

3

Escalonamento usando múltiplos períodos

O escalonamento usando múltiplos períodos consiste em dividir a carga total em fragmentos de carga e enviá-las a cada processador mais de uma vez, segundo períodos. Em cada período, frações de carga são enviadas para cada processador uma única vez. Apresenta-se, por exemplo, o escalonamento mostrado na Figura 3.1, onde o envio de dados é feito em três períodos. No primeiro e no segundo, uma determinada quantidade de carga é enviada apenas aos processadores P_3 e P_9 . No último, um número maior de processadores é utilizado.

No exemplo previamente descrito, a mesma ordem de envio foi respeitada em todos os períodos, porém o número de processadores utilizados foi diferente em diferentes períodos. Enquanto isto, no escalonamento apresentado na Figura 3.2, além da ordem de envio ser sempre a mesma, o número de processadores utilizados em cada período também permanece fixo.

Para o escalonamento usando múltiplos períodos deve-se definir o número total de períodos p , bem como a ordem de envio, quantos processadores serão utilizados em cada período e a quantidade de carga $\alpha_{i,j}$ que cada processador P_i receberá em cada rodada j . Devido ao grande número de incógnitas, em todas as heurísticas encontradas na literatura [3, 5, 23] é estipulado que a ordem de envio e o número de processadores permanecem sempre os mesmos durante toda a computação, facilitando a elaboração de técnicas de escalonamento em múltiplos períodos. Neste trabalho também considera-se uma única ordem de envio, porém são estudados casos onde o número de processadores utilizados varia durante o processamento.

Inicialmente, neste capítulo é apresentada uma técnica proposta em [5] para sistemas com latências desprezíveis. Logo após são apresentadas duas heurísticas encontradas em [3, 4] para o escalonamento considerando latências. A primeira delas utiliza períodos de mesma duração e a segunda utiliza períodos com durações que diminuem a medida que a computação termina.

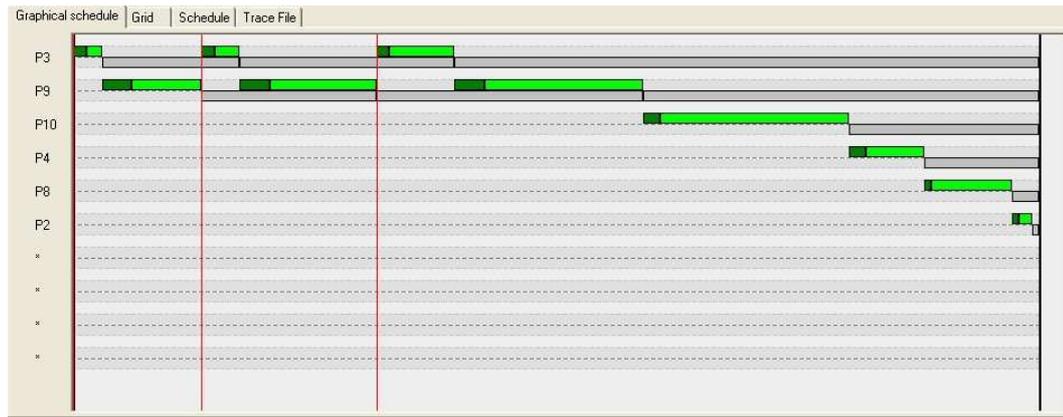


Figura 3.1: Escalonamento em três períodos

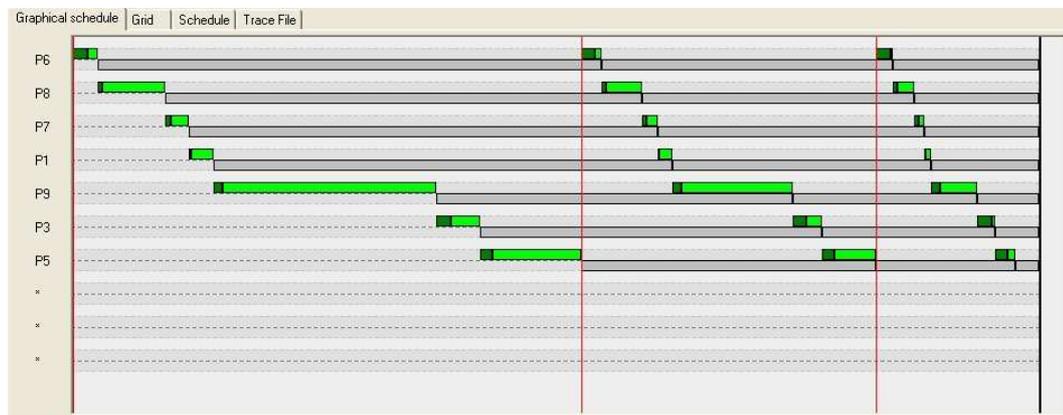


Figura 3.2: Escalonamento em três períodos sempre utilizando sete processadores

Também considerando latências e estendendo o desenvolvimento feito em [5], é proposta neste trabalho uma nova técnica de escalonamento que será apresentada também neste capítulo.

Partindo-se para o estudo de uma solução exata para o problema, não se encontrou na literatura formulações como modelos de programação linear ou não-linear inteiras. Neste capítulo serão apresentados modelos de programação linear inteira mista para o problema, sendo esta mais uma das contribuições deste trabalho.

3.1

Resultados anteriores para sistemas sem latências

Em [5] foi apresentada uma técnica para a distribuição da carga aos processadores em múltiplos períodos. Nela, o conjunto de processadores utilizados e a ordem de envio das tarefas permanecem inalterados durante todos os períodos. Além disso, no momento que um processador recebe sua carga, esse irá processá-la e, imediatamente após o término do processamento, irá começar a receber a carga do próximo período e assim por diante. No último período, todos os processadores terminam a computação ao mesmo tempo. Dadas estas restrições, as cargas a serem enviadas para cada processador em cada período podem ser definidas como:

$$\alpha_{i,j} = \frac{1}{w_i} \left(\sum_{k=i+1}^n \alpha_{k,j} G_k + \sum_{k=1}^i \alpha_{k,j+1} G_k \right), \quad \forall (1 \leq i \leq n, 1 \leq j \leq p-1) \quad (3-1)$$

$$\alpha_{i,p} = \alpha_{n,p} \prod_{k=i+1}^{n-1} \frac{w_k + G_k}{w_{k-1}}, \quad \forall (1 \leq i \leq n-1) \quad (3-2)$$

Utilizando-se a equação $\alpha_{1,1} + \alpha_{1,2} + \dots + \alpha_{n,p} = W$, que implica na computação de toda a carga, tem-se um sistema de equações com pn equações lineares e pn variáveis que pode ser facilmente resolvido.

As igualdades 3-1 e 3-2 garantem uma utilização ininterrupta dos processadores após o início da primeira computação, bem como garantem que não ocorrerá ociosidade de comunicação por parte do processador mestre. Ambos conjuntos de igualdades são muito semelhantes às equações 3-3 e 3-4, que serão apresentadas em detalhe na Seção 3.3.

Esta técnica é apresentada em [5] como um melhoramento da técnica de escalonamento em período único descrita na Seção 2.1. Desta forma, a ordem de envio utilizada é a mesma descrita para o escalonamento em período único, ou seja, em ordem não crescente das taxas de comunicação.

3.2

Resultados anteriores para sistemas com latências

3.2.1

Heurística com período fixo *HFix*

Esta heurística, proposta em [4], visa resolver o problema fixando o tamanho do período a priori. Consiste basicamente de três passos:

1. Reordenar os processadores em ordem não crescente das taxas de comunicação, ou seja, $G_1 \leq G_2 \leq \dots \leq G_n$;
2. Calcular uma estimativa de tempo total de processamento T . Esta estimativa é feita assumindo um balanceamento perfeito de carga, negligenciando todas as latências. Neste trabalho utiliza-se a mesma estimativa apresentada em [4], onde $T = W / (\sum_{i=1}^n \frac{1}{(w_i + G_i)})$. Com esta estimativa, estipula-se a duração do período T_p como sendo \sqrt{T} . Caso T_p seja menor que a latência de comunicação mais o tempo de envio de uma unidade de carga ao primeiro computador, então T_p recebe este valor.
3. Para determinar a quantidade $\alpha_{i,j}$ a ser processada pelo processador P_i num período j , calcula-se $\alpha_{i,j} = T_p / w_i$. Nota-se porém que, para um processador P_k , se $\sum_{i=1}^k (g_i + G_i \alpha_{i,j}) > T_p$, então $\alpha_{k,j} = \max\{0, (T_p - \sum_{i=1}^{k-1} (g_i + G_i \alpha_{i,j}) - g_k) / G_k\}$. Desta forma, o tempo total de comunicação não ultrapassa o fim do período corrente. Dependendo das latências de comunicação e do tamanho de T_p , alguns processadores podem não participar da execução (possuem $\alpha_{i,j} = 0$).

Pode-se encontrar na Figura 3.3 um exemplo de escalonamento usando esta técnica. Nota-se que no último período alguns processadores deixam de ser utilizados pelo fato de toda a carga já ter sido computada.

Neste exemplo, a razão entre o tempo de comunicação e o tempo de processamento é muito baixa, o que causa a necessidade de uma aproximação de 20x no gráfico para visualizar melhor os períodos de comunicação, como apresentado na Figura 3.4.

3.2.2

Heurística adaptativa *HAdapt*

Uma pequena modificação na heurística descrita anteriormente foi também proposta em [4]. Ao invés de se calcular a duração do período uma

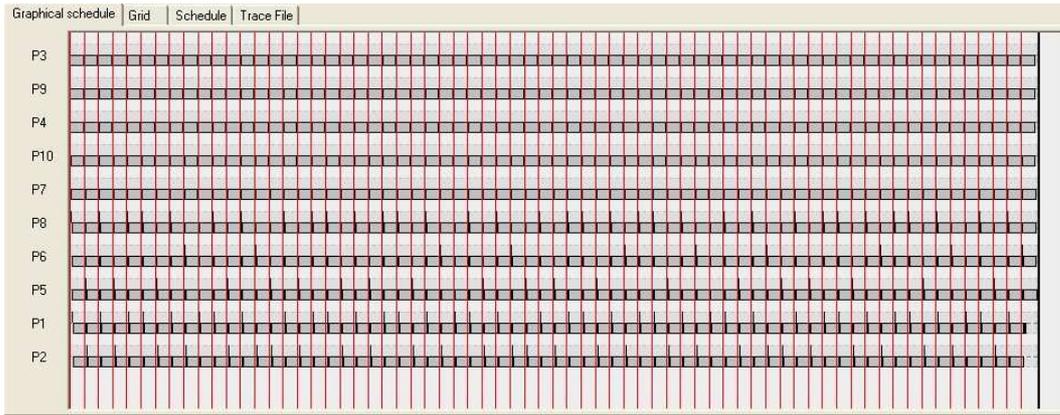


Figura 3.3: Escalonamento usando a heurística de período fixo (aproximação 1x)

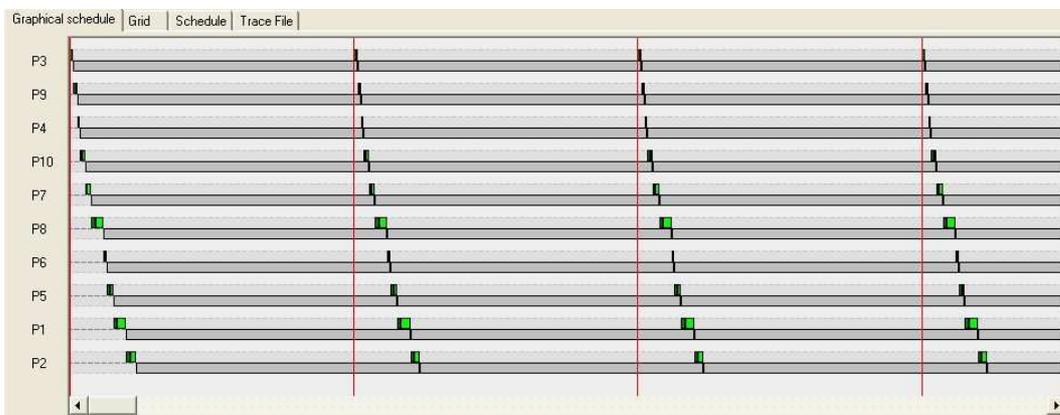


Figura 3.4: Escalonamento usando a heurística de período fixo (aproximação 20x)

única vez no início da computação, o tamanho do período é recalculado a cada período, calculando-se a estimativa de término conforme a quantidade de dados ainda não computados. Desta forma, o tamanho dos períodos vai diminuindo conforme a carga vai sendo computada, como pode ser visto na Figura 3.5.

3.3

Nova heurística *HeuMul* para escalonamento em múltiplos períodos

Fixando-se a ordem de envio, o número de processadores, o número de períodos a serem utilizados e mantendo-se o mesmo número de processadores utilizados em todos os períodos, desenvolveu-se um método para calcular o total de carga a ser enviado a cada processador. Com uma distribuição similar à apresentada na Seção 3.1 para sistemas sem latências, pode-se encontrar uma

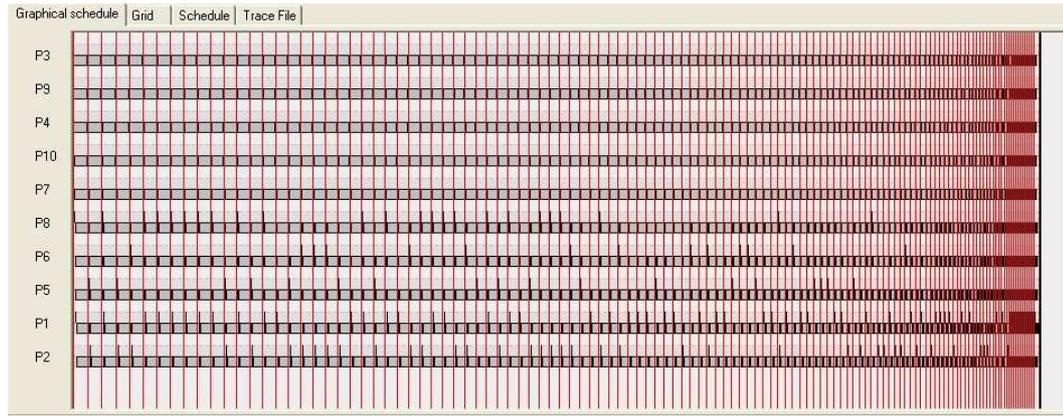


Figura 3.5: Escalonamento usando a heurística adaptativa (aproximação 1x)

distribuição para um cenário onde as latências são consideradas:

$$\alpha_{i,j}w_i = \sum_{k=i+1}^n (g_k + \alpha_{k,j}G_k) + \sum_{k=1}^i (g_k + \alpha_{k,j+1}G_k),$$

$$\forall(1 \leq i \leq n, 1 \leq j \leq p-1) \quad (3-3)$$

$$\alpha_{i,p}w_i = g_{i+1} + \alpha_{i+1,p}(w_{i+1} + G_{i+1}) \quad , \forall(1 \leq i \leq n-1) \quad (3-4)$$

Utilizando-se a equação $\alpha_{1,1} + \alpha_{1,2} + \dots + \alpha_{n,p} = W$, que implica na computação de toda a carga, tem-se um sistema de equações com pn equações lineares e pn variáveis, que pode ser facilmente resolvido com um algoritmo de complexidade $O((pn)^3)$.

As equações 3-3 implicam que todo processador P_i , num período j , permanecerá processando a carga $\alpha_{i,j}$ até o momento em que inicia-se o processamento da carga $\alpha_{i,j+1}$. No período j , após o término do recebimento de $\alpha_{i,j}$ dados por P_i , o processador mestre enviará dados a todos os processadores que o sucedem na ordem de envio, gastando com isso um período de tempo igual a $\sum_{k=i+1}^n (g_k + \alpha_{k,j}G_k)$. Após terminar o envio dos dados referentes ao período j , o processador mestre começará o envio de dados do período $j+1$. Antes porém de enviar dados a P_i , o processador mestre enviará dados a todos os processadores que o precedem na ordem de envio, gastando para isso um período de tempo igual a $\sum_{k=1}^{i-1} (g_k + \alpha_{k,j+1}G_k)$. Somando-se o tempo necessário para P_i receber $\alpha_{i,j+1}$ a estes dois períodos de tempo, tem-se o tempo total decorrido desde o término do recebimento de $\alpha_{i,j}$ até o término

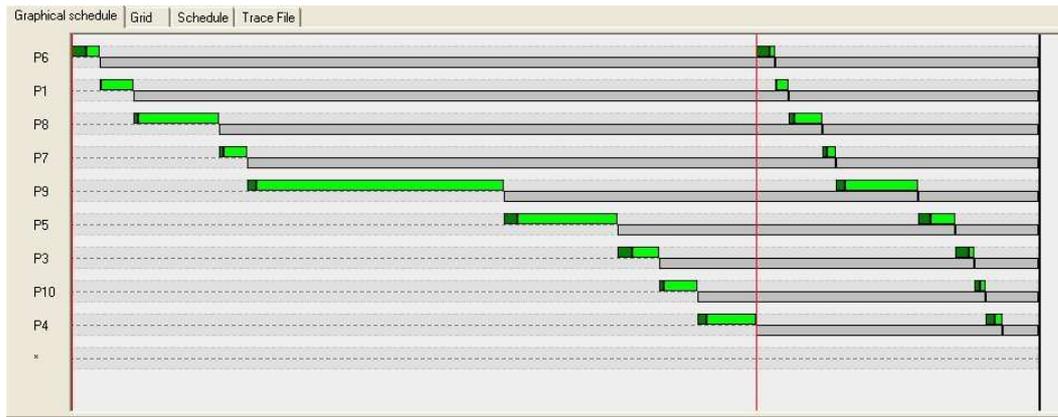


Figura 3.6: Escalonamento com número fixo de processadores e períodos

do recebimento de $\alpha_{i,j+1}$, que é igual a $\alpha_{i,j}w_i$. Garante-se desta forma o processamento ininterrupto por parte deste computador. Esta propriedade pode ser vista na Figura 3.6, onde, por exemplo, a barra que indica o tempo de computação do processador P_9 tem a mesma extensão do somatório dos comprimentos das barras de comunicação de P_5 , P_3 , P_{10} e P_4 no primeiro período, mais o comprimento das barras de comunicação de P_6 , P_1 , P_8 , P_7 e P_9 no último período.

As equações 3-4 implicam em que, no último período, o tempo de processamento $w_i\alpha_{i,p}$ do processador P_i deve ser igual ao tempo de comunicação e processamento do próximo processador na ordem de envio.

Utilizando-se a heurística *HeuRet* para escalonamento em período único, obtém-se a ordem de envio *MelhorOrder* e o número de processadores utilizados *MelhorL**. Com estes dados de entrada, mais o número máximo de períodos *MaxP* a serem utilizados, desenvolveu-se a heurística *HeuMul*. Esta heurística varia o número de processadores utilizados de um a *MelhorL** e o número de períodos de um a *MaxP*, verificando qual a configuração que gera o menor *makespan* dado o escalonamento descrito pelo sistema de equações 3-3 e 3-4.

3.4

Novos modelos de programação linear inteira mista

Nesta seção serão apresentados dois modelos de programação linear inteira mista para o problema de escalonamento em múltiplos períodos. É importante salientar que, em ambos, a ordem de envio é mantida sempre a

mesma em todos os períodos. Salienta-se porém que é possível que existam soluções melhores com ordens de envio diferentes, porém estas não estão sendo consideradas. Optou-se por esta abordagem para que seja possível comparar os resultados da resolução dos modelos com os resultados das heurísticas, que também utilizam uma ordem única. Com o primeiro modelo apresentado é possível obter-se o escalonamento ótimo para um determinado número fixo de períodos utilizando-se sempre um mesmo número de processadores, ou seja, obtém-se a ordem ótima de envio, o número de processadores a serem utilizados e as cargas a serem enviadas. No segundo modelo apresentado é possível obter-se o número ótimo de períodos e quantos processadores devem ser utilizados em cada. Por fim, propõe-se um terceiro modelo com aprimoramentos em relação ao segundo.

3.4.1

Modelo com número fixo de processadores e períodos

Numa primeira formulação, estipulou-se que o número de períodos que se deseja utilizar é conhecido. Além disso, cada processador ou não é utilizado em nenhum período, ou é utilizado em todos, seguindo sempre a mesma ordem de envio.

Neste modelo tem-se as características do sistema e o total W de dados a serem processados como constantes juntamente com as seguintes variáveis:

$$x_{i,j} = \begin{cases} 1, & \text{caso } P_i \text{ seja o } j\text{-ésimo a receber dados em todos os períodos,} \\ 0, & \text{caso contrário.} \end{cases}$$

$$\alpha_{k,i,j} = \begin{cases} 0, & \text{caso } x_{i,j} = 0, \\ \geq 0, & \text{quantidade de dados transmitida ao processador } P_i \text{ no} \\ & \text{período } k \text{ caso } x_{i,j} = 1. \end{cases}$$

$$T = \text{tempo total ou } makespan$$

$$t_{k,j} = \text{momento em que o } j\text{-ésimo processador começará a receber dados no período } k$$

Com o Modelo 6, é possível determinar a ordem de envio ótima, bem

como a quantidade a ser enviada a cada processador em cada período, como pode ser visto na solução apresentada na Figura 3.7. Nota-se que este modelo possui o mesmo número de variáveis inteiras que os modelos apresentados para o problema com período único, porém o número de variáveis contínuas aumenta para $pn^2 + pn + 1$.

Com este modelo pode-se obter soluções ótimas em que nem todos os processadores são efetivamente utilizados em todos os períodos, como pode ser visto no exemplo da Figura 3.8. Porém, mesmo sem receber dados, o tempo de latência de comunicação é considerado, pois depende apenas de $x_{i,j}$, o que é sanado no próximo modelo que será apresentado.

$$\text{Minimize } T \quad (2-3)$$

s.a.

$$\sum_{i=1}^n x_{i,j} \leq 1 \quad \forall(1 \leq j \leq n) \quad (2-13)$$

$$\sum_{j=1}^n x_{i,j} \leq 1 \quad \forall(1 \leq i \leq n) \quad (2-14)$$

$$\sum_{i=1}^n x_{i,j} \geq \sum_{i=1}^n x_{i,j+1} \quad \forall(1 \leq j < n) \quad (2-20)$$

$$\sum_{k=1}^p \sum_{i=1}^n \sum_{j=1}^n \alpha_{k,i,j} = W \quad (3-5)$$

$$\sum_{k=1}^p \alpha_{k,i,j} \leq W_i x_{i,j} \quad \forall(1 \leq i, j \leq n) \quad (3-6)$$

$$t_{1,1} = 0 \quad (3-7)$$

$$t_{k,j} \geq t_{k,j-1} + \sum_{i=1}^n (g_i x_{i,j-1} + G_i \alpha_{k,i,j-1}) \quad \forall(2 \leq j \leq n, 1 \leq k \leq p) \quad (3-8)$$

$$t_{k,j} + \sum_{i=1}^n (g_i x_{i,j} + G_i \alpha_{k,i,j}) \geq$$

$$t_{k-1,j} + \sum_{i=1}^n (g_i x_{i,j} + G_i \alpha_{k-1,i,j} + w_i \alpha_{k-1,i,j}) \quad \forall(1 \leq j \leq n, 2 \leq k \leq p) \quad (3-9)$$

$$t_{k,1} \geq t_{k-1,n} + \sum_{i=1}^n (g_i x_{i,n} + G_i \alpha_{k-1,i,n}) \quad \forall(2 \leq k \leq p) \quad (3-10)$$

$$t_{p,j} + \sum_{i=1}^n (g_i x_{i,j} + (G_i + w_i) \alpha_{p,i,j}) = T \quad \forall(1 \leq j \leq n) \quad (3-11)$$

$$x_{i,j} \in \{0, 1\} \quad \forall(1 \leq i, j \leq n) \quad (2-8)$$

$$\alpha_{k,i,j} \geq 0 \quad \forall(1 \leq i, j \leq n, 1 \leq k \leq p) \quad (3-12)$$

Modelo 6: Modelo com número fixo de processadores e períodos

As restrições 3-5 implicam que o total de dados enviados aos processadores deve ser igual ao total de dados a serem tratados.

As restrições 3-6 implicam que, em qualquer período k , um processador só poderá receber dados como sendo o j -ésimo, caso este processador esteja alocado como tal. Além disso, a quantidade máxima W_i de carga que um processador P_i poderá receber é igual a $(LimiteSuperior - g_i)/w_i$, onde $LimiteSuperior$ é um valor máximo para o *makespan*, que pode ser obtido a partir de alguma heurística. Para o cálculo de W_i , considera-se uma taxa de comunicação infinita ($G_i = 0$), que o processador será o primeiro a receber

carga e que permanecerá processando ininterruptamente, o que garante que $\sum_{k=1}^p \alpha_{k,i,j} \leq W_i = (\text{LimiteSuperior} - g_i)/w_i, \forall (1 \leq i \leq n, 1 \leq j \leq p)$, para toda solução viável do problema.

A restrição 3-7 indica que o primeiro processador começará a receber os dados do primeiro período no instante 0.

As restrições 3-8 implicam que, para todo período k , cada j -ésimo processador a receber dados neste período, só pode começar a receber carga após o anterior na ordem de envio (deste mesmo período) ter terminado de receber seus dados.

As restrições 3-9 implicam que, para todo período k , cada j -ésimo processador a receber dados neste período só pode começar a processar a carga (instante $t_{k,j} + \sum_{i=1}^n (g_i x_{i,j} + G_i \alpha_{k,i,j})$) após ele mesmo ter terminado de processar seus dados do período anterior (instante $t_{k-1,j} + \sum_{i=1}^n (g_i x_{i,j} + G_i \alpha_{k-1,i,j} + w_i \alpha_{k-1,i,j})$). Salienta-se que, uma vez que a ordem de envio se mantém sempre a mesma, o j -ésimo processador num período é também o j -ésimo em qualquer outro período.

As restrições 3-10 implicam que, para todo período k , o primeiro processador a receber dados só poderá começar a receber após o último do período anterior ter terminado de receber. Estas restrições, juntamente com as 3-8, garantem que não haverá concorrência de comunicações entre o processador mestre e processadores escravos.

As restrições 3-11 implicam que no último período todos os processadores devem terminar o processamento ao mesmo tempo. Isto é uma característica da solução ótima, sendo facilmente verificado por contradição. Suponha que se tem uma solução ótima onde no último período um processador P_i termina o processamento no instante I e algum processador P_j no instante $I - r$. Desta forma, o escalonador poderia enviar menos carga a P_i e mais a P_j , de forma que ambos terminassem no mesmo instante I' , obtendo-se $I - r < I' < I$. Se for aplicado este procedimento até todos os processadores terminarem no mesmo instante, obter-se-ia um *makespan* menor, que é uma contradição.

3.4.2

Modelo com número máximo de períodos

Foi verificado que o escalonamento obtido pela resolução do Modelo 6 para certos casos acaba por utilizar alguns processadores apenas nos últimos

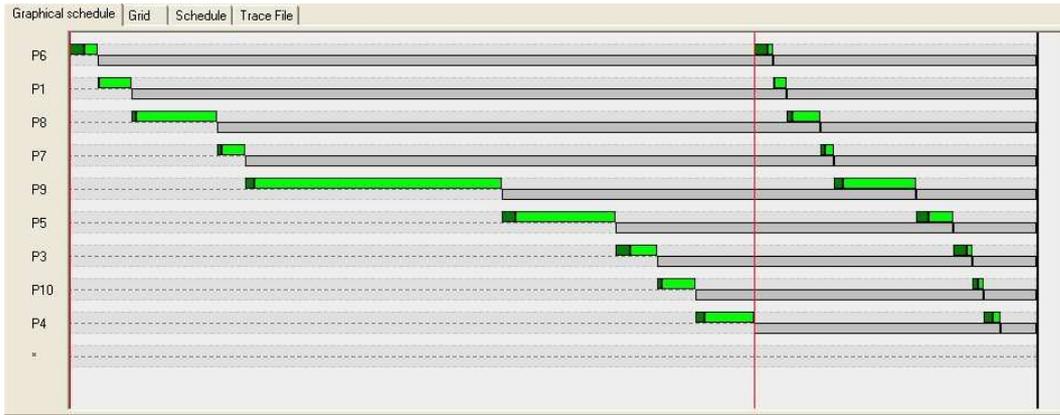


Figura 3.7: Escalonamento ótimo resultante do modelo linear inteiro com número fixo de processadores e períodos

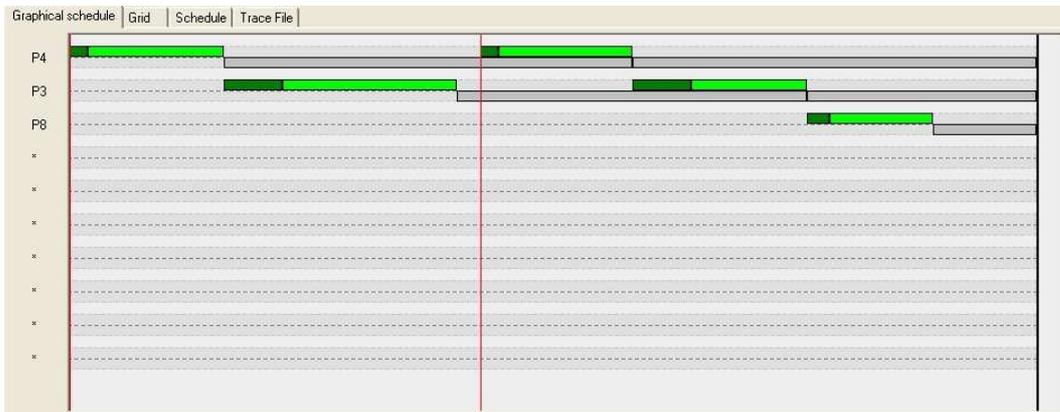


Figura 3.8: Escalonamento ótimo resultante do modelo linear inteiro com número fixo de processadores e períodos

períodos (assim como no escalonamento apresentado na Figura 3.8). Devido à inflexibilidade do Modelo 6, mesmo não utilizando alguns processadores nos primeiros períodos, seus tempos de latência de comunicação eram contados e o processador mestre ficava ocioso. Foi então elaborado o Modelo 7, que visa permitir que alguns processadores só sejam utilizados a partir de algum período, sem que seus tempos de latência sejam contados (assim como no escalonamento apresentado da Figura 3.9). Além desta característica, o Modelo 7 permite também que seja encontrado automaticamente o número ótimo de períodos a serem utilizados no escalonamento (dado um máximo p).

Para atingir estes objetivos, as variáveis inteiras x foram redefinidas:

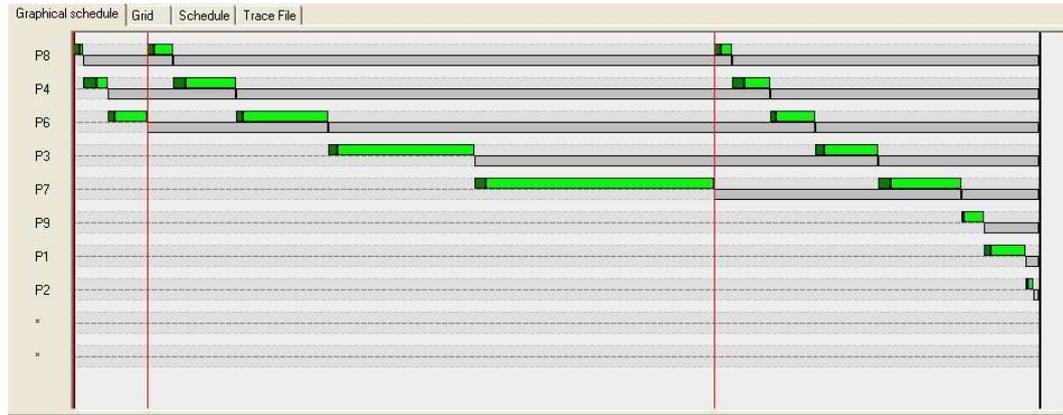


Figura 3.9: Escalonamento ótimo resultante do modelo linear inteiro com número máximo de períodos

$$x_{k,i,j} = \begin{cases} 1, & \text{caso } P_i \text{ seja o } j\text{-ésimo a receber dados no período } k \\ 0, & \text{caso contrário} \end{cases}$$

Como pode ser visto no Modelo 7, o modelo é composto por pn^2 variáveis inteiras, $pn^2 + pn + 1$ variáveis contínuas e $p(5n + 2n^2 - 1) + 1$ restrições (não incluindo as que definem as variáveis x como sendo binárias e as variáveis α como sendo contínuas e maiores ou iguais a zero)

As restrições 3-13 implicam que, para todo período k , no máximo um processador pode ser alocado como sendo o j -ésimo. Por exemplo, para $k = 3$ e $j = 2$, a restrição implica que poderá haver no máximo um processador como sendo o segundo na ordem de envio do terceiro período.

As restrições 3-14 implicam que, para todo período k , um processador pode ser alocado em no máximo uma posição dentro da ordem de envio. Por exemplo, para $i = 3$ a restrição implica que o processador P_3 poderá receber dados no máximo uma vez (ser alocado no máximo em uma posição na ordem de envio).

As restrições 3-15 implicam que, para todo período k , só pode ser alocado um processador como sendo o j -ésimo caso haja um processador alocado na posição $j - 1$ da ordem de envio. Por exemplo, para $j = 4$ e $k = 1$ tem-se que, no período 1, só poderá haver um quinto processador na ordem de envio caso haja um processador alocado como quarto a receber dados.

$$\text{Minimize } T \quad (2-3)$$

s.a.

$$\sum_{i=1}^n x_{k,i,j} \leq 1 \quad \forall(1 \leq j \leq n, 1 \leq k \leq p) \quad (3-13)$$

$$\sum_{j=1}^n x_{k,i,j} \leq 1 \quad \forall(1 \leq i \leq n, 1 \leq k \leq p) \quad (3-14)$$

$$\sum_{i=1}^n x_{k,i,j} \geq \sum_{i=1}^n x_{k,i,j+1} \quad \forall(1 \leq j < n, 1 \leq k \leq p) \quad (3-15)$$

$$x_{k,i,j} \geq x_{k-1,i,j} \quad \forall(1 \leq i, j \leq n, 2 \leq k \leq p) \quad (3-16)$$

$$\sum_{k=1}^p \sum_{i=1}^n \sum_{j=1}^n \alpha_{k,i,j} = W \quad (3-5)$$

$$\alpha_{k,i,j} \leq W_i x_{k,i,j} \quad \forall(1 \leq i, j \leq n, 1 \leq k \leq p) \quad (3-6)$$

$$t_{1,1} = 0 \quad (3-7)$$

$$t_{k,j} \geq t_{k,j-1} + \sum_{i=1}^n (g_i x_{k,i,j-1} + G_i \alpha_{k,i,j-1}) \quad \forall(2 \leq j \leq n, 1 \leq k \leq p) \quad (3-8)$$

$$t_{k,j} + \sum_{i=1}^n (g_i x_{k,i,j} + G_i \alpha_{k,i,j}) \geq t_{k-1,j} + \sum_{i=1}^n (g_i x_{k-1,i,j} + (G_i + w_i) \alpha_{k-1,i,j}) \quad \forall(1 \leq j \leq n, 2 \leq k \leq p) \quad (3-9)$$

$$t_{k,1} \geq t_{k-1,n} + \sum_{i=1}^n (g_i x_{k-1,i,n} + G_i \alpha_{k-1,i,n}) \quad \forall(2 \leq k \leq p) \quad (3-10)$$

$$t_{p,j} + \sum_{i=1}^n (g_i x_{p,i,j} + (G_i + w_i) \alpha_{p,i,j}) = T \quad \forall(1 \leq j \leq n) \quad (3-11)$$

$$\sum_{f=1}^n x_{p,f,j} \geq \sum_{h=j}^n x_{p,i,h} \quad \forall(1 \leq j, i \leq n) \quad (3-17)$$

$$x_{k,i,j} \in \{0, 1\} \quad \forall(1 \leq i, j \leq n, 1 \leq k \leq p) \quad (3-18)$$

$$\alpha_{k,i,j} \geq 0 \quad \forall(1 \leq i, j \leq n, 1 \leq k \leq p) \quad (3-12)$$

Modelo 7: Modelo com número máximo de períodos

As restrições 3-16 implicam que um processador só pode ser alocado como sendo o j -ésimo a receber dados no período $k - 1$, se este mesmo processador tiver sido alocado como sendo o j -ésimo a receber dados no período k .

As desigualdades válidas 2-32 criadas para as variáveis x do modelo elaborado para escalonamento em período único, apresentadas na Seção 2.4.4, são também utilizadas na formulação aqui proposta. Nota-se que as relações entre as variáveis x do modelo para período único possuem as mesmas características das variáveis x deste modelo, referentes a cada período. A utilização destas restrições em todos os períodos causa porém uma grande perda de performance na resolução do modelo. Desta forma optou-se por utilizá-las apenas nas variáveis x referentes ao último período, como descrito nas desigualdades 3-17.

