Sistemas de Tempo Real
Ano Lectivo 2001/2002

Escalonamento de processos num sistema computacional multi-processo e uni-processador

1. Objectivo

Notação	***Descrição***
C	Máximo tempo de computação de uma tarefa
T	Periodicidade de uma tarefa (ou o menor intervalo entre duas activações consecutivas de uma tarefa)
D	*Deadline* (meta temporal) relativo da tarefa

Num sistema computacional podem existir *N* tarefas, t_1, t_2, ..., t_N, sendo então cada uma delas caracterizada da seguinte forma:

$$t_i = (C_i, T_i, D_i) \tag{1}$$

Numa perspectiva de tempo-real, as políticas de escalonamento de processos (ou tarefas) mais utilizadas em sistemas computacionais são:

- o *Rate Monotonic* (RM);
- e o *Earliest Deadline First* (EDF).

Pretende-se:

1. Determinar se um conjunto de tarefas é ou não escalonável
2. Simular graficamente o escalonamento das tarefas (admitindo um escalonamento RM ou EDF em ambientes pre-emptivos ou não pre-emptivos)

2. Relativo ao Objectivo 1

Assume-se sempre que todas as tarefas são independentes e que o tempo de *switching* (passagem da execução de uma tarefa para a outra) é nulo.

2.1. Testes Baseados na Utilização do Processador

Existem dois testes simples (baseados na utilização do processador) que podem ser aplicados. Estes testes só são válidos para sistemas pre-emptivos e para conjuntos de tarefas que tenham *deadlines* relativos iguais ao período. Para além disso, e particularmente no que se refere ao RM, o teste é pessimista (é uma condição suficiente mas não necessária).

Assim, para o RM, o teste baseado na utilização é:

$$\sum_{i=1}^{N} \frac{C_i}{T_i} \leq N \times \left(2^{1/N} - 1\right) \tag{2}$$

Para o EDF, o teste baseado na utilização é:

$$\sum_{i=1}^{N} \frac{C_i}{T_i} \leq 1 \tag{3}$$

2.2. Testes Baseados na Resposta Temporal da Tarefa (RM Pre-emptivo)

Outra forma de verificar se um conjunto de tarefas é ou não escalonável é determinar o seu pior tempo de resposta. O tempo de resposta de uma tarefa corresponde ao intervalo de tempo desde que ela é lançada no sistema até acabar a sua execução.

Como é óbvio, o tempo de resposta de uma tarefa é dado por:

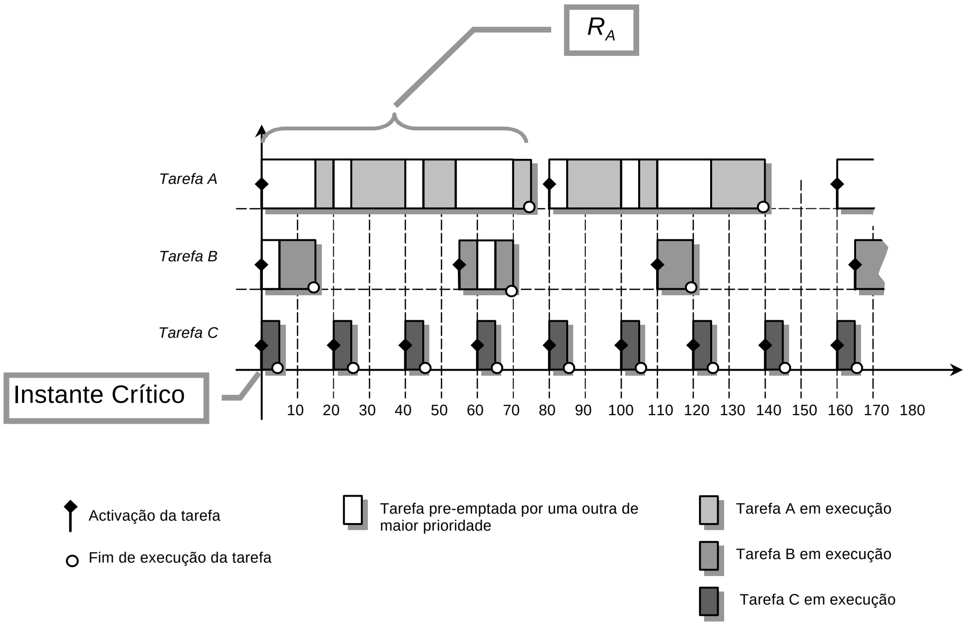
$$R_i = C_i + I_i \tag{4}$$

no qual I_i é a máxima interferência que uma tarefa *i* pode sofrer.

Admita por exemplo o seguinte conjunto de tarefas independentes.

Tarefa	***C***	***T(=D)***
A	35	80
B	10	55
C	5	20

Admitindo que elas são escalonadas de acordo com o RM num ambiente pre-emptivo, e como se sabe que a situação crítica é quando todas são lançadas em simultâneo, R_A = 75 (I_A = 40), R_B = 15 (I_B = 5), R_C = 5 (I_C = 0), como se pode ver graficamente:



2.2.1. RM Pre-Emptivo

No caso de o ambiente ser pre-emptivo, a tarefa pode ser interrompida por uma outra de maior prioridade (interferência) até ao fim da sua execução. Ora, até ao fim da sua execução significa até ao valor de R_i, que é precisamente a incógnita.

No caso do exemplo de conjunto de tarefas dado, e para a tarefa A, esta tarefa até acabar a sua execução (que acontecerá no instante R_A) poderá sofrer interferência das tarefas B e C, que de acordo com RM tem maior prioridade. O número de vezes que a tarefa B aparece no intervalo de tempo que vai desde 0 até R_A (a nossa incógnita) é dado pela função *ceiling* (arredondamento para cima):

$$\left\lceil \frac{R_A}{T_B} \right\rceil$$

A partir do gráfico da página anterior, nós "sabemos" que R_A = 75, e que por isso o número máximo de vezes que a tarefa B pode interferir com a tarefa A é 2 vezes ($\lceil 75 \ / \ 55 \rceil = 2$). De cada vez que a tarefa B é executada antes da tarefa A, a interferência causada vai ser de C_B = 10, o que considerando as duas vezes vai dar 20.

Ainda a partir do gráfico da página anterior, "sabendo" nós que R_A = 75, o número máximo de vezes que a tarefa C pode interferir com a tarefa A é 4 vezes ($\lceil 75 \ / \ 20 \rceil = 4$). De cada vez que a tarefa C é executada antes da tarefa A, a interferência causada vai ser de C_C = 5, o que considerando as quatro vezes vai dar 20.

Então, como determinar R_i de uma forma automática?

Como a interferência que uma tarefa de maior prioridade (*j*) pode causar numa de menor prioridade (*i*) é dada por:

$$\left\lceil \frac{R_i}{Tj} \right\rceil \times C_j$$

A interferência total corresponde ao somatório das interferências causadas por todas as tarefas *j* com prioridade maior ou igual à da tarefa *i* ($j \in$ hp (*i*)):

$$I_i = \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{R_i}{Tj} \right\rceil \times C_j \right) \tag{5}$$

Então, substituindo I_i na equação (4) temos que, para o RM, caso pre-emptivo e tarefas independentes, o tempo de resposta de uma tarefa *i* é dado por:

$$\boxed{R_i = C_i + \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{R_i}{Tj} \right\rceil \times C_j \right)} \tag{6}$$

Ora esta equação é pouco "jeitosa", já que R_i (que é o que se pretende determinar) aparece dos dois lados da equação, e não é possível reformular a equação por forma a ter R_i só do lado esquerdo (a função de *ceiling* não é uma função linear ...).

Estamos por isso perante uma equação recorrente (incógnita está dos dois lados da equação). Como é que se resolve uma equação recorrente? Por recorrência:

$$\boxed{R_i^{n+1} = C_i + \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{R_i^n}{Tj} \right\rceil \times C_j \right)} \tag{7}$$

Começam-se as iterações com:

$$\boxed{R_i^0 = C_i} \tag{8}$$

Quando

$$R_i^{n+1} = R_i^n$$

está encontrada a solução da equação (6).

Por razões óbvias para que o número de iterações seja finito (o algoritmo converge) é necessário que a utilização do processador seja inferior a 1 (porquê?):

$$\sum_{i=1}^{N} \frac{C_i}{T_i} \le 1$$

Para que se possa dizer que o conjunto de tarefas é escalonável, é necessário que se verifique:

$$\boxed{R_i \le D_i \qquad , \forall_i} \tag{9}$$

Por isso, e ao contrário do teste baseado na utilização (equação (2)), o teste baseado na resposta temporal permite que os *deadlines* relativos das tarefas sejam inferiores aos períodos.

Vamos ver para o que se passa com o R_A do exemplo (é só para mostrar o óbvio das equações (6) e (7)):

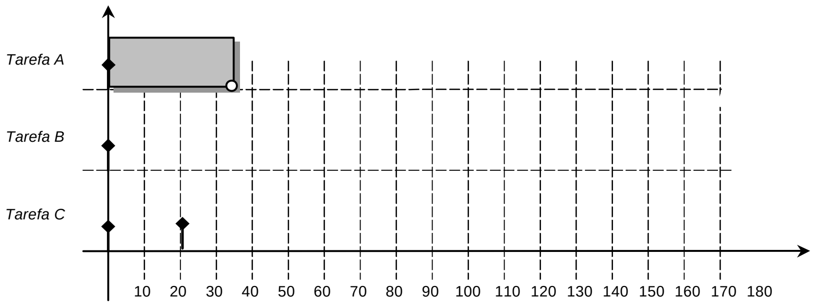
Tarefa	***C***	***T(=D)***
A	35	80
B	10	55
C	5	20

No mínimo a tarefa A vai ter um R_A = 35 (que é o seu tempo máximo de execução).

Isto corresponde, em relação algoritmo representado pela equação (7), a dizer que

$$R_A^0 = C_A = 35$$

Ora, se considerarmos R_A = 35, durante essas 35 unidades de tempo, a tarefa A vai sofrer interferências de B (1 vez no máximo) e de C (2 vezes no máximo):

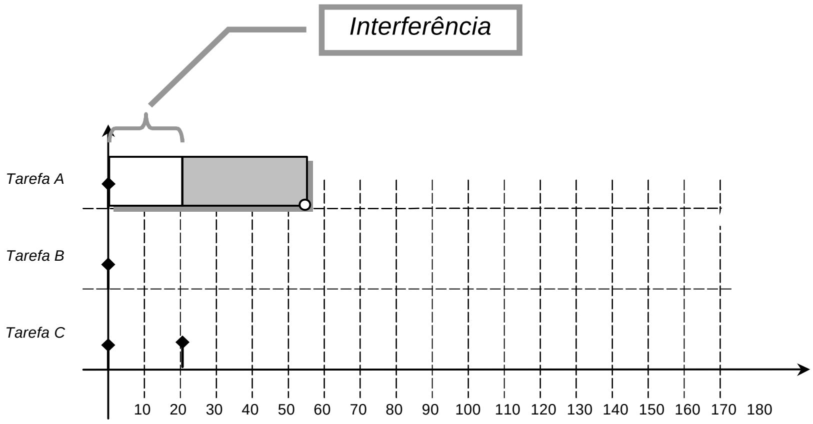


Isto quer dizer que R_A não poderá ser 35 mas sim 35 mais 1 $\times$ 10 (resultado da interferência de B) mais 2 $\times$ 5 (resultado da interferência de C), isto é, 55.

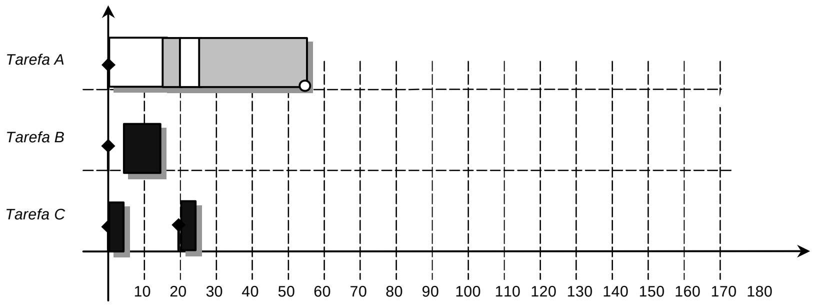
Este é o valor que é dado na segunda iteração relativo à aplicação da equação (7):

$$R_A^1 = C_A + \sum_{j \in \mathrm{hp}(A)} \left(\left\lceil \frac{R_A^0}{Tj} \right\rceil \times C_j \right) = 35 + \left\lceil \frac{35}{55} \right\rceil \times 10 + \left\lceil \frac{35}{20} \right\rceil \times 5 = 35 + 1 \times 10 + 2 \times 5 = 55$$

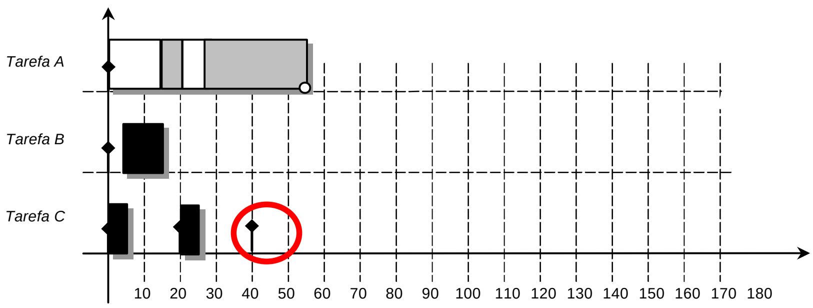
Isto quer dizer que no mínimo R_A será 55.



Na figura anterior, a interferência aparece toda seguida por conveniência, o que se passa na realidade é (do ponto de vista do algoritmo é irrelevante):



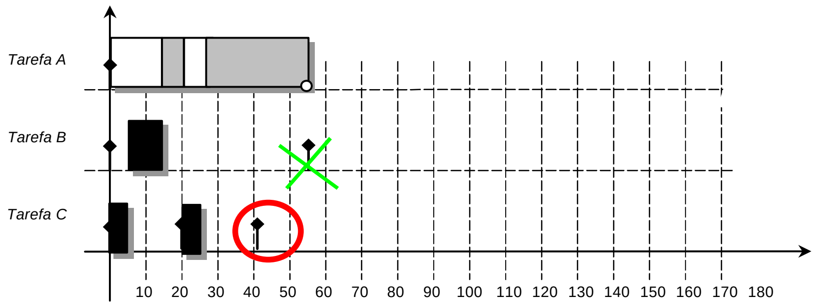
Mas se R_A for 55, será que a interferência não será maior? Obviamente que é, já que aparece outra activação da tarefa C no instante 40, que dada a sua prioridade vai ter de executar antes:



De facto,

$$R_A^2 = C_A + \sum_{j \in \mathrm{hp}(A)} \left(\left\lceil \frac{R_A^1}{Tj} \right\rceil \times C_j \right) = 35 + \left\lceil \frac{55}{55} \right\rceil \times 10 + \left\lceil \frac{55}{20} \right\rceil \times 5 = 35 + 1 \times 10 + 3 \times 5 = 60$$

Note-se que a activação de B no instante 55 não interferiria (no limite) com a execução de A, por que A acabaria a sua execução no instante 55:

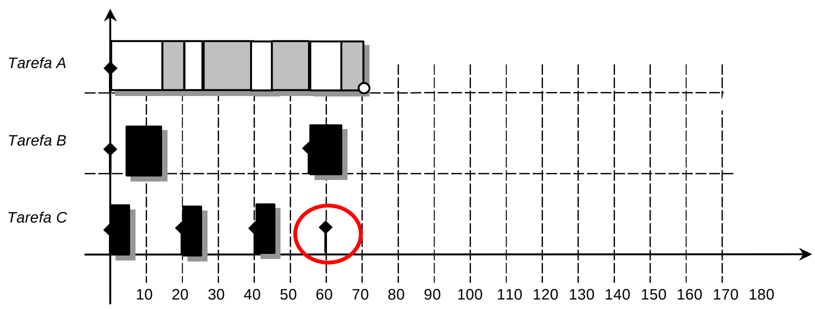


Mas ao considerar a terceira interferência de C, sendo R_A = 60, tem de ser considerada uma nova interferência, já que a tarefa B é activada outra vez em 55 (antes de 60):

De facto,

$$R_A^3 = C_A + \sum_{j \in \mathrm{hp}(A)} \left(\left\lceil \frac{R_A^2}{Tj} \right\rceil \times C_j \right) = 35 + \left\lceil \frac{60}{55} \right\rceil \times 10 + \left\lceil \frac{55}{20} \right\rceil \times 5 = 35 + 2 \times 10 + 3 \times 5 = 70$$

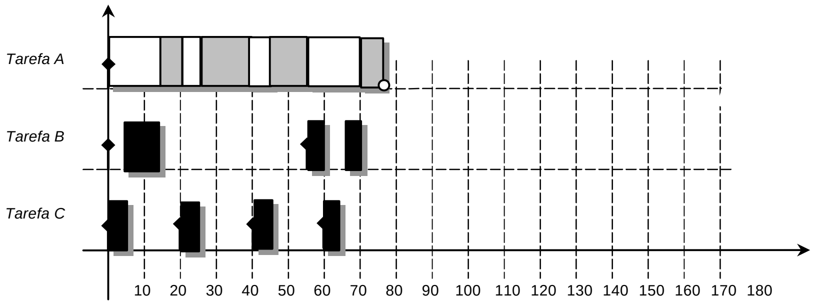
Mas se na realidade o valor hipotético de R_A é 70, é preciso entrar com uma outra interferência da tarefa C:



De facto,

$$R_A^4 = C_A + \sum_{j \in \mathrm{hp}(A)} \left(\left\lceil \frac{R_A^3}{Tj} \right\rceil \times C_j \right) = 35 + \left\lceil \frac{70}{55} \right\rceil \times 10 + \left\lceil \frac{70}{20} \right\rceil \times 5 = 35 + 2 \times 10 + 4 \times 5 = 75$$

o que quer dizer que a realidade é:



A novidade é que para o valor R_A é 75, não existe mais nenhuma interferência adicional:

$$R_A^5 = C_A + \sum_{j \in \mathrm{hp}(A)} \left(\left\lceil \frac{R_A^4}{Tj} \right\rceil \times C_j \right) = 35 + \left\lceil \frac{75}{55} \right\rceil \times 10 + \left\lceil \frac{75}{20} \right\rceil \times 5 = 35 + 2 \times 10 + 4 \times 5 = 75$$

E por isso R_A = 75 corresponde à solução:

$$R_A^5 = R_A^4 = 75 \quad \Rightarrow \quad R_A = 75$$

2.2.2. RM **Não** Pre-Emptivo

No caso de o ambiente ser não pre-emptivo, nenhuma tarefa que comece a sua execução pode ser interrompida.

Assim, e se no caso pre-emptivo determinar o valor de R_i corresponde a determinar o instante de tempo (a partir de 0, onde todas as tarefas são activadas - instante crítico)

- no qual o processador não tem nenhuma tarefa pendente com maior prioridade do que a da tarefa *i* ;
- já acabou a execução completa da tarefa *i*

no caso não pre-emptivo, determinar o valor de R_i corresponde a determinar o instante de tempo (a partir de 0, onde todas as tarefas são activadas - instante crítico)

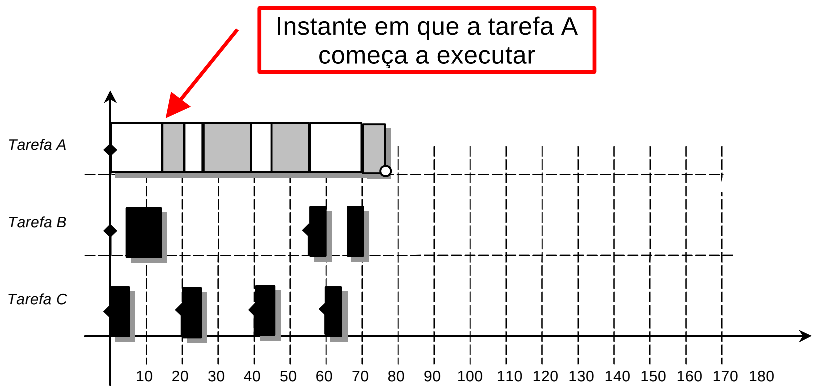
- no qual o processador não tem nenhuma tarefa pendente com maior prioridade do que a da tarefa *i* ;

Nesse instante de tempo a tarefa i começará a executar, até ao fim...

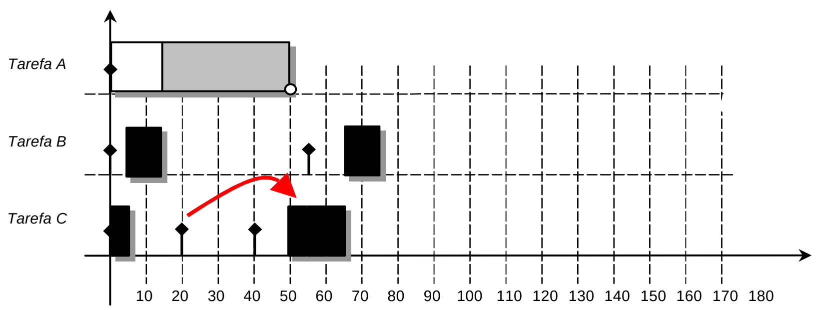
Assim, e relativamente ao exemplo da tarefa A:

Tarefa	***C***	***T(=D)***
A	35	80
B	10	55
C	5	20

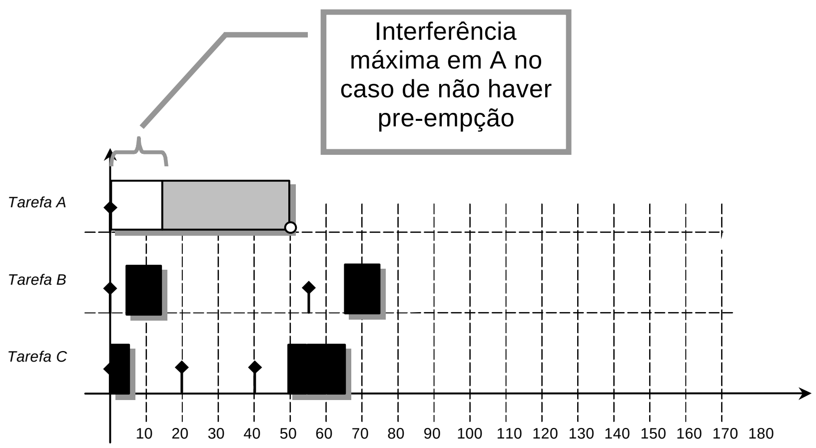
Para o qual se volta a apresentar o cenário num contexto pre-emptivo:



o que se passaria é (note-se que a instância da tarefa A que é activada em 20 perderá o seu *deadline* se o seu *deadline* relativo for igual ao período = 20):



Portanto, e em relação ao caso pre-emptivo, a interferência não depende do tempo de execução da própria tarefa, e por isso não depende do tempo de resposta (R_i):



Assim, e se no caso pre-emptivo, o número de vezes que uma tarefa de maior prioridade *j* pode interferir na execução de uma tarefa *i* é dado por:

$$\left\lceil \frac{R_i}{T_j} \right\rceil$$

no caso não pre-emptivo a definição de interferência é o intervalo de tempo desde o instante crítico (todas as tarefas activadas) até ao instante em que o processador não tem tarefas de maior prioridade ou prioridade igual para processar:

$$I_i = \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{I_i}{Tj} \right\rceil \times C_j \right) \tag{10}$$

Esta equação é igual (a menos de um pormenor) à equação (5).

A diferença, é que, ao contrário do caso pre-emptivo, primeiro determina-se o I_i. O valor obtido para I_i é depois somado C_i a para determinar o R_i (ver equação (4)).

Exemplo para determinar a interferência I_A.

A equação (10) é uma equação recorrente. Por isso resolve-se da seguinte forma:

$$I_i^{n+1} = \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{I_i^n}{Tj} \right\rceil \times C_j \right) \tag{11}$$

O valor inicial ($I_i^{\,0}$) não pode ser 0 (porquê?), mas logo o somatório dos valores de C das tarefas que têm mais prioridade:

$$I_i^0 = \sum_{j \in \mathrm{hp}(i)} C_j \tag{12}$$

Assim,

$$I_A^0 = C_B + C_C = 10 + 5 = 15$$

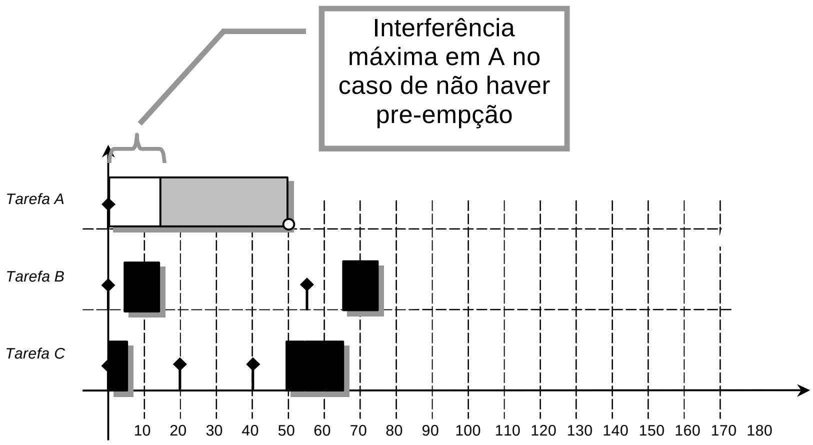
Então,

$$I_A^1 = \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{I_A^0}{Tj} \right\rceil \times C_j \right) = \left\lceil \frac{15}{55} \right\rceil \times 10 + \left\lceil \frac{15}{20} \right\rceil \times 5 = 15$$

Como

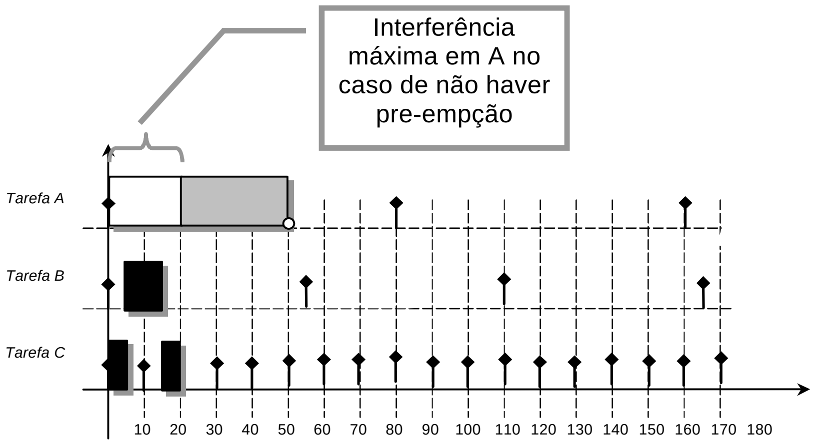
$$I_A^1 = I_A^0 = 15 \qquad \Rightarrow \qquad I_A = 15$$

Que é um valor que já tínhamos visto graficamente:



Se alterarmos ligeiramente o exemplo de conjunto de tarefas:

Tarefa	***C***	***T(=D)***
A	35	80
B	10	55
C	5	**10**



Determinar I_A recorrendo à equação (11), será:

$$I_A^0 = C_B + C_C = 10 + 5 = 15$$

$$I_A^1 = \left\lceil \frac{15}{55} \right\rceil \times 10 + \left\lceil \frac{15}{10} \right\rceil \times 5 = 1 \times 10 + 2 \times 5 = 20$$

$$I_A^2 = \left\lceil \frac{20}{55} \right\rceil \times 10 + \left\lceil \frac{20}{10} \right\rceil \times 5 = 1 \times 10 + 2 \times 5 = 20$$

Ainda para o exemplo correspondente ao seguinte conjunto de tarefas:

Tarefa	*C*	*T(=D)*
A	35	80
B	10	55
C	5	20

Determinar o I_C seria:

$$I_C^0 = 0 \quad \Rightarrow \quad I_C = 0$$

já que não existe nenhuma tarefa j de maior prioridade, e I_B seria:

$$I_B^0 = C_C = 5$$
$$I_B^1 = \left\lceil \frac{5}{20} \right\rceil \times 5 = 5$$

O que quer dizer que

$$I_B^1 = I_B^0 = 5 \quad \Rightarrow \quad I_B = 5$$

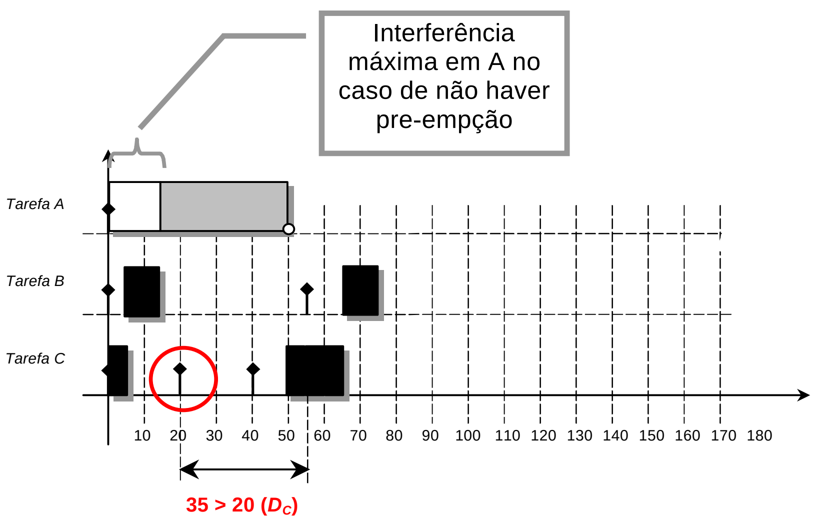
Então, e aplicando a equação (4), os piores tempos de resposta para as 3 tarefas seriam:

$$R_A = C_A + I_A = 35 + 15 = 50$$
$$R_B = C_B + I_B = 10 + 5 = 15$$
$$R_C = C_C + I_C = 5 + 0 = 5$$

O que quer dizer que o conjunto de tarefas seria escalonável (nenhuma perdia o *deadline*), já que a determinação de R_i corresponde a considerar a pior situação:

$$R_A = 50 \quad \leq \quad D_A = 80$$
$$R_B = 15 \quad \leq \quad D_B = 55$$
$$R_C = 5 \quad \leq \quad D_C = 20$$

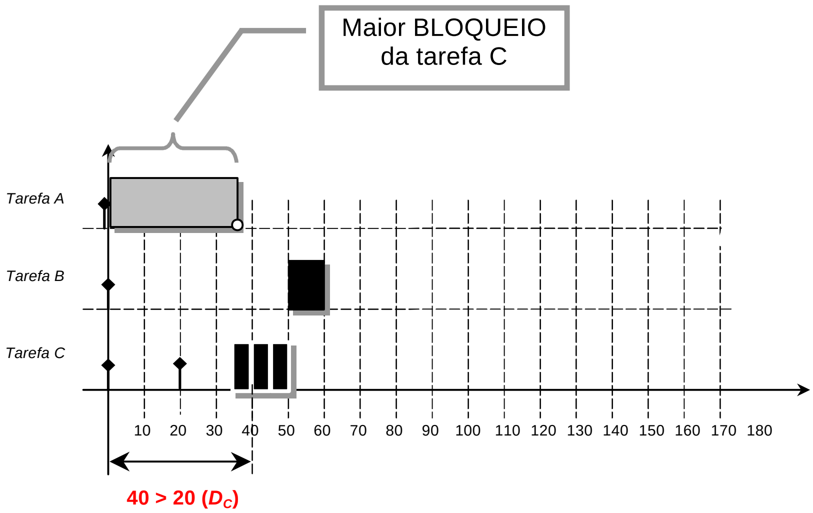
Algo está mal, já que quando se esteve a analisar o como determinar a interferência na tarefa A, se via que pelo menos a tarefa C apresentava situações em perdia o *deadline*:



O que se passa, é que pelo facto de não haver pre-empção, uma tarefa de menor prioridade pode atrasar a execução de uma de maior prioridade: **situação de bloqueio**.

Qual é o maior bloqueio que a tarefa C pode sofrer?

O maior bloqueio é considerar que a tarefa A é activada no sistema marginalmente antes da tarefa B e C:



A equação (10) tem de ser actualizada para incluir o bloqueio:

$$I_i = B_i + \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{I_i}{Tj} \right\rceil \times C_j \right) \quad (13)$$

No caso da tarefa A, B_A é 0, já que não existe nenhuma tarefa de menor prioridade.

Quer para o caso das tarefas B e C o bloqueio corresponde a 35.

Sendo assim, o valor da interferência para a tarefa C é:

$$I_C = B_C + 0 = 35 + 0 = 35$$

O valor da interferência para a tarefa B é:

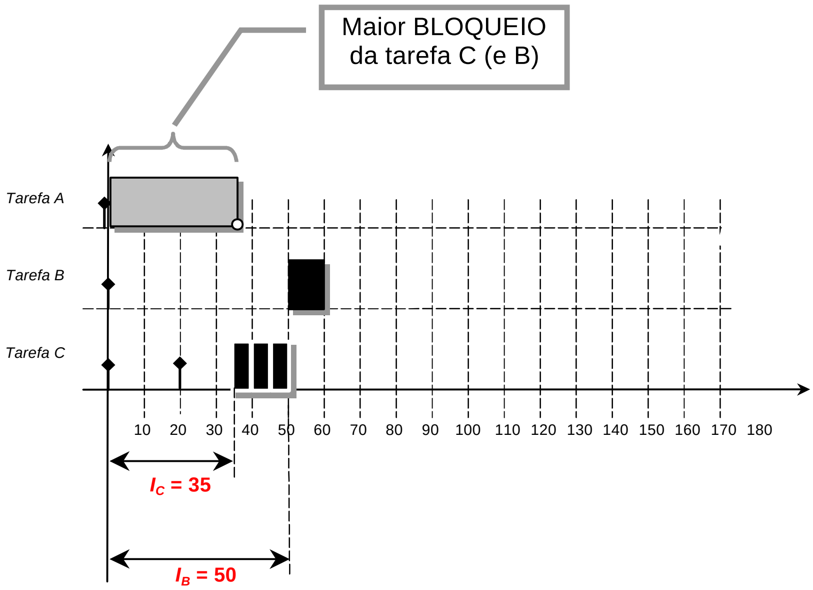
$$I_B^0 = B_B + C_C = 35 + 5 = 40$$

$$I_B^1 = 35 + \left\lceil \frac{40}{20} \right\rceil \times 5 = 45$$

$$I_B^2 = 35 + \left\lceil \frac{45}{20} \right\rceil \times 5 = 50$$

$$I_B^3 = 35 + \left\lceil \frac{50}{20} \right\rceil \times 5 = 50$$

Aliás como já aparecia graficamente ilustrado:



Então, e aplicando a equação (4), os piores tempos de resposta para as 3 tarefas seriam:

$$R_A = C_A + I_A = 35 + 15 = 50$$
$$R_B = C_B + I_B = 10 + 50 = 60$$
$$R_C = C_C + I_C = 5 + 35 = 40$$

O que quer dizer que o conjunto de tarefas não seria escalonável, já que há tarefas que podem perder os *deadlines* (tarefas B e C):

$$R_A = 50 \quad \leq \quad D_A = 80$$
$$R_B = 15 \quad \leq \quad D_B = 55$$
$$R_C = 5 \quad \leq \quad D_C = 20$$

Note-se que se alterarmos as características do conjunto de tarefas:

Tarefa	**C**	***T(=D)***
A	**9**	80
B	10	55
C	5	20

Os valores dos bloqueios para cada tarefa serão:

$$B_A = 0$$
$$B_B = 9$$
$$B_C = 10$$

A definição que foi dada para o caso pre-emptivo relativamente a "instante crítico" tem de ser ligeiramente alterada:

A pior situação ocorre quando a tarefa é activada em simultâneo com todas as que têm prioridade maior ou igual, e a maior (em termos de C), das que têm menor prioridade é activada marginalmente antes.

Assim, para determinar a interferência utilizam a seguinte formula recorrente:

$$I_i^{n+1} = B_i + \sum_{j \in \mathrm{hp}(i)} \left(\left\lceil \frac{I_i^n}{Tj} \right\rceil \times C_j \right) \qquad (14)$$

na B_i é dado por:

$$B_i = \begin{cases} 0 & \text{, caso } i \text{ seja a tarefa de menor prioridade} \\ \max\{C_j\} & \text{, em que } j \text{ são tarefas de menor prioridade do que a de } i \end{cases} \qquad (15)$$

o valor inicial para I_i deverá ser (só é importante para a determinação da interferência na tarefa de menor prioridade - para a qual $B_i = 0$)

$$I_i^0 = B_i + \sum_{j \in \mathrm{hp}(i)} C_j \qquad (16)$$

2.2.3. EDF Pre-Emptivo

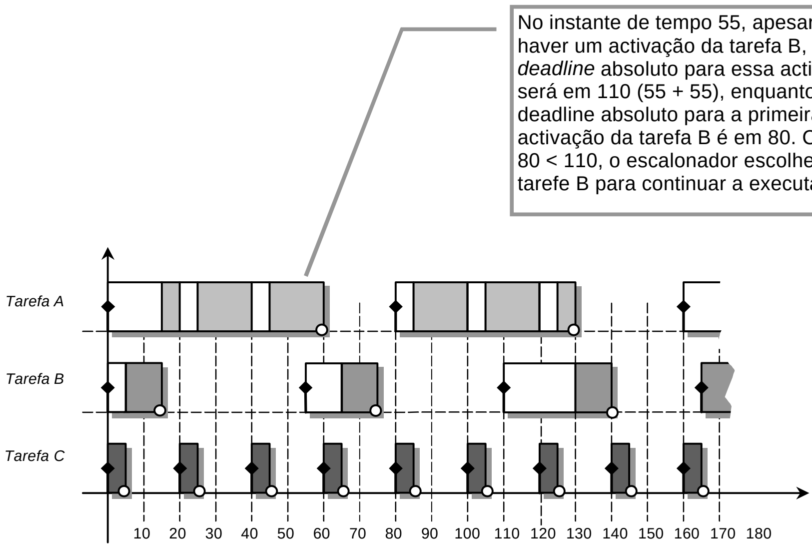
Ao contrário do que acontece com o RM, no qual as tarefas vão mantendo sempre o mesmo nível de prioridade (prioridades estáticas), no EDF a prioridade relativa de uma tarefa vai variando ao longo do tempo (prioridades dinâmicas).

Admita o mesmo conjunto de tarefas independentes:

Tarefa	***C***	***T (=D)***
A	35	80
B	10	55
C	5	20

No caso do EDF, a tarefa que é escolhida para executar é a que tem o seu *deadline* (absoluto) mais cedo.

Assim, e admitindo que o instante crítico corresponde à situação em que todas as tarefas são activadas simultaneamente, o escalonamento para o conjunto de tarefas apresentado será:



Como é que se determina o valor de R_A (genericamente R_i) para o EDF pre-emptivo?

A filosofia de base é igual à utilizada no caso RM:

$$R_i = C_i + I_i \tag{17}$$

As pequenas diferenças ocorrem para o calculo da interferência.

Vamos a seguir analisar a interferência que existe na primeira activação da tarefa A (que pela figura nós sabemos que é 25).

Vamos repescar a equação apresentada para o caso do RM pre-emptivo:

$$R_i = C_i + \sum_{j=?} \left(\left\lceil \frac{R_i}{Tj} \right\rceil \times C_j \right) \tag{18}$$

O que difere é a noção de quais são as tarefas *j* (*j* ≠ *i*) que vão tendo maior prioridade do que a da tarefa *i*.

Por exemplo, em relação ao cenário apresentado, e se a análise é em relação à activação no instante 0 da tarefa A, o *deadline* absoluto dessa activação vai ocorrer em 80 (uma vez que o *deadline* relativo D_A = 80). A tarefa B tem um *deadline* relativo D_B = 55 e a tarefa C tem um *deadline* relativo D_C = 20. Obviamente, quer B quer C vão contribuir para a interferência em A pelo menos uma vez. Isto é, as primeiras activações (no instante 0) das tarefas B e C vão ter *deadlines* absolutos respectivamente em 55 e 20 (o que é antes do *deadline* absoluto da primeira activação da tarefa A).

Assim, e relativamente à equação (18), a definição das tarefas j de maior prioridade poderá ser:

$$R_i = C_i + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\left\lceil \frac{R_i}{Tj} \right\rceil \times C_j \right) \tag{19}$$

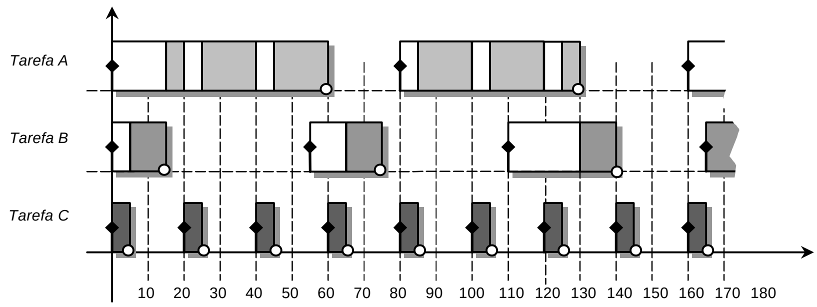
Ora mas isto seria considerar uma política de escalonamento que é conhecida como *Deadline Monotonic* (DM), no qual as prioridades das tarefas são estáticas em função do seu *deadline* relativo (e não do período). Note-se que quando os *deadlines* relativos das tarefas são iguais aos seus períodos, o DM é exactamente igual ao RM.

De facto, e para o caso do RM pre-emptivo, a equação (6) pode ser apresentada de uma forma equivalente:

$$R_i = C_i + \sum_{\substack{j \neq i \\ T_j \leq T_i}} \left(\left\lceil \frac{R_i}{Tj} \right\rceil \times C_j \right) \tag{20}$$

Para o caso do EDF pre-emptivo, não basta considerar que, durante R_i, todas as activações de tarefas *j* (em que $j \neq i$ e $D_j \leq D_i$) podem interferir na execução da tarefa *i* que é activada no instante 0.

Olhando para o gráfico de execução EDF



é fácil concluir que a partir da segunda (inclusive) activação da tarefa B e da quinta activação da tarefa C, essas tarefas já não vão interferir na primeira activação da tarefa A.

Isto porque a segunda activação de B vai ter um *deadline* absoluto em 110, a terceira em 165, e a quinta activação de C vai ter um *deadline* absoluto em 100 a sexta activação de C em 120, etc. Isto é, em instantes posteriores ao *deadline* absoluto da primeira activação de A (que ocorre em 80).

Então, qual é o máximo número de activações de uma tarefa *j* (em que $j \neq i$ e $D_j \leq D_i$) que pode interferir na primeira activação de uma tarefa *i*?

Esse valor é dado por ($\lfloor \ \rfloor$ corresponde a função *floor* - arredondar para baixo):

$$1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \tag{21}$$

Voltemos ao exemplo:

Tarefa	*C*	*T (=D)*
A	35	80
B	10	55
C	5	20

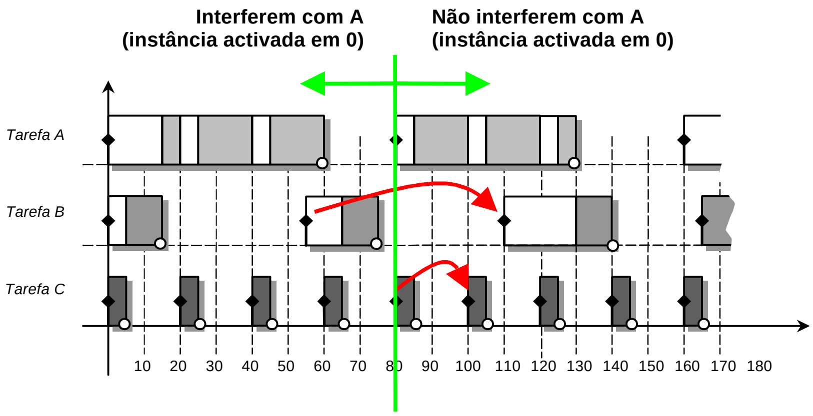
Qual o número máximo de vezes que B pode de facto interferir com a activação de A que tem o *deadline* absoluto em 80?

$$1 + \left\lfloor \frac{D_A - D_B}{T_B} \right\rfloor = 1 + \left\lfloor \frac{80 - 55}{55} \right\rfloor = 1 + 0 = 1$$

Qual o número máximo de vezes que C pode de facto interferir com a activação de A que tem o *deadline* absoluto em 80?

$$1 + \left\lfloor \frac{D_A - D_C}{T_C} \right\rfloor = 1 + \left\lfloor \frac{80 - 20}{20} \right\rfloor = 1 + 3 = 4$$

Estes valores são óbvios olhando para o gráfico:



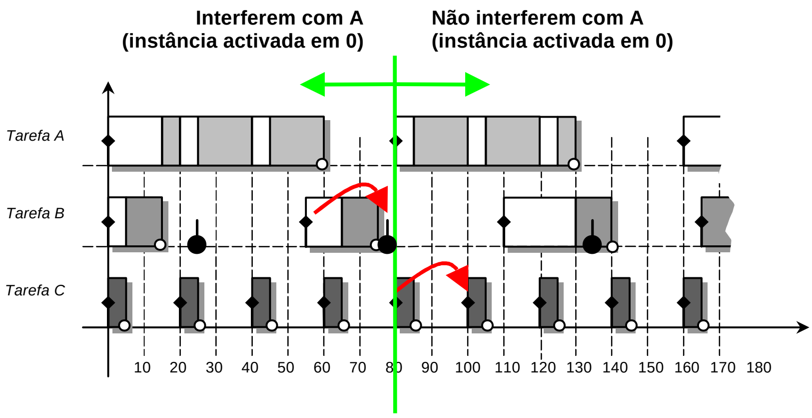
Alteremos ligeiramente o exemplo do conjunto de tarefas:

Tarefa	***C***	***T***	***D***
A	35	80	80
B	10	55	**24**
C	5	20	20

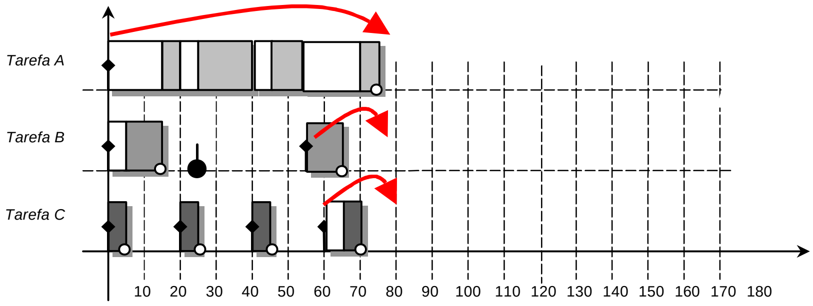
Para este caso, qual o número máximo de vezes que B pode de facto interferir com a activação de A que tem o *deadline* absoluto em 80?

$$1+\left\lfloor \frac{D_A - D_B}{T_B} \right\rfloor = 1+\left\lfloor \frac{80-24}{55} \right\rfloor = 1+\left\lfloor \frac{56}{55} \right\rfloor = 1+1=2$$

De facto, a activação de B em 55 vai ter um *deadline* absoluto em 79 (55 + D_B = 55 + 24 = 79).



O que quer dizer que o escalonamento correcto (considerando D_B = 24) é:



Então, o que é que tem de ser alterado na equação (19)?

A partir de determinada altura de "pesquisa" de R_i, o número de activações que interfere é limitado ao máximo que é dado por (21), o que quer dizer:

$$R_i = C_i + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_i}{Tj} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) \qquad (22)$$

Para resolver esta equação recorrente:

$$R_i^{n+1} = C_i + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_i^n}{Tj} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) \qquad (23)$$

Em cada passo, e para cada tarefa j que possa interferir com a tarefa i, é escolhido o menor de entre os dois valores:

$$\left(\min\{a, b\} \times C_j\right), \quad \text{no qual } a = \left\lceil \frac{R_i^n}{Tj} \right\rceil \text{ e } b = 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor$$

α corresponde ao número de activações de j até ao instante R_i (tal qual como no RM). β corresponde ao número máximo de activações de *j* que podem de facto eventualmente interferir com a activação (no instante 0) da tarefa *i*.

Voltemos ao exemplo:

Tarefa	*C*	*T (=D)*
A	35	80
B	10	55
C	5	20

e determinemos R_A.

$$R_A^0 = C_A = 35$$

$$R_A^1 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^0}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) =$$

$$= 35 + \left(\min\left\{ \left\lceil \frac{R_A^0}{T_B} \right\rceil, 1 + \left\lfloor \frac{D_A - D_B}{T_B} \right\rfloor \right\} \times C_B \right) + \left(\min\left\{ \left\lceil \frac{R_A^0}{T_C} \right\rceil, 1 + \left\lfloor \frac{D_A - D_C}{T_C} \right\rfloor \right\} \times C_C \right) =$$

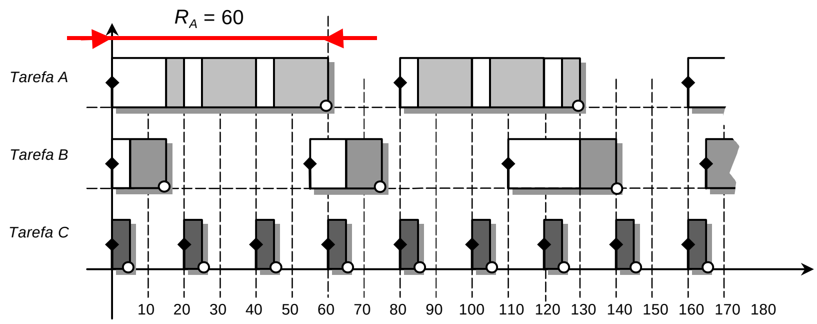
$$= 35 + \left(\min\left\{ \left\lceil \frac{35}{55} \right\rceil, 1 + \left\lfloor \frac{80-55}{55} \right\rfloor \right\} \times 10 \right) + \left(\min\left\{ \left\lceil \frac{35}{20} \right\rceil, 1 + \left\lfloor \frac{80-20}{20} \right\rfloor \right\} \times 5 \right) =$$

$$= 35 + (\min\{1,1\} \times 10) + (\min\{2,4\} \times 5) = 35 + 10 + 2 \times 5 = 55$$

$$R_A^2 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^1}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) = 35 + (\min\{1,1\} \times 10) + (\min\{3,4\} \times 5) = 60$$

$$R_A^3 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^2}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) = 35 + (\min\{2,1\} \times 10) + (\min\{3,4\} \times 5) = 60$$

Portanto a solução é 60, como é comprovado pelo gráfico já apresentado (note-se para determinar R^3_A que com R^2_A = 60, passa a haver duas activações de B, mas só uma é que pode interferir (min{2,**1**} = 1). No caso da interferência causada pela tarefa C, o "limitador" não chegou a condicionar o resultado (min{3,**4**} = 3).



Vejamos agora o mesmo cálculo, mas considerando as seguintes alterações no conjunto de tarefas:

Tarefa	*C*	*T*	*D*
A	35	80	80
B	10	55	**24**
C	5	20	20

Para determinar R_A:

$$R_A^0 = C_A = 35$$

$$R_A^1 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^0}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) =$$

$$= 35 + \left(\min\left\{ \left\lceil \frac{R_A^0}{T_B} \right\rceil, 1 + \left\lfloor \frac{D_A - D_B}{T_B} \right\rfloor \right\} \times C_B \right) + \left(\min\left\{ \left\lceil \frac{R_A^0}{T_C} \right\rceil, 1 + \left\lfloor \frac{D_A - D_C}{T_C} \right\rfloor \right\} \times C_C \right) =$$

$$= 35 + \left(\min\left\{ \left\lceil \frac{35}{55} \right\rceil, 1 + \left\lfloor \frac{80 - 24}{55} \right\rfloor \right\} \times 10 \right) + \left(\min\left\{ \left\lceil \frac{35}{20} \right\rceil, 1 + \left\lfloor \frac{80 - 20}{20} \right\rfloor \right\} \times 5 \right) =$$

$$= 35 + (\min\{1,2\} \times 10) + (\min\{2,4\} \times 5) = 35 + 10 + 2 \times 5 = 55$$

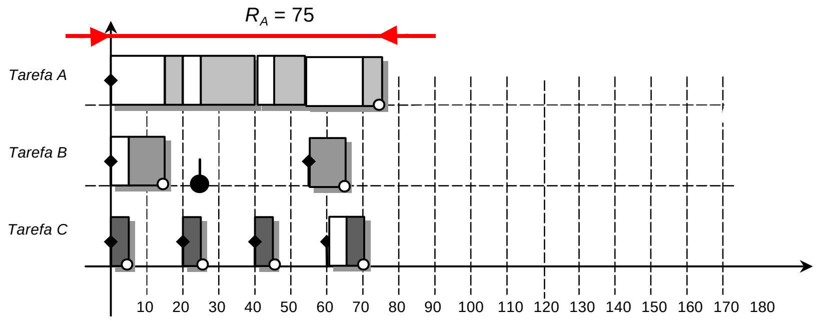
$$R_A^2 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^1}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) = 35 + (\min\{1,2\} \times 10) + (\min\{3,4\} \times 5) = 60$$

$$R_A^3 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^2}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) = 35 + (\min\{2,2\} \times 10) + (\min\{3,4\} \times 5) = 70$$

O facto de o limitador para B ser 2 e não 1, começa a fazer os seus "estragos", e vai fazer com que sejam consideradas 4 activações de C:

$$R_A^4 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^3}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) = 35 + (\min\{2,2\} \times 10) + (\min\{4,4\} \times 5) = 75$$

$$R_A^5 = 35 + \sum_{\substack{j \neq i \\ D_j \leq D_i}} \left(\min\left\{ \left\lceil \frac{R_A^4}{T_j} \right\rceil, 1 + \left\lfloor \frac{D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) = 35 + (\min\{2,2\} \times 10) + (\min\{4,4\} \times 5) = 75$$



Voltemos ao exemplo:

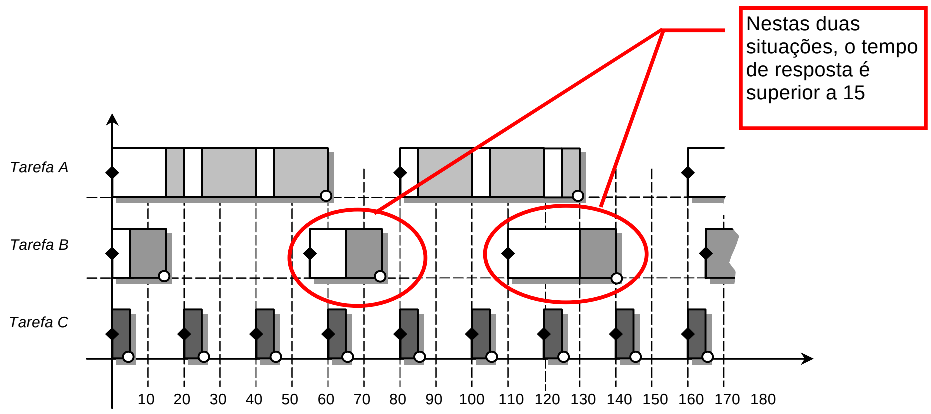
Tarefa	*C*	*T (=D)*
A	35	80
B	10	55
C	5	20

Os valores de R_B e R_C obtidos serão:

$$R_B = 15$$

$$R_C = 5$$

Mas basta voltar a olhar atentamente para o gráfico:



Então, o que é que está mal?

O que acontece, é que ao contrário do RM pre-emptivo, a situação crítica para uma tarefa é quando todas as outras são activadas simultaneamente no instante de tempo 0, e a tarefa *i* é activada no instante ***a***, em que ***a*** $\geq$ 0.

Se quisermos determinar o tempo de resposta da instância da tarefa B que é activada em 55, temos de considerar a seguinte actualização da equação (22):

$$R_A(a=55) = L_A(a=55) - a, \qquad \text{na qual}$$

$$L_A(a=55) = \left(1 + \left\lfloor \frac{55}{55} \right\rfloor\right) \times C_i + \sum_{\substack{j \neq i \\ D_j \leq 55 + D_i}} \left(\min\left\{ \left\lceil \frac{L_A(a=55)}{Tj} \right\rceil, 1 + \left\lfloor \frac{55 + D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) \tag{24}$$

Isto porque o L_A(a=55) vai conduzir ao valor 75. Como a instância de B é activada em 55 (a=55), a resposta temporal é 75 - 55 = 20.

Assim, genericamente,

$$L_i(a) = \left(1 + \left\lfloor \frac{a}{T_i} \right\rfloor\right) \times C_i + \sum_{\substack{j \neq i \\ D_j \leq a + D_i}} \left(\min\left\{ \left\lceil \frac{L_i(a)}{Tj} \right\rceil, 1 + \left\lfloor \frac{a + D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) \tag{25}$$

De facto, ao considerar o "historial" desde 0 até $L_i(a)$, podemos ter de considerar mais do que uma vez a tarefa i. Para a tarefa B e se a = 55, corresponde a considerar 2 vezes C_B. Daí aparecer na equação (25):

$$\left(1 + \left\lfloor \frac{a}{T_i} \right\rfloor\right) \times C_i$$

em vez de simplesmente

$$C_i$$

Por outro lado, a primeira activação da tarefa A vai interferir na segunda activação da tarefa B. Daí aparecer por baixo do somatório $D_j \leq a + D_i$.

No caso, da tarefa B que é activada em 55 (a = 55), $D_A \leq a + D_B$ (80 ≤ 55 + 55).

Obviamente que o número máximo de interferências (desde o instante 0) depende da variável a. Por exemplo, e para a mesma activação de B em 55, o número máximo de vezes que C vai executar antes dessa activação de B é:

$$1 + \left\lfloor \frac{a + D_B - D_C}{T_C} \right\rfloor = 1 + \left\lfloor \frac{55 + 55 - 20}{20} \right\rfloor = 1 + \left\lfloor \frac{90}{20} \right\rfloor = 5$$

Exemplo

Vamos então determinar o $L_B(a=55)$, utilizando a equação (25)

$$L_B(a=55) = \left(1+\left\lfloor \frac{55}{T_B} \right\rfloor\right)\times C_B + \sum_{\substack{j\neq B \\ D_j\leq 55+D_B}} \left(\min\left\{ \left\lceil \frac{L_B(a=55)}{Tj} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_j}{T_j} \right\rfloor \right\} \times C_j \right)$$

Para variar, resolve-se iterativamente a equação recorrente:

$$L_B^0(a=55) = \left(1+\left\lfloor \frac{55}{T_B} \right\rfloor\right)\times C_B = \left(1+\left\lfloor \frac{55}{55} \right\rfloor\right)\times 10 = 2\times 10 = 20$$

$$\begin{aligned} L_B^1(a=55) &= 20+\min\left\{ \left\lceil \frac{L_B^0(a=55)}{T_A} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_A}{T_A} \right\rfloor \right\}\times C_A + \min\left\{ \left\lceil \frac{L_B^0(a=55)}{T_C} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_C}{T_C} \right\rfloor \right\}\times C_C = \\ &= 20+\min\left\{ \left\lceil \frac{20}{80} \right\rceil, 1+\left\lfloor \frac{55+55-80}{80} \right\rfloor \right\}\times 35 + \min\left\{ \left\lceil \frac{20}{20} \right\rceil, 1+\left\lfloor \frac{55+55-20}{20} \right\rfloor \right\}\times 5 = \\ &= 20+\min\{1,1\}\times 35+\min\{1,5\}\times 5 = 20+1\times 35+1\times 5 = 60 \end{aligned}$$

$$\begin{aligned} L_B^2(a=55) &= 20+\min\left\{ \left\lceil \frac{L_B^1(a=55)}{T_A} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_A}{T_A} \right\rfloor \right\}\times C_A + \min\left\{ \left\lceil \frac{L_B^1(a=55)}{T_C} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_C}{T_C} \right\rfloor \right\}\times C_C = \\ &= 20+\min\left\{ \left\lceil \frac{60}{80} \right\rceil, 1+\left\lfloor \frac{55+55-80}{80} \right\rfloor \right\}\times 35 + \min\left\{ \left\lceil \frac{60}{20} \right\rceil, 1+\left\lfloor \frac{55+55-20}{20} \right\rfloor \right\}\times 5 = \\ &= 20+\min\{1,1\}\times 35+\min\{3,5\}\times 5 = 20+1\times 35+3\times 5 = 70 \end{aligned}$$

$$\begin{aligned} L_B^3(a=55) &= 20+\min\left\{ \left\lceil \frac{L_B^2(a=55)}{T_A} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_A}{T_A} \right\rfloor \right\}\times C_A + \min\left\{ \left\lceil \frac{L_B^2(a=55)}{T_C} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_C}{T_C} \right\rfloor \right\}\times C_C = \\ &= 20+\min\left\{ \left\lceil \frac{70}{80} \right\rceil, 1+\left\lfloor \frac{55+55-80}{80} \right\rfloor \right\}\times 35 + \min\left\{ \left\lceil \frac{70}{20} \right\rceil, 1+\left\lfloor \frac{55+55-20}{20} \right\rfloor \right\}\times 5 = \\ &= 20+\min\{1,1\}\times 35+\min\{3,5\}\times 5 = 20+1\times 35+4\times 5 = 75 \end{aligned}$$

$$\begin{aligned} L_B^4(a=55) &= 20+\min\left\{ \left\lceil \frac{L_B^3(a=55)}{T_A} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_A}{T_A} \right\rfloor \right\}\times C_A + \min\left\{ \left\lceil \frac{L_B^3(a=55)}{T_C} \right\rceil, 1+\left\lfloor \frac{55+D_B-D_C}{T_C} \right\rfloor \right\}\times C_C = \\ &= 20+\min\left\{ \left\lceil \frac{75}{80} \right\rceil, 1+\left\lfloor \frac{55+55-80}{80} \right\rfloor \right\}\times 35 + \min\left\{ \left\lceil \frac{75}{20} \right\rceil, 1+\left\lfloor \frac{55+55-20}{20} \right\rfloor \right\}\times 5 = \\ &= 20+\min\{1,1\}\times 35+\min\{3,5\}\times 5 = 20+1\times 35+4\times 5 = 75 \end{aligned}$$

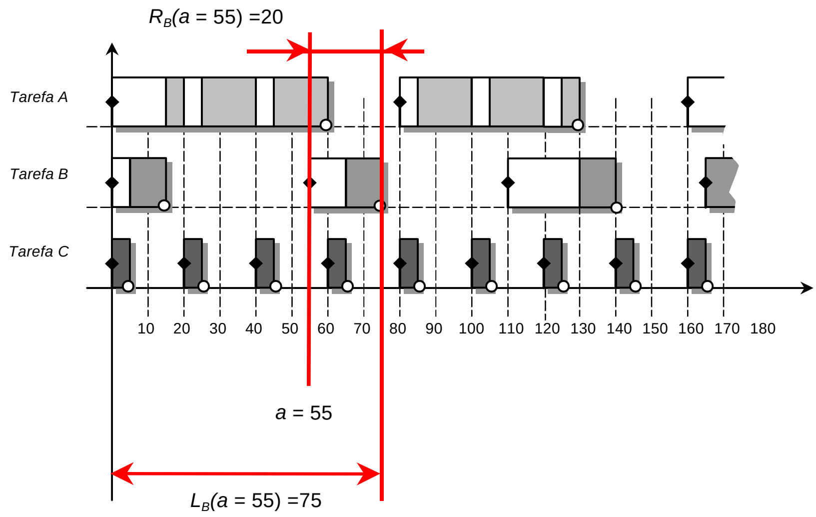
Assim, se

$$L_B(a=55) = 75$$

então

$$R_B(a=55) = L_B(a=55) - a = 75-55 = 20$$

o que coincide com o que já tínhamos visto graficamente:



Como também é possível ver no gráfico, 20 não é o pior caso de tempo de resposta para a tarefa B. Por exemplo a instância que é activada em 110 tem um tempo de resposta pior (30). E se B for lançada em 5, ou 10, ou 15, ou 18, ou, ...

Põe-se então o problema:

Para que valores de *a* se deve testar a equação (25), que por conveniência se reproduz a seguir?

$$L_i(a) = \left(1 + \left\lfloor \frac{a}{T_i} \right\rfloor\right) \times C_i + \sum_{\substack{j \neq i \\ D_j \leq a + D_i}} \left(\min\left\{ \left\lceil \frac{L_i(a)}{Tj} \right\rceil, 1 + \left\lfloor \frac{a + D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right)$$

Está provado que os valores de a que conduzem ao pior resultado de tempo de resposta estão no intervalo contínuo:

$$a \in [0, L[\qquad (26)$$

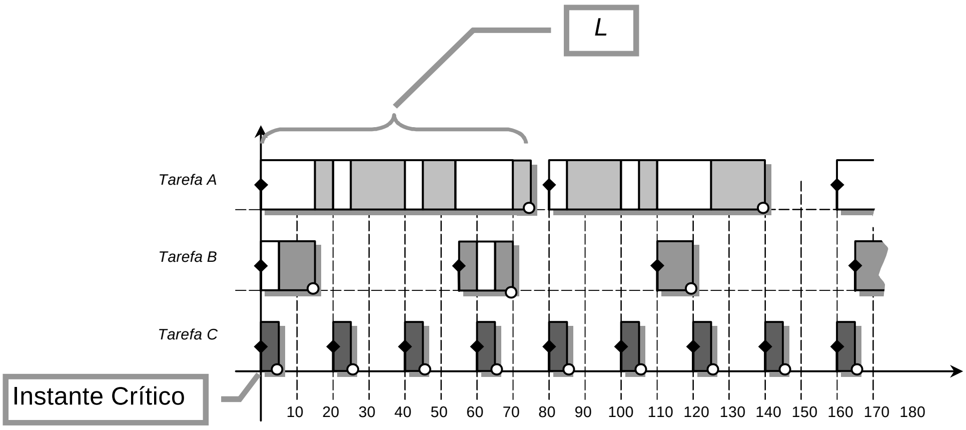
em que *L* é o chamado *Synchronous Busy Period*, ou seja, o intervalo de tempo que o processador vai estar ocupado a processar tarefas sendo todas lançadas simultaneamente no instante 0.

Para o conhecido conjunto de tarefas:

Tarefa	*C*	*T (=D)*
A	35	80
B	10	55
C	5	20

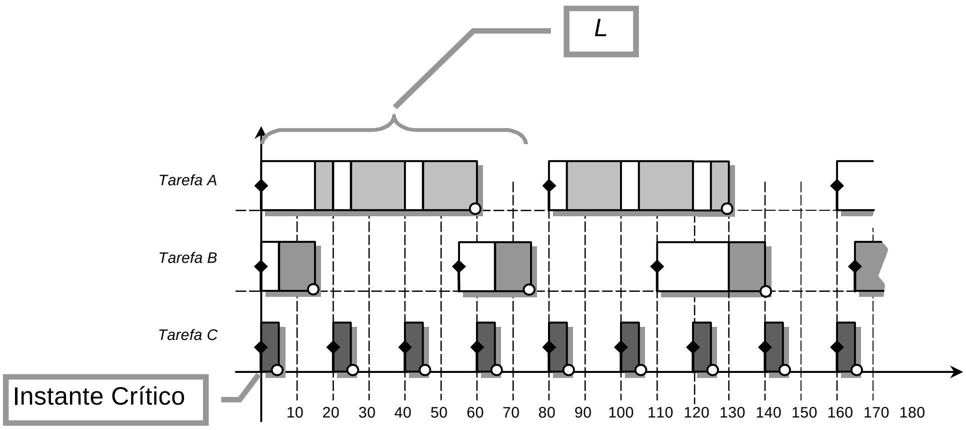
L é 75.

Voltemos a ver o gráfico:



O gráfico anterior representa o escalonamento de acordo com o RM pre-emptivo. Mas *L* é indiferente do tipo de escalonamento e de o ambiente ser ou não pre-emptivo. **(Porquê?)**.

Vejamos outra vez o que se passa com o EDF pre-emptivo:



E então, como é que se determina o *L*?

É óbvio (é como determinar a interferência causada por todas as *N* tarefas):

$$L = \sum_{i=1}^{N} \left(\left\lceil \frac{L}{T_i} \right\rceil \times C_i \right) \tag{27}$$

Trata-se de uma equação recorrente, e por isso determina-se o valor de *L* iterativamente:

$$L^{n+1} = \sum_{i=1}^{N}\left(\left\lceil \frac{L^n}{T_i} \right\rceil \times C_i\right) \tag{28}$$

Por razões práticas, o valor inicial para *L* (L^0) não pode ser 0. E por isso é óbvio que seja o somatório dos tempos de execução de todas as tarefas (1 activação de cada).

$$L^0 = \sum_{i=1}^{N} C_i \tag{29}$$

Vamos então calcular o valor de L para o exemplo dado:

$$L^0 = \sum_{i=1}^{N} C_i = 35 + 10 + 5 = 50$$

$$L^1 = \sum_{i=1}^{N}\left(\left\lceil \frac{L^0}{T_i} \right\rceil \times C_i\right) = \left\lceil \frac{50}{80} \right\rceil \times 35 + \left\lceil \frac{50}{55} \right\rceil \times 10 + \left\lceil \frac{50}{20} \right\rceil \times 5 = 1 \times 35 + 1 \times 10 + 3 \times 5 = 60$$

$$L^2 = \sum_{i=1}^{N}\left(\left\lceil \frac{L^1}{T_i} \right\rceil \times C_i\right) = \left\lceil \frac{60}{80} \right\rceil \times 35 + \left\lceil \frac{60}{55} \right\rceil \times 10 + \left\lceil \frac{60}{20} \right\rceil \times 5 = 1 \times 35 + 2 \times 10 + 3 \times 5 = 70$$

$$L^3 = \sum_{i=1}^{N}\left(\left\lceil \frac{L^2}{T_i} \right\rceil \times C_i\right) = \left\lceil \frac{70}{80} \right\rceil \times 35 + \left\lceil \frac{70}{55} \right\rceil \times 10 + \left\lceil \frac{70}{20} \right\rceil \times 5 = 1 \times 35 + 2 \times 10 + 4 \times 5 = 75$$

$$L^4 = \sum_{i=1}^{N}\left(\left\lceil \frac{L^3}{T_i} \right\rceil \times C_i\right) = \left\lceil \frac{75}{80} \right\rceil \times 35 + \left\lceil \frac{75}{55} \right\rceil \times 10 + \left\lceil \frac{75}{20} \right\rceil \times 5 = 1 \times 35 + 2 \times 10 + 4 \times 5 = 75$$

Mas, no intervalo

$$a \in [0, L[$$

que valores particulares de *a* é que deverão ser testados?

Como é fácil de ver, na equação

$$L_i(a) = \left(1 + \left\lfloor \frac{a}{T_i} \right\rfloor\right) \times C_i + \sum_{\substack{j \neq i \\ D_j \leq a + D_i}} \left(\min\left\{\left\lceil \frac{L_i(a)}{Tj} \right\rceil, 1 + \left\lfloor \frac{a + D_i - D_j}{T_j} \right\rfloor\right\} \times C_j\right)$$

só para alguns valores de *a* é que a interferência máxima causada por uma tarefa *j* altera.

De facto, a função de *Ceiling* (⌈ ⌉) é uma *step function*, que só pode assumir os valores 0, 1, 2, 3, 4, 5, ... Isto é, os números naturais, incluindo o 0.

Por isso, os valores de *a* que interessa testar para uma determinada tarefa *i* são os:

$$\frac{a + D_i - D_j}{T_j} = k, \qquad \text{em que } k \in \aleph_0$$

Isto é,

$$a = k \times T_j + D_j - D_i, \qquad \text{em que } k \in \aleph_0$$

O que quer dizer que:

$$a \in \bigcup_{j=1}^{N} \{k \times T_j + D_j - D_i, k \in \aleph_0\} \cap [0, L[\tag{30}$$

Voltemos então ao conjunto de tarefas:

Tarefa	*C*	*T (=D)*
A	35	80
B	10	55
C	5	20

Para este conjunto de tarefas, *L* = 75.

Como determinar, por exemplo, os valores de *a* relevantes para a tarefa B?

Solução:

$$a \in \bigcup_{j=1}^{3} \{k \times T_j + D_j - 80, k \in \aleph_0\} \cap [0{,}75[$$

então,

$k = 0$

$j = 1, \qquad a = 0 \times 80 + 80 - 80 = 0$

$j = 2, \qquad a = 0 \times 55 + 55 - 80 = -25$

$j = 3, \qquad a = 0 \times 20 + 20 - 80 = -60$

$k = 1$

$j = 1, \qquad a = 1 \times 80 + 80 - 80 = 80$

$j = 2, \qquad a = 1 \times 55 + 55 - 80 = 30$

$j = 3, \qquad a = 1 \times 20 + 20 - 80 = -40$

$k = 2$

$j = 1, \qquad a = 2 \times 80 + 80 - 80 = 160$

$j = 2, \qquad a = 2 \times 55 + 55 - 80 = 85$

$j = 3, \qquad a = 2 \times 20 + 20 - 80 = -20$

$k = 3$

$j = 1, \quad a = 3 \times 80 + 80 - 80 = 240$

$j = 2, \quad a = 3 \times 55 + 55 - 80 = 140$

$j = 3, \quad a = 3 \times 20 + 20 - 80 = 0$

$k = 4$

$j = 1, \quad a = 4 \times 80 + 80 - 80 = 320$

$j = 2, \quad a = 4 \times 55 + 55 - 80 = 195$

$j = 3, \quad a = 4 \times 20 + 20 - 80 = 20$

$k = 5$

$j = 1, \quad a = 5 \times 80 + 80 - 80 = 400$

$j = 2, \quad a = 5 \times 55 + 55 - 80 = 250$

$j = 3, \quad a = 5 \times 20 + 20 - 80 = 40$

$k = 6$

$j = 1, \quad a = 6 \times 80 + 80 - 80 = 480$

$j = 2, \quad a = 6 \times 55 + 55 - 80 = 305$

$j = 3, \quad a = 6 \times 20 + 20 - 80 = 60$

$k = 7$

$j = 1, \quad a = 7 \times 80 + 80 - 80 = 560$

$j = 2, \quad a = 7 \times 55 + 55 - 80 = 360$

$j = 3, \quad a = 7 \times 20 + 20 - 80 = 80$

A partir do valor *k* = 7, todos os valores para a já são maiores do que 75.

Isto quer dizer que os valores de *a* a testar no caso da tarefa B são:

0, 20, 30, 40 e 60.

Então, para determinar os priores tempos de resposta das tarefas escalonadas pelo EDF em ambiente pre-emptivo, é preciso:

<u>**Primeiro** (determinar o valor de *L*):</u>

$$L = \sum_{i=1}^{N} \left(\left\lceil \frac{L}{T_i} \right\rceil \times C_i \right) \tag{31}$$

Este valor é obtido implementando o seguinte algoritmo:

$$L^{n+1} = \sum_{i=1}^{N} \left(\left\lceil \frac{L^n}{T_i} \right\rceil \times C_i \right) \tag{32}$$

Para o qual

$$L^0 = \sum_{i=1}^{N} C_i \tag{33}$$

<u>**Segundo** (para cada tarefa *i*, determinar os valores de *a* relevantes):</u>

Estes valores são obtidos implementando um algoritmo que implemente (note que o valor de *L* já foi obtido do primeiro passo):

$$a \in \bigcup_{j=1}^{N} \left\{ k \times T_j + D_j - D_i, k \in \aleph_0 \right\} \cap [0, L[\tag{34}$$

<u>**Terceiro** (para cada tarefa *i*, e para cada valor de *a* relevante para essa tarefa):</u>

Implementar um algoritmo (**recorrente**) que permita obter o resultado de:

$$L_i(a) = \left(1 + \left\lfloor \frac{a}{T_i} \right\rfloor \right) \times C_i + \sum_{\substack{j \neq i \\ D_j \leq a + D_i}} \left(\min \left\{ \left\lceil \frac{L_i(a)}{Tj} \right\rceil, 1 + \left\lfloor \frac{a + D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) \tag{35}$$

A partir do valor determinado obter (notar que para alguns valores de a pode acontecer que $L_i(a) - a < C_i$:

$$R_i(a) = \max\{C_i, L_i(a) - a\} \tag{36}$$

Por fim, R_i é o maior dos $R_i(a)$:

$$R_i = \max\{R_i(a)\} \tag{37}$$

2.2.4. EDF **Não** Pre-Emptivo

As razões das pequenas diferenças, são as que já foram apontadas para o caso do RM não pre-emptivo.

Em vez da equação (35), é:

$$L_i(a) = \left\lfloor \frac{a}{T_i} \right\rfloor \times C_i + \max_{D_j > a + D_i} \{C_j\} + \sum_{\substack{j \neq i \\ D_j \leq a + D_i}} \left(\min\left\{ 1 + \left\lfloor \frac{L_i(a)}{T_j} \right\rfloor, 1 + \left\lfloor \frac{a + D_i - D_j}{T_j} \right\rfloor \right\} \times C_j \right) \qquad (38)$$

BLOQUEIO!

Em vez da equação (36), é:

$$R_i(a) = \max\{C_i, L_i(a) + C_i - a\} \qquad (39)$$