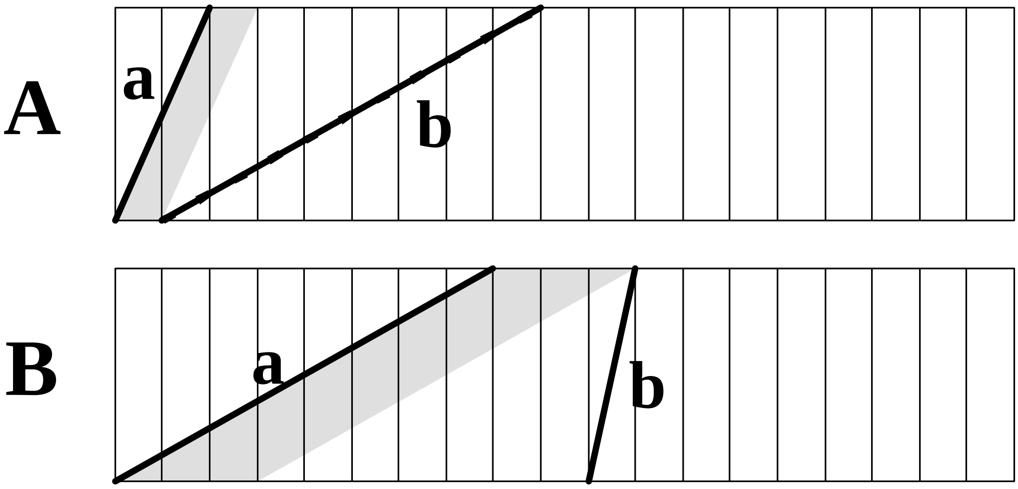
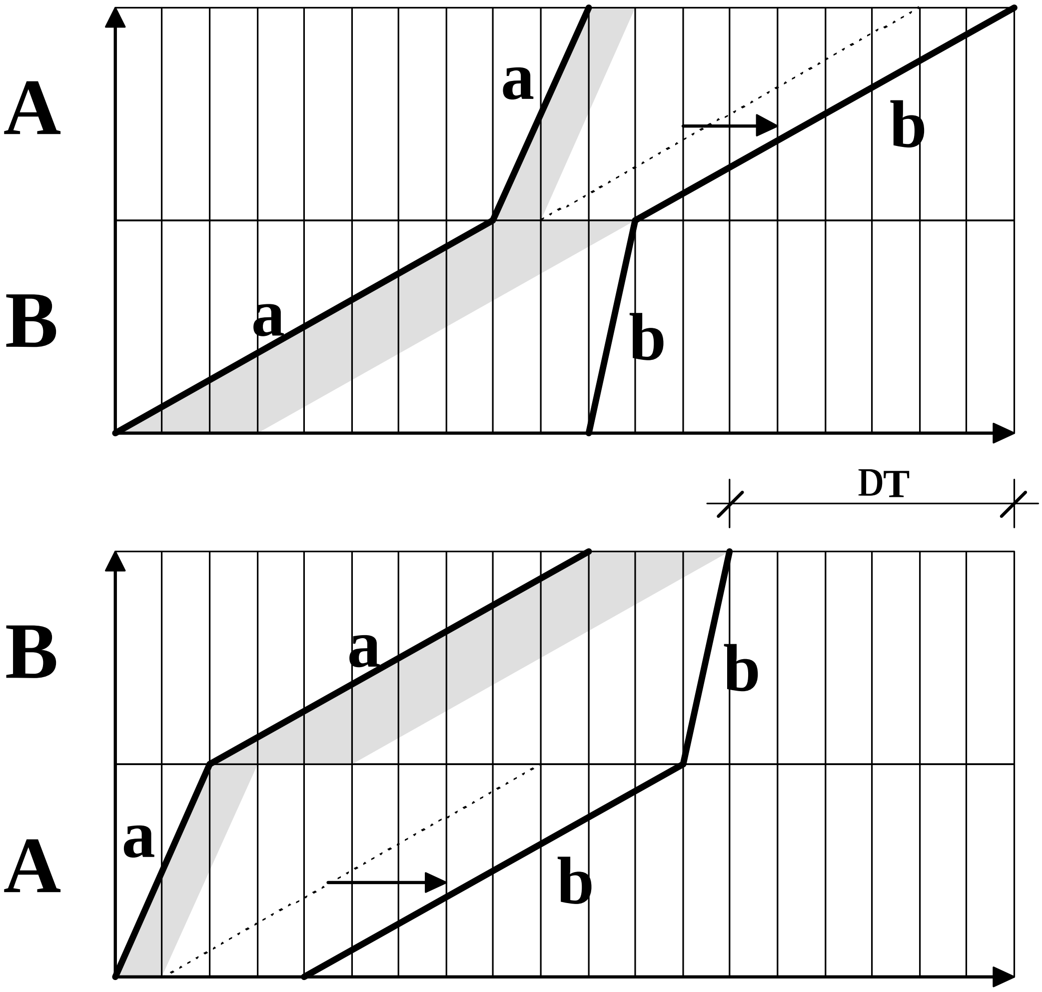
S O R R E N D T E R V E Z É S

Projektek, építmények sorolása

(Multi-Project Management)





Az F½overlap½C^{max} Feladat

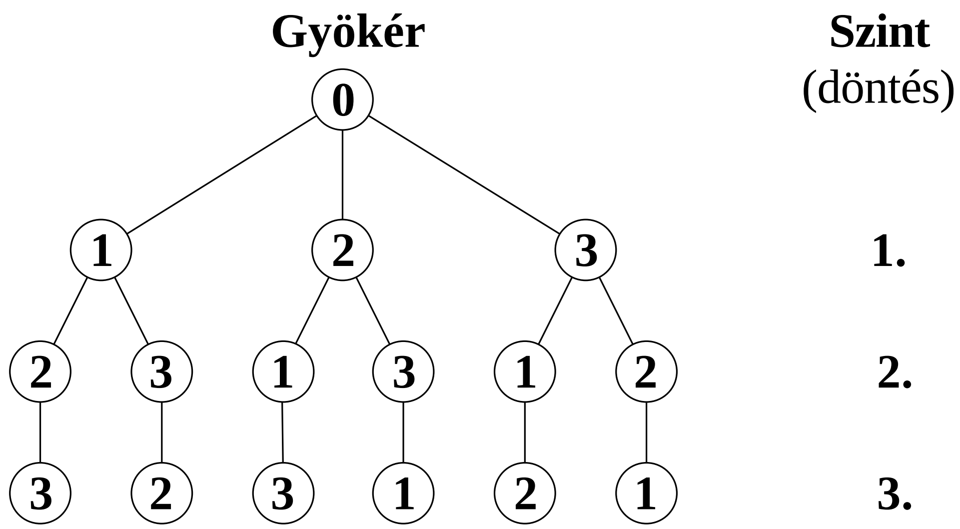
Munkadarabok sorolása párhuzamos gépeken „Flow-Shop Problem”

ID: Graham, Lenstra, Rinnooy Kan, 1979

Megszorítások:

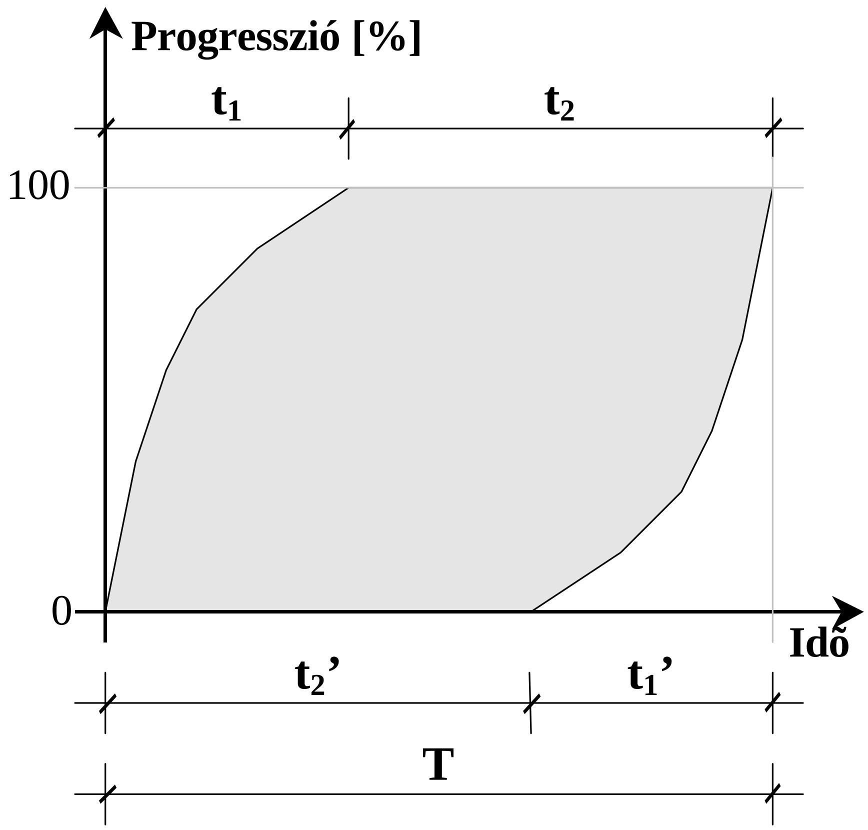
- **Minden munkafolyamat (tevékenység) minden munkadarabon (építményen) elvégzendõ elõre megadott technológiai sorrendben –** hiányzó munkafolyamat nem megengedett … „flow-shop”
- **Minden gép (brigád) egyetlen munkafolyamatot végez minden egyes munkadarabon (építményen)**
- **Minden munkafolyamatot (tevékenységet) egy-egy arra specializált gép (munkacsoport) végez**
- **A munkadarabok (építmények) munkába vételi sorrendje minden gép (brigád) számára azonos –** „elõzés nem megengedett”
- **Minden gép (brigád) folyamatosan, megszakítás nélkül dolgozik –** „megszakítás nem megengedett”
- **Munkafolyamatok idõbeli átlapolása a munkadarabokon megengedett –** „átlapolás megengedett”
- **Cél a teljes átfutási idõ minimálása –** „az összes munkadarab a legrövidebb idõn belül”

A Sorolás Döntési Fája



123 < 132 < 213 < 231 < 312 < 321

A TELJES ÁTFUTÁSI IDÕ
állandó és változó része



$T = t_1 + t_2$

$T = t_1' + t_2'$

$t_1 = \text{const}$

$t_1' = \text{const}$

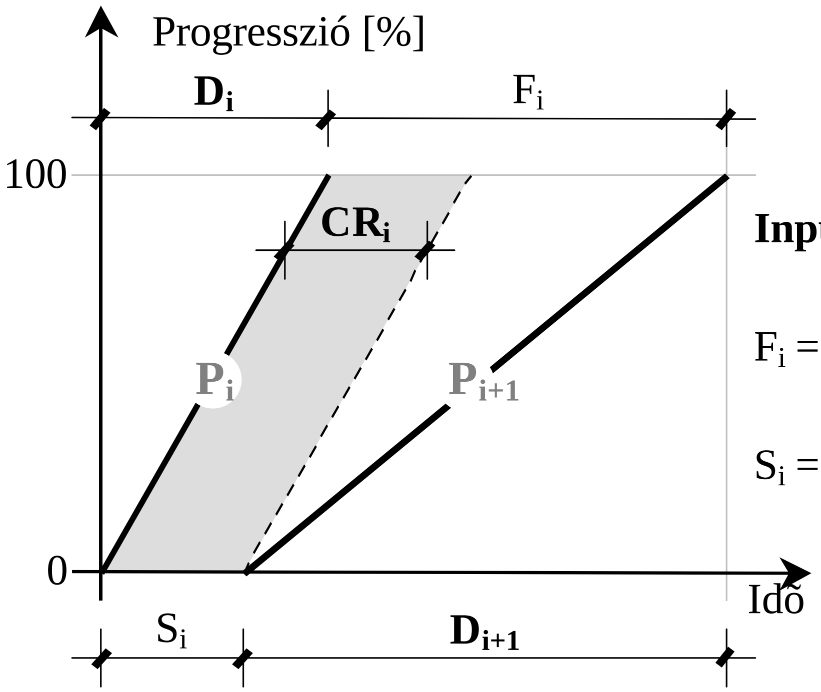
T_{min} Þ $t_{2\ min}$

T_{min} Þ $t_2'_{min}$

Követési idõk meghatározása

Átlapolás megengedett

elõre megadott technológiai követési idõvel

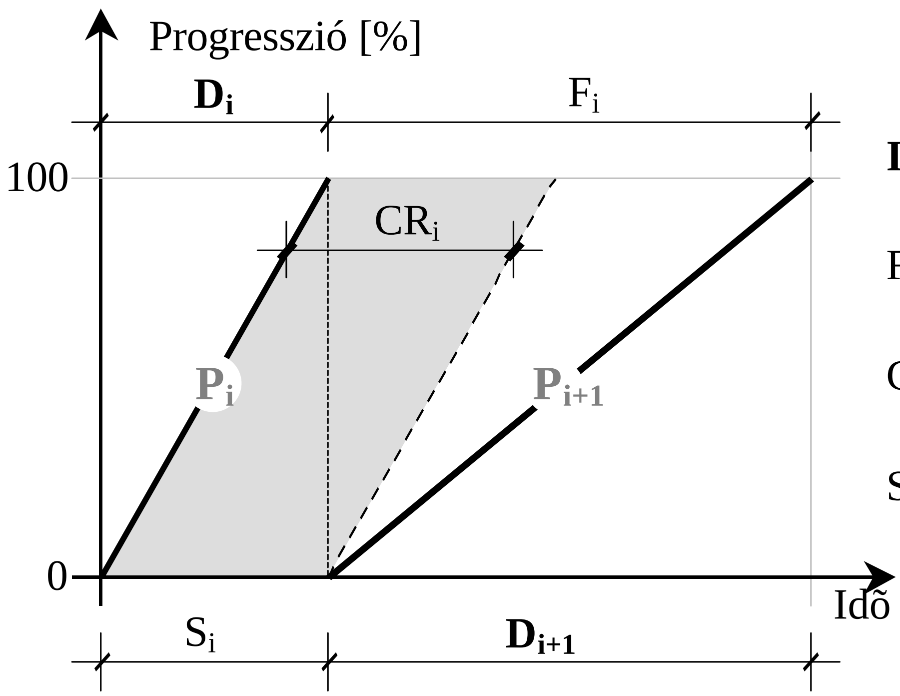


Input adat : $\mathbf{D_i}$ **,** $\mathbf{D_{i+1}}$ **,** $\mathbf{CR_i}$

$F_i = \max\{ CR_i ; CR_i + D_{i+1} - D_i \}$

$S_i = \max\{ CR_i ; CR_i + D_i - D_{i+1} \}$

Átlapolás nem megengedett



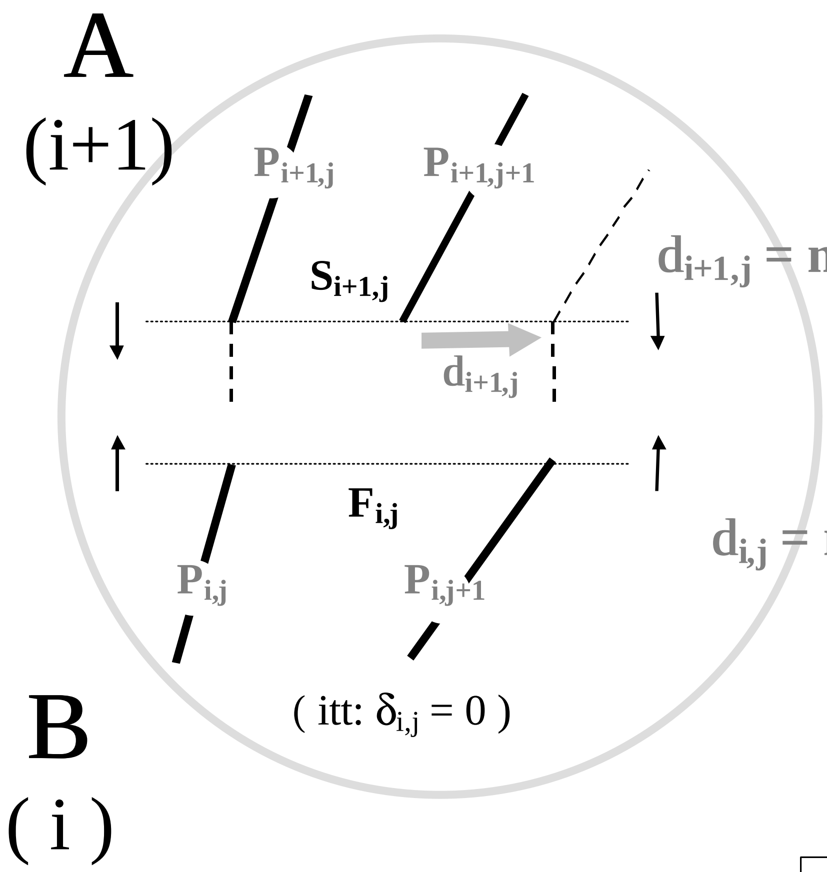
Input adat : $\mathbf{D_i}$ **,** $\mathbf{D_{i+1}}$

$F_i = D_{i+1}$

$CR_i = \min\{ D_i ; D_{i+1} \}$

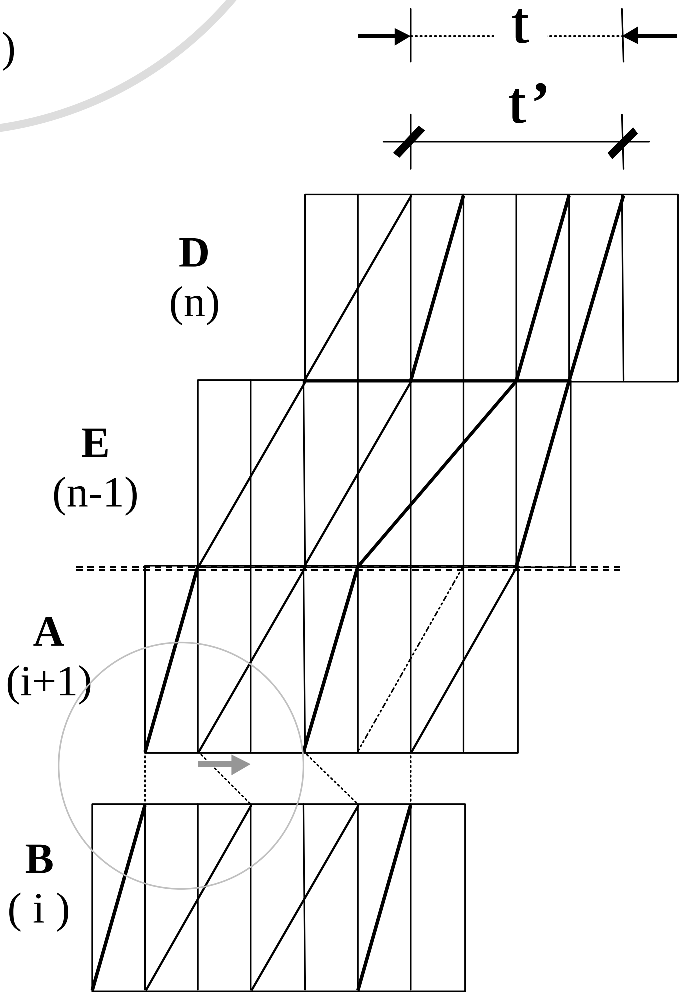
$S_i = D_i$

Egyesített ütemterv kialakítása

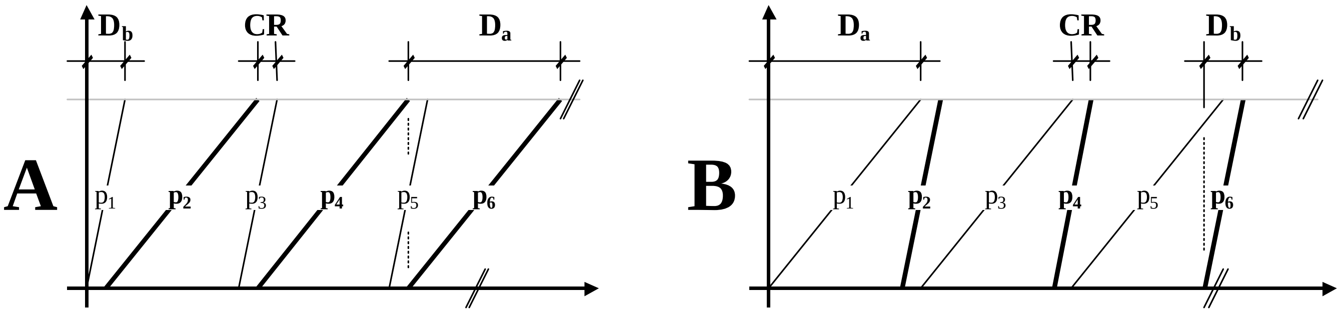


$$d_{i+1,j} = \max\{\, 0 \,;\, F_{i,j} - S_{i+1,j} \,\}$$

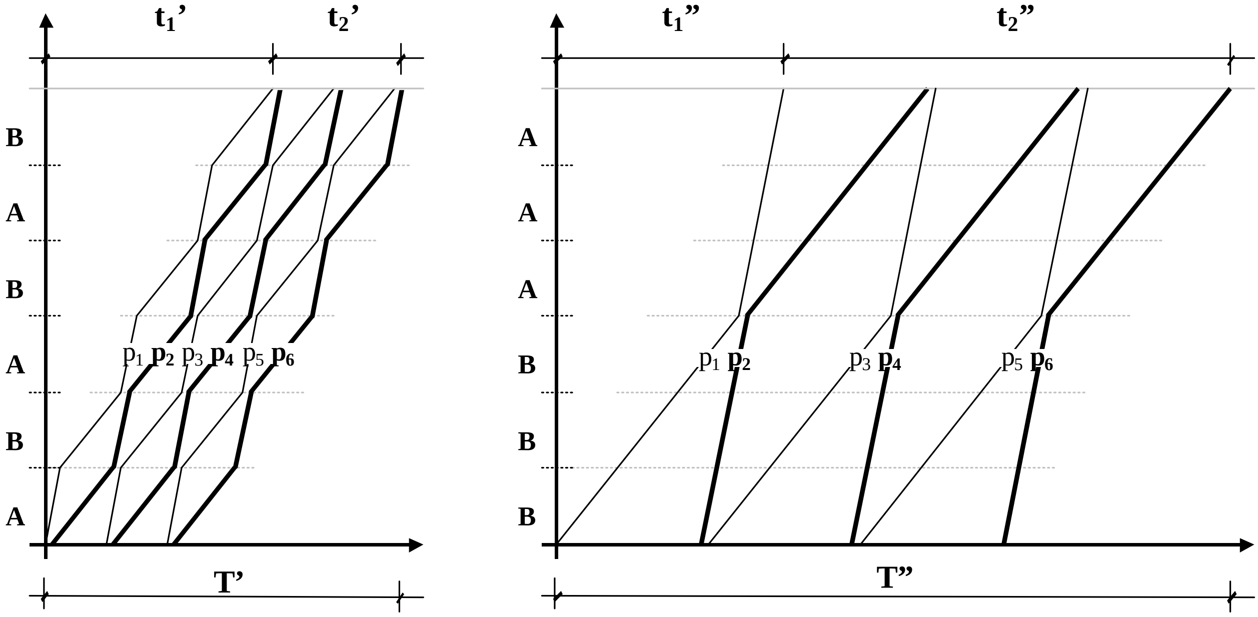
$$d_{i,j} = \max\{\, 0 \,;\, S_{i+1,j} - F_{i,j} \,\}$$



A sorrend „korlátlan” hatása



$$D_a \gg D_b, CR$$



$$\frac{T'}{T''} = \frac{t_1'+t_2'}{t_1''+t_2''} \approx \frac{m\cdot D_a + (n-1)\cdot D_a}{m\cdot D_a + m\cdot n\cdot D_a} = \frac{(m+n-1)\cdot D_a}{m\cdot(n+1)\cdot D_a} = \frac{m+n-1}{m\cdot(n+1)}$$

$$\lim_{m\to\infty}\frac{T'}{T''} \approx \lim_{m\to\infty}\frac{m+n-1}{m\cdot(n+1)} = \lim_{m\to\infty}\frac{m}{m\cdot(n+1)} + \lim_{m\to\infty}\frac{n}{m\cdot(n+1)} - \lim_{m\to\infty}\frac{1}{m\cdot(n+1)} = \frac{1}{n+1} + 0 - 0$$

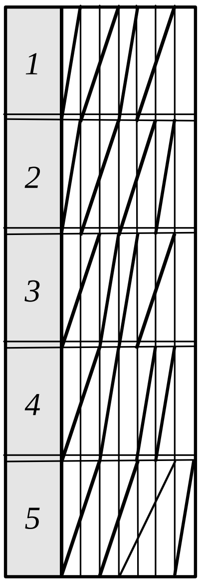
$$\lim_{n\to\infty}\frac{T'}{T''} \approx \lim_{n\to\infty}\frac{m+n-1}{m\cdot(n+1)} = \lim_{n\to\infty}\frac{m}{m\cdot(n+1)} + \lim_{n\to\infty}\frac{n}{m\cdot(n+1)} - \lim_{n\to\infty}\frac{1}{m\cdot(n+1)} = 0 + \frac{1}{m} - 0$$

$$\lim_{\substack{m\to\infty\\ n\to\infty}}\frac{T'}{T''} \approx \lim_{\substack{m\to\infty\\ n\to\infty}}\frac{m+n-1}{m\cdot(n+1)} = \lim_{\substack{m\to\infty\\ n\to\infty}}\frac{m}{m\cdot(n+1)} + \lim_{\substack{m\to\infty\\ n\to\infty}}\frac{n}{m\cdot(n+1)} - \lim_{\substack{m\to\infty\\ n\to\infty}}\frac{1}{m\cdot(n+1)} = 0 + 0 - 0$$

SZÛRÕÉRTÉK MEGVÁLASZTÁSA
a célérték (optimum) becslése alapján

		a	b	c	d	
	i \ j	1	2	3	4	
A	1	1	2	1	2	← $D_{i,j}$
			1	1	1	← $CR_{i,j}$
B	2	1	2	2	1	
			1	2	1	
C	3	2	1	1	2	
			1	1	1	
D	4	2	1	1	1	
			1	2	1	
E	5	2	2	3	1	
			2	1	1	
		8	8	8	7	← $\sum_{i=1}^{m} D_{i,j}$
			1	1	1	← $\min_i\{CR_{i,j}\}$

		a	b	c	d
	i \ j	1	2	3	
A	1	2	1	2	← $F_{i,j}$
		1	2	1	← $S_{i,j}$
B	2	2	2	1	
		1	2	2	
C	3	1	1	2	
		2	1	1	
D	4	1	2	1	
		2	2	1	
E	5	2	2	1	
		2	1	3	



A	1	**5**	← $\sum_{j=1}^{n-1} F_{i,j}$
		4	← $\sum_{i=1}^{n-1} S_{i,j}$
B	2	**5**	
		5	
C	3	**4**	
		4	
D	4	**4**	
		5	
E	5	**5**	
		6	

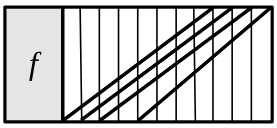
Feltételezve az elsõ/utolsó projekt változtatásának szükségtelenségét:

$$E_1 = \sum_{i=1}^{m} D_{i,1} + \min_i\{\sum_{j=1}^{n-1} F_{i,j}\}$$

$$E_2 = \sum_{i=1}^{m} D_{i,n} + \min_i\{\sum_{j=1}^{n-1} S_{i,j}\}$$

	a	b	c	d	
j	1	2	3	4	
f	8	8	8	7	← $\sum_{i=1}^{m} D_{i,j}$
		1	1	1	← $\min_i\{CR_{i,j}\}$

	a	b	c	d
j	1	2	3	
f	1	1	1	← $F_{f,j}$
	1	1	2	← $S_{f,j}$



3 ← $\sum_{j=1}^{n-1} F_{f,j}$

4 ← $\sum_{i=1}^{n-1} S_{f,j}$

Feltételezve a folyamatok párhuzamosságát:

$$E_3 = \sum_{i=1}^{m} D_{i,1} + \sum_{j=1}^{n-1} F_{f,j}$$

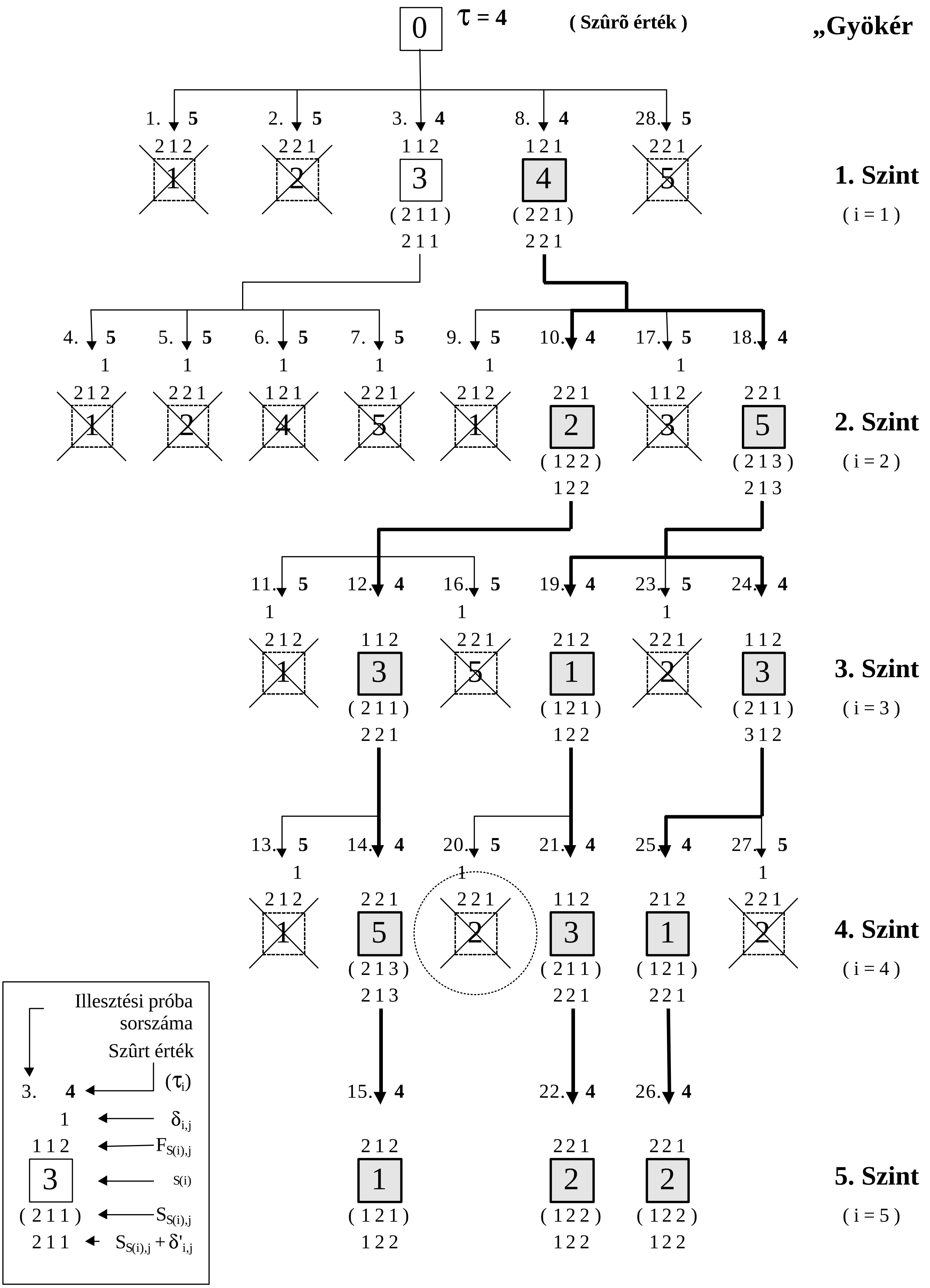
$$E_4 = \sum_{i=1}^{m} D_{i,n} + \sum_{j=1}^{n-1} S_{f,j}$$

Becsült OPTIMUM-érték:

$$E = \max\{ E_1, E_2, E_3, E_4 \}$$

SZÛRÕ: $$\tau = E - \sum_{i=1}^{m} D_{i,1}$$

A KERESÉS (implicit enumeráció)



Hatékonyság vizsgálat

„Teljes leszámlálás”

$$\pi(m) = m! + \sum_{i=1}^{m-1} \frac{m!}{i!}$$

„Hatékonyság”

$$\nu = 100 \cdot \frac{\pi(m) - \pi_S}{\pi(m)}$$

„Expozíció”

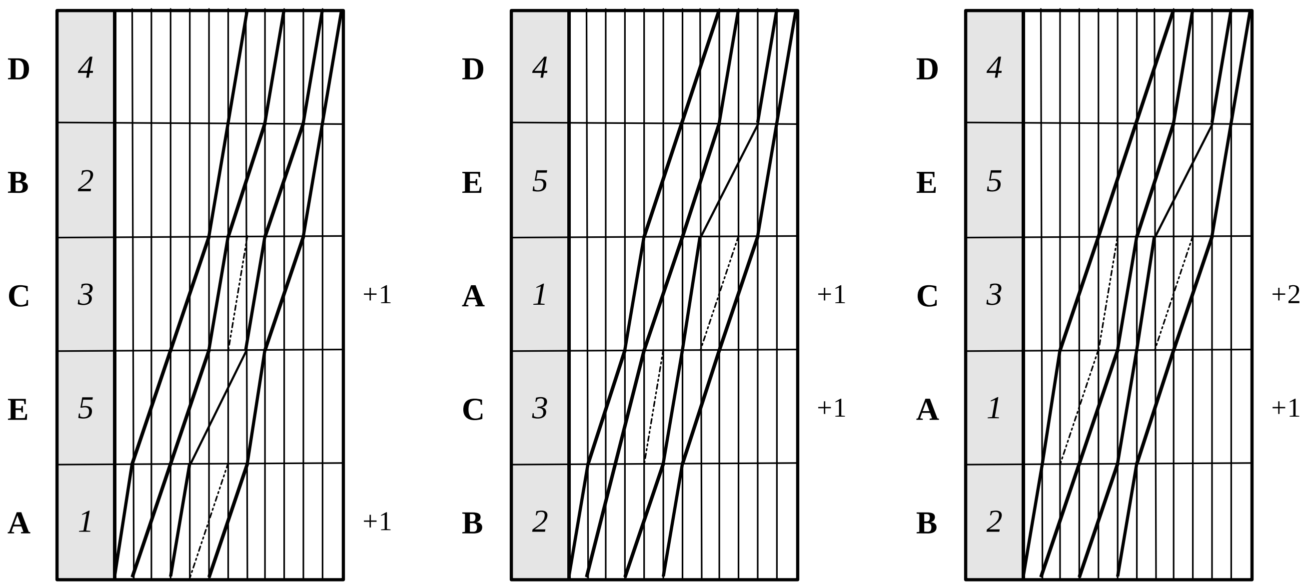
$$\mu = 100 \cdot \frac{\tau^{avr} - \tau^{min}}{\tau^{avr}}$$

„Játéktér”

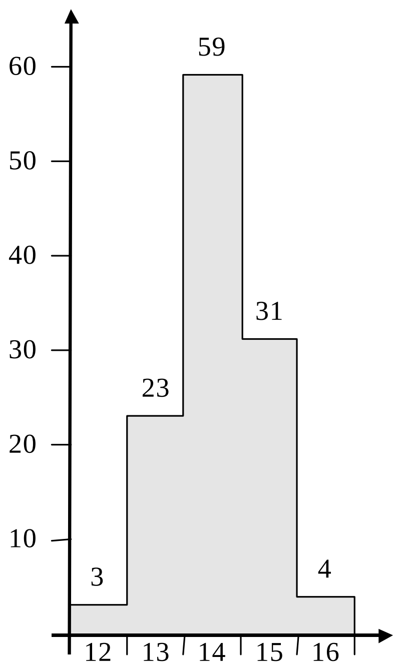
$$\varepsilon = 100 \cdot \frac{\tau^{max} - \tau^{min}}{\tau^{min}}$$

Eredmények értékelése

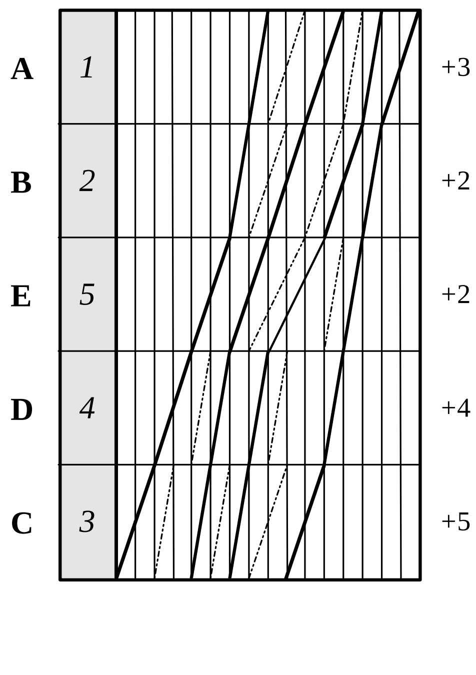
Optimális sorrendek és ütemtervek …



Célértékek eloszlása

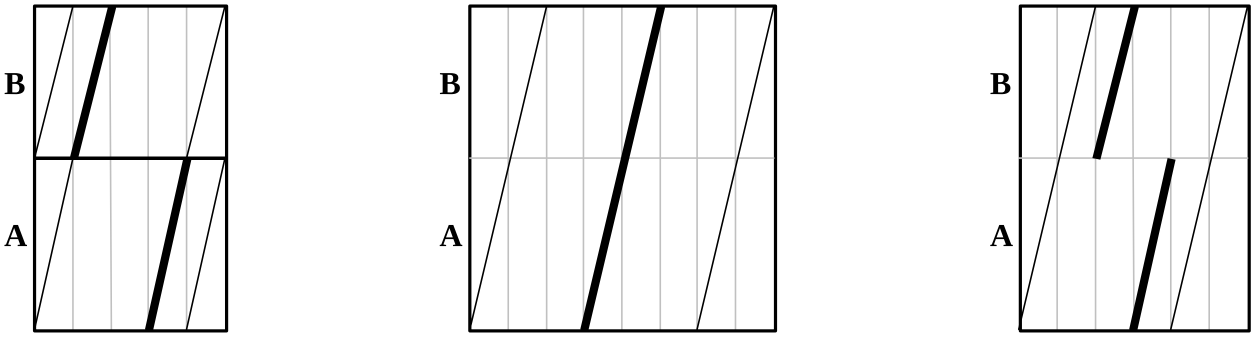


Egy a legkevésbé megfelelõk közül

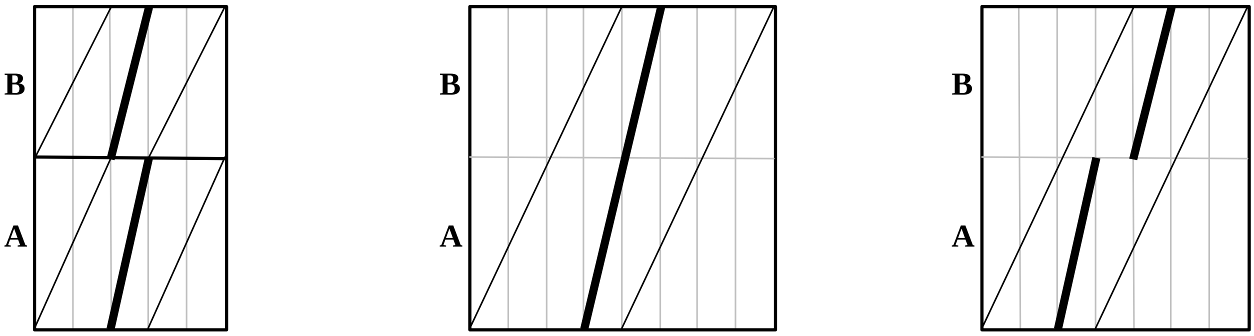


Megszorítások elemzése

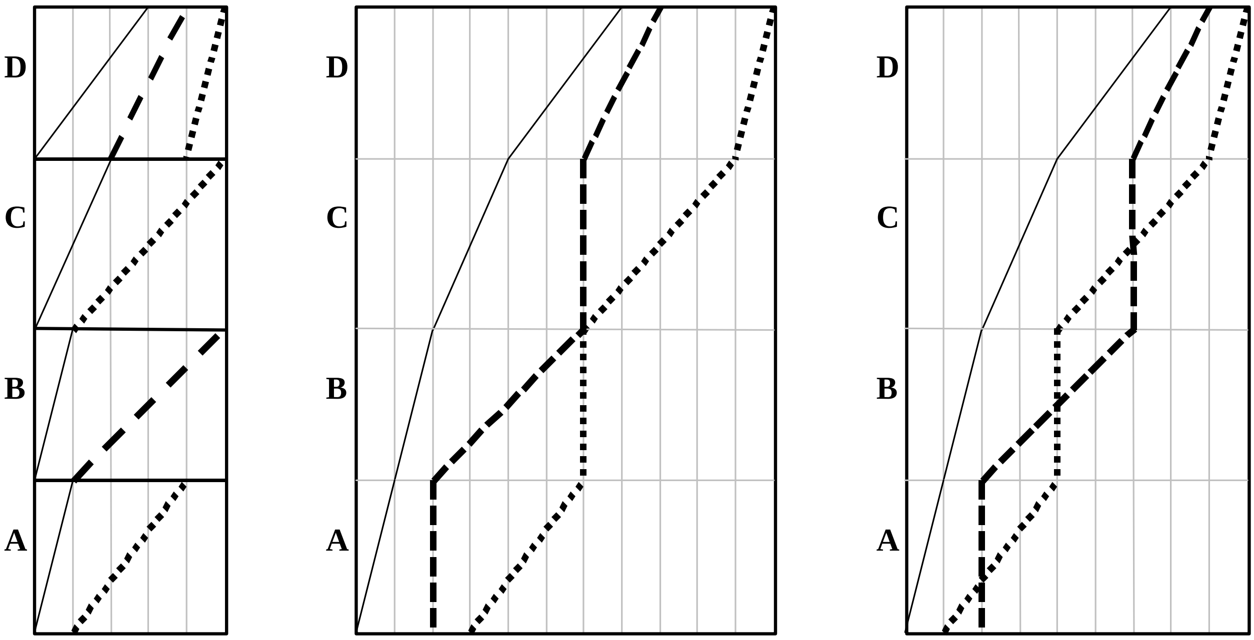
Elõzés nem megengedett …



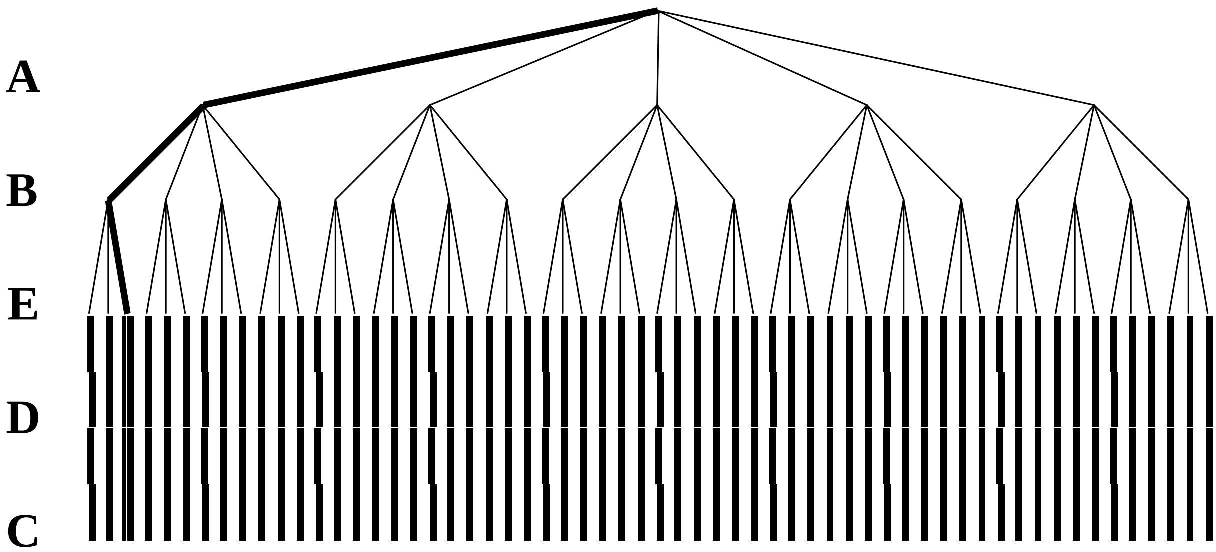
Megszakítás nem megengedett …



Hiányzó munkafolyamat nem megengedett …

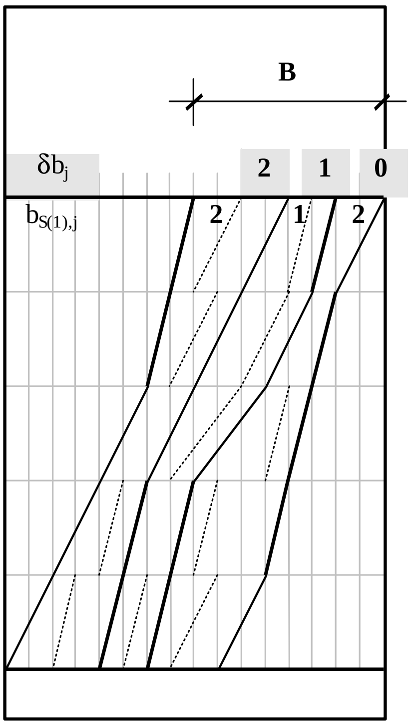


Tetszõleges sorrend-változatú egyesített ütemterv számítása



$a_{1,j} = 0$ $\quad$ $a_{i,j} = (b_{S(i),j} - k_{S(i-1),j})$, $i > 1$ $\quad$ $c_{i,j} = \sum_{p=1}^{i} a_{p,j}$ $\quad$ $db_j = \max_i c_{i,j}$ $\quad$ $B = \sum_{j=1}^{n-1} (b_{S(1),j} + db_j)$

	i	S(i)	*j* = *1*			*2*			*3*		
			$b_{S(i),j}$ / $k_{S(i),j}$	$a_{i,j}$	$c_{i,j}$	$b_{S(i),j}$ / $k_{S(i),j}$	$a_{i,j}$	$c_{i,j}$	$b_{S(i),j}$ / $k_{S(i),j}$	$a_{i,j}$	$c_{i,j}$
A	*1*	1	**2**	0	0	**1**	0	0	**2**	0	0
			1			2			1		
B	*2*	2	2	1	1	2	0	0	1	0	0
			1			2			2		
E	*3*	5	2	1	2	2	0	0	1	-1	-1
			2			1			3		
D	*4*	4	1	-1	1	2	1	1	1	-2	-3
			2			2			1		
C	*5*	3	1	-1	0	1	-1	0	2	1	-2
			2			1			1		
	δb_j				**2**			**1**			**0**



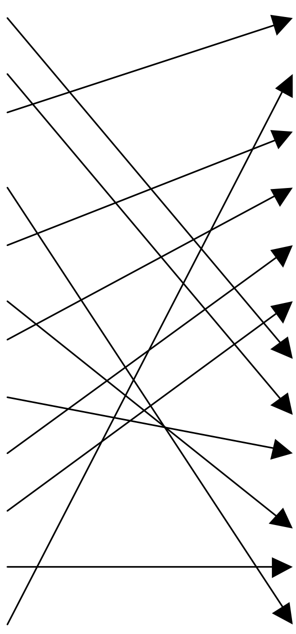
Az F2//C^{max} Feladat

Algoritmus *(Johnson 1954)*

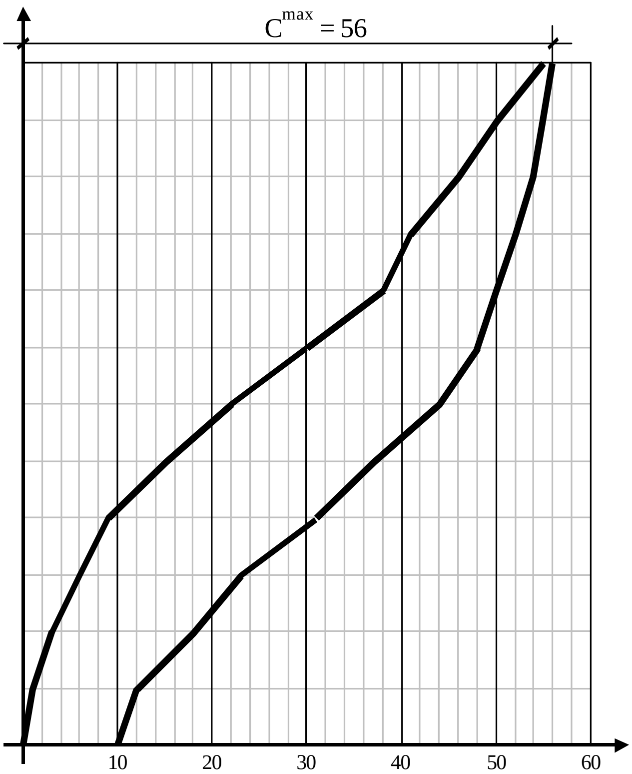
Két vége felõl kialakítva az ütemtervet, a (t_i) tevékenységidõk növekvõ sorrendjében az alábbiak szerint döntsünk a munkadarabok sorrendbeli helyérõl:

1. Ha a vizsgált „t_i” tevékenységidõ az elsõ gépen jelentkezik (t_i=t_1), soroljuk az adott munkadarabot a sorrend elejére (a már sorolt elsõk mögé) !
2. Ha a vizsgált „t_i” tevékenységidõ a második gépen jelentkezik (t_i=t_2), soroljuk az adott munkadarabot a sorrend végére (a már sorolt utolsók elé) !
3. Ha a vizsgált tevékenységidõ „t_i” az elsõ és a második gépen megegyezik (t_i=t_1=t_2), szabadon dönthetünk, hogy {1.} vagy {2.} szerint járunk-e el !

	t_1	t_2
A	8	7
B	6	6
C	5	1
D	1	2
E	5	2
F	3	5
G	3	2
H	3	8
I	8	2
J	7	4
K	2	6
L	4	1



S_i
C
L
E
G
I
J
A
B
H
F
K
D



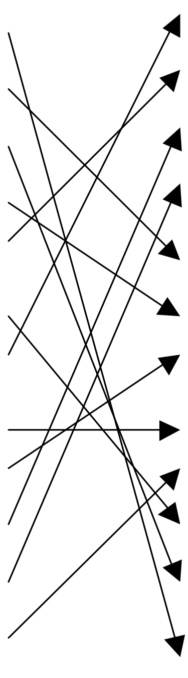
Az F2/overlap/C^{max} Feladat

Algoritmus *(módosított Johnson-algoritmus / Vattai 1993)*

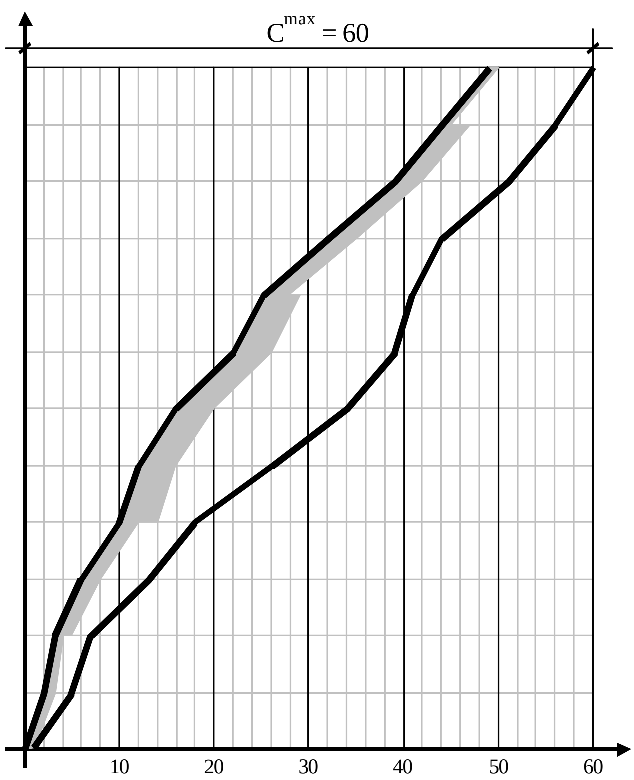
Két vége felõl kialakítva az ütemtervet, a (cr) követési idõk növekvõ sorrendjében az alábbiak szerint döntsünk a munkadarabok sorrendbeli helyérõl:

1. Ha a vizsgált „cr” követési idõ a munkák kezdésénél jelentkezik („V” alakú ütemterv), soroljuk az adott munkadarabot a sorrend elejére (a már sorolt elsõk mögé) !
2. Ha a vizsgált „cr” követési idõ a munkák befejezésénél jelentkezik („A” alakú ütemterv), soroljuk az adott munkadarabot a sorrend végére (a már sorolt utolsók elé) !
3. Ha a vizsgált „cr” követési idõ a munkák kezdésénél és befejezésénél megegyezik („párhuzamos” ütemvonalak), szabadon dönthetünk {1.} vagy {2.} alkalmazásáról !

	t_1	**cr**	t_2	
A	2	**1**	4	*V*
B	3	**4**	2	*A*
C	1	**1**	2	*V*
D	6	**4**	5	*A*
E	5	**3**	5	*//*
F	3	**2**	6	*V*
G	5	**1**	4	*A*
H	2	**4**	8	*V*
I	4	**4**	8	*V*
J	7	**3**	7	*//*
K	7	**3**	3	*A*
L	4	**2**	5	*V*



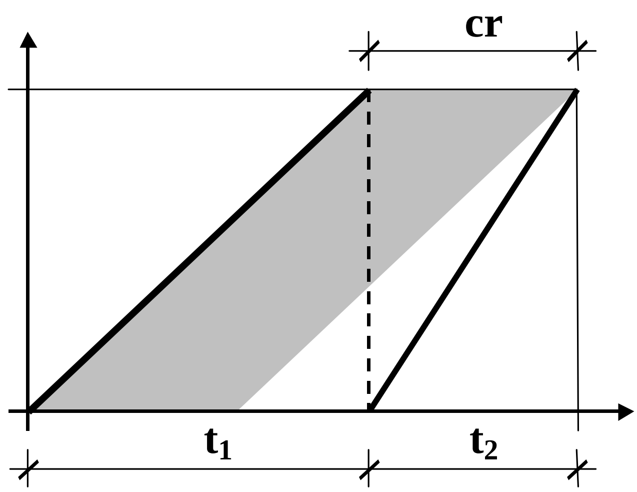
S_i
G
E
J
K
B
D
I
H
L
F
C
A



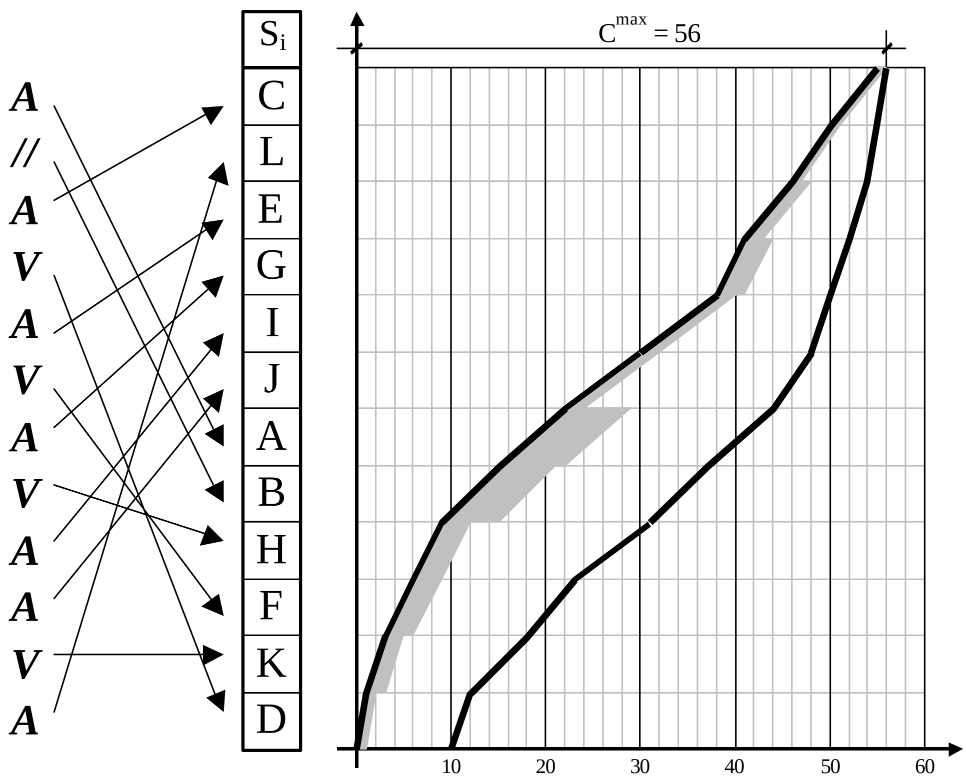
F2//C^{max} Þ F2/overlap/C^{max}

Következmény :

A „cr” követési idõk meghatározásával [cr = *min*{t_1,t_2}]
(„átlapolás nélküli esetek” „átlapoltkénti” kezelésével) minden olyan feladat, mely megoldható Johnson algoritmusával, megoldható a módosított Johnson-algoritmussal is.



	t_1	**cr**	t_2
A	8	**7**	7
B	6	**6**	6
C	5	**1**	1
D	1	**1**	2
E	5	**2**	2
F	3	**3**	5
G	3	**2**	2
H	3	**3**	8
I	8	**2**	2
J	7	**4**	4
K	2	**2**	6
L	4	**1**	1



A módosított Johnson-algoritmussal elõállítható ütemtervek optimalitása

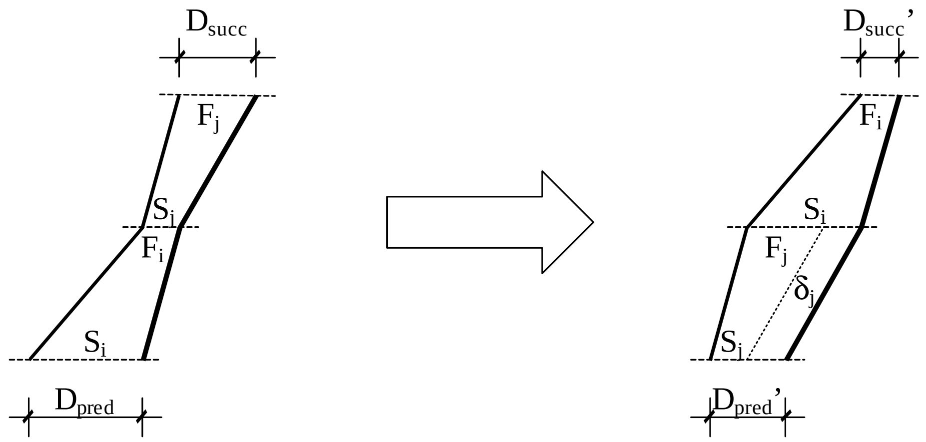
Definíció : Egy ütemterv „kvázi ó-alakú” ha

$F_i \leq S_i$ és $F_j \geq S_j$ | $i < j$

(„V-alakúak” a sorrend elején, „A-alakúak” a sorrend végén)

Állítás : Létezik olyan optimális ütemterv, mely kvázi ó-alakú

Bizonyítás : Tegyük fel, hogy találtunk olyan optimális ütemtervet, mely nem kvázi ó-alakú.
Tegyük kvázi ó-alakúvá … !



Mint látható: $D_{succ}' = F_i = S_j < F_j = D_{succ}$

$D_{pred}' = S_j + \delta_j = S_j + S_i - F_j < S_i = D_{pred}$

Egy nem kvázi ó-alakú ütemtervet kvázi ó-alakúvá téve a teljes átfutási idõ (C^{max}) nem nõtt.

Következtetés : Bármely olyan optimális ütemtervbõl kiindulva, mely nem kvázi ó-alakú, elõ tudunk állítani olyan optimális ütemtervet, mely kvázi ó-alakú.

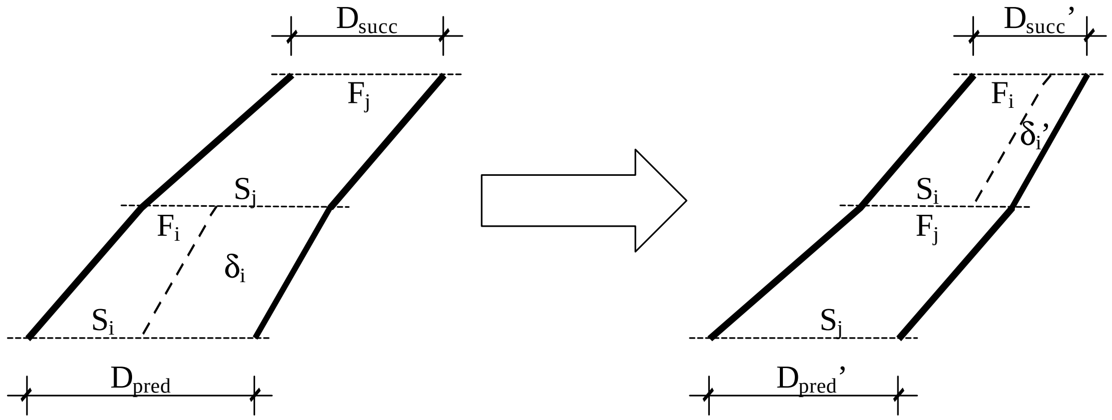
A módosított Johnson-algoritmussal elõállítható ütemtervek optimalitása

Definíció : Jelölje *g* az utolsó olyan munkadarabot egy kvázi ó-alakú ütemtervben, melynek saját ütemterve V-alakú ($S_g<F_g$), valamint jelölje *h* az elsõ olyan munkadarabot ugyanebben az ütemtervben, melynek saját ütemterve A-alakú ($S_h>F_h$). (*A kvázi ó-alakú ütemterv definíciója szerint g<h .*) Egy kvázi ó-alakú ütemterv szigorúan véve is ó-alakú, ha

$$S_i > S_j \quad | \quad i < j \text{ £ } g \qquad \text{és} \qquad F_k < F_l \quad | \quad h \text{ £ } k < l$$

Állítás: Létezik olyan optimális ütemterv, mely szigorúan véve is ó-alakú.

Bizonyítás: Tegyüket fel, hogy találtunk olyan kvázi ó-alakú optimális ütemtervet, mely szigorúan véve nem ó-alakú. Tegyük szigorúan véve is ó-alakúvá … !



Mint látható: $D_{succ}’ = F_i + \delta_i’ = F_i + F_j - S_i < F_j = D_{succ}$

(például) $D_{pred}’ = S_j = F_i + \delta_i = F_i + S_j - F_i \leq S_i + S_j - F_i = D_{pred}$

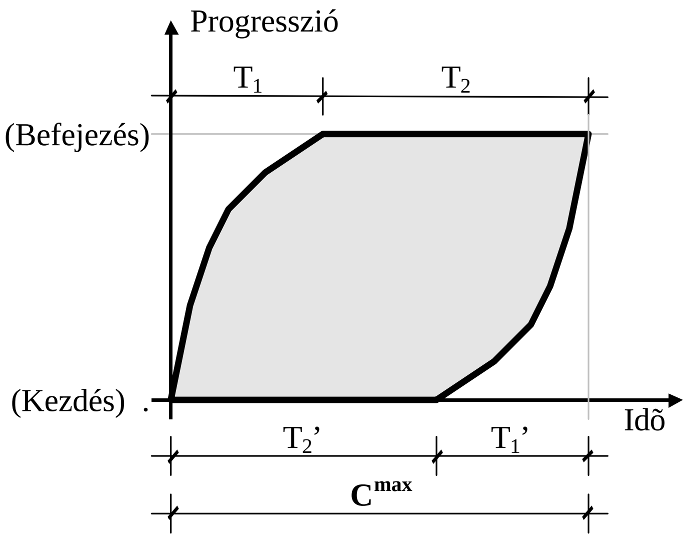
Egy kvázi ó-alakú, de szigorúan véve nem ó-alakú optimális ütemtervet szigorúan véve is ó-alakúvá téve a teljes átfutási idõ (C^{max}) nem nõtt. (Hasonló logika alkalmazható más esetekre is.)

Következtetés: Bármely kvázi ó-alakú optimális ütemtervbõl kiindulva, mely szigorúan véve nem ó-alakú, elõ tudunk állítani olyan optimális ütemtervet, mely szigorúan véve is ó-alakú.

A módosított Johnson-algoritmussal elõállítható ütemtervek optimalitása

Állítás : Ha egy ütemterv szigorúan véve is ó-alakú, akkor bizonyosan optimális is.

Bizonyítás :



$C^{max} = T_1 + T_2 = T_1' + T_2'$

$T_1 = \Sigma\, t_{i,1} = \text{const}$ és $T_1' = \Sigma\, t_{i,2} = \text{const}$

$C^{max} = \min \;\big|\; T_2 = \min$ és $T_2' = \min$

Lásd a szigorúan véve is ó-alakú ütemtervek definícióját …

Felismerés : Mind Johnson algoritmusa, mind a módosított Johnson-algoritmus szigorúan véve is ó-alakú ütemtervet állít elõ.

Megjegyzés : A tevékenységek meg-nem-szakíthatóságának elõírása az F2//C^{max} és F2/overlap/C^{max} feladatoknál irreleváns.