3. O Problema do Sequenciamento em uma Única Máquina

O problema do sequenciamento em uma única máquina é frequentemente muito simples e quase sempre parte de um problema de programação complexo. Segundo Pinedo (2008), os problemas do sequenciamento em uma única máquina muitas vezes têm propriedades que os de em máquinas em paralelo ou em série não possuem. Os resultados que podem ser obtidos para os problemas do sequenciamento em uma única máquina não só fornecem o conhecimento para o ambiente de uma única máquina, como também fornecem base para heurísticas aplicáveis a ambientes mais complexos.

Na prática, os problemas de programação em ambientes mais complicados são frequentemente decompostos em subproblemas de uma única máquina.

Por exemplo, um ambiente complexo, com um único gargalo, pode dar origem a um modelo de sequenciamento em uma única máquina.

Dessa forma, o problema do sequenciamento em uma única máquina é importante por diversas razões, dentre elas pode-se citar:

a) O processo de aprendizado, já que o problema do sequenciamento pode ilustrar uma variedade de tópicos de sequenciamento em um modelo tratável. Esse problema fornece um contexto para que se investigue muitas medidas de desempenho e técnicas de solução. Além disso, é uma base para o desenvolvimento do entendimento de conceitos de sequenciamento úteis para modelar sistemas mais complexos.

b) Para entender completamente o comportamento de um sistema complexo, é vital entender como funciona cada um de seus componentes e muito frequentemente o problema de uma única máquina aparece como componente elementar em um problema de sequenciamento maior.

c) Algumas vezes é possível resolver o problema do sequenciamento em uma única maquina independentemente e então incorporar o resultado

c) Algumas vezes é possível resolver o problema do sequenciamento em uma única maquina independentemente e então incorporar o resultado em um problema maior. Por exemplo, em um processo com múltiplas operações, frequentemente existe uma operação gargalo e o tratamento dessa operação gargalo, vista como uma análise de um problema de uma única máquina, determina as propriedades de todo o sequenciamento.

d) Em outros casos, o nível em que as decisões devem ser tomadas pode permitir que as instalações de processamento sejam tratadas como um conjunto, como se fossem uma única máquina.

De acordo com Baker (1974), o problema básico de sequenciamento em uma única máquina (também conhecido como sequenciamento de permutação) pode ser caracterizado pelas seguintes condições (nem sempre satisfeitas em situações reais):

1) Um conjunto de *n* trabalhos de uma única operação independente está disponível para processamento no tempo zero;
2) Os tempos de *set up* (preparação de máquina) para os trabalhos são independentes da sequência dos trabalhos e podem ser incluídos nos tempos de processamento;
3) A descrição dos trabalhos é conhecida previamente;
4) Uma máquina está continuamente disponível e nunca haverá tempo ocioso enquanto houver uma tarefa esperando para processamento; e
5) Uma vez que um trabalho é iniciado, ele é processado até seu término, sem interrupção.

Sobre essas condições, há uma correspondência entre a sequência dos *n* trabalhos um-a-um e uma permutação dos índices dos trabalho 1, 2, ..., n. O número total de soluções distintas para o problema básico de sequenciamento em uma única máquina é dado por *n*! que é o número de permutações possíveis dos *n* elementos. Sempre que um sequenciamento é completamente caracterizado por uma permutação de inteiros, ele é chamado de sequência de permutação. Na descrição de sequências de permutação, é útil usar colchetes para indicar a posição na sequência. Desse modo, [5] = 2 significa que o quinto trabalho na sequência é o trabalho 2. De maneira análoga, $d_{[2]}$ refere-se à data prometida do segundo trabalho na sequência.

Reitera-se que a condição 2, para um problema básico de sequenciamento, o tempo de processamento t_j incluirá geralmente tanto os tempos diretos de processamento quanto o tempo de preparação das instalações (tempo de *set up*).

O tempo inicial pode ser entendido como o tempo de chegada – o momento em que o trabalho *j* está disponível nas instalações de processamento – e no modelo básico a suposição da condição 1 é que $r_j = 0$ para todos os trabalhos.

3.1. Definição de Atributos

Ao lidar com atributos de trabalhos para o modelo de uma única máquina, é útil distinguir entre informações conhecidas previamente e informações que são geradas como resultados de decisões de sequenciamento. A informação que é conhecida previamente serve como parâmetro de entrada para função de sequenciamento e é usualmente conveniente usar letras minúsculas para denotar esse tipo de informação. As três informações básicas que ajudam a descrever trabalhos no problema básico determinístico de uma única máquina são:

- Tempo de processamento (t_j): tempo de processamento requerido pelo trabalho *j*;
- Data inicial (r_j): o ponto no tempo em que o trabalho *j* está disponível para processamento; e
- Data prometida (d_j): é o ponto no tempo em que o processamento do trabalho *j* está prometido para ser concluído.

As datas prometidas podem não ser pertinentes em certos problemas, mas estabelecer os prazos (*deadlines)* é um problema comum na indústria e o problema básico pode auxiliar na determinação do prazo de entrega.

É conveniente usar letras maiúsculas para denotar as informações resultantes do sequenciamento.

Data de Conclusão (C_j). O tempo no qual o processamento do trabalho *j* é terminado.

Os critérios quantitativos para escolher uma sequência são geralmente funções dos tempos de conclusão. Duas quantidades importantes são:

Tempo de Fluxo (*Flowtime*) – F_j – Tempo total que o trabalho *j* fica no sistema:

$$F_j = C_j - r_j \tag{3.1}$$

Defasagem (*Lateness*) – L_j – Diferença entre a data de conclusão e a data prometida do trabalho *j*, podendo assumir valores positivos ou negativos:

$$L_j = C_j - d_j \tag{3.2}$$

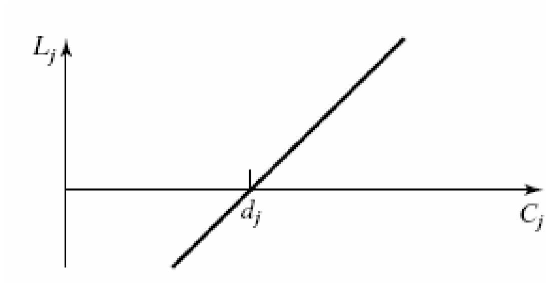

Figura 4 – Gráfico do Atraso (L_j) x Tempo de Conclusão (C_j).
Fonte: Pinedo (2008)

Essas duas quantidades refletem dois critérios importantes. O tempo de fluxo mede a resposta do sistema para demandas individuais por serviço e representa o intervalo que um trabalho leva entre sua chegada e sua saída estando, portanto, ligado ao estque em processo. (Esse intervalo é chamado de "tempo de atravessamento".) A defasagem, $L_{j,}$ mede a conformidade do sequenciamento em relação ao prometido. É importante notar que a defasagem terá valor negativo quando um trabalho é finalizado antecipadamente. Defasagens negativas podem representar serviços ou melhores do que solicitados, enquanto atrasos positivos representam, quase sempre, serviços piores do que requisitados. Em muitas situações, penalidades distintas e outros custos serão associados para defasagens positivas e para defasagens negativas. Dessa forma, tem-se as definições de Atraso (*Tardiness*) e Antecipação (*Earliness*):

Atraso (*Tardiness*) – T_j – é quanto o trabalho *j* é terminado com atraso em relação à sua data prometida, caso contrário será considerado zero:

$$T_j = \max\{0, L_j\} \tag{3.3}$$

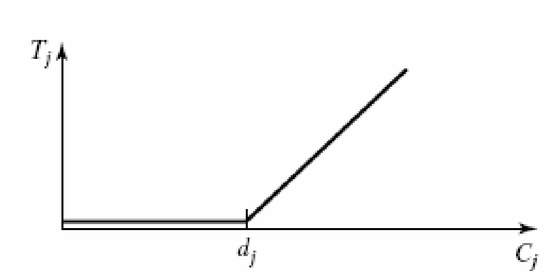

Figura 5 – Gráfico (T_j) x (C_j).

Fonte: Pinedo (2008)

Antecipação (*Earliness*) – E_j – é quanto o trabalho *j* é terminado com antecipação em relação à sua data prometida, caso contrário será considerado zero:

$$E_j = \max\{d_j - C_j, 0\} \tag{3.4}$$

Makespan – C_{max} – O *makespan*, definido como o maior dentre as datas de conclusão ($C_1, \ldots, C_n$), ou seja, ao tempo de conclusão do último trabalho a sair do sistema ($C_{[n]}$). Minimizar o *makespan* usualmente implica em uma boa utilização dos recursos (máquinas).

3.2. Medidas de Desempenho

O sequenciamento é geralmente avaliado pelo conjunto de quantidades que envolvem as informações sobre todos os trabalhos, resultando em uma medida de desempenho uni-dimensional. As medidas de desempenho do sequenciamento geralmente são funções do conjunto de datas de conclusão no sequenciamento. Por exemplo, suponha que *n* trabalhos precisem ser sequenciados. As medidas de desempenho globais podem ser definidas como:

a) Média do tempo de fluxo:

$$\overline{F} = \frac{1}{n}\sum_{j=1}^{n} F_j \tag{3.5}$$

b) Média dos Atrasos:

$$\overline{T} = \frac{1}{n}\sum_{j=1}^{n} T_j \tag{3.6}$$

c) Tempo de Fluxo Máximo:

$$F_{\max} = \max_{1 \leq j \leq n}\{F_j\} \tag{3.7}$$

d) Atraso Máximo:

$$T_{\max} = \max_{1 \leq j \leq n}\{T_j\} \tag{3.8}$$

e) Número de trabalhos atrasados:

$$N_T = \frac{1}{n}\sum_{j=1}^{n}\delta(T_j)\,, \tag{3.9}$$

onde $\delta(x) = 1$ se $x > 0$ e $\delta(x) = 0$ se $x \leq 0$.

f) *Makespan* (supondo que $r_j = 0, \ \forall j$):

$$C_{\max} = \mathrm{Max}\,(C_1, \ldots, C_n) \tag{3.10}$$

Cada uma das medidas de desempenho é uma função do conjunto de datas de conclusão do trabalho, então a forma geral é sempre

$$Z = f(C_1, C_2, \ldots, C_n) \tag{3.11}$$

3.2.1. Medidas de Desempenho Regulares

As medidas de desempenho apresentadas pertencem a uma importante classe, conhecidas como medidas de desempenho regulares. Uma medida de desempenho Z é regular se:

(a) O objetivo do sequenciamento é minimizar Z, e

(b) Z pode aumentar somente se pelo menos um dos datas de conclusão no sequenciamento aumentar.

Mais formalmente, supondo que $Z = f(C_1, C_2, ..., C_n)$ é o valor da medida que caracteriza a sequência S e que $Z' = f(C'_1, C'_2, ..., C'_n)$ representa o valor da mesma medida de uma sequência diferente S'. Então Z é regular se a seguinte condição for satisfeita:

$$Z' > Z \text{ implica que } C'_j > C_j \text{ para algum trabalho } j$$

As medidas globais introduzidas acima são medidas regulares, que são praticamente todas consideradas como critérios importantes para o sequenciamento.

Os critérios de otimização regulares mais comumente usados dentre as medidas de desempenho enumeradas são: minimizar o instante de término da última operação processada (*makespan*); minimizar o atraso máximo; minimizar o tempo médio de fluxo; entre outros. (GOMES JÚNIOR, 2007)

Os primeiros modelos de programação da produção utilizavam em sua maioria a minimização do *makespan* como critério de decisão (PINEDO, 2008). Com o passar dos anos, novos critérios passaram a ser explorados, tendo em vista a crescente importância para o mercado de fatores ligados à determinação e ao cumprimento de datas de entrega (KIM, 1995). Para acompanhar essa tendência, foram elaborados modelos cujos critérios de decisão relacionavam-se com o atraso, como por exemplo, minimizar o número de operações atrasadas, o somatório dos atrasos de todas as operações da programação ou minimizar o atraso médio de todas as operações.

Estes critérios, em particular, levam em consideração as datas prometidas, que, no entanto, ignoram a consequência dos trabalhos que são finalizados antes destas datas. Porém, a ênfase tem mudado com o interesse em produções baseadas na filosofia JIT, que adere à noção de que antecipação, assim como atraso da produção, devem ser desencorajados. Em um ambiente de programação JIT, os produtos que são finalizados antecipadamente devem ser mantidos em estoque até a sua data de prometida, acarretando em custos de estoque que devem ser computados, enquanto produtos que são finalizados tardiamente podem incorrer em penalidades contratuais, em perda de credibilidade da empresa e consequentemente, na perda de clientes. (BAKER e SCUDDER, 1990)

Dessa forma, são desenvolvidas as medidas de desempenho não regulares.

3.2.2. Medidas de Desempenho Não Regulares

Baker e Scudder (1990) apresentam uma revisão sobre os problemas de programação que adotam medidores de desempenho não regulares. Neste trabalho, pode claramente ser visto que tais problemas diferenciam-se basicamente em relação a duas características: a data de prometida e os custos de antecipação e atraso, como será visto mais a frente neste trabalho.

Em algumas formulações do problema do sequenciamento com penalidades por atraso e antecipação da produção, a data prometida é conhecida a priori, enquanto que em outros o problema é otimizar a data prometida e a sequência de operações simultaneamente. Uma formulação mais geral adota datas prometidas distintas.

Determinadas formulações adotam datas prometidas comuns a todos os trabalhos (quando todos os trabalhos devem ser finalizados em uma mesma data, por exemplo, para permitir a montagem do produto final). Uma formulação mais geral adota datas distintas de entrega (quando cada trabalho deve ser finalizado em uma data específica).

Em outras formulações, adota-se uma janela de entrega ao invés de uma data prometida fixa, ou seja, os produtos que forem finalizados dentro desta janela de tempo não terão penalidades por antecipação nem por atraso da produção.

Baker e Scudder (1990) classificam os problemas com medidores de desempenho não regulares, em relação aos custos unitários de antecipação e atraso, em:

- Problemas com custos dependentes: quando as penalidades por antecipação e atraso são dependentes das operações, assumindo valores diferentes para cada operação.
- Problemas com custos iguais: quando as penalidades por antecipação são iguais às penalidades por atraso.
- Problemas com custos diferentes: quando as penalidades por antecipação são diferentes das penalidades por atraso.

Esta dissertação adota como critério de otimização a minimização dos custos distintos de antecipação e atraso da produção com datas prometidas também distintas e custos dependentes das operações.

3.3. Problemas Sem Datas Prometidas

Algumas vezes os custos associados às decisões de sequenciamento envolvem serviços aos clientes, através do gasto despendido no sistema e o objetivo de sequenciamento é fazer as entregas o mais brevemente possível. Em outras situações, o custo envolve investimentos nos recursos do sistema, como refletidos pelo comportamento dos estoques em processo e o objetivo de sequenciamento é manter baixos os níveis de estoque. A íntima relação entre esses dois objetivos pode ser ilustrada no modelo básico de uma única máquina.

O tempo gasto por um trabalho no sistema foi definido como seu tempo de fluxo e o objetivo de efetuar as entregas o mais brevemente possível pode ser interpretado como a minimização do tempo de fluxo. Da mesma forma, o objetivo de minimizar o nível de estoque pode ser interpretado como minimizar o número de trabalhos no sistema. Seja J(t) o número de trabalhos no sistema no tempo *t* e, para um dado intervalo [a, b], define-se que o número médio de trabalhos no sistema como sendo $\overline{\mathrm{J}}$, onde

$$\overline{J} = \frac{1}{b-a}\int_a^b J(t) \quad dt \tag{3.12}$$

Em outras palavras, $\overline{\mathrm{J}}$ é a média de tempo da função J(t). Para o modelo básico de uma única máquina, o comportamento de J(t) é fácil de ser visualizado. No tempo zero, existem *n* trabalhos disponíveis do sistema, J(0) = *n*. Não existe mudança em J(t) até que o primeiro trabalho seja terminado, o que ocorre no tempo $F_{[1]} = t_{[1]}$. Então J(t) passa a ser *n-1* e permanece com esse valor até o término do segundo trabalho, o que ocorre no tempo $F_{[2]} = t_{[1]} + t_{[2]}$. Prosseguindo com o raciocínio, é fácil ver que J(t) é uma função degrau decrescente ao longo do intervalo $[0, t_{[1]} + t_{[2]} + ... + t_{[n]}]$, como mostrado na Figura 6. Nota-se também que $F_{max} = t_{[1]} + t_{[2]} + ... + t_{[n]}$ e é independente da sequência que os trabalhos são processados. Para o intervalo $[0, F_{max}]$ tem-se que:

$$\bar{J} = \frac{1}{F_{max}} \{nt_{[1]} + (n\text{-}1)t_{[2]} + ... + 2\, t_{[n\text{-}1]} + t_{[n]}\} \tag{3.13}$$

A soma entre colchetes é a área A sob a função J(t), expressa como a soma das faixas verticais de cada $t_{[n]}$. Portanto:

$$\bar{J} = \frac{A}{F_{max}} \tag{3.14}$$

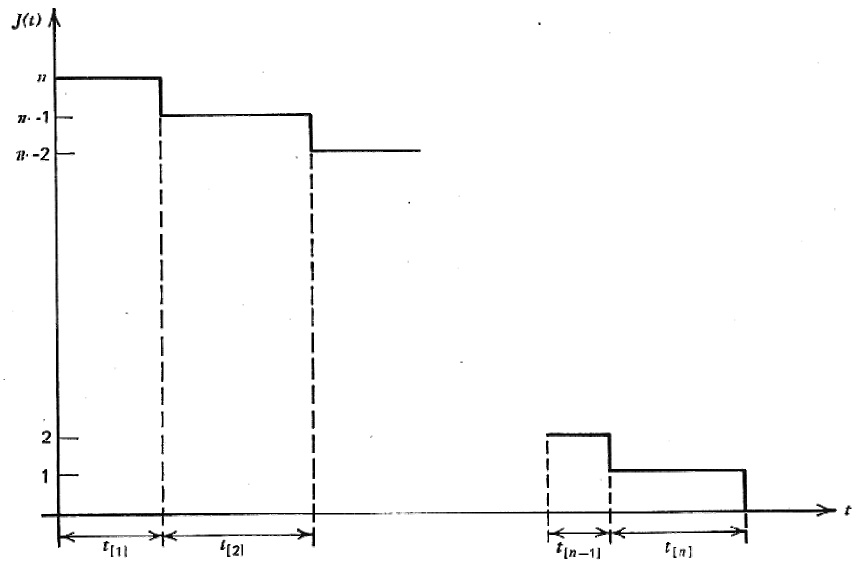

Figura 6 – Gráfico da Função J(t)

Fonte: Baker (1974)

Lembrando a média do tempo de fluxo, $\bar{F}$, definida em (3.5):

$$\bar{F} = \frac{1}{n}\sum_{j=1}^{n} F_j = \frac{1}{n}(F_{[1]} + F_{[2]} + ... + F_{[n]})$$

Nota-se que a soma entre parênteses também é a área A, expressa como a soma das faixas horizontais, como Figura 7. Portanto:

$$\bar{F} = \frac{A}{n} \tag{3.15}$$

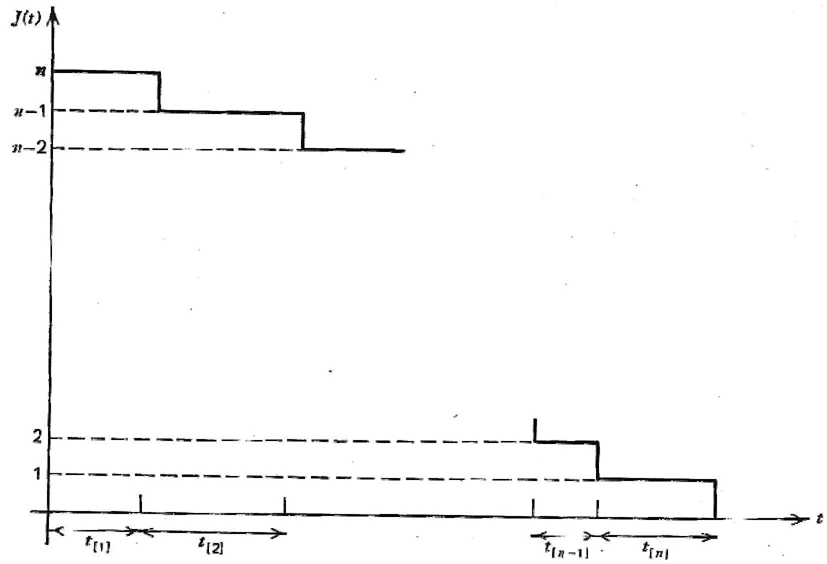

Figura 7 – Outra alternativa da Função J(t)
Fonte: Adaptado de Baker (1974)

Dessa forma A= $\bar{F}$ n = $\bar{J}$ F_{max}

Em outras palavras, desde que *n* e F_{max} sejam constantes, $\bar{J}$ é diretamente proporcional a $\bar{F}$. Isso significa, em particular, que a sequência de trabalhos que minimiza $\bar{F}$ (média das entregas) também minimizará $\bar{J}$ (média do estoque em processo). Quer do ponto de vista de otimizar os serviços dos clientes ou de minimizar os níveis de estoque em processo, o problema é o mesmo: achar a sequência que minimiza o tempo de fluxo médio ($\bar{F}$).

A relação entre o tempo de fluxo e o estoque estende-se bem além do problema do sequenciamento em uma única máquina. Ela surge no ambiente dinâmico (aonde os trabalhos chegam ao longo do tempo), em modelos com horizonte infinito, em sistemas probabilísticos e em situações onde o custo de estoque pode variar entre trabalhos. Muitos dos trabalhos teóricos em sequenciamento têm sido direcionados para problemas de tempo de fluxo e suas generalizações. O que pode em princípio parecer uma ênfase indevida no critério de entrega não é realmente tão restritiva, à luz dessa relação entre Tempo de Fluxo e estoque, porque significa que o tempo de fluxo realmente engloba um aspecto mais amplo nos custos relacionados ao sequenciamento.

3.3.1. Minimizando o Tempo de Fluxo Médio

Uma sequência de trabalhos em ordem não decrescente de tempos de processamento é conhecida como sequenciamento de menor tempo de processamento (*Shortest Processing Time* – SPT)

De acordo com Baker (1974), o tempo de fluxo médio, $\overline{F}$, é minimizado em uma sequência SPT.

3.3.2. Minimizando a Média Ponderada do Tempo de Fluxo Médio

Em algumas variações do problema do tempo de fluxo médio, os trabalhos não têm a mesma importância quando comparados entre si. Uma maneira de levar em conta essa característica é atribuir um fator de ponderação, w_j, para cada trabalho e incorporar os fatores de ponderação na medida de desempenho. A versão ponderada da média do tempo de fluxo é a média ponderada do tempo de fluxo, definida como

$$\overline{F}_w = \frac{\sum_{j=1}^{n} w_j F_j}{\sum_{j=1}^{n} w_j} \tag{3.16}$$

3.4. Problemas com Datas Prometidas

3.4.1. Critério de Atraso

Conforme definido na subseção 3.1, a defasagem, L_j, é a diferença entre o tempo de conclusão, C_j, e a data prometida, d_j. Baker (1974) prova que a minimização da média das defasagens, $\overline{L}$, é alcançada por uma sequência SPT, como apresentado abaixo:

$$\overline{L} = \frac{1}{n}\sum_{j=1}^{n} L_j = \frac{1}{n}\sum_{j=1}^{n}\left(C_j - d_j\right) = \frac{1}{n}\sum_{j=1}^{n}\left(F_j - d_j\right) = \frac{1}{n}\sum_{j=1}^{n} F_j - \frac{1}{n}\sum_{j=1}^{n} d_j = \overline{F} - \overline{d} \qquad (3.17)$$

Onde $\overline{d}$ é a média do conjunto de datas prometidas e, consequentemente, é um valor constante. Como $\overline{L}$ difere de $\overline{F}$ por uma constante que é independente da sequência, a sequência que minimiza $\overline{L}$ será a sequência que minimiza $\overline{F}$, que é dada por uma sequência com menor tempo de processamento (SPT).

O fato interessante é que uma sequência que não leva em consideração a data de prometida (uma sequência SPT) é ideal para um critério voltado para a data de prometida.

Em contrapartida, sequenciando os trabalhos por sua data de prometida mais próxima (*Earliest Due Date* – EDD) não garante que a média das defasagens, $\overline{L}$, será minimizada, porém uma sequência EDD minimiza a máxima defasagem, L_{max}, na sequência, como será mostrado a seguir:

Considere uma sequência *S* que não esteja organizada por sua data de prometida mais próxima (EDD). Isso significa que em algum ponto de S deve existir um par de trabalhos adjacentes, *i* e *j*, com *j* após *i*, tal que $d_i > d_j$, conforme mostrado na Figura 8.

Seja uma nova sequência, S', onde os trabalhos *i* e *j* estejam com suas posições trocadas no sequenciamento e todos os outros trabalhos são concluídos no mesmo instante que em S.

Considere também que B denota o conjunto de trabalhos que precedem os trabalhos *i* e *j* em ambas as sequências, A denota o conjunto de trabalhos que sucedem os trabalhos *i* e *j*, em *S* e em S', e t_B é o ponto em que o trabalho *i* é iniciado em S e o ponto em que o trabalho *j* é iniciado em S'.

Sendo assim:

$$L_i(S) = t_B + t_i - d_i \qquad (3.18)$$

Ou seja, a defasagem do trabalho *i* na sequência *S* é igual a soma do tempo correspondente ao início do trabalho *i* com o seu Tempo de Processamento menos a data de prometida. Analogamente, a defasagem do trabalho *j* é dado pela eq. (3.18):

$$L_j(S) = t_B + t_i + t_j - d_j \tag{3.19}$$

De forma similar para a sequência S', tem-se:

$$L_j(S') = t_B + t_j - d_j \tag{3.20}$$

$$L_i(S') = t_B + t_j + t_i - d_i \tag{3.21}$$

De (3.19) e (3.21) e com $d_i > d_j$, tem-se que:

$$L_j(S) > L_i(S') \tag{3.22}$$

De (3.19) e (3.20), tem-se que:

$$L_j(S) > L_j(S') \tag{3.23}$$

De (3.22) e (3.23), tem-se que:

$$L_j(S) > \max\{L_j(S'), L_i(S')\} \tag{3.24}$$

Seja $L = \max\ \{L_k \mid k \in A \text{ ou } k \in B\}$, notando que L é o mesmo para *S* ou S'. Então tem-se que:

$$L_{max}(S) = \max\ \{L, L_i(S'), L_j(S')\} = L_{max}(S') \tag{3.25}$$

Em outras palavras, a troca de posição entre os trabalhos *i* e *j* não aumenta o valor da defasagem máxima, L_{max}, e pode na verdade reduzi-lo (melhorá-lo).

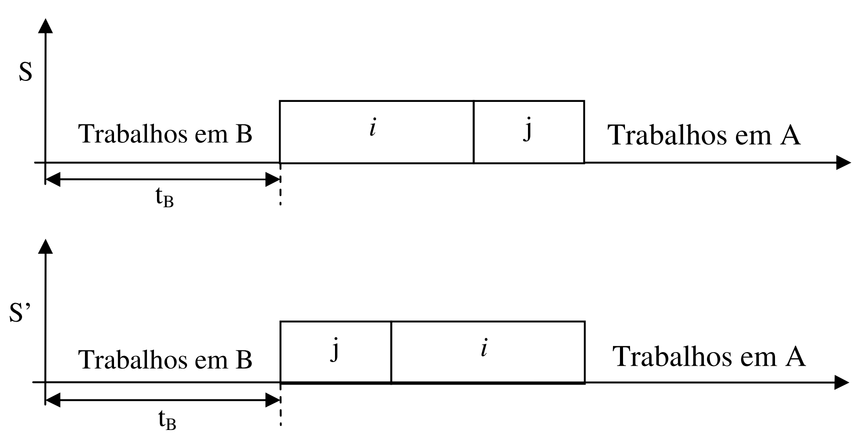

Figura 8 – Troca de pares de trabalhos adjacentes
Fonte: Adaptado de Baker (1974)

3.4.2. Minimizando o número de trabalhos Atrasados

De acordo com Baker (1970), se a sequência EDD resultar em nenhum trabalho atrasado ou resultar em exatamente um trabalho atrasado, então essa sequência é ótima para N_T. No entanto, se existir mais que um trabalho atrasado, a sequência EDD pode não ser ótima.

3.4.3. Minimizando a Média dos Trabalhos Atrasados

O objetivo de definir as datas de entrega dos trabalhos é um dos critérios encontrados com mais frequência em problemas práticos. Embora encontrar as datas prometidas seja só um objetivo qualitativo, ele normalmente implica que as penalidades por dependência de tempo são avaliadas nos trabalhos atrasados, mas nenhum benefício será alcançado quando se finaliza os trabalhos antes da data prometida. Uma quantificação natural do objetivo de se encontrar datas prometidas, portanto, envolve a medida de atraso e um problema fundamental de sequenciamento é a minimização da média de trabalhos atrasados. A dificuldade em lidar com a média de trabalhos atrasados e também com a maioria das outras medidas de desempenho baseadas no atraso é devido ao fato de que o atraso não é uma função linear do tempo de conclusão. Isso significa que, para encontrar soluções ótimas para tais problemas, é necessário contar com conceitos de otimização combinatória. Além disso, devido à complexidade dos métodos combinatórios, existe uma tendência a se ter mais atenção em técnicas eficientes de solução, porém subótimas.

A primeira aproximação lógica para o problema $\overline{T}$ é usar as propriedades de trocas de pares adjacentes e do procedimento de Wilkerson-Irwin (1971), cujo algoritmo utiliza uma regra de decisão em um procedimento heurístico que emprega comparações entre pares de trabalhos na construção da sequência.