečeno v kapitole 3.5, subkritické činnosti jsou takové, jejichž trvání se od kritických liší maximálně o hodnotu  $\delta$  – odchylku od kritické cesty.

Subkritické jsou tedy všechny činnosti, jejichž celkové rezervy jsou menší než  $\delta$ . Vytvářejí tak subkritické cesty, jejichž trvání  $L$  vyhovuje nerovnosti:  $T_{kr} - \delta \leq L \leq T_{kr}$ .

Pro náš příklad zvolíme  $\delta = 1$  a určíme cesty, které se liší od kritické nejvýše o tuto hodnotu.

Upravíme si síťový graf tak, aby v něm byly všechny potřebné údaje – uzel rozdělíme a do levé části napíšeme hodnotu  $T_i^{(0)}$  a číslo uzlu pro který je platné. Vpravo napíšeme  $\mu_i^*$  a jemu odpovídající číslo. Číslo v kroužku u hrany vyjadřuje celkové rezervy.



Obr. 14. Graf s hodnotami pro určování subkritických činností a cest.

Nyní procházíme jednotlivé cesty a určíme všechny, které jsou subkritické:

$(P_0, P_1, P_3, P_4, P_7)$ ,

$(P_0, P_1, P_5, P_6, P_7)$ ,

$(P_0, P_3, P_5, P_7)$ ,

a dále k nim přiřadíme již nalezené kritické cesty (ty samozřejmě také vyhovují podmínce subkritičnosti):

$(P_0, P_1, P_5, P_7)$ ,

$(P_0, P_3, P_4, P_7)$ .

Tímto jsme vyšetřili všechny hlavní parametry síťového grafu. Známe celkovou dobu trvání projektu, máme určené kritické činnosti a kritickou cestu. Určili jsme také časové rezervy jednotlivých činností a nyní jsme si stanovili subkritické činnosti (dle zvolené proměnné blízké kritickým činnostem) a cesty, které jsou jimi ovlivněny.



## ZÁVĚR

Cílem této bakalářské práce bylo vypracovat přehled optimalizačních metod a seznámit se s metodami síťové analýzy. Vybranou z nich potom popsat a zhodnotit.

V úvodu jsme si uvedli poměrně rozsáhlý přehled metod optimalizace, rozdělený podle základní klasifikace. U jednotlivých metod byly stručně popsány principy a postupy se kterými pracují.

Stručně jsme se také dotkli základních pojmů z oblasti síťových grafů. Popsali jsme si některá pravidla jejich sestavování a popsali si algoritmy pro číslování uzlů a vyhledávání cyklů.

V hlavní části této práce jsme si popsali jednotlivé metody optimalizace projektů popsaných síťovým grafem. Nejdříve jsme si v rámci projektu stanovili kritické termíny a činnosti, dále jsme se věnovali metodám optimalizujícím dané činnosti z různých pohledů. Vzhledem k tomu, že o peníze jde až v první řadě, skutečným výsledkem jakékoli analýzy projektu je jeho výsledná rentabilita. V některém případě jde přímo o minimalizaci nákladů. Jindy slouží jako hlavní kritérium zkrácení doby projektu (obzvláště, pokud je projekt součástí dalšího většího projektu, nebo jen střípkem ve strategickém rozvoji firmy) – zde nám může pomoci optimalizovat zdroje, nebo maximalizovat toky. V praxi jde většinou o stanovení kompromisu mezi cenou a termínem, což nám popsané metody také pomohou řešit.

V praktické části bylo vybráno analyzování síťového grafu metodou kritické cesty. Jsou pomocí ní stanoveny hodnoty, které slouží jako podklad i pro některé další metody optimalizace. Vzhledem k problému poměrně velkého počtu kroků při procházení síťových grafů byl vybrán jednoduchý model s menším počtem uzlů.

Řešení úloh podle zadaných kritérií je z pohledu laika možná trochu těžkopádné a vzhledem k velkému počtu kroků při procházení síťového grafu (a jeho opakovanému procházení) se může zdát i nepřehledné. Tyto komplikace rostou s rozsahem grafu a počtem činností. Při současném nasazení výpočetní techniky v praxi a vzhledem k dobré algoritmizovatelnosti jednotlivých metod ovšem může tento problém být přenesen na stroj. Na zadavateli potom může zůstat „jen“ zadání jednotlivých činností. Slůvko jen musíme brát opravdu v uvozovkách, protože bez řádného zadání nám i ten nejdokonalejší algoritmus dá v praxi nepoužitelné výsledky.

## CONCLUSION

Objective of this thesis was to come up with the list of optimization methods and learn about the methods of network analysis and to describe and evaluate the selected one.

At the beginning we presented an extensive list of methods of optimization divided according to the basic classification. In particular methods were briefly described the principles and procedures with which they work.

We touch briefly on the basic notions from the field of network charts. We have described some of the values of their composition and described the algorithm of enumeration of nodes and searching for cycles.

In the main body of this thesis we have described the particular methods of the optimization of projects represented by the network charts. First we have established the essential terms and activities of this project. Then we focused on the methods of optimization of the required activities from different angles. Because the money always comes first, the real results of any analysis is benefit. In some cases were the cost reduction the main criterion. Other times the time saving in the project comes first of order importance, especially if the project is a part of any larger one, or just a fragment in strategic evolution of the company. There it can help to optimize the resources or maximize the flow. In real life it's mostly the question of finding a compromise between the price and project deadline, which the methods presented help us solve.

In the practice part we have chosen the analysis of the network chart through the method of the critical path. It helps us determine values, which provide a base for the other methods of optimization. In view of the problem of considerably large number of steps in pathing through the network chart, a simple model was chosen with the smaller number of nodes.

Solving of the task according to the established criteria might seem a little cumbersome in regard of the large amount of the steps in pathing through the network charts and the repetitive pathing through might seem complicated. These complications grow with the size of the chart and number of activities. With the present application of information technologies in practical life and with respect to a good algorithmizability of particular methods we can leave this problem to the computer. The submitter of the project will provide "only" the entry of particular activities. We have to consider the word only in quotation mark because without the proper inputs even the best algorithm will return the results that can be used in the practice life.



**SEZNAM POUŽITÉ LITERATURY**

- [1] ZUCHOVICKIJ, S. I. – RADČIKOVÁ, I. A., *Matematické metody síťové analýzy*, 2. nezměněné vydání, Praha: SNTL – Nakladatelství technické literatury, 1973
- [2] MAŇAS, Miroslav, RNDr., *Teorie her a optimální rozhodování*, Praha: SNTL – Nakladatelství technické literatury, 1974
- [3] JUREK, Miloš, Bc, PROKOP, Roman, prof. Ing. (vedoucí), *Optimalizace*, prezentace ve formátu MS PowerPoint

**SEZNAM OBRÁZKŮ**

Obr. 1. Zavedení pomocných fiktivních činností s nulovou délkou trvání k oddělení činností vystupujících z jednoho uzlu a vstupujících společně opět do jednoho uzlu. ....	19
Obr. 2. Zavedení pomocných fiktivních činností s nulovou délkou trvání a uzlů k oddělení činností, vstupujících a vystupujících do/z jednoho uzlu, které na sebe nutně nenavazují.....	20
Obr. 3. Zavedení pomocných fiktivních činností s nulovou délkou trvání pro činnosti, které nemají a) předcházející, b) následující činnost. ....	20
Obr. 4. Zavedení pomocných fiktivních činností s nulovou délkou trvání pro činnosti, které nemají a) předcházející, b) následující činnost. ....	21
Obr. 5. Příklad sestavení agregované sítě – nahrazení dílčí části grafu (nezávislé na ostatních činnostech) jednou činností. ....	21
Obr. 6. Ukázka síťového grafu před přečíslováním. ....	22
Obr. 7. Ukázka přečíslovaného síťového grafu.....	23
Obr. 8. Ukázka grafu připraveného pro výpočet (čísla uzlů v horní části). ....	32
Obr. 9. Graf s vypočtenými hodnotami a kritickou cestou (zvýrazněná). ....	32
Obr. 10. Ukázka lineárního projektu diagramu. ....	34
Obr. 11. Přečíslovaný graf pro praktickou ukázkou.....	58
Obr. 12. Graf se zapsanými dobami trvání jednotlivých činností. ....	59
Obr. 13. Síťový graf se zaznačenými kritickými cestami. ....	62
Obr. 14. Graf s hodnotami pro určování subkritických činností a cest.....	64

**SEZNAM TABULEK**

Tab. 1. Ukázka tabulky pro výpočty hodnot.....	33
Tab. 2. Zápis grafu z obrázku (12) do tabulky .....	59
Tab. 3. Tabulka pro určení kritické posloupnosti činností a volných rezerv (dva díly tabulky pod sebou).....	61
Tab. 4. Přehled časových rezerv pro jednotlivé činnosti.....	63