

## Programando n tarefas em 3 máquinas (Regra de Johnson)

São consideradas 3 máquinas com ordem de sequência técnica das tarefas obrigatória, começando em M1, depois em M2 e por último em M3. Isto é, as tarefas tem que serem processadas na mesma ordem nas três máquinas.

As condições para este algoritmo ser aplicado a este tipo de problema são:

$$\min t_{i1} \geq \max t_{i2} \quad \text{ou} \quad \min t_{i3} \geq \max t_{i2}$$

Para resolver o problema, as três máquinas serão vistas como duas máquinas artificiais (M'1 e M'2). Os novos tempos das tarefas nestas máquinas (artificiais) serão: em M'1  $t_{i1} + t_{i2}$  e em M'2  $t_{i2} + t_{i3}$ .

Feito isto, aplica-se normalmente o método de Johnson para  $n \times 2$  (para M'1 e M'2).

Exemplo: Encontre a sequência ótima para as seis tarefas listadas a seguir, a serem processadas em M1, M2 e M3.

Tarefa	Tempos das tarefas		
	M1	M2	M3
1	5	3	9
2	7	2	5
3	4	3	7
4	8	4	3
5	6	2	2
6	7	0	8

$$\min t_{i1} = \min \{t_{11}, t_{21}, t_{31}, t_{41}, t_{51}, t_{61}\} = 4$$

$$\max t_{i2} = \max \{t_{12}, t_{22}, t_{32}, t_{42}, t_{52}, t_{62}\} = 4$$

A condição para ser aplicado este algoritmo foi satisfeita. Então cria-se máquinas artificiais, obtendo-se também os novos tempos para estas:

Tarefa	M'1	M'2
1	8	12
2	9	7
3	7	10
4	12	7
5	8	4
6	7	8

As sequências possíveis são:

3-6-1-2-4-5, 6-3-1-2-4-5, 3-6-1-4-2-5, 6-3-1-4-2-5