Regra de Johnson

Regra de Johnson

❖ Outra regra de sequenciamento é a heurística conhecida como "Regra de Johson", que minimiza o *lead time* total de um conjunto de ordens processadas em dois recursos sucessivos (máquina A e máquina B) desde que as seguintes condições sejam satisfeitas:

a) Os tempos de processamento das ordens (incluindo os *setups*) devem ser conhecidos e constantes, bem como independentes da sequência de processamento escolhida;

b) Todas as ordens são processadas na mesma direção, da máquina A para a B;

c) Não existem prioridades; e,

d) As ordens são transferidas de uma máquina para outra apenas quando completadas.

Observação: Johnson sugere no caso de empate fazer uso do índice ou posição da tarefa para o desempate, ou seja, a tarefa posicionada na lista de tarefas próximo do início da lista de tarefas deve ser alocada em primeiro lugar.

Regra de Johnson

- Uma vez cumpridas essas condições, a determinação da sequência pela regra de Johnson segue os seguintes passos:

1) Selecione o menor tempo entre todos os tempos de processamento da lista de ordens a serem processadas nas máquinas A e B (no caso de empate escolha qualquer uma).
2) Se o tempo escolhido for na máquina A, programe esta ordem no início. Se o tempo escolhido for na B, programe esta ordem para o final.
3) Elimine a ordem escolhida da lista de ordens a serem processadas e retorne ao 1º passo até programar todas as ordens.

Exemplo:

Cinco ordens de fabricação precisam ser estampadas na máquina A e, em seguida, usinadas na máquina B. Os tempos de processamento, incluindo *setups*, e as prioridades atribuídas são apresentadas na Tabela. Para aplicação da regra PEPS, consideraremos que as ordens deram entradas em carteira na sequência de sua numeração.

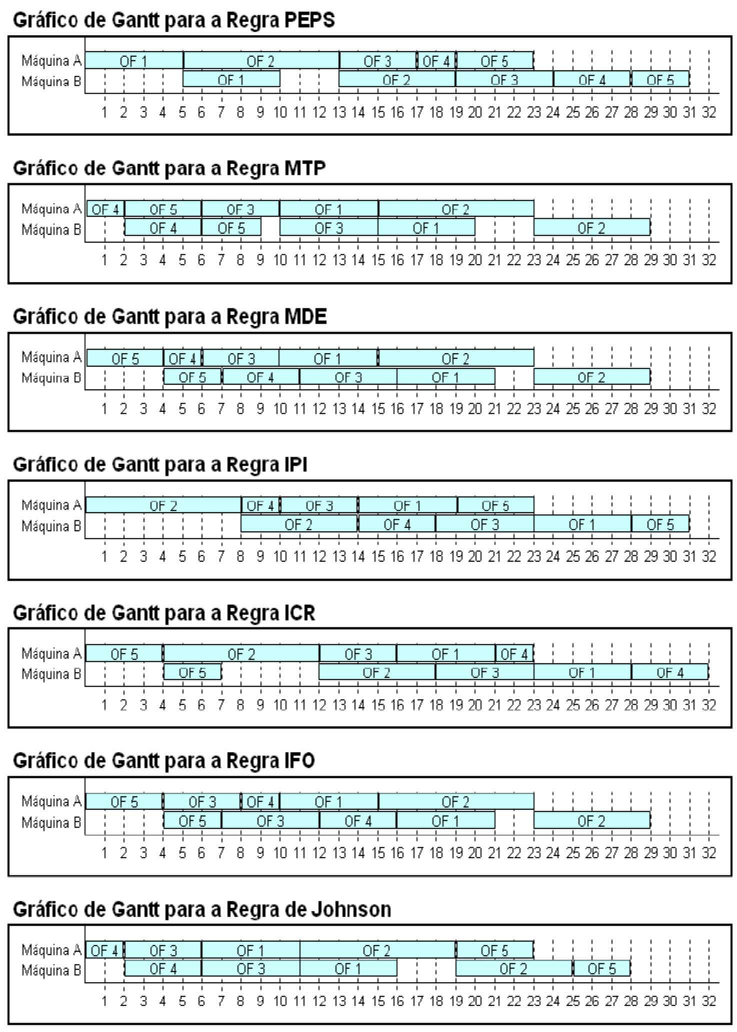
Regra de Johnson (Fonte: Nilson Ribeiro)

Ordens de fabricação	Tempo de processamento (horas)		Entrega (horas)	(Prioridade)
	Produto A	Produto B		
OF1	5	5	15	4
OF2	8	6	20	1
OF3	4	5	13	3
OF4	2	4	10	2
OF5	4	3	9	5

Regra de Johnson (Fonte: Nilson Ribeiro)

- A Tabela a seguir mostra as sequências obtidas pelas regras apresentadas, exceto a regra do índice de falta (IFA) por falta de dados de estoque.

Regras	Sequências
PEPS – primeiro que entra – primeiro que sai	OF1 – OF2 – OF3 – OF4 – OF5
MTP – mínimo tempo de processamento	OF4 – OF5 – OF3 – OF1 – OF2
MDE – menor data de entrega	OF5 – OF4 – OF3 – OF1 – OF2
IPI – índice de prioridade	OF2 – OF4 – OF3 – OF1 – OF5
ICR – índice crítico	OF5 – OF2 – OF3 – OF1 – OF4
IFO – índice de folga	OF5 – OF3 – OF4 – OF1 – OF2
Johnson	OF4 – OF3 – OF1 – OF2 – OF5



Fonte: Nilson Ribeiro (PlanProd-Cap4)

Regra de Johnson

Regras	*Lead Time* (total)	*Lead Time* médio (horas)	Atraso médio (horas)	Tempo de espera médio (horas)
PEPS – primeiro que entra – primeiro que sai	31	31 / 5 = 6,2	(0+0+11+18+22)/5 = 10,2	(0+0+2+5+5) / 5 = 2,4
MTP – mínimo tempo de processamento	29	29 / 5 = 5,8	(0+0+2+5+9)/5 = 3,2	(0+0+0+0+0) / 5 = 0
MDE – menor data de entrega	29	29 / 5 = 5,8	(0+1+3+6+9)/5 = 3,8	(0+1+1+1+0) / 5 = 0,6
IPI – índice de prioridade	31	31 / 5 = 6,2	(0+8+10+13+22)/5 = 10,6	(0+4+4+4+5) / 5 = 4,2
ICR – índice crítico	32	32 / 5 = 6,4	(0+0+10+13+22)/5 = 9,0	(0+0+2+2+5) / 5 = 1,8
IFO – índice de folga	29	29 / 5 = 5,8	(0+0+7+7+9)/5 = 4,6	(0+0+3+2+0) / 5 = 1,0
Johnson	28	28 / 5 = 5,6	(0+0+1+5+19)/5 = 5,0	(0+0+0+0+2) / 5 = 0,4

Regra de Johnson

COMENTÁRIOS:

Regra PEPS:

· É a mais simples, sendo pouco eficiente.

· É muito empregada quando o cliente está presente.

· Faz com que lotes com tempo grande retardem toda a sequência de produção, gerando tempo ocioso nos processos à frente, fazendo com que o tempo de espera médio seja elevado (2,4 h).

Regra MTP

· Obtém um índice de *lead time* médio baixo, reduzindo os estoques em processo, agilizando o carregamento das máquinas à frente e melhorando o nível de atendimento ao cliente.

· No exemplo foi a regra com o melhor desempenho global, perdendo apenas para a regra de Johnson no que se refere ao *lead time.*

· Como ponto negativo, a regra MTP faz com que as ordens com tempos de processamento longos sejam sempre preteridas, principalmente se for grande a dinâmica de chegada de novas ordens com tempos menores.

· Uma solução para este problema seria associarmos ume regra complementar que possibilitasse à uma ordem que fosse preterida um determinado número de vezes, ou após um determinado tempo, avançar para o topo da lista.

Regra de Johnson

Regra MDE

· Como prioriza as datas de entrega dos lotes, faz com que os atrasos se reduzam, o que é conveniente nos processos sob encomenda.

· Porém, como não leva em consideração o tempo de processamento, pode fazer com que lotes com potencial de conclusão rápido fiquem aguardando.

· Nos processos repetitivos em lotes, onde trabalhamos com estoques, as vantagens em priorizar apenas as datas de entrega não são muito claras.

Regra IPI

· Baseada em atribuirmos um índice de prioridade para cada ordem, esta regra teve o pior desempenho entre as sete regras testadas quanto ao atraso e tempo de espera médios.

· Seu uso é mais conveniente apenas como critério de desempate para outra regra.

Regra de Johnson

Regras ICR, IFO e ITA

· Baseadas em cálculos de índices, são normalmente empregadas em sistemas MRPII, dentro de um módulo chamado "controle de fábrica", que se encarrega de gerar prioridades para as ordens liberadas pelo módulo MRP.

· As regras ICR e IFO são baseadas no conceito de folga entre a data de entrega do lote e o tempo de processamento, sendo que a regra IFO considera não só a operação imediata, como todas as demais à frente.

· As regras ICR e IFO privilegiam o atendimento ao cliente, porém, devido a simplicidade do exemplo, a regra ICR obteve o pior *lead time* (32 h) e um atraso médio alto.

· A regra IFA relaciona os estoques atuais com a demanda, buscando evitar que os estoquem se esgotem, o que causa prejuízo ao fluxo produtivo, sendo mais empregada para os itens intermediários que compõem os produtos acabados.

Regra de Johnson

Regra de Johnson

· Apresentou o menor *lead time* (28 h) e um baixo tempo de espera para processamento na 2ª máquina, garantido pela sua heurística de sequenciar tempos rápidos de início para o 1º recurso e tempos rápidos de conclusão para o 2º.

· Infelizmente as restrições desta regra são muito fortes, fazendo com que ela seja de aplicação limitada.

Regra de Johnson

- Exercícios – Programação da Produção

EXERCÍCIO 1

Regra de Johnson

1) Sequenciar as tarefas pelas regras MTP (Menor Tempo de Processamento) e DD (Data Devida) e, em ambos os casos determinar:

   Tempo médio de término;

   Atraso médio;

   Número de tarefas atrasadas.

Tarefa	Tempo de Processamento (horas)	Data Devida (horas)
I	12	30
II	25	28
III	4	8
IV	8	12
V	22	42

EXERCÍCIO 2

Regra de Johnson

2) Existem seis ordens de fabricação aguardando processamento em duas máquinas I e II, sendo essa a ordem obrigatória de processamento, com os seguintes tempos de processamento em horas:

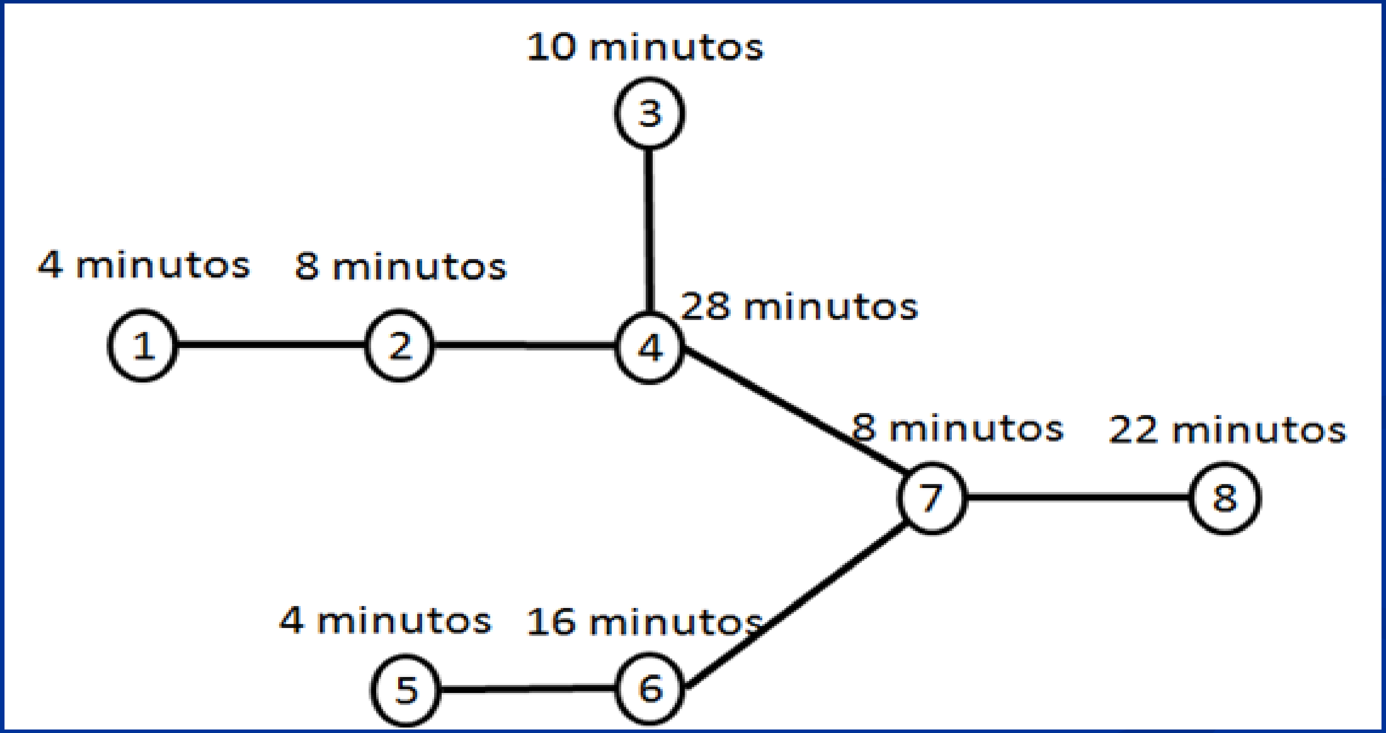
Ordem	Máquina I	Máquina II
14	1	1
25	7	2
8	3	4
30	4	3
35	8	2
32	2	6

Sequenciar as tarefas pela regra de Johnson. Reparar que a ordem 14 tanto pode ser sequenciada em primeiro como em último lugar. Sequenciar então dessas duas formas e calcular o tempo de término do último trabalho sequenciado. Calcular também nos dois casos a eficiência conjunta das máquinas. Existe alguma diferença conforme se sequencie a ordem 14 em primeiro ou em último lugar?

EXERCÍCIO 3

Regra de Johnson

3) Balancear a linha seguinte pela técnica do peso da posição. Usar 30 minutos como tempo de ciclo.



a) Qual é o número mínimo de estações de trabalho?

b) Qual é o número definitivo de estações de trabalho após o balanceamento?

EXERCÍCIO 4

Regra de Johnson

4) A empresa Alfa deve definir a sequência do plano de produção das ordens de produção da Tabela.

Número da tarefa	Tempo de Produção (horas)	Data Devida (horas)
161	3,8	6,0
162	2,1	3,0
163	4,5	14,0
164	3,0	10,0
165	4,2	20,0
166	2,9	19,0

a) Determine a sequência de produção usando:

a.1) Menor Tempo de Processamento

a.2) Razão Crítica

a.3) Data Devida

b) Compare as sequências usando como critério o Atraso Médio.

c) Construa o gráfico de *Gantt* para a melhor sequência.

EXERCÍCIO 5

Regra de Johnson

5) Para os trabalhos descritos na Tabela, que chegaram na ordem 1, 2, 3, 4, 5 e devem ser obrigatoriamente sequenciados na Furação e depois no Corte. Calcular a economia de tempo obtida pela aplicação do algoritmo de Johnson em relação ao critério PEPS.

Tarefas	Tempo de Processamento em horas	
	Furação	Corte
1	7	4
2	3	10
3	8	1
4	6	9
5	2	5

EXERCÍCIO 6

Regra de Johnson

6) Cinco produtos estão aguardando para serem produzidos em uma marcenaria. Todos os produtos têm a mesma sequência de produção, isto é, passam pelas mesmas máquinas e na mesma ordem de processamento. Os tempos de processamento em cada máquina são apresentados na Tabela.

Produto	**Máquina**			
	1	2	3	4
A	5	8	8	8
B	9	5	6	2
C	8	6	8	0
D	7	4	5	9
E	3	8	4	6

Regra de Johnson

Determine a sequência de produção, e compare os resultados fornecidos pelas heurísticas de:

a) Gupta

b) Palmer

c) Johnson (sem atender à condição de otimalidade)

c.1) Somando os tempos

c.2) Só para as duas primeiras máquinas.

Regras de prioridades (πJ) – Heurística de Gupta (vale para 3 máquinas)

$\pi J = [eJ] / [\min (tJ1 + tJ2 + tJ3]$

$eJ = 1$ se $tJ1 < tJ2$

$eJ = - 1$ se $tJ1 \geq tJ2$

Regras de Palmer (prioridades)

Para 3 máquinas = $\pi J = -2tJ1 - 0tJ2 + 2tJ3$

Para 4 máquinas = $\pi J = -3tJ1 - 1tJ2 + 1tJ3 + 3tJ4$

SOLUÇÃO

EXERCÍCIOS
PROGRAMAÇÃO DA PRODUÇÃO

EXERCÍCIO 1

Regra de Johnson

1) Solução.

Alternativa a) MTP – Menor Tempo de Processamento

	Tempo de Processamento		Tempo Total	Data Devida	Atraso
III	4	↓	4	8	0
IV	8		12	12	0
I	12		24	30	0
V	22		46	42	4
II	25		71	28	43
Total			157		47
Média			31,4		9,4

Número de tarefas atrasadas = 2.

Atraso médio = (2 / 5) *100 = 40%.

Alternativa b) DD – Data Devida (***Due date***)

Regra de Johnson

1) Solução.

	Tempo de Processamento	Tempo Total	Data Devida	Atraso
III	4	4	8	0
IV	8	12	12	0
II	25	37	28	9
I	12	49	30	19
V	22	71	42	29
Total		173		57
Média		34,6		11,4

Número de tarefas atrasadas = 3.

Atraso médio = (3 / 5) * 100 = 60%.

EXERCÍCIO 2

HEURÍSTICA

JOHNSON

Regra de Johnson

1) Solução.

Usando a Regra de Johnson: **Sequência 1**

Ordem	Máquina I	Máquina II
14	***1***	***1***
25	7	2
8	3	4
30	4	3
35	8	2
32	2	6

Sequência: __ __ __ __ __ __

Regra de Johnson

1) Solução.

Sequência: <u>14</u> <u>32</u> __ __ __ __

Ordem	Máquina I	Máquina II
25	7	2
8	3	4
30	4	3
35	8	2
32	***2***	***6***

Critério de desempate: menor tempo da 2ª operação em análise
Observação: se o critério de desempate for pelo índice de acordo com Johnson, nesse caso, o resultado final é o mesmo

Ordem	Máquina I	Máquina II
14	**1**	**1**
32	***2***	***6***
25	7	2
8	3	4
30	4	3
35	8	2

Regra de Johnson

1) Solução. **Sequência: 14 32 __ __ __ 25**

Ordem	Máquina I	Máquina II
25	7	2
8	*3*	*4*
30	4	3
35	8	2

Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*
8	3	4
30	4	3
35	8	2
25	7	2
Ordem	Máquina I	Máquina II
8	*3*	*4*
30	*4*	*3*
35	8	2

Regra de Johnson

1) Solução.

Sequência: 14 32 __ __ 35 25

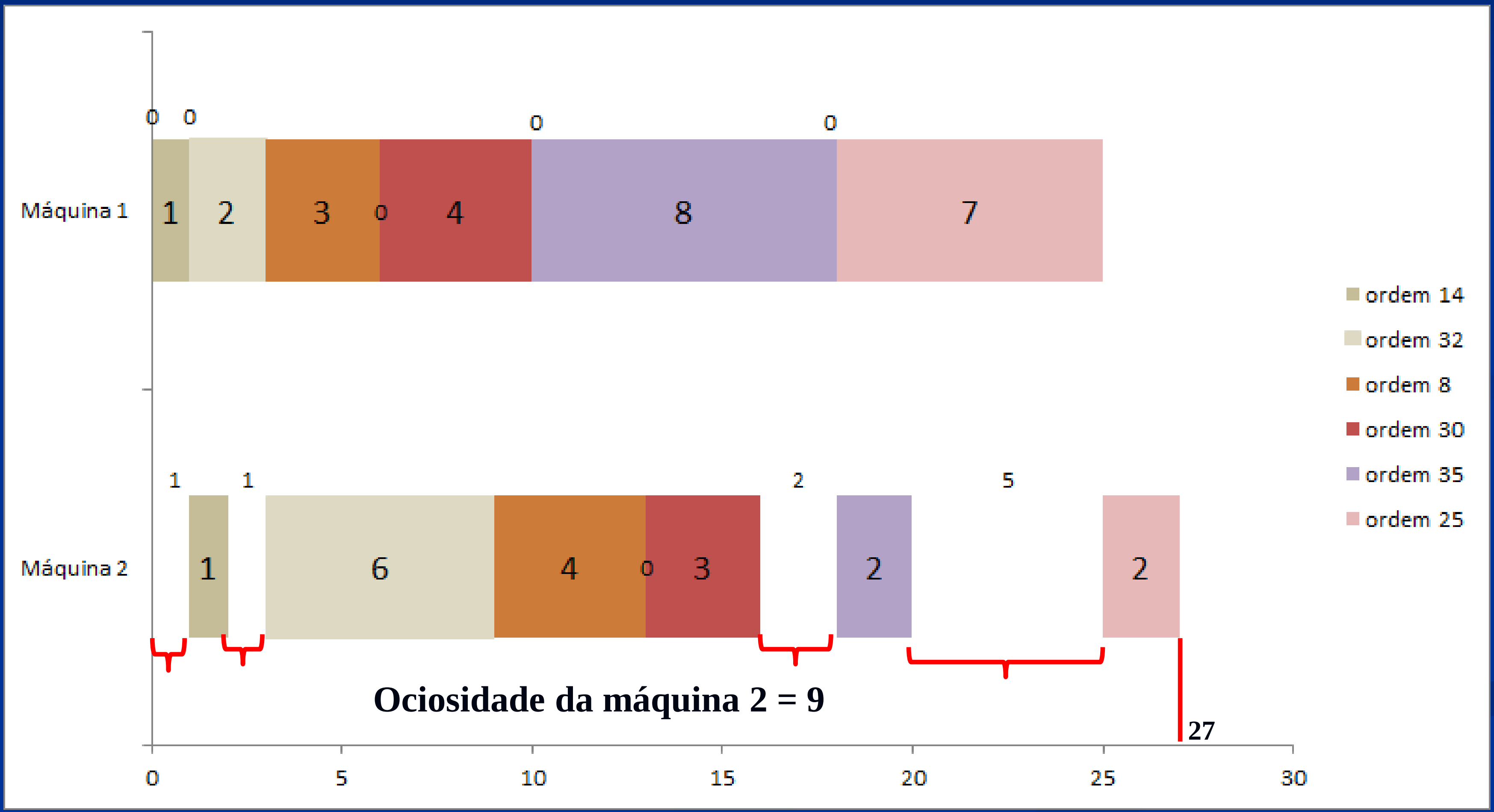
Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*
8	3	4
30	4	3
35	8	2
25	7	2

Ordem	Máquina I	Máquina II
8	3	4
30	4	3

Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*
8	*3*	*4*
30	4	3
35	8	2
25	7	2

Regra de Johnson

1) Solução.



ANÁLISE

SCHEDULING

SEQUÊNCIA 1
MENOR TEMPO DE PROCESSAMENTO
MÁQUINA I

Regra de Johnson

1) Solução.

Usando a Regra de Johnson: **Sequência 1**

Ordem	Máquina I	Máquina II
14	***1***	***1***
25	7	2
8	3	4
30	4	3
35	8	2
32	2	6

Regra de Johnson

1) Solução.

Ordem	Máquina I	Máquina II
25	7	2
8	3	4
30	4	3
35	8	2
32	***2***	***6***

Ordem	Máquina I	Máquina II
14	1	1
32	***2***	***6***
25	7	2
8	3	4
30	4	3
35	8	2

Regra de Johnson

1) Solução.

Ordem	Máquina I	Máquina II
25	7	2
8	***3***	***4***
30	4	3
35	8	2

Ordem	Máquina I	Máquina II
14	1	1
32	***2***	***6***
8	***3***	***4***
25	7	2
30	4	3
35	8	2
Ordem	Máquina I	Máquina II
25	7	2
30	***4***	***3***
35	8	2

Regra de Johnson

1) Solução.

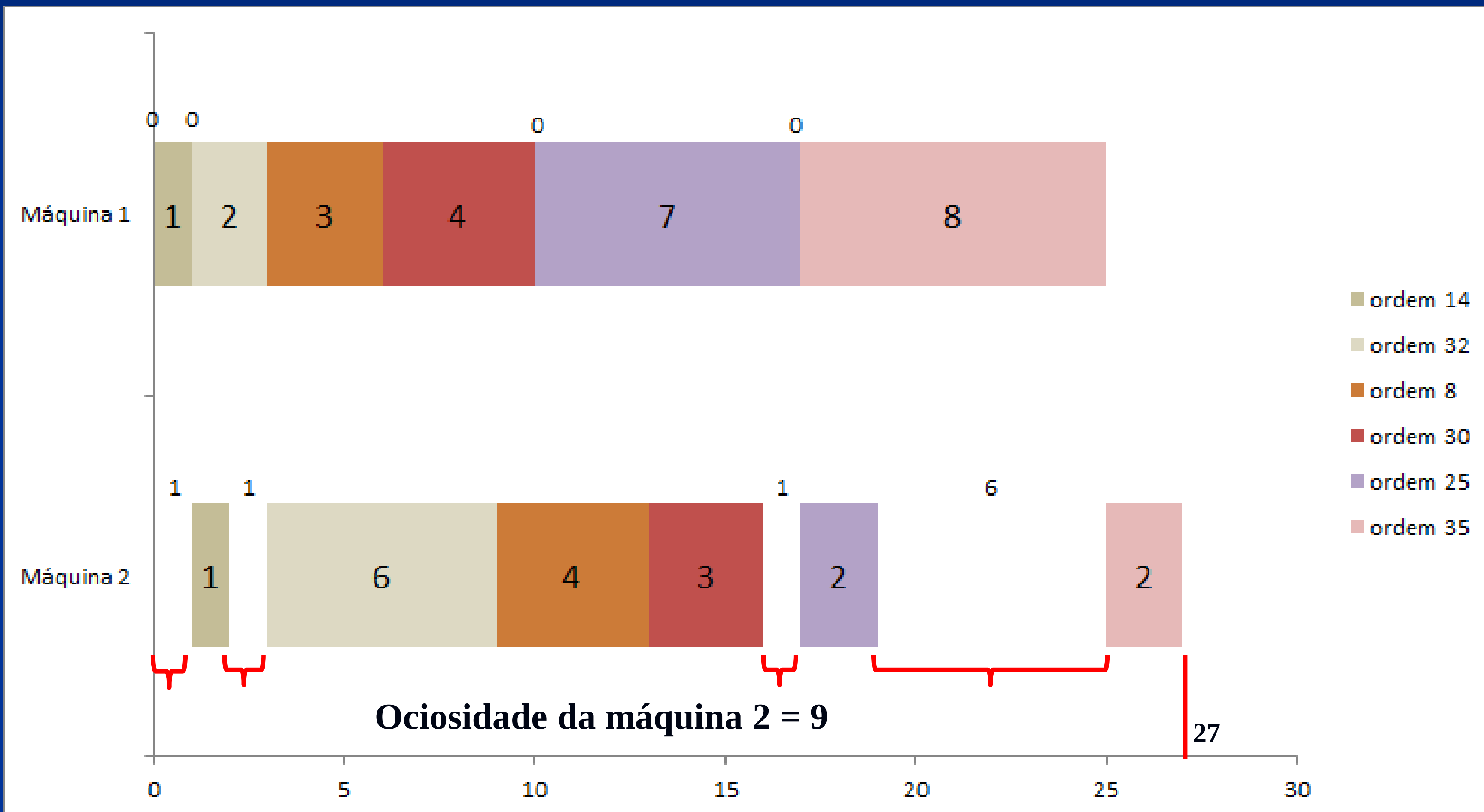
Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*
8	*3*	*4*
30	*4*	*3*
25	7	2
35	8	2

Ordem	Máquina I	Máquina II
25	*7*	*2*
35	8	2

Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*
8	*3*	*4*
30	*4*	*3*
25	*7*	*2*
35	8	2

Regra de Johnson

1) Solução.



SEQUÊNCIA 2
MENOR TEMPO DE PROCESSAMENTO
MÁQUINA I

Regra de Johnson

2) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
8	3	4
30	4	3
35	8	2
32	***2***	***6***

Ordem	Máquina I	Máquina II
32	***2***	***6***
14	1	1
25	7	2
8	3	4
30	4	3
35	8	2

Regra de Johnson

2) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
8	*3*	*4*
30	4	3
35	8	2

Ordem	Máquina I	Máquina II
32	*2*	*6*
8	*3*	*4*
14	1	1
25	7	2
30	4	3
35	8	2

Regra de Johnson

2) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
30	*4*	*3*
35	8	2

Ordem	Máquina I	Máquina II
32	*2*	*6*
8	*3*	*4*
30	*4*	*3*
14	1	1
25	7	2
35	8	2

Regra de Johnson

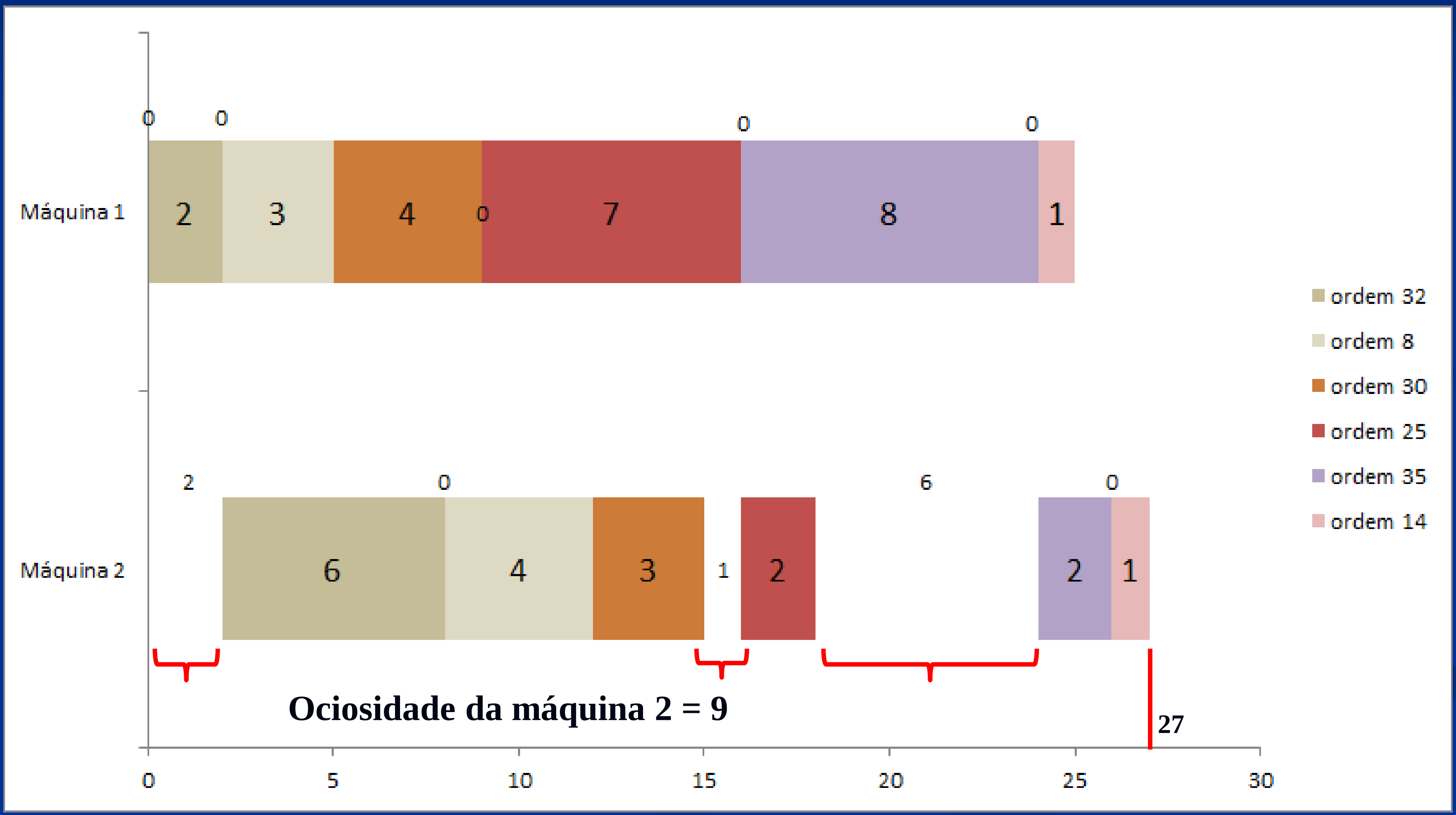
2) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	*7*	*2*
35	8	2

Ordem	Máquina I	Máquina II
32	*2*	*6*
8	*3*	*4*
30	*4*	*3*
25	*7*	*2*
35	8	2
14	1	1

Regra de Johnson

2) Solução.



SEQUÊNCIA 3
MENOR TEMPO DE PROCESSAMENTO MÁQUINA II AÇÃO CONTRÁRIA DA HEURÍSTICA DE JOHNSON

Regra de Johnson

3) Solução.

Não usando a Regra de Johnson: **Sequência 3**

Ordem	Máquina I	Máquina II
14	***1***	***1***
25	7	2
8	3	4
30	4	3
35	8	2
32	2	6

Regra de Johnson

3) Solução.

Ordem	Máquina I	Máquina II
25	7	2
8	3	4
30	4	3
35	8	2
32	*2*	*6*

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
35	8	2
8	3	4
30	4	3
32	*2*	*6*

Regra de Johnson

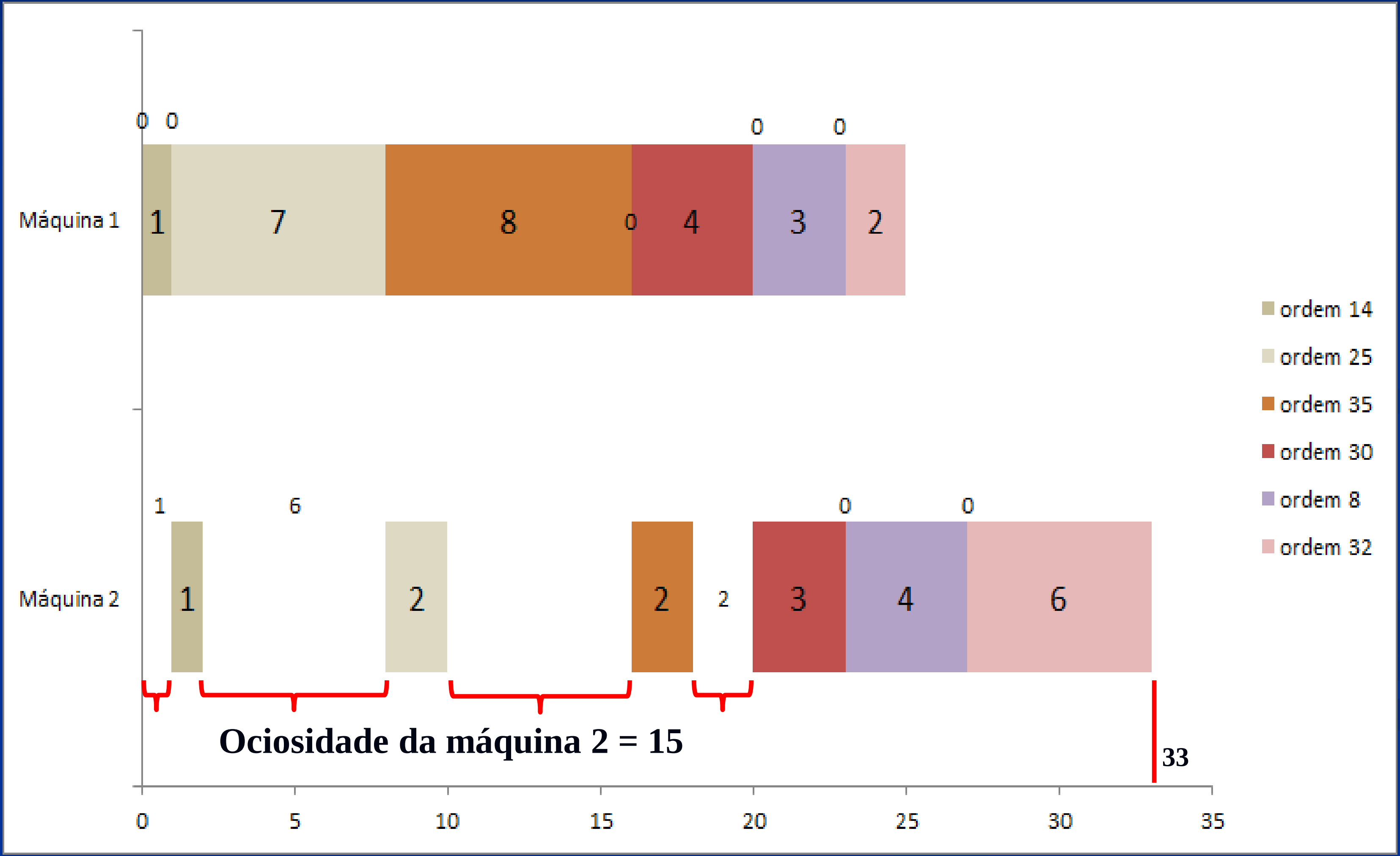
3) Solução.

Ordem	Máquina I	Máquina II
8	3	4
30	4	3
32	*2*	*6*

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
35	8	2
30	*4*	*3*
8	3	4
32	*2*	*6*

Regra de Johnson

3) Solução.



SEQUÊNCIA 4
MENOR TEMPO DE PROCESSAMENTO MÁQUINA II
AÇÃO CONTRÁRIA DA HEURÍSTICA DE JOHNSON

Regra de Johnson

4) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
8	3	4
30	4	3
35	8	2
32	*2*	*6*

Ordem	Máquina I	Máquina II
25	7	2
14	1	1
8	3	4
30	4	3
35	8	2
32	*2*	*6*

Regra de Johnson

4) Solução.

Ordem	Máquina I	Máquina II
14	1	1
8	3	4
30	4	3
35	8	2
32	*2*	*6*

Ordem	Máquina I	Máquina II
25	7	2
35	8	2
14	1	1
8	3	4
30	4	3
32	*2*	*6*

Regra de Johnson

4) Solução.

Ordem	Máquina I	Máquina II
14	1	1
8	3	4
30	**4**	**3**
32	*2*	*6*

Ordem	Máquina I	Máquina II
25	**7**	**2**
35	**8**	**2**
30	**4**	**3**
14	1	1
8	3	4
32	*2*	*6*

Regra de Johnson

4) Solução.

Ordem	Máquina I	Máquina II
14	1	1
8	3	4
32	*2*	*6*

Ordem	Máquina I	Máquina II
25	7	2
35	8	2
30	4	3
8	3	4
14	1	1
32	*2*	*6*

Regra de Johnson

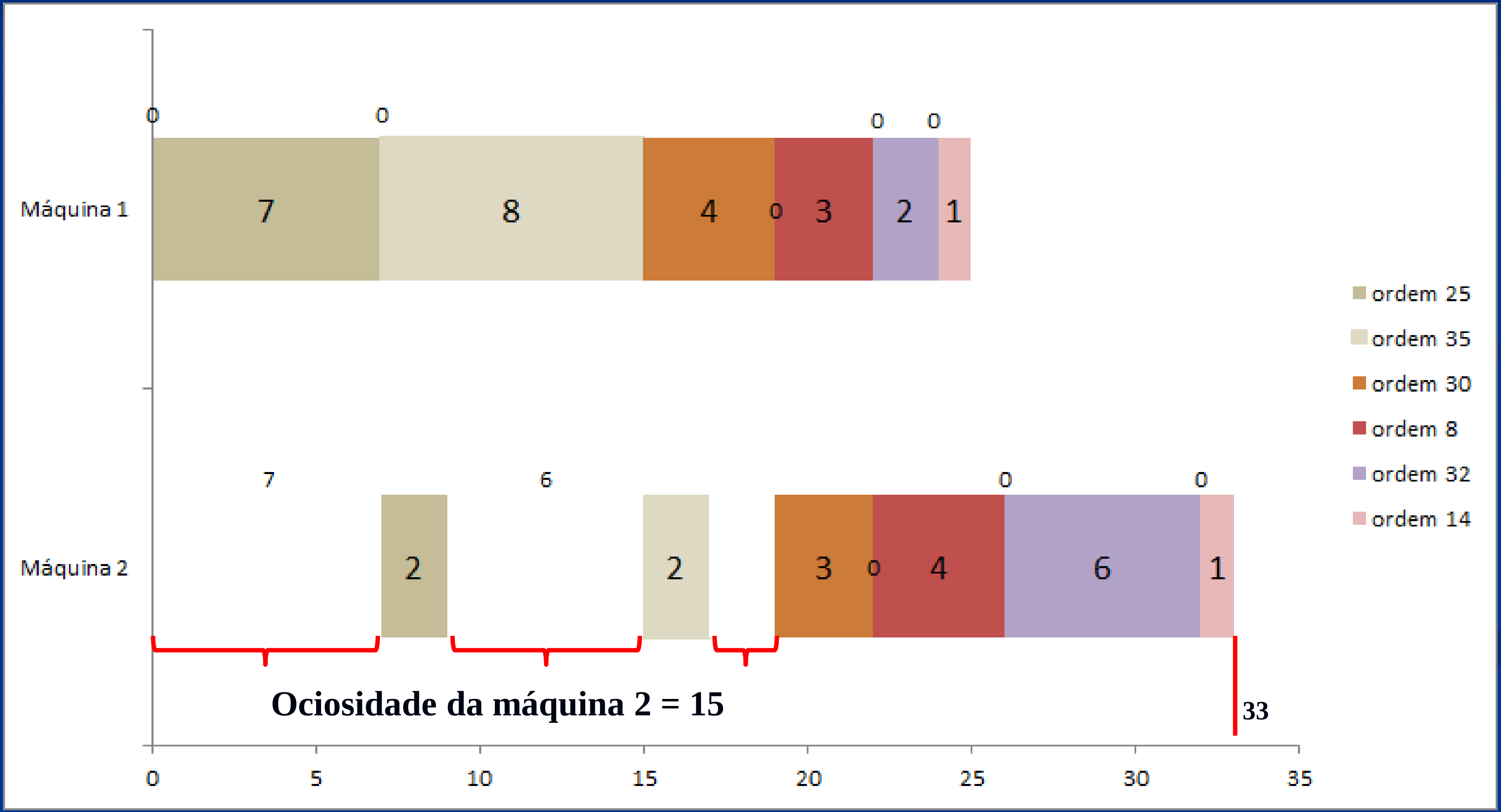
4) Solução.

Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*

Ordem	Máquina I	Máquina II
25	7	2
35	8	2
30	4	3
8	3	4
32	*2*	*6*
14	1	1

Regra de Johnson

4) Solução.



Solução	Ociosidade máquina 2	*Makespan*
1	9	27
2	9	27
3	15	33
4	15	33

HEURÍSTICA DE JOHNSON

MÁQUINA II

Maior tempo de processamento para o menor

SEQUÊNCIA 5
MAIOR TEMPO DE PROCESSAMENTO
MÁQUINA II
HEURÍSTICA DE JOHNSON

Regra de Johnson

5) Solução.

Usando a Regra de Johnson: **Sequência 5**

Ordem	Máquina I	Máquina II
14	***1***	***1***
25	7	2
8	3	4
30	4	3
35	8	2
32	2	6

Regra de Johnson

5) Solução.

Ordem	Máquina I	Máquina II
25	7	2
8	3	4
30	4	3
35	8	2
32	*2*	*6*

Ordem	Máquina I	Máquina II
14	1	1
32	***2***	***6***
25	7	2
8	3	4
30	4	3
35	8	2

Regra de Johnson

5) Solução.

Ordem	Máquina I	Máquina II
25	7	2
8	3	4
30	4	3
35	8	2

Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*
8	3	4
25	7	2
30	4	3
35	8	2

Regra de Johnson

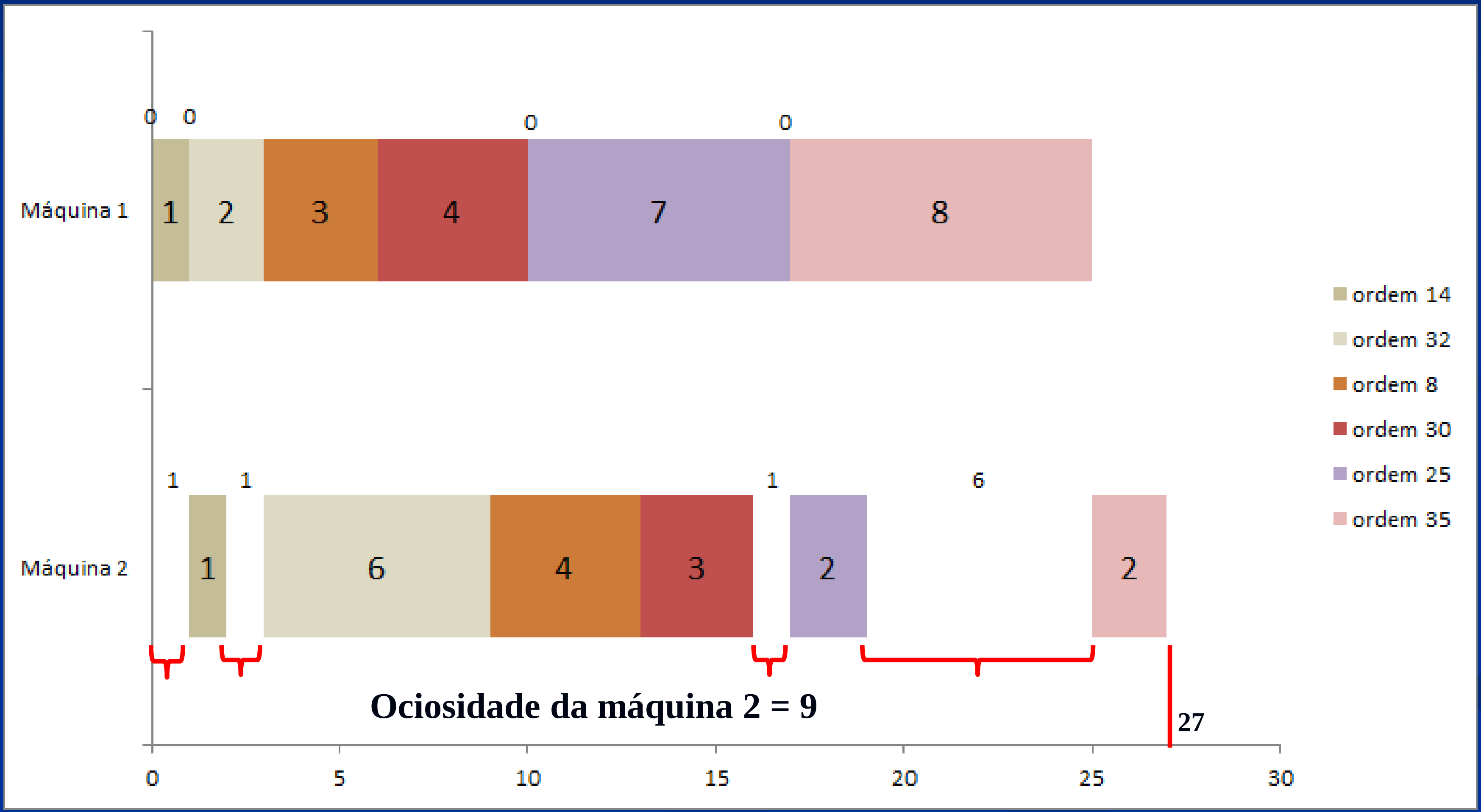
5) Solução.

Ordem	Máquina I	Máquina II
25	7	2
30	4	3
35	8	2

Ordem	Máquina I	Máquina II
14	1	1
32	*2*	*6*
8	3	4
30	4	3
25	7	2
35	8	2

Regra de Johnson

5) Solução.



SEQUÊNCIA 6
MAIOR TEMPO DE PROCESSAMENTO MÁQUINA II HEURÍSTICA DE JOHNSON

Regra de Johnson

6) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
8	3	4
30	4	3
35	8	2
32	*2*	*6*

Ordem	Máquina I	Máquina II
32	***2***	***6***
14	1	1
25	7	2
8	3	4
30	4	3
35	8	2

Regra de Johnson

6) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
8	**3**	**4**
30	4	3
35	8	2

Ordem	Máquina I	Máquina II
32	***2***	***6***
8	**3**	**4**
14	1	1
25	7	2
30	4	3
35	8	2

Regra de Johnson

6) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
30	4	3
35	8	2

Ordem	Máquina I	Máquina II
32	*2*	*6*
8	3	4
30	4	3
14	1	1
25	7	2
35	8	2

Regra de Johnson

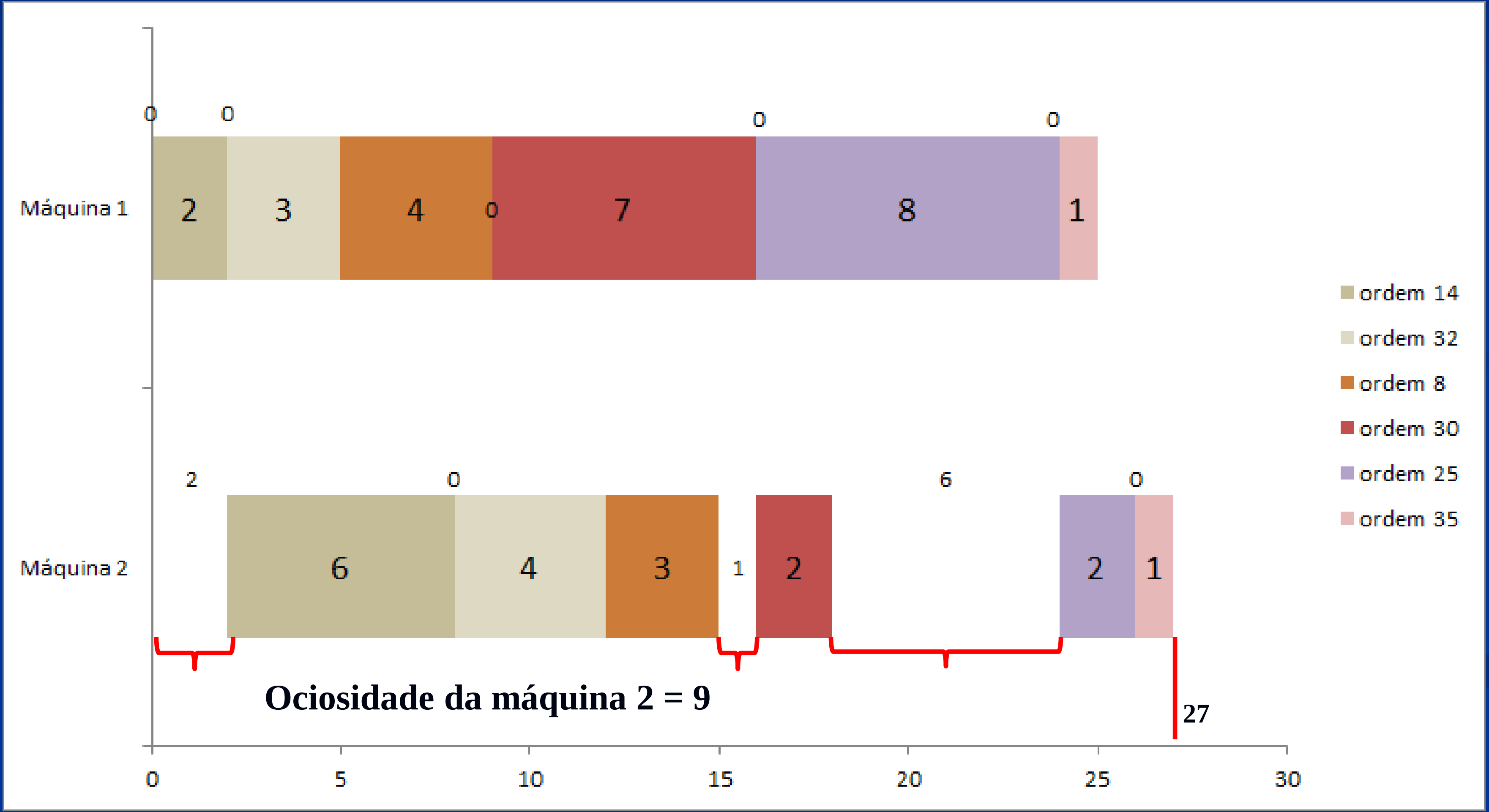
6) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
35	8	2

Ordem	Máquina I	Máquina II
32	*2*	*6*
8	3	4
30	4	3
25	7	2
35	8	2
14	1	1

Regra de Johnson

6) Solução.



Regra de Johnson

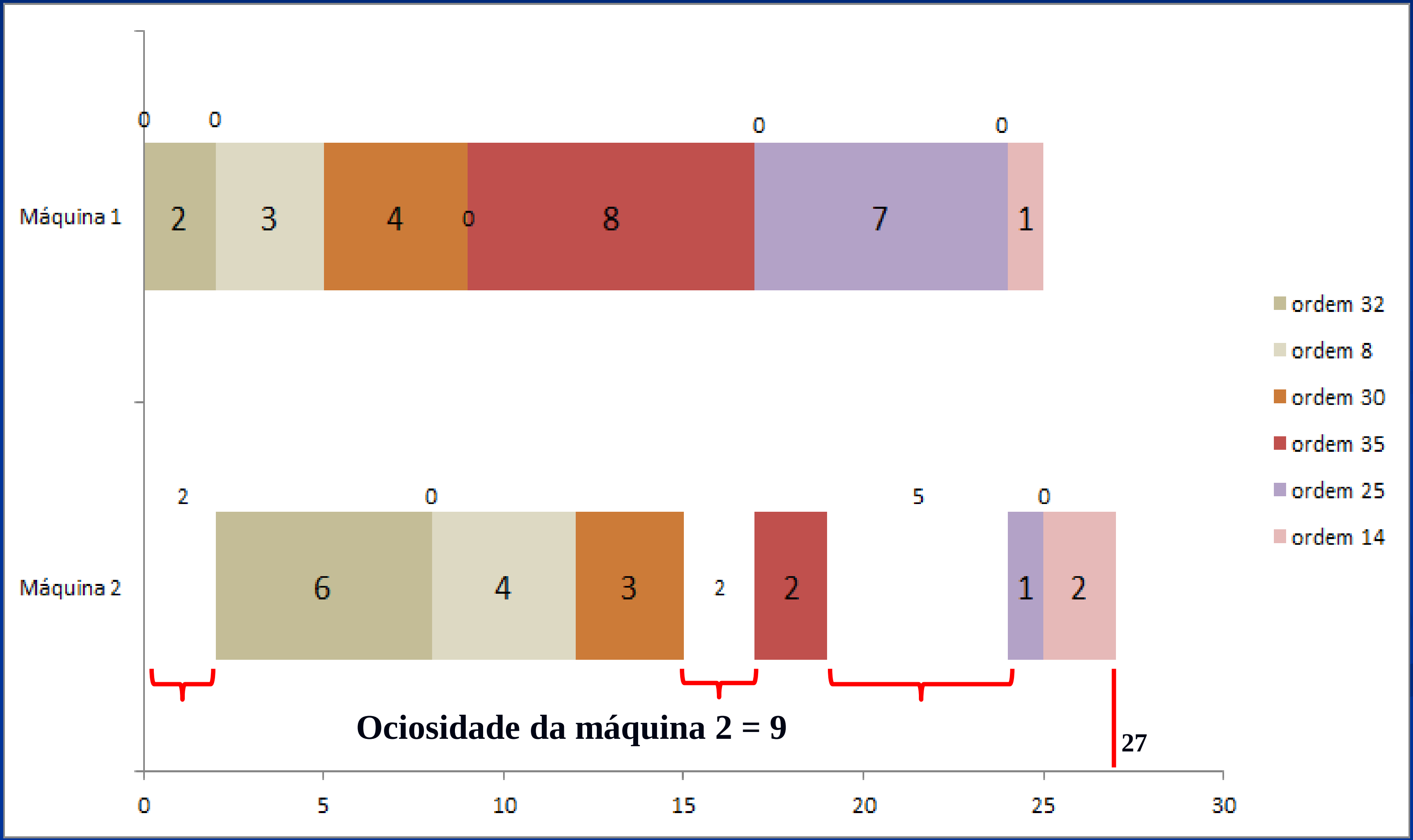
7) Solução.

Ordem	Máquina I	Máquina II
14	1	1
25	7	2
35	8	2

Ordem	Máquina I	Máquina II
32	*2*	*6*
8	3	4
30	4	3
35	8	2
25	7	2
14	1	1

Regra de Johnson

7) Solução.



Solução	Ociosidade máquina 2	*Makespan*
1	9	27
2	9	27
3	15	33
4	15	33
5	9	27
6	9	27
7	9	27

Não seguiu o procedimento da heurística de Johnson

EXERCÍCIO 3

Regra de Johnson

3) Solução.

Atividade	Peso da posição	Ordem decrescente do peso
1	70	1
2	66	3
3	68	2
4	58	4
5	50	5
6	46	6
7	30	7
8	22	8

Atividade	Peso da posição	Ordem decrescente do peso
1	70	1
3	68	2
2	66	3
4	58	4
5	50	5
6	46	6
7	30	7
8	22	8

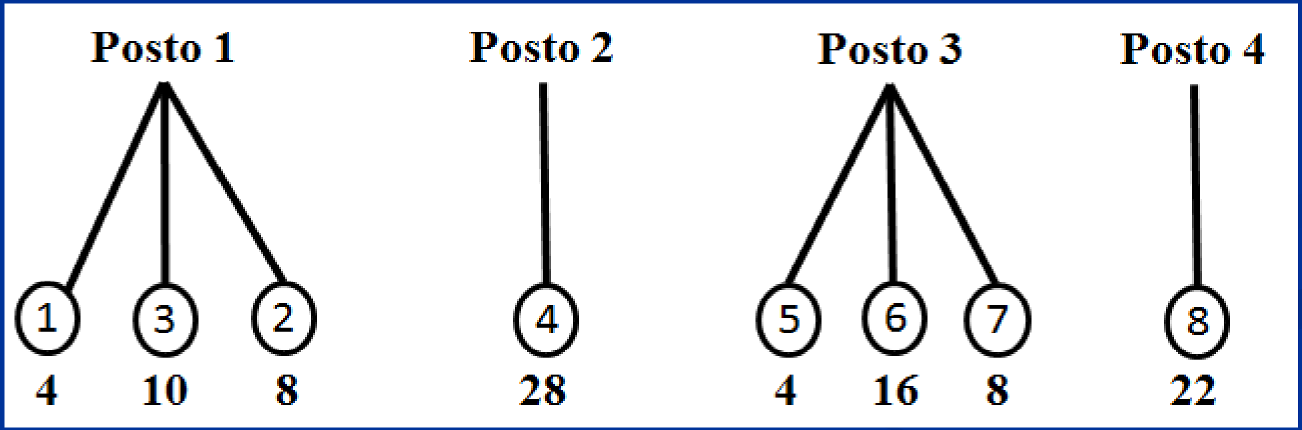
Regra de Johnson

3) Solução.

Tempo total do fluxo = 4 + 8 + 10 + 28 + 4 + 16 + 8 + 22 = 100 minutos

Tempo de ciclo = 30 minutos

N = 100 / 30 = 3,3 = 4 postos de trabalho



	Posto 1	Posto 2	Posto 3	Posto 4	Total	Eficiência
Tempo – posto de trabalho	22	28	28	22	100	83,33%
Eficiência	22 / 30	28 / 30	28 / 30	22 / 30	100 / 120	

EXERCÍCIO 4

Regra de Johnson

4) Solução.

a) Menor Tempo de Processamento

Menor Tempo de Processamento	Número da Tarefa	Tempo de Produção Acumulado	Data Devida	Atraso
2,1	162 (1º)	2,1	3	0
2,9	166 (2º)	5,0	19	0
3,0	164 (3º)	8,0	10	0
3,8	161 (4º)	11,8	6	5,8
4,2	165 (5º)	16,0	20	0
4,5	163 (6º)	20,5	14	6,5

Regra de Johnson

4) Solução.

b) Razão Crítica

Tarefa	Razão Crítica	Ordem	Sequência	Tempo de Processamento Acumulado	Data Devida	Atraso
161	0,63	2º	162	2,1	3	0
162	0,70	1º	161	5,9	6	0
163	0,32	3º	163	10,4	14	0
164	0,30	4º	164	13,4	10	3,4
165	0,21	5º	165	17,6	20	0
166	0,15	6º	166	20,5	19	1,5

Regra de Johnson

4) Solução.

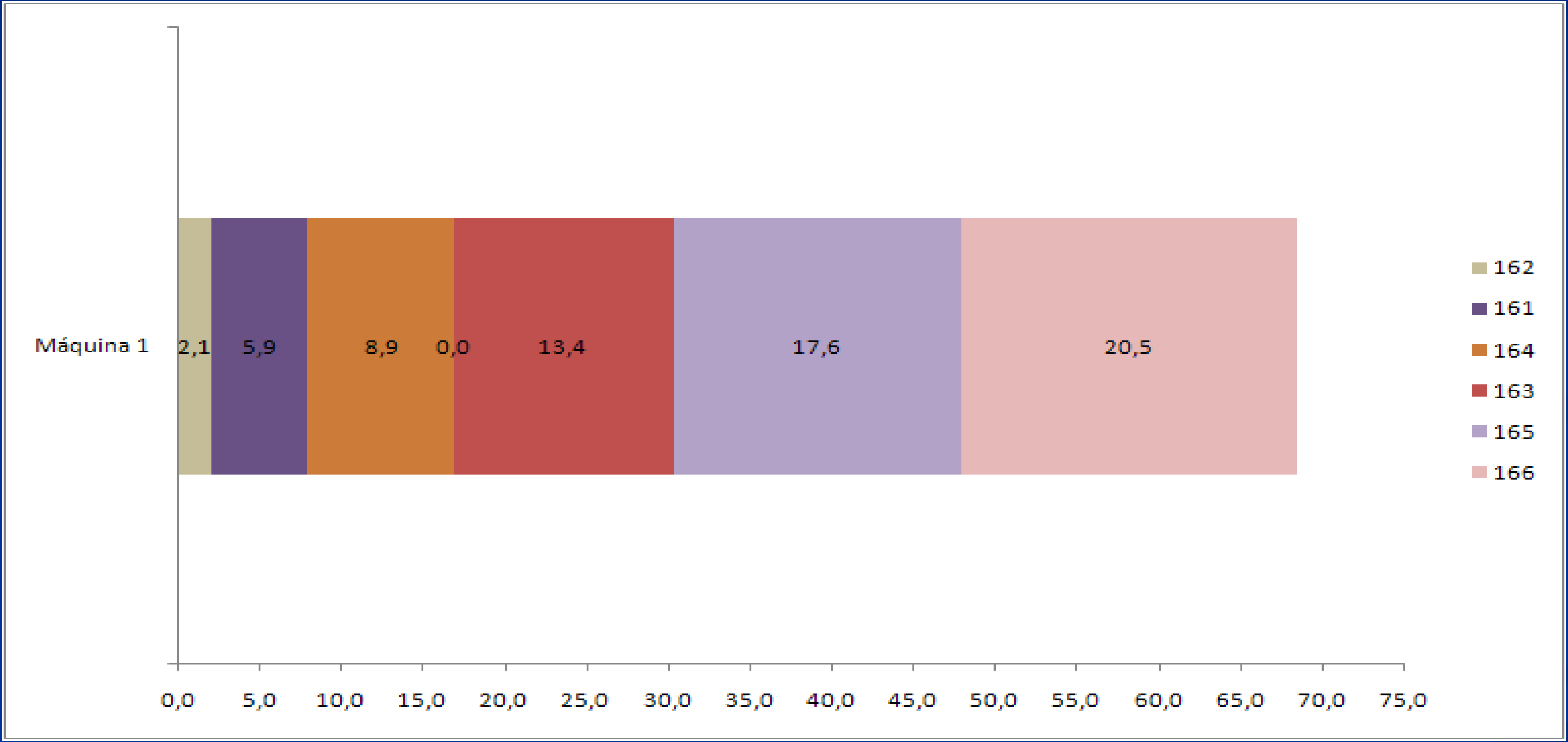
c) Data Devida

Data Devida	Sequência	Tempo de Produção Acumulado	Data Devida	Atraso
3	162	2,1	3	0
6	161	5,9	6	0
10	164	8,9	10	0
14	163	13,4	14	0
19	165	17,6	19	0
20	166	20,5	20	0,5

Regra de Johnson

4) Solução.

c) Data Devida



De acordo com o critério do atraso médio, a melhor sequência é aquela baseada na Data Devida. Além de ter o menor atraso médio, atrasa somente na Data Devida do último produto da sequência.

EXERCÍCIO 5

Regra de Johnson

5) Solução.

Tarefa	Furação	Corte	Acumulo em horas	
			Furação	Corte
5	2	5	2ª hora	7ª hora
2	3	10	5ª hora	17ª hora
4	6	9	11ª hora	26ª hora
1	7	4	18ª hora	30ª hora
3	8	1	26ª hora	31ª hora

Entre as operações 3 e 4 – 24ª – 22ª = 2 horas de espera.

→ 38 – 31 = 7 horas.

Em relação ao critério PEPS, a aplicação do algoritmo de Johnson gerou uma economia de 7 horas.

Regra de Johnson

5) Solução.

PEPS – Primeiro que entra, Primeiro que sai	Acumulo em horas	
Tarefa	Furação	Corte
1	7ª hora	11ª hora
2	10ª hora	21ª hora
3	18ª hora	22ª hora
4	24ª hora	33ª hora
5	26ª hora	38ª hora

EXERCÍCIO 6

Regra de Johnson

6) Solução.

Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10

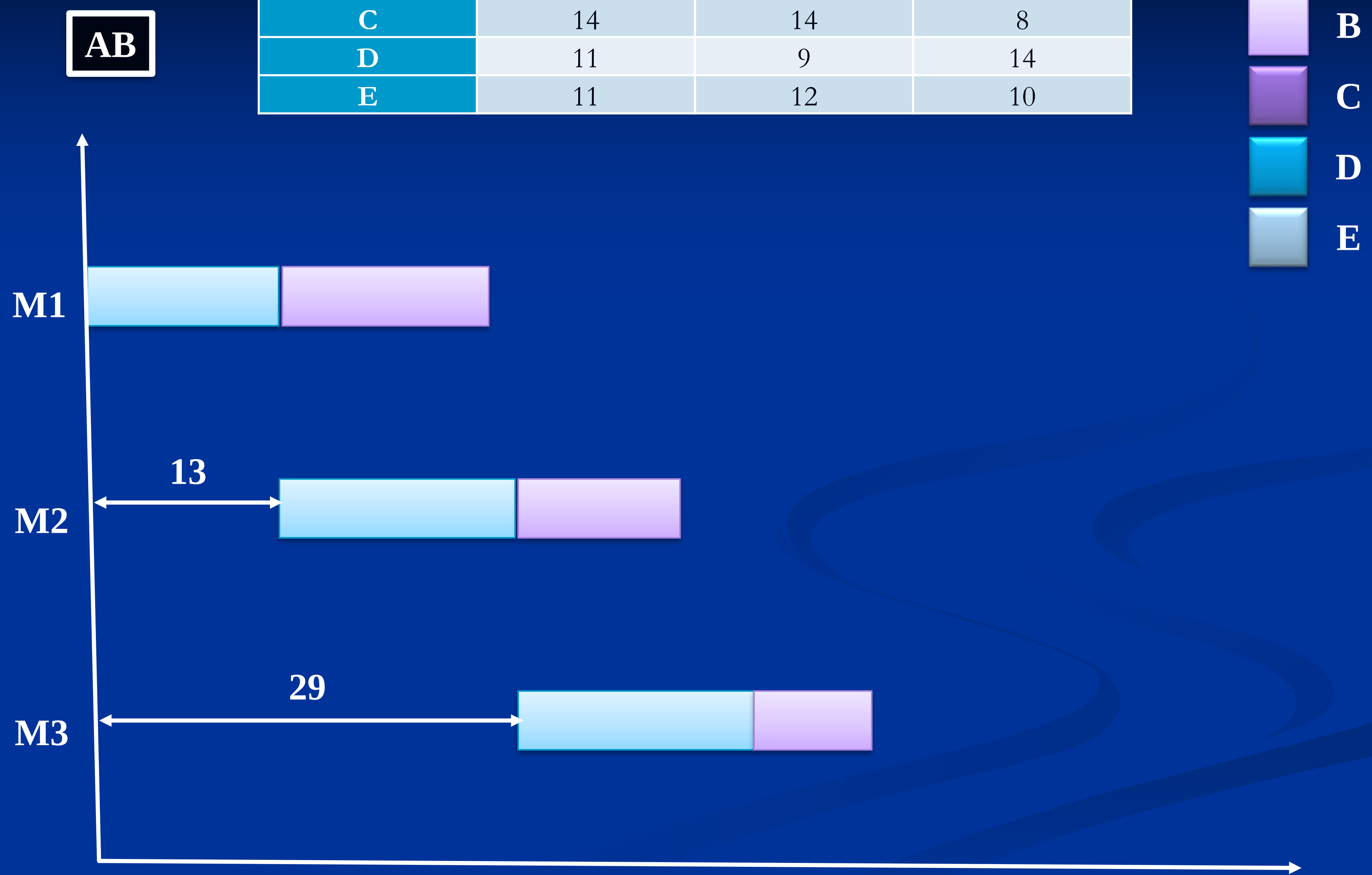
De acordo com Gupta ⇒ n * (n – 1) ⇒ 5 tarefas ⇒ 5 * (5 – 1) = 20 pares:

AB BA CA DA EA
AC BC CB DB EB
AD BD CD DC EC
AE BE CE DE ED

Subgrupo de tarefas

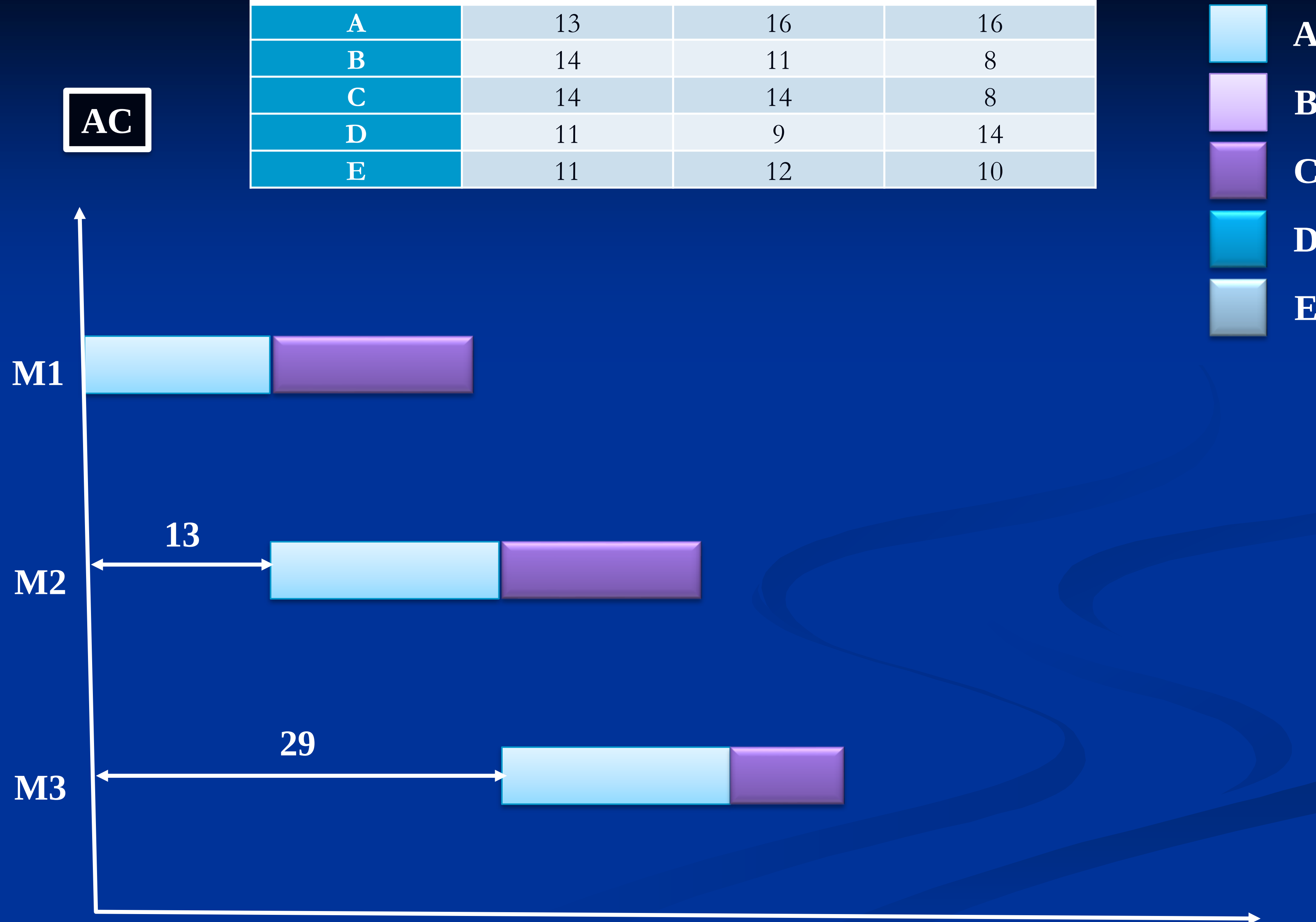
πA

Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10



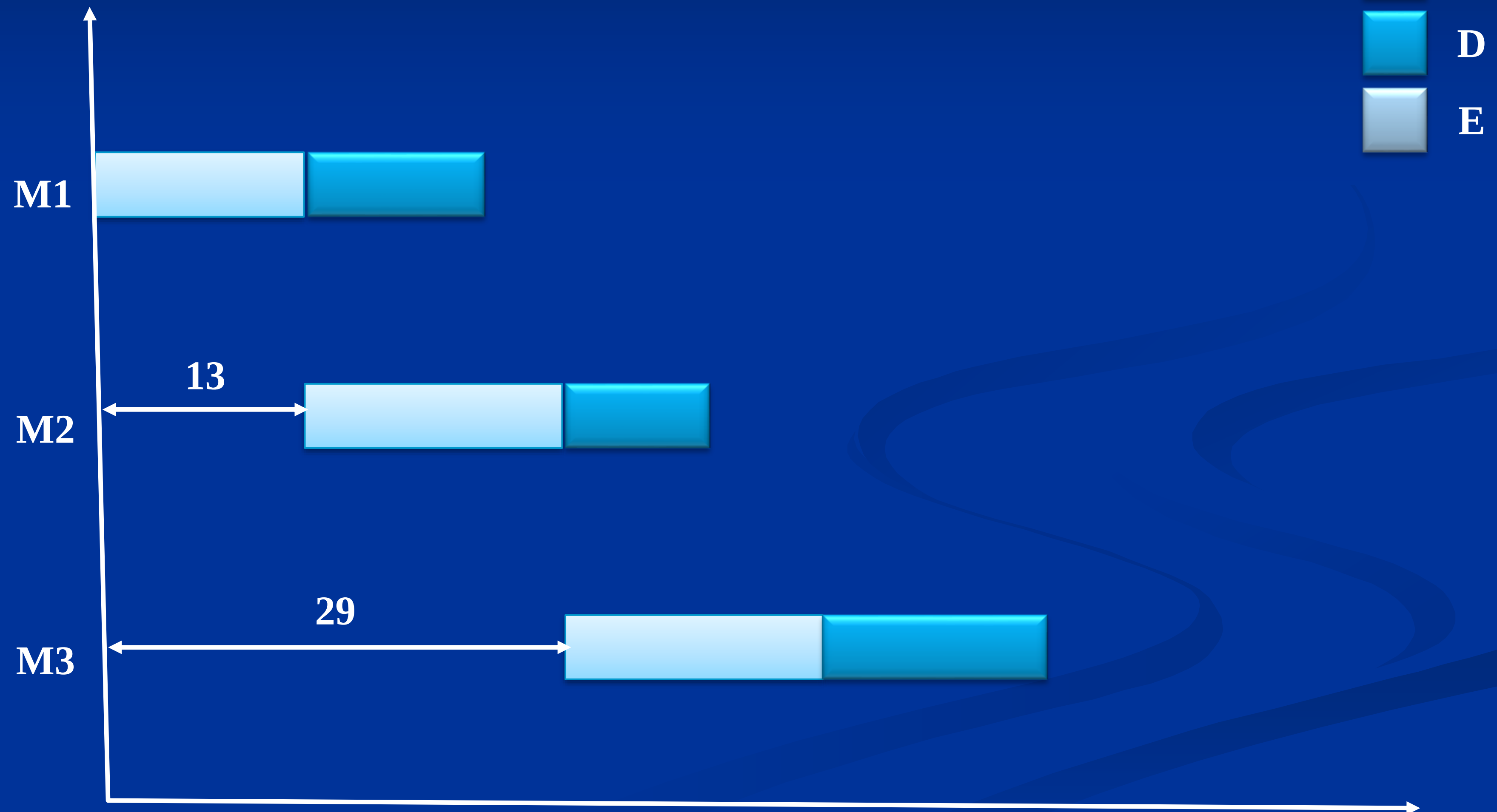
Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10

AC

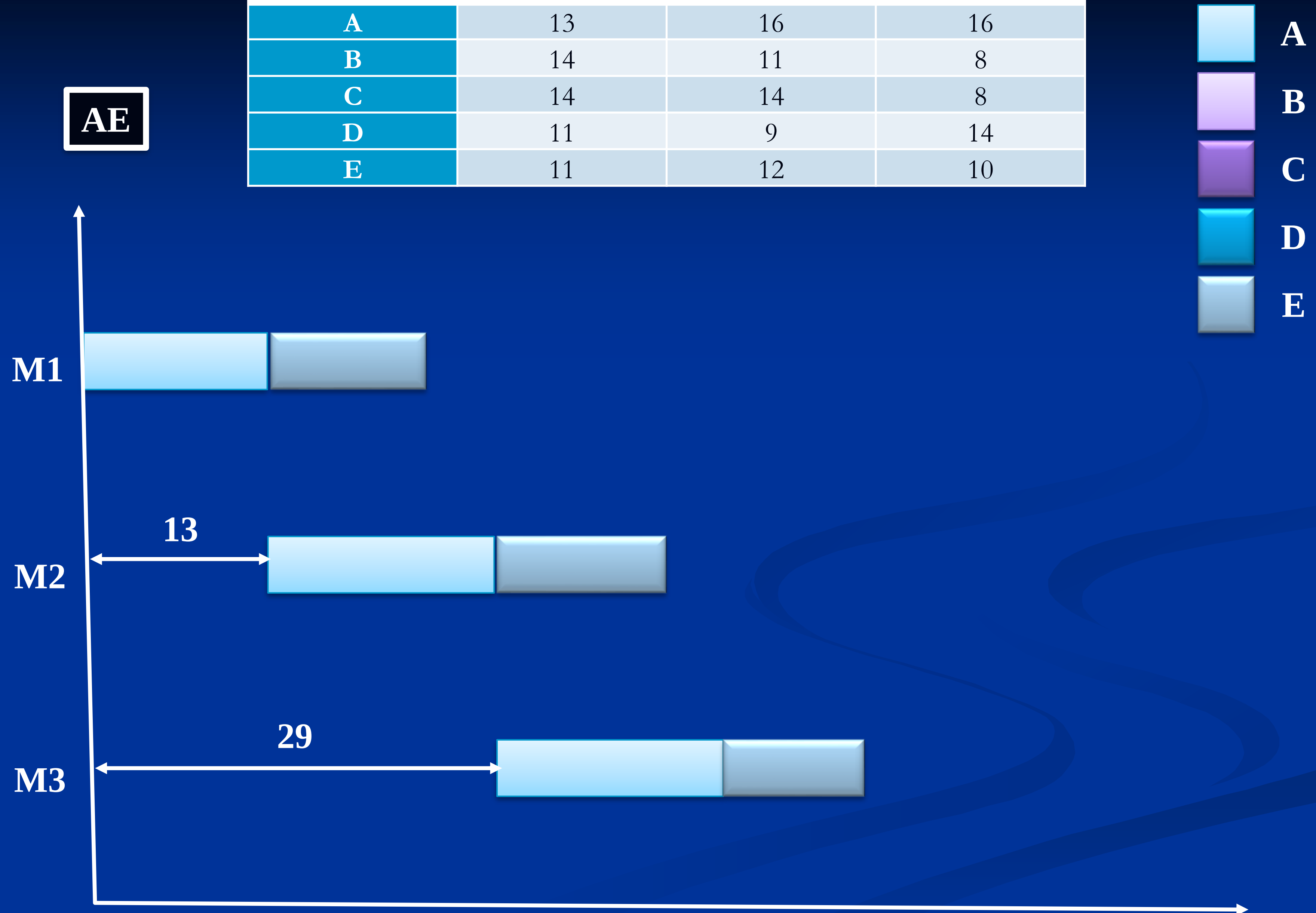


Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10

AD



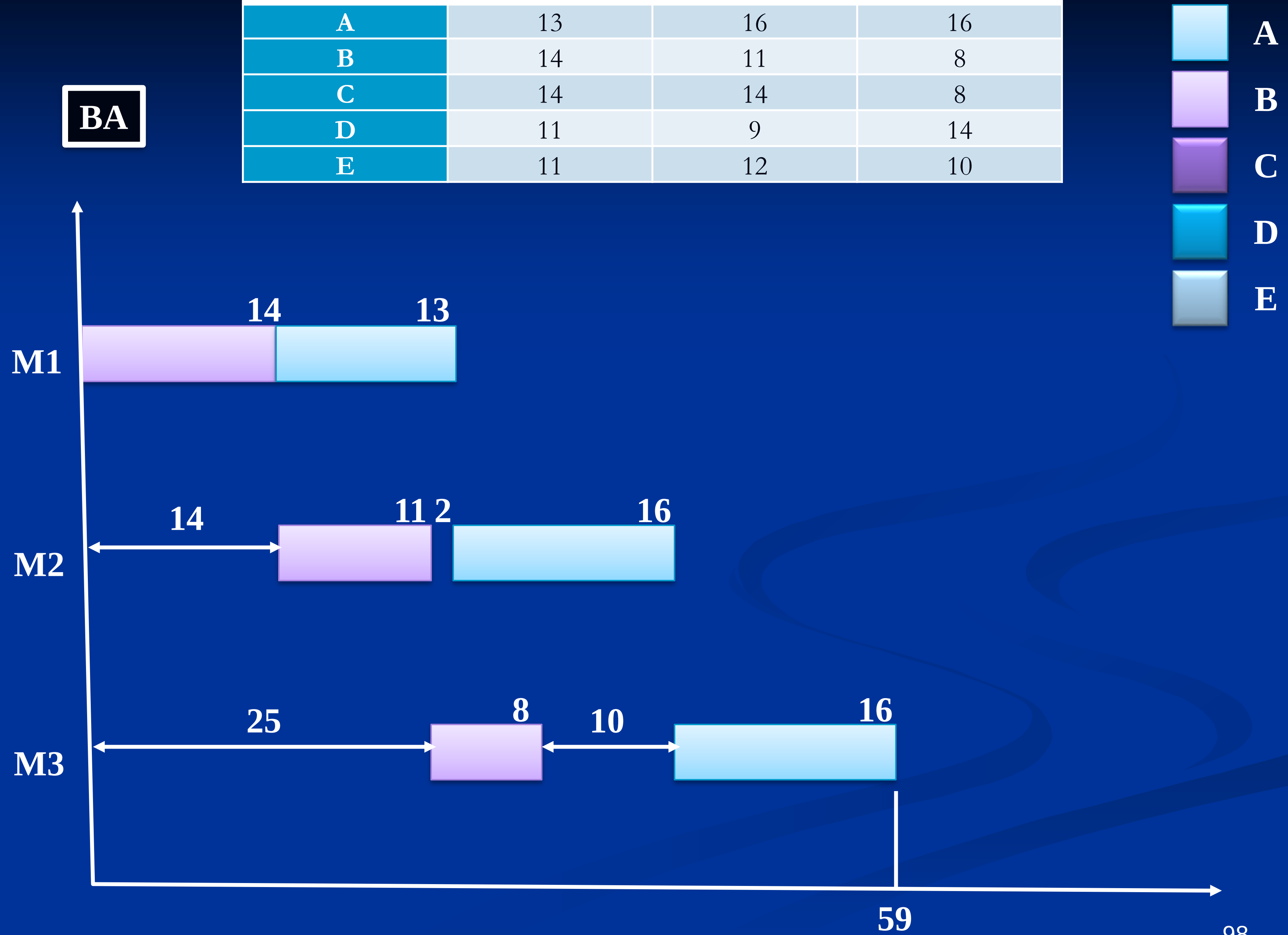
Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10



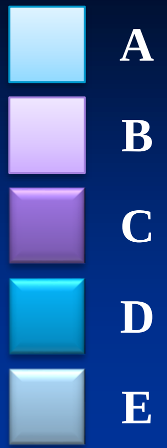
Subgrupo de tarefas

πB

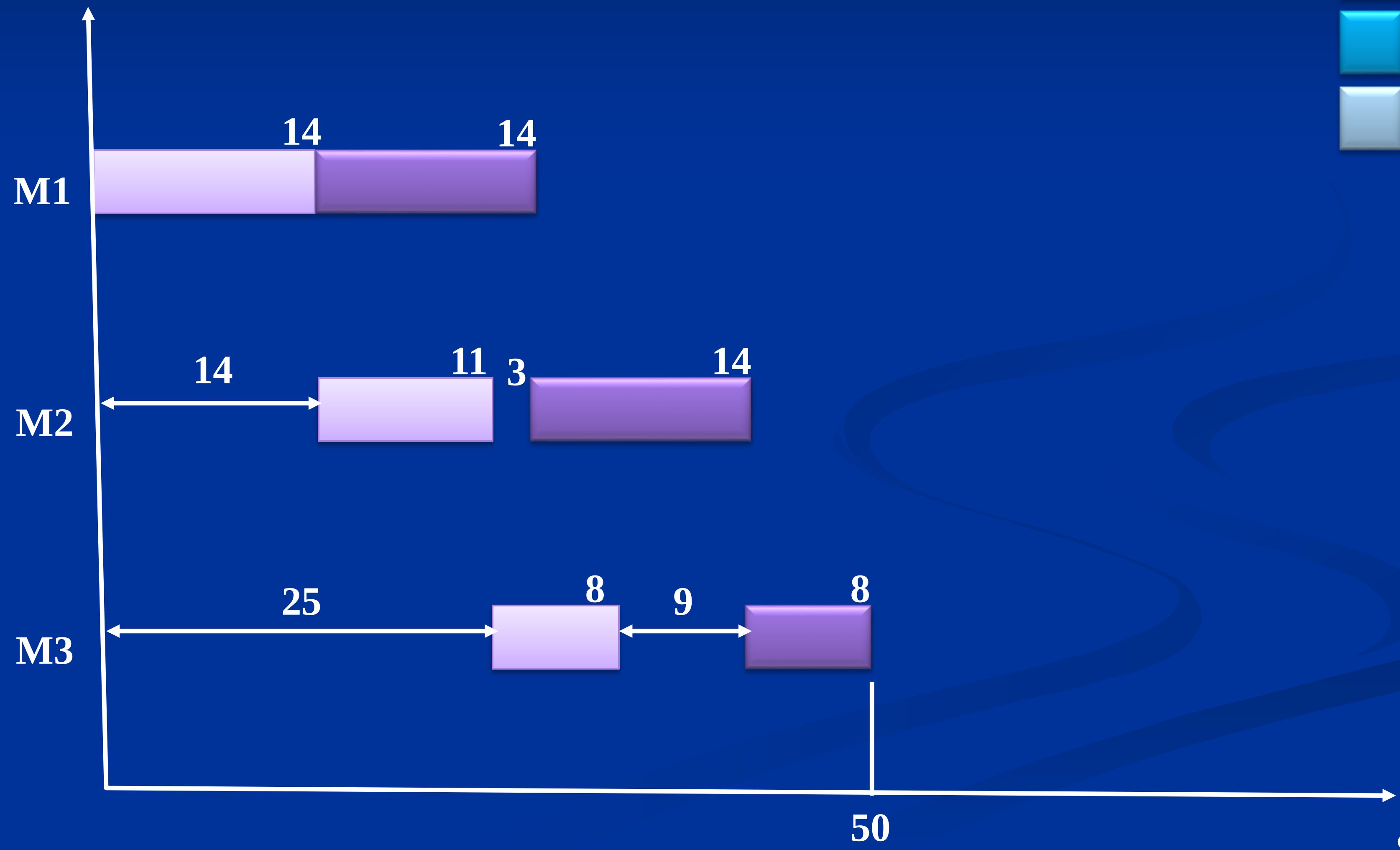
Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10



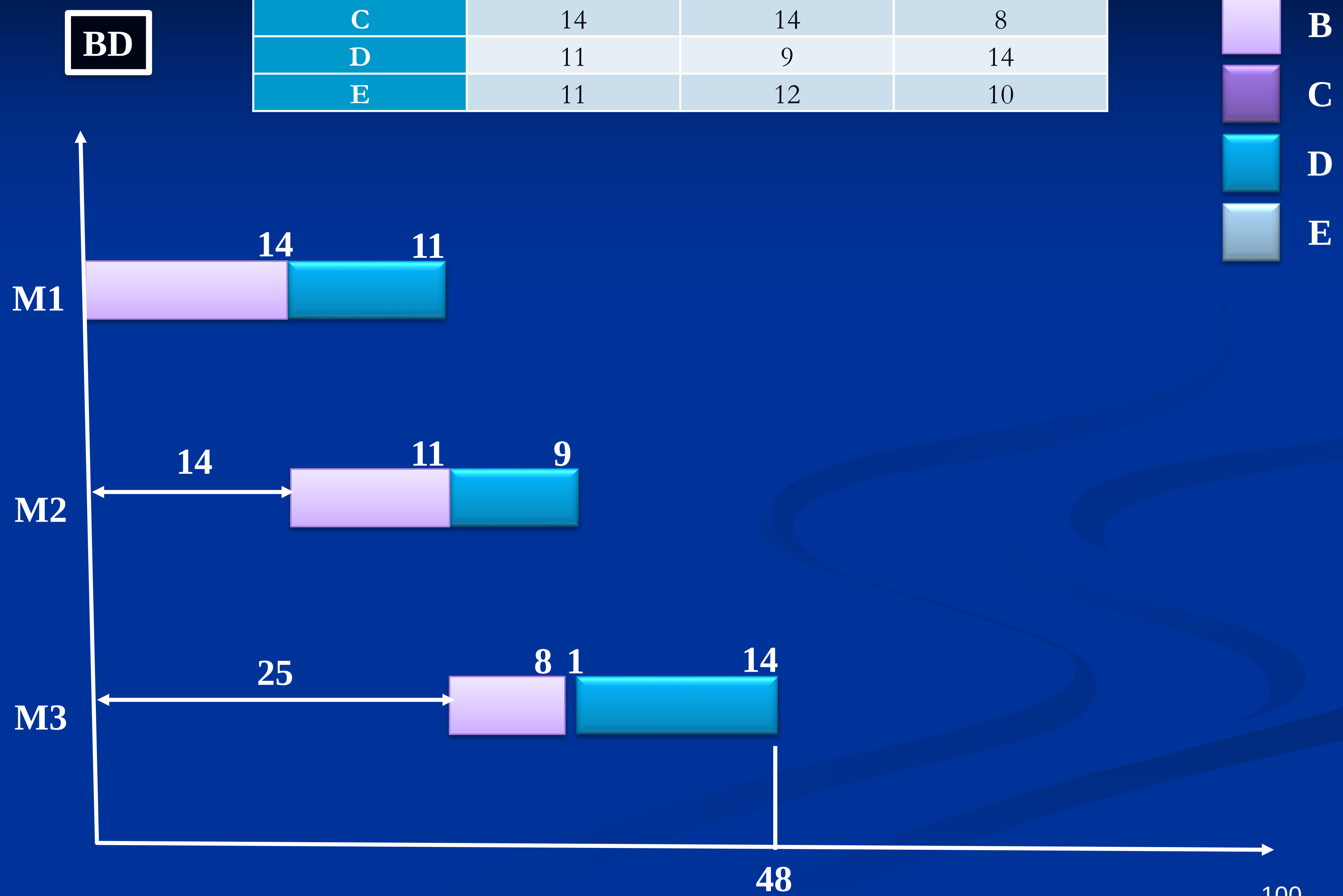
Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10



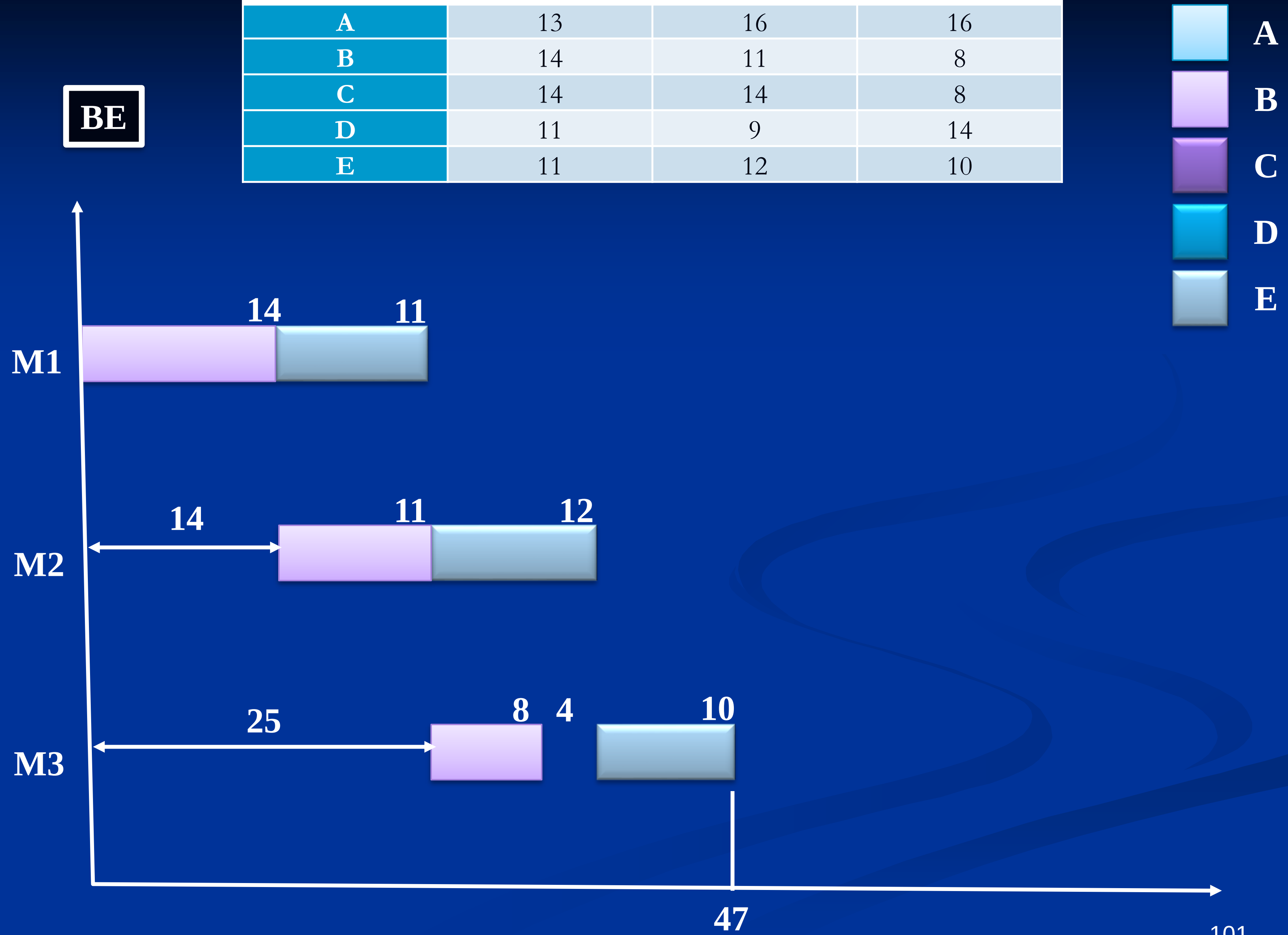
BC



Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10



Produto	Máquina 12	Máquina 23	Máquina 34
A	13	16	16
B	14	11	8
C	14	14	8
D	11	9	14
E	11	12	10



Regra de Johnson

6) Solução.

Gupta

para máquina 1, 2 e 3

- $\pi A = + 1 / 13$
- $\pi B = - 1 / 11$
- $\pi C = - 1 / 14$
- $\pi D = - 1 / 9$
- $\pi E = + 1 / 11$
- ✓ Sequência: E, A, C, B, D

para máquina 2, 3 e 4

- $\pi A = - 1 / 16$
- $\pi B = + 1 / 8$
- $\pi C = + 1 / 8$
- $\pi D = + 1 / 9$
- $\pi E = - 1 / 10$
- ✓ Sequência: B, C, D, A, E ou C, B, D, A, E

Regra de Johnson

6) Solução.

Palmer

- $\pi A = -3(5) - 1(8) + 1(8) + 3(8) = +9$
- $\pi B = -3(9) - 1(5) + 1(6) + 3(2) = -20$
- $\pi C = -3(8) - 1(6) + 1(8) + 3(0) = -22$
- $\pi D = -3(7) - 1(4) + 1(5) + 3(9) = +7$
- $\pi E = -3(3) - 1(8) + 1(4) + 3(6) = +5$
- ✓ Sequência: A, D, E, B, C

Regra de Johnson

6) Solução.

Johnson

- * somando os tempos

Produto	Máquina 1 + 2	Máquina 3 + 4
A	13	16
B	14	8
C	14	8
D	11	14
E	11	10

- → sequência D, A, E, C, B ou D, A, E, B, C
- Johnson só para as duas primeiras máquinas

✓ Sequência: E, A, C, B, D

Regra de Johnson

6) Solução.

✓ Relação de resultados

Heurística	Sequência
Gupta Máquina 1, 2, 3	E, A, C, B, D
Gupta Máquina 2, 3, 4	B, C, D, A, E
Palmer	A, D, E, B, C
Johnson (1+2) (3+4)	D, A, E, B, C
Johnson Máquina 1 e 2	E, A, C, B, D

✓ As sequências de Palmer e de Johnson (somando os tempos) são muito parecidas, e provavelmente são as melhores dentre as sequências descritas na Tabela.

Exemplo Johnson

Ji	J1	J2	J3	J4	J5
Pi1	**3**	**5**	**1**	**6**	**7**
pi1	**6**	**2**	**2**	**6**	**5**

Etapa	Tarefas ainda não programadas	Mínimo pij	designação	Sequência parcial
1	J_1, J_2, J_3, J_4, J_5	p_{31}	$J_3 = J_{[1]}$	J3
2	J_1, J_2, J_4, J_5	p_{22}	$J_2 = J_{[5]}$	J3_ _ _J2
3	J_1, J_4, J_5	p_{11}	$J_1 = J_{[2]}$	J3J1_ _J2
4	J_4, J_5	p_{52}	$J_5 = J_{[4]}$	J3J1 _ J5J2
5	J_4	$p_{41} = p_{42}$	$J_4 = J_{[3]}$	$J_3J_1J_4J_5J_2$

Exemplo

Matriz D

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: _ _ _ _ _ _ _ A8

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: _ _ _ _ _ _ A6A8

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: A1 _ _ _ _ _ A6A8

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: A1 _ _ _ _ A5A6A8

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: A1A7 _ _ _ A5A6A8

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: A1A7 _ _ A3A5A6A8

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: A1A7 _ A2A3A5A6A8

D:	M1	M2
A1	2	4
A2	7	5
A3	6	4
A4	9	8
A5	8	2
A6	7	1
A7	3	9
A8	5	1

Sequência: A1A7A4A2A3A5A6A8

Johnson

Caso especial

EXEMPLO

D:	M1	M2	M3
A1	4	2	5
A2	6	4	2
A3	6	4	6
A4	7	2	4
A5	5	3	6
A6	5	1	5

EXEMPLO

D:	M1	M2
A1	6	7
A2	10	6
A3	10	10
A4	9	6
A5	8	9
A6	6	6

Sequência: A1

EXEMPLO

D:	M1	M2
A1	6	7
A2	10	6
A3	10	10
A4	9	6
A5	8	9
A6	6	6

Sequência: A1A6

EXEMPLO

D:	M1	M2
A1	6	7
A2	10	6
A3	10	10
A4	9	6
A5	8	9
A6	6	6

Sequência: A1A6 _ _ _ A4

EXEMPLO

D:	M1	M2
A1	6	7
A2	10	6
A3	10	10
A4	9	6
A5	8	9
A6	6	6

Sequência: A1A6 _ _ A2A4

EXEMPLO

D:	M1	M2
A1	6	7
A2	10	6
A3	10	10
A4	9	6
A5	8	9
A6	6	6

Sequência: A1A6A5_ A2A4

EXEMPLO

D:	M1	M2
A1	6	7
A2	10	6
A3	10	10
A4	9	6
A5	8	9
A6	6	6

Sequência: A1A6A5A3A2A4

Heurística de Johnson

- **Passo 1.** Encontrar min $\{t_{i1},t_{i2}\}$
- **Passo 2 (a).** Se o menor tempo de processamento requer a 1ª máquina, colocar a entidade respectiva na primeira posição disponível. Ir para o passo 3.
- **Passo 2 (b).** Se o menor tempo de processamento requer a 2ª máquina, colocar a entidade respectiva na última posição disponível. Ir para o passo 3.
- **Passo 3.** Retirar a entidade atribuída e voltar ao passo 1 até que todos as entidades sejam atribuídas.

Observação: Johnson sugere no caso de empate fazer uso do índice ou posição da tarefa para o desempate, ou seja, a tarefa posicionada na lista de tarefas próximo do início da lista de tarefas deve ser alocada em primeiro lugar.

	Máquina A	Máquina B
Tarefa 1	2	4
Tarefa 2	3	2
Tarefa 3	5	4
Tarefa 4	3	6
Tarefa 5	4	7

Apenas ordenando as tarefas na máquina 1

Do menor para o maior tempo de processamento

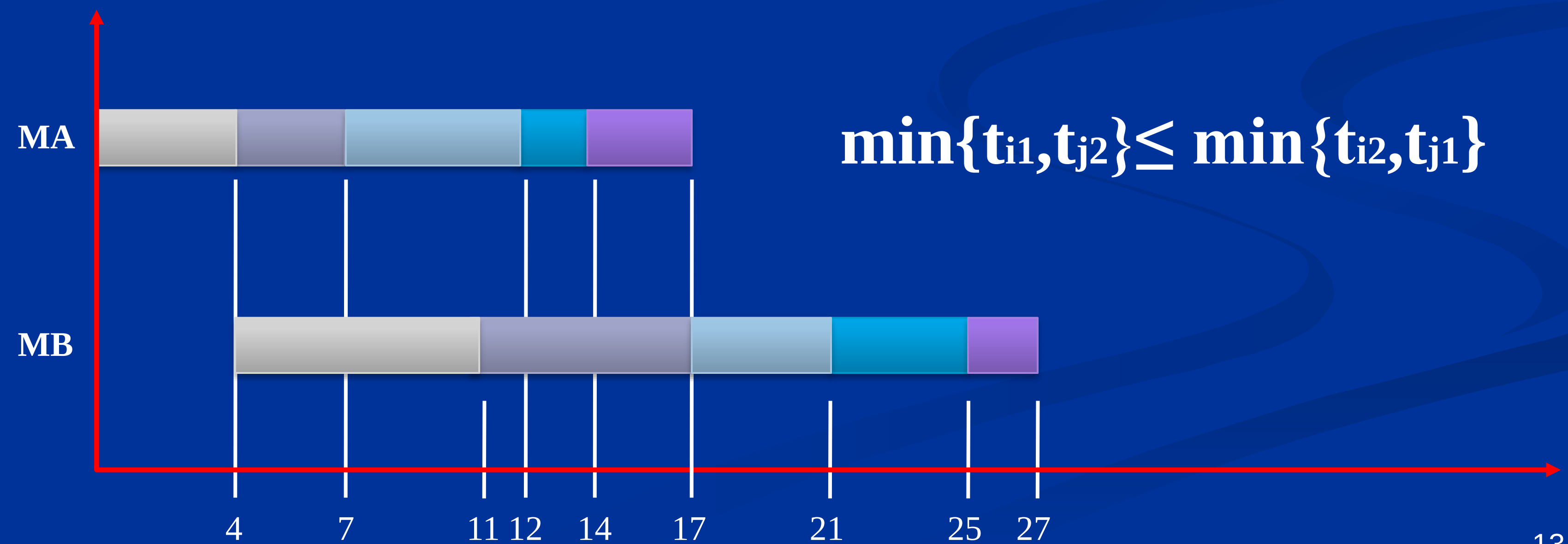
	Máquina A	Máquina B
Tarefa 1	2	4
Tarefa 2	3	2
Tarefa 4	3	6
Tarefa 5	4	7
Tarefa 3	5	4
Total	17	23

$$\min\{t_{i1},t_{j2}\} \leq \min\{t_{i2},t_{j1}\}$$

Apenas ordenando as tarefas na máquina 2

Do maior para o menor tempo de processamento

	Máquina A	Máquina B
Tarefa 5	4	7
Tarefa 4	3	6
Tarefa 3	5	4
Tarefa 1	2	4
Tarefa 2	3	2



$$\min\{t_{i1},t_{j2}\} \leq \min\{t_{i2},t_{j1}\}$$

Heurística de Johnson

Alocação das tarefas: menor tempo de processamento na máquina 1 (alocação máquina 1), menor tempo de processamento na máquina 2 (alocação máquina 2)

	Máquina A	Máquina B
Tarefa 1	2	4
Tarefa 2	3	2
Tarefa 3	5	4
Tarefa 4	3	6
Tarefa 5	4	7

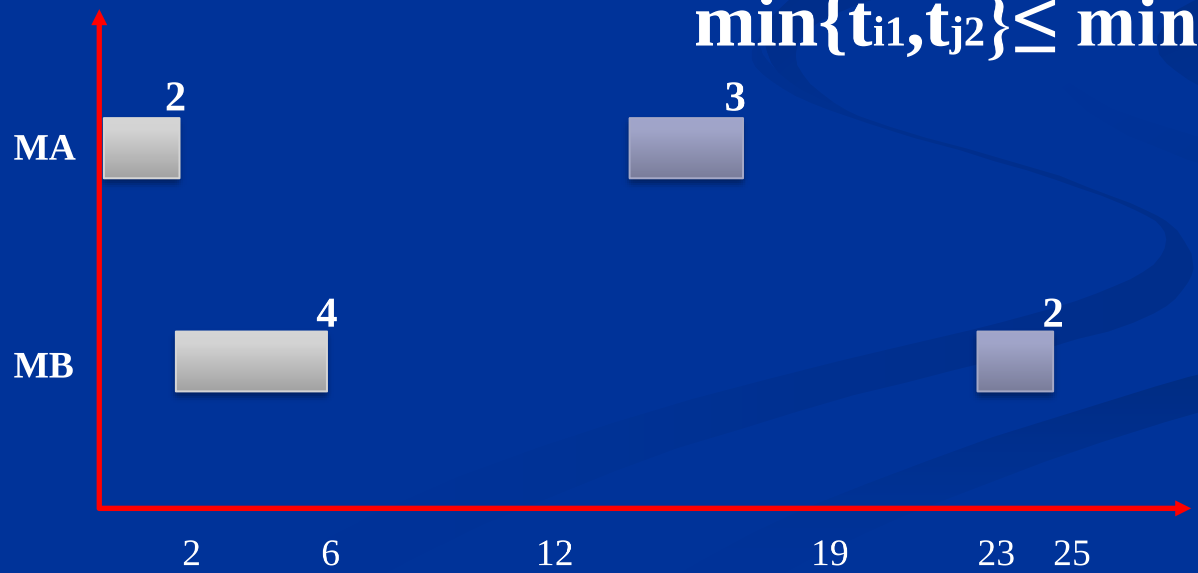
$$\min\{t_{i1},t_{j2}\} \leq \min\{t_{i2},t_{j1}\}$$

	Máquina A	Máquina B
Tarefa 1	**2**	**4**
Tarefa 2	3	2
Tarefa 3	5	4
Tarefa 4	3	6
Tarefa 5	4	7

$$\min\{t_{i1},t_{j2}\} \leq \min\{t_{i2},t_{j1}\}$$

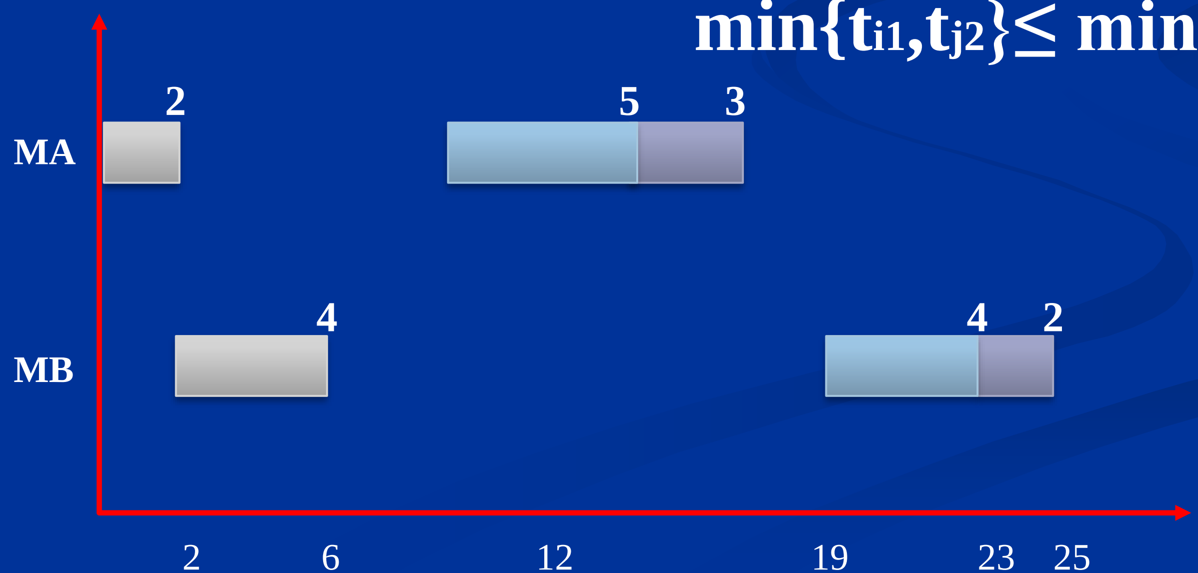
	Máquina A	Máquina B
Tarefa 1	**2**	**4**
Tarefa 2	**3**	**2**
Tarefa 3	5	4
Tarefa 4	3	6
Tarefa 5	4	7

$$\min\{t_{i1},t_{j2}\} \leq \min\{t_{i2},t_{j1}\}$$



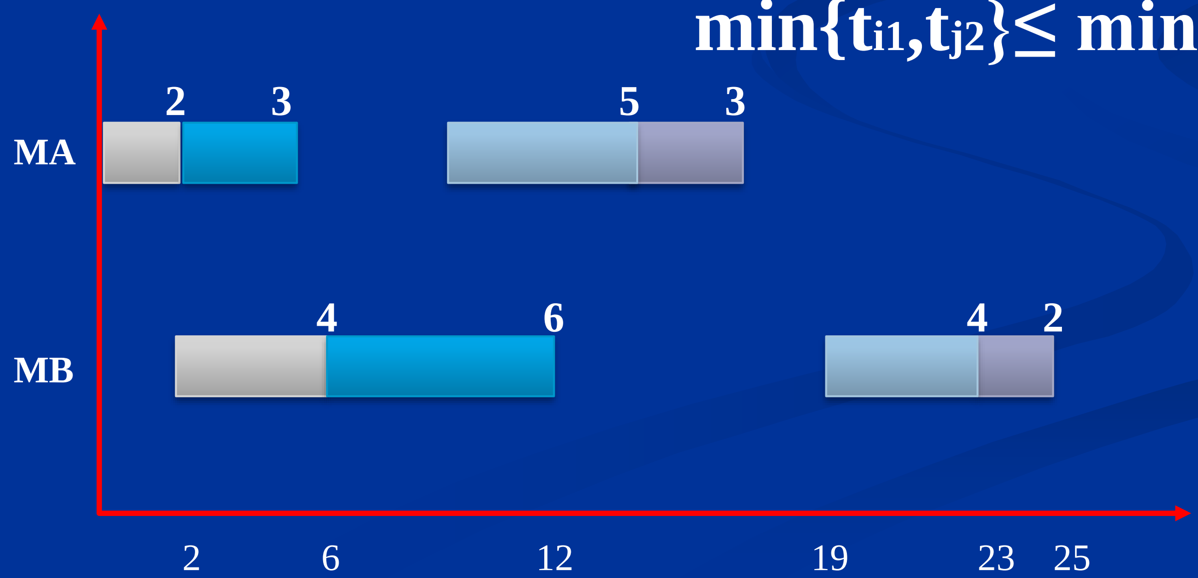
	Máquina A	Máquina B
Tarefa 1	**2**	**4**
Tarefa 2	**3**	**2**
Tarefa 3	**5**	**4**
Tarefa 4	3	6
Tarefa 5	4	7

$$\min\{t_{i1},t_{j2}\} \leq \min\{t_{i2},t_{j1}\}$$



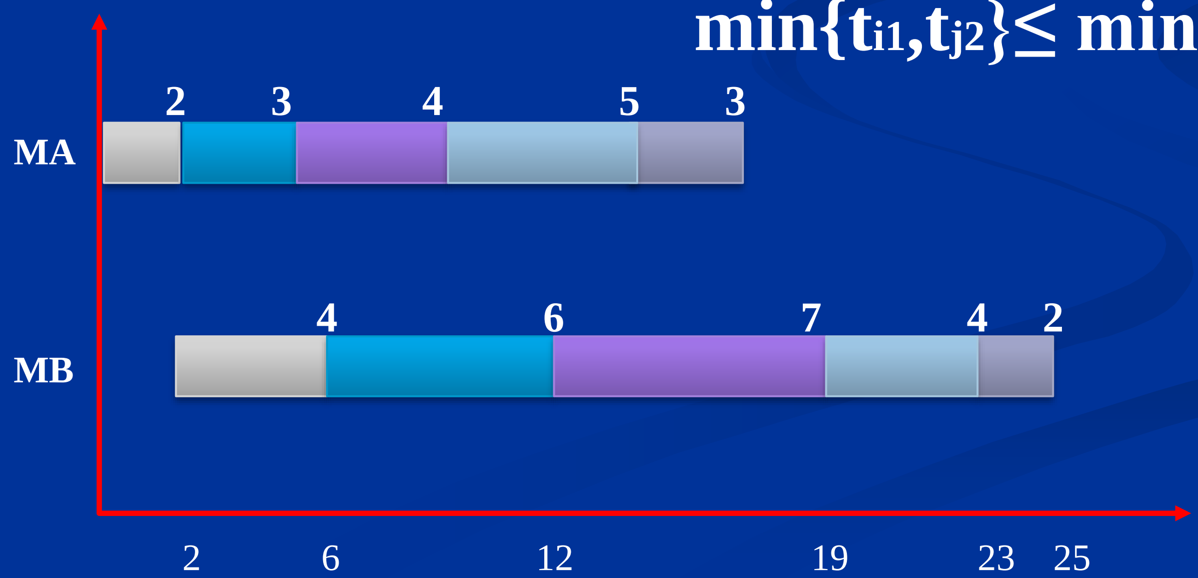
	Máquina A	Máquina B
Tarefa 1	**2**	**4**
Tarefa 2	**3**	**2**
Tarefa 3	**5**	**4**
Tarefa 4	**3**	**6**
Tarefa 5	4	7

$$\min\{t_{i1},t_{j2}\} \leq \min\{t_{i2},t_{j1}\}$$



	Máquina A	Máquina B
Tarefa 1	2	4
Tarefa 2	3	2
Tarefa 3	5	4
Tarefa 4	3	6
Tarefa 5	4	7

$$\min\{t_{i1},t_{j2}\}\leq \min\{t_{i2},t_{j1}\}$$



EXEMPLO GUPTA

Heuristic Algorithms for Multistage Flowshop

Scheduling Problem

1972

Procedimento

Heurística de Gupta

Procedimento heurística de Gupta

1) Formar o par (a, b) a partir das n tarefas disponíveis e calcular C(ab, m) para todas as máquinas “m”, pares a e b. encontrar o C(ab, m) mínimo. Se um único par ab é encontrado, vá para o passo 2(c), caso contrário para o passo 2(a);
2) (a) Considere os pares com vínculo (i, j) e (r, s) – o menor C(ab, m). Selecione o par com o tempo ocioso (*idle*) mínimo acumulado nas máquinas (M – 1, (M – 2) .... até a última máquina de modo a quebrar o empate. Se o vínculo for resolvido, processe o passo 2(c) até encontrar (a,b) como o par selecionado, caso contrário processe o passo 2(b).

Procedimento heurística de Gupta

2) (b) se (i, j) = (s, r) selecione algum par arbitrariamente como par (a, b) e processe o passo 2(c), caso contrário quebre o vínculo a partir da seleção do par com a maior soma dos tempos de processo na máquina M, (M – 1), ... até a máquina a qual é possível resolver o vínculo e chamar o par selecionado (a, b). Processar o passo 2(c).

   (c) considere σ = ab, k = 2. Vá para o passo 3.

3) Entre as tarefas não alocadas, examine cada uma das tarefas e calcule C(σa, m) para todo a e m. Encontre o mínimo C(σa, M). Se um único σa existe, vá para o passo 5; caso contrário vá para o passo 4.

Procedimento heurística de Gupta

4) Considere o vínculo entre σa e σb. Resolva o vínculo em favor de uma tarefa com tempo de processo máximo na máquina M, (M – 1), ..., até a máquina suficiente para quebrar o vínculo. No caso de uma ruptura, selecione o *schedule* parcial com o menor tempo *idle* nas máquinas m para m = (M – 1), M – 2), ... até a máquina suficiente para quebrar o vínculo. Processar o passo 5.

5) Considere k = k + 1, σ = σa.

é K = (n – 2)?

(a) Sim. Vá para o passo 6.

(b) Não. Retorne para o passo 3.

Procedimento heurística de Gupta

6) Gere duas *schedules* (programações) completas através da alocação das duas tarefas remanescentes em ambas as posições das sequências, e aceite o *schedule* que tem o mínimo custo quanto a melhor *schecule*.

Exemplo

- Para compreender a operação ou execução do algoritmo MINIT, considere a Tabela 1.

Tabela 1: Matriz dos tempos de processamento			
m (máquina) / i (tarefa)	**1**	**2**	**3**
	operação		
1	5	8	20
2	6	30	6
3	30	4	5
4	2	5	3
5	3	10	4
6	4	1	4

tarefa

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	**1**	**2**	**3**
	operação		
1	5	8	20
2	6	30	6
3	30	4	5
4	2	5	3
5	3	10	4
6	4	1	4

tarefa

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

Exemplo

- Há seis tarefas e contudo 30 pares (número de pares = n * (n – 1) ⇒ 6 * (6 – 1) = 6 * 5 = 30 pares). Estes 30 pares são formados e os tempos de ociosidade e de execução são calculados de acordo com a Tabela 2. Os pares (4, 6) e (6, 4) tem um tempo de ociosidade mínimo na máquina 3. Visto que **C(46,2) < C(64,2)** (Etapa 2(a) do algoritmo acima), par (4,6) é selecionado **σ = 46**. A partir dos passos 3, 4 e 5 do algoritmo, os resultados da Tabela 3 são obtidos.

C(46,2) = 2 < C(64,2) = 5

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	**8** 2 + 5 + 1 (4,1 ⇒ 4,2 ⇒ 6,2)	14	C(46,2) = 2 **2**	7

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação			
4	2	5	3
6	4	1	4

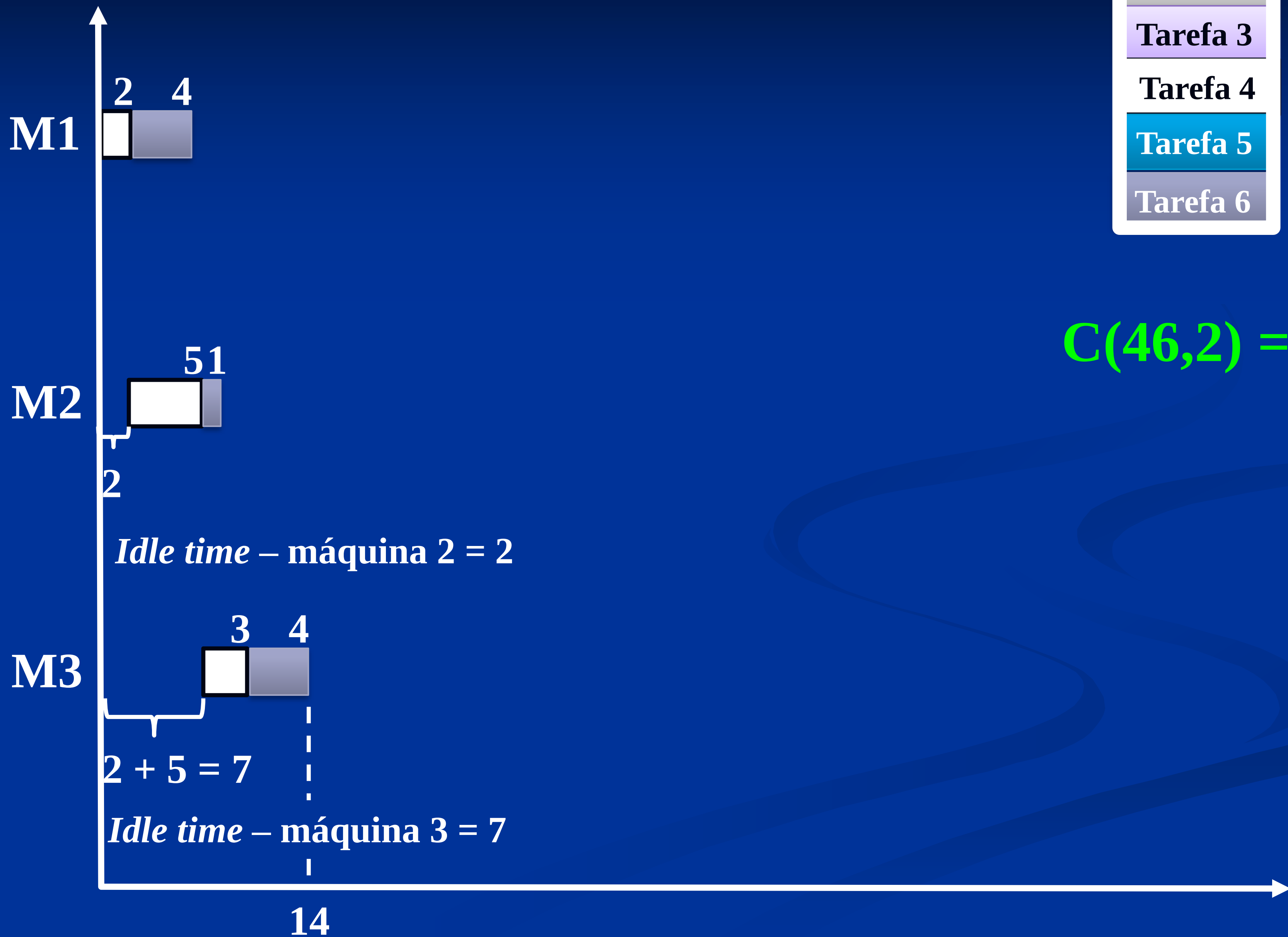
tarefa

4,1	4,2	4,3
6,1	6,2	6,3

Se $t_{6,1} \leq t_{4,2} \Rightarrow$ *idle* = $t_{4,1}$

Quinto par

Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6



C(46,2) = 2

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	**11** 6 + 5	14	**C(64,2) = 5** 5	**7**
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	**1**	**2**	**3**
	operação →		
6	4	1	4
4	2	5	3

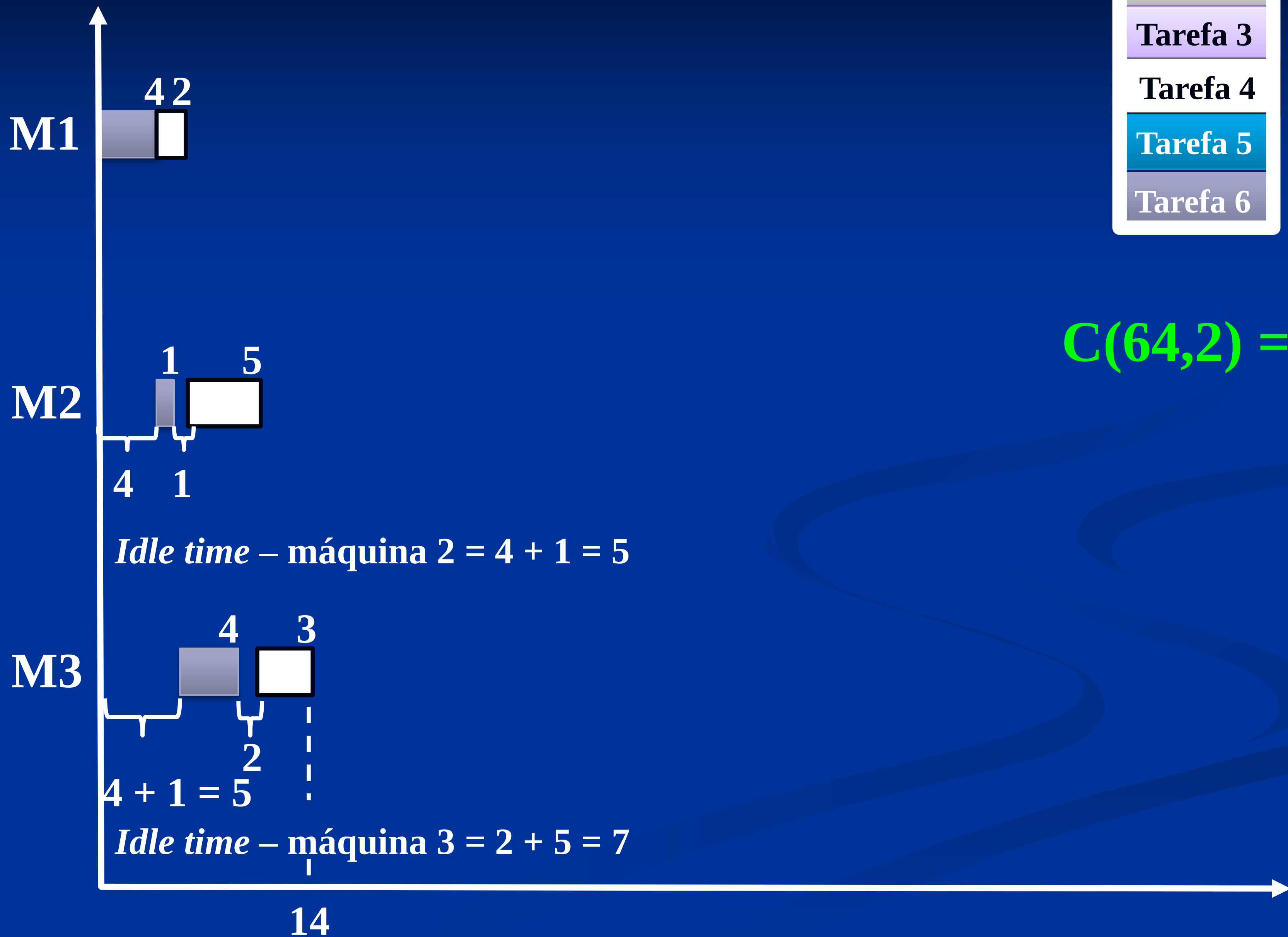
tarefa ↓

6,1	6,2	6,3
4,1	4,2	4,3

Se $t_{4,1} > t_{6,2} \Rightarrow$
idle **$= (t_{4,1} - t_{6,2}) + t_{6,1}$**
$(2 - 1) + 4 = 5$

Quarto par

Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6



C(64,2) = 5

Passo 1 do procedimento do algoritmo MINIT

Passo 1 – Formar o par (a, b) entre as tarefas disponíveis e calcular C(ab, m) para todas as máquinas m, a e b. Encontrar o C(ab, m) mínimo. Se um único par ab é encontrado, vá para o passo 2(c), caso contrário vá para o passo 2(a).

1ª série

Tarefa 1

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(**1**,2)	5 + 6 = 11	11 + **30** = 41 (5 + 8) + **30** = **43**	43 + **6** = **49** (5 + 8 + 20) + **6** = 39		
(**1**,3)	5 + 30 = 35	35 + **4** = **39** (5 + 8) + **4** = 17	39 + **5** = **44** (5 + 8 + 20) + **5** = 38		
(**1**,4)	5 + 2 = 7	7 + **5** = 12 (5 + 8) + **5** = **18**	18 + **3** = 21 (5 + 8 + 20) + **3** = **36**		
(**1**,5)	5 + 3 = 8	8 + **10** = 18 (5 + 8) + **10** = **23**	23 + **4** = 27 (5 + 8 + 20) + **4** = **37**		
(**1**,6)	5 + 4 = 9	9 + **1** = 10 (5 + 8) + **1** = **14**	14 + **4** = 18 (5 + 8 + 20) + **4** = **37**		

2ª operação da 2ª tarefa

Tempo da 3ª operação da 2ª tarefa

Soma dos tempos de operação da 1ª tarefa

Repete a 2ª opeação da 2ª tarefa

maior valor

1ª operação das tarefas

Soma da 1ª e da 2ª operação da 1ª tarefa

Máquina 2

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43 5 + 8 + 30 (1,1 ⇒ 1,2 ⇒ 2,2)	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1	2	3
	operação		
tarefa 1	5	8	20
2	6	30	6

	1,1	1,2	1,3
	2,1	2,2	2,3

Se $t_{2,1} \leq t_{1,2} \Rightarrow$ *idle* $= t_{1,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	**18** **5 + 8 + 5 (1,1 ⇒ 1,2 ⇒ 4,2)**	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
1	5	8	20
4	2	5	3

tarefa

1,1	1,2	1,3
4,1	4,2	4,3

Se $t_{4,1} \leq t_{1,2} \Rightarrow$ *idle* $= t_{1,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	**23** **5 + 8 + 10 (1,1 ⇒ 1,2 ⇒ 5,2)**	37	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** operação	**2**	**3**
tarefa 1	5	8	20
5	3	10	4

1,1	1,2	1,3
5,1	5,2	5,3

Se $t_{5,1} \leq t_{1,2} \Rightarrow$ *idle* $= t_{1,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	**14** 5 + 8 + 1 (1,1 ⇒ 1,2 ⇒ 6,2)	37	5	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação →			
tarefa 1	5	8	20
tarefa 6	4	1	4

	1	2	3
	1,1	1,2	1,3
	6,1	6,2	6,3

Se $t_{6,1} \leq t_{1,2} \Rightarrow$ *idle* $= t_{1,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43	49	5	23
(1,3)	35	**39** 35 + 4	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento

i (tarefa) / m (máquina)	1	2	3
tarefa / operação			
1	5	8	20
3	30	4	5

1,1	1,2	1,3
3,1	3,2	3,3

Se $t_{3,1} > t_{1,2} \Rightarrow$

idle **$= (t_{3,1} - t_{1,2}) + t_{1,1}$**

$(30 - 8) + 5 = 27$

Máquina 3

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43	**49** 43 + 6	5	**23**
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	1 operação	2	3
tarefa 1	5	8	2,2 > 1,3 20
2	6	30	6

1,1	1,2	1,3
2,1	2,2	2,3

$idle_2 = \{[(t_{1,2} + t_{2,2}) + idle_1] - [(t_{1,1} + t_{1,2}) + t_{1,3}]\} + (t_{1,1} + t_{1,2}) = [(8 + 30) + 5] - [(5 + 8) + 20] + 5 + 8 = 23$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43	49	5	23
(1,3)	35	39	**44** (8 + 4 + 27) + 5	27	**19**
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
	operação		
tarefa 1	5	8	20
3	30	4	5

3,2 < 1,3

1,1	1,2	1,3
3,1	3,2	3,3

$$idle_2 = \{[(t_{1,2} + t_{3,2}) + idle_1] - [(t_{1,1} + t_{1,2}) + t_{1,3}]\} + (t_{1,1} + t_{1,2}) = \underbrace{[(8 + 4) + 27]} - [(5 + 8) + 20] + 5 + 8 = 19$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	**36** (5 + 8 + 20) + 3	5	**13**
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	**1**	**2**	**3**
	operação		
1	5	8	4,2 < 1,3 20
4	2	5	3

tarefa

1,1	1,2	1,3
4,1	4,2	4,3

~~$idle_2 = \{[(t_{1,2} + t_{4,2}) + idle_1] - [(t_{1,1} + t_{1,2}) + t_{1,3}]\} + (t_{1,1} + t_{1,2}) = [(8 + 5) + 5] - [(5 + 8) + 20] + 5 + 8 = 18 - 33 + 5 + 8 = -2$~~

Se: $[(t_{1,2} + t_{4,2}) + idle_1] \leq [(t_{1,1} + t_{1,2}) + t_{1,3}] \Rightarrow [(8 + 5) + 5] < [(5 + 8) + 20] \Rightarrow 18 < 33$, *sendo nesse caso:* $idle_2 = (t_{1,1} + t_{1,2}) = (5 + 8) = 13$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37 (5 + 8 + 20) + 4	5	13
(1,6)	9	14	37	5	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
	operação		
tarefa 1	5	8	5,2 < 1,3 20
5	3	10	4

1,1	1,2	1,3
5,1	5,2	5,3

Se: $[(t_{1,2} + t_{5,2}) + idle_1] \leq [(t_{1,1} + t_{1,2}) + t_{1,3}] \Rightarrow [(8 + 10) + 5] < [(5 + 8) + 20] \Rightarrow 23 < 33$

$idle_2 = (t_{1,1} + t_{1,2}) = (5 + 8) = 13$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	**37** (5 + 8 + 20) + 4	5	**13**

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** operação	**2**	**3**
1	5	8	6,2 < 1,3 20
6	4	1	4

tarefa

1,1	1,2	1,3
6,1	6,2	6,3

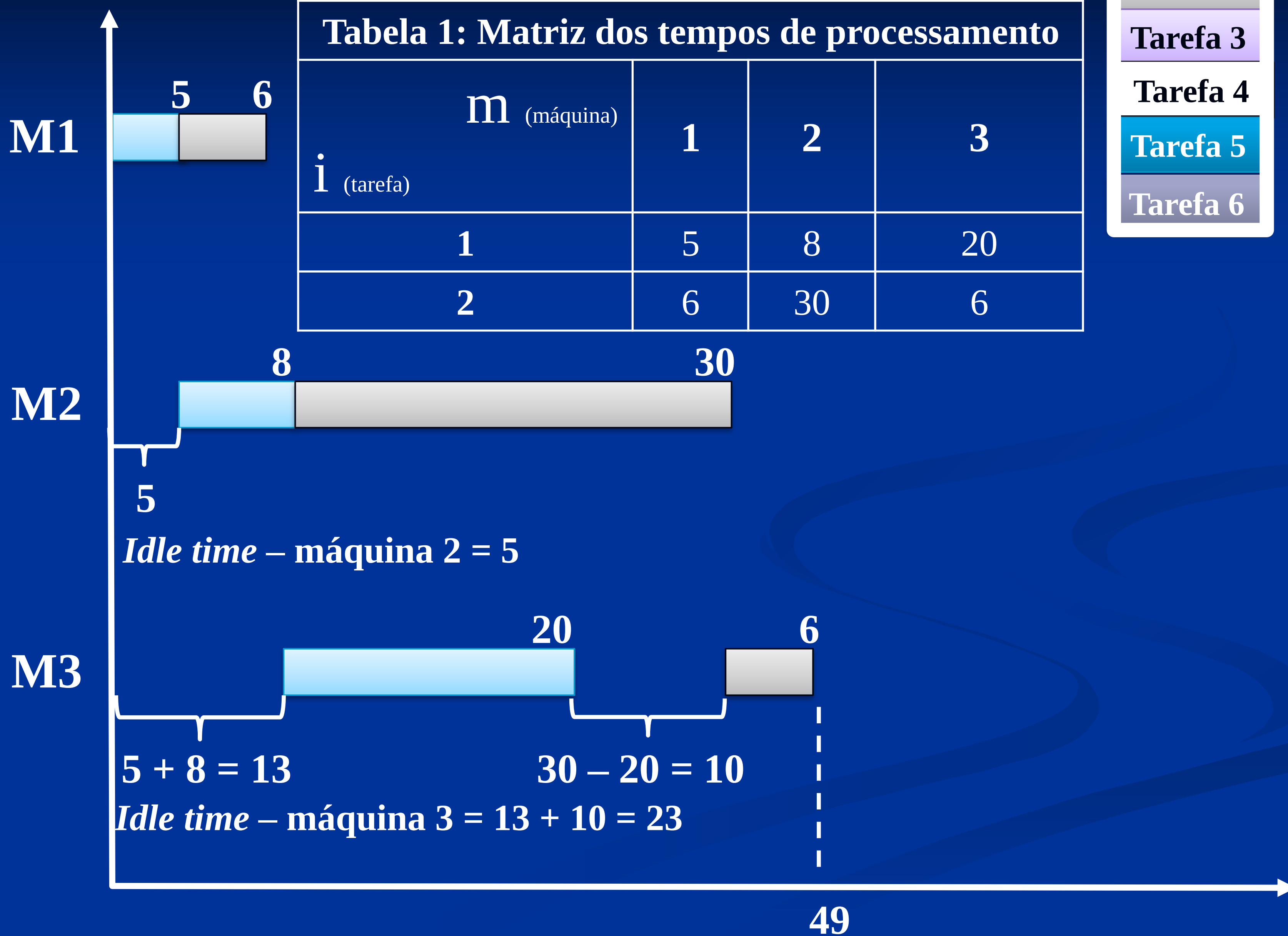
Se: $[(t_{1,2} + t_{6,2}) + idle_1] \leq [(t_{1,1} + t_{1,2}) + t_{1,3}] \Rightarrow [(8 + 1) + 5] < [(5 + 8) + 20] \Rightarrow 14 < 33$

$idle_2 = (t_{1,1} + t_{1,2}) = (5 + 8) = 13$

Primeiro par

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
1	5	8	20
2	6	30	6



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

Segundo par

- Tarefa 1
- Tarefa 2
- Tarefa 3
- Tarefa 4
- Tarefa 5
- Tarefa 6

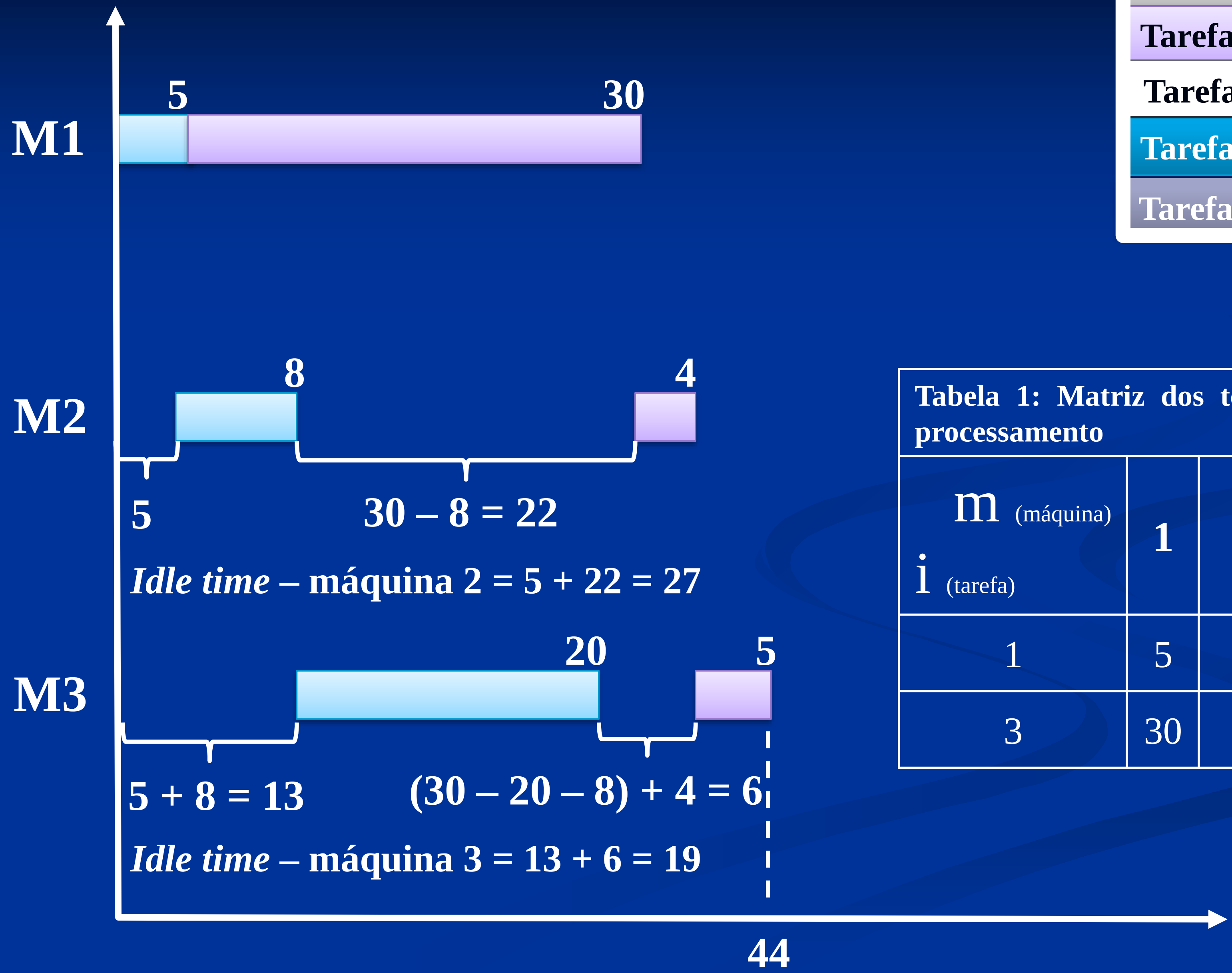
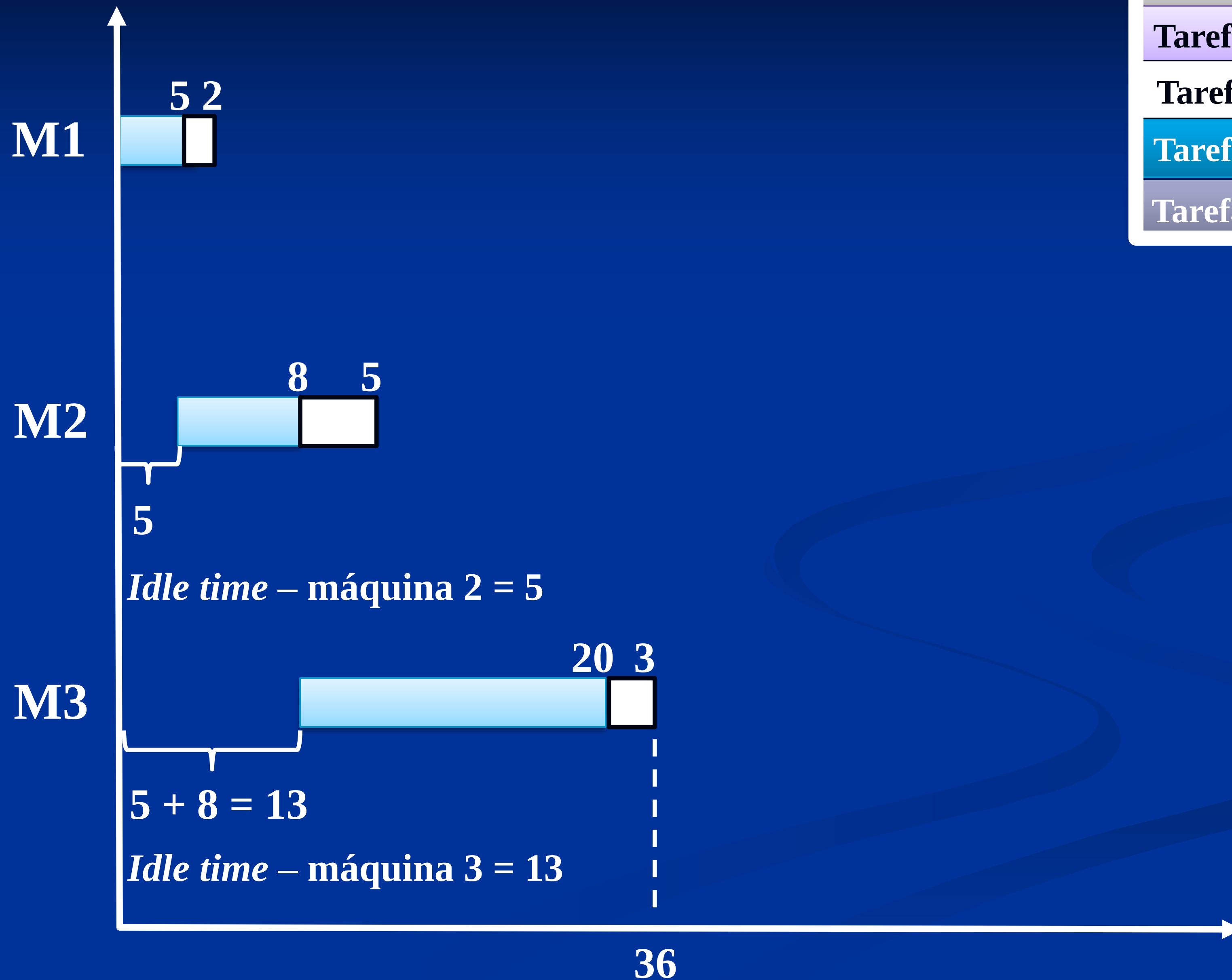


Tabela 1: Matriz dos tempos de processamento

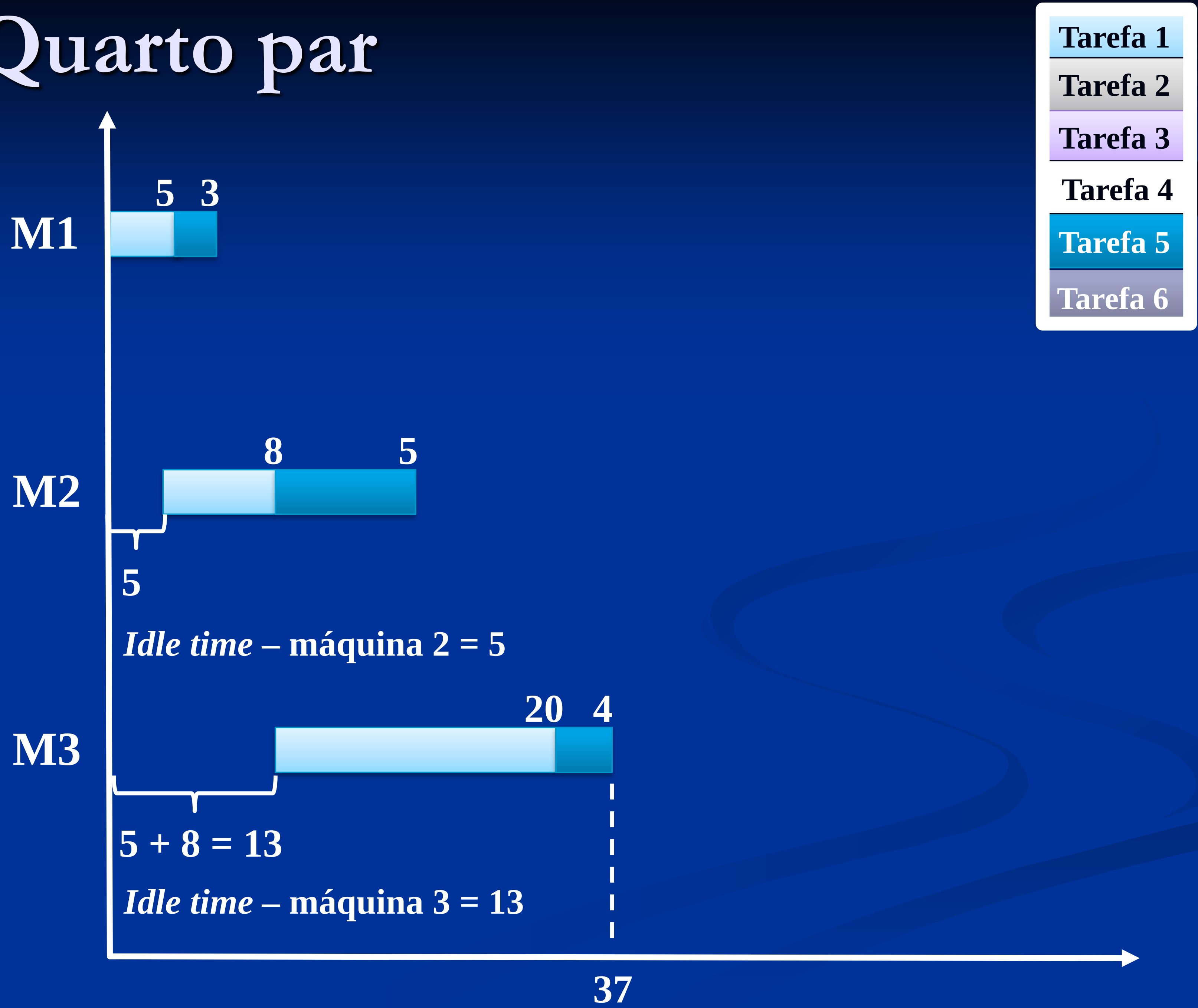
m (máquina) / i (tarefa)	**1**	**2**	**3**
1	5	8	20
3	30	4	5

Terceiro par

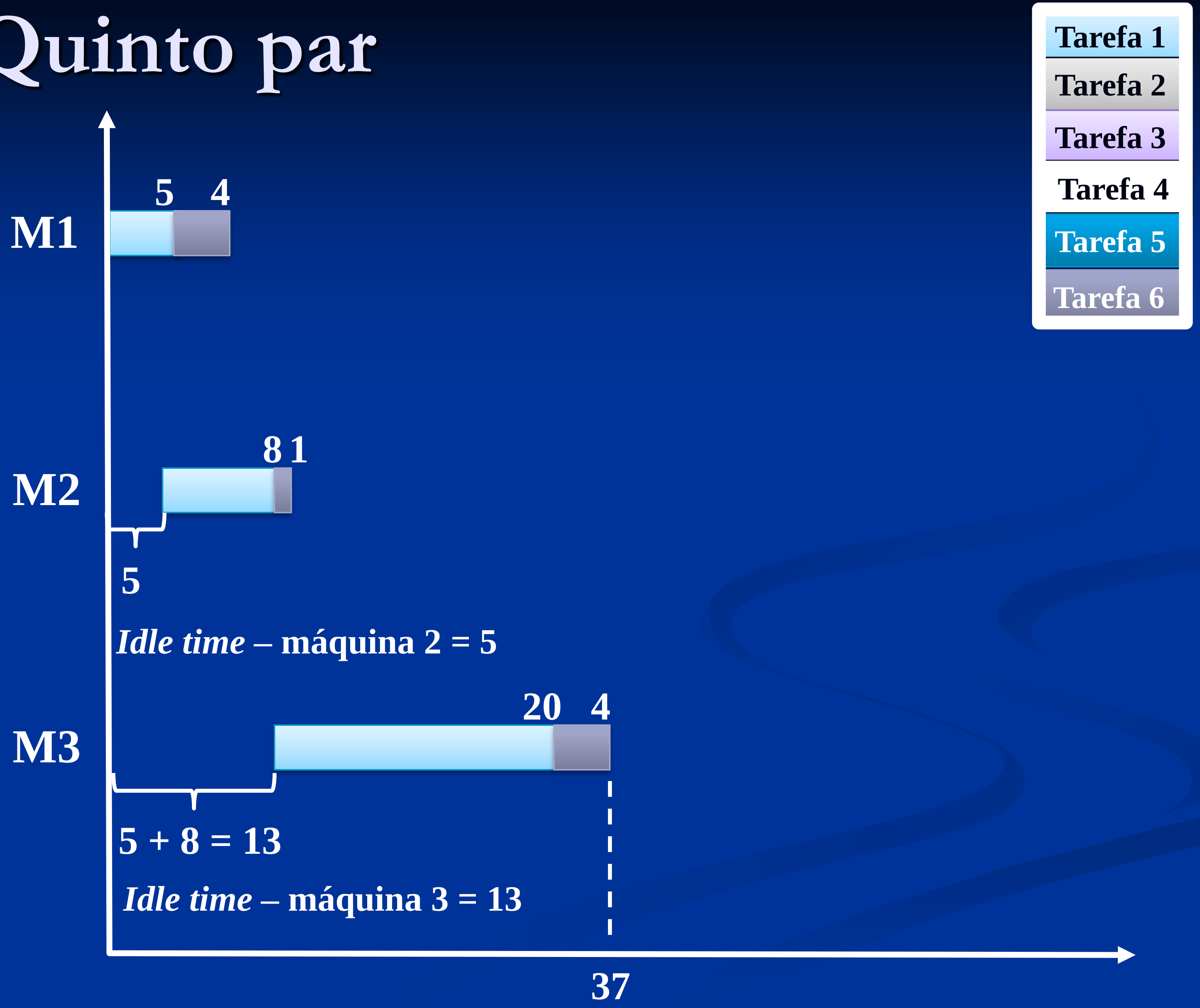
Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6



Quarto par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
5 3
M2
8 5
5
Idle time – máquina 2 = 5
M3
20 4
5 + 8 = 13
Idle time – máquina 3 = 13
37

Quinto par



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
5
4
M2
8
1
5
Idle time – máquina 2 = 5
M3
20
4
5 + 8 = 13
Idle time – máquina 3 = 13
37

2ª série

Tarefa 2

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	6 + 5 = 11	11 + **8** = 19 (6 + 30) + **8** = **44**	44 + **20** = **64** (6 + 30 + 6) + **20** = 62		
(2,3)	6 + 30 = 36	36 + **4** = **40** (6 + 30) + **4** = **40**	40 + **5** = 45 (6 + 30 + 6) + **5** = **47**		
(2,4)	6 + 2 = 8	8 + **5** = 13 (6 + 30) + **5** = **41**	41 + **3** = 44 (6 + 30 + 6) + **3** = **45**		
(2,5)	6 + 3 = 9	9 + **10** = 19 (6 + 30) + **10** = **46**	46 + **4** = **50** (6 + 30 + 6) + **4** = 46		
(2,6)	6 + 4 = 10	10 + **1** = 11 (6 + 30) + **1** = **37**	37 + **4** = 41 (6 + 30 + 6) + **4** = **46**		

2ª operação da 2ª tarefa

Tempo da 3ª operação da 2ª tarefa

Soma dos tempos de operação da 1ª tarefa

ou

Repete a 2ª opeação da 2ª tarefa

maior valor

1ª operação das tarefas

Soma da 1ª e da 2ª operação da 1ª tarefa

Máquina 2

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(2,1)	11	**44** 6 + 30 + 8 (2,1 ⇒ 2,2 ⇒ 1,2)	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
tarefa / operação			
2	6	30	6
1	5	8	20

2,1	2,2	2,3
1,1	1,2	1,3

Se $t_{1,1} \leq t_{2,2} \Rightarrow$ *idle* $= t_{2,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	64	6	38
(2,3)	36	40 6 + 30 + 4 (2,1 ⇒ 2,2 ⇒ 3,2)	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento

i (tarefa) / m (máquina)	1	2	3
operação →			
tarefa 2	6	30	6
tarefa 3	30	4	5

2,1	2,2	2,3
3,1	3,2	3,3

Se $t_{3,1} \leq t_{2,2} \Rightarrow$ *idle* $= t_{2,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	**41** 6 + 30 + 5 (2,1 ⇒ 2,2 ⇒ 4,2)	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1	2	3
tarefa / operação			
2	6	30	6
4	2	5	3

2,1	2,2	2,3
4,1	4,2	4,3

Se $t_{4,1} \leq t_{2,2} \Rightarrow$ *idle* $= t_{2,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	**46** 6 + 30 + 10 (2,1 ⇒ 2,2 ⇒5,2)	50	6	40
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1 operação	2	3
2	6	30	6
5	3	10	4

tarefa

2,1	2,2	2,3
5,1	5,2	5,3

Se $t_{5,1} \leq t_{2,2} \Rightarrow$ *idle* $= t_{2,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	**37** 6 + 30 + 1 (2,1 ⇒ 2,2 ⇒ 6,2)	46	6	36

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação →			
tarefa 2	6	30	6
tarefa 6	4	1	4

2,1	2,2	2,3
6,1	6,2	6,3

Se $t_{6,1} \leq t_{2,2} \Rightarrow$ *idle* $= t_{2,1}$

Máquina 3

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	**64** 44 + 20	6	**38**
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1	2	3
	operação		
tarefa 2	6	30	1,2 > 2,3 6
1	5	8	20

2,1	2,2	2,3
1,1	1,2	1,3

$idle_2 = \{[(t_{2,2} + t_{1,2}) + idle_1] – [(t_{2,1} + t_{2,2}) + t_{2,3}]\} + (t_{2,1} + t_{2,2}) = [30 + 8) + 6] – [(6 + 30) + 6] + 6 + 30 = 38$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(2,1)	11	44	64	6	38
(2,3)	36	40	**47** (6 + 30 + 6) + 5	6	**36**
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
2	6	30	**3,2 < 2,3** 6
3	30	4	5

tarefa

2,1	2,2	2,3
3,1	3,2	3,3

Se: $[(t_{2,2} + t_{3,2}) + idle_1] \leq [(t_{2,1} + t_{2,2}) + t_{2,3}] \Rightarrow [(30 + 4) + 6] < \underbrace{[(6 + 30) + 6]} \Rightarrow 40 < 42$

$idle_2 = (t_{2,1} + t_{2,2}) = (6 + 30) = 36$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45 (6 + 30 + 6) + 3	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1	2	3
	operação		
2	6	30	4,2 < 2,3 6
4	2	5	3

tarefa

2,1	2,2	2,3
4,1	4,2	4,3

Se: $[(t_{2,2} + t_{4,2}) + idle_1] \leq [(t_{2,1} + t_{2,2}) + t_{2,3}] \Rightarrow [(6 + 5) + 6] < \underbrace{[(6 + 30) + 6]} \Rightarrow 17 < 42$

$idle_2 = (t_{2,1} + t_{2,2}) = (6 + 30) = 36$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	**50** 46 + 4	6	**40**
(2,6)	10	37	46	6	36

Tabela 1: Matriz dos tempos de processamento			
m (máquina) / i (tarefa)	**1** operação	**2**	**3**
tarefa 2	6	30	5,2 > 2,3 6
5	3	10	4

2,1	2,2	2,3
5,1	5,2	5,3

$idle_2 = \{[(t_{2,2} + t_{5,2}) + idle_1] - [(t_{2,1} + t_{2,2}) + t_{2,3}]\} + (t_{2,1} + t_{2,2}) = [30 + 10) + 6] - [(6 + 30) + 6] + 6 + 30 = 40$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46 (6 + 30 + 6) + 4	6	36

Tabela 1: Matriz dos tempos de processamento

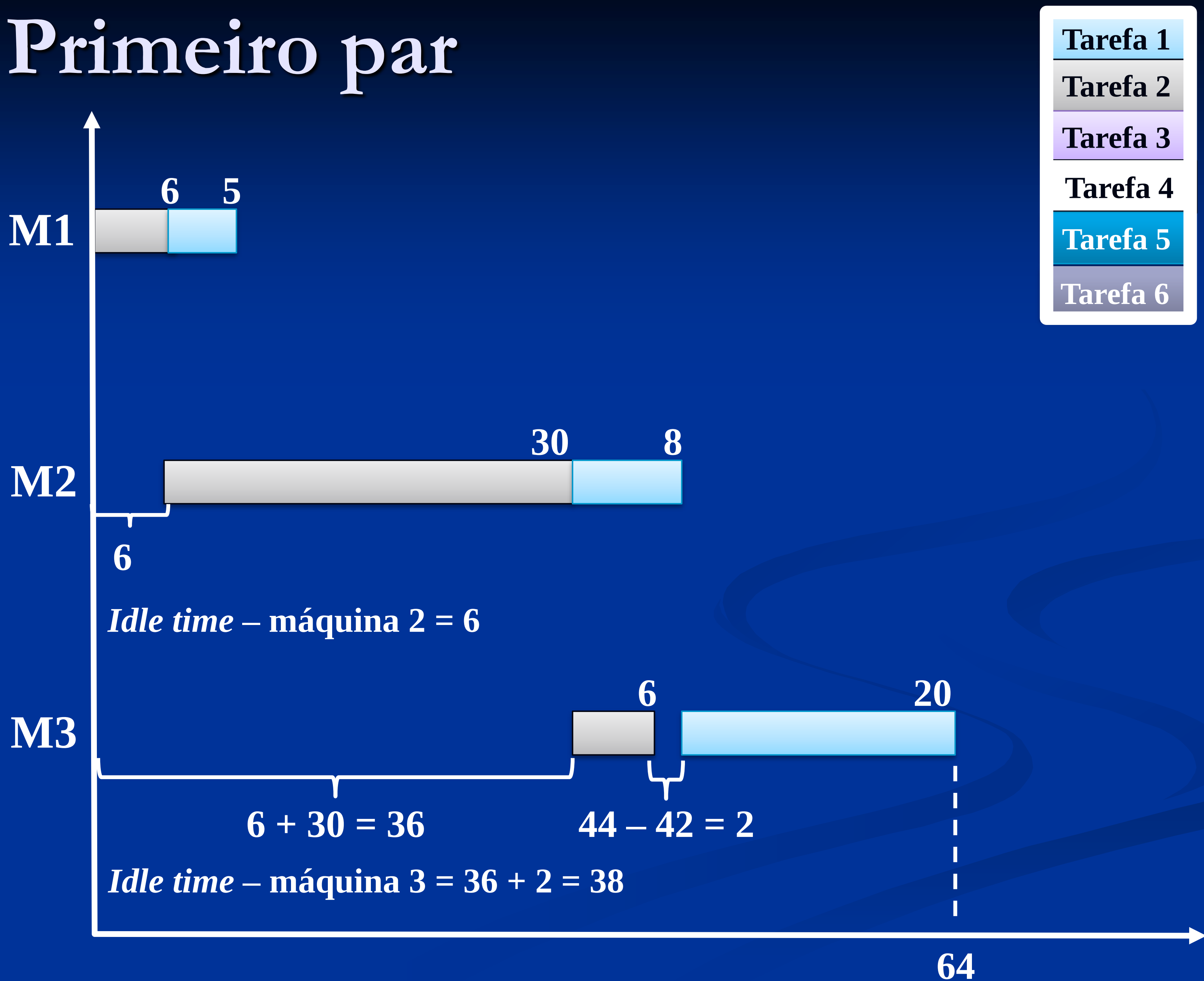
m (máquina) i (tarefa)	1	2	3
	operação		
tarefa 2	6	30	6,2 < 2,3 6
6	4	1	4

2,1	2,2	2,3
6,1	6,2	6,3

Se: $[(t_{2,2} + t_{6,2}) + idle_1] \leq [(t_{2,1} + t_{2,2}) + t_{2,3}] \Rightarrow [(30 + 1) + 6] < \underbrace{[(6 + 30) + 6]} \Rightarrow 37 < 42$

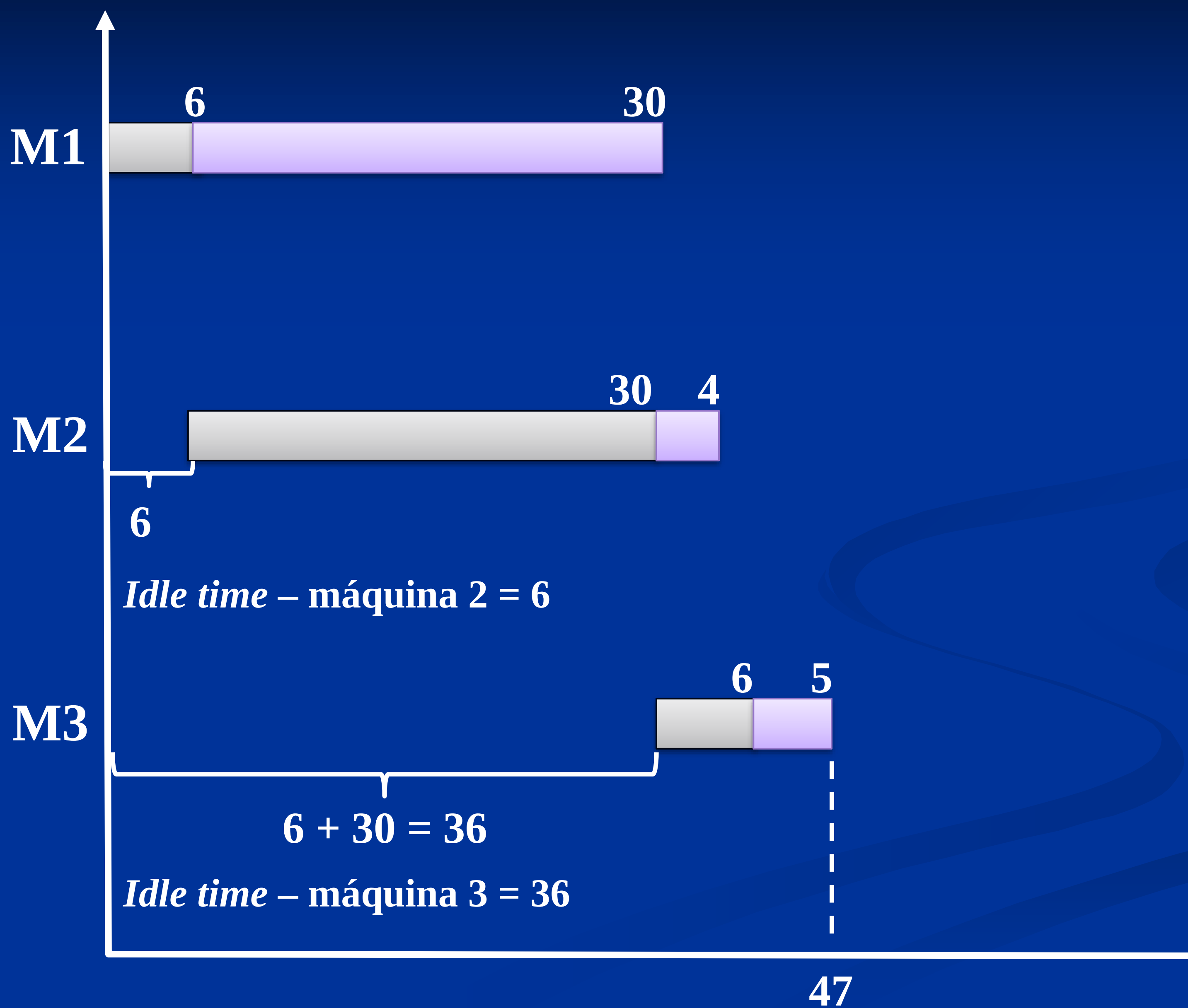
$idle_2 = (t_{2,1} + t_{2,2}) = (6 + 30) = 36$

Primeiro par

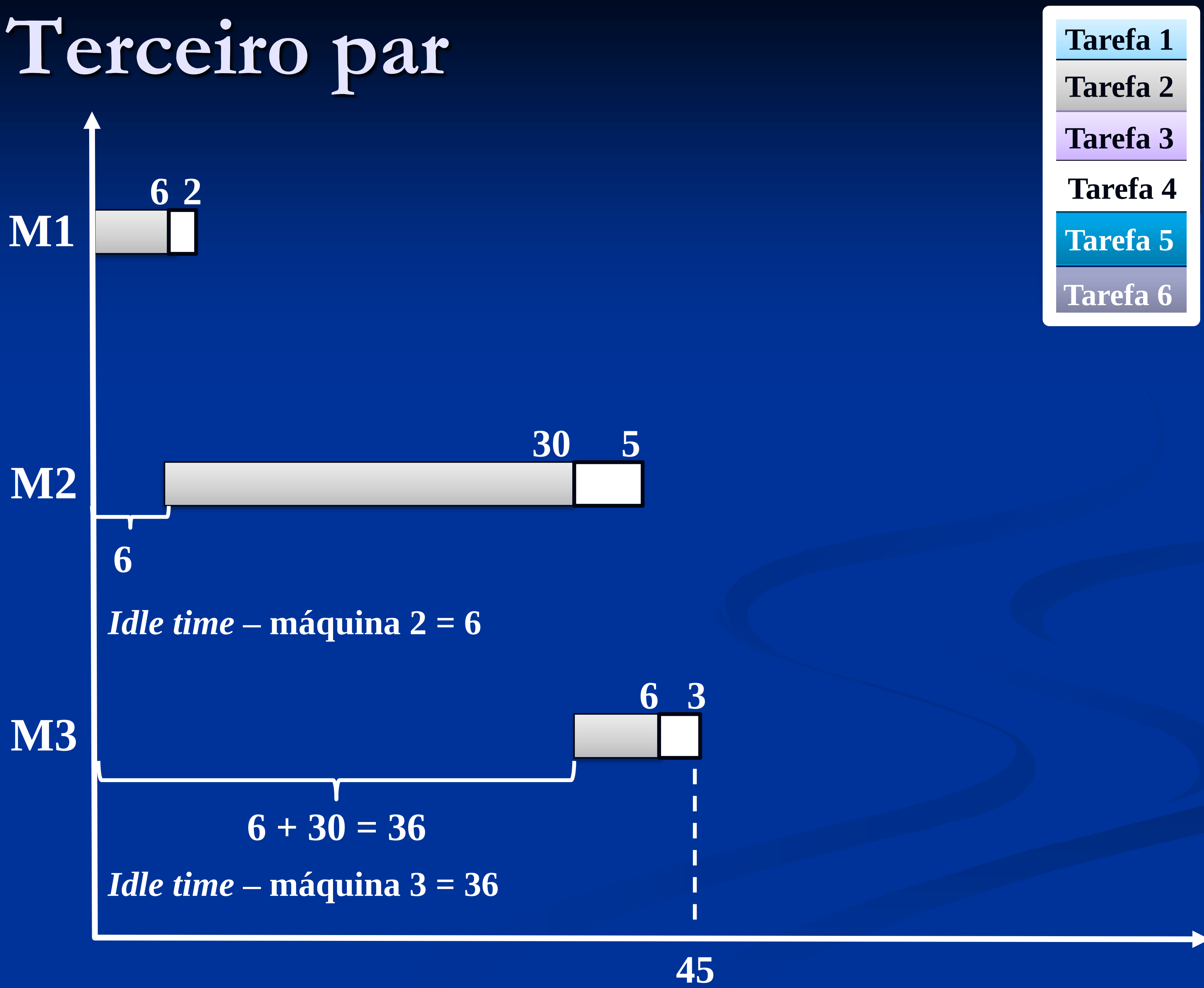


Segundo par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

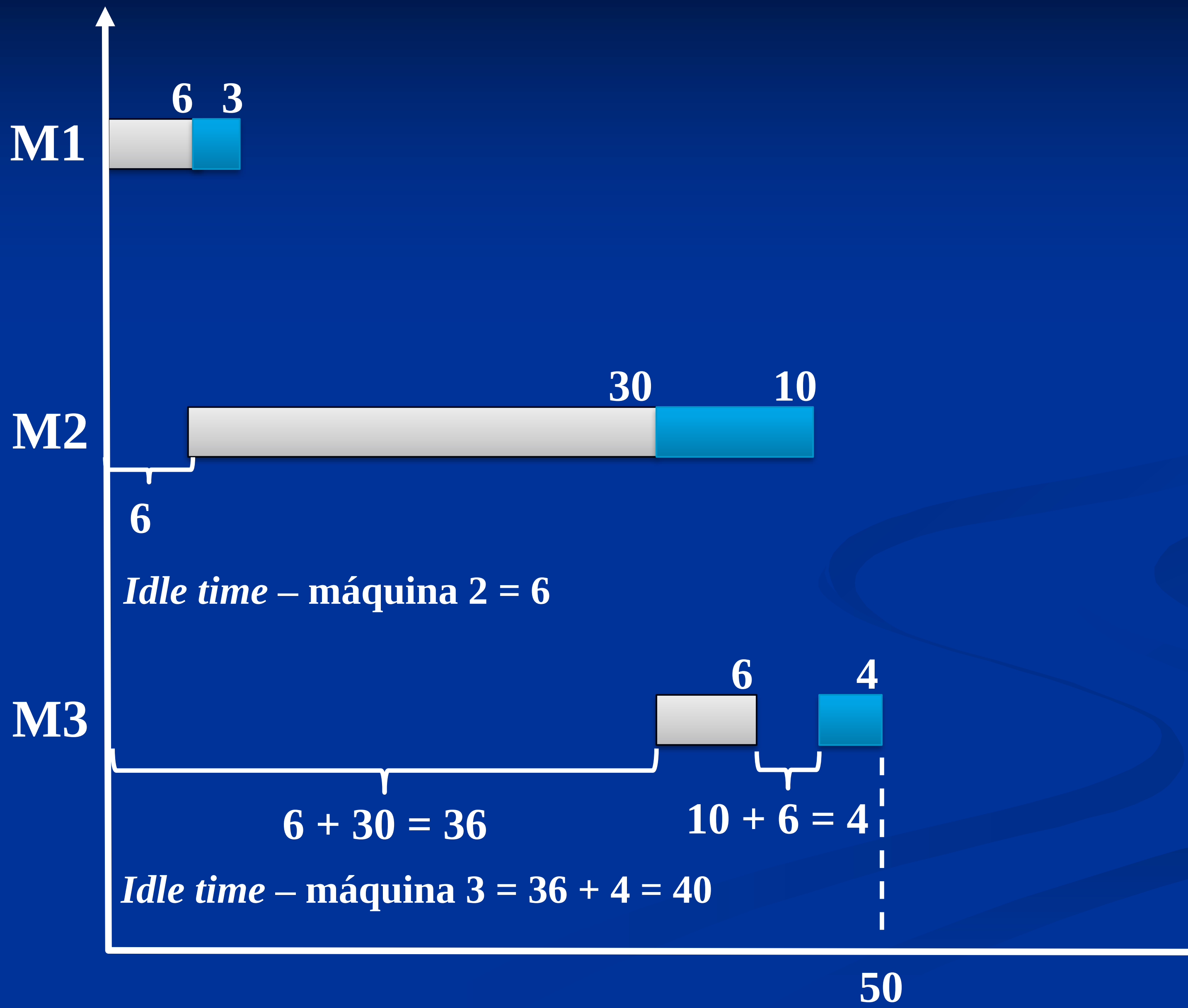

M1
6
30
M2
30
4
6
Idle time – máquina 2 = 6
M3
6
5
6 + 30 = 36
Idle time – máquina 3 = 36
47

Terceiro par



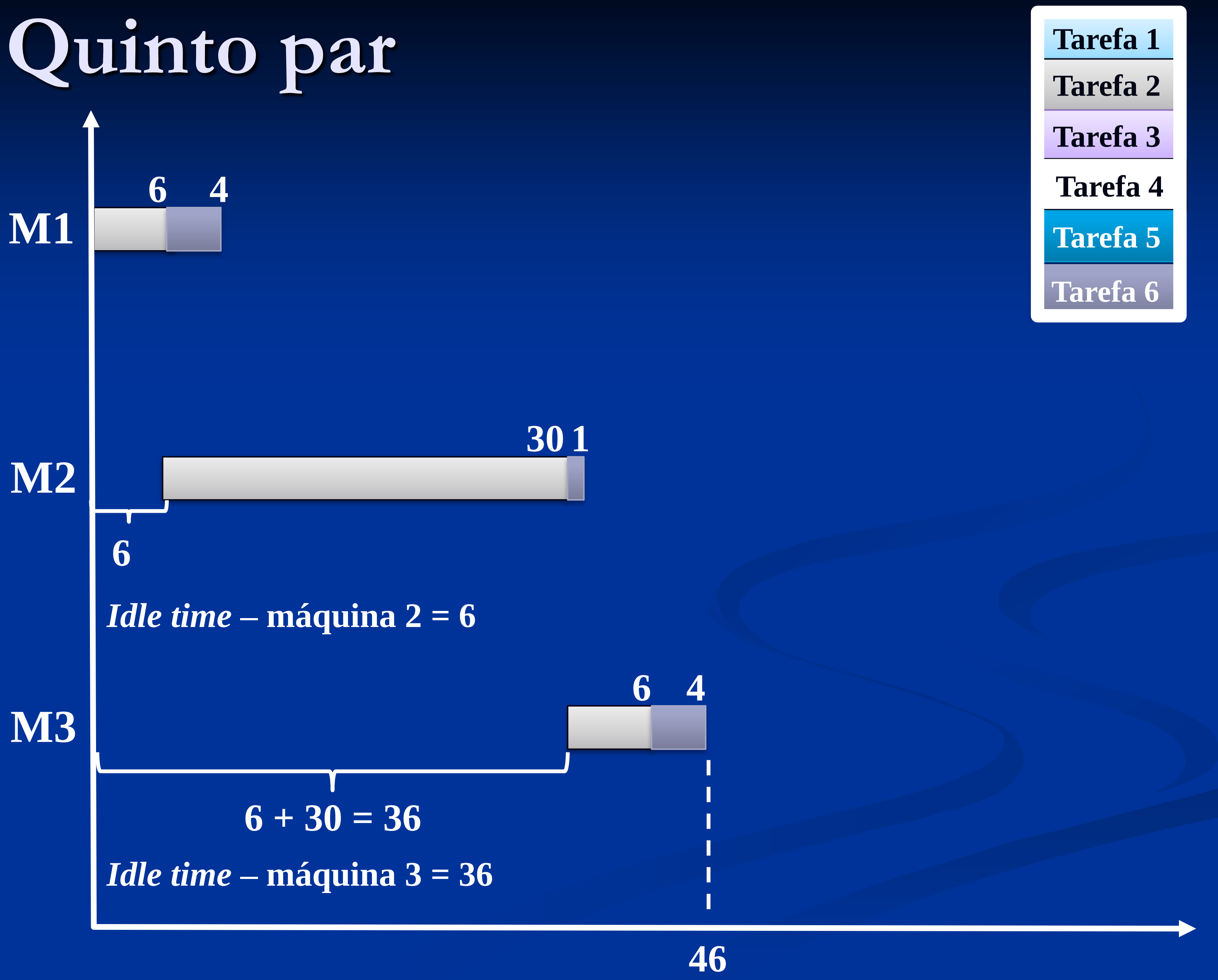
Quarto par

Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6



M1
6
3
M2
30
10
6
Idle time – máquina 2 = 6
M3
6
4
6 + 30 = 36
10 + 6 = 4
Idle time – máquina 3 = 36 + 4 = 40
50

Quinto par



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
6
4
M2
30
1
6
Idle time – máquina 2 = 6
M3
6
4
6 + 30 = 36
Idle time – máquina 3 = 36
46

3ª série

Tarefa 3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	30 + 5 = 35	35 + **8** = **43** (30 + 4) + **8** = 42	43 + **20** = **63** (30 + 4 + 5) + **20** = 59		
(3,2)	30 + 6 = 36	36 + **30** = **66** (30 + 4) + **30** = 64	66 + **6** = **72** (30 + 4 + 5) + **6** = 45		
(3,4)	30 + 2 = 32	32 + **5** = 37 (30 + 4) + **5** = **39**	39 + **3** = **42** (30 + 4 + 5) + **3** = **42**		
(3,5)	30 + 3 = 33	33 + **10** = 43 (30 + 4) + **10** = **44**	44 + **4** = **48** (30 + 4 + 5) + **4** = 43		
(3,6)	30 + 4 = 34	34 + **1** = **35** (30 + 4) + **1** = **35**	35 + **4** = 39 (30 + 4 + 5) + **4** = **43**		

2ª operação da 2ª tarefa

Tempo da 3ª operação da 2ª tarefa

Soma dos tempos de operação da 1ª tarefa

Repete a 2ª opeação da 2ª tarefa

maior valor

ou

1ª operação das tarefas

Soma da 1ª e da 2ª operação da 1ª tarefa

Máquina 2

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	35	43 35 + 8	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1 **operação**	2	3
3	30	4	5
1	5	8	20

tarefa

3,1	3,2	3,3
1,1	1,2	1,3

Se $t_{1,1} > t_{3,2} \Rightarrow$
idle **$= (t_{1,1} - t_{3,2}) + t_{3,1}$**
$(5 - 4) + 30 = 31$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	35	43	63	31	38
(3,2)	36	**66** 36 + 30	72	**32**	61
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação →			
tarefa 3	30	4	5
tarefa 2	6	30	6

3,1	3,2	3,3
2,1	2,2	2,3

Se $t_{2,1} > t_{3,2} \Rightarrow$

idle $= (t_{2,1} - t_{3,2}) + t_{3,1}$

(6 – 4) + 30 = 32

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	**39** 30 + 4 + 5 (3,1 ⇒ 3,2 ⇒ 4,2)	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação →			
tarefa 3	30	4	5
tarefa 4	2	5	3

3,1	3,2	3,3
4,1	4,2	4,3

Se $t_{4,1} \leq t_{3,2} \Rightarrow$ *idle* $= t_{3,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	**44** 30 + 4 + 10 (3,1 ⇒ 3,2 ⇒5,2)	48	30	39
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
	operação		
tarefa 3	30	4	5
5	3	10	4

3,1	3,2	3,3
5,1	5,2	5,3

Se $t_{5,1} \leq t_{3,2} \Rightarrow$ *idle* $= t_{3,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	**35** **30 + 4 + 1 (3,1 ⇒ 3,2 ⇒ 6,2)**	43	30	34

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação** →	**2**	**3**
tarefa ↓ 3	30	4	5
6	4	1	4

Se $t_{5,1} \leq t_{3,2} \Rightarrow$ *idle* = $t_{3,1}$	1,1	1,2	1,3
	3,1	3,2	3,3

Máquina 3

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	35	43	**63** 43 + 20	31	**38**
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
	operação		
3	30	4	1,2 > 3,3 5
1	5	8	20

tarefa

3,1	3,2	3,3
1,1	1,2	1,3

$idle_2 = \{[(t_{3,2} + t_{1,2}) + idle_1] - [(t_{3,1} + t_{3,2}) + t_{3,3}]\} + (t_{3,1} + t_{3,2}) = [(4 + 8) + 31] - [(30 + 4) + 5] + 30 + 4 = 38$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(3,1)	35	43	63	31	38
(3,2)	36	66	**72** 66 + 6	32	**61**
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** operação	**2**	**3**
tarefa 3	30	4	2,2 > 3,3 5
2	6	30	6

3,1	3,2	3,3
2,1	2,2	2,3

$$\textit{idle}_2 = \{[(t_{3,2} + t_{2,2}) + \textit{idle}_1] - [(t_{3,1} + t_{3,2}) + t_{3,3}]\} + (t_{3,1} + t_{3,2}) = [(4 + 30) + 32] - [(30 + 4) + 5] + 30 + 4 = 61$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	**42** 39 + 3	30	**34**
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
	operação		
3	30	4	4,2 = 3,3 5
4	2	5	3

tarefa

3,1	3,2	3,3
4,1	4,2	4,3

Se: $[(t_{3,2} + t_{4,2}) + idle_1] \leq [(t_{3,1} + t_{3,2}) + t_{3,3}] \Rightarrow [(4 + 5) + 30] = [(30 + 4) + 5] \Rightarrow 39 = 39$

$idle_2 = (t_{3,1} + t_{3,2}) = (30 + 4) = 34$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	44	**48** 44 + 4	30	**39**
(3,6)	34	35	43	30	34

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** operação	**2**	**3**
tarefa 3	30	4	5,2 > 3,3 5
5	3	10	4

3,1	3,2	3,3
5,1	5,2	5,3

$$idle_2 = \{[(t_{3,2} + t_{5,2}) + idle_1] - [(t_{3,1} + t_{3,2}) + t_{3,3}]\} + (t_{3,1} + t_{3,2}) = [(4 + 10) + 30] - [(30 + 4) + 5] + 30 + 4 = 39$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	**43** (30 + 4 + 5) + 4	30	**34**

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
	operação →		
tarefa 3	30	4	5
tarefa 6	4	1	4

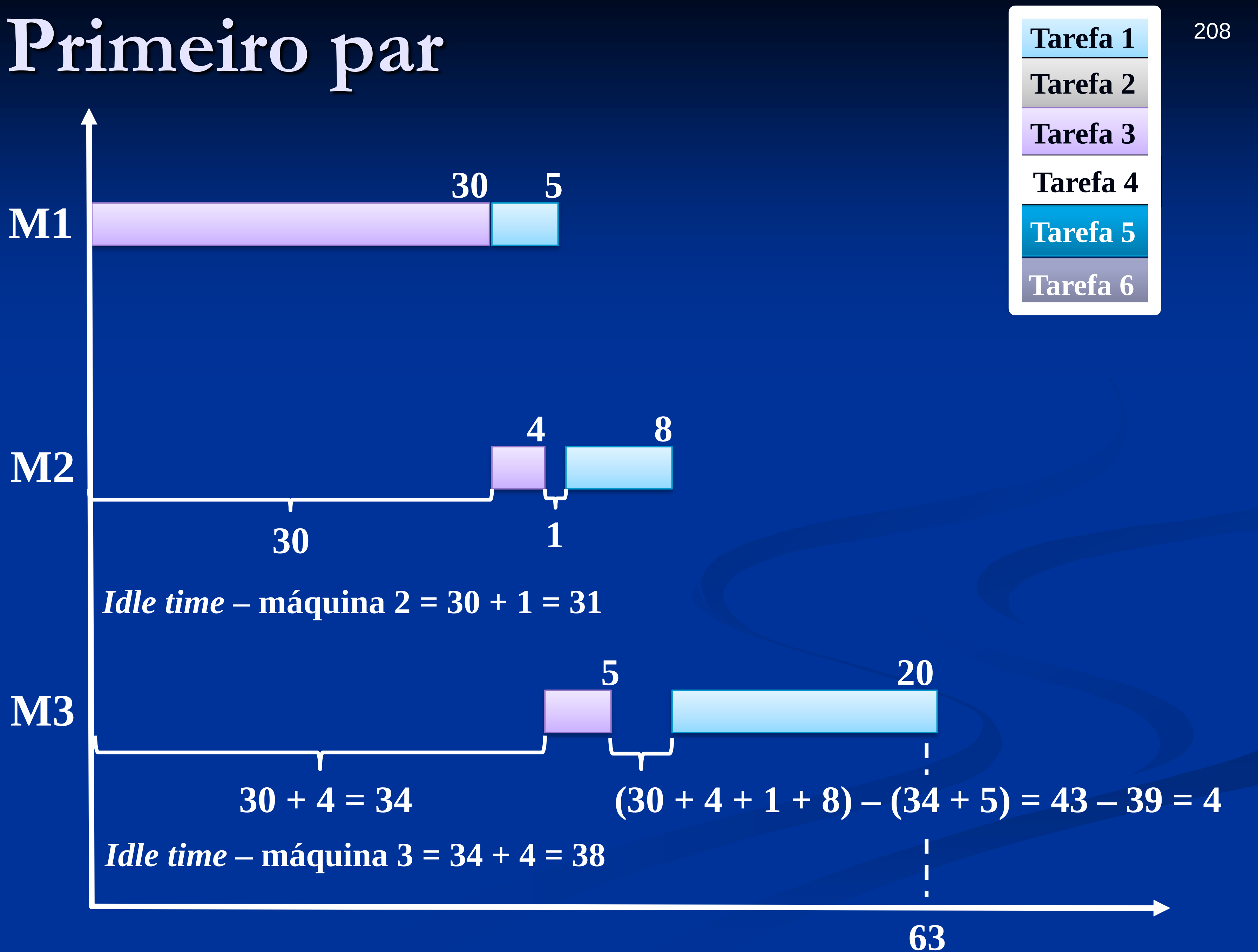
6,2 < 3,3

3,1	3,2	3,3
6,1	6,2	6,3

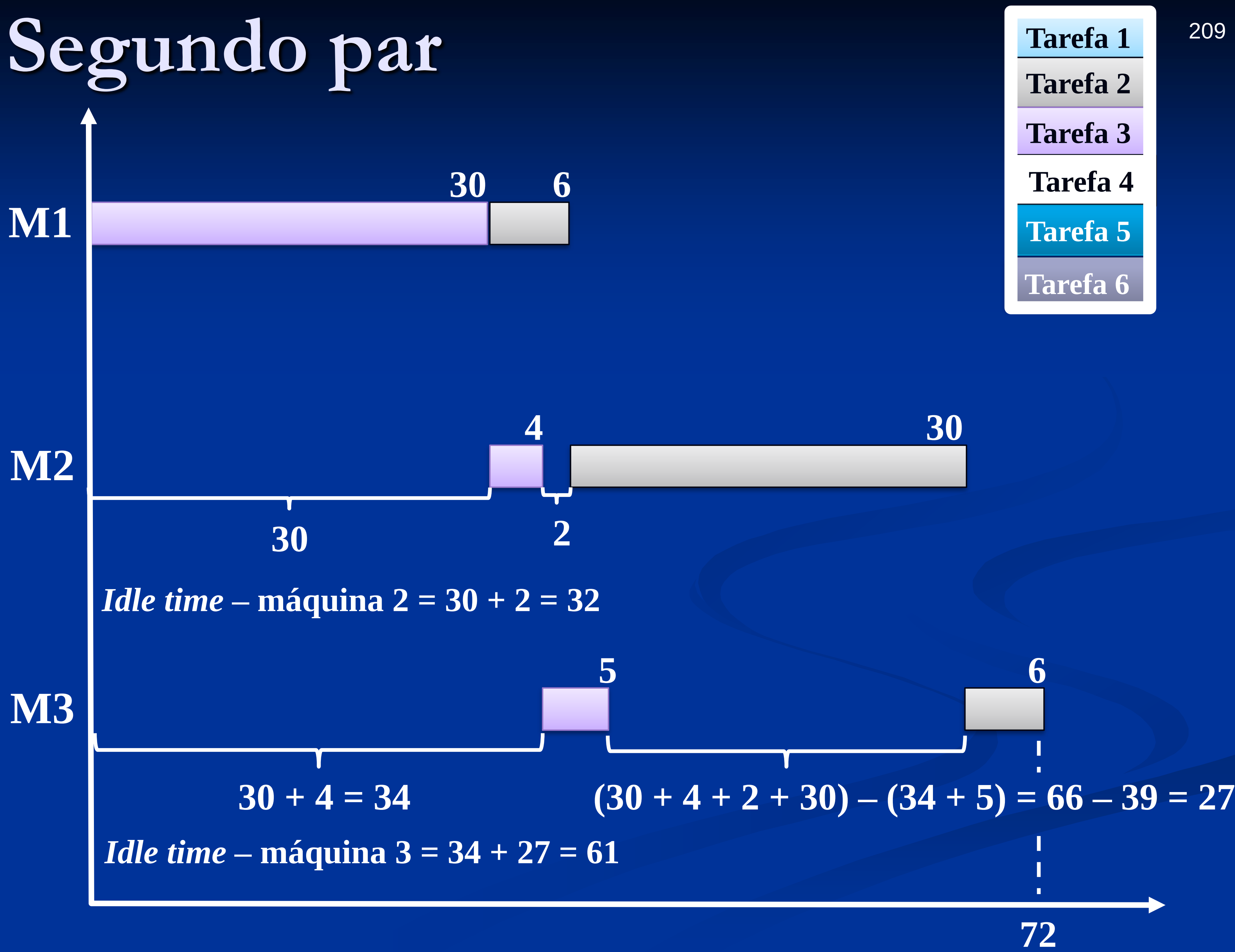
Se: $[(t_{3,2} + t_{6,2}) + idle_1] \leq [(t_{3,1} + t_{3,2}) + t_{3,3}] \Rightarrow [(4 + 1) + 30] < [(30 + 4) + 5] \Rightarrow 35 < 39$

$idle_2 = (t_{3,1} + t_{3,2}) = (30 + 4) = 34$

Primeiro par

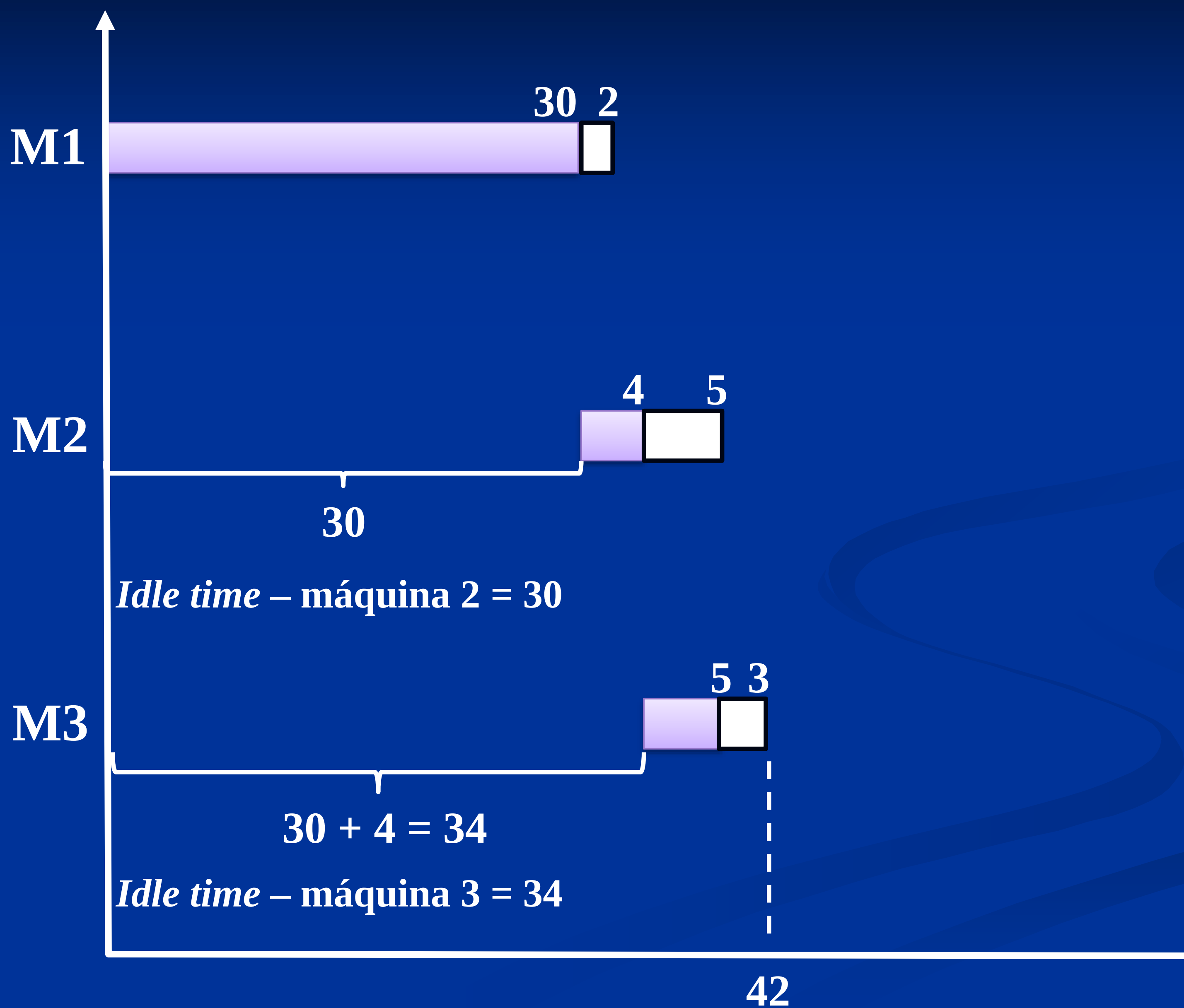


Segundo par

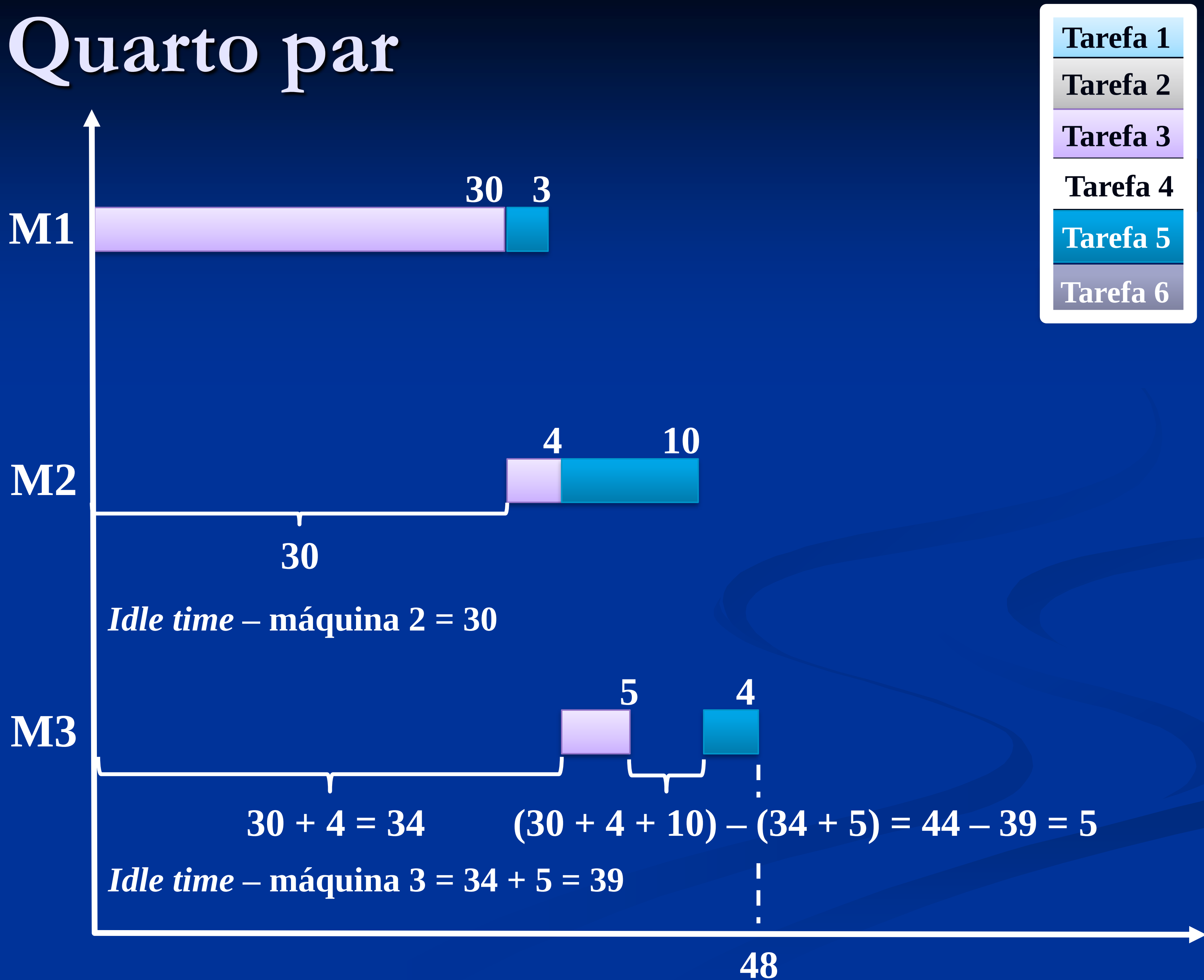


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
30
6
M2
4
30
30
2
Idle time – máquina 2 = 30 + 2 = 32
M3
5
6
30 + 4 = 34
(30 + 4 + 2 + 30) – (34 + 5) = 66 – 39 = 27
Idle time – máquina 3 = 34 + 27 = 61
72

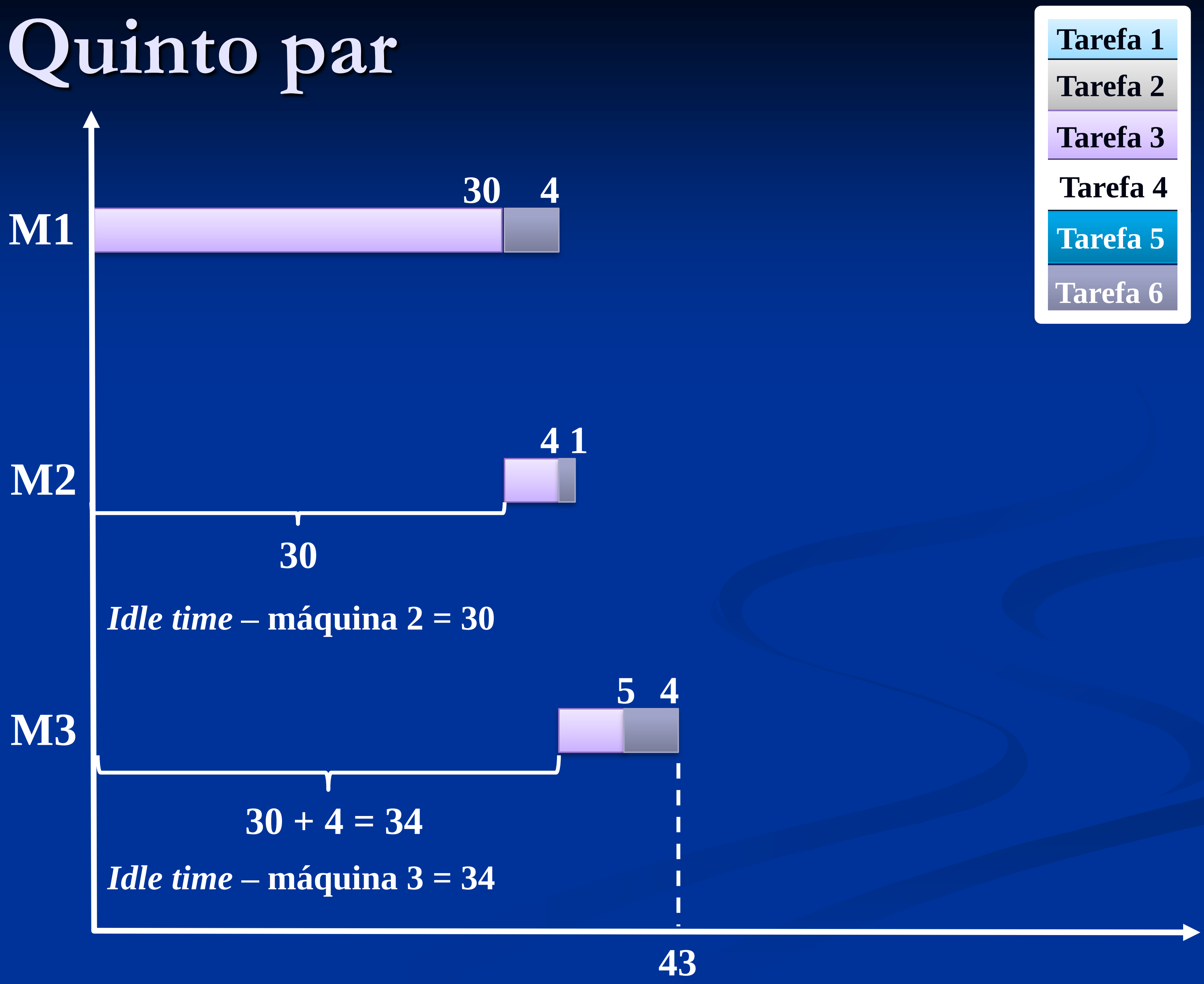
Terceiro par



Quarto par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
30
3
M2
4
10
30
Idle time – máquina 2 = 30
M3
5
4
30 + 4 = 34
(30 + 4 + 10) – (34 + 5) = 44 – 39 = 5
Idle time – máquina 3 = 34 + 5 = 39
48

Quinto par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
30
4
M2
4
1
30
Idle time – máquina 2 = 30
M3
5
4
30 + 4 = 34
Idle time – máquina 3 = 34
43

4ª série

Tarefa 4

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(**4**,1)	7	15	35	2	12
(**4**,2)	8	38	44	3	35
(**4**,3)	32	36	41	27	33
(**4**,5)	5	17	21	2	14
(**4**,6)	6	8	14	2	7

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(**4**,1)	2 + 5 = 7	7 + **8** = **15** (2 + 5) + **8** = **15**	15 + **20** = **35** (2 + 5 + 3) + **20** = 30		
(**4**,2)	2 + 6 = 8	8 + **30** = **38** (2 + 5) + **30** = 37	38 + **6** = **44** (2 + 5 + 3) + **6** = 16		
(**4**,3)	2 + 30 = 32	32 + **4** = **36** (2 + 5) + **4** = 11	36 + **5** = **41** (2 + 5 + 3) + **5** = 15		
(**4**,5)	2 + 3 = 5	5 + **10** = 15 (2 + 5) + **10** = **17**	17 + **4** = **21** (2 + 5 + 3) + **4** = 14		
(**4**,6)	2 + 4 = 6	6 + **1** = **7** (2 + 5) + **1** = **8**	8 + **4** = 12 (2 + 5 + 3) + **4** = **14**		

2ª operação da 2ª tarefa

Tempo da 3ª operação da 2ª tarefa

ou

Soma dos tempos de operação da 1ª tarefa

Repete a 2ª opeação da 2ª tarefa

maior valor

1ª operação das tarefas

Soma da 1ª e da 2ª operação da 1ª tarefa

Máquina 2

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,1)	7	**15** **2 + 5 + 8 (4,1 ⇒ 4,2 ⇒ 1,2)**	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	**1**	**2**	**3**
	operação →		
tarefa 4	2	5	3
1	5	8	20

	4,1	4,2	4,3
Se $t_{1,1} \leq t_{4,2} \Rightarrow$ *idle* $= t_{4,1}$	1,1	1,2	1,3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,1)	7	15	35	2	12
(4,2)	8	**38** 8 + 30	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** operação	**2**	**3**
tarefa 4	2	5	3
2	6	30	6

4,1	4,2	4,3
2,1	2,2	2,3

Se $t_{2,1} > t_{4,2} \Rightarrow$

***idle* = $(t_{2,1} - t_{4,2}) + t_{4,1}$**

(6 – 5) + 2 = 3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	**36** 32 + 4	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1**	**2**	**3**
tarefa	**operação**		
4	2	5	3
3	30	4	5

Se $t_{3,1} > t_{4,2} \Rightarrow$
***idle* = $(t_{3,1} - t_{4,2}) + t_{4,1}$**
(30 – 5) + 2 = 27

4,1	4,2	4,3
3,1	3,2	3,3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17 2 + 5 + 10 (4,1 ⇒ 4,2 ⇒ 5,2)	21	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1	2	3
operação			
tarefa 4	2	5	3
5	3	10	4

1,1	1,2	1,3
6,1	6,2	6,3

Se $t_{5,1} \leq t_{4,2} \Rightarrow$ *idle* $= t_{4,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	**8** 2 + 5 + 1 (4,1 ⇒ 4,2 ⇒ 6,2)	14	2	7

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação →			
4	2	5	3
6	4	1	4

tarefa ↓

4,1	4,2	4,3
6,1	6,2	6,3

Se $t_{6,1} \leq t_{4,2} \Rightarrow$ *idle* $= t_{4,1}$

Máquina 3

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	**35** 15 + 20	2	**12**
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1	2	3
	operação		
tarefa 4	2	5	1,2 > 4,3 3
1	5	8	20

4,1	4,2	4,3
1,1	1,2	1,3

$$idle_2 = \{[(t_{4,2} + t_{1,2}) + idle_1] - [(t_{4,1} + t_{4,2}) + t_{4,3}]\} + (t_{4,1} + t_{4,2}) = [(5 + 8) + 2] - [(2 + 5) + 3] + 2 + 5 = 12$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,1)	7	15	35	2	12
(4,2)	8	38	**44** 38 + 6	3	**35**
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
4	2	5	2,2 > 4,3 3
2	6	30	6

tarefa

4,1	4,2	4,3
2,1	2,2	2,3

$$\boldsymbol{idle_2 = \{[(t_{4,2} + t_{2,2}) + idle_1] - [(t_{4,1} + t_{4,2}) + t_{4,3}]\} + (t_{4,1} + t_{4,2}) = [(5 + 30) + 3] - [(2 + 5) + 3] + 2 + 5 = 35}$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41 36 + 5	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1	2	3
	operação		
4	2	5	3,2 > 4,3 3
3	30	4	5

tarefa

4,1	4,2	4,3
3,1	3,2	3,3

$idle_2 = \{[(t_{4,2} + t_{3,2}) + idle_1] - [(t_{4,1} + t_{4,2}) + t_{4,3}]\} + (t_{4,1} + t_{4,2}) = [(5 + 4) + 27] - [(2 + 5) + 3] + 2 + 5 = 33$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21 17 + 4	2	14
(4,6)	6	8	14	2	7

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1	2	3
	operação →		
tarefa 4	2	5	3
tarefa 5	3	10	4

5,2 > 4,3

4,1	4,2	4,3
5,1	5,2	5,3

$$idle_2 = \{[(t_{4,2} + t_{5,2}) + idle_1] - [(t_{4,1} + t_{4,2}) + t_{4,3}]\} + (t_{4,1} + t_{4,2}) = [(5 + 10) + 2] - [(2 + 5) + 3] + 2 + 5 = 14$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14 (2 + 5 + 3) + 4	2	7

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
tarefa / operação			
4	2	5	3
6	4	1	4

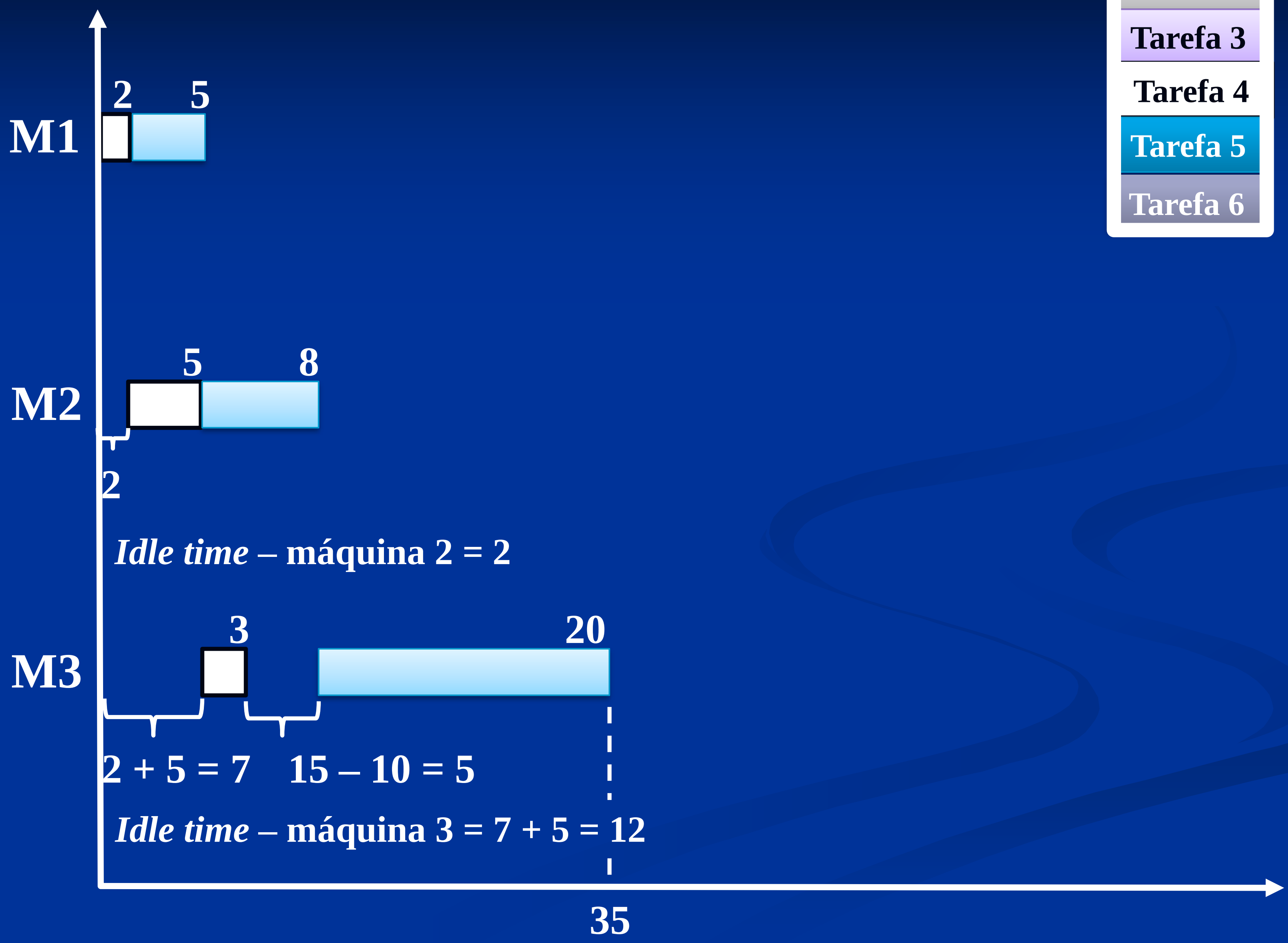
6,2 < 4,3

4,1	4,2	4,3
6,1	6,2	6,3

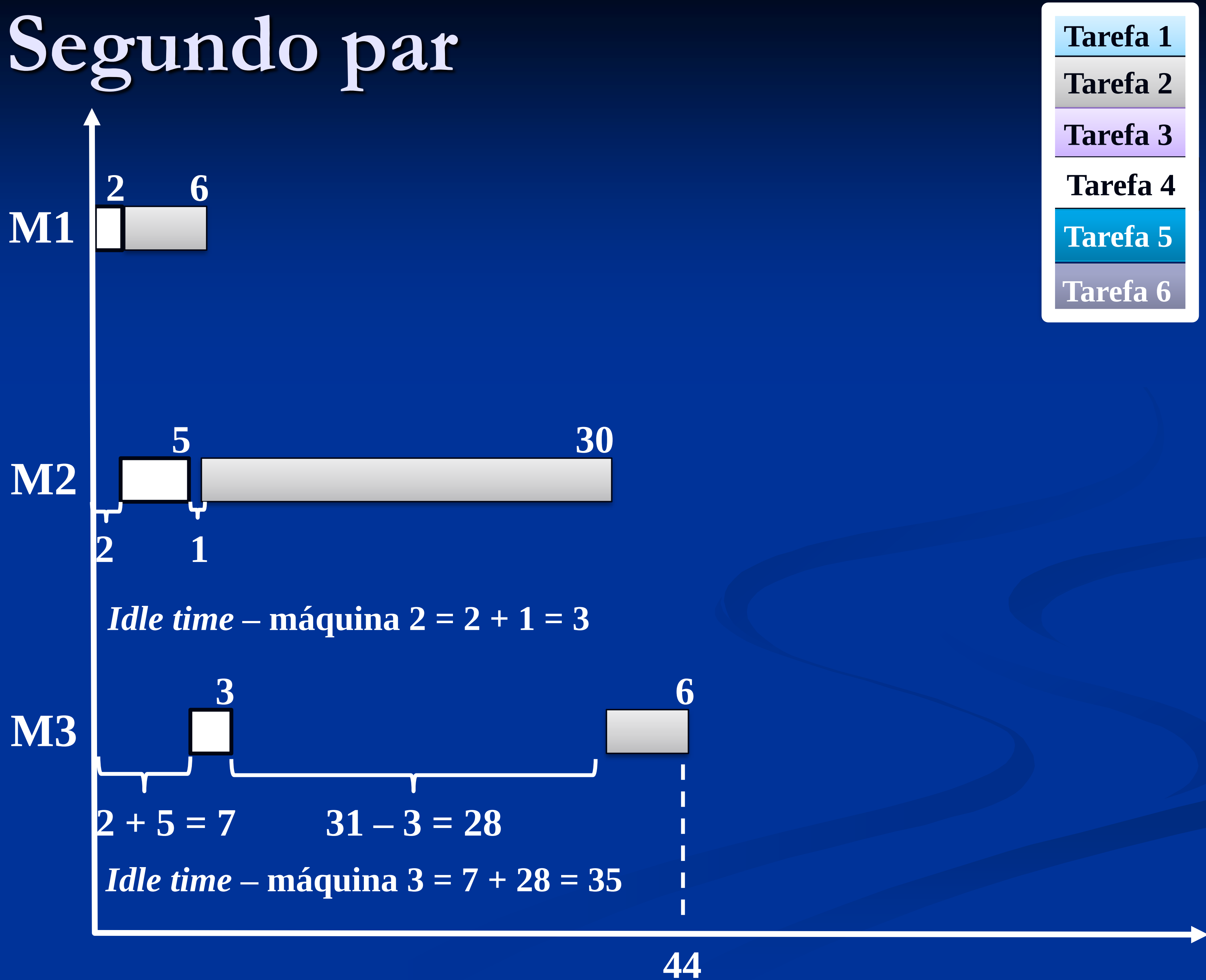
Se: $[(t_{4,2} + t_{6,2}) + idle_1] \leq [(t_{4,1} + t_{4,2}) + t_{4,3}] \Rightarrow [(5 + 1) + 2] < \underbrace{[(2 + 5) + 3]} \Rightarrow 8 < 10$

$idle_2 = (t_{4,1} + t_{4,2}) = (2 + 5) = 7$

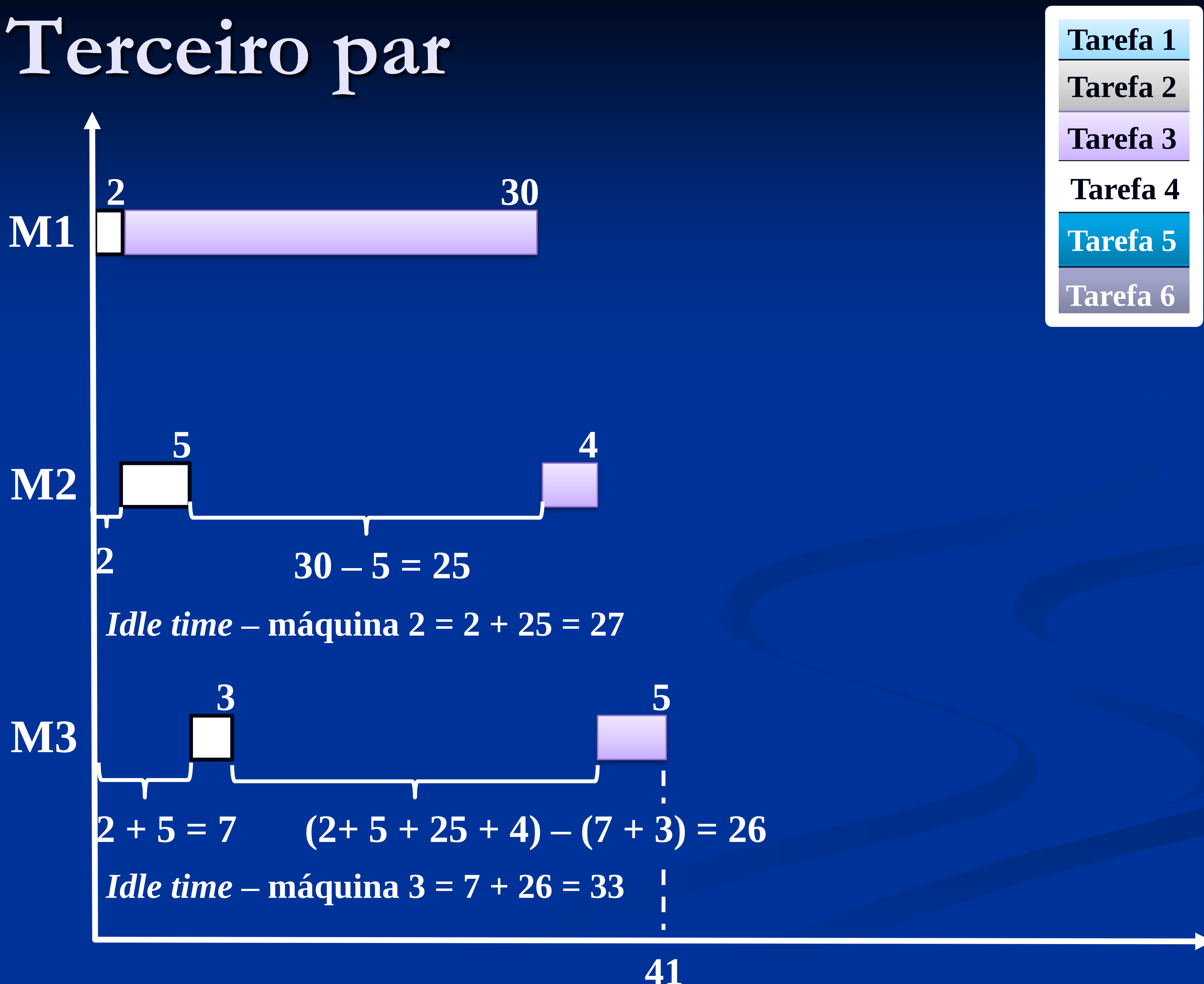
Primeiro par



Segundo par

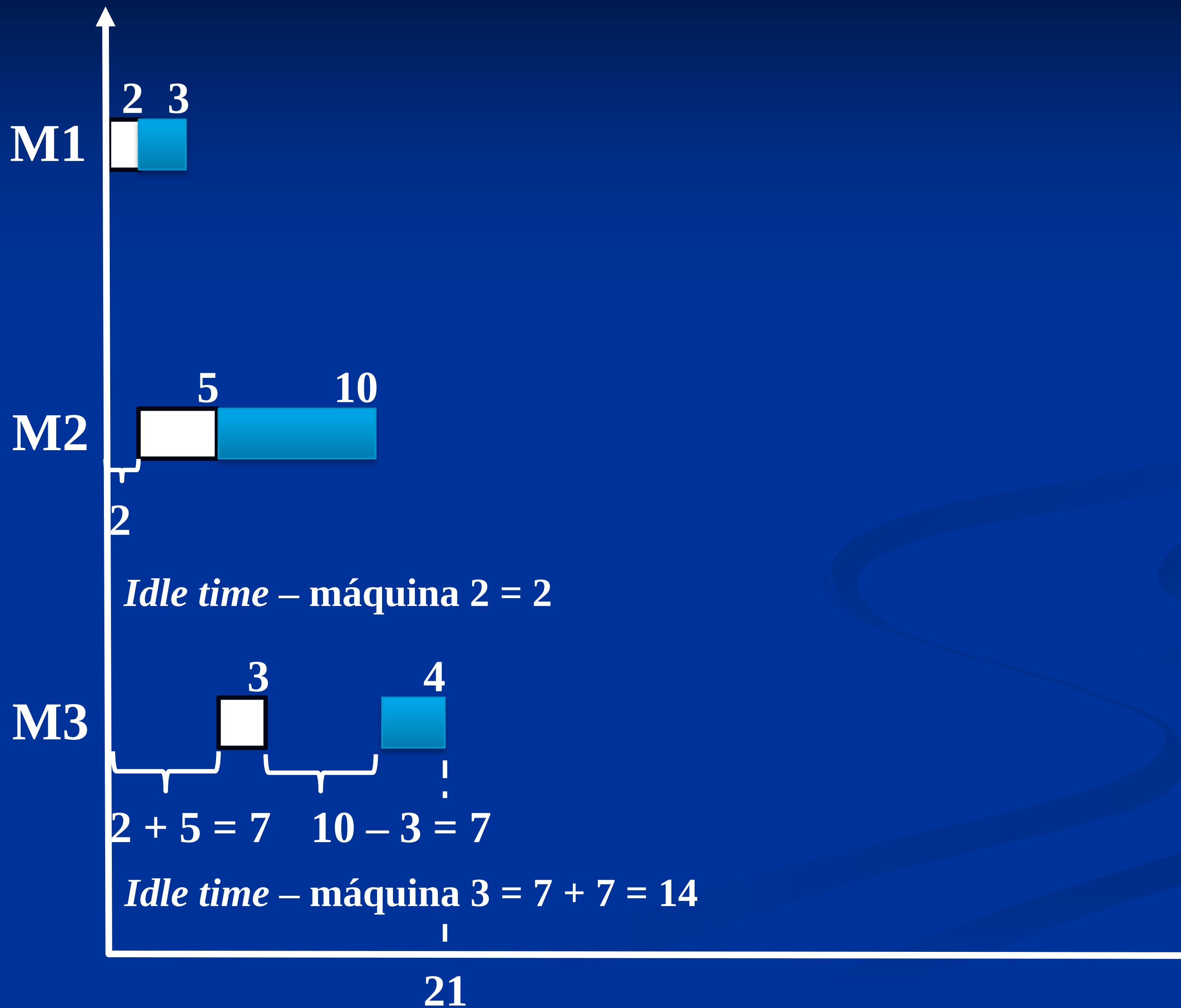

Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
2
6
M2
5
30
2
1
Idle time – máquina 2 = 2 + 1 = 3
M3
3
6
2 + 5 = 7
31 – 3 = 28
Idle time – máquina 3 = 7 + 28 = 35
44

Terceiro par

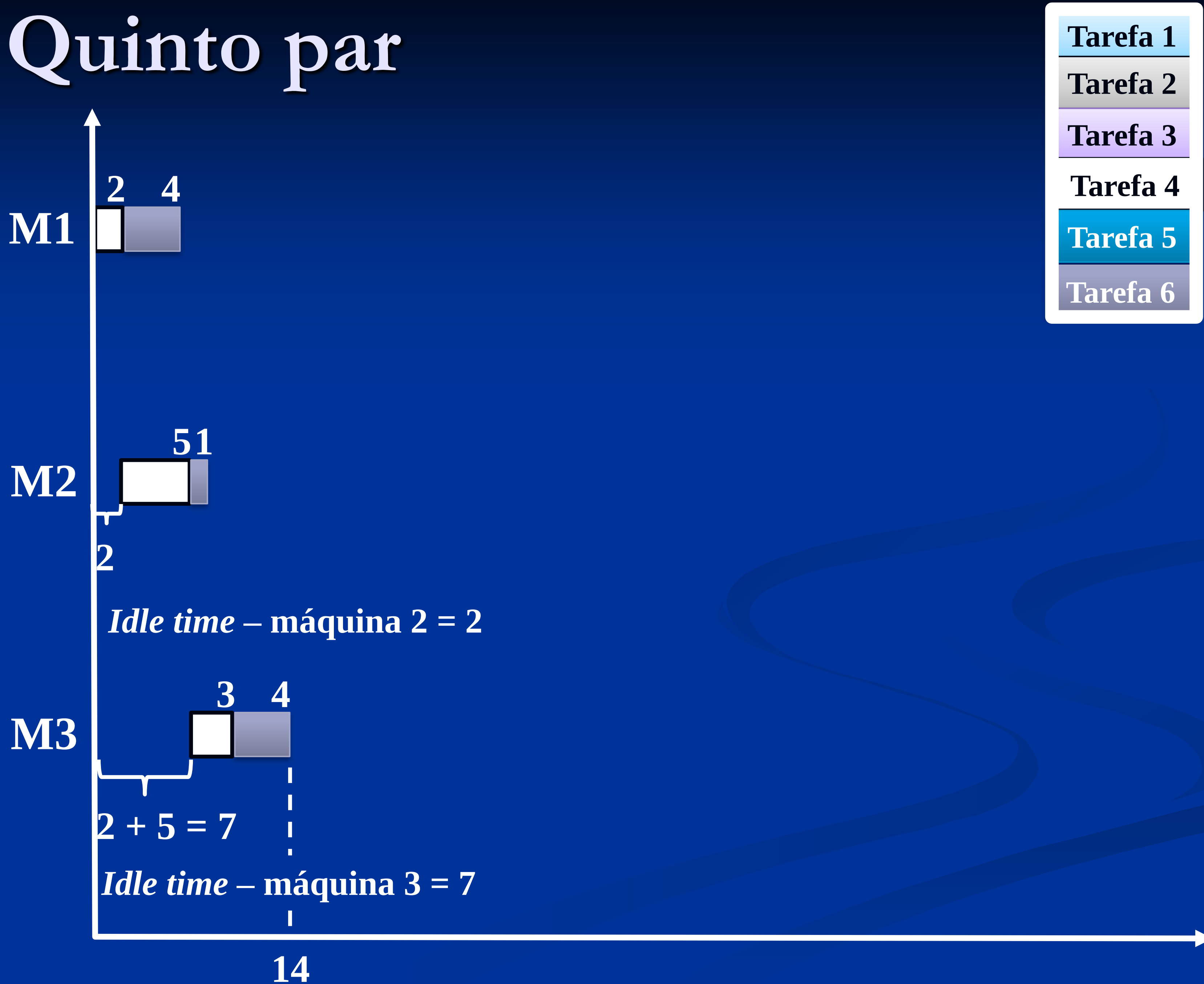


Quarto par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6


M1
2
3
M2
5
10
2
Idle time – máquina 2 = 2
M3
3
4
2 + 5 = 7
10 – 3 = 7
Idle time – máquina 3 = 7 + 7 = 14
21

Quinto par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
2
4
M2
51
2
Idle time – máquina 2 = 2
M3
3
4
2 + 5 = 7
Idle time – máquina 3 = 7
14

5ª série

Tarefa 5

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(**5**,1)	8	21	41	3	17
(**5**,2)	9	43	49	3	39
(**5**,3)	33	37	42	23	33
(**5**,4)	5	18	21	3	14
(**5,**6)	7	14	21	3	13

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(5,1)	3 + 5 = 8	8 + **8** = 16 (3 + 10) + **8** = **21**	21 + **20** = **41** (3 + 10 + 4) + **20** = 37		
(5,2)	3 + 6 = 9	9 + **30** = 39 (3 + 10) + **30** = **43**	43 + **6** = **49** (3 + 10 + 4) + **6** = 23		
(5,3)	3 + 30 = 33	33 + **4** = **37** (3 + 10) + **4** = 17	37 + **5** = **42** (3 + 10 + 4) + **5** = 22		
(5,4)	3 + 2 = 5	5 + **5** = 10 (3 + 10) + **5** = **18**	18 + **3** = **21** (3 + 10 + 4) + **3** = 20		
(5,6)	3 + 4 = 7	7 + **1** = **8** (3 + 10) + **1** = **14**	14 + **4** = 18 (3 + 10 + 4) + **4** = **21**		

2ª operação da 2ª tarefa

Tempo da 3ª operação da 2ª tarefa

Soma dos tempos de operação da 1ª tarefa

Repete a 2ª opeação da 2ª tarefa

maior valor

1ª operação das tarefas

Soma da 1ª e da 2ª operação da 1ª tarefa

Máquina 2

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(5,1)	8	**21** 3 + 10 + 8 (5,1 ⇒ 5,2 ⇒ 1,2)	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) / i (tarefa)	**1** operação	**2**	**3**
tarefa 5	3	10	4
1	5	8	20

5,1	5,2	5,3
1,1	1,2	1,3

Se $t_{1,1} \leq t_{5,2} \Rightarrow$ *idle* $= t_{5,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(5,1)	8	21	41	3	17
(5,2)	9	43 3 + 10 + 30 (5,1 ⇒ 5,2 ⇒2,2)	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1	2	3
operação			
tarefa 5	3	10	4
2	6	30	6

Se $t_{2,1} \leq t_{5,2} \Rightarrow$ *idle* $= t_{5,1}$	5,1	5,2	5,3
	2,1	2,2	2,3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(5,1)	8	21	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	**37** 33 + 4	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
tarefa 5	3	10	4
3	30	4	5

Se $t_{3,1} > t_{5,2} \Rightarrow$	5,1	5,2	5,3
***idle* = $(t_{3,1} - t_{5,2}) + t_{5,1}$** **(30 – 10) + 3 = 23**	3,1	3,2	3,3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(5,1)	8	21	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	**18** **3 + 10 + 5 (5,1 ⇒ 5,2 ⇒ 4,2)**	21	3	14
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
5	3	10	4
4	2	5	3

tarefa

5,1	5,2	5,3
4,1	4,2	4,3

Se $t_{4,1} \leq t_{5,2} \Rightarrow$ *idle* $= t_{5,1}$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(5,1)	8	21	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	**14** 3 + 10 + 1 (5,1 ⇒ 5,2 ⇒ 6,2)	21	3	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
	operação		
5	3	10	4
6	4	1	4

tarefa

Se $t_{6,1} \leq t_{5,2} \Rightarrow$ *idle* $= t_{5,1}$

5,1	5,2	5,3
6,1	6,2	6,3

Máquina 3

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(5,1)	8	21	41 21 + 20	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1 **operação**	2	3
tarefa 5	3	10	4
1	5	8	20

1,2 > 5,3

5,1	5,2	5,3
1,1	1,2	1,3

$$\textit{idle}_2 = \{[(t_{5,2} + t_{1,2}) + \textit{idle}_1] - [(t_{5,1} + t_{5,2}) + t_{5,3}]\} + (t_{5,1} + t_{5,2}) = [(10 + 8) + 3] - [(3 + 10) + 4] + 3 + 10 = 17$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(5,1)	8	21	41	3	17
(5,2)	9	43	**49** 43 + 6	3	**39**
(5,3)	33	37	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** operação	**2**	**3**
tarefa 5	3	10	2,2 > 5,3 4
2	6	30	6

5,1	5,2	5,3
2,1	2,2	2,3

$$\textit{idle}_2 = \{[(t_{5,2} + t_{2,2}) + \textit{idle}_1] - [(t_{5,1} + t_{5,2}) + t_{5,3}]\} + (t_{5,1} + t_{5,2}) = [(10 + 30) + 3] - [(3 + 10) + 4] + 3 + 10 = 39$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(5,1)	8	21	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	**42** 37 + 5	23	**33**
(5,4)	5	18	21	3	14
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1**	**2**	**3**
	operação		
tarefa 5	3	10	4
3	30	4	5

3,2 = 5,3

5,1	5,2	5,3
3,1	3,2	3,3

$$idle_2 = \{[(t_{5,2} + t_{3,2}) + idle_1] - [(t_{5,1} + t_{5,2}) + t_{5,3}]\} + (t_{5,1} + t_{5,2}) = [(10 + 4) + 23] - [(3 + 10) + 4] + 3 + 10 = 33$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(5,1)	8	21	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	18	**21** 18 + 3	3	**14**
(5,6)	7	14	21	3	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) / i (tarefa)	**1**	**2**	**3**
	operação		
tarefa 5	3	10	4
tarefa 4	2	5	3

4,2 > 5,3

5,1	5,2	5,3
4,1	4,2	4,3

$$idle_2 = \{[(t_{5,2} + t_{4,2}) + idle_1] - [(t_{5,1} + t_{5,2}) + t_{5,3}]\} + (t_{5,1} + t_{5,2}) = [(10 + 5) + 3] - [(3 + 10) + 4] + 3 + 10 = 14$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(5,1)	8	21	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	14	**21** (3 + 10 + 4) + 4	3	**13**

Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1	2	3
	operação		
5	3	10	6,2 < 5,3 4
6	4	1	4

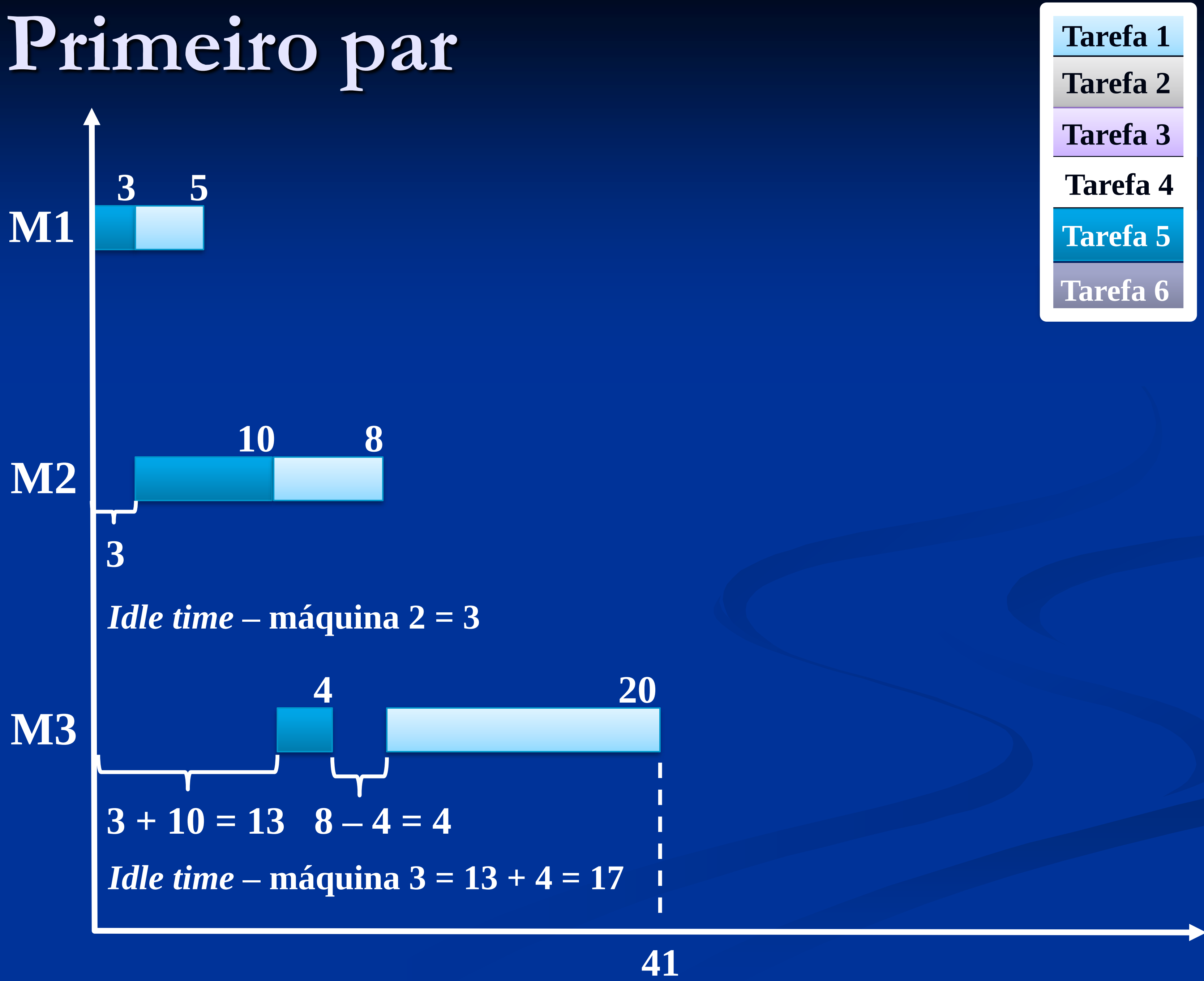
tarefa

5,1	5,2	5,3
6,1	6,2	6,3

Se: $[(t_{5,2} + t_{6,2}) + \textit{idle}_1] \leq [(t_{5,1} + t_{5,2}) + t_{5,3}] \Rightarrow [(10 + 1) + 3] < \underbrace{[(3 + 10) + 4]} \Rightarrow 14 < 17$

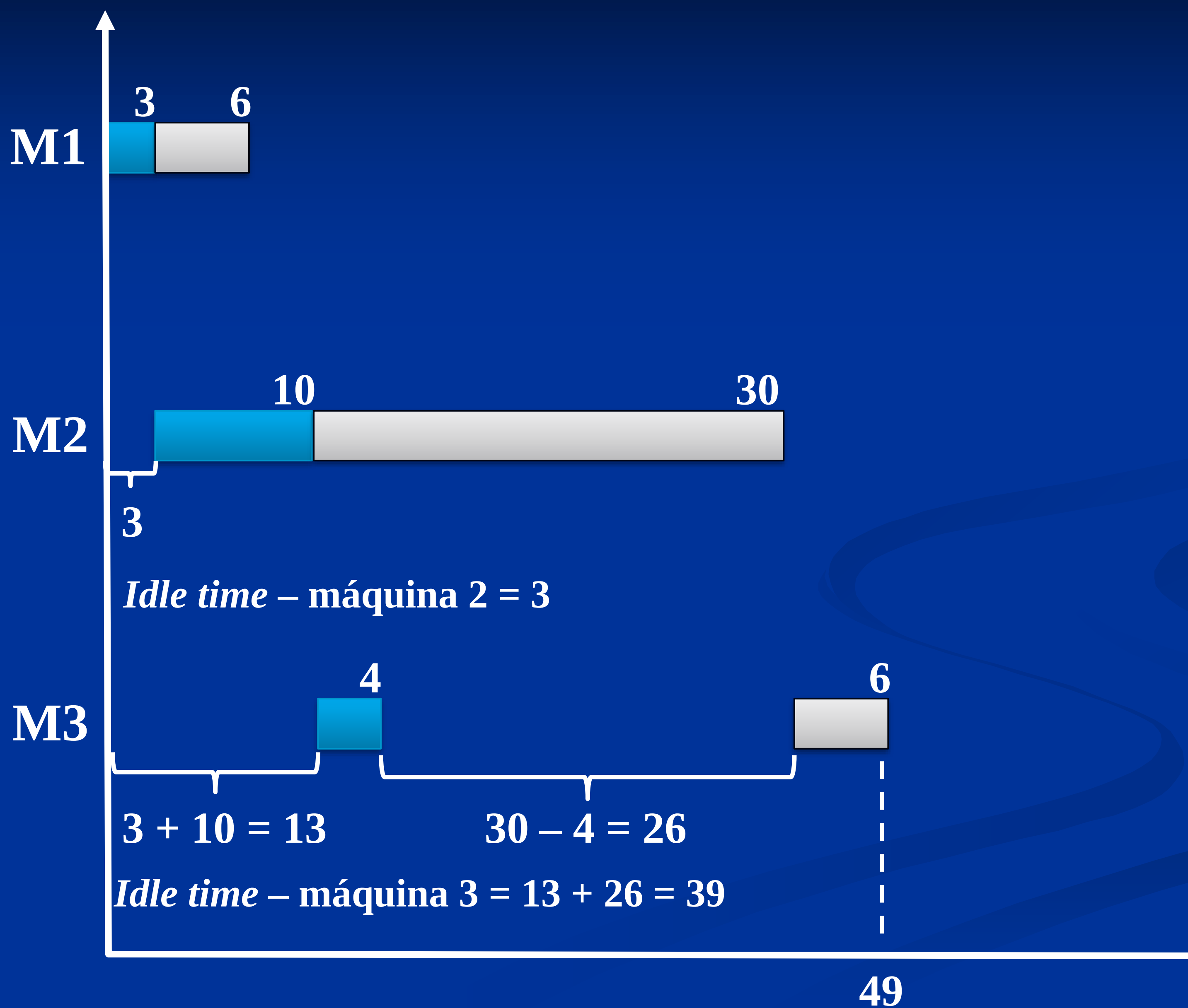
$\textit{idle}_2 = (t_{5,1} + t_{5,2}) = (3 + 10) = 13$

Primeiro par

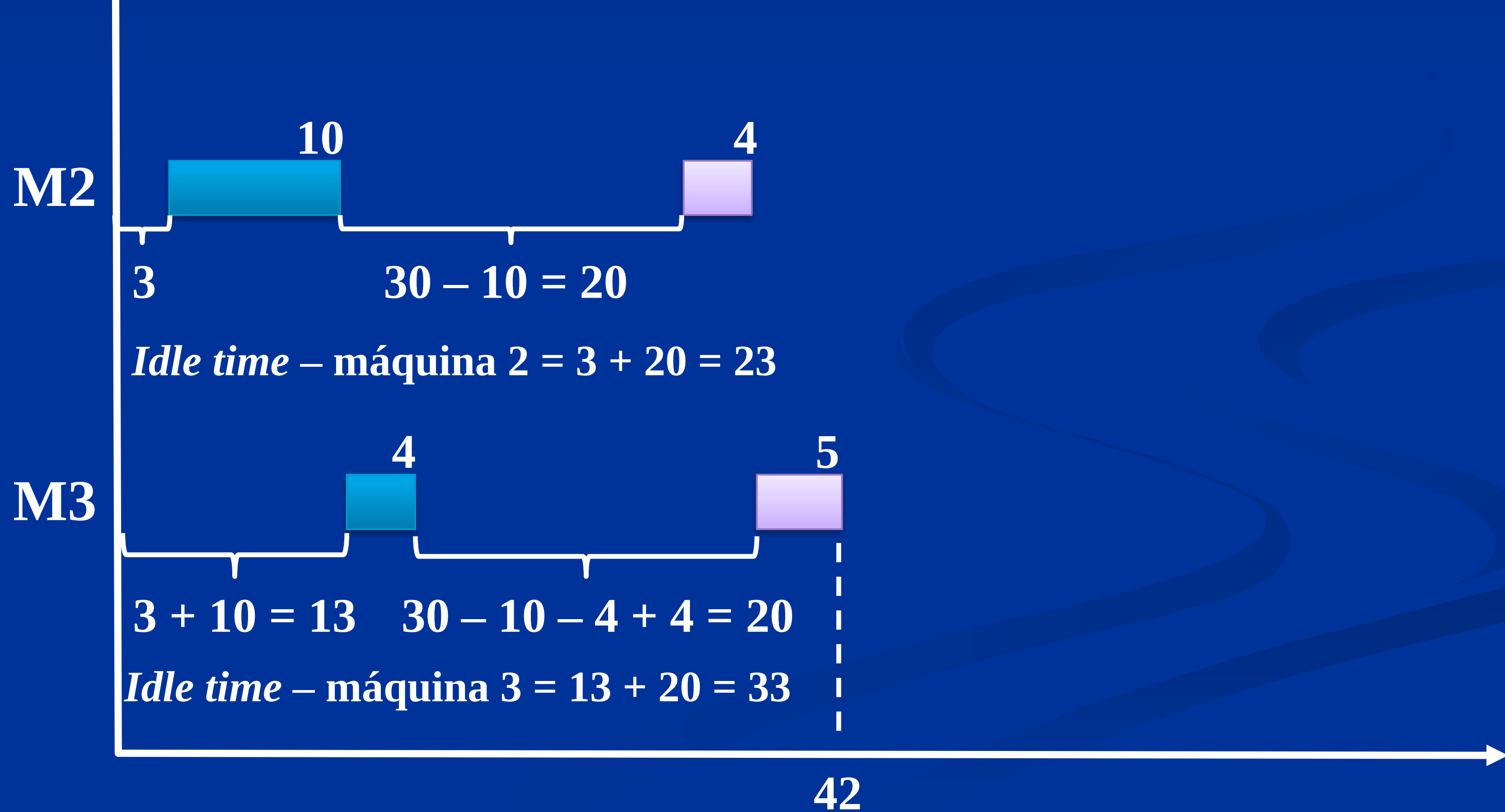


Segundo par

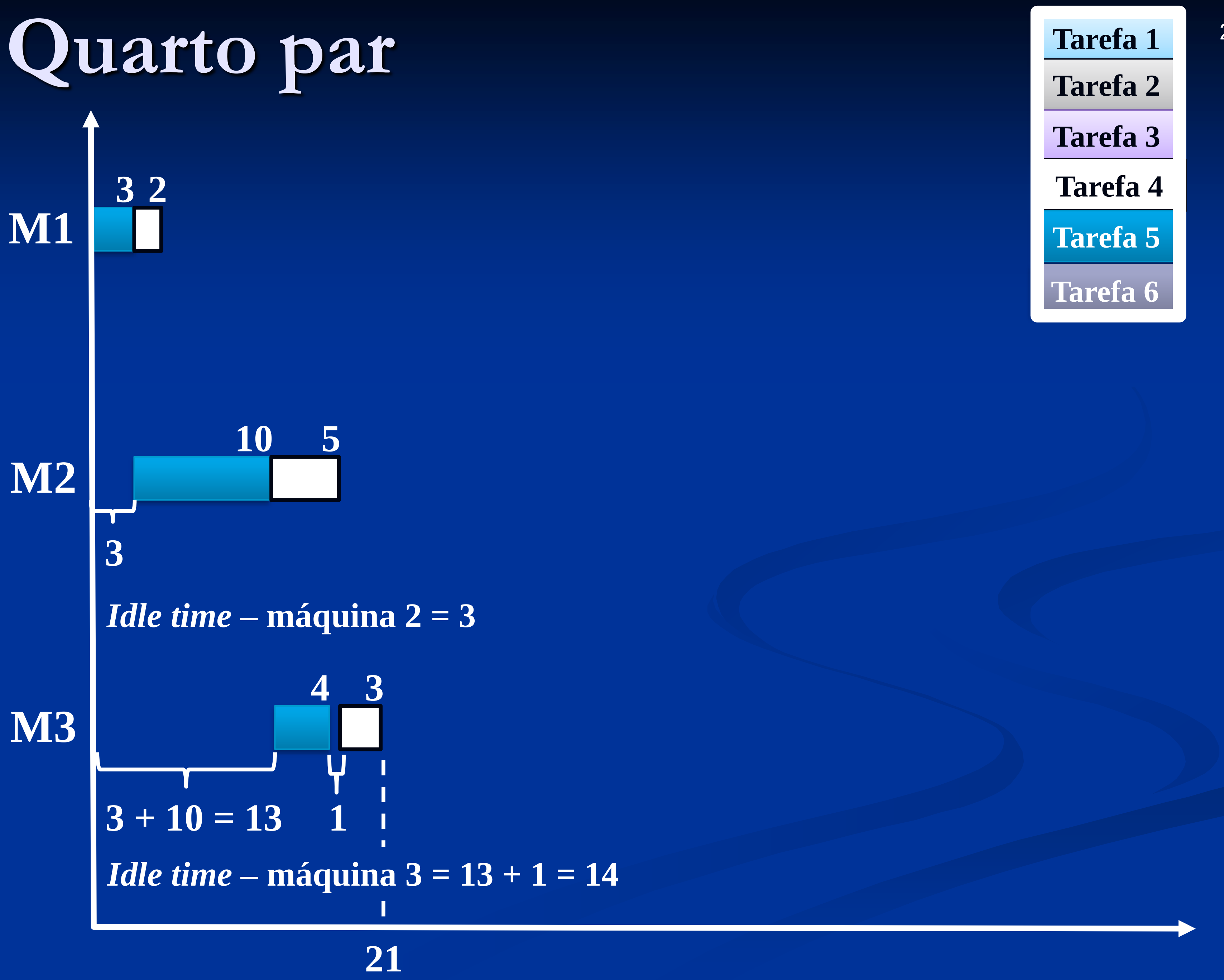

Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6


M1
3
6
M2
10
30
3
Idle time – máquina 2 = 3
M3
4
6
3 + 10 = 13
30 – 4 = 26
Idle time – máquina 3 = 13 + 26 = 39
49

Terceiro par

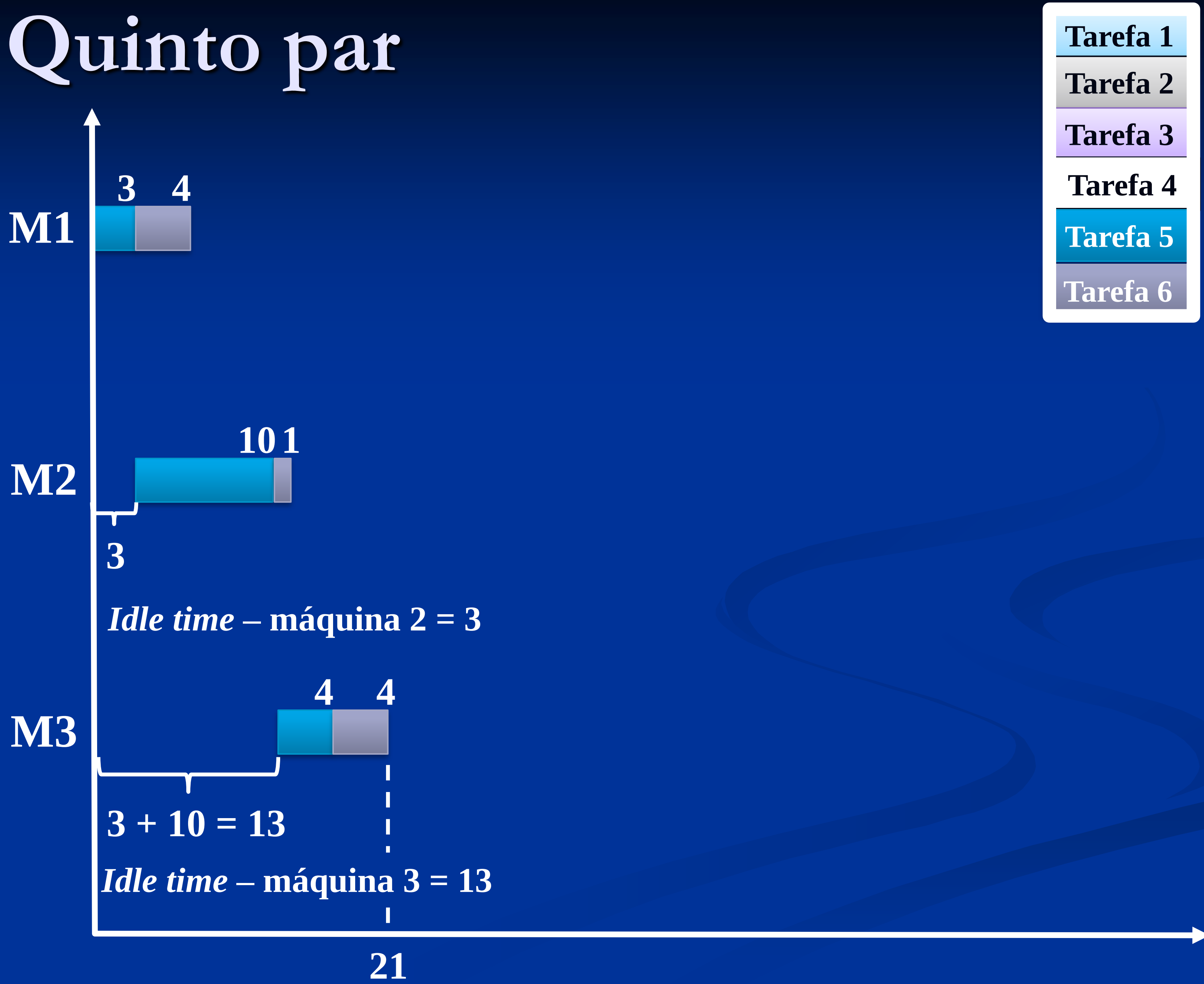


Quarto par



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
3
2
M2
10
5
3
Idle time – máquina 2 = 3
M3
4
3
3 + 10 = 13
1
Idle time – máquina 3 = 13 + 1 = 14
21

Quinto par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
3
4
M2
10
1
3
Idle time – máquina 2 = 3
M3
4
4
3 + 10 = 13
Idle time – máquina 3 = 13
21

6ª série

Tarefa 6

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	**1**	**2**	**3**	**2**	**3**
(**6**,1)	9	17	37	8	13
(**6**,2)	10	40	46	9	36
(**6**,3)	34	38	43	33	34
(**6**,4)	6	11	14	5	7
(**6**,5)	7	17	21	6	13

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(6,1)	4 + 5 = 9	9 + **8** = **17** (4 + 1) + **8** = 13	17 + **20** = **37** (4 + 1 + 4) + **20** = 29		
(6,2)	4 + 6 = 10	10 + **30** = **40** (4 + 1) + **30** = 34	40 + **6** = **46** (4 + 1 + 4) + **6** = 15		
(6,3)	4 + 30 = 34	34 + **4** = **38** (4 + 1) + **4** = 9	38 + **5** = **43** (4 + 1 + 4) + **5** = 14		
(6,4)	4 + 2 = 6	6 + **5** = **11** (4 + 1) + **5** = 10	11 + **3** = **14** (4 + 1 + 4) + **3** = 12		
(6,5)	4 + 3 = 7	7 + **10** = **17** (4 + 1) + **10** = 15	17 + **4** = **21** (4 + 1 + 4) + **4** = 13		

2ª operação da 2ª tarefa

Tempo da 3ª operação da 2ª tarefa

Soma dos tempos de operação da 1ª tarefa

Repete a 2ª opeação da 2ª tarefa

maior valor

1ª operação das tarefas

Soma da 1ª e da 2ª operação da 1ª tarefa

Máquina 2

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(6,1)	9	**17** 9 + 8	37	8	**13**
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	11	14	5	7
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1 operação	2	3
tarefa 6	4	1	4
tarefa 1	5	8	20

6,1	6,2	6,3
1,1	1,2	1,3

Se $t_{1,1} > t_{6,2} \Rightarrow$

idle **$= (t_{1,1} - t_{6,2}) + t_{6,1}$**

$(5 - 1) + 4 = 8$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(6,1)	9	17	37	8	13
(6,2)	10	**40** 10 + 30	46	9	**36**
(6,3)	34	38	43	33	34
(6,4)	6	11	14	5	7
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
tarefa 6	4	1	4
2	6	30	6

6,1	6,2	6,3
2,1	2,2	2,3

Se $t_{2,1} > t_{6,2} \Rightarrow$

***idle* = $(t_{2,1} - t_{6,2}) + t_{6,1}$**

(6 – 1) + 4 = 9

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	**38** 34 + 4	43	33	**34**
(6,4)	6	11	14	5	7
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação			
6	4	1	4
3	30	4	5

tarefa

Se $t_{3,1} > t_{6,2} \Rightarrow$
idle **$= (t_{3,1} - t_{6,2}) + t_{6,1}$**
$(30 - 1) + 4 = 33$

6,1	6,2	6,3
3,1	3,2	3,3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(**6**,1)	9	17	37	8	13
(**6**,2)	10	40	46	9	36
(**6**,3)	34	38	43	33	34
(**6**,4)	6	**11** 6 + 5	14	5	**7**
(**6**,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	1	2	3
operação			
6	4	1	4
4	2	5	3
	6,1	6,2	6,3
	4,1	4,2	4,3

tarefa

Se $t_{4,1} > t_{6,2} \Rightarrow$

idle **$= (t_{4,1} - t_{6,2}) + t_{6,1}$**

$(2 - 1) + 4 = 5$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	11	14	5	7
(6,5)	7	**17** 7 + 10	21	6	**13**

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1	2	3
	operação		
tarefa 6	4	1	4
5	3	10	4

	6,1	6,2	6,3
	5,1	5,2	5,3

Se $t_{5,1} > t_{6,2} \Rightarrow$

idle **$= (t_{5,1} - t_{6,2}) + t_{6,1}$**

$(3 - 1) + 4 = 6$

Máquina 3

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(6,1)	9	17	37 17 + 20	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	11	14	5	7
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	**1**	**2**	**3**
	operação		
tarefa 6	4	1	1,2 > 6,3 4
1	5	8	20

6,1	6,2	6,3
1,1	1,2	1,3

$$idle_2 = \{[(t_{6,2} + t_{1,2}) + idle_1] - [(t_{6,1} + t_{6,2}) + t_{6,3}]\} + (t_{6,1} + t_{6,2}) = [(1 + 8) + 8] - [(4 + 1) + 4] + 4 + 1 = 13$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(6,1)	9	17	37	8	13
(6,2)	10	40	46 40 + 6	9	36
(6,3)	34	38	43	33	34
(6,4)	6	11	14	5	7
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	1 operação	2	3
tarefa 6	4	1	2,2 > 6,3 4
2	6	30	6

6,1	6,2	6,3
2,1	2,2	2,3

$idle_2 = \{[(t_{6,2} + t_{2,2}) + idle_1] – [(t_{6,1} + t_{6,2}) + t_{6,3}]\} + (t_{6,1} + t_{6,2}) = [(1 + 30) + 9] – [(4 + 1) + 4] + 4 + 1 = 36$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	**43** 38 + 5	33	**34**
(6,4)	6	11	14	5	7
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) / i (tarefa)	**1** operação	**2**	**3**
tarefa 6	4	1	3,2 = 6,3 4
3	30	4	5

6,1	6,2	6,3
3,1	3,2	3,3

$$idle_2 = \{[(t_{6,2} + t_{3,2}) + idle_1] - [(t_{6,1} + t_{6,2}) + t_{6,3}]\} + (t_{6,1} + t_{6,2}) = [(1 + 4) + 33] - [(4 + 1) + 4] + 4 + 1 = 34$$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	11	14 11 + 3	5	7
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento

m (máquina) i (tarefa)	1	2	3
	operação		
tarefa 6	4	1	4,2 > 6,3 4
4	2	5	3

6,1	6,2	6,3
4,1	4,2	4,3

$idle_2 = \{[(t_{6,2} + t_{4,2}) + idle_1] - [(t_{6,1} + t_{6,2}) + t_{6,3}]\} + (t_{6,1} + t_{6,2}) = [(1 + 5) + 5] - [(4 + 1) + 4] + 4 + 1 = 7$

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	11	14	5	7
(6,5)	7	17	**21** 17 + 4	6	**13**

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)	**1**	**2**	**3**
	operação		
6	4	1	4
5	3	10	4

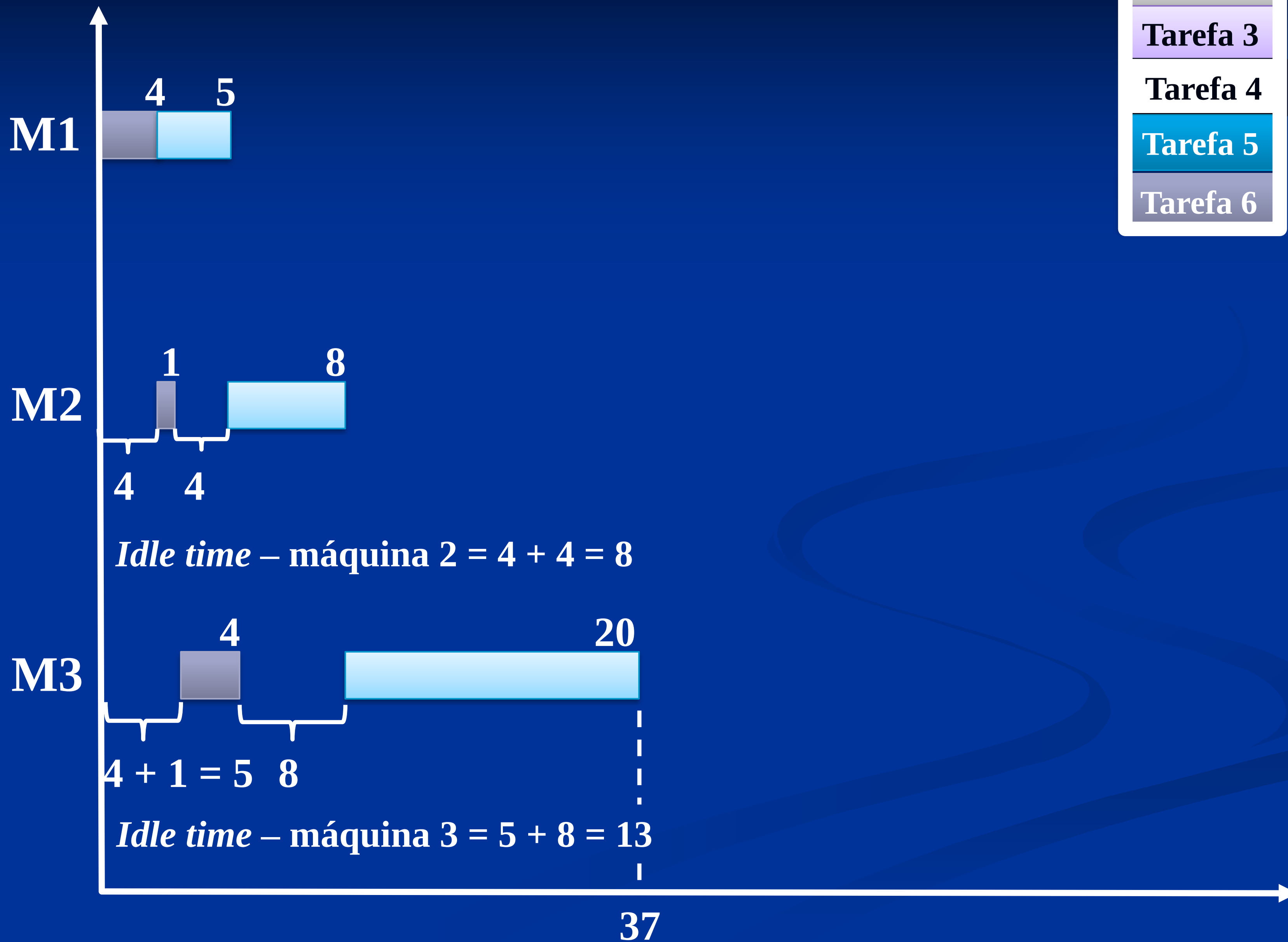
tarefa

5,2 > 6,3

6,1	6,2	6,3
5,1	5,2	5,3

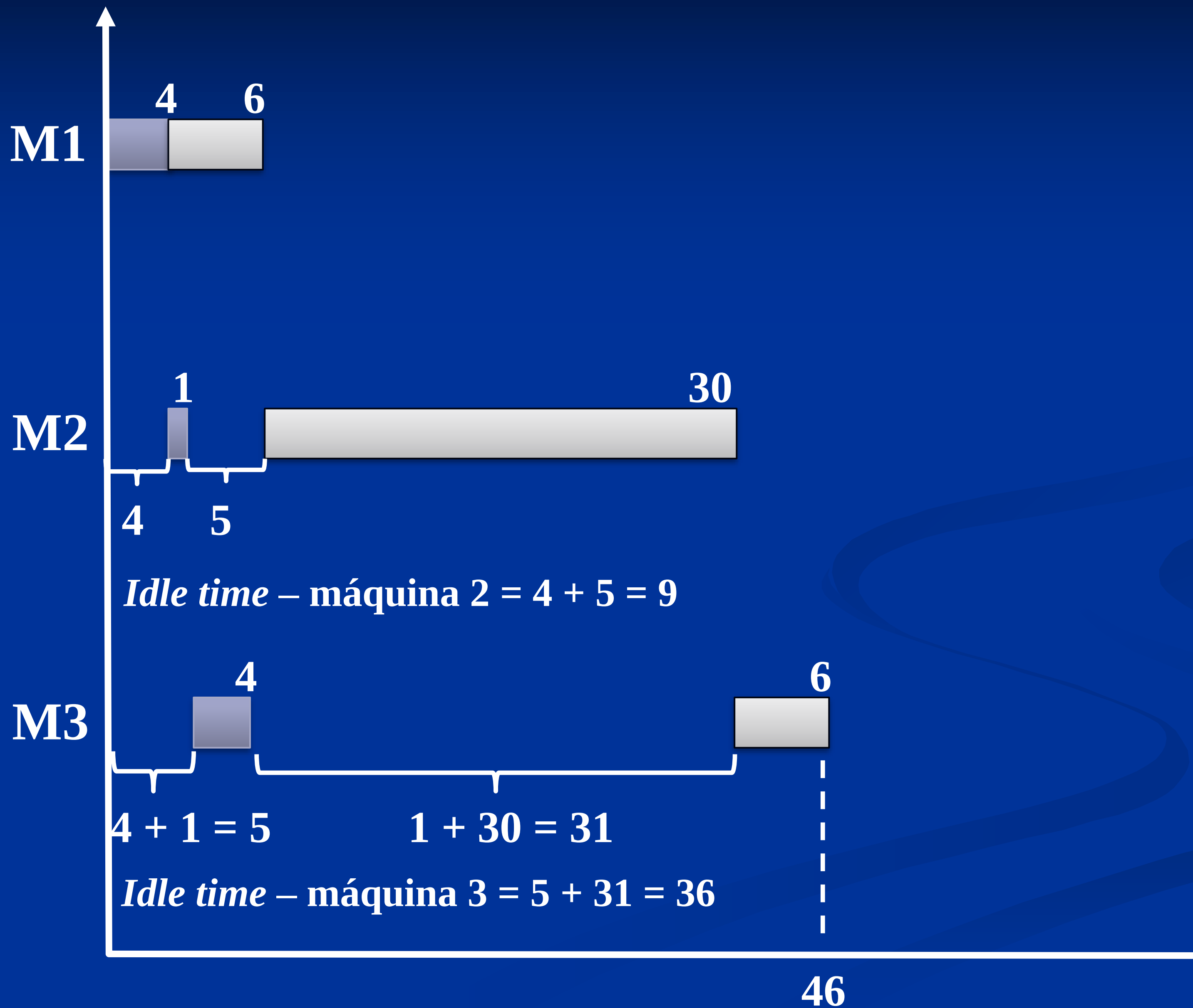
idle$_2$ **= {[(t$_{6,2}$ + t$_{5,2}$) +** ***idle***$_1$**] – [(t$_{6,1}$ + t$_{6,2}$) + t$_{6,3}$]} + (t$_{6,1}$ + t$_{6,2}$) = [(1 + 10) + 6] – [(4 + 1) + 4] + 4 + 1 = 13**

Primeiro par

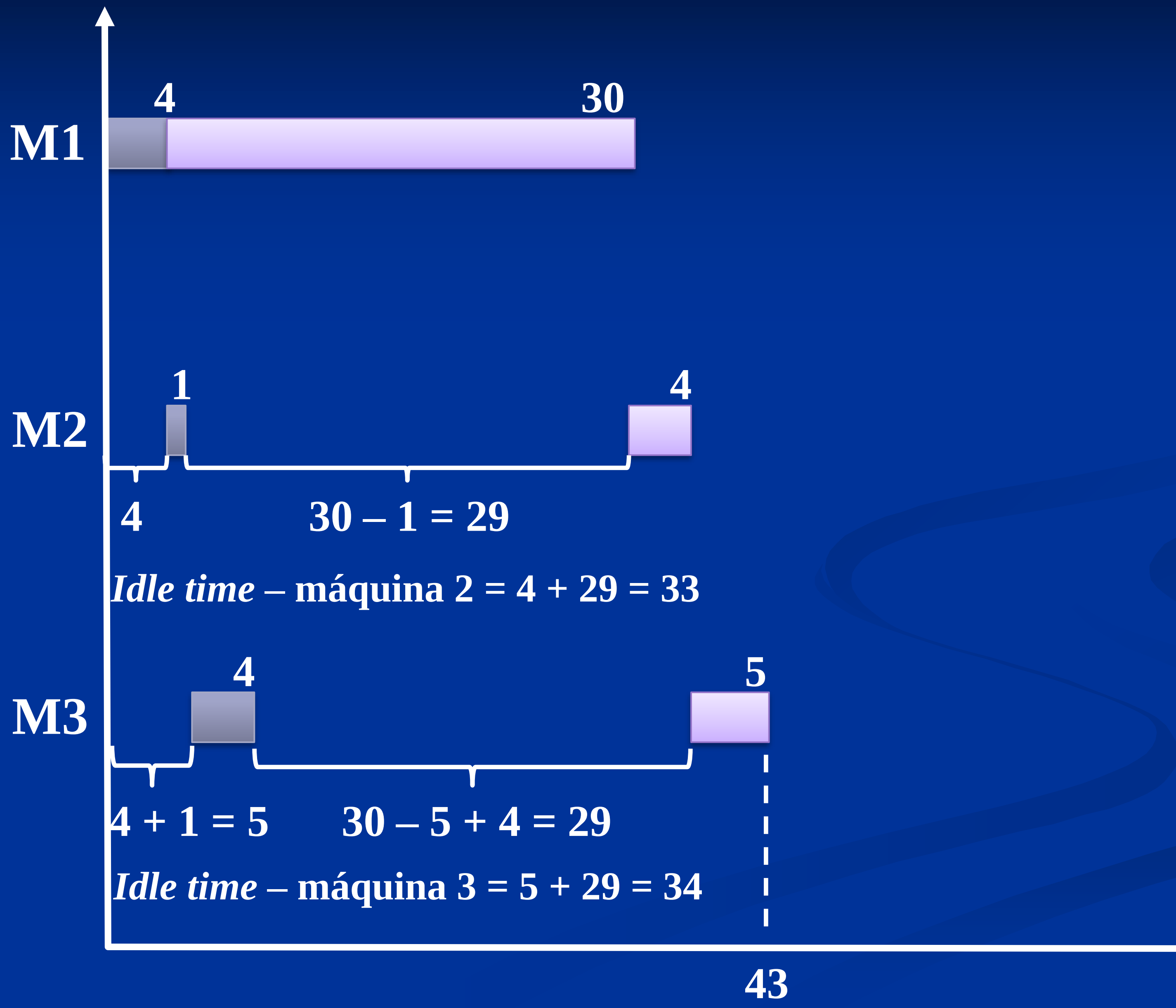


Segundo par


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

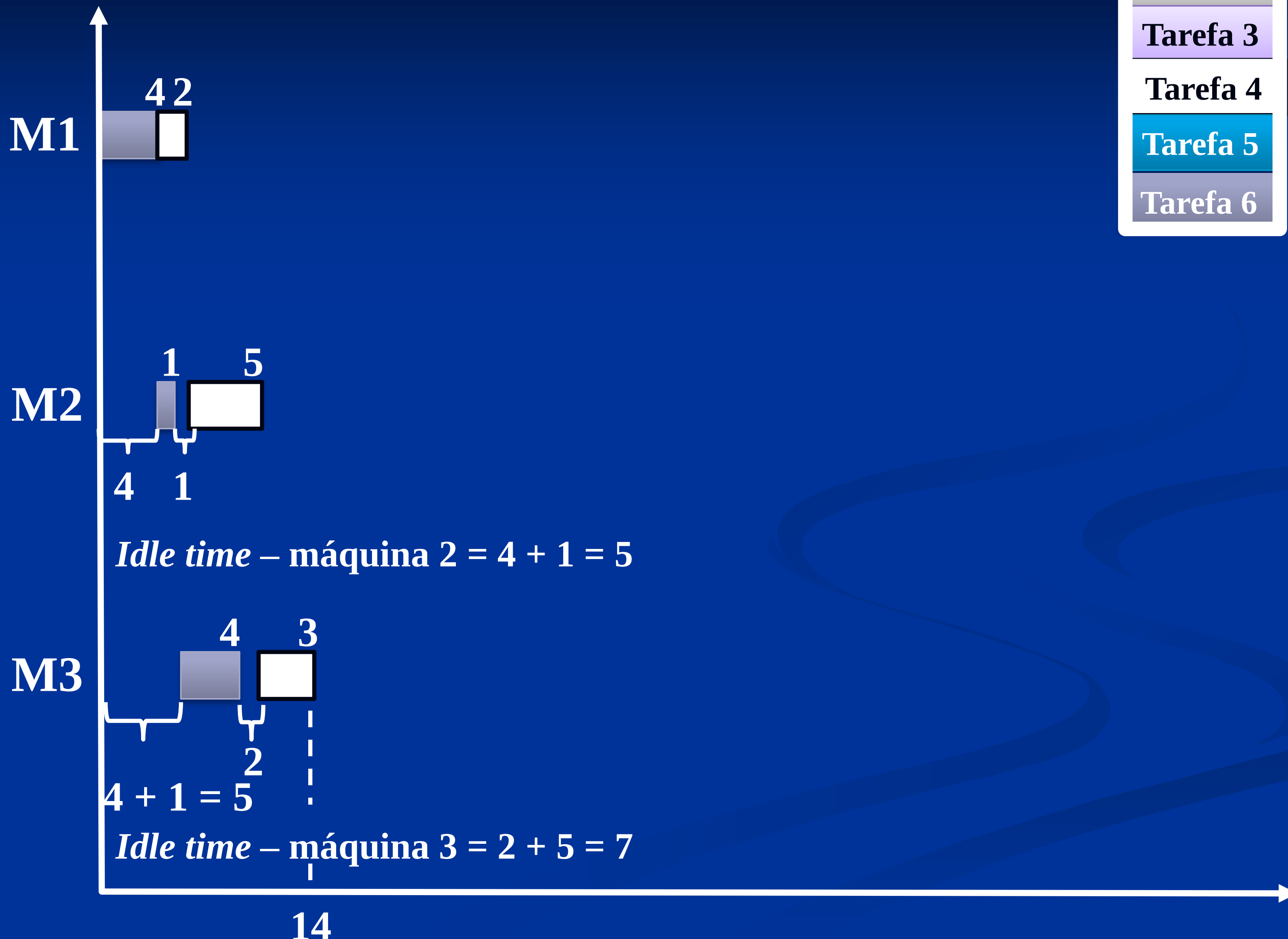

M1
4
6
M2
1
30
4
5
Idle time – máquina 2 = 4 + 5 = 9
M3
4
6
4 + 1 = 5
1 + 30 = 31
Idle time – máquina 3 = 5 + 31 = 36
46

Terceiro par

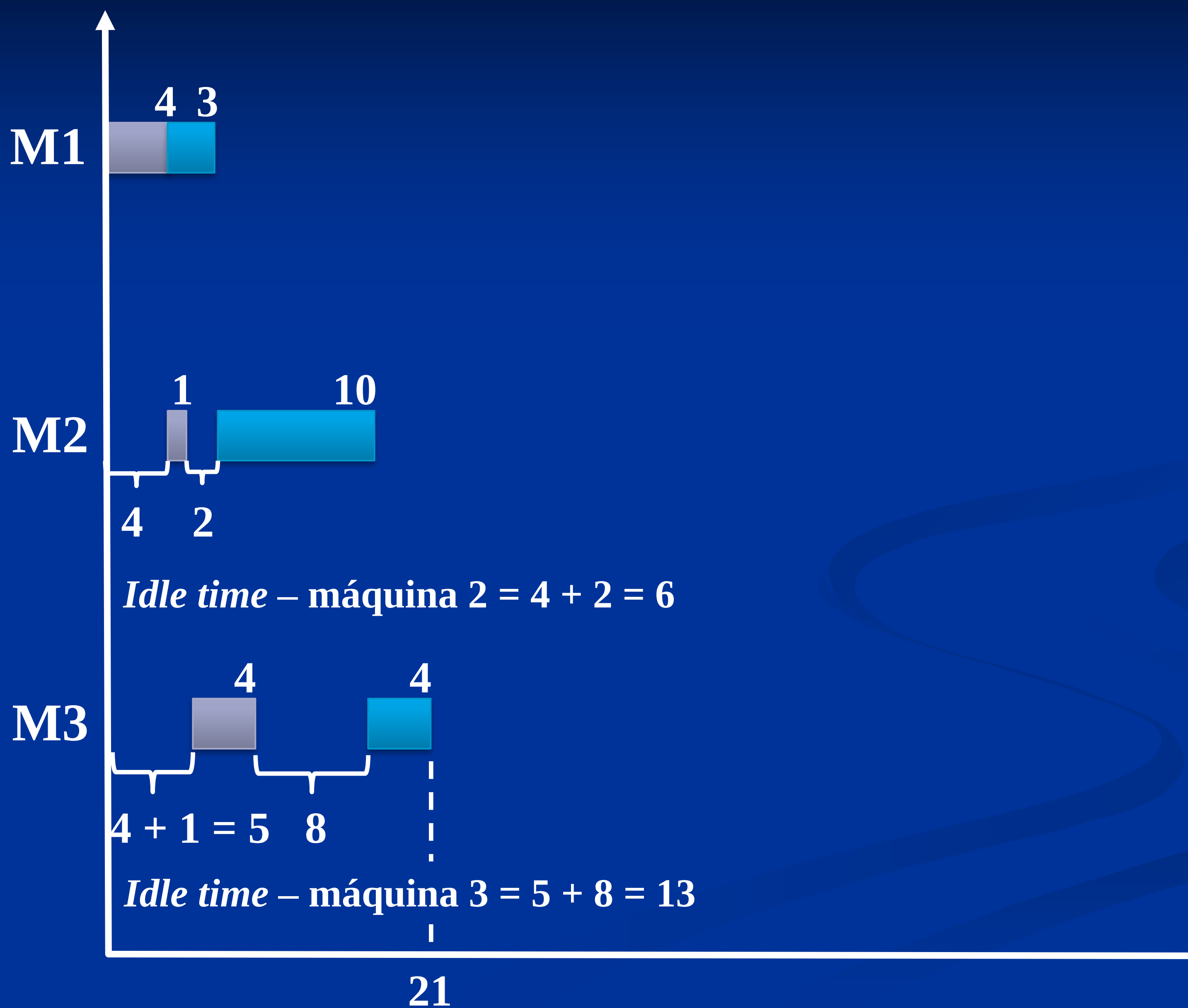


Quarto par

Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6



Quinto par



Idle time – máquina 2 = 4 + 2 = 6

Idle time – máquina 3 = 5 + 8 = 13

Término do passo 1

Tabela 2 completa

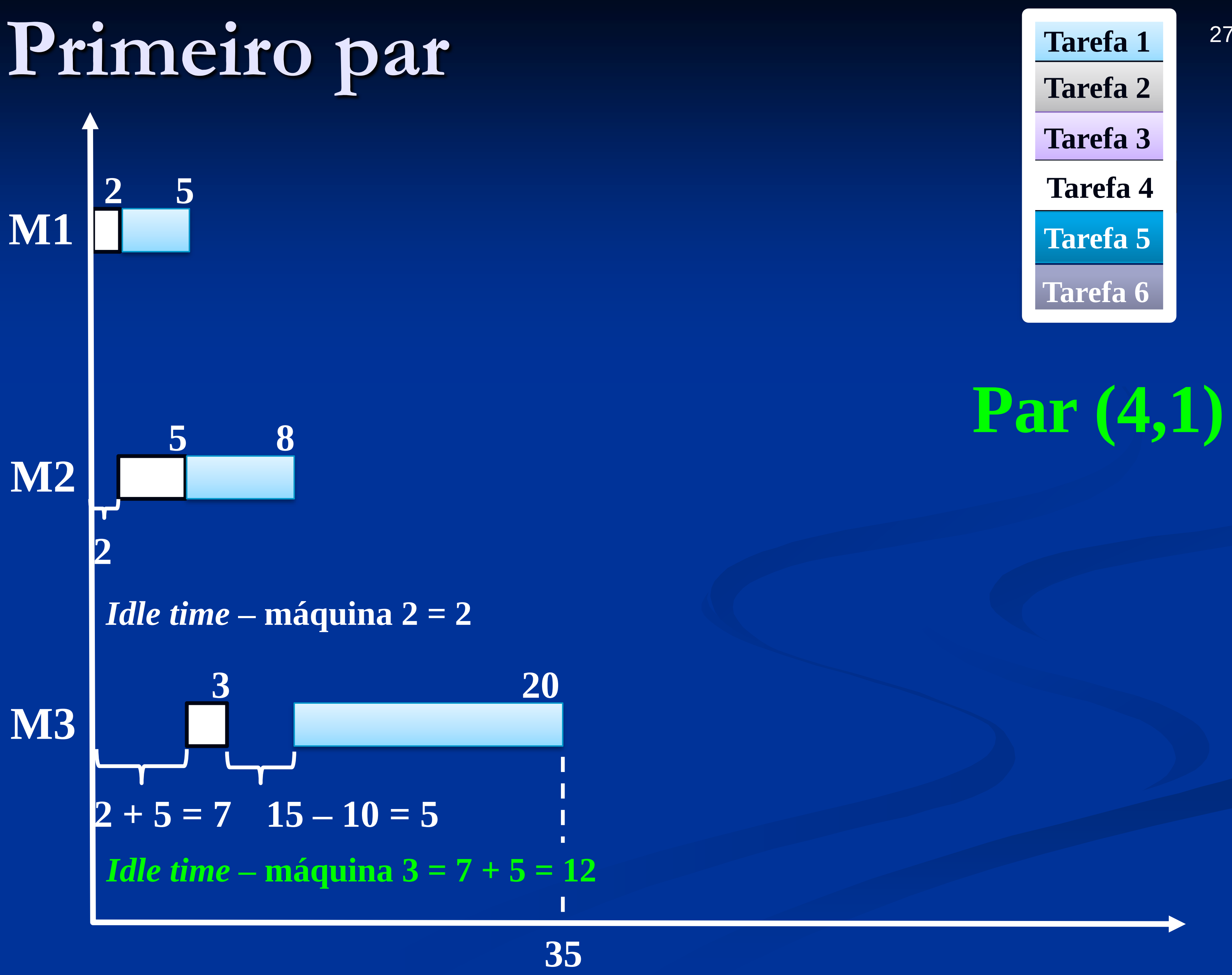
Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(1,2)	11	43	49	5	23
(1,3)	35	39	44	27	19
(1,4)	7	18	36	5	13
(1,5)	8	23	37	5	13
(1,6)	9	14	37	5	13
(2,1)	11	44	64	6	38
(2,3)	36	40	47	6	36
(2,4)	8	41	45	6	36
(2,5)	9	46	50	6	40
(2,6)	10	37	46	6	36
(3,1)	35	43	63	31	38
(3,2)	36	66	72	32	61
(3,4)	32	39	42	30	34
(3,5)	33	44	48	30	39
(3,6)	34	35	43	30	34
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	8	14	2	7
(5,1)	8	21	41	3	17
(5,2)	9	43	49	3	39
(5,3)	33	37	42	23	33
(5,4)	5	18	21	3	14
(5,6)	7	14	21	3	13
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	11	14	5	7
(6,5)	7	17	21	6	13

Pares selecionados na etapa 1, C(ab,m) mínimo $\Rightarrow$ (4,1), (4,5) e (4,6)

Passo 2 do procedimento do algoritmo MINIT

Passo 2 (a) – Considere os pares com vínculo (i, j) e (r, s) – o menor C(ab, m). Selecione o par com o tempo ocioso (*idle*) mínimo nas máquinas (M – 1, (M – 2) .... até a última máquina de modo a quebrar o empate. Se o vínculo for resolvido, processe o passo 2(c) até encontrar (a,b) como o par selecionado, caso contrário processe o passo 2(b).

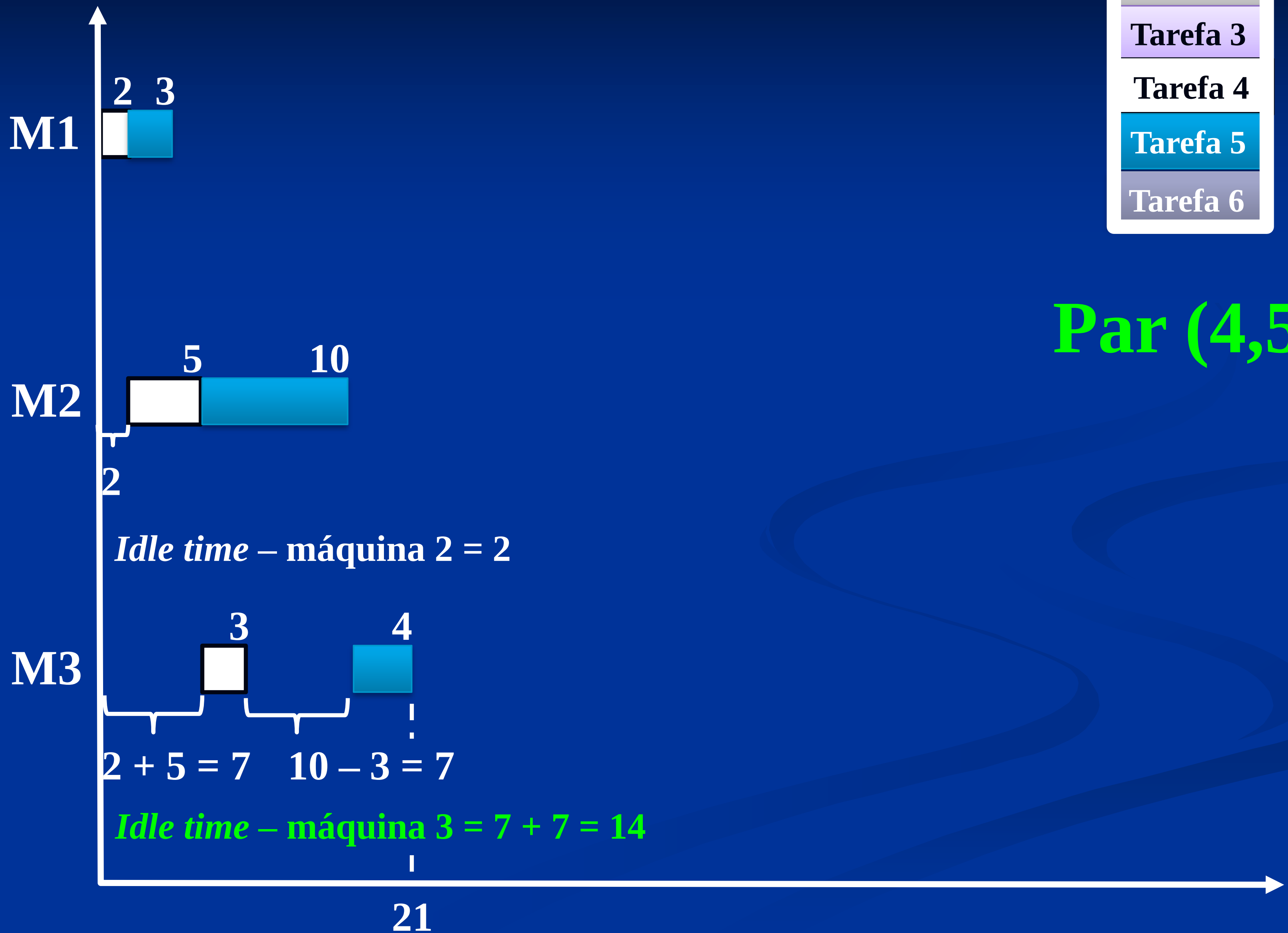
Primeiro par



Quarto par

Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

Par (4,5)



Quinto par

- Tarefa 1
- Tarefa 2
- Tarefa 3
- Tarefa 4
- Tarefa 5
- Tarefa 6

Par (4,6)

Passo 2 do procedimento do algoritmo MINIT

Passo 2 (c) – considere σ = ab (46), k igual a 2. Vá para o passo 3

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14
(4,6)	6	**8** 2 + 5 + 1 (4,1 ⇒ 4,2 ⇒ 6,2)	14	C(46,2) = 2 **2**	7

Tabela 1: Matriz dos tempos de processamento

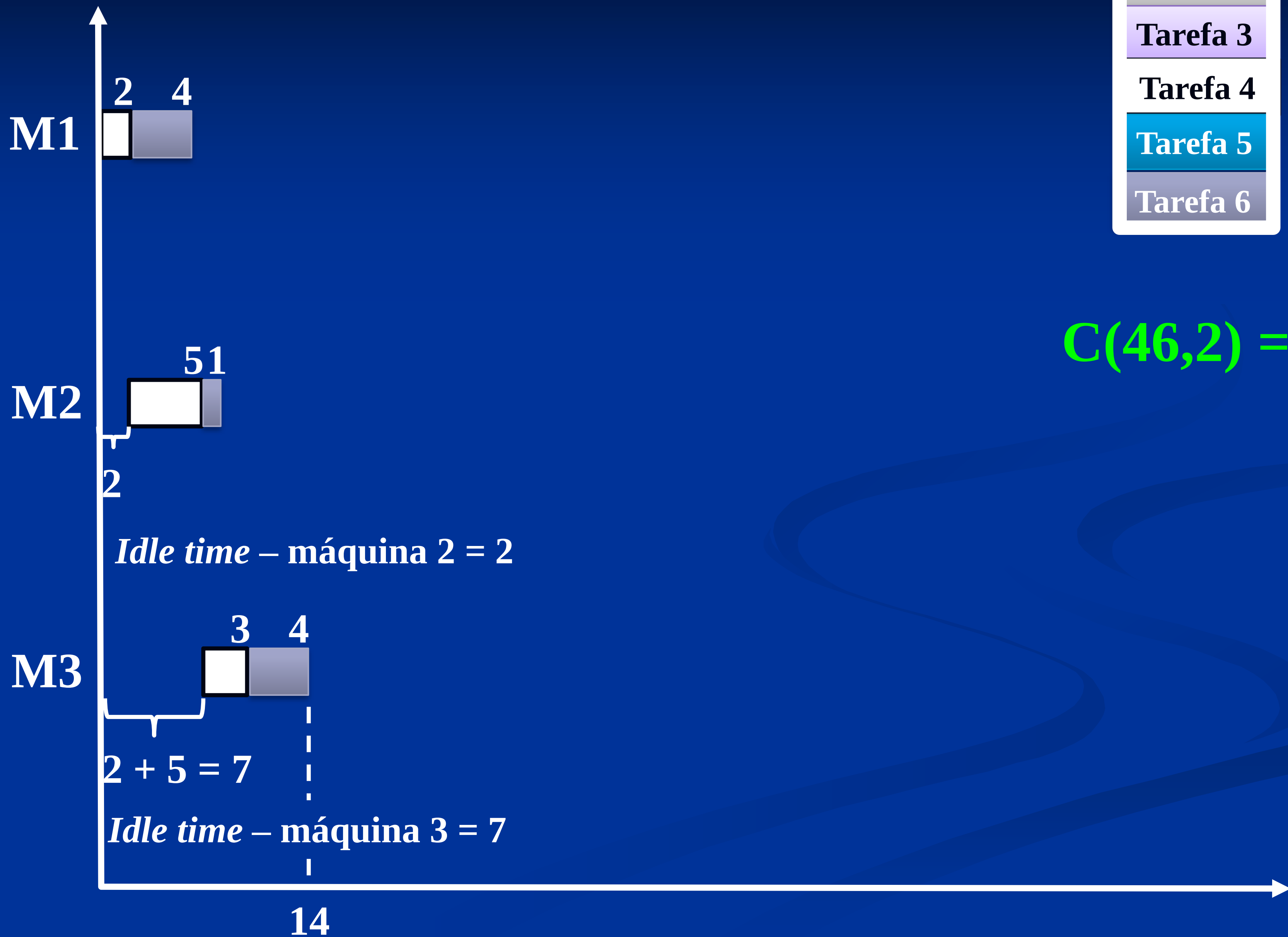
m (máquina) / i (tarefa)	1	2	3
operação →			
tarefa 4	2	5	3
6	4	1	4

4,1	4,2	4,3
6,1	6,2	6,3

Se $t_{6,1} \leq t_{4,2} \Rightarrow$ *idle* $= t_{4,1}$

Quinto par

- Tarefa 1
- Tarefa 2
- Tarefa 3
- Tarefa 4
- Tarefa 5
- Tarefa 6



$C(46,2) = 2$

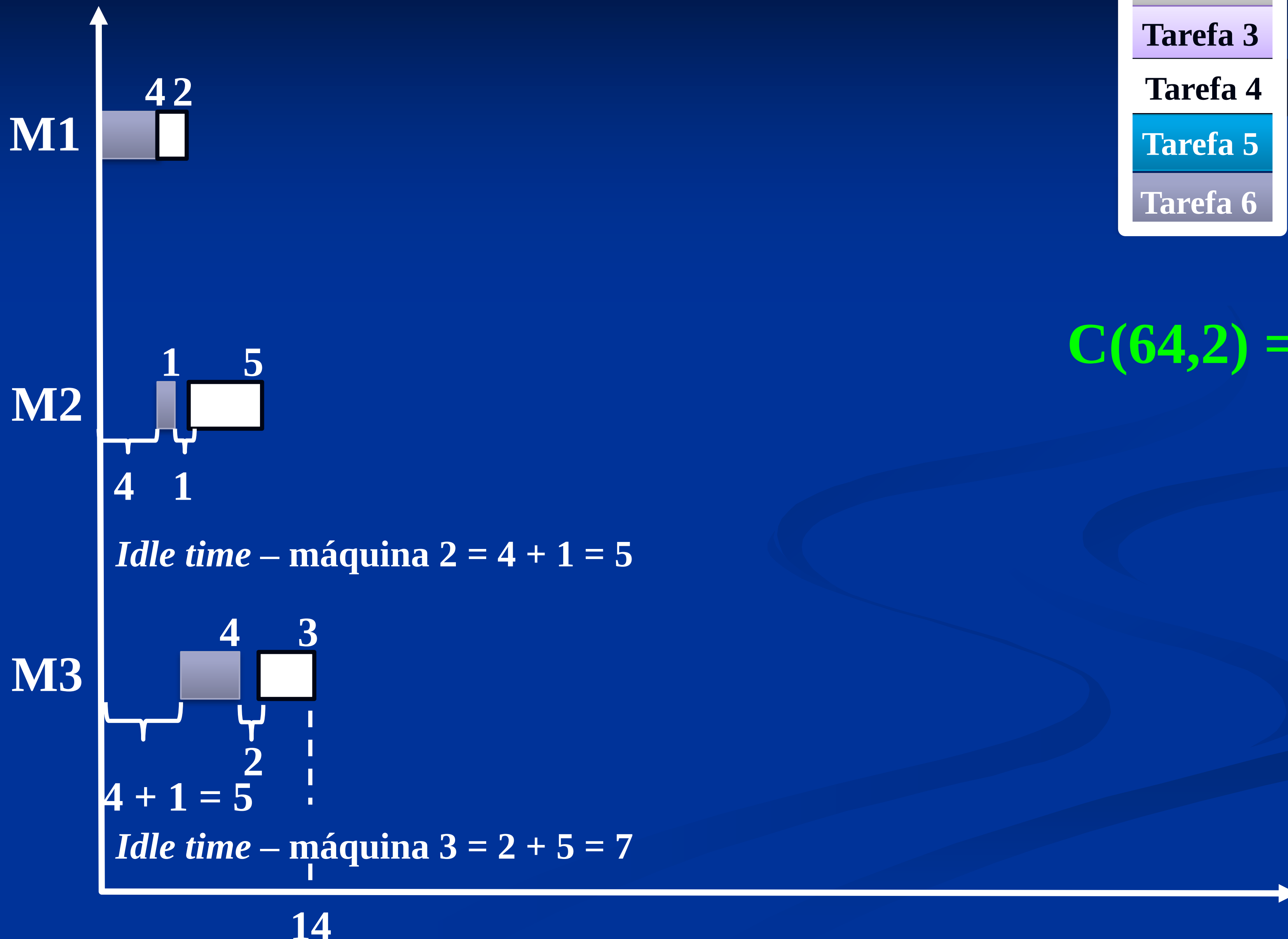
Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(6,1)	9	17	37	8	13
(6,2)	10	40	46	9	36
(6,3)	34	38	43	33	34
(6,4)	6	**11** 6 + 5	14	**C(64,2) = 5** 5	**7**
(6,5)	7	17	21	6	13

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
tarefa 6	4	1	4
4	2	5	3

Se $t_{4,1} > t_{6,2} \Rightarrow$
idle **$= (t_{4,1} - t_{6,2}) + t_{6,1}$**
$(2 - 1) + 4 = 5$

6,1	6,2	6,3
4,1	4,2	4,3

Quarto par



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

C(64,2) = 5

Passo 3 do procedimento do algoritmo MINIT

Passo 3 – entre as tarefas não alocadas, examine cada uma das tarefas e calcule C(σa, m) para todo a e m. Encontre o mínimo C(σa, M). Se um único σa existe, vá para o passo 5; caso contrário vá para o passo 4.

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,1)	7	15	35	2	12
(4,2)	8	38	44	3	35
(4,3)	32	36	41	27	33
(4,5)	5	17	21	2	14

Tempos de processamento das tarefas não alocadas

Par	σ	a	$C(\sigma a, m)$	
			$m = 2$	$m = 3$
(4,6,1)	46	1		
(4,6,2)		2		
(4,6,3)		3		
(4,6,5)		5		

Máquina 2 & 3 par (461)

Idle time

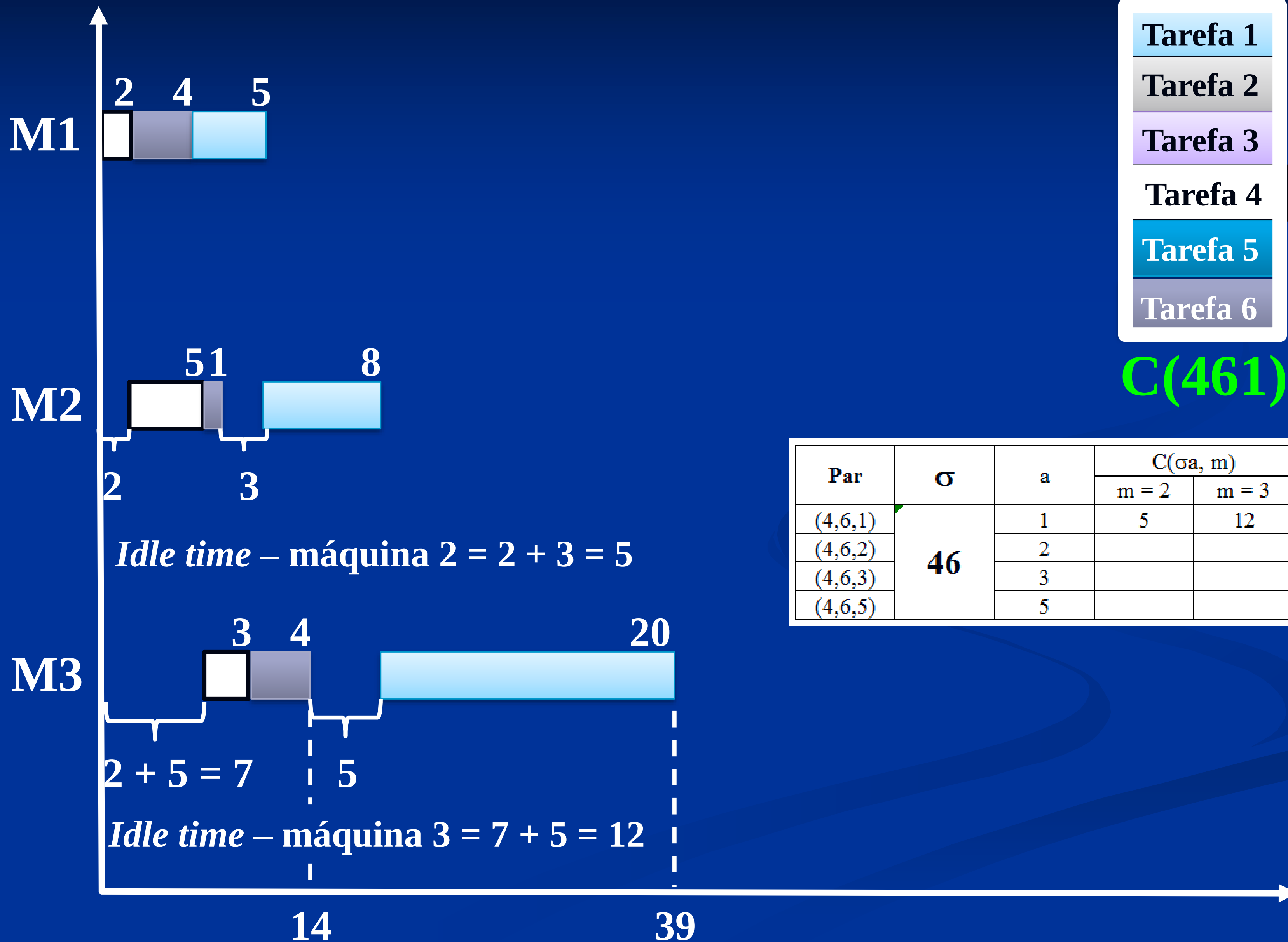
Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,6)	6	8 461,1 > 46,2	14 461,2 > 46,3	2	7
(4,6,1)	6 + 5 = 11	11+ 8 = 19	19 + 20 = 39	2 + (11 – 8) = 5	7 + (19 – 14) = 12

Tabela 1: Matriz dos tempos de processamento			
m (máquina) / i (tarefa)	1	2	3
operação			
4	2	5	3
6	4	1	4
1	5	8	20

tarefa

4,1	4,2	4,3
6,1	6,2	6,3
1,1	1,2	1,3

Primeiro grupo de três tarefas



C(461)

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,1)	46	1	5	12
(4,6,2)		2		
(4,6,3)		3		
(4,6,5)		5		

Máquina 2 & 3 par (462)

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,6)	6	8 462,1 > 46,2	14 462,2 > 46,3	2	7
(4,6,2)	6 + 6 = 12	12+ 30 = 42	42 + 6 = 48	2 + (12 – 8) = 6	7 + (42 – 14) = 35

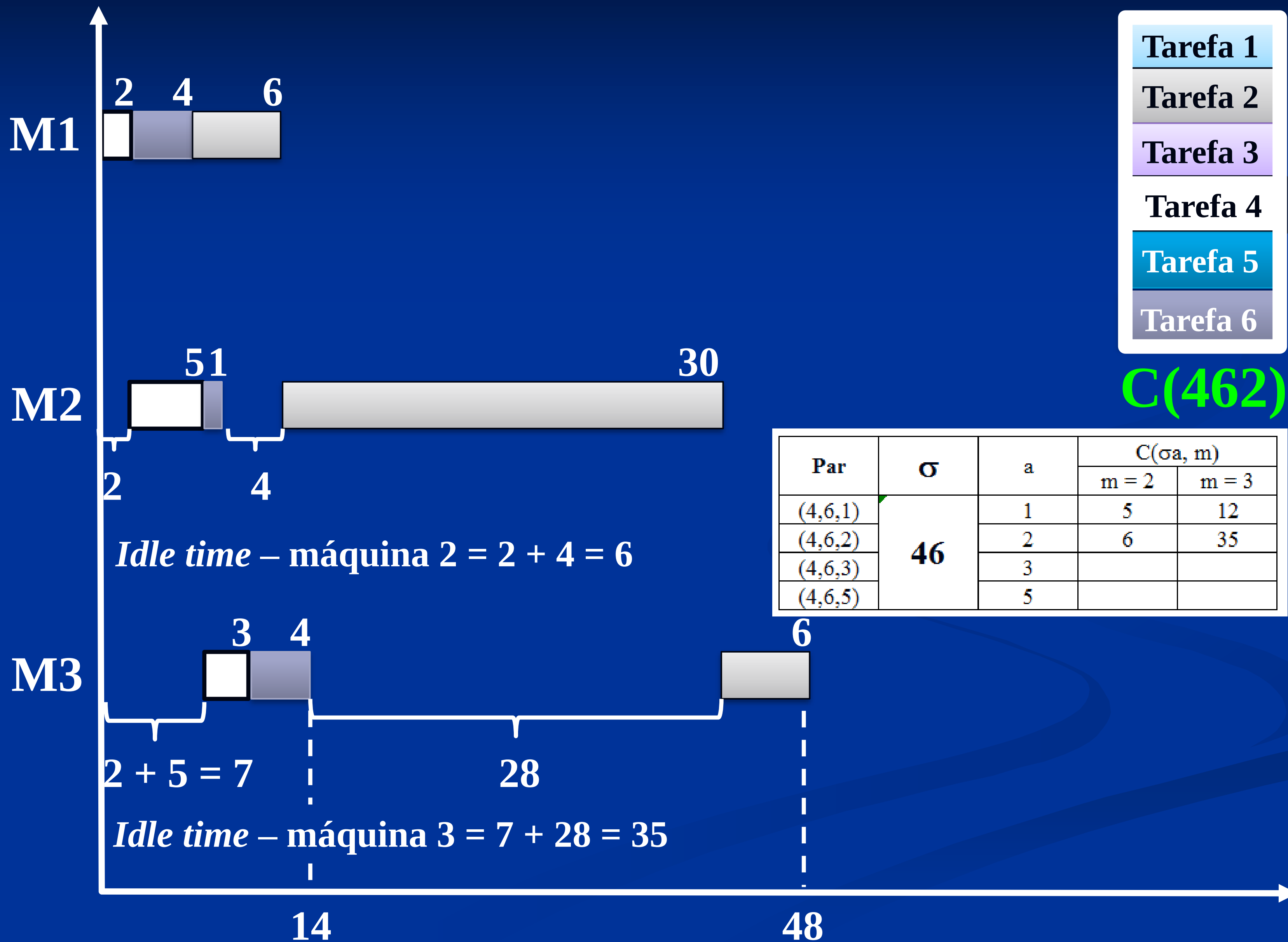
Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1 operação	2	3
4	2	5	3
6	4	1	4
2	6	30	6

tarefa

4,1	4,2	4,3
6,1	6,2	6,3
2,1	2,2	2,3

Segundo grupo de três tarefas



Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,1)	46	1	5	12
(4,6,2)		2	6	35
(4,6,3)		3		
(4,6,5)		5		

Máquina 2 & 3 par (463)

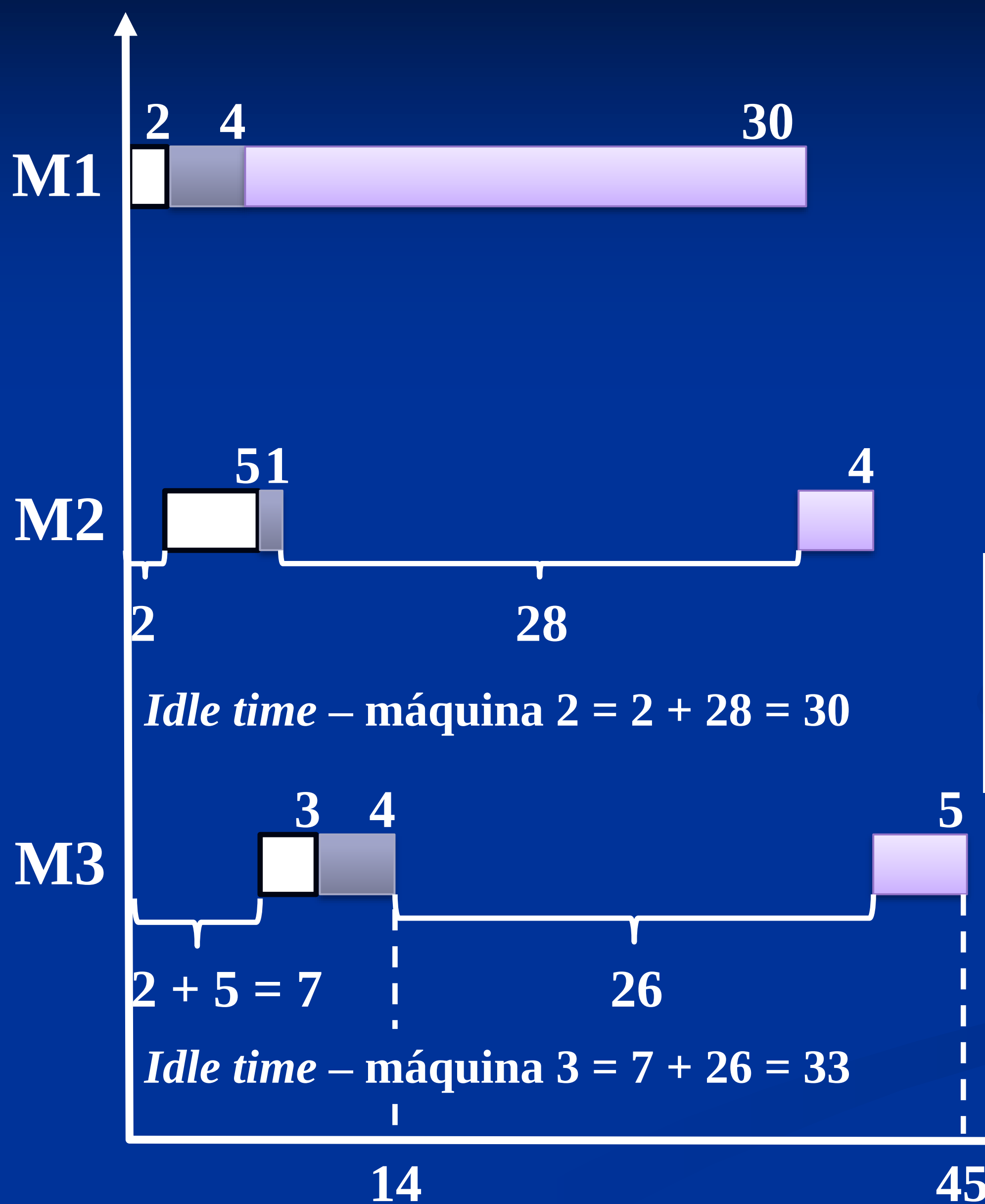
Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	T(ab, m)			C(ab, m)	
(a,b)	1	2	3	2	3
(4,6)	6	8 463,1 > 46,2	14 463,2 > 46,3	2	7
(4,6,3)	6 + 30 = 36	36+ 4 = 40	40 + 5 = 45	2 + (36 – 8) = 30	7 + (40 – 14) = 33

Tabela 1: Matriz dos tempos de processamento			
m (máquina) / i (tarefa)	1 operação	2	3
4	2	5	3
6	4	1	4
3	30	4	5
	4,1	4,2	4,3
	6,1	6,2	6,3
	3,1	3,2	3,3

tarefa

Terceiro grupo de três tarefas



- Tarefa 1
- Tarefa 2
- Tarefa 3
- Tarefa 4
- Tarefa 5
- Tarefa 6

C(463)

Par	σ	a	C(σa, m) m = 2	C(σa, m) m = 3
(4,6,1)	46	1	5	12
(4,6,2)		2	6	35
(4,6,3)		3	30	33
(4,6,5)		5		

Máquina 2 & 3 par (465)

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,6)	6	8 465,1 > 46,2	14 465,2 > 46,3	2	7
(4,6,5)	6 + 3 = 9	9 + 10 = 19	19 + 4 = 23	2 + (9 – 8) = 3	7 + (19 – 14) = 12

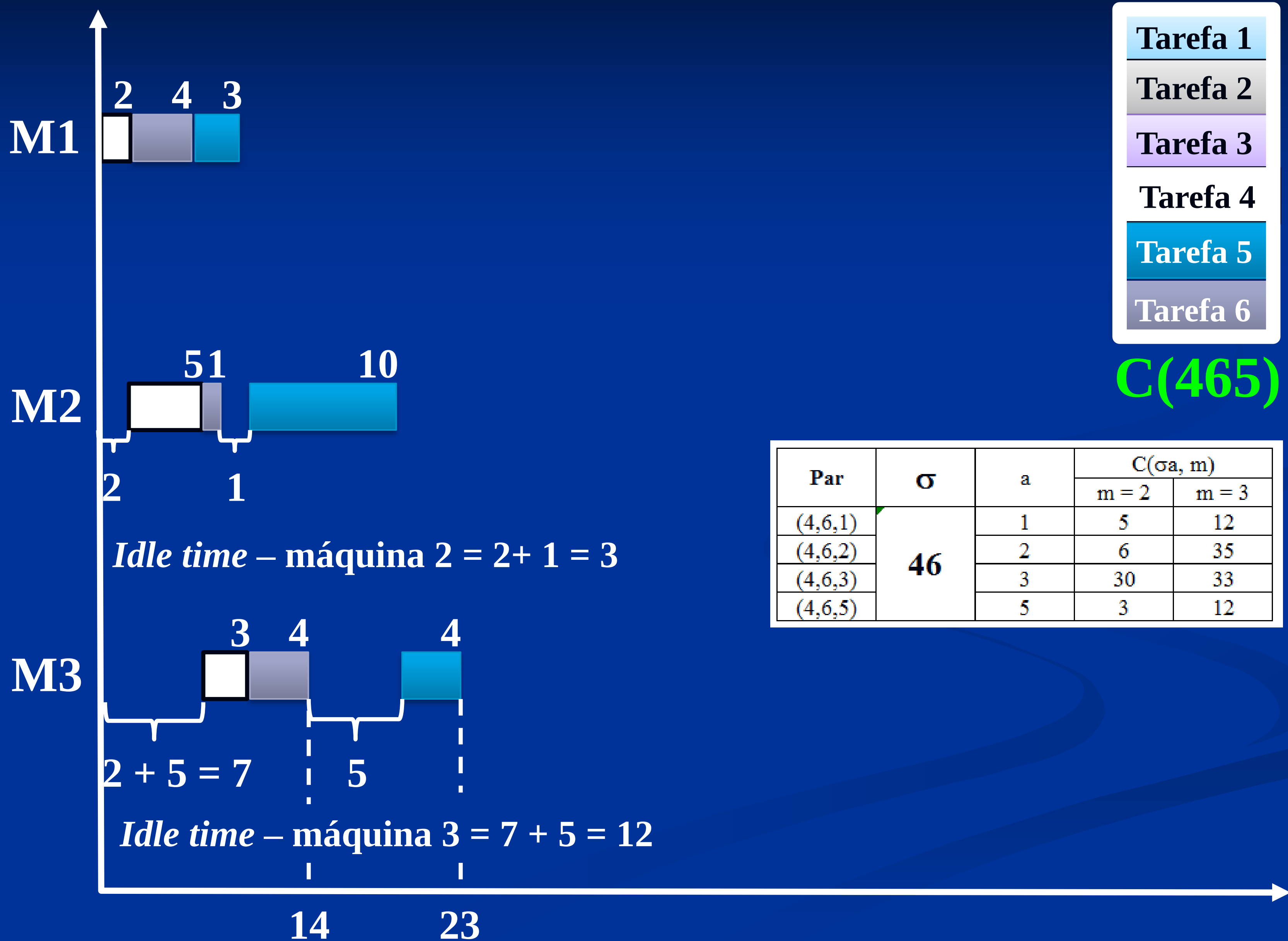
Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1	2	3
operação			
4	2	5	3
6	4	1	4
5	3	10	4

tarefa

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

Quarto grupo de três tarefas



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

C(465)

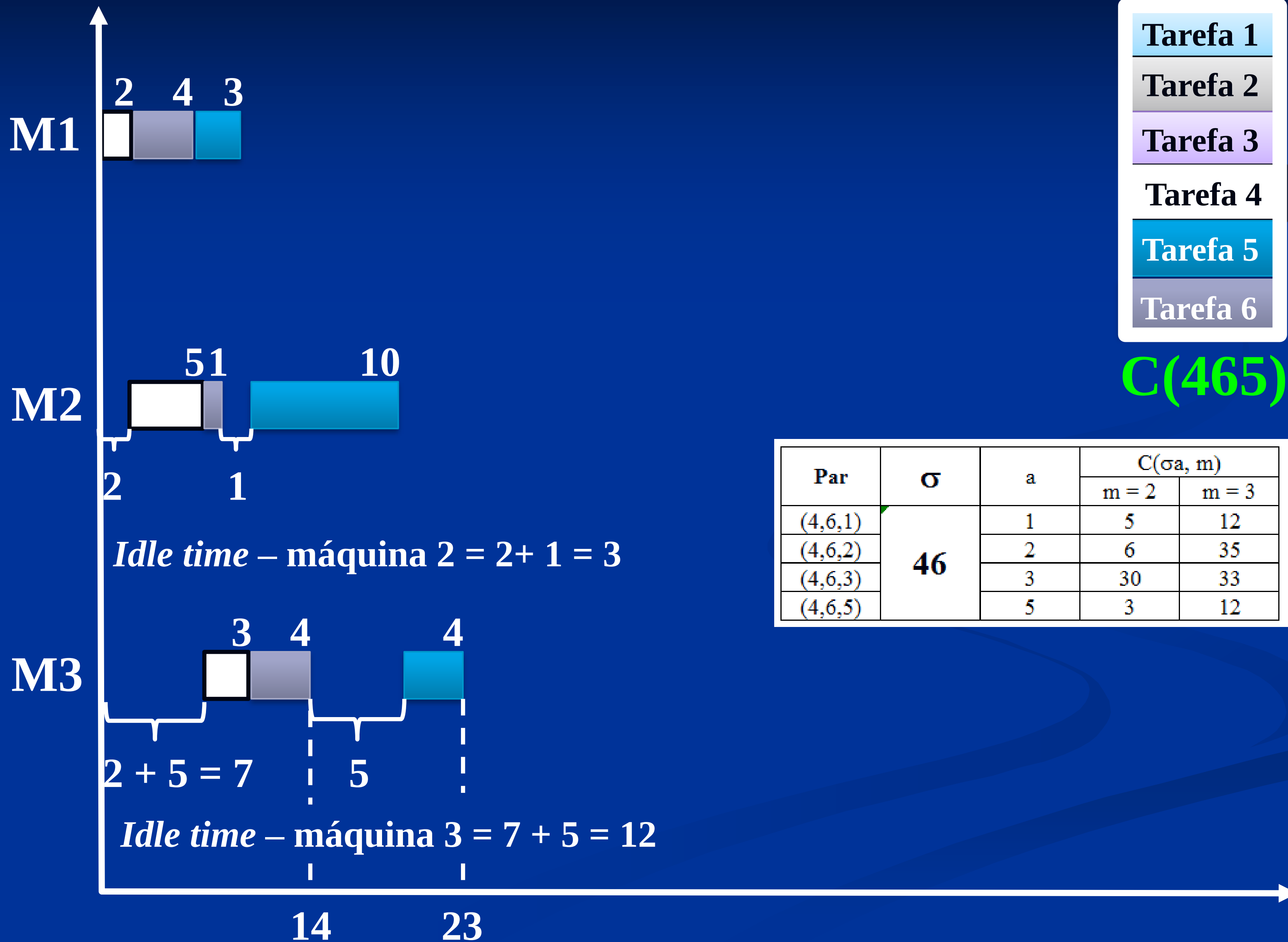
Par	σ	a	C(σa, m) m = 2	C(σa, m) m = 3
(4,6,1)	46	1	5	12
(4,6,2)		2	6	35
(4,6,3)		3	30	33
(4,6,5)		5	3	12

Tempos de processamento das tarefas não alocadas

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,1)	46	1	5	12
(4,6,2)		2	6	35
(4,6,3)		3	30	33
(4,6,5)		5	3	12

Menor C(σa, m)

Quarto grupo de três tarefas



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

C(465)

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,1)	46	1	5	12
(4,6,2)		2	6	35
(4,6,3)		3	30	33
(4,6,5)		5	3	12

Tempos de processamento das tarefas não alocadas

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,1)	46	1	5	12
(4,6,2)		2	6	35
(4,6,3)		3	30	33

Tempos de processamento das tarefas não alocadas

Par	σ	a	$C(\sigma a, m)$	
			$m = 2$	$m = 3$
(4,6,5,1)	**465**			
(4,6,5,2)				
(4,6,5,3)				

Máquina 2 & 3 par (4651)

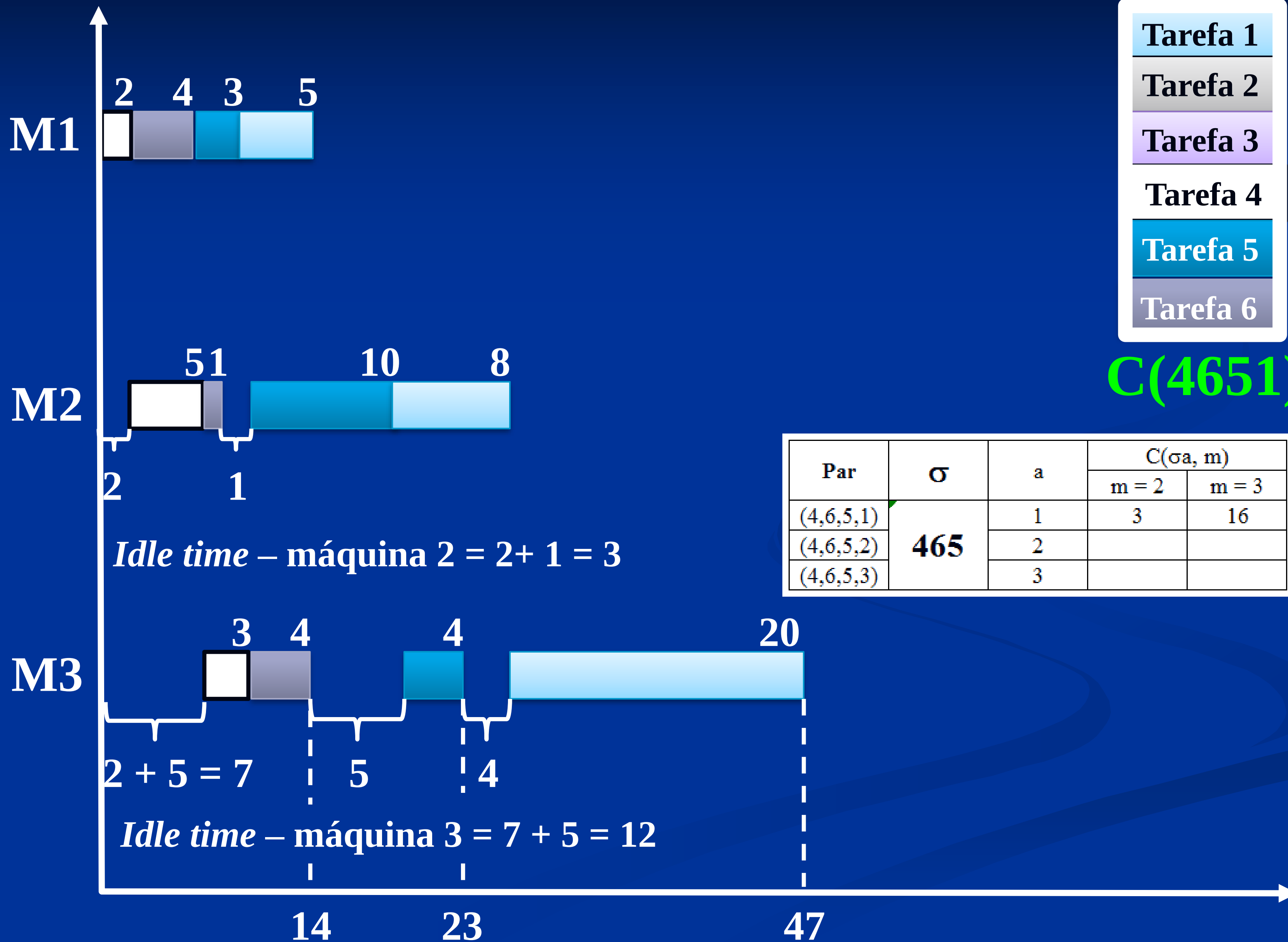
Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(**4**,6,5)	9	**4651,1 < 465,2** **19**	**4651,2 > 465,3** **23**	**3**	**12**
(**4**,6,5,1)	9 + 5 = **14**	**19** + 8 = **27**	27 + 20 = 47	3 = 3	**12** + **(27 – 23)** = 16

Tabela 1: Matriz dos tempos de processamento			
i (tarefa) m (máquina)	**1** **operação**	**2**	**3**
4	2	5	3
tarefa 6	4	1	4
5	3	10	4
1	5	8	20

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

Primeiro grupo de quatro tarefas



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6

C(4651)

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,5,1)	465	1	3	16
(4,6,5,2)		2		
(4,6,5,3)		3		

Máquina 2 & 3 par (4652)

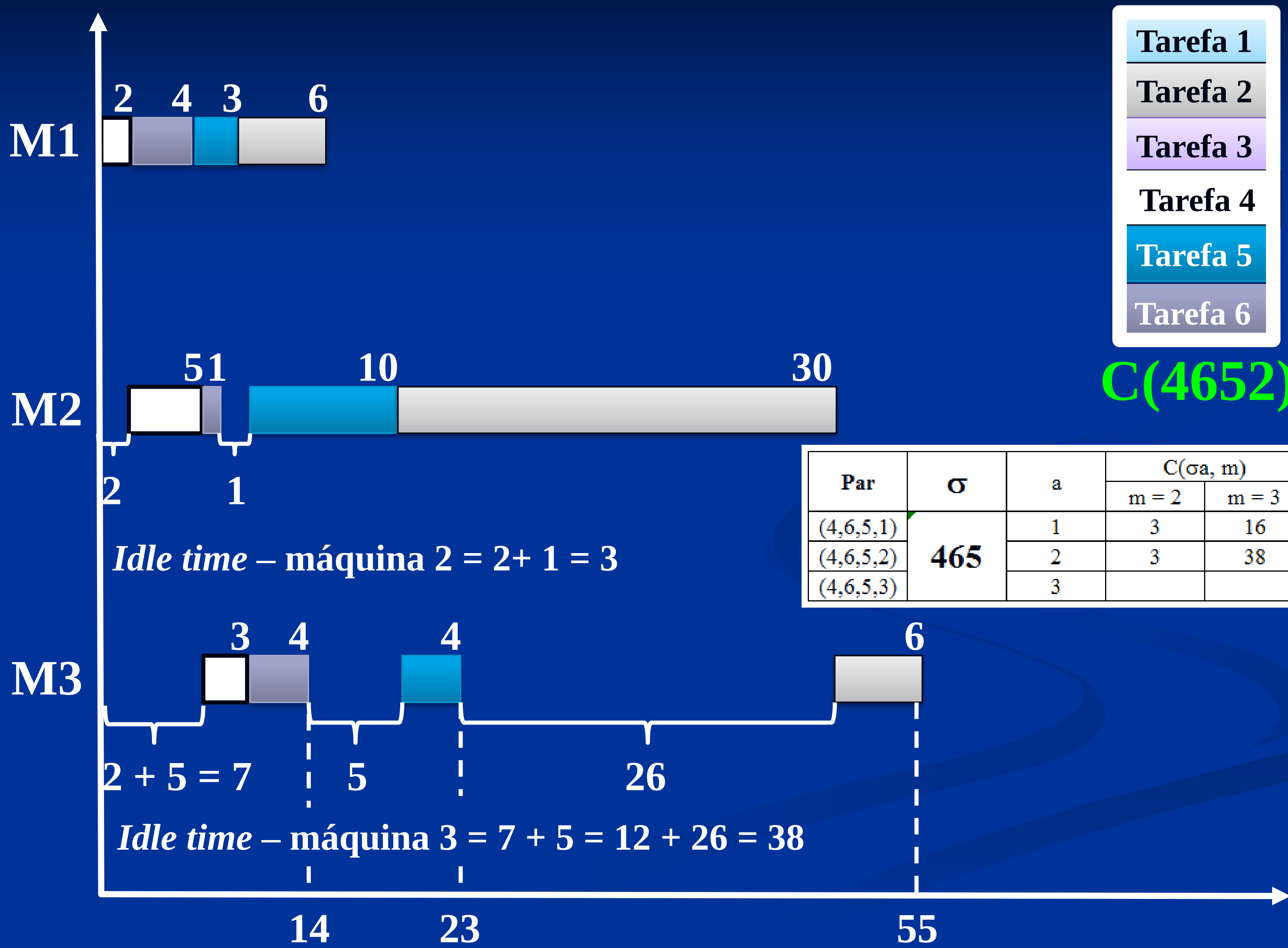
Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,6,5)	9	4652,1 < 465,2 **19**	4652,2 > 465,3 **23**	**3**	**12**
(4,6,5,2)	9 + 6 = **15**	**19** + 30 = **49**	49 + 6 = 55	3 = 3	**12** + **(49 – 23)** = 38

Tabela 1: Matriz dos tempos de processamento				
i (tarefa)	m (máquina)	**1** operação	**2**	**3**
tarefa	4	2	5	3
	6	4	1	4
	5	3	10	4
	2	6	30	6

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

Segundo grupo de quatro tarefas



Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,5,1)	465	1	3	16
(4,6,5,2)		2	3	38
(4,6,5,3)		3		

Máquina 2 & 3 par (4653)

Idle time

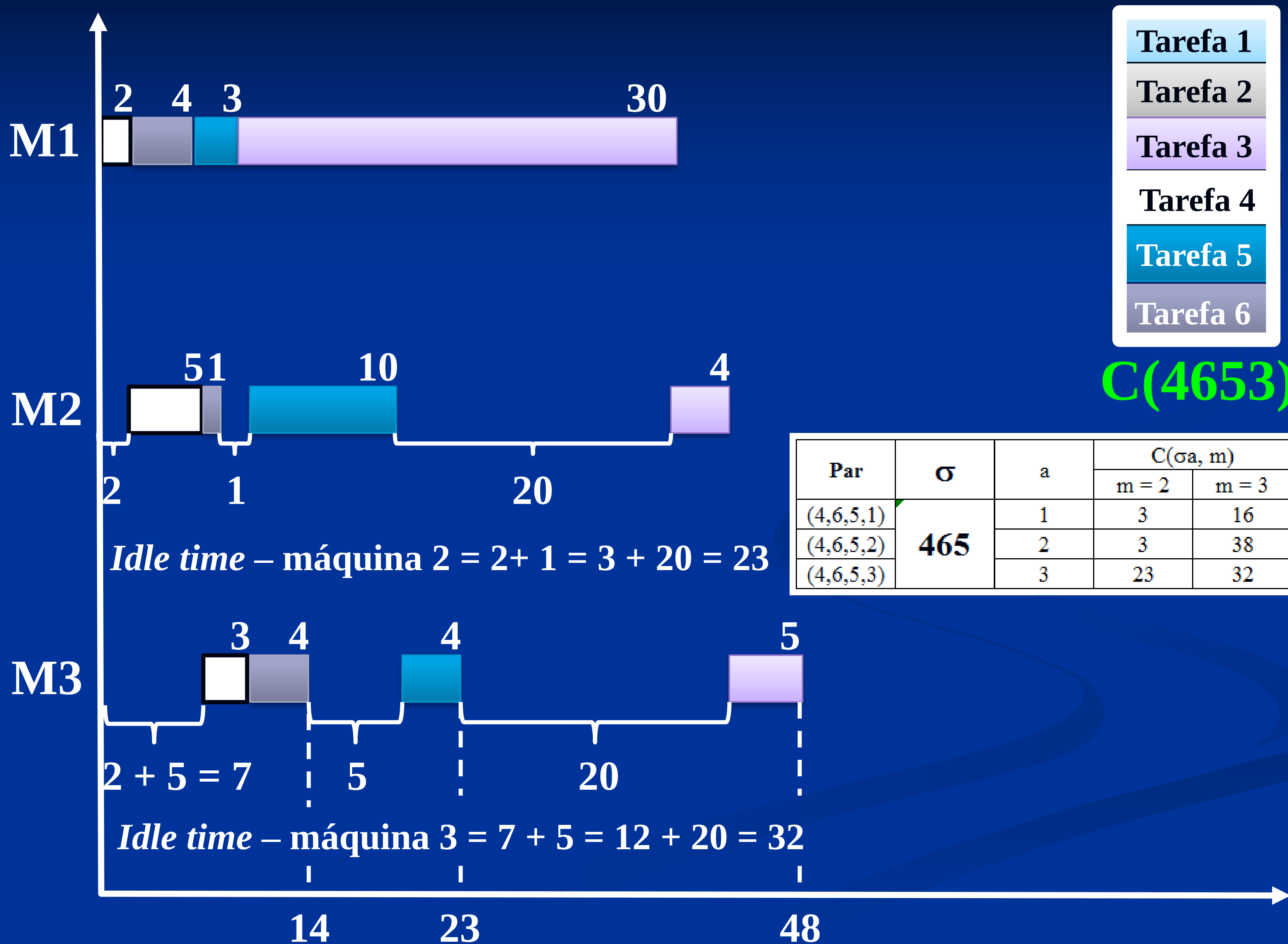
Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,6,5)	9	4653,1 > 465,2 **19**	4653,2 > 465,3 **23**	**3**	**12**
(4,6,5,3)	9 + 30 = **39**	**39** + 4 = **43**	43 + 5 = 48	3 + (39 – 19) = 23	**12** + **(43 – 23)** = 32

Tabela 1: Matriz dos tempos de processamento			
m (máquina) i (tarefa)	**1** **operação**	**2**	**3**
4	2	5	3
6	4	1	4
5	3	10	4
3	30	4	5

tarefa

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

Terceiro grupo de quatro tarefas

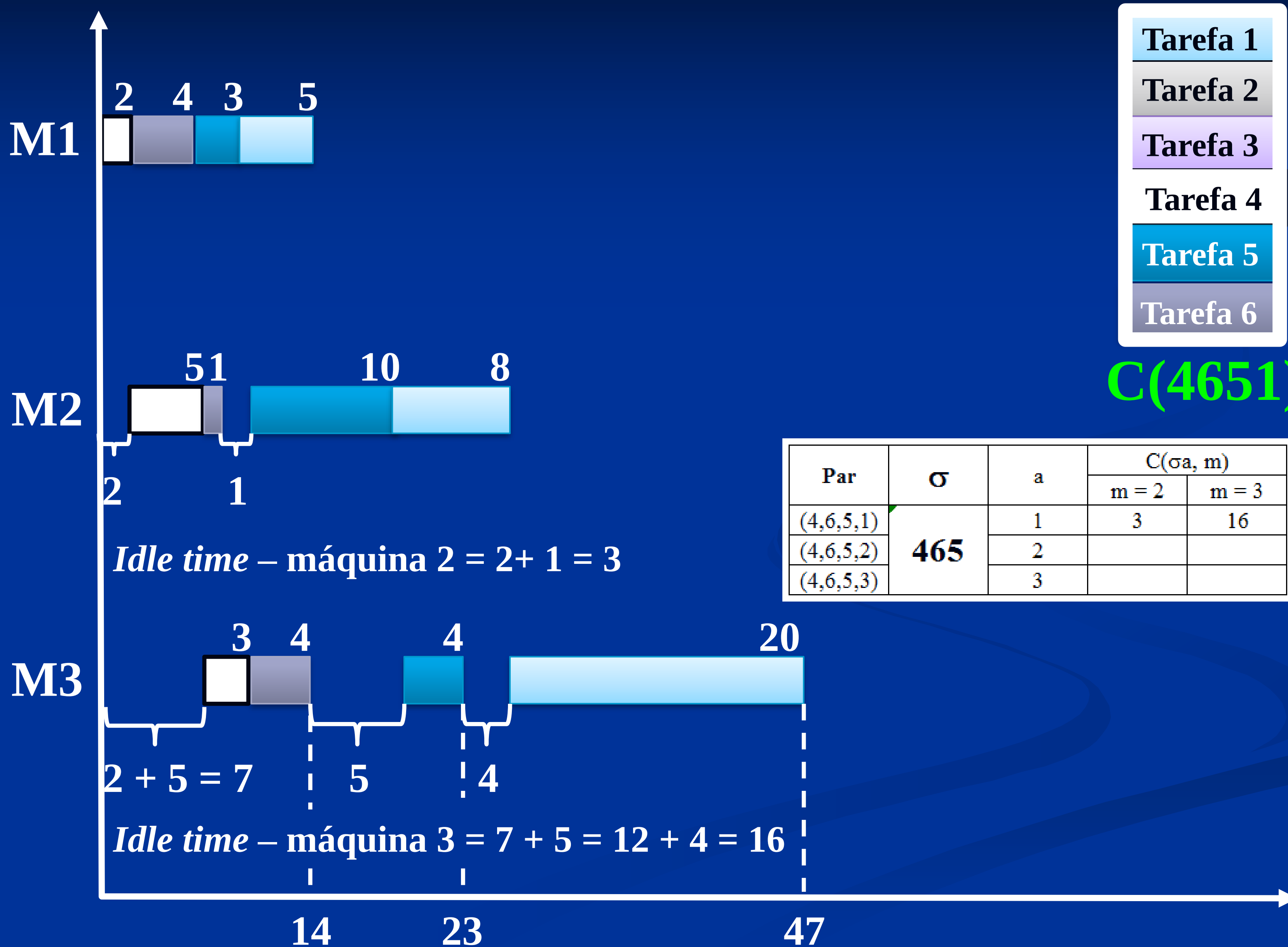


Par	σ	a	C(σa, m) m = 2	C(σa, m) m = 3
(4,6,5,1)	465	1	3	16
(4,6,5,2)		2	3	38
(4,6,5,3)		3	23	32

Tempos de processamento das tarefas não alocadas

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,5,1)	465	1	3	16
(4,6,5,2)		2	3	38
(4,6,5,3)		3	23	32

Primeiro grupo de quatro tarefas



C(4651)

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,5,1)	465	1	3	16
(4,6,5,2)		2		
(4,6,5,3)		3		

Tempos de processamento das tarefas não alocadas

Par	σ	a	$C(\sigma a, m)$	
			$m = 2$	$m = 3$
(4,6,5,1,2)				
(4,6,5,1,3)				

Máquina 2 & 3 par (46512)

Idle time

Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)

Par	T(ab, m)			C(ab, m)	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,6,5,1)	14	46512,1 < 4651,2 27	46512,2 > 4651,3 47	3	16
(4,6,5,1,2)	14 + 6 = 20	27 + 30 = 57	57 + 6 = 63	3 = 3	16 + (57 – 47) = 26

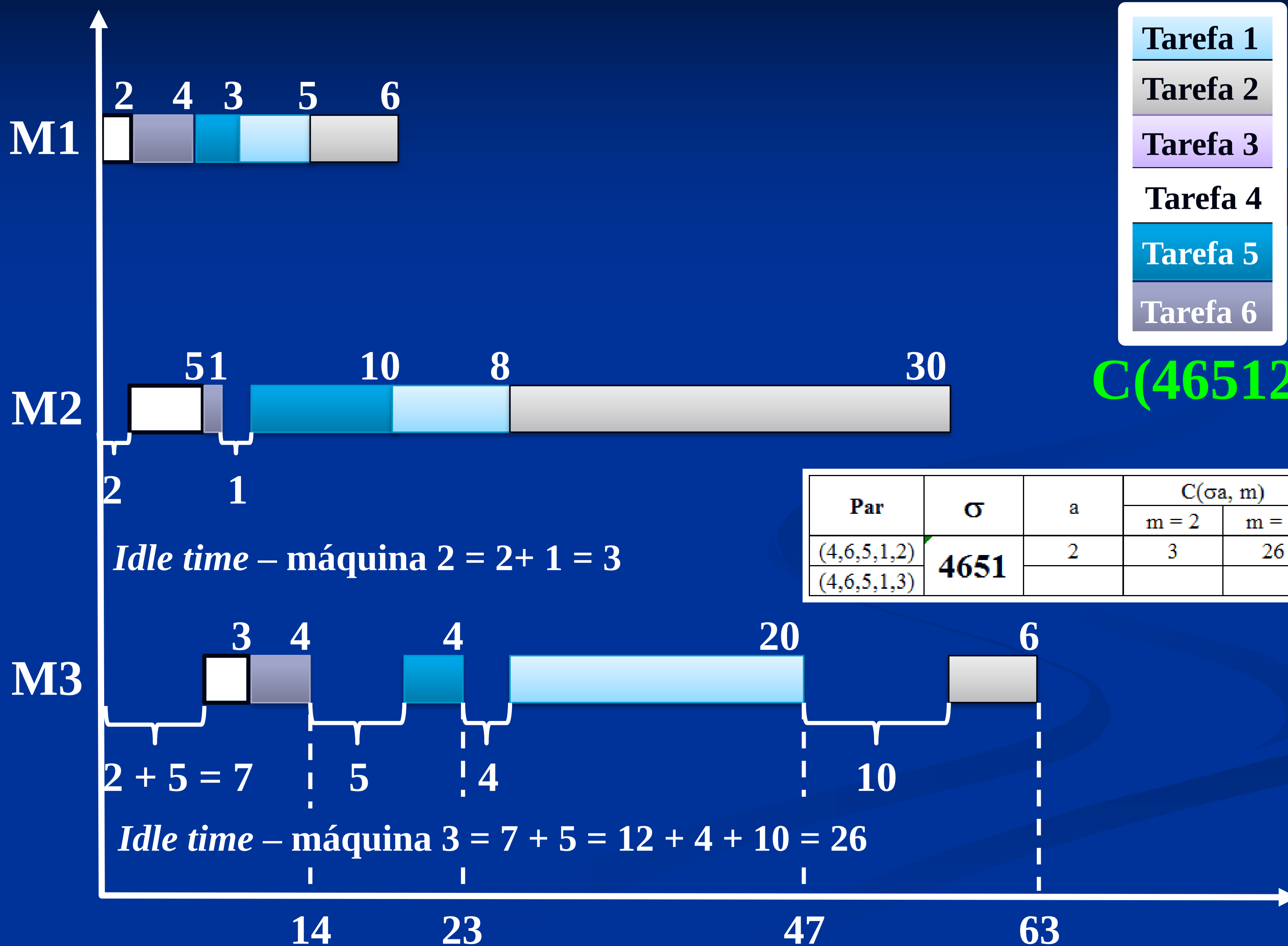
Tabela 1: Matriz dos tempos de processamento

i (tarefa) \ m (máquina)	1 operação	2	3
4	2	5	3
6	4	1	4
5	3	10	4
1	5	8	20
2	6	30	6

tarefa

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

Primeiro grupo de cinco tarefas



C(46512)

Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,5,1,2)	4651	2	3	26
(4,6,5,1,3)				

Máquina 2 & 3 par (46513)

Idle time

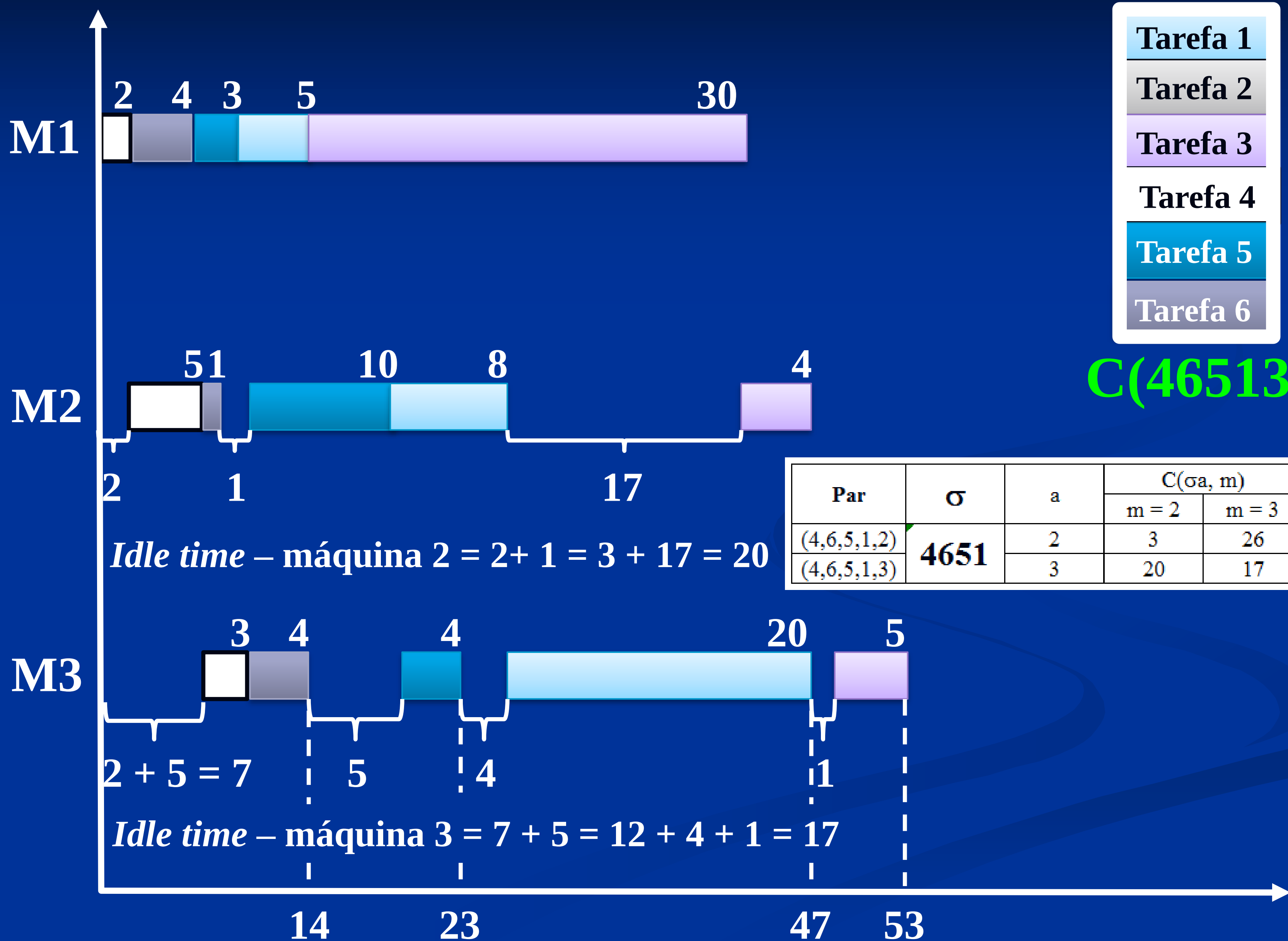
Tabela 2 – Tempos de execução e ociosidade dos vários pares (1ª parte)					
Par	**T(ab, m)**			**C(ab, m)**	
(a,b)	**1**	**2**	**3**	**2**	**3**
(4,6,5,1)	14	**46513,1 < 4651,2** 27	**46513,2 > 4651,3** 47	3	16
(4,6,5,1,3)	14 + 30 = 44	44 + 4 = 48	48 + 5 = 53	3 + (44 – 27) = 20	16 + (48 – 47) = 17

Tabela 1: Matriz dos tempos de processamento

m (máquina) / i (tarefa)		1 operação	2	3
tarefa	4	2	5	3
	6	4	1	4
	5	3	10	4
	1	5	8	20
	3	30	4	5

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

Segundo grupo de cinco tarefas



C(46513)

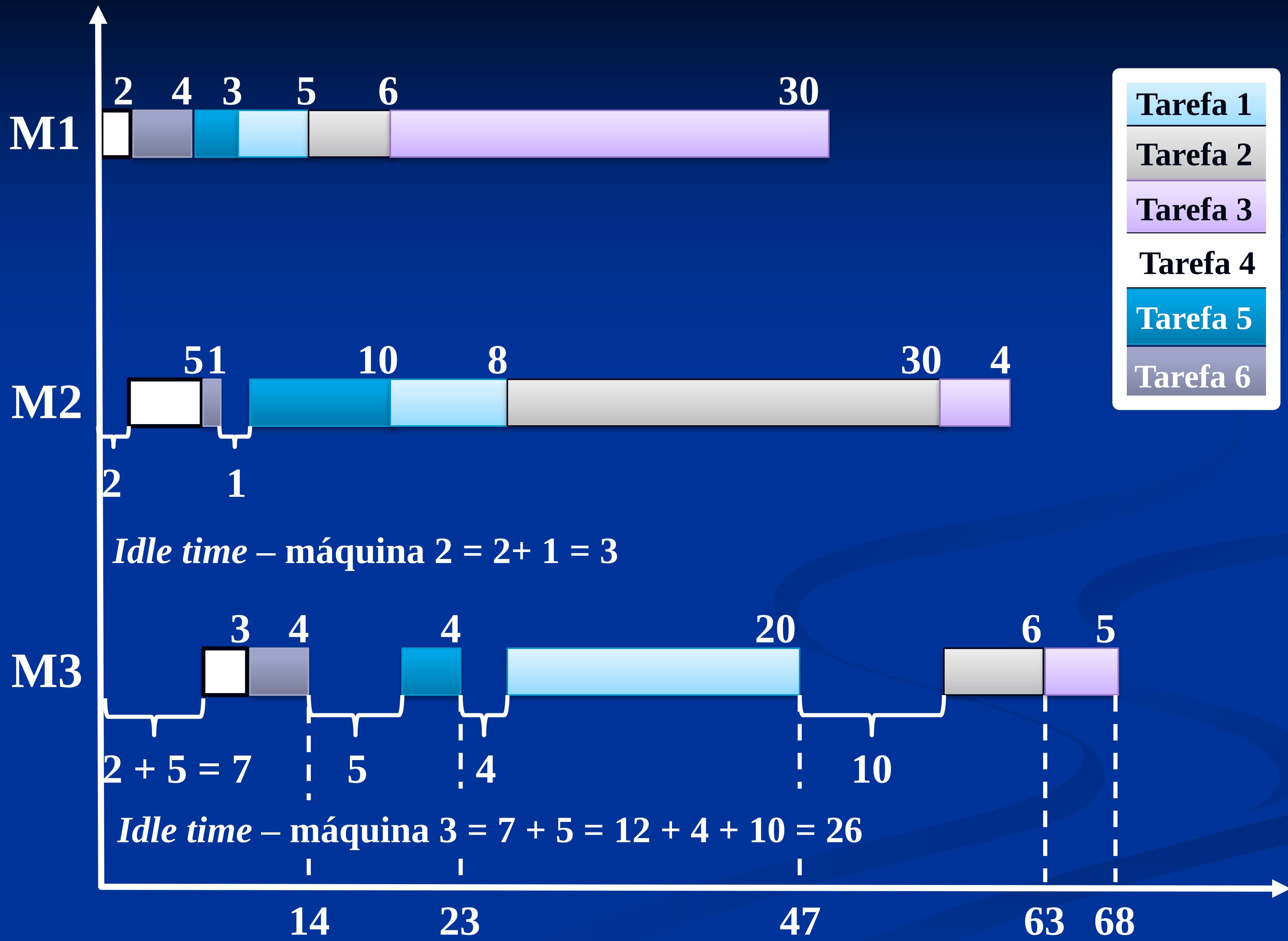
Par	σ	a	C(σa, m)	
			m = 2	m = 3
(4,6,5,1,2)	4651	2	3	26
(4,6,5,1,3)		3	20	17

***Idle time* – máquina 2 = 2+ 1 = 3 + 17 = 20**

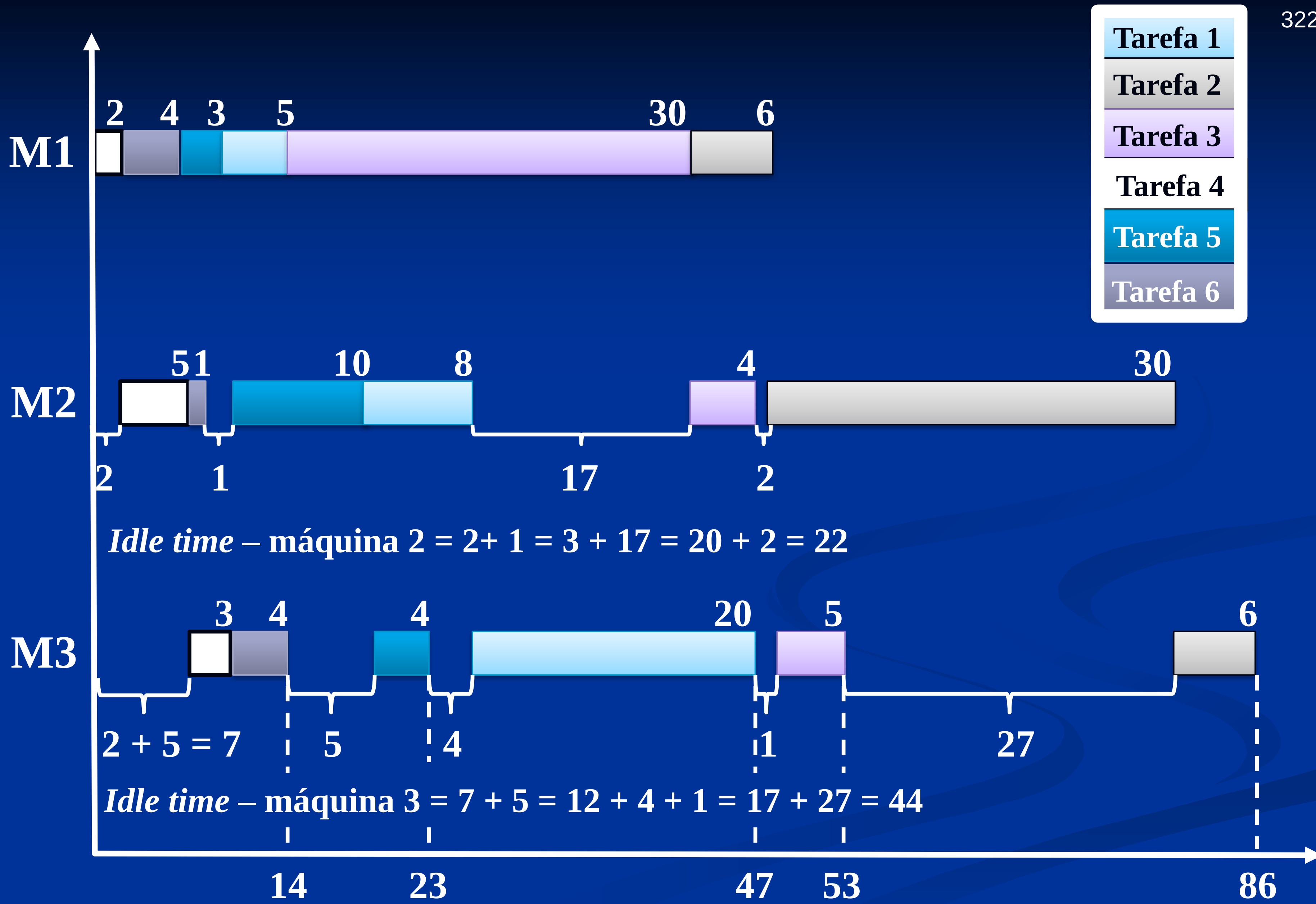
***Idle time* – máquina 3 = 7 + 5 = 12 + 4 + 1 = 17**

Duas opções finais

Últimas tarefas: 2 ou a 3?



Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
2 4 3 5 6 30
M2
5 1 10 8 30 4
2 1
Idle time – máquina 2 = 2+ 1 = 3
M3
3 4 4 20 6 5
2 + 5 = 7
5
4
10
Idle time – máquina 3 = 7 + 5 = 12 + 4 + 10 = 26
14 23 47 63 68


Tarefa 1
Tarefa 2
Tarefa 3
Tarefa 4
Tarefa 5
Tarefa 6
M1
2 4 3 5 30 6
M2
51 10 8 4 30
2 1 17 2
Idle time – máquina 2 = 2+ 1 = 3 + 17 = 20 + 2 = 22
M3
3 4 4 20 5 6
2 + 5 = 7 5 4 1 27
Idle time – máquina 3 = 7 + 5 = 12 + 4 + 1 = 17 + 27 = 44
14 23 47 53 86

√