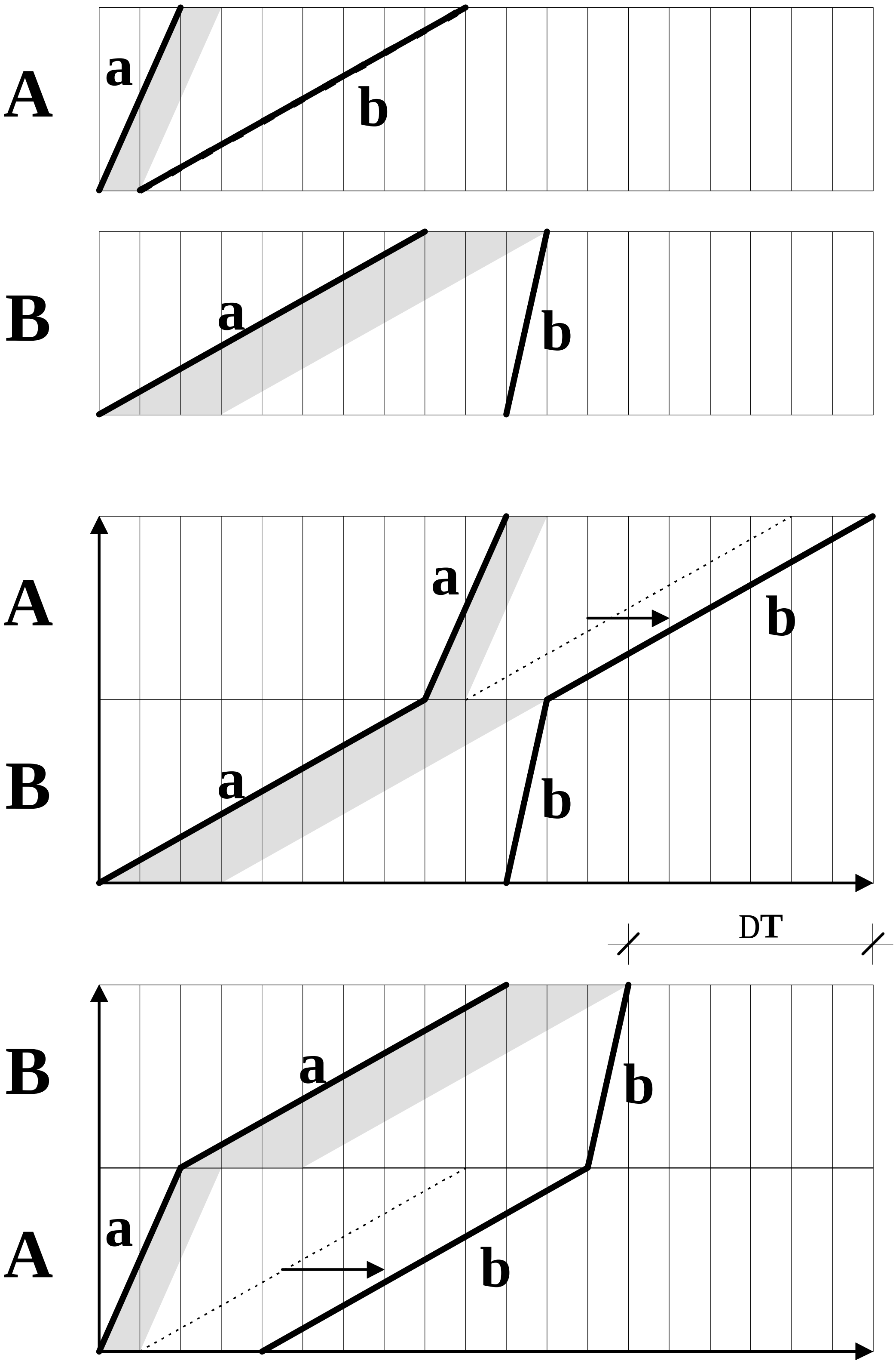
S C H E D U L I N G

Sequencing Projects

(Multi-Project Management)



The F½overlap½C^{max} Problem

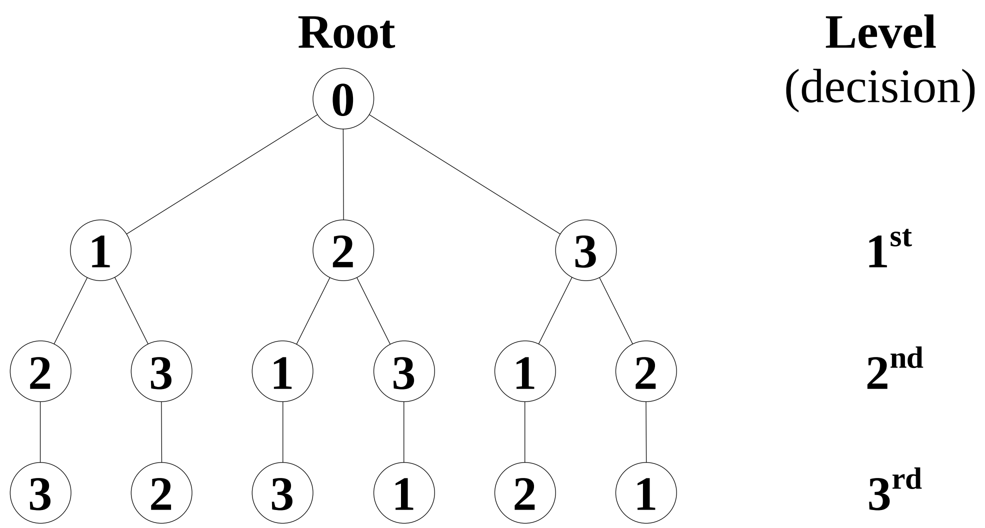
Sequencing on parallel machines „Flow-Shop Problem”

ID: Graham, Lenstra, Rinnooy Kan, 1979

Assumptions:

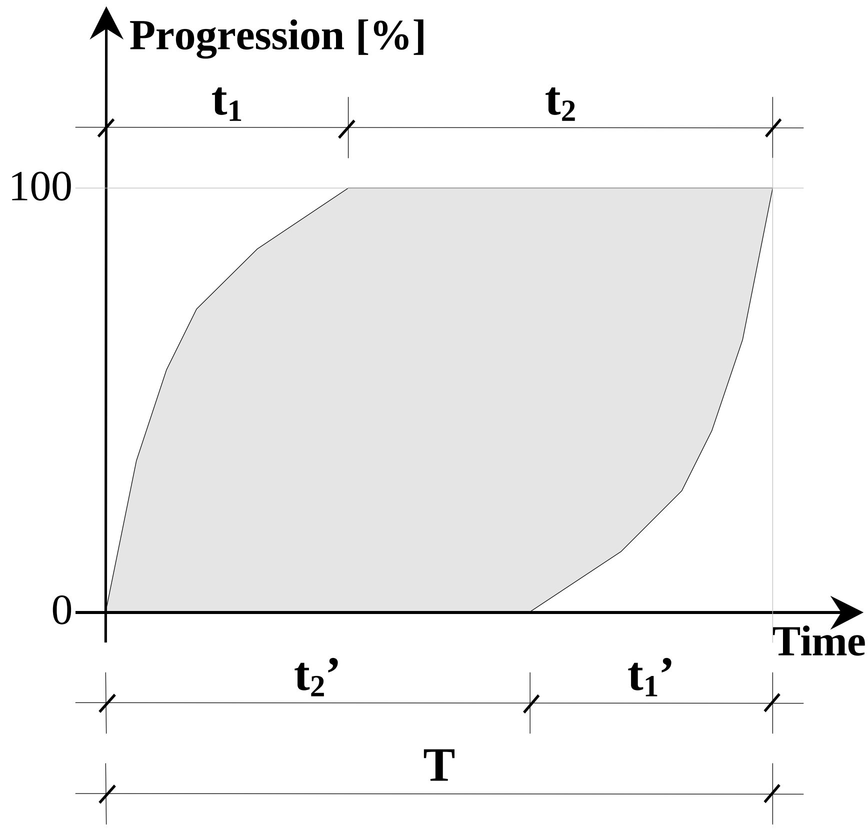
- **Each work (activity) should be performed on each piece (project) in a preset technological order –** „flow-shop”
- **Each machine (group) performs its only single (specialized) work (activity) on each building**
- **Each work (activity) is performed by its only (specialized) machine (group)**
- **Sequence of pieces (projects) must be the same for each machine (group) –** „no passing allowed”
- **Each machine (group) should work with no break –** „pre-emption not allowed”
- **Overlapping performance in time on a piece (project) allowed –** „ovelapping allowed”
- **The aim is to minimize the overall complition time –** „complition time to be minimized”

A Decision Tree for Sequencing



123 < 132 < 213 < 231 < 312 < 321

Constant and varying segments of the OVERALL EXECUTION TIME



$$T = t_1 + t_2$$

$$T = t_1' + t_2'$$

$$t_1 = \text{const}$$

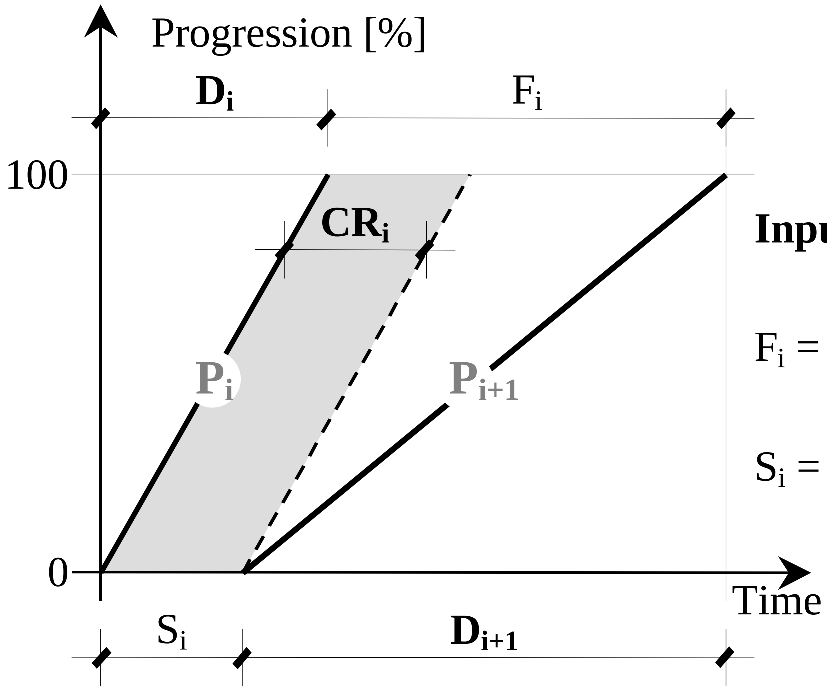
$$t_1' = \text{const}$$

$$T_{min} \text{ Þ } t_{2\ min}$$

$$T_{min} \text{ Þ } t_2'_{min}$$

Calculating succession times

Overlapping allowed
technological break preset

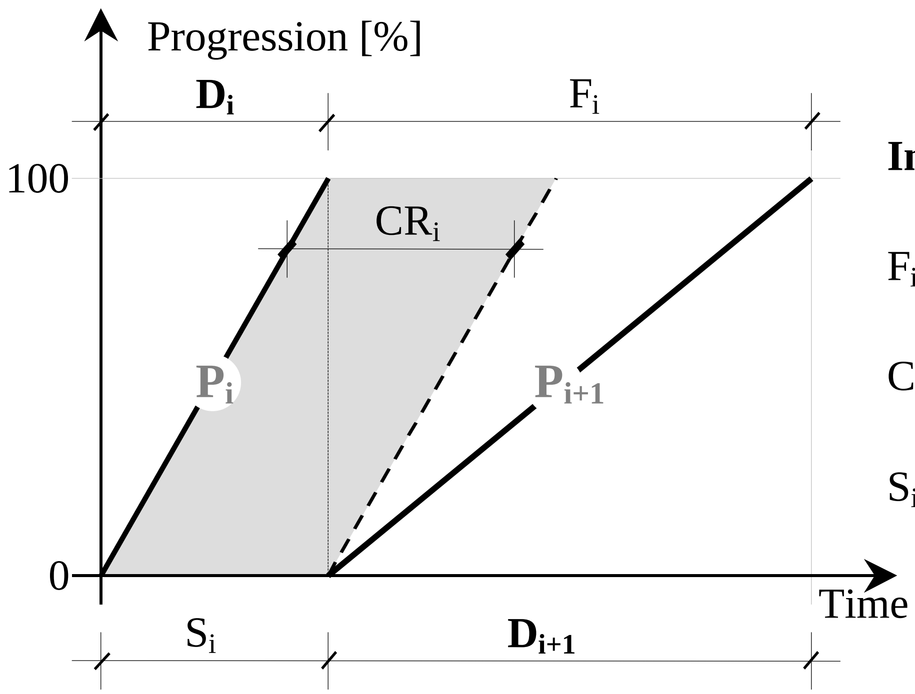


Input data : D_i , D_{i+1} , CR_i

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

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

With no overlapping



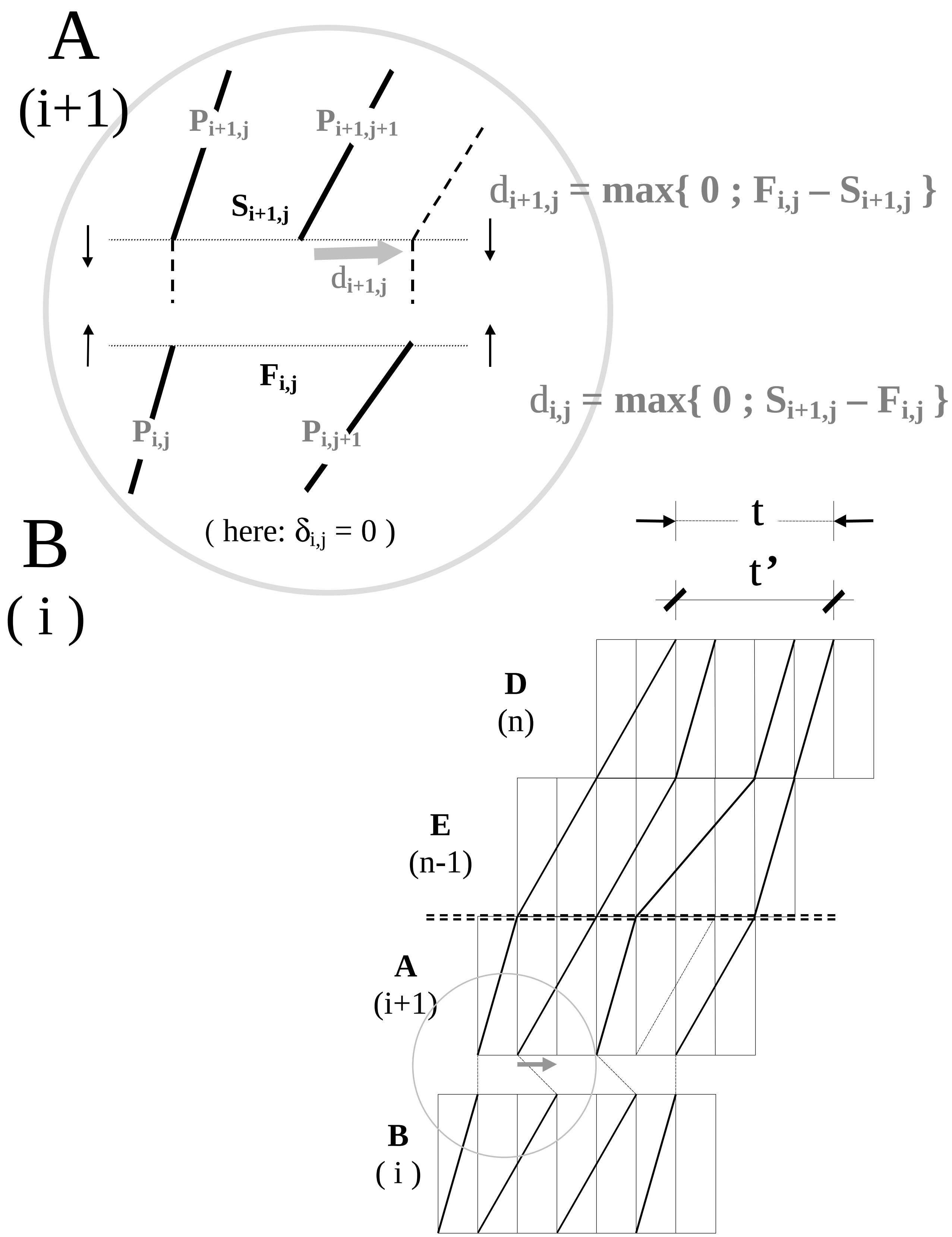
Input data : D_i , D_{i+1}

$F_i = D_{i+1}$

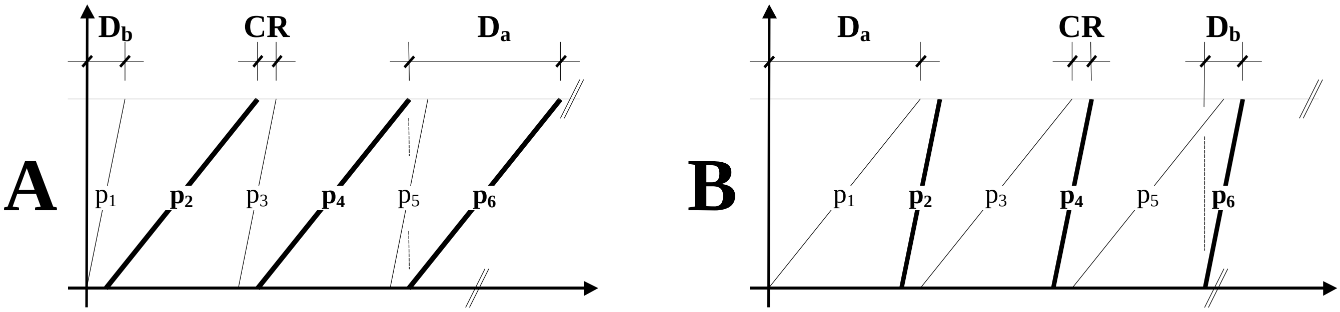
$CR_i = \min\{ D_i ; D_{i+1} \}$

$S_i = D_i$

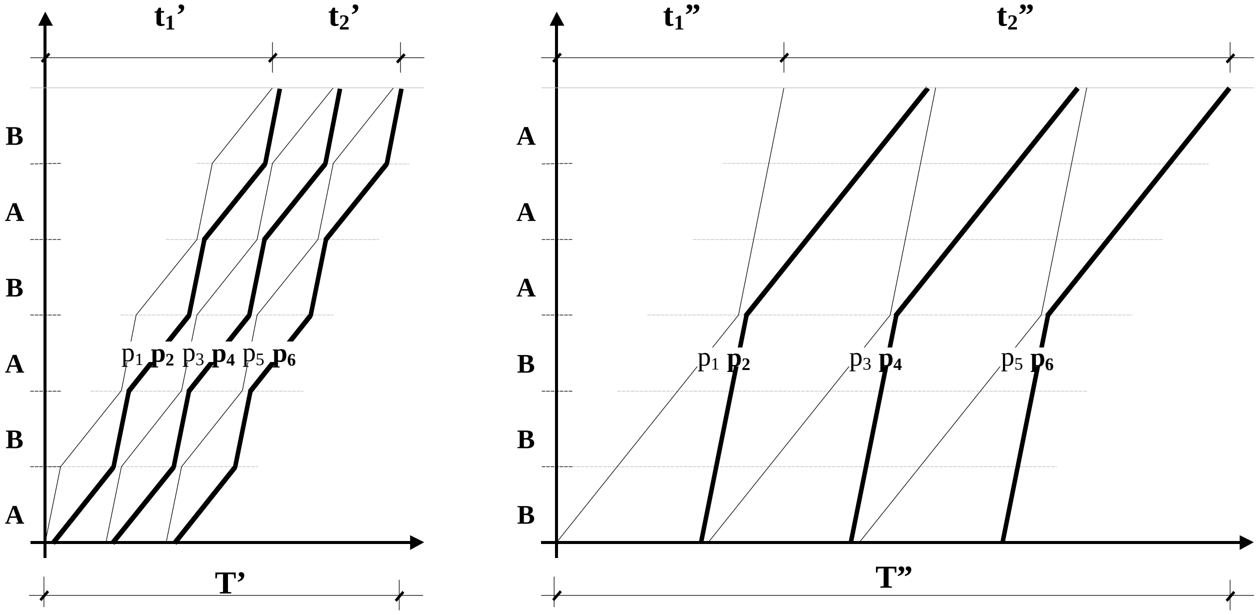
Forming a Master Schedule



Theoretical effect of Sequence



$D_a >> D_b, CR$



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

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

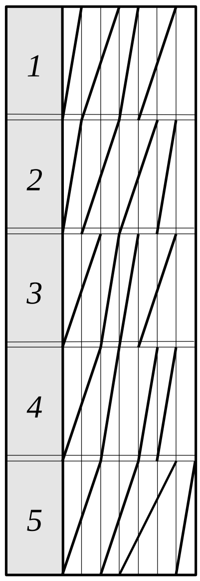
$$\lim_{n\to\infty}\frac{T'}{T''} \approx \lim_{n\to\infty}\frac{m+n-1}{m\cdot(n+1)} = \lim_{n\to\infty}\frac{m}{m\cdot(n+1)} + \lim_{n\to\infty}\frac{n}{m\cdot(n+1)} - \lim_{n\to\infty}\frac{1}{m\cdot(n+1)} = 0 + \frac{1}{m} - 0$$

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

SETTING THE FILTER VALUE
(Estimating the optimum)

	i \ j	1 (a)	2 (b)	3 (c)	4 (d)	
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		← $\min_i\{CR_{i,j}\}$

	i \ j	1 (a)	2 (b)	3 (c)	
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	



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

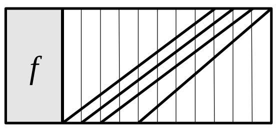
Assuming the last/first project unreleased :

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

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

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

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



3 ← $\sum_{j=1}^{n-1} F_{f,j}$

4 ← $\sum_{j=1}^{n-1} S_{f,j}$

Assuming no inner coincidences :

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

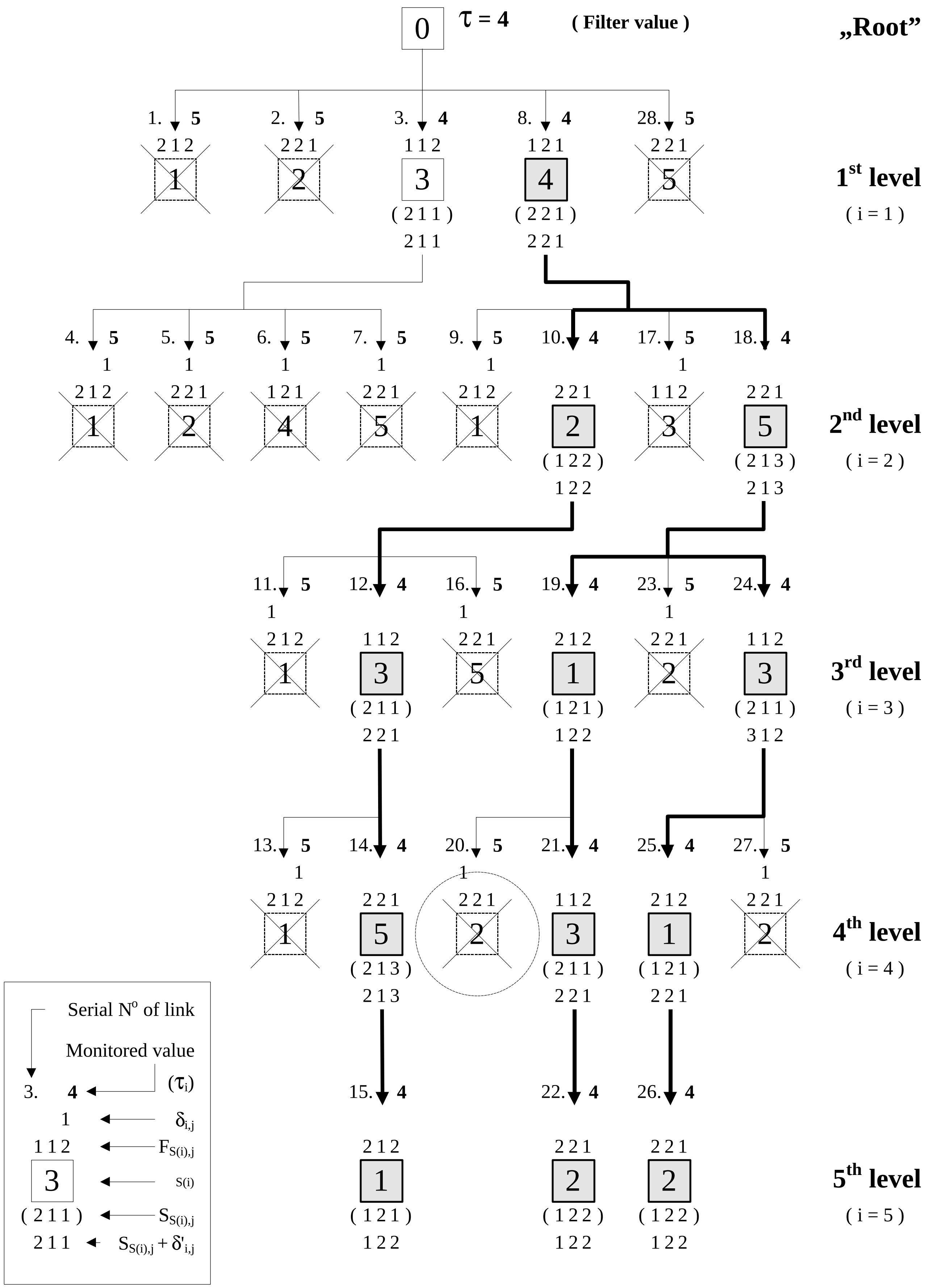
$$E_4 = \sum_{i=1}^{m} D_{i,n} + \sum_{j=1}^{n-1} S_{f,j}$$

Estimated OPTIMUM:

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

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

THE SEARCH (implicite enumeration)



Evaluating solution

„Enumeration”

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

„Effectivity”

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

„Exposition”

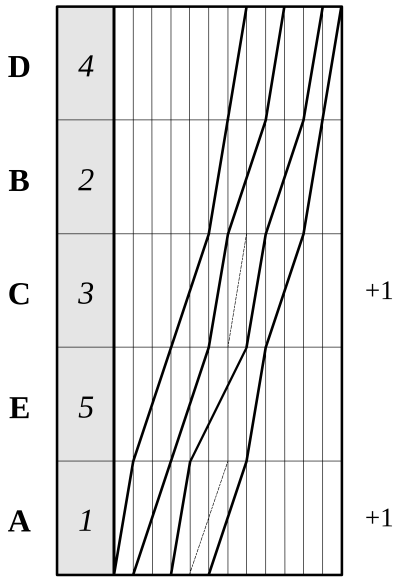
$$\mu = 100 \cdot \frac{\tau^{avr} - \tau^{min}}{\tau^{avr}}$$

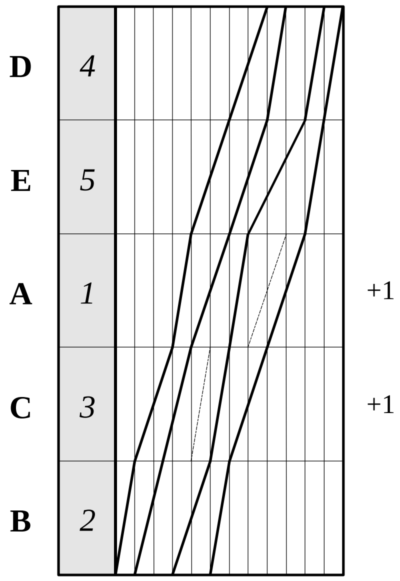
„Gap”

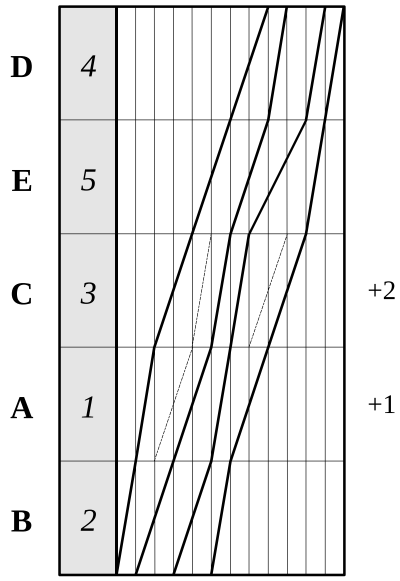
$$\varepsilon = 100 \cdot \frac{\tau^{max} - \tau^{min}}{\tau^{min}}$$

Evaluating results

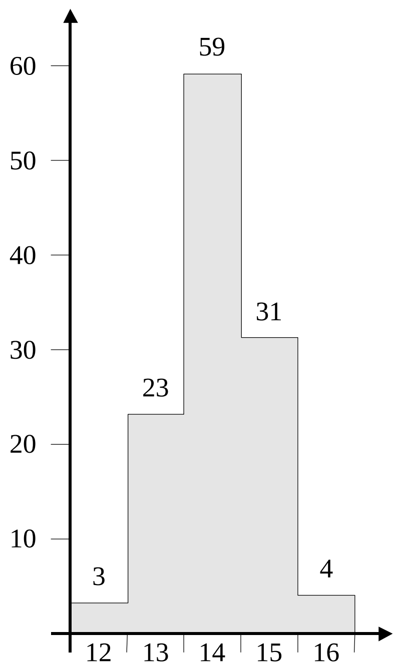
Optimal sequences and schedules …



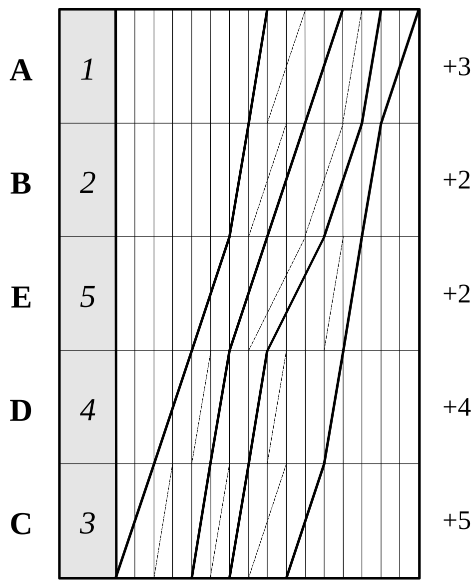




Distribution of results

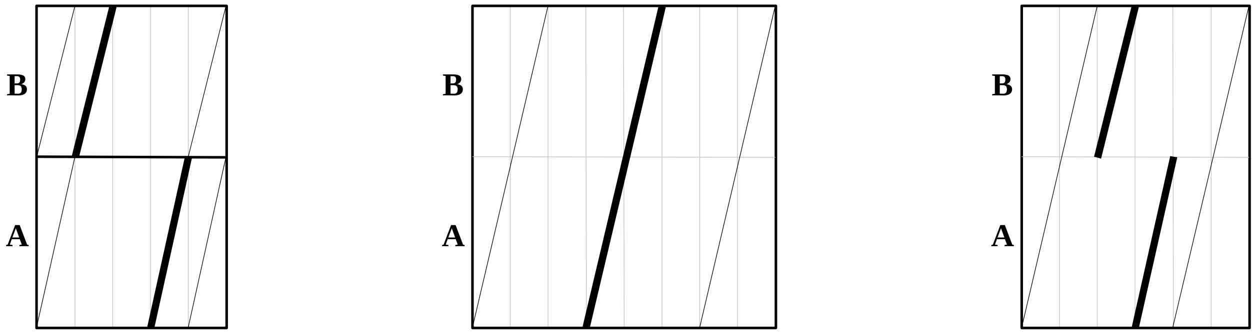


One of the most unfavorables

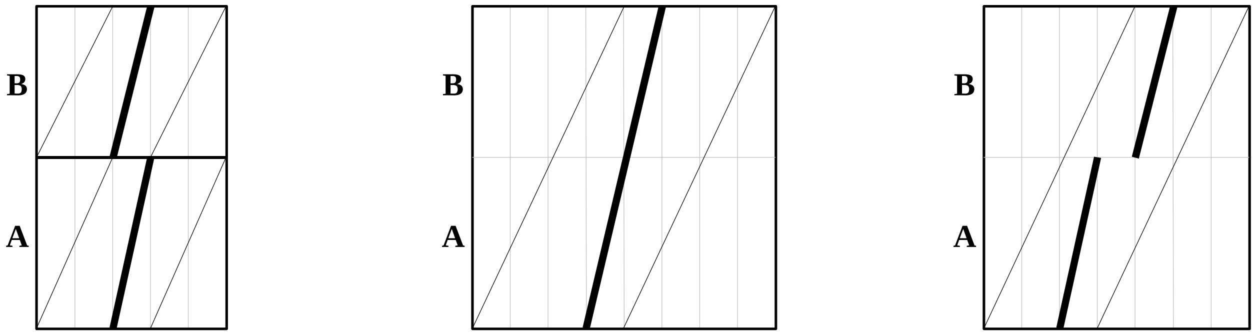


Analysing some restrictions

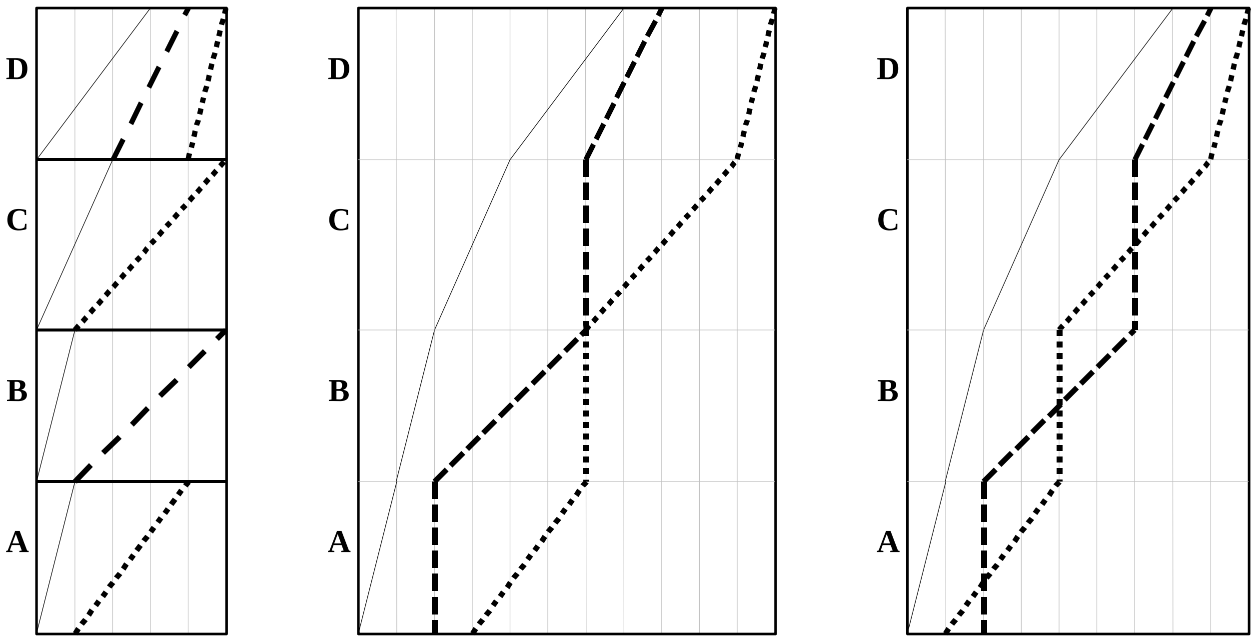
No passing …



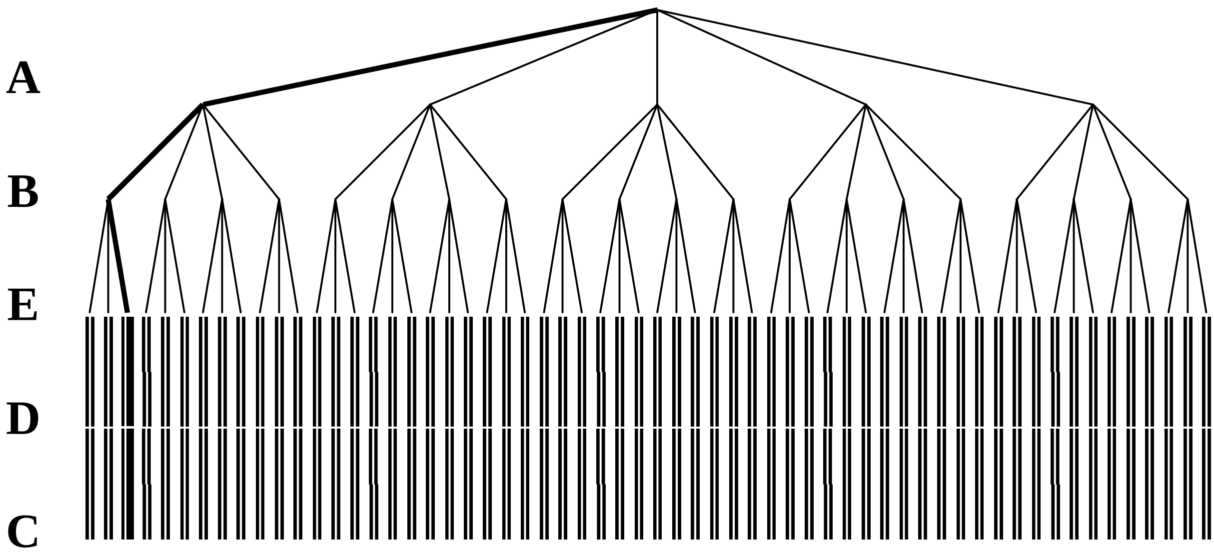
No idle-times …



No missing processes …

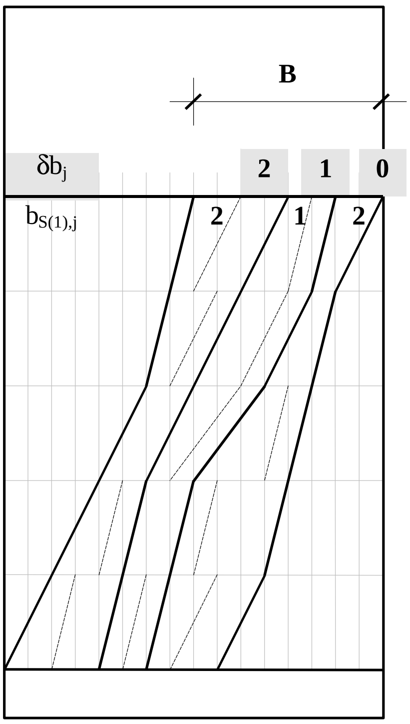


Calculating a single permutation



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

	i	S(i)	*j* = *1*: $b_{S(i),j}$ / $k_{S(i),j}$	$a_{i,j}$	$c_{i,j}$	*2*: $b_{S(i),j}$ / $k_{S(i),j}$	$a_{i,j}$	$c_{i,j}$	*3*: $b_{S(i),j}$ / $k_{S(i),j}$	$a_{i,j}$	$c_{i,j}$
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**



The F2//C^{max} Problem

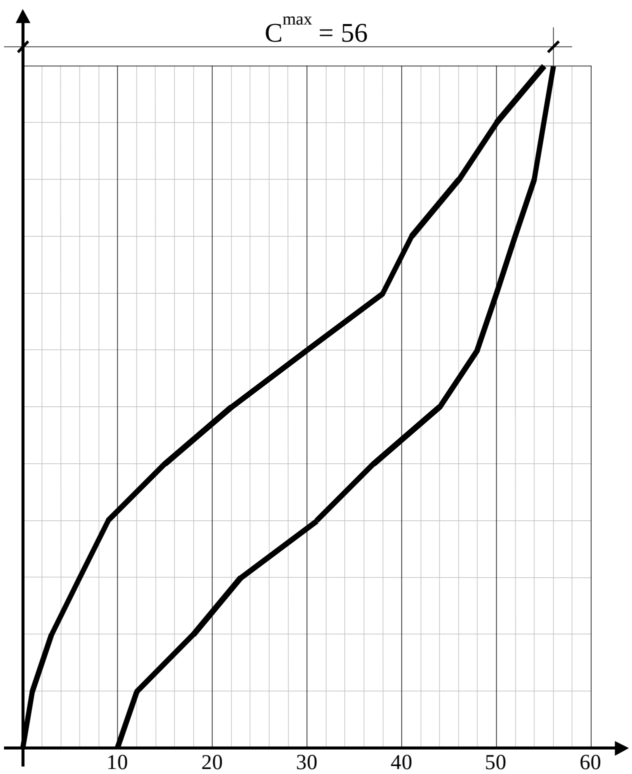
Algorithm *(Johnson 1954)*

Constructing the schedule from both ends, considering production time values (t_i) in ascending sequence, do order the pieces as listed below :

1. If „t_i" value consedered appeared on the first machine (t_i=t_1) do order the piece to the beginning of the schedule (after the already scheduled ones) !
2. If „t_i" value considered appeared on the second machine (t_i=t_2) do order the piece to the end of the schedule (before the already scheduled ones) !
3. If „t_i" value considered appeared both on the first and on the second machine (t_i=t_1=t_2) you are free to chose either {1.} or {2.} !

	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



The F2/overlap/C^{max} Problem

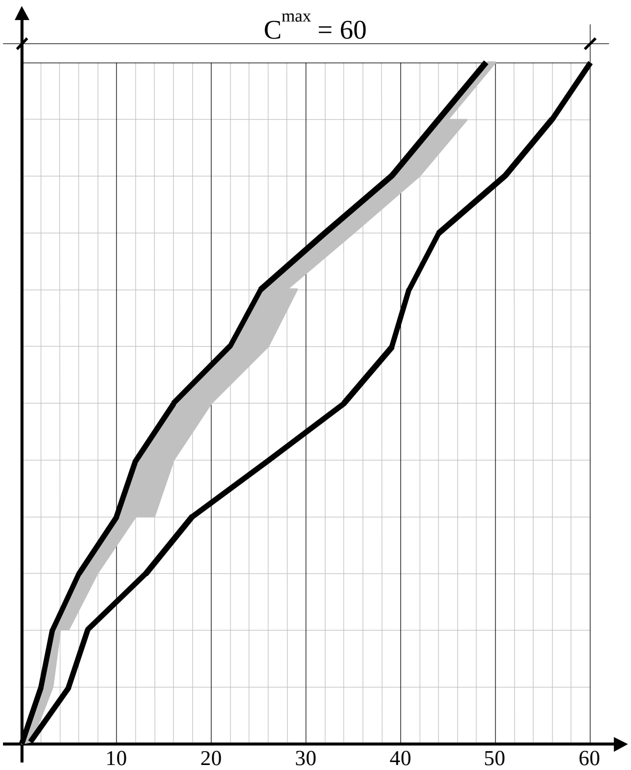
Algorithm *(modified Johnson-algorithm / Vattai 1993)*

Constructing the schedule from both ends, considering minimum succession time values (cr) in ascending sequence, do order the pieces as listed below :

1. If „cr" value consedered appeared at start of the piece („V" shaped progression) do order the piece to the beginning of the schedule (after the already scheduled ones) !
2. If „cr" value considered appeared at finish of the piece („A" shaped progression) do order the piece to the end of the schedule (before the already scheduled ones) !
3. If „cr" value considered appeared both at start and at finish of the piece („parallel" progression) you are free to chose either {1.} or {2.} !

	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*

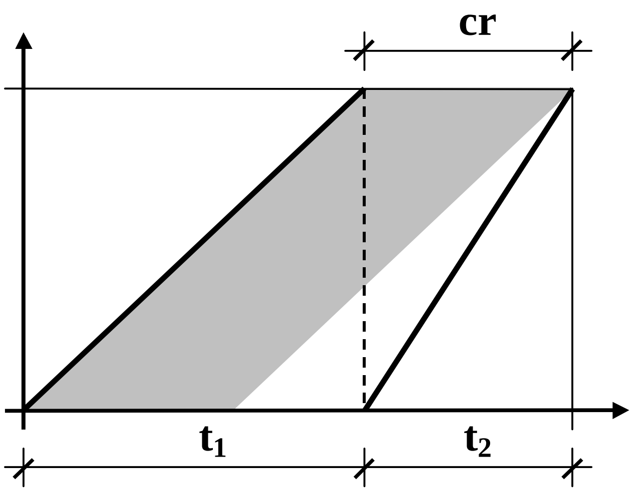
S_i
G
E
J
K
B
D
I
H
L
F
C
A



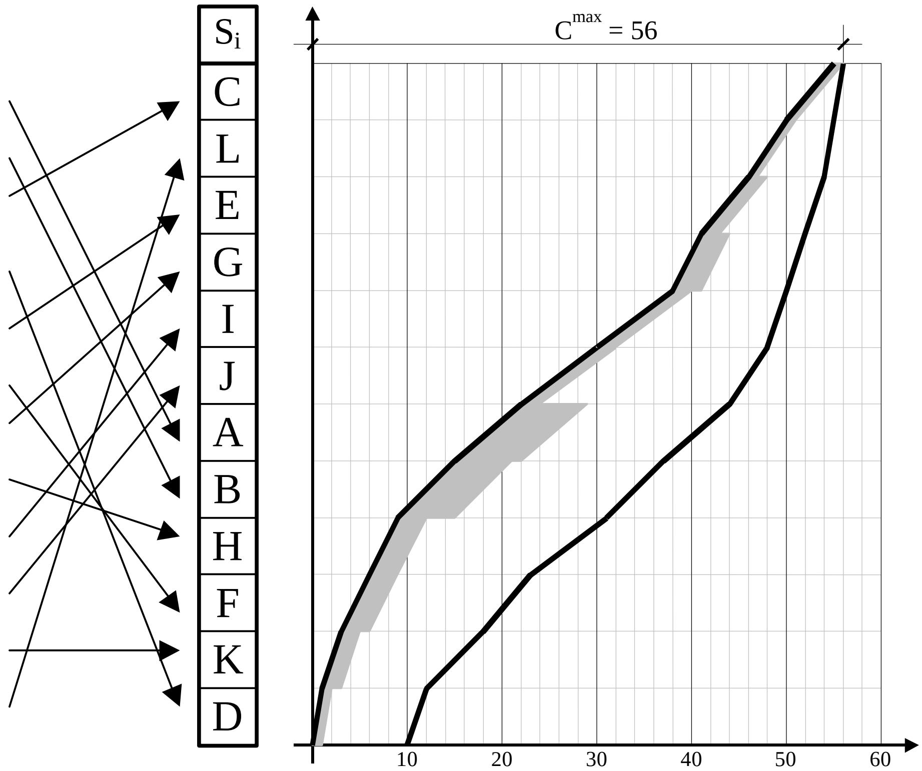
F2//C^{max} Þ F2/overlap/C^{max}

Conclusion :

Defining „cr" values for each piece as [cr = *min*{t_1,t_2}] („consecutive-" to „overlapped processing") any problem that can be solved by Johnson's algorithm can be solved by modified Johnson-algorithm too.



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



Proofing optimality of schedules constructed by modified Johnson-algorithm

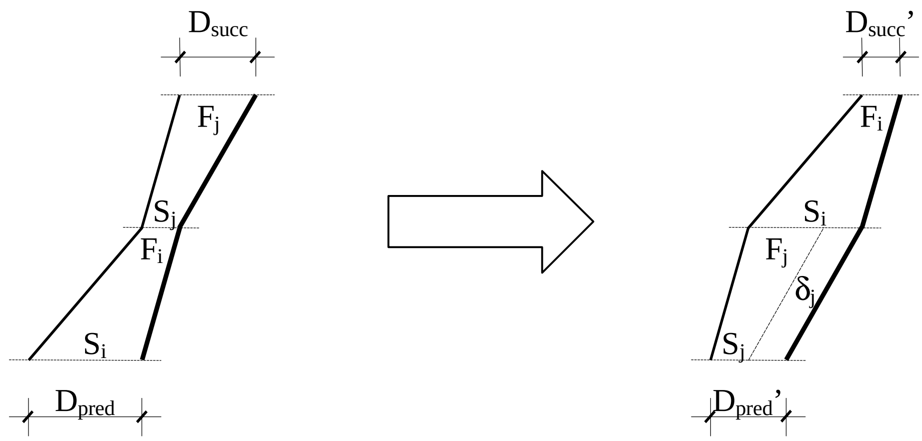
Definition : A schedule is callad „quasi o-shaped" if

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

(„V-shaped" progression at start, „A-shaped" progression at end)

Theorem : There exists at least one optimal schedule that is quasi o-shaped

Proof : Let assume we found an optimal schedule that is not quasi o-shaped. Make it quasi o-shaped ! …



We find: $D_{succ}' = F_i = S_j < F_j = D_{succ}$

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

After transforming an optimal schedule that had not been quasi o-shaped into a quasi o-shaped one completion time (C^{max}) did not increase.

Conclusion : Originating from any (optimal) schedule that is not quasi o-shaped we can construct an other (optimal) schedule that is quasi o-shaped.

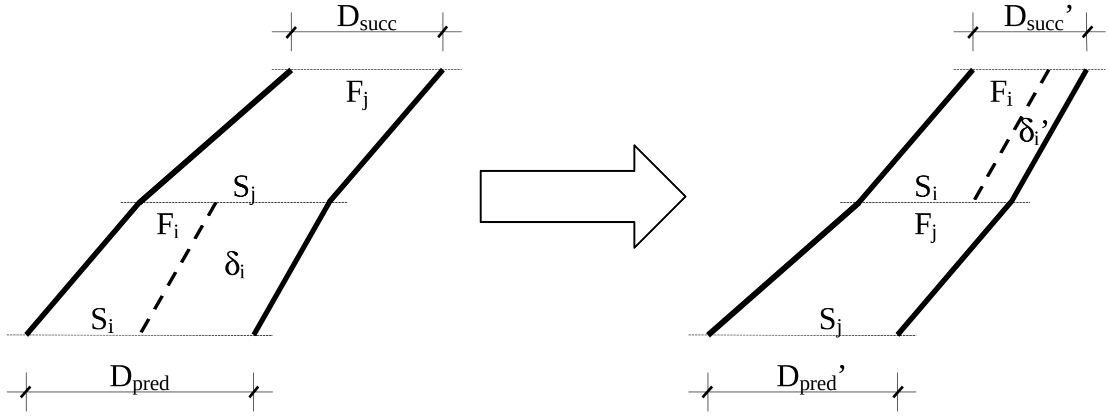
Proofing optimality of schedules constructed by modified Johnson-algorithm

Definition : Let *g* indicate the last piece with V-shaped progression ($S_g<F_g$) in a quasi o-shaped schedule. Also let *h* indicate the first piece with A-shaped progression ($S_h>F_h$) in the same quasi o-shaped schedule. (*By definition of quasi o-shaped schedule g<h .*) A schedule is „strictly o-shaped" if

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

Theorem : There exists at least one optimal schedule that is strictly o-shaped

Proof : Let assume we found a quasi o-shaped optimal schedule that is not strictly o-shaped. Make it strictly o-shaped ! …



We find (e.g.): $D_{succ}' = F_i + \delta_i' = F_i + F_j - S_i < F_j = D_{succ}$

$D_{pred}' = S_j = F_i + \delta_i = F_i + S_j - F_i \leq S_i + S_j - F_i = D_{pred}$

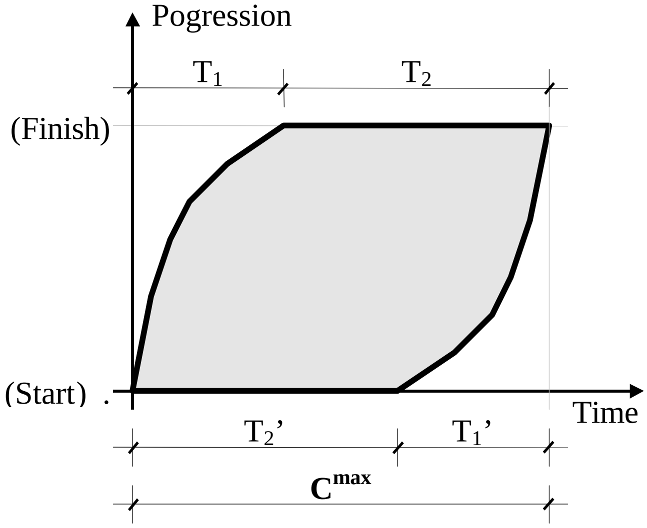
After transforming a quasi o-shaped optimal schedule that had not been strictly o-shaped into a strictly o-shaped one completion time (C^{max}) did not increase. (Use the same logic for any other cases)

Conclusion : Originating from any quasi o-shaped (optimal) schedule that is not strictly o-shaped we can construct an other (optimal) schedule that is strictly o-shaped.

Proofing optimality of schedules constructed by modified Johnson-algorithm

Theorem : If a schedule is strictly o-shaped than it is surely optimal too.

Proof :



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

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

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

See definition of strictly o-shaped schedule …

Recognition : Using Johson's algorithm or modified Johnson-algorithm we make a strictly o-shaped schedule.

Remark : The condition if any of „pre-emption allowed" (*machines need not work with no break*) is irrelevant at $F2//C^{max}$ and $F2/overlap/C^{max}$ problems.