

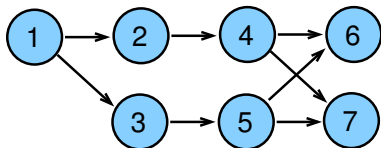
Plánování projektu

28. března 2023

- 1 Úvod
- 2 Repräsentace projektu
- 3 Neomezené zdroje
- 4 Variabilní doba trvání
- 5 Přidání pracovní síly

- Zprostředkování, instalace a testování rozsáhlého počítačového systému
 - projekt zahrnuje
 - evaluace a výběr hardware, vývoj software, nábor a školení lidí, testování a ladění systému, . . .
 - precedenční vztahy
 - některé úlohy mohou být prováděny paralelně
 - úloha musí být realizována až po dokončení jiných úloh
 - cíl: **minimalizovat čas** na realizaci celého projektu
- Příklady dalších problémů
 - stavba nemovitostí, konstrukce center elektráren, vojenský průmysl

- Základní problém **plánování projektu**
 - precedenční podmínky
 - paralelní stroj s **neomezeným počtem strojů**
 - minimalizace maximálního času konce úloh (*makespan*)
 - relativně jednoduché na řešení

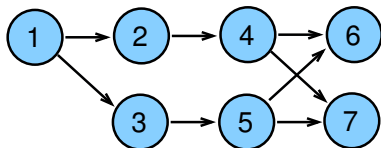


- Rozšíření
 - **variabilní doba trvání** (spojena s cenou provádění)
 - optimalizace: kompromis mezi cenou na ukončení projektu a cenou za zkrácení délky úloh
 - **pracovní síla** (skupiny operátorů s odlišnou specializací)
 - při sdílení omezeného množství operátorů nutno uvažovat **disjunktivní hrany**

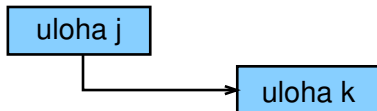
⇒ komplexní problém, jehož řešení je velmi obtížné

- Reprezentace: **úloha jako uzel**
 - hrany reprezentují precedenční podmínky
 - síť neobsahuje žádné orientované cykly

Úloha	Doba trvání	Předchůdci
1	2	–
2	3	1
3	1	1
4	4	2
5	2	3
6	1	4,5
7	3	4,5



- Běžně používána reprezentace úloha jako hrana
- Výhodnější ale úloha jako uzel
 - nejsou nutné redundantní hrany (pomocné úlohy) pro korektní vyjádření precedencí
 - úloha jako uzel lze převést na **úloha jako obdelník**
 - horizontální strany obdelníku použity jako časové osy odpovídající době provádění úlohy

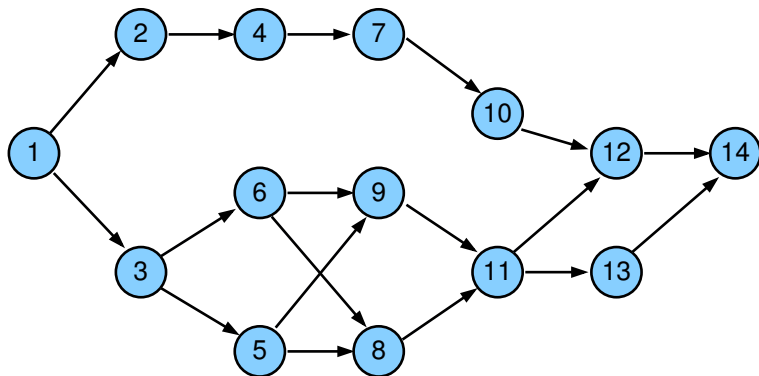


- Popis problému
 - m paralelně zapojených strojů
 - n úloh s precedenčními omezeními
 - doba provádění p_j
 - objektivní funkce: minimalizace maximálního času konce úloh (*makespan*)
- $P_\infty | prec | C_{max}$ (a $m \geq n$) polynomiální složitost, metoda kritické cesty
- $P_m | prec | C_{max}$ $2 \leq m < n$ NP úplný problém
- Značení
 - S'_j nejdřívější startovní čas úlohy j
 - $C'_j = S'_j + p_j$ nejdřívější koncový čas úlohy j
 - S''_j nejpozdější startovní čas úlohy j
 - C''_j nejpozdější koncový čas úlohy j
 - $Prec_j$ (přímí) předchůdci úlohy j
 $\forall k \in Prec_j$ všechny úlohy k , které předcházejí úlohu j
 $\forall j : k \in Prec_j$ všechny úlohy j , které následují úlohu k

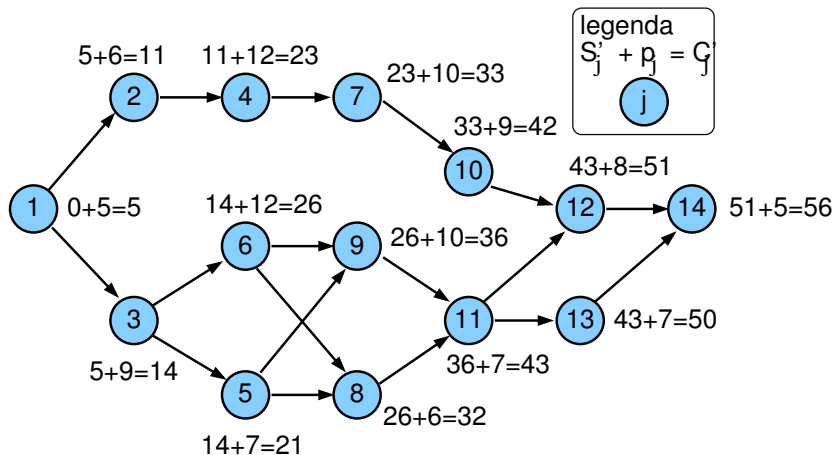
- Popis algoritmu pro nalezení **kritické cesty**
 - **dopředná procedura**
 - start v čase 0
 - výpočet **nejdřívějšího startovního času** každé úlohy
 - čas dokončení poslední úlohy je *makespan*
 - **zpětná procedura**
 - start v čase rovném *makespan*
 - výpočet **nejpozdějšího startovního času**, aby byl realizován tento *makespan*
- **Úloha s rezervou (*slack job*)**
 - její startovní čas může být odložen aniž je navýšen *makespan*
 - úloha, jejichž nejdřívější startovní čas je menší než nejpozdější startovní čas
- **Kritická úloha**
 - úloha, která nesmí být odložena
 - úlohy, jejichž nejdřívější startovní čas je roven nejpozdějšímu startovnímu čas
- **Kritická cesta**
 - řetěz úloh začínající v čase 0 a končící v čase C_{max}
 - v grafu může existovat více kritických cest
kritické cesty se mohou částečně překrývat
 - **graf kritických cest G_{CP}** : podgraf daný množinou kritických úloh a kritických cest

Kritická cesta: zadání příkladu

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p _j	5	6	9	12	7	12	10	6	10	9	7	8	7	5

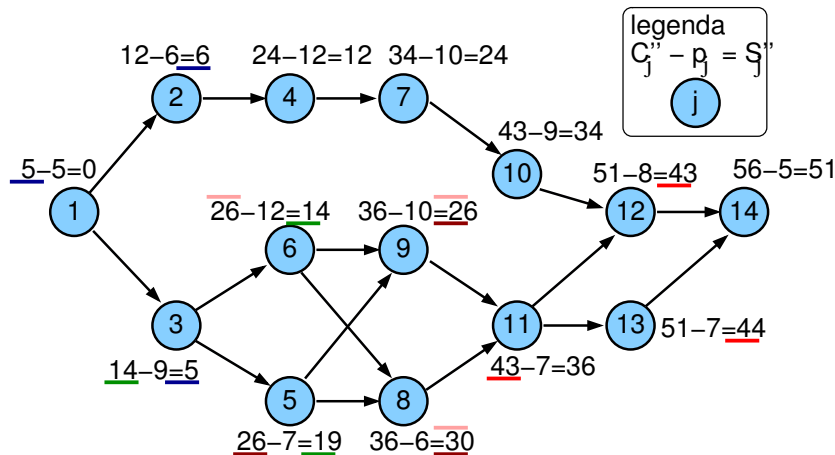


Příklad: dopředná procedura



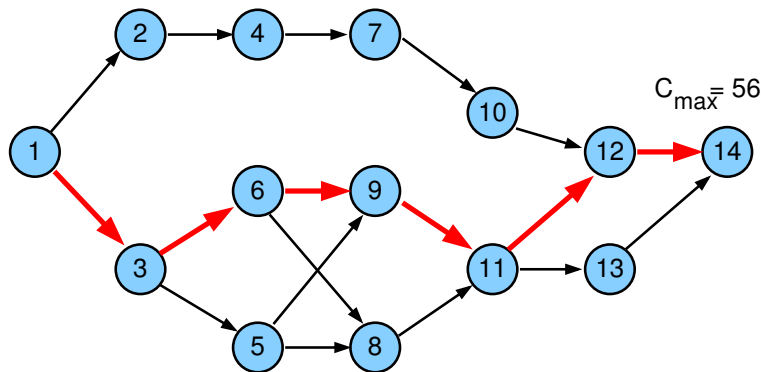
j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p _j	5	6	9	12	7	12	10	6	10	9	7	8	7	5

Příklad: zpětná procedura



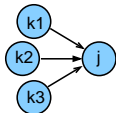
j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p _j	5	6	9	12	7	12	10	6	10	9	7	8	7	5

Kritická cesta



1 Dopředná procedura

- 1 $t = 0$
- 2 pro všechny úlohy j bez předchůdce: $S'_j = 0, C'_j = p_j$
- 3 vypočítej postupně pro všechny zbývající úlohy j :

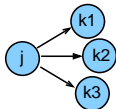


$$S'_j = \max_{\forall k \in \text{Prec}_j} C'_k, \quad C'_j = S'_j + p_j$$

- 4 optimální *makespan* je $C_{max} = \max(C'_1, \dots, C'_n)$

2 Zpětná procedura

- $t = C_{max}$
- pro všechny úlohy j bez následníka: $C''_j = C_{max}, S''_j = C_{max} - p_j$
- vypočítej postupně pro všechny zbývající úlohy j :

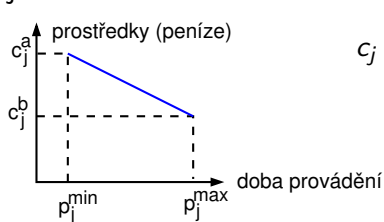


$$C''_j = \min_{\forall k: j \in \text{Prec}_k} S''_k, \quad S''_j = C''_j - p_j$$

- ověř, že $0 = \min(S''_1, \dots, S''_n)$

Kompromis mezi časem a cenou

- Lze uvažovat **variabilní dobu trvání úloh**
 - za předpokladu vyšší ceny lze zkrátit dobu provádění
- **Lineární cena**
- Doba trvání $p_j^{min} \leq p_j \leq p_j^{max}$
- **Marginální cena:** cena za zkrácení doby trvání úlohy o 1 časovou jednotku



$$c_j = \frac{c_j^a - c_j^b}{p_j^{max} - p_j^{min}}$$

- př. $p_j^{min} = 10$, $p_j^{max} = 15$, $c_j^b = 10$, $c_j^a = 20$, $c_j = 2$

⇒ cena provádění úlohy j po dobu p_j : $c_j^b + c_j(p_j^{max} - p_j)$

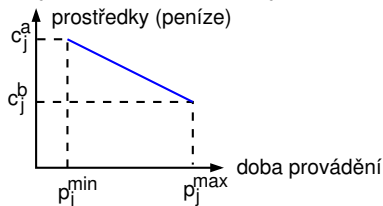
Cena za provádění projektu

- **Fixní režijní náklady c_0**
na časovou jednotku doby provádění projektu
- **Cena $F(p_j)$ za provádění projektu**
 - při době provádění úloh p_jurčena jako součet
 - ceny za provádění všech úloh
 - fixních režijních nákladů

celkem: $C_{\max} c_0$

$$F(p_j) = C_{\max} c_0 + \sum_j (c_j^b + c_j(p_j^{\max} - p_j))$$

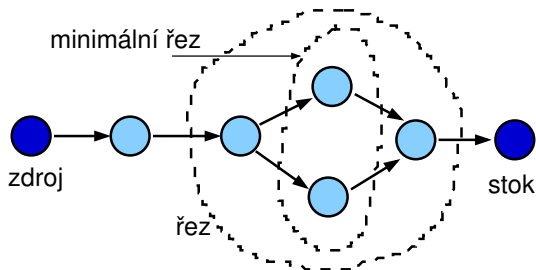
- Cíl: $\forall j$ nalézt p_j a S_j tak, aby byla $F(p_j)$ minimální



- Objektivní funkce: minimální cena projektu
- Kompromisní heuristika mezi časem a cenou
 - dobrá kvalita rozvrhu
 - použité i pro nelineární cenu
- Formulace lineárního programování
 - optimální rozvrh
 - nelineární verze obtížně řešitelné

Opakování: Řez, minimální řez

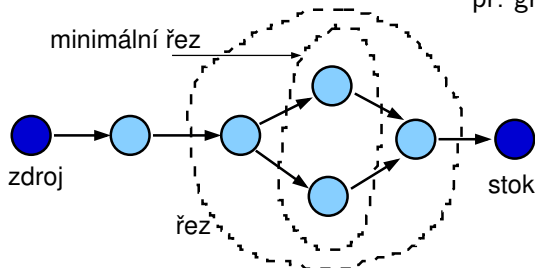
- Orientovaný graf $G = (V, E)$
- Počáteční uzel: zdroj $s \in V$
- Koncový uzel: stok $t \in V$
- **Řez:** ... také mluvíme o vrcholovém řezu
množina uzlů V' , jejíž smazáním z grafu se rozpojí zdroj a stok
 - E' množina hran incidentních s V'tj. v $G'=(V-V',E-E')$ neexistuje orientovaná cesta z s to t
- **Minimální řez:** řez U takový, že neexistuje řez $W \subset U$
 - tj. vrácení libovolného uzlu z U do grafu znovu spojí zdroj a stok



Řez, minimální řez II.

- Uvažujme orientovaný graf $G_0 = (V_0, E_0)$
- Do grafu přidáme zdroj:
 - nový vrchol s a
 - hrany S vedoucí z s do všech vrcholů G_0 bez předchůdců
- Do grafu přidáme stok:
 - nový vrchol t a
 - hrany T vedoucí ze všech vrcholů G_0 bez následníků do t
- Nový graf $G = (V, E)$: $V = V_0 \cup \{s, t\}$, $E = E_0 \cup S \cup T$
- Budeme hledat řezy a minimální řezy z s do t v G

př. graf má 4 minimální řezy

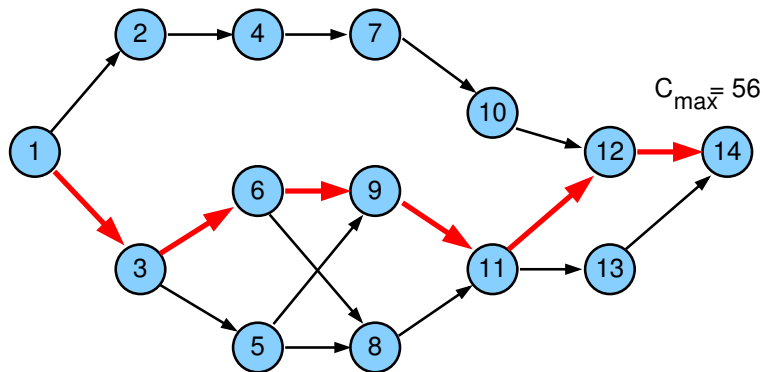


Kompromisní heuristika: příklad

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p_j^{\max}	5	6	9	12	7	12	10	6	10	9	7	8	7	5
p_j^{\min}	3	5	7	9	5	9	8	3	7	5	4	5	5	2
c_j^b	20	25	20	15	30	40	35	25	30	20	25	35	20	10
c_j	7	2	4	3	4	3	4	4	4	5	2	2	4	8

fixní režijní náklady na časovou jednotku $c_0=6$

Příklad (pokračování): maximální doba provádění



Kompromisní heuristika: příklad

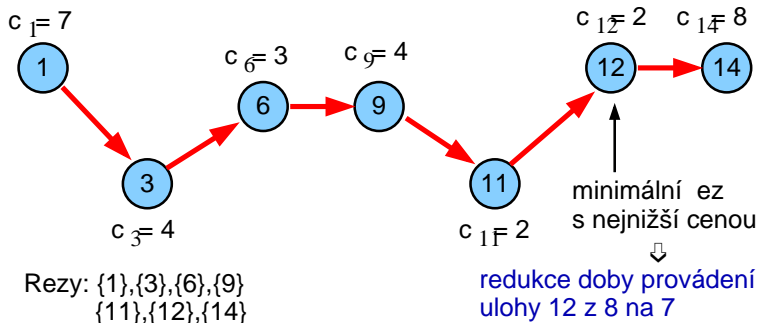
j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p_j^{\max}	5	6	9	12	7	12	10	6	10	9	7	8	7	5
$c_0=6$ p_j^{\min}	3	5	7	9	5	9	8	3	7	5	4	5	5	2
c_j^b	20	25	20	15	30	40	35	25	30	20	25	35	20	10
c_j	7	2	4	3	4	3	4	4	4	5	2	2	4	8

Náklady na provedení projektu při maximální době trvání úloh

$$\begin{aligned}F(p_j^{\max}) &= C_{\max}c_0 + \sum_j (c_j^b + c_j(p_j^{\max} - p_j^{\min})) = \\&= C_{\max}c_0 + \sum_j c_j^b = \\&= 56 \times 6 + 20 + 25 + 20 + 15 + 30 + 40 + 35 + 25 + \\&\quad + 30 + 20 + 25 + 35 + 20 + 10 = \\&= 336 + 350 = 686\end{aligned}$$

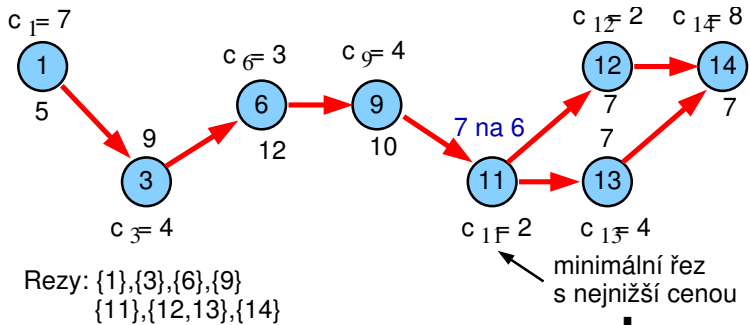
Podgraf s kritickou cestou (G_{CP})

- Kandidáti na redukci: uzel 11 a uzel 12, vybereme uzel 12



- Fixní režijní náklady se redukuje z 56×6 na $55 \times 6 = 330$
- Cena za provádění úloh naroste o $c_{12} = 2$, tj. $350 + 2 = 352$
- Celková cena klesla z 686 na $330 + 352 = 682$

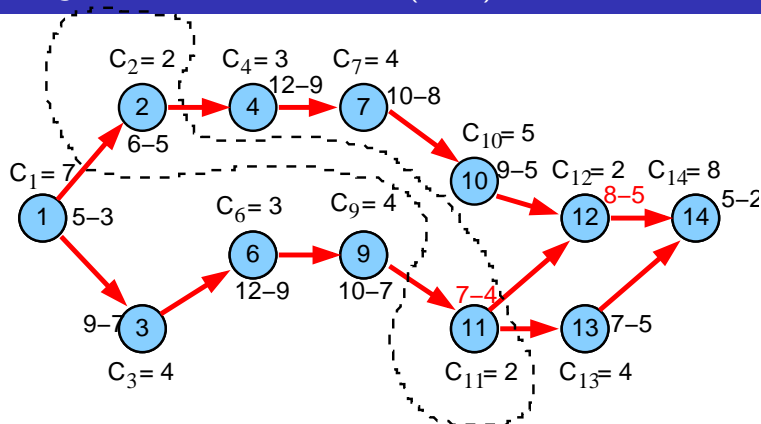
Podgraf s kritickou cestou (G_{CP})



redukce doby trvání úlohy 11 ze 7 na 6

- Fixní režijní náklady se redukovávají z 55×6 na $54 \times 6 = 324$
- Cena za provádění úloh naroste o $c_{11} = 2$, tj. $324 + 2 = 326$
- Celková cena klesla z 682 na $324 + 354 = 678$

Podgraf s kritickou cestou (G_{CP})



další redukce: pro uzel 2 na 5 a pro uzel 11 na 5, ...

- Fixní režijní náklady se redukovávají z 54×6 na $53 \times 6 = 318$
- Cena za provádění úloh naroste o $c_2 + c_{11} = 2 + 2$, tj. $354 + 4 = 358$
- Celková cena klesla z 678 na $318 + 358 = 676$

Algoritmus kompromisní heuristiky

- 1
 - Nastav doby provádění na jejich maximum: $p_j = p_j^{max}$
 - Urči všechny kritické cesty s těmito dobami provádění
 - Zkonstruuuj graf G_{CP} kritických cest
- 2
 - Urči všechny **minimální řezy v G_{CP}**
 - pozn. pokud zkrátíme dobu provádění všech úloh v minimálním řezu, podaří se nám i zkrátit dobu provádění projektu!
 - Najdi **řezy, jejichž doba provádění je větší než jejich minimum:**
 $p_j > p_j^{min} \quad \forall j \in G_{CP}$
 - Pokud takový řez neexistuje STOP, jinak běž na krok 3
- 3
 - Pro každý minimální řez: spočítej cenu redukující všechny doby provádění o 1 časovou jednotku
 - Vyber **minimální řez s nejnižší cenou**
 - pozn. abychom za snížení zaplatili co nejnižší cenu
 - Jestliže je cena řezu menší než fixní režijní náklady c_0 za časovou jednotku běž na krok 4, jinak STOP
- 4
 - Redukuj všechny doby provádění v minimálním řezu o 1 časovou jednotku
 - Urči novou množinu kritických cest
 - Reviduj graf G_{CP} a běž na krok 2

Lineární program

- Heuristika negarantuje nalezení optima
- Celková cena je lineární

$$c_0 C_{max} + \sum_{j=1}^n \left(c_j^b + c_j (p_j^{max} - p_j) \right)$$

všimněte si: stejná účelová funkce jako cena za provádění projektu

- Lineární program:

minimalizace:

$$c_0 C_{max} - \sum_{j=1}^n c_j p_j$$

c_j^b a $c_j p_j^{max}$

se nemění

za předpokladu:

$$\begin{aligned} x_k - p_j - x_j &\geq 0 && \forall j \in Prec_k \\ p_j &\leq p_j^{max} && \forall j \\ p_j &\geq p_j^{min} && \forall j \\ x_j &\geq 0 && \forall j \\ C_{max} - x_j - p_j &\geq 0 && \forall j \end{aligned}$$

kde x_j je nejdřívější možný startovní čas úlohy j

- Pracovní síla = operátor = zdroj
- Problém popsán v literatuře jako
problém plánování projektu s omezenými zdroji
resource-constrained project scheduling problem (RCPSP)
 - n úloh
 - N zdrojů
 - R_i kapacita zdroje i
 - p_j doba provádění úlohy j
 - R_{ij} požadavek úlohy j na zdroj i
 - $Prec_j$ (přímí) předchůdci úlohy j

Formulace celočíselného programování

- Pomocná úloha $n + 1$ jako stok, $p_{n+1} = 0$
- $x_{jt} = 1$ úloha j je dokončena v čase t
 $x_{jt} = 0$ jinak
- Kapacita zdroje i , který potřebuje úloha j

v intervalu $[t - 1, t]$:

$$R_{ij} \sum_{u=t}^{t+p_j-1} x_{ju}$$

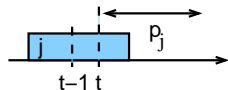
$\sum_{u=t}^{t+p_j-1} x_{ju}$... počítá, zda úloha j běží v čase $[t - 1, t]$

př. úloha s $S_j = 2, p_j = 2$ a $t = 2, 3, 4, 5$ ($x_{j4} = 1$, pro $t = 3, 4$ úloha započítána)

- H jako horní hranice *makespan*:
$$H = \sum_{j=1}^n p_j$$

- Koncový čas úlohy j :
$$C_j = \sum_{t=1}^H t x_{jt}$$

- *Makespan*:
$$C_{max} = \sum_{t=1}^H t x_{n+1,t}$$
 koncový čas stoku



Minimalizace $\sum_{t=1}^H t x_{n+1,t}$ minimalizace *makespan*

za předpokladu

jestliže j je předchůdce k , pak $C_j \leq S_k$, tj. $C_j \leq C_k - p_k$, tj. $C_j + p_k - C_k \leq 0$

$$\sum_{t=1}^H t x_{jt} + p_k - \sum_{t=1}^H t x_{kt} \leq 0 \quad \forall j \in \text{Prec}_k$$

pro každý čas t : požadavek na zdroj i nepřeroste kapacitu i

$$\sum_{j=1}^n \left(R_{ij} \sum_{u=t}^{t+p_j-1} x_{ju} \right) \leq R_i \quad \forall i \forall t$$

každá úloha j skončí právě jednou

$$\sum_{t=1}^H x_{jt} = 1 \quad \forall j$$

- Řešení celočíselného programu obtížné
- Při velkém počtu úloh a dlouhém časovém horizontu
 - použití heuristik
- Lze použít programování s omezujícími podmínkami
 - kumulativní zdroje
 - precedenční podmínky
- Probírané speciální případy problému
 - *job shop + makespan*
 - *timetabling*

- Základní problém s neomezenými zdroji
 - metoda kritické cesty
- Neomezené zdroje + variabilní doba trvání (lineární)
 - kompromisní heuristika mezi časem a cenou
 - lineární programování
- Problém plánování projektu s omezenými zdroji
 - celočíselné programování
 - heuristiky
 - programování s omezujícími podmínkami