Řešení vybraných příkladů ze sbírky k předmětu PA167 Rozvrhování

http://www.fi.muni.cz/~hanka/rozvrhovani

Hana Rudová

Fakulta informatiky, Masarykova univerzita

19. února 2018

Poděkování

Řešení příkladů jsou převzata z domacích úkolů studentů předmětu. Všem děkuji za pěkná řešení, která jsou zde použita.

Příklad 5.4

Pravidlo minimální rezervy – jsme v určitém čase t (například na začátku, kdy $t = 0$) a potřebujeme rozhodnout, kterou úlohu z dosud neprovedených úloh (tzn. na začátku úplně ze všech) máme spustit. Pro každou neprovedenou úlohu (tzn. na začátku úplně pro každou) se vypočítá rezerva $A_j = \max(d_j - p_j - t, 0)$. Je spuštěna ta úloha, jejíž rezerva je nejmenší.

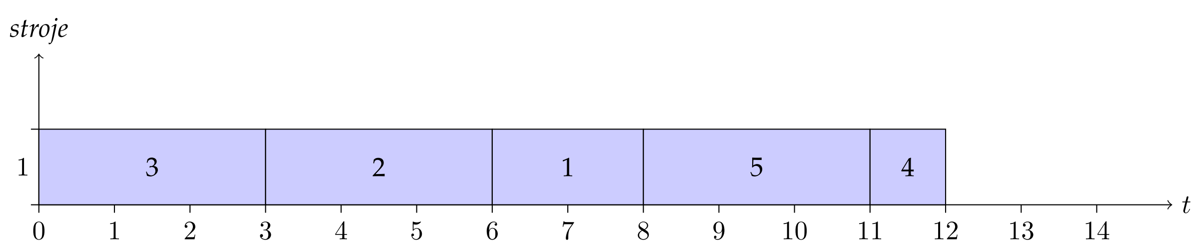
$\boldsymbol{t = 0}$
$A_1 = \max(7 - 2 - 0, 0) = 5$
$A_2 = \max(7 - 3 - 0, 0) = 4$
$A_3 = \max(4 - 3 - 0, 0) = 1$
$A_4 = \max(11 - 1 - 0, 0) = 10$
$A_5 = \max(12 - 3 - 0, 0) = 9$
Úloha 3 poběží v čase (0, 3).

$\boldsymbol{t = 3}$
$A_1 = \max(7 - 2 - 3, 0) = 2$
$A_2 = \max(7 - 3 - 3, 0) = 1$
$A_4 = \max(11 - 1 - 3, 0) = 7$
$A_5 = \max(12 - 3 - 3, 0) = 6$
Úloha 2 poběží v čase (3, 6).

$\boldsymbol{t = 6}$
$A_1 = \max(7 - 2 - 6, 0) = 0$
$A_4 = \max(11 - 1 - 6, 0) = 4$
$A_5 = \max(12 - 3 - 6, 0) = 3$
Úloha 1 poběží v čase (6, 8).

$\boldsymbol{t = 8}$
$A_4 = \max(11 - 1 - 8, 0) = 2$
$A_5 = \max(12 - 3 - 8, 0) = 1$
Úloha 5 poběží v čase (8, 11) a úloha 4 poběží v čase (11, 12).

Celkový rozvrh je znázorněn následujícím Ganttovým diagramem:



$\boldsymbol{L_{max}} = \max(L_1, L_2, L_3, L_4, L_5) = \max(C_1 - d_1, C_2 - d_2, C_3 - d_3, C_4 - d_4, C_5 - d_5) =$
$= \max(8 - 7, 6 - 7, 3 - 4, 12 - 11, 11 - 12) = \max(1, -1, -1, 1, -1) = 1$

$\sum_{\boldsymbol{j}} \boldsymbol{T_j} = T_1 + T_2 + T_3 + T_4 + T_5 =$
$= \max(C_1 - d_1, 0) + \max(C_2 - d_2, 0) + \max(C_3 - d_3, 0) + \max(C_4 - d_4, 0) + \max(C_5 - d_5, 0) =$
$= \max(8 - 7, 0) + \max(6 - 7, 0) + \max(3 - 4, 0) + \max(12 - 11, 0) + \max(11 - 12, 0) =$
$= \max(1, 0) + \max(-1, 0) + \max(-1, 0) + \max(1, 0) + max(-1, 0) = 1 + 0 + 0 + 1 + 0 = 2$

Příklad 5.5

(1) Minimum Slack

čas	dostupné úlohy	výběr
0	úloha 4 s rezervou 11	4
1	–	–
2	úloha 1 s rezervou 0 úloha 3 s rezervou 1	1
7	úloha 3 s rezervou 0 úloha 2 s rezervou 1 úloha 5 s rezervou 0	3
8	úloha 2 s rezervou 0 úloha 5 s rezervou 0	2
11	úloha 5 s rezervou 0	5

$L_{\max} = \max(0, 0, 4, -11, 6) = 6$
$\sum_j w_j T_j = 1 \cdot 4 + 1 \cdot 6 = 10$

(2) Earliest Due Date

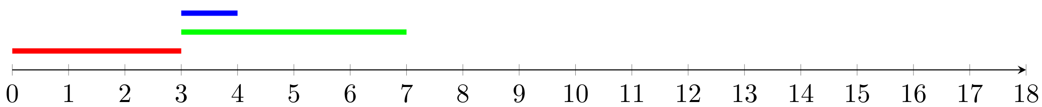
čas	dostupné úlohy	výběr
0	úloha 4	4
1	–	–
2	úloha 1 úloha 3	3
3	úloha 1 úloha 2 úloha 5	1
8	úloha 2 úloha 5	5
11	úloha 2	2

$L_{\max} = \max(1, 3, -1, -11, 3) = 3$
$\sum_j w_j T_j = 2 \cdot 1 + 1 \cdot 3 + 1 \cdot 3 = 8$

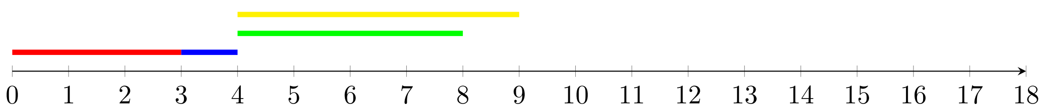
Příklad 5.9

Pravidlo *WSPT* vybírá úlohy podle doby trvání s váhami. Budeme úlohy řadit podle vzrůstajícího p_j/w_j, protože chceme nejdříve brát úlohy s kratším časem a větší váhou. Minimalizuje tedy vážený součet časů konců úloh.

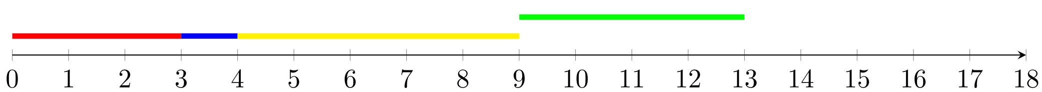
V čase 0 máme jedinou úlohu 4. Proto ji můžeme vybrat. Doběhne v čase 3 a musíme vybrat další úlohu.



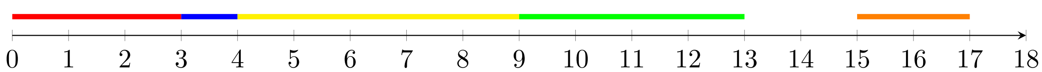
Vybíráme z úloh 3 a 5, jejichž vážené doby trvání jsou $1/2$ a $4/1$, proto vybereme úlohu 3.



Znovu vybíráme v čase 4, tentokrát z úloh 2 a 5. Jejich vážené doby trvání jsou $5/3$ a $4/1$, proto vybíráme úlohu 2.



V čase 9 musíme vybrat další úlohu. Máme jedinou možnost: úlohu 5. Po jejím doběhnutí v čase 13 nemáme žádnou další dostupnou úlohu. Musíme tedy počkat do času 15, kdy přijde úloha 1, kterou můžeme hned spustit.



Výsledný rozvrh tedy bude (4, 3, 2, 5, 1) s celkovou hodnotou účelové funkce
$5 \cdot 3 + 2 \cdot 4 + 3 \cdot 9 + 1 \cdot 13 + 3 \cdot 17 = 114$.

Kdybychom neměli termíny dostupnosti, mohli bychom na začátku naplánovat celý rozvrh a najít i optimální řešení. Pro úlohu 1 bychom navíc nemuseli čekat na její zadání a zbavili bychom se tedy intervalu, kdy stroj nic nedělá. Vážené doby trvání pro úlohy 1–5 by byly postupně $2/3$, $5/3$, $1/2$, $3/5$, $4/1$.

Vytvořený rozvrh tedy bude $(3, 4, 1, 2, 5)$ s hodnotou účelové funkce
$2 \cdot 1 + 5 \cdot 4 + 3 \cdot 6 + 3 \cdot 11 + 1 \cdot 15 = 88$.

Příklad 6.4

Vyjádření problému $1|r_j, p_j = 1|\sum w_j C_j$ pomocí celočíselného programování:

Proměnné: x_{jt} kde $x_{jt} = 1$ pokud úloha j začne v čase t, jinak $x_{jt} = 0$.

Optimalizační funkce: (minimalizovat)

$$\sum_{j=1}^{n} \sum_{t=0}^{H} w_j (t+1) x_{jt}$$

kde n je počet úloh.
Jako poslední možný startovní čas uvažujeme $H = \max(r_1, r_2, .., r_n) + n$, tím se kryjeme proti případu, kdy všechny úlohy jsou dostupné ve stejném čase, který je větší než n počet úloh. Dále budeme používat H s tímto významem.

Předpoklady:
Omezení z definice proměnné platí.

$$\forall j \in \{0, .., n\}, \forall t \in \{0, .., H\} : x_{jt} \in \{0, 1\}$$

Úloha nezačne před svým termínem dostupnosti r_j.

$$\forall j \in \{0, .., n\} : \sum_{t=0}^{r_j - 1} x_{jt} = 0$$

Úloha začne někdy po té co se stane dostupnou.

$$\forall j \in \{0, .., n\} : \sum_{t=r_j}^{H} x_{jt} = 1$$

V každém čase běží nejvýše jedna úloha (v rozvrhu ale mohou vzniknout mezery kvůli termínům dostupnosti). Suma přes všechny úlohy stačí, díky jednotkovému trvání úloh (tj. nemohou se překrývat, pokud nezačínají ve stejném čase. Překrývání by tedy způsobilo, že tato suma bude větší než 1).

$$\forall t \in \{0, .., H\} : \sum_{i=0}^{n} x_{it} \leq 1$$

Příklad 8.4

1. iterace

$S_1 = (1, 2, 3, 4)$
$C_1 = 1, C_2 = 3, C_3 = 6, C_4 = 7$
$T_1 = 0, T_2 = 2, T_3 = 4, T_4 = 0$
$F(S_1) = \sum w_j T_j = 1 \cdot 0 + 2 \cdot 2 + 5 \cdot 4 + 3 \cdot 0 = 24 = F(S_{best})$
$N(S_1) = \{(2, 1, 3, 4), (1, 3, 2, 4), (1, 2, 4, 3)\}$

$\boldsymbol{F(2, 1, 3, 4)}$
$C_2 = 2, C_1 = 3, C_3 = 6, C_4 = 7$
$T_2 = 1, T_1 = 1, T_3 = 4, T_4 = 0$
$F(2, 1, 3, 4) = 2 \cdot 1 + 1 \cdot 1 + 5 \cdot 4 + 3 \cdot 0 = 23$

$\boldsymbol{F(1, 3, 2, 4)}$
$C_1 = 1, C_3 = 4, C_2 = 6, C_4 = 7$
$T_1 = 0, T_3 = 2, T_2 = 5, T_4 = 0$
$F(1, 3, 2, 4) = 1 \cdot 0 + 5 \cdot 2 + 2 \cdot 5 + 3 \cdot 0 = 20$

$\boldsymbol{F(1, 2, 4, 3)}$
$C_1 = 1, C_2 = 3, C_4 = 4, C_3 = 7$
$T_1 = 0, T_2 = 2, T_4 = 0, T_3 = 5$
$F(1, 2, 4, 3) = 1 \cdot 0 + 2 \cdot 2 + 3 \cdot 0 + 5 \cdot 5 = 29$

Nejlepší rozvrh z okolí: $(1, 3, 2, 4)$; $F(1, 3, 2, 4) = 20 = F(S_{best})$
Tabu seznam: $\langle(2, 3)\rangle$

2. iterace

$S_2 = (1, 3, 2, 4)$
$N(S_2) = (3, 1, 2, 4), (1, 2, 3, 4), (1, 3, 4, 2)$

$\boldsymbol{F(3, 1, 2, 4)}$
$C_3 = 3, C_1 = 4, C_2 = 6, C_4 = 7$
$T_3 = 1, T_1 = 2, T_2 = 5, T_4 = 0$
$F(3, 1, 2, 4) = 5 \cdot 1 + 1 \cdot 2 + 2 \cdot 5 + 3 \cdot 0 = 17$

$\boldsymbol{F(1, 2, 3, 4)}$
TABU

$\boldsymbol{F(1, 3, 4, 2)}$
$C_1 = 1, C_3 = 4, C_4 = 5, C_2 = 7$
$T_1 = 0, T_3 = 2, T_4 = 0, T_2 = 6$
$F(1, 3, 2, 4) = 1 \cdot 0 + 5 \cdot 2 + 3 \cdot 0 + 2 \cdot 6 = 22$

Nejlepší rozvrh z okolí: $(3, 1, 2, 4)$; $F(3, 1, 2, 4) = 17 = F(S_{best})$
Tabu seznam: $\langle(1, 3), (2, 3)\rangle$

Příklad 9.5

$S_{best} = S_1 = (3, 1, 4, 2)$
$F(S_1) = \sum w_j T_j = 1 \cdot 7 + 14 \cdot 11 + 12 \cdot 0 + 12 \cdot 25 = 461 = F(S_{best})$
$t_0 = 0.99$

$N(S_1) = \{(1, 3, 4, 2), (3, 4, 1, 2), (3, 1, 2, 4)\}$
$F((1, 3, 4, 2)) = 14 \cdot 0 + 1 \cdot 16 + 12 \cdot 0 + 12 \cdot 25 = 316$
$F((3, 4, 1, 2)) = 1 \cdot 7 + 12 \cdot 0 + 14 \cdot 14 + 12 \cdot 25 = 503$
$F((3, 1, 2, 4)) = 1 \cdot 7 + 14 \cdot 11 + 12 \cdot 22 + 12 \cdot 5 = 485$

$S_{new} = (1, 3, 4, 2)$
$F(S_{new}) = 316 < F(S_{best})$
$S_{best} = S_2 = (1, 3, 4, 2)$
$t = 0.99 \cdot 0.9 = 0.891$

$N(S_2) = \{(3, 1, 4, 2), (1, 4, 3, 2), (1, 3, 2, 4)\}$
$F((3, 1, 4, 2)) = 461$
$F((1, 4, 3, 2)) = 14 \cdot 0 + 12 \cdot 0 + 1 \cdot 19 + 12 \cdot 25 = 319$
$F((1, 3, 2, 4)) = 0 + 16 + 12 \cdot 22 + 12 \cdot 5 = 340$

$S_{new} = (1, 4, 3, 2)$
$F(S_{new}) > F(S_2)$
$U_2 = 0.02 < e^{(316-319)/0.891}$
$S_3 = (1, 4, 3, 2)$
$t = 0.891 \cdot 0.9 = 0.8019$

$N(S_3) = \{(4, 1, 3, 2), (1, 3, 4, 2), (1, 4, 2, 3)\}$
$F((4, 1, 3, 2)) = 12 \cdot 0 + 14 \cdot 2 + 1 \cdot 19 + 12 \cdot 25 = 347$
$F((1, 3, 4, 2)) = 316$
$F((1, 4, 2, 3)) = 0 + 0 + 12 \cdot 13 + 28 = 184$

$S_{new} = (1, 4, 2, 3)$
$F(S_{new}) = 184 < F(S_{best})$
$S_{best} = S_4 = (1, 4, 2, 3)$
$t = 0.72171$

Příklad 10.12

Modře zbarvený potomek bude nahrazovat zeleně zbarveného jedince v populaci.

0. populace:

<table>
<tr><td>populace</td><td colspan="3">3421</td><td colspan="3">2314</td><td colspan="3">4231</td></tr>
<tr><td>vhodnost</td><td colspan="3">17</td><td colspan="3">15</td><td colspan="3">18</td></tr>
<tr><td>potomci</td><td>4321</td><td>3241</td><td>3412</td><td>3214</td><td>2134</td><td>2341</td><td>2431</td><td>4321</td><td>4213</td></tr>
<tr><td>vhodnost</td><td>17</td><td>15</td><td>14</td><td>14</td><td>13</td><td>16</td><td>18</td><td>17</td><td>16</td></tr>
</table>

1. populace:

<table>
<tr><td>populace</td><td colspan="3">3421</td><td colspan="3">2314</td><td colspan="3">2134</td></tr>
<tr><td>vhodnost</td><td colspan="3">17</td><td colspan="3">15</td><td colspan="3">13</td></tr>
<tr><td>potomci</td><td>4321</td><td>3241</td><td>3412</td><td>3214</td><td>2134</td><td>2341</td><td>1234</td><td>2314</td><td>2143</td></tr>
<tr><td>vhodnost</td><td>17</td><td>15</td><td>14</td><td>14</td><td>13</td><td>16</td><td>9</td><td>15</td><td>14</td></tr>
</table>

2. populace:

populace	1234	2314	2134
vhodnost	9	15	13

Rozvrh s nejlepší vhodností po dvou iteracích: 1234.

Příklad 11.8

Iniciální domény: $A \in \{1..4\},\ B \in \{1..4\},\ C \in \{1..4\}$
Iniciální fronta: $Q = \{A, B, C\}$

1. výběr A z fronty
   (a) revise(c_1): beze změn
   (b) revise(c_3) $\rightarrow A \in \{1..3\},\ B \in \{1..3\}$

   $Q = \{B, C\} \cup \{A\} = \{B, C, A\}$ (B už ve frontě je, nepřidáváme)
2. výběr B z fronty
   (a) revise(c_1): beze změn
   (b) revise(c_2): beze změn
   (c) revise(c_3): beze změn
   (d) revise(c_4) $\rightarrow B \in \{1..2\},\ C \in \{1..2\}$

   $Q = \{C, A\} \cup \{B\} = \{C, A, B\}$ (C už ve frontě je, nepřidáváme)
3. výběr C z fronty
   (a) revise(c_2): beze změn
   (b) revise(c_4): beze změn

   $Q = \{A, B\}$
4. výběr A z fronty
   (a) revise(c_1): beze změn
   (b) revise(c_3) $\rightarrow A \in \{2..3\}$

   $Q = \{B\} \cup \{A\} = \{B, A\}$ (B už ve frontě je, nepřidáváme)
5. výběr B z fronty
   (a) revise(c_1): beze změn
   (b) revise(c_2): beze změn
   (c) revise(c_3): beze změn
   (d) revise(c_4): beze změn

   $Q = \{A\}$
6. výběr A z fronty
   (a) revise(c_1): beze změn
   (b) revise(c_3): beze změn

   $Q = \{\}$

Finální domény: $A \in \{2, 3\},\ B \in \{1, 2\},\ C \in \{1, 2\}$

Příklad 11.13

Algoritmus MAC postupně přiřazuje proměnným hodnoty. Při každém přiřazení kontroluje, jestli jsou proměnné konzistentní. Pokud ne, zkusí jinou hodnotu. Když některé proměnné dojdou hodnoty, algoritmus oznámí neexistenci řešení.

Nejprve provedeme konzistenční algoritmus. Podmínky 1, 2, 4 jsou konzistentní. Pro podmínku 3 ale hodnota 0 proměnné B nemá podporu a proto ji můžeme odstranit.

Nové domény tedy jsou A in $0 \ldots 1$, B in $1 \ldots 3$, C in $0 \ldots 2$.

Nyní můžeme spustit vlastní prohledávací algoritmus. První krok: vybereme proměnnou. A má dvě možné hodnoty, B a C po třech hodnotách. Proto vybereme A. Nyní musíme vybrat některou její hodnotu.

Pro hodnotu 0 máme $3+2+3+2=10$ podpor (postupně pro jednotlivé podmínky $\{1,2,3\}$, $\{1,2\}$, $\{1,2,3\}$, $\{(2,2),(3,1)\}$). Pro hodnotu 1 máme $2+2+2+3=9$ podpor ($\{2,3\}$, $\{0,2\}$, $\{2,3\}$, $\{(1,2),(2,1),(3,0)\}$). Vezmeme hodnotu 0 jako první a přiřadíme $A=0$. Nyní musíme zrevidovat domény ostatních proměnných. První a třetí podmínka neodstraní žádné hodnoty. Druhá podmínka odstraní hodnotu 0 proměnné C. Poslední podmínka odstraní hodnotu 1 z domény proměnné B.

Nové domény tedy jsou $A=0$, B in $2 \ldots 3$, C in $1 \ldots 2$.

Vybereme další proměnnou. Obě zbývající proměnné mají po dvou hodnotách, zvolíme tedy například proměnnou B. Její hodnoty mají po jedné podpoře v doméně C. Jako první tedy zkusíme například hodnotu 2 a budeme propagovat omezení. Čtvrté omezení nám odstraní hodnotu 1 z domény C. Tím se doména poslední proměnné zmenšila na jedinou hodnotu a dostali jsme řešení.

Výsledek tedy je $A=0$, $B=2$, $C=2$.

Příklad 13.9

Nejprve určíme domény proměnných $start(A)$ a $end(A)$ a dobu provádění $p(A)$ pro všechny úlohy z Ω:

$start(X) \in \langle 0, 7 \rangle$	$start(Y) \in \langle 2, 7 \rangle$	$start(Z) \in \langle 3, 8 \rangle$	$start(V) \in \langle 1, 7 \rangle$
$end(X) \in \langle 3, 10 \rangle$	$end(Y) \in \langle 5, 10 \rangle$	$end(Z) \in \langle 7, 12 \rangle$	$end(V) \in \langle 3, 9 \rangle$
$p(X) = 3$	$p(Y) = 3$	$p(Z) = 4$	$p(V) = 2$

Dále zkoušíme aplikovat odvozovací pravidla pro každou kombinaci $(A, \Omega - \{A\})$.

Hledání hran

$\boldsymbol{(X, \{Y, Z, V\}), end(X)}$
$lct(\{Y, Z, V\} \cup \{X\}) - est(\{Y, Z, V\}) = 12 - 1 = 11$
$p(\{Y, Z, V\} \cup \{X\}) = 12$
$11 < 12 \Rightarrow X << \{Y, Z, V\} \Rightarrow$
$\Rightarrow end(X) \leq \min(lct(\{Y, Z, V\}) - p(\{Y, Z, V\}), lct(\{Y, Z\}) - p(\{Y, Z\}), lct(\{Z, V\}) - p(\{Z, V\}), lct(\{Y, V\}) - p(\{Y, V\}), lct(\{Y\}) - p(\{Y\}), lct(\{Z\}) - p(\{Z\}), lct(\{V\}) - p(\{V\})) \Rightarrow$
$\Rightarrow end(X) \leq \min(12 - 9, 12 - 7, 12 - 6, 10 - 5, 10 - 3, 12 - 4, 9 - 2) \Rightarrow$
$\Rightarrow end(X) \leq \min(3, 5, 6, 5, 7, 8, 7) \Rightarrow$
$\Rightarrow end(X) \leq 3 \Rightarrow end(X) \in \{3\}$

$\boldsymbol{(X, \{Y, Z, V\}), start(X)}$
$lct(\{Y, Z, V\}) - est(\{Y, Z, V\} \cup \{X\}) = 12 - 0 = 12$
$p(\{Y, Z, V\} \cup \{X\}) = 12$
$12 \geq 12 \Rightarrow$ žádná změna

$\boldsymbol{(Y, \{X, Z, V\}), end(Y)}$
$lct(\{X, Z, V\} \cup \{Y\}) - est(\{X, Z, V\}) = 12 - 0 = 12$
$p(\{X, Z, V\} \cup \{Y\}) = 12$
$12 \geq 12 \Rightarrow$ žádná změna

$\boldsymbol{(Y, \{X, Z, V\}), start(Y)}$
$lct(\{X, Z, V\}) - est(\{X, Z, V\} \cup \{Y\}) = 12 - 0 = 12$
$p(\{X, Z, V\} \cup \{Y\}) = 12$
$12 \geq 12 \Rightarrow$ žádná změna

$\boldsymbol{(Z, \{X, Y, V\}), end(Z)}$
$lct(\{X, Y, V\} \cup \{Z\}) - est(\{X, Y, V\}) = 12 - 0 = 12$
$p(\{X, Y, V\} \cup \{Z\}) = 12$
$12 \geq 12 \Rightarrow$ žádná změna

$\boldsymbol{(Z, \{X, Y, V\}), start(Z)}$
$lct(\{X, Y, V\}) - est(\{X, Y, V\} \cup \{Z\}) = 10 - 0 = 10$
$p(\{X, Y, V\} \cup \{Z\}) = 12$

$10 < 12 \Rightarrow \{X, Y, V\} << Z \Rightarrow$
$\Rightarrow start(Z) \geq \max(est(\{X,Y,V\})+p(\{X,Y,V\}), est(\{X,Y\})+p(\{X,Y\}), est(\{Y,V\})+p(\{Y,V\}), est(\{X,V\})+p(\{X,V\}), est(\{X\})+p(\{X\}), est(\{Y\})+p(\{Y\}), est(\{V\})+p(\{V\})) \Rightarrow$
$\Rightarrow start(Z) \geq \max(0+8, 0+6, 1+5, 0+5, 0+3, 2+3, 1+2) \Rightarrow$
$\Rightarrow start(Z) \geq \max(8, 6, 6, 5, 3, 5, 3) \Rightarrow$
$\Rightarrow start(Z) \geq 8 \Rightarrow start(Z) \in \{8\}$

$\boldsymbol{(V, \{X, Y, Z\}), end(V)}$
$lct(\{X,Y,Z\} \cup \{V\}) - est(\{X,Y,Z\}) = 12 - 0 = 12$
$p(\{X,Y,Z\} \cup \{V\}) = 12$
$12 \geq 12 \Rightarrow$ žádná změna

$\boldsymbol{(V, \{X, Y, Z\}), start(V)}$
$lct(\{X,Y,Z\}) - est(\{X,Y,Z\} \cup \{V\}) = 12$
$p(\{X,Y,Z\} \cup \{V\}) = 12$
$12 \geq 12 \Rightarrow$ žádná změna

Ne-první / ne-poslední

$\boldsymbol{(X, \{Y, Z, V\}), start(X)}$
$p(\{Y,Z,V\} \cup \{X\}) = 12$
$lct(\{Y,Z,V\}) - est(X) = 12 - 0 = 12$
$12 \leq 12 \Rightarrow$ žádná změna

$\boldsymbol{(X, \{Y, Z, V\}), end(X)}$
$p(\{Y,Z,V\} \cup \{X\}) = 12$
$lct(X) - est(\{Y,Z,V\}) = 10 - 1 = 9$
$12 > 9 \Rightarrow \neg\{Y,Z,V\} << X \Rightarrow$
$\Rightarrow end(X) \leq \max(lst(Y), lst(Z), lst(V)) \Rightarrow$
$\Rightarrow end(X) \leq \max(7, 8, 7) \Rightarrow$
$\Rightarrow end(X) \leq 8 \Rightarrow end(X) \in \langle 3, 8 \rangle$

$\boldsymbol{(Y, \{X, Z, V\}), start(Y)}$
$p(\{X,Z,V\} \cup \{Y\}) = 12$
$lct(\{X,Z,V\}) - est(Y) = 12 - 2 = 10$
$12 > 10 \Rightarrow \neg Y << \{X,Z,V\} \Rightarrow$
$\Rightarrow start(Y) \geq \min(ect(X), ect(Z), ect(V)) \Rightarrow$
$\Rightarrow start(Y) \geq \min(3, 7, 3) \Rightarrow$
$\Rightarrow start(Y) \geq 3 \Rightarrow start(Y) \in \langle 3, 7 \rangle$

$\boldsymbol{(Y, \{X, Z, V\}), end(Y)}$
$p(\{X,Z,V\} \cup \{Y\}) = 12$
$lct(Y) - est(\{X,Z,V\}) = 10 - 0 = 10$
$12 > 10 \Rightarrow \neg\{X,Z,V\} << Y \Rightarrow$
$\Rightarrow end(Y) \leq \max(lst(X), lst(Z), lst(V)) \Rightarrow$
$\Rightarrow end(Y) \leq \max(7, 8, 7) \Rightarrow$
$\Rightarrow end(Y) \leq 8 \Rightarrow end(Y) \in \langle 5, 8 \rangle$

$\boldsymbol{(Z, \{X, Y, V\}), start(Z)}$
$p(\{X, Y, V\} \cup \{Z\}) = 12$
$lct(\{X, Y, V\}) - est(Z) = 10 - 3 = 7$
$12 > 7 \Rightarrow \neg Z << \{X, Y, V\} \Rightarrow$
$\Rightarrow start(Z) \geq \min(ect(X), ect(Y), ect(V)) \Rightarrow$
$\Rightarrow start(Z) \geq \min(3, 5, 3) \Rightarrow$
$\Rightarrow start(Z) \geq 3 \Rightarrow start(Z) \in \langle 3, 8 \rangle$ (ve skutečnosti žádná změna)

$\boldsymbol{(Z, \{X, Y, V\}), end(Z)}$
$p(\{X, Y, V\} \cup \{Z\}) = 12$
$lct(Z) - est(\{X, Y, V\}) = 12 - 0 = 12$
$12 \leq 12 \Rightarrow$ žádná změna

$\boldsymbol{(V, \{X, Y, Z\}), start(V)}$
$p(\{X, Y, Z\} \cup \{V\}) = 12$
$lct(\{X, Y, Z\}) - est(V) = 12 - 1 = 11$
$12 > 11 \Rightarrow \neg V << \{X, Y, Z\} \Rightarrow$
$\Rightarrow start(V) \geq \min(ect(X), ect(Y), ect(Z)) \Rightarrow$
$start(V) \geq \min(3, 5, 7) \Rightarrow$
$start(V) \geq 3 \Rightarrow start(V) \in \langle 3, 7 \rangle$

$\boldsymbol{(V, \{X, Y, Z\}), end(V)}$
$p(\{X, Y, Z\} \cup \{V\}) = 12$
$lct(V) - est(\{X, Y, Z\}) = 9 - 0 = 9$
$12 > 9 \Rightarrow \neg \{X, Y, Z\} << V \Rightarrow$
$\Rightarrow end(V) \leq \max(lst(X), lst(Y), lst(Z)) \Rightarrow$
$\Rightarrow end(V) \leq \max(7, 7, 8) \Rightarrow$
$end(V) \leq 8 \Rightarrow end(V) \in \langle 3, 8 \rangle$

Příklad 13.18

V nejlepším možném případě budou úlohy B a E naplánovány před úlohou C.

$$\begin{aligned} orp(C) = InitLevel \quad & +cap(C) \quad + \sum\nolimits_{X<<C} cap(X) \quad + \sum\nolimits_{X??C \,\&\, cap(X)>0} cap(X) \\ = 4 \quad & +(-4) \quad +0 \quad +1+1 \\ = 2 \quad & \end{aligned}$$

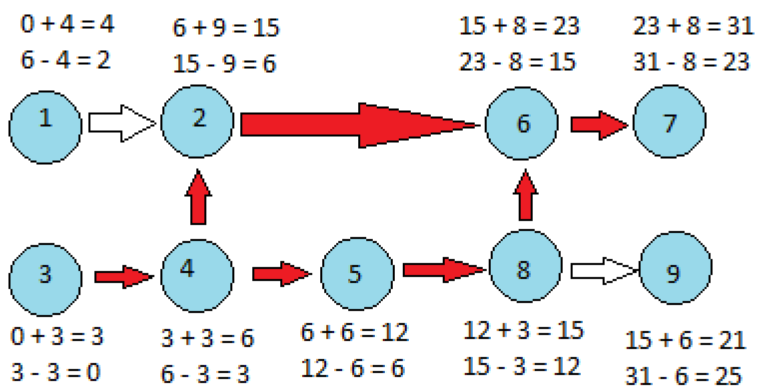
Může být úloha C naplánována po úloze A?

$$\begin{aligned} orp(C|A << C) = InitLevel \quad & +cap(C) \quad + \sum\nolimits_{X<<C} cap(X) \quad + \sum\nolimits_{X??C \,\&\, cap(X)>0} cap(X) \\ = 4 \quad & +(-4) \quad +(-3) \quad +1+1 \\ = -1 \quad & \end{aligned}$$

To znamená, že pokud by úloha A byla spuštěna před úlohou C, tak by úloze C v době jejího spuštění chyběla jedna jednotka zdroje. Tedy C nemůže být naplánováno po A.

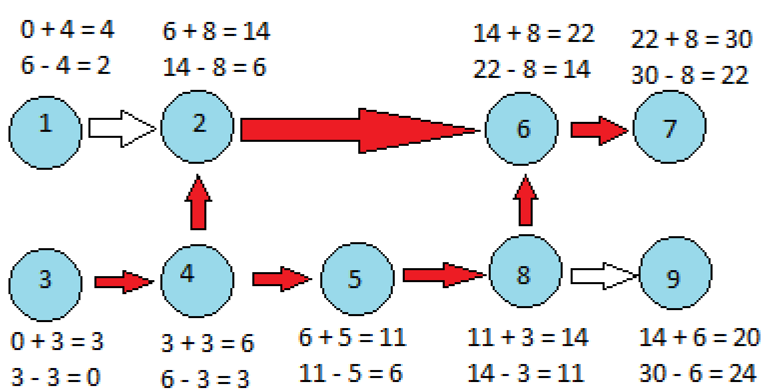
Příklad 19.5 i)

- $p_j = p_j^{\max}$ pro $j = \{1, \ldots, 9\}$
- Určení kritických cest



Horní rovnice u každé úlohy představuje $S'_j + p_j = C'_j$ při dopředné proceduře, spodní rovnice ukazuje $C''_j - p_j = S''_j$ při zpětné proceduře. Červeně vybarvené šipky značí hrany kritických cest.

- Minimální řezy grafu kritických cest: $\{3\}, \{4\}, \{2, 5\}, \{2, 8\}, \{6\}, \{7\}$
- Minimální řez s nejnižší cenou je $\{2, 5\}$.
- Redukce dob provádění: $p_2 = 8$, $p_5 = 5$
- Určení nových kritických cest



Iniciální cena
Fixní režijní náklady: $C_{\max} c_0 = 31 \cdot 4 = 124$
Cena za provádění úloh: $\sum_{j=1}^{9} c_j^b = 4 + 7 + 5 + 4 + 5 + 8 + 9 + 7 + 6 = 55$
$F(p_j^{\max}) = 124 + 55 = 179$

Cena po jednom kroku algoritmu
Fixní režijní náklady: $30 \cdot 4 = 120$

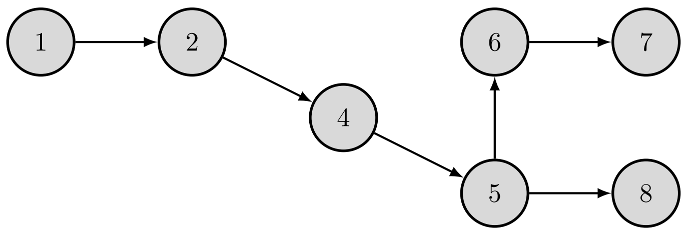
Cena za provádění úloh naroste o $c_2 + c_5 = 2$.
$F(p_j) = 120 + 57 = 177$

Algoritmus skončí, když v grafu kritických cest neexistuje minimální řez R takový, že $\forall j \in R : p_j > p_j^{\min}$, nebo když cena minimálního řezu s nejnižší cenou není menší než fixní režijní náklady c_0.

Příklad 19.5 ii)

iniciální kritické cesty:

- $1 \to 2 \to 4 \to 5 \to 6 \to 7$
- $1 \to 2 \to 4 \to 5 \to 8$



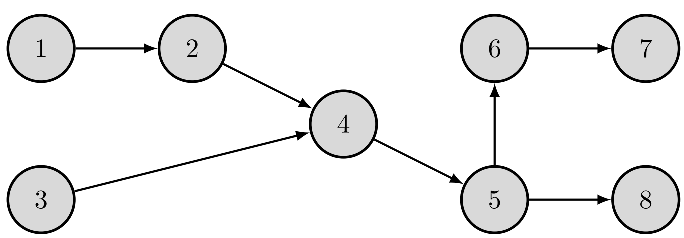
G_{CP}

minimální řezy	$\{1\}$	$\{2\}$	$\{4\}$	$\{5\}$	$\{6,8\}$	$\{7,8\}$
lze vylepšit?	Ano	Ano	Ano	Ano	Ano	Ano
cena	3	2	3	3	5	2

Jeden z řezů s nejmenší cenou je $\{2\}$. Zredukování doby provádění všech úloh v tomto řezu, tedy pouze úlohu 2 na $p_2 = 1$.

nové kritické cesty:

- $1 \to 2 \to 4 \to 5 \to 6 \to 7$
- $1 \to 2 \to 4 \to 5 \to 8$
- $3 \to 4 \to 5 \to 6 \to 7$
- $3 \to 4 \to 5 \to 8$



revidovaný G_{CP}

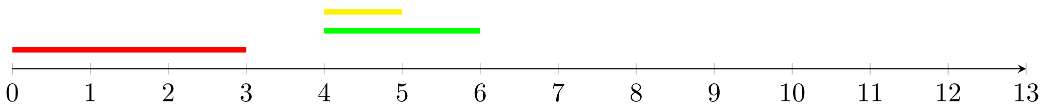
Iniciální cena:
$F(p_j) = F(p_j^{max}) = C_{max}c_0 + \sum_j c_j^b = 21 \cdot 5 + 9 + 7 + 5 + 6 + 7 + 9 + 9 + 6 = 163$

Cena po jednom kroku:
$F(p_j) = F(p_j^{max}) - c_0 + c_2(p_2^{max} - p_2) = 163 - 5 + 2 = 160$

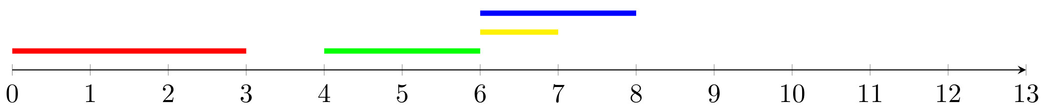
Příklad 21.8

Pravidlo *EDD* vybírá prioritně úlohy s nejdřívějším termínem dokončení. Protože máme termíny dostupnosti, není možné rozvrh vytvořit jednorázově na začátku.

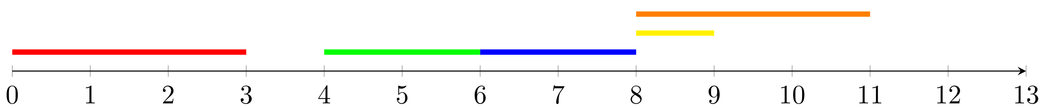
V čase 0 máme dostupnout pouze úlohu 3, musíme ji tedy vybrat.



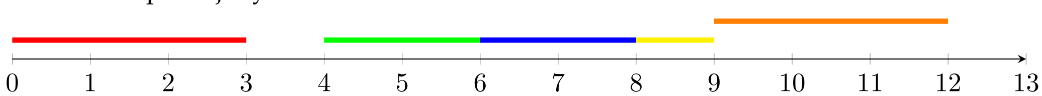
Znovu se rozhodujeme po dokončení úlohy 3 v čase 3. Nemáme ale žádné dostupné úlohy. V čase 4 přijdou úlohy 2 a 4. Vybereme tu s menším d_i, tedy úlohu 2.



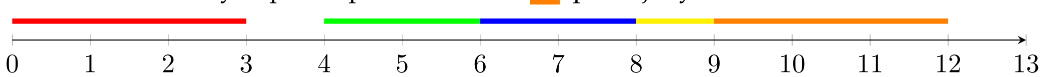
V čase 6 se rozhodujeme mezi úlohami 4 a 1, vybíráme úlohu 1.



Další rozhodování následuje v čase 8, vybíráme z úloh 4 a 5. Úloha 4 má dřívější termín dokončení a proto ji vybereme.



V čase 9 už nám zbývá pouze poslední úloha 5, proto ji vybereme.



Koncové časy tedy jsou $C_1 = 8$, $C_2 = 6$, $C_3 = 3$, $C_4 = 9$ a $C_5 = 12$.

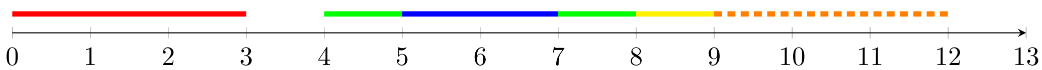
Hodnota účelové funkce
$L_{\max} = \max(\{C_i - d_i | 1 \leq i \leq 5\}) = \max(\{8-7, 6-7, 3-4, 9-11, 12-12\}) = 1$

Pokud můžeme přerušovat úlohy, mohli bychom se teoreticky dostat k lepšímu výsledku. Začátek algoritmu proběhne stejně: vybereme jedinou dostupnou úlohu 3, po jejím doběhnutí budeme muset počkat jednu časovou jednotku, až se v čase 4 objeví další dostupné úlohy 2 a 4, přičemž z nich vybereme například úlohu 2.

V čase 5 se objeví nová úloha 1. Musíme tedy rozhodnout, jestli ji nemáme spustit. Máme $d_1 = 7$, $d_2 = 7$, $d_4 = 11$. Úlohy 2 a 1 mají stejné termíny dokončení, vybereme úlohu 1, protože má nižší identifikátor. Další rozhodování nastane v čase 7, kdy aktuální úloha doběhne. Vybíráme z $d_2 = 7$, $d_4 = 11$. Vrátíme se tedy k úloze 2 a necháme ji doběhnout.

V čase 8 se objevuje úloha 5. Přednost ale dostane dostane úloha 4 (protože $d_4 = 11 < 12 = d_5$). Nakonec spustíme úlohu 5.

Výsledný rozvrh bude vypadat následovně:



Jediný rozdíl oproti rozvrhu bez přerušování je v úlohách 2 a 1, jejich časy dokončení se změnily na $C_1 = 7$, $C_2 = 8$. Hodnota účelové funkce se nezmění, největší zpoždění pořád zůstane 1, ale bude na úloze 2, nikoli 1.

Příklad 22.5

Uvažujeme první úroveň větvení. Nemá smysl rozvrhovat úlohu 3 jako první, tj. (3,*,*), protože úloha 2 musí být vždy rozvržena před úlohou 3 vzhledem k $r_2 + p_2 \leq r_3$. Zbývají nám tedy dvě možnosti pro první úlohu. Spočítáme řešení pro problém $1|r_j, prmp|L_{\max}$ s úlohou 1 jako první, tj. (1,*,*), a následně s úlohou 2 jako první, tj. (2,*,*). Výsledné hodnoty účelové funkce nám dají dolní odhad pro jednotlivé uzly.

Pro (1,*,*) dostaneme rozvrh

- 1 (0..4) $L_1 = -4$,
- 3 (4..10) $L_3 = 0$,
- 2 (10..12) $L_2 = 1$

s $L_{\max} = 1$. Tento rozvrh je zároveň nepreemtivní a tedy víme, že ve větvi (1,*,*) už nic lepšího nenajdeme a pro tuto větev máme řešení (1,3,2).

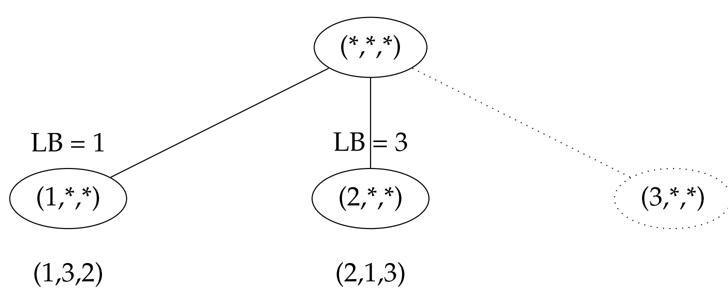
Pro větev (2,*,*) najdeme rozvrh

- 2 (1..3) $L_2 = -8$,
- 1 (3..7) $L_1 = -1$,
- 3 (7..13) $L_3 = 3$

s $L_{\max} = 3$. Tento rozvrh je také nepreemtivní (a nic lepšího tedy v této větvi nenajdeme). Tento rozvrh je zároveň horší než předchozí rozvrh (1,3,2). Dále tedy už nemá smysl dále prohledávat.

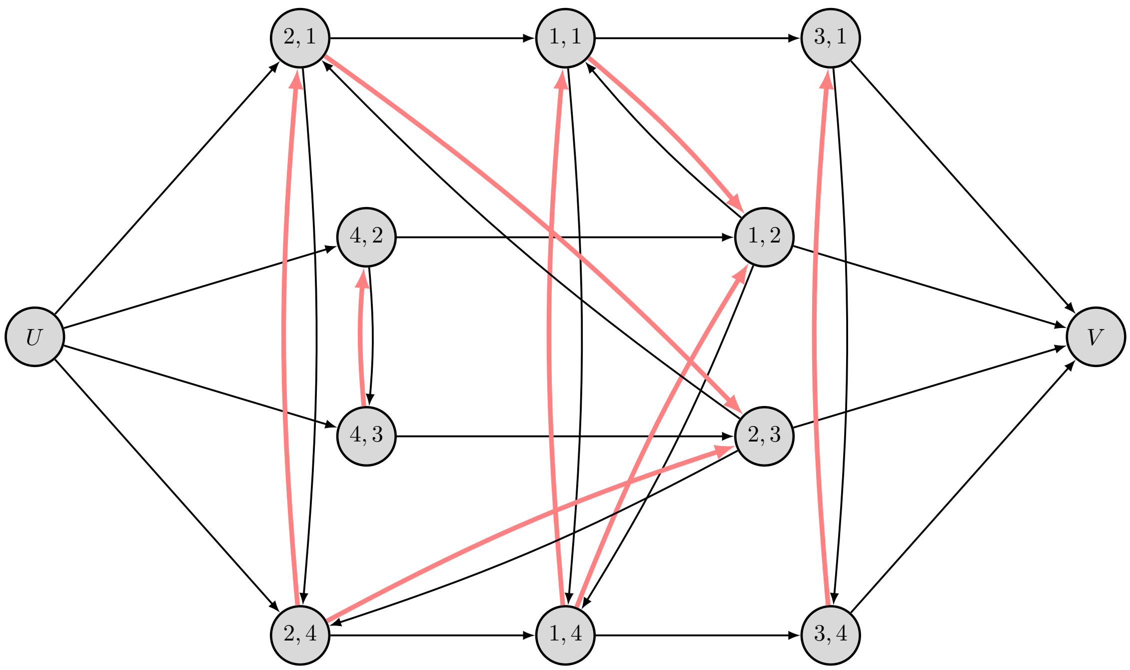
Optimální rozvrh pro naše zadání je tedy (1,3,2).

Následující obrázek zobrazuje prohledávaný strom. Tečkovaný uzel nebylo třeba prohledávat.



Příklad 24.2

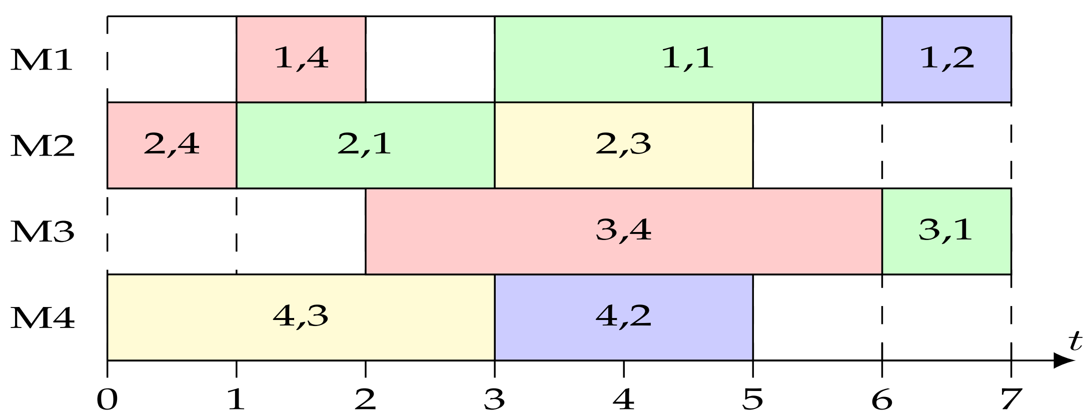
Řešení numerické části příkladu:



disjunktní grafová reprezentace

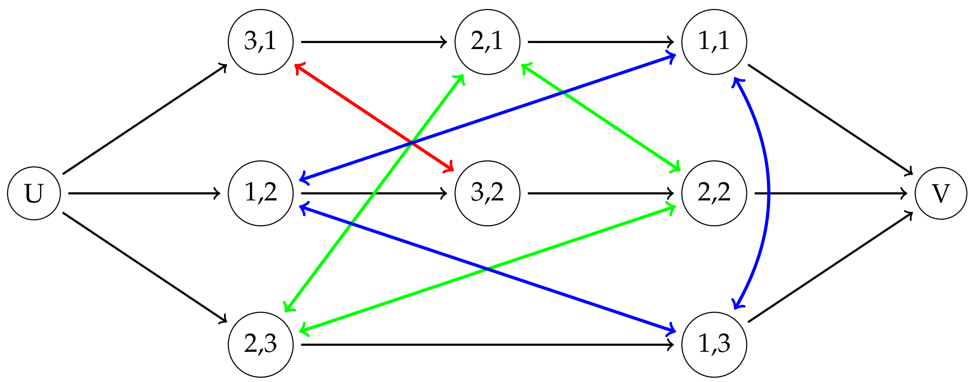
Splnitelný výběr ukazují červeně označené hrany v grafové reprezentaci tedy
$\{((1,1),(1,2));((1,4),(1,1));((1,4),(1,2));$
$((2,4),(2,1));((2,4),(2,3));((2,1),(2,3));$
$((3,4),(3,1));$
$((4,3),(4,2))\}$

Níže je rozvrh splňující zadání s co nejmenšími startovními časy.

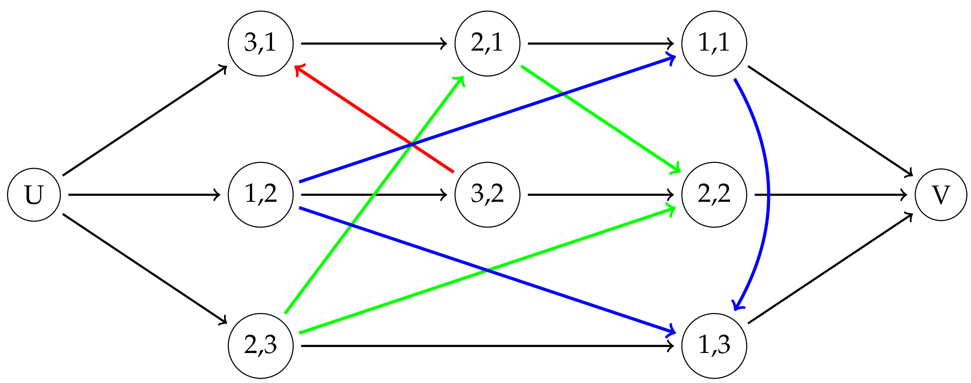


Příklad 24.3

Máme tři stroje a tři úlohy. Disjunktivní grafová reprezentace je na následujícím obrázku. Všechny hrany se šipkami na obou koncích (červené, zelené a modré) reálně reprezentují dvě jednosměrné hrany s opačnými orientacemi.



Splnitelný výběr reprezentovaným obrázkem ze zadání je na následujícím obrázku.



Disjunktivní grafová reprezentace (DGR) určuje zadání problému, splnitelný výběr (SV) je už jeho řešení. DGR obsahuje disjunktivní hrany v obou směrech, SV obsahuje z každého páru disjunktivních hran právě jednu hranu a vybrané hrany reprezentují uspořádání úloh na strojích. Splnitelný výběr je vlastním acyklickým podgrafem DGR (některé hrany byly vypuštěny).

Zadání úlohy pro *job shop*:

- Úlohy:
  - J1: (3,1) → (2,1) → (1,1)
  - J2: (1,2) → (3,2) → (2,2)
  - J3: (2,3) → (1,3)
- Doby provádění:
  - $p_{11} = 1$, $p_{21} = 2$, $p_{31} = 4$,
  - $p_{12} = 3$, $p_{22} = 1$, $p_{32} = 3$,
  - $p_{13} = 4$, $p_{23} = 2$

Příklad 26.3

Nejprve je třeba provést inicializaci:

$$\begin{aligned} \Omega &:= \{(2,1),(1,2),(3,3)\} \\ r_{ij} &:= 0 \quad \forall (i,j) \in \Omega \end{aligned}$$

Nyní začínáme s prázdným rozvrhem.

V první iteraci musíme určit, kdy nejdříve může nějaká úloha z Ω skončit.

$$\begin{aligned} t(\Omega) &:= min_{(i,j)\in\Omega}\{r_{ij} + p_{ij}\} \\ &= min\{4+0, 2+0, 3+0\} = 2 \end{aligned}$$

Minimum je realizováno úlohou $(1,2)$, tedy na stroji $i^* = 1$. Nyní musíme určit množinu $\Omega' \subseteq \Omega$ úloh, které přidáme do rozvrhu.

$$\Omega' \quad := \quad \{(i^*, j) | r_{i^*j} < t(\Omega)\} = \{(1,2)\}$$

Do rozvrhu tedy přidáme úlohu $(1,2)$, která poběží od času 0 do času 2 na stroji 1.

Zároveň musíme aktualizovat množinu úloh s narozvrhovanými předchůdci, tedy odstraníme rozvrhnutou úlohu a přidáme její následníky.

$$\begin{aligned} \Omega &:= \Omega \setminus \{(1,2)\} \cup \{(2,2)\} = \{(2,1),(2,2),(3,3)\} \\ r_{22} &:= 2 \end{aligned}$$

Nyní můžeme jít znovu na výběr první úlohy, která by měla skončit.

$$\begin{aligned} t(\Omega) &:= min_{(i,j)\in\Omega}\{r_{ij} + p_{ij}\} \\ &= min\{4+0, 1+2, 3+0\} = 3 \end{aligned}$$

Nyní je minimum dosaženo na strojích 2 a 3. Vybereme z nich pro větvení stroj s nejnižším identifikátorem, tedy stroj 2.

Určíme množinu úloh pro přidání do rozvrhu:

$$\Omega' \quad := \quad \{(2,j) | r_{2j} < 3\} = \{(2,1),(2,2)\}$$

Do jednoho rozvrhu přidáme úlohu (2,1), která poběží od času 0 do času 4 na stroji 2 a aktualizujeme množinu Ω.

$$\begin{aligned} \Omega &:= \Omega \setminus \{(2,1)\} \cup \{(3,1)\} = \{(3,1),(2,2),(3,3)\} \\ r_{31} &:= 4 \end{aligned}$$

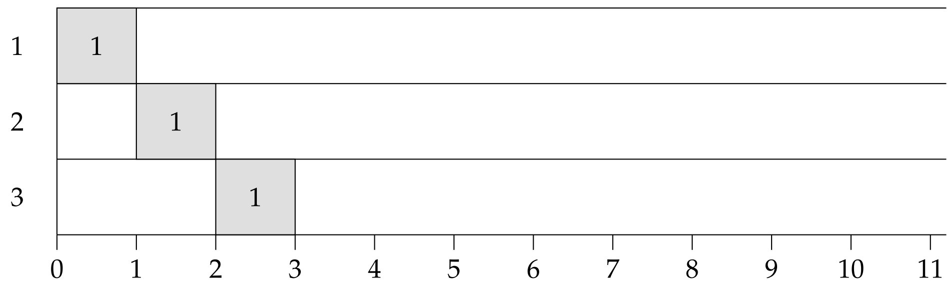
Do druhého rozvrhu přidáme úlohu (2,2), která poběží na stroji č. 2 od času 2 do času 3. Opět aktualizuje množinu dostupných úloh a počátečních časů.

$$\begin{aligned} \Omega &:= \Omega \setminus \{(2,2)\} \cup \{(3,2)\} = \{(2,1),(3,2),(3,3)\} \\ r_{32} &:= 3 \end{aligned}$$

Nyní bychom mohli pro každý částečný rozvrh pokračovat dál.

Příklad 28.8

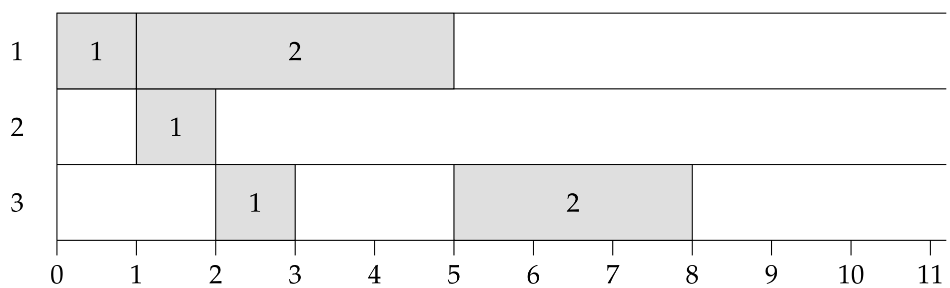
Rozvrh pro první úlohu:



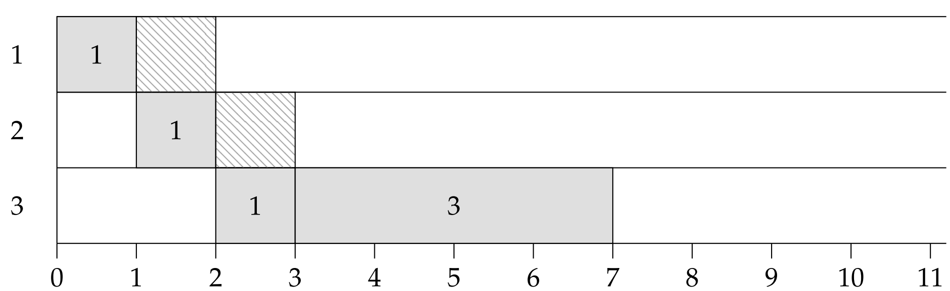
Označíme N_{i,j_k} jako neproduktivní dobu na stroji i pro úlohu, která je kandidátem na pozici k.

2. pozice

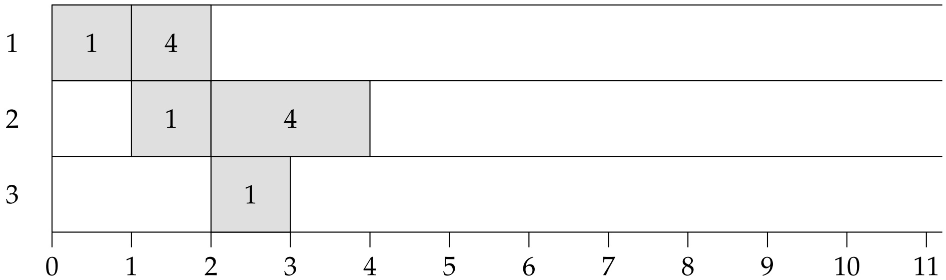
Rozvrh pro kandidátní úlohu 2 na druhou pozici ($j_2 = 2$)



Rozvrh pro kandidátní úlohu 3 na druhou pozici ($j_2 = 3$)



Rozvrh pro kandidátní úlohu 4 na druhou pozici ($j_2 = 4$)

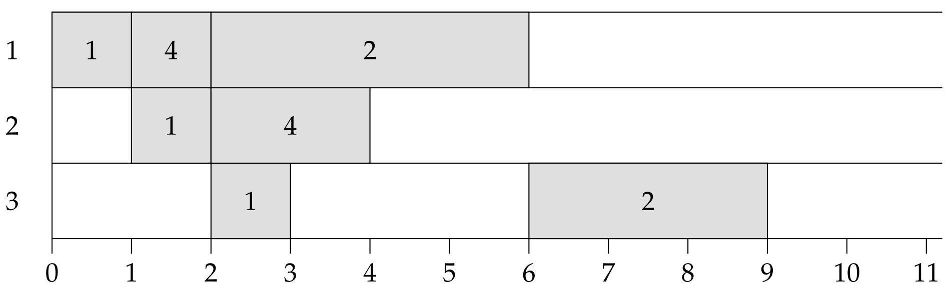


Výpočet neproduktivní doby pro výše uvedené kandidátní úlohy na druhou pozici:

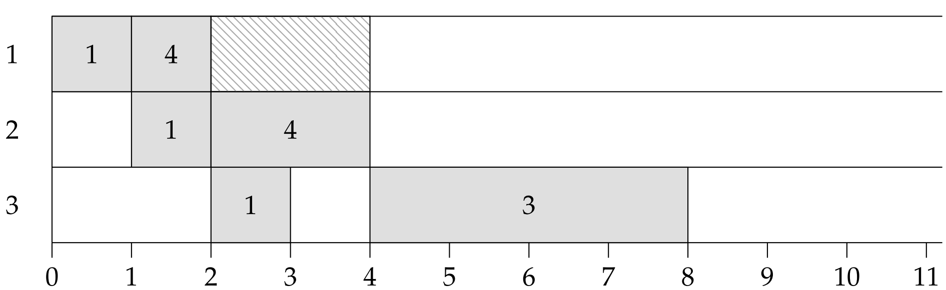
	X_{i,j_2}			N_{i,j_2}			
	i			i			
j_2	1	2	3	1	2	3	$\sum_{i=1}^{3} N_{i,j_2}$
2	5	5	8	0	3	2	5
3	2	3	7	1	1	0	2
4	2	4	4	0	0	1	1

3. pozice

Rozvrh pro kandidátní úlohu 2 na třetí pozici ($j_3 = 2$)



Rozvrh pro kandidátní úlohu 3 na třetí pozici ($j_3 = 3$)

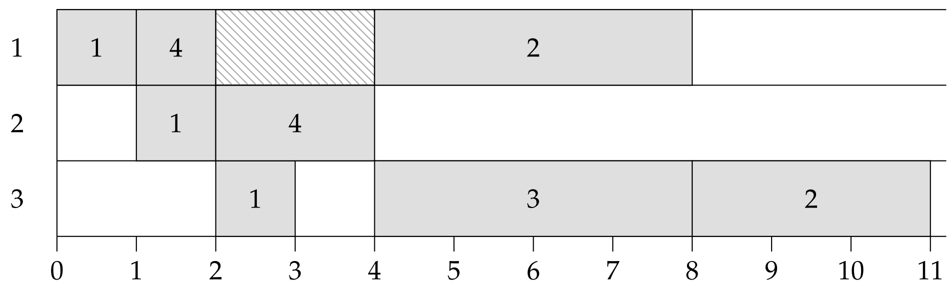


Výpočet neproduktivní doby pro výše uvedené kandidátní úlohy na třetí pozici:

	X_{i,j_3}			N_{i,j_3}			
	i			i			
j_3	1	2	3	1	2	3	$\sum_{i=1}^{3} N_{i,j_3}$
2	6	6	9	0	2	2	4
3	4	4	8	2	0	0	2

4. pozice

Rozvrh pro poslední úlohu 2 na čtvrtou pozici ($j_4 = 2$)

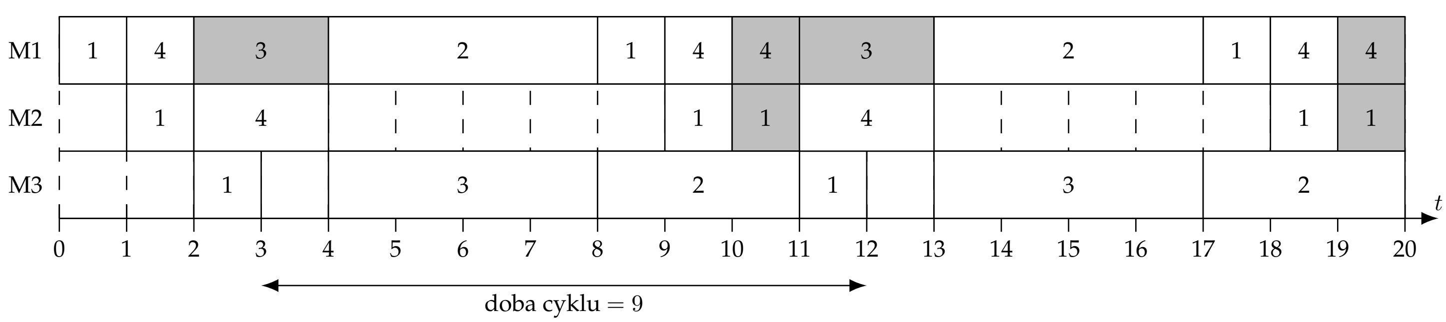


Výpočet neproduktivní doby pro poslední úlohu na čtvrté pozici:

	X_{i,j_4}			N_{i,j_4}			
	i			i			
j_4	1	2	3	1	2	3	$\sum_{i=1}^{3} N_{i,j_4}$
2	8	8	11	0	4	0	4

Výsledný rozvrh

Pořadí úloh při použití heuristiky padnoucího profilu při fixované první úloze je 1, 4, 3 a 2.



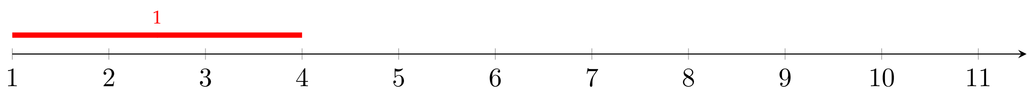
Příklad 35.5

Řešíme problém intervalového rozvrhování. Všechny úlohy mají jednotkovou váhu a máme dva identické stroje. Máme k dispozici optimální algoritmus.

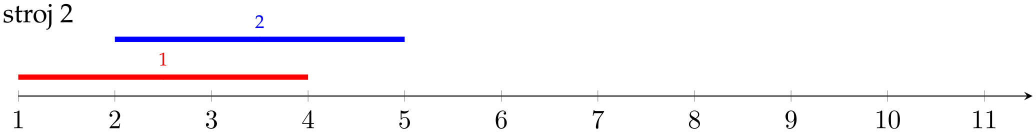
Nejprve musíme seřadit úlohy podle času dostupnosti. Dostaneme stejné pořadí jako v zadání.

Nyní můžeme začít s jednotlivými iteracemi:

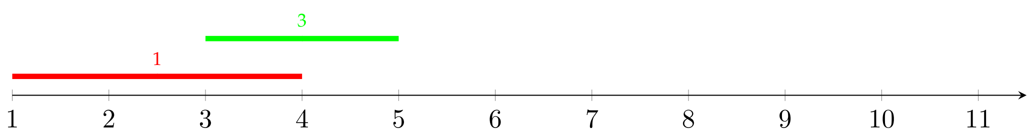
$j = 1$: V čase $r_1 = 1$ máme dostupný stroj, proto úlohu 1 rozvrhneme na stroj 1



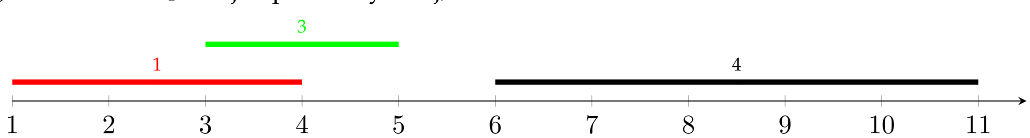
$j = 2$: V čase $r_2 = 2$ máme volný druhý stroj, proto i úlohu 2 můžeme rovnou rozvrhnout, a to na stroj 2



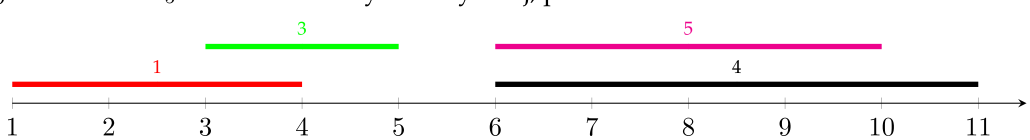
$j = 3$: V čase $r_3 = 3$ není volný žádný stroj. Poslední dokončenou úlohou bude úloha 2 (tedy $j^* = 2$) v čase 5. Úloha 3 končí ve stejný čas, podle algoritmu tedy rozvrhneme úlohu 3 místo úlohy 2, protože neplatí podmínka, že $C_3 > C_2$. Optimální řešení bychom ovšem získali i v případě, že bychom ponechali rozvrhnutou úlohu 2 (maximální počet realizovaných úloh by byl stejný), navíc je však v tomto řešení obsazený interval 2–3.



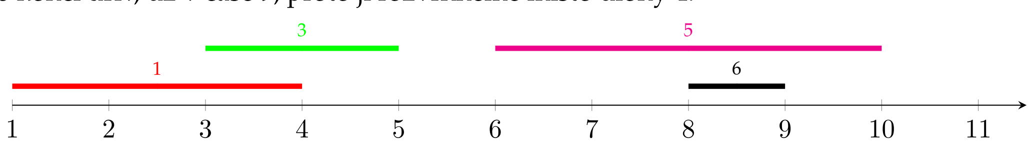
$j = 4$: V čase $r_4 = 6$ je opět volný stroj, úlohu 4 můžeme rozvrhnout.



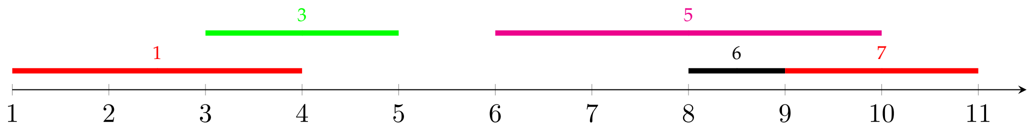
$j = 5$: V čase $r_5 = 6$ máme volný i druhý stroj, proto úlohu 5 máme kde rozvrhnout.



$j = 6$: V čase $r_6 = 8$ nemáme žádný volný stroj. Poslední úlohou je úloha 4 končící v čase 11. Úloha 8 končí dřív, už v čase 9, proto ji rozvrhneme místo úlohy 4.



$j = 7$: V čase $r_7 = 9$ máme dostupný stroj, proto úlohu 7 můžeme rozvrhnout.

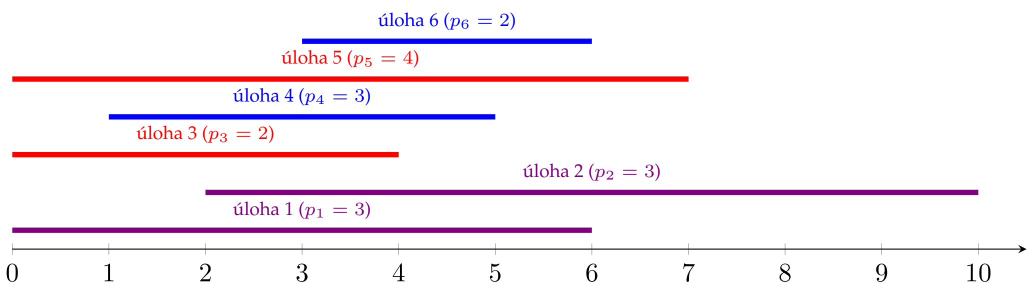


Množina všech rozvržených úloh $J = \{1, 3, 5, 6, 7\}$.

Příklad 36.3

Máme zadané úlohy s časovou rezervou a váhami, chceme je rozvrhnout na dva stroje. Použijeme prioritní index $I_j = \frac{|M_j|}{w_j/p_j}$.

Následující diagram znázorňuje intervaly, ve kterých je možné jednotlivé úlohy rozvrhnout. Červené úlohy vyžadují stroj 1, modré stroj 2 a fialové stroj 1 nebo 2 (v dalších diagramech už barvy nenesou žádný význam).



Nyní můžeme spočítat priority pro jednotlivé úlohy:

úloha	1	2	3	4	5	6
I_j	3	6	1/2	1/2	2	2/3

Dále budeme potřebovat počty úloh, které lze přiřadit stroji i v intervalu $[t-1, t]$, tj. ψ_{it}.

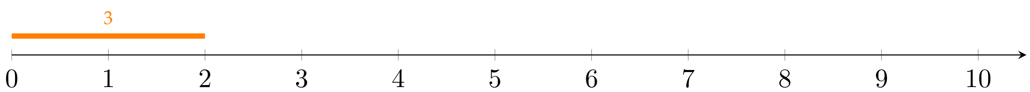
čas	1	2	3	4	5	6	7	8	9	10
ψ_{1t}	3	3	4	4	3	3	2	1	1	1
ψ_{2t}	1	2	3	4	4	3	1	1	1	1

Prioritní index stroje budeme počítat jako $g(\psi_{i,t+1}, \ldots, \psi_{i,t+p_j}) = (\sum_{l=1}^{p_j} \psi_{i,t+l})/p_j$

1. Vybereme úlohu s nejnižším indexem, tj. úlohu 3 ($p_3 = 2$) (úloha 4 má stejný prioritní index, ale vyšší identifikátor). Nyní pro ni musíme vybrat stroj a čas, tedy určit jejich priority. Úloha 3 vyžaduje stroj 1, takže stačí uvažovat startovní časy 0, 1 a 2 na tomto stroji.

$$\begin{aligned} g(\psi_{1,1}, \psi_{1,2}) &= (3+3)/2 = 3 \\ g(\psi_{1,2}, \psi_{1,3}) &= (3+4)/2 = 3,5 \\ g(\psi_{1,3}, \psi_{1,4}) &= (4+4)/2 = 4 \end{aligned}$$

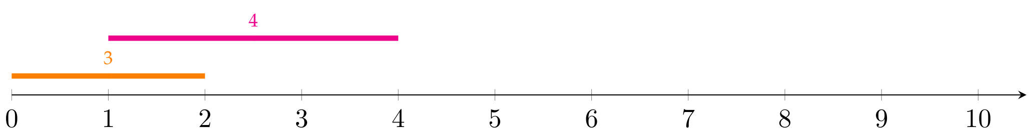
Minimum vyjde pro čas 0. Proto úlohu 3 rozvrhneme v čase 0–2 na stroji 1.



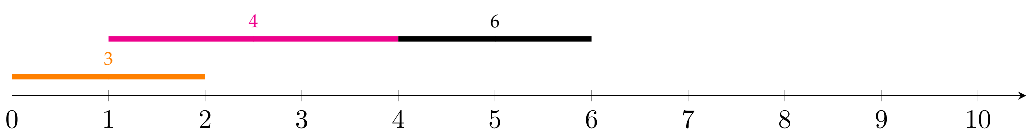
2. Vybereme další úlohu s nejnižší prioritou, tj. úlohu 4 ($p_4 = 3$). Tato úloha musí běžet na stroji 2. Opět musíme vybrat čas.

$$\begin{aligned} g(\psi_{2,2}, \dots, \psi_{2,4}) &= (2+3+4)/3 = 3 \\ g(\psi_{2,3}, \dots, \psi_{2,5}) &= (3+4+4)/3 \cong 3,7 \end{aligned}$$

Minimum vyjde pro počáteční čas 1. Proto úlohu 4 rozvrhneme na stroj 2 v čase 1–4.



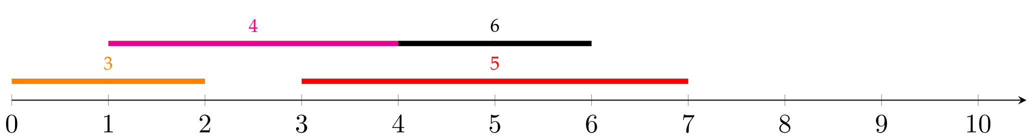
3. Vybereme další úlohu, tentokrát úlohu 6 ($p_6 = 2$). Ta musí také běžet na stroji 2. Stroj 2 se uvolní až v čase 4, úloha 6 musí skončit v čase 6, zbývá tedy jediná možnost, jak úlohu rozvrhnout, a to je interval 4–6 na stroji 2.



4. Další úloha k rozvrhnutí je úloha 5 ($p_5 = 4$), která musí běžet na stroji 1. Potenciální startovní časy jsou 2 a 3 (po dokončení úlohy 3).

$$\begin{aligned} g(\psi_{1,3}, \dots, \psi_{1,6}) &= (4+4+3+3)/4 = 3.5 \\ g(\psi_{1,4}, \dots, \psi_{1,7}) &= (4+3+3+2)/4 = 3 \end{aligned}$$

Zvolíme tedy čas 3 a úlohu 5 rozvrhneme na stroji 1 v intervalu 3–7.



5. Další úloha je číslo 1 ($p_1 = 3$), která může běžet na libovolném stroji. V možném čase 0–6 jsou ovšem volné pouze intervaly délky 1, kam se tato úloha nemůže vejít a proto nebude rozvržena.

6. Poslední úlohou je číslo 2 ($p_2 = 3$), která také může běžet na libovolném stroji. Může být poprvé na stroji 1 rozvržena až v čase 7, což je také poslední možný čas. Pro stroj 2 připadají v úvahu časy 6 a 7.

$$\begin{aligned} g(\psi_{1,8},\ldots,\psi_{1,10}) &= (1+1+1)/3 = 1 \\ g(\psi_{2,7},\ldots,\psi_{2,9}) &= (1+1+1)/3 = 1 \\ g(\psi_{2,8},\ldots,\psi_{2,10}) &= (1+1+1)/3 = 1 \end{aligned}$$

Zjevně jsou všechny možnosti totožné, proto z nich vybereme stroj 2 a čas 7, který minimalizuje čas dokončení poslední úlohy.

Výsledný rozvrh tedy bude vypadat následovně. Úlohu 1 se nepodařilo rozvrhnout.