

## 4. O *Makespan* com Tempos de Preparação de Máquina Dependentes da Sequência

Para problemas de sequenciamento em um única máquina com todos os  $r_j = 0$  e tempos de preparação de máquinas (tempos de *set up*) independentes da sequência, o *makespan* é igual à soma dos tempos de processamento ( $t_j$ ).

No entanto, quando os tempos de *set up* são dependentes da sequência, o *makespan* depende do sequenciamento.

Sejam dois parâmetros ( $a_j$  e  $b_j$ ) associados ao trabalho  $j$  e o parâmetro  $s_{jk} = |a_k - b_j|$ . Essa estrutura de tempo de *set up* pode ser descrita como segue: após a conclusão do trabalho  $j$ , a máquina é posta em estado  $b_j$  e para ser capaz de iniciar o trabalho  $k$ , a máquina deve retornar ao estado  $a_k$ . O tempo total de preparação necessário para passar a máquina do estado  $b_j$  para o estado  $a_k$  é proporcional à diferença absoluta entre os dois estados. Essa variável denominada “estado” poderia ser, por exemplo, temperatura (no caso de um forno) ou a medida de algum outro ajuste da máquina. Além disso, assume-se que, no tempo zero, o estado é  $b_0$  e que, após completar o último trabalho, a máquina tem de ser deixada no estado  $a_0$  (isso implica que um tempo adicional de limpeza é necessário após o último trabalho ser completado).

E, na prática, os tempos de preparação muitas vezes têm uma estrutura especial. Algumas estruturas de tempo de *set up* em particular permitem a resolução do algoritmo em tempo polinomial. (PINEDO, 2008)

### 4.1. O Problema do Caixeiro Viajante

A descrição de um algoritmo com uma estrutura de tempo de preparação especial que pode ser resolvida em tempo polinomial é melhor compreendido no contexto do Problema do Caixeiro Viajante (TSP – *Travelling Salesman Problem*).

Os algoritmos, portanto, apresentados no contexto de um TSP consideram  $n + 1$  cidades; Seja a cidade adicional chamada de cidade 0 com parâmetros  $a_0$  e  $b_0$ . Sem perda de generalidade, pode-se supor que  $b_0 \leq b_1 \leq \dots \leq b_n$ . O caixeiro-viajante saindo da cidade  $j$  para a cidade  $k$  (ou, equivalentemente, o trabalho  $k$  é o trabalho seguinte ao trabalho  $j$ ) é denotado por  $k = \varnothing(j)$ . A sequência de cidades em uma viagem é denotada por  $\Phi$ , que é um vetor que mapeia cada elemento do conjunto  $\{0, 1, 2, \dots, n\}$  em um único elemento de  $\{0, 1, 2, \dots, n\}$  pelas relações  $k = \varnothing(j)$ , indicando que o caixeiro-viajante visita a cidade  $k$  após da cidade  $j$  (ou, equivalentemente, o trabalho  $k$  é a tarefa seguinte à tarefa  $j$ ). Tais mapeamentos são chamados mapeamentos de permutação. Note que nem todos os mapeamentos de permutação possíveis do conjunto  $\{0, 1, 2, \dots, n\}$  constituem caminhos do TSP viáveis. Por exemplo,  $\{0, 1, 2, 3\}$  mapeadas em  $\{2, 3, 1, 0\}$  representa um TSP viável, como mostrado na Figura 9-a. Nesse caso viável, o Mapeamento de Permutação é  $\Phi = \{2 = \varnothing(0); 3 = \varnothing(1); 1 = \varnothing(2); 0 = \varnothing(3)\}$ , ou seja,  $0 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 0$ . No entanto,  $\{0, 1, 2, 3\}$  mapeadas em  $\{2, 1, 3, 0\}$  não representa um caminho viável, já que  $\Phi = \{2 = \varnothing(0); 1 = \varnothing(1); 3 = \varnothing(2); 0 = \varnothing(3)\}$  e esta estrutura representa duas subviagens disjuntas, a saber: as subviagens  $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$  e a subviagem  $1 \rightarrow 1$ , que consiste de uma única cidade, como apresentado na Figura 9-b. Dessa forma, define-se  $\varnothing(k) = k$  para significar uma viagem redundante, que começa e acaba em  $k$ .

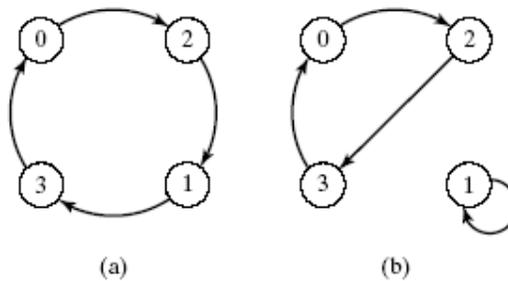


Figura 9 – Mapas de Permutação

Fonte: Adaptado de Pinedo (2008)

(a)  $\{0, 1, 2, 3\} \rightarrow \{2, 3, 1, 0\}$  (b)  $\{0, 1, 2, 3\} \rightarrow \{2, 1, 3, 0\}$

Analogamente como descrito acima para a estrutura de tempo de *set up*, para a estrutura de custos especial de ir da cidade  $j$  para a cidade  $k$  é evidente que esse custo é igual à altura vertical da reta que liga  $b_j$  com  $a_k$ , ou seja é a diferença

absoluta entre os parâmetros  $a_k$  e  $b_j$  referentes às duas cidades, como mostrado na Figura 10.

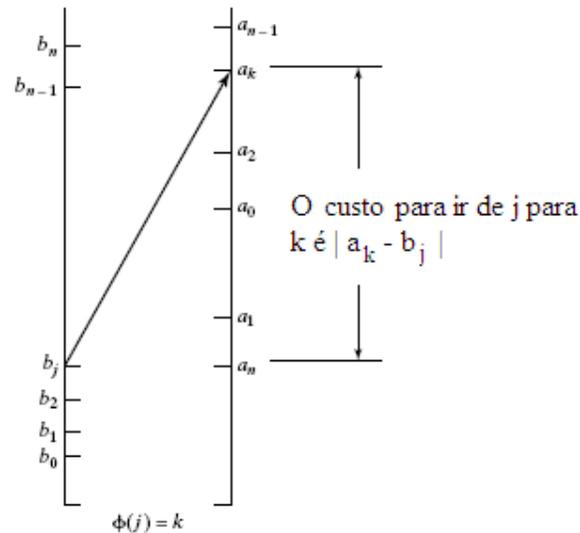


Figura 10 – Custo de ir de  $j$  para  $k$  ( $s_{jk} \rightarrow \emptyset(j) = k$ )

Fonte: Adaptado de Pinedo (2008)

Define-se o custo de uma sub-viagem redundante, ou seja,  $\emptyset(k) = k$ , como a altura vertical de uma reta de  $b_k$  para  $a_k$ .

Assim, qualquer mapeamento de permutação (o que poderá eventualmente ser constituído por sub-viagens) pode ser representado como um conjunto de retas que ligam  $b_j$ ,  $j = 0, \dots, n$ , para  $a_k$ ,  $k = 0, \dots, n$ , e os custos associados com esse mapeamento é simplesmente a soma das alturas verticais das  $n+1$  retas.

Definindo uma troca  $I(k, j)$  como o procedimento que, quando aplicado a um mapeamento de permutação  $\Phi$ , produz outro mapeamento  $\Phi'$ , afetando apenas as atribuições de  $j$  e  $k$  e deixando as outras inalteradas, mais precisamente, a nova atribuição  $\Phi' = \Phi I(j, k)$  é definida como:

$$\emptyset'(k) = \emptyset(j),$$

$$\emptyset'(j) = \emptyset(k),$$

e

$$\emptyset'(l) = \emptyset(l), l \in \{0, 1, \dots, n\} - \{j, k\}$$

para todos  $l$  não iguais a  $j$  ou  $k$ . Seja por exemplo o seguinte Mapeamento de Permutação  $\Phi = \{2 = \emptyset(0); 4 = \emptyset(2); 1 = \emptyset(4); 5 = \emptyset(1); 3 = \emptyset(5); 0 = \emptyset(3)\}$ ,

ou seja,  $0 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 3 \rightarrow 0$ . A troca  $I(j, k) = I(4,5)$  promoverá um novo Mapa de Permutação tal que:  $\varnothing'(k) = \varnothing(j)$ ,  $\varnothing'(j) = \varnothing(k)$  e  $\varnothing'(l) = \varnothing(l)$ .

$$\varnothing'(5) = \varnothing(4) = 1$$

$$\varnothing'(4) = \varnothing(5) = 3$$

$$\varnothing'(0) = \varnothing(0) = 2$$

$$\varnothing'(1) = \varnothing(1) = 5$$

$$\varnothing'(2) = \varnothing(2) = 4$$

$$\varnothing'(3) = \varnothing(3) = 0$$

O novo Mapa de Permutação  $\Phi'$  é composto por duas sub-viagens  $0 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 0$  e  $1 \rightarrow 5 \rightarrow 1$ . Observa-se nesse novo Mapa de Permutação que somente as posições  $j$  e  $k$  foram alteradas. Essa transformação também pode ser denotada por  $\varnothing'(j) = \varnothing(j) I(j, k)$ . Nota-se que isso não é equivalente a uma troca de pares adjacentes de uma sequência, uma vez que um mapa de permutação  $\Phi$  nem sempre representa uma sequência inicial (uma viagem do TSP viável). Mais intuitivamente, isso representa apenas uma troca das retas saindo de  $b_j$  e  $b_k$  e deixando todas as outras retas inalteradas. Em particular, se essas retas cruzam umas com as outras antes, elas não se cruzarão agora e vice-versa. O que tal troca implica, em termos da viagem atual e de subviagens, é bastante surpreendente. Conforme apresentado no exemplo acima, a troca  $I(j, k)$  tem o efeito de criar duas subviagens a partir de uma, se  $j$  e  $k$  pertencem à mesma subviagem em  $\Phi$ . Por outro lado, quando uma troca  $I(j, k)$  é aplicada a duas subviagens diferentes  $j$  e  $k$  pertencentes a  $\Phi$ , elas combinam-se em uma só subviagem.

A definição a seguir quantifica o custo total da troca  $I(j, k)$ , aplicado à sequência  $\Phi$ . O custo da troca é denotado por  $c_{\Phi} I(j,k)$ . Na definição, o intervalo dos pares  $[a, b]$  que não estão em uma sequência se refere a um intervalo pertencente à reta dos números reais. Dessa forma, tem-se:

$$\| [a, b] \| = \begin{cases} 2(b-a) & \text{se } b > a \\ 2(a-b) & \text{se } b < a \end{cases} \quad (4.1)$$

Onde  $\| [a, b] \|$  é a norma do intervalo  $[a, b]$ .

Se for aplicada uma troca  $I(j,k)$ , quando as duas setas que não se cruzavam (representadas em linhas cheias na Figura 11), passam a se cruzar (representadas em linhas tracejadas na Figura 11), então o custo da viagem é aumentado e vice-versa. A magnitude desse aumento ou decréscimo é dada por:

$$c_{\Phi} I(j,k) = \left\| [b_j, b_k] \cap [a_{\phi(j)}, a_{\phi(k)}] \right\| \quad (4.2)$$

Onde a mudança no custo  $\left\| [b_j, b_k] \cap [a_{\phi(j)}, a_{\phi(k)}] \right\|$  é igual ao comprimento da sobreposição vertical dos intervalos  $[b_j, b_k]$  e  $[a_{\phi(j)}, a_{\phi(k)}]$ , como mostrado na Figura 11.

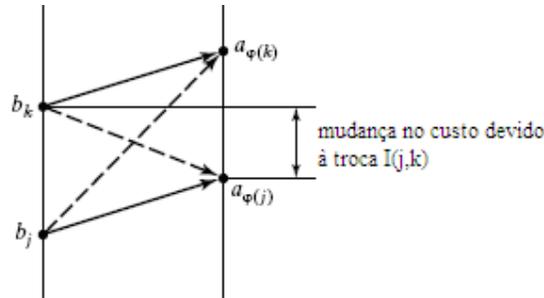


Figura 11 – Mudança do custo devido à troca  $I(j,k)$

Fonte: Adaptado de Pinedo (2008)

Essa definição é importante, pois fornece uma indicação visual para redução de custos quando as setas são descruzadas e ajuda a quantificar a redução de custos em termos de quantidades de sobreposição de determinados intervalos. A partir dessa interpretação visual, conclui-se que um Mapeamento de Permutação ótimo,  $\Phi^*$ , é obtido se  $b_j \leq b_k \rightarrow a_{\phi^*(j)} \leq a_{\phi^*(k)}$ .

Apesar desse Mapeamento de Permutação ótimo não ser necessariamente uma viagem viável, ele fornece um limite inferior para o custo ótimo de qualquer TSP. Esse Mapeamento de Permutação ótimo,  $\Phi^*$ , pode consistir em  $p$  subviagens distintas,  $(Subviagem)_1, (Subviagem)_2, \dots, (Subviagem)_p$ . Como apresentado anteriormente, ao se realizar uma troca  $I(j,k)$ , com  $j$  e  $k$  pertencentes a subviagens distintas, essas subviagens se unem em uma única viagem e o custo aumentará (isso se as retas que não se cruzavam antes da troca agora se cruzem) de  $c_{\Phi^*} I(j,k)$ . É desejável selecionar  $j$  e  $k$  de subviagens diferentes de tal forma que esse custo da união das subviagens  $c_{\Phi^*} I(j,k)$  seja minimizado.

Para determinar essas trocas, em vez de considerar o gráfico direto que representa as subviagens do caixeiro-viajante, considera-se a versão não direcionada de um mesmo gráfico. Neste caso, as subviagens representam ciclos distintos de subviagens redundantes e são simplesmente nós independentes. Para

conectar os elementos disjuntos (ou seja, os ciclos correspondentes às subviagens) e construir um gráfico conectado, arcos adicionais devem ser inseridos neste gráfico indireto.

Os custos dos arcos entre as cidades que pertencem a diferentes subviagens neste gráfico não direcionado são escolhidos para serem iguais ao custo de realizar as trocas correspondentes na viagem do caixeiro-viajante no gráfico direto. O custo desta troca pode ser calculado facilmente por 4.2. Os arcos utilizados para ligar subviagens disjuntas são selecionados de acordo com o Algoritmo Guloso<sup>1</sup>: selecione o arco mais barato que liga duas das subviagens  $p$  no gráfico indireto; selecionar entre os arcos restantes não utilizados, o mais barato arco ligando duas das  $(p - 1)$  subviagens restantes e assim por diante. Uma propriedade dos arcos selecionados diz que o conjunto de arcos que conectam o gráfico indireto com o mínimo custo contém somente arcos que conectam a cidade  $j$  à cidade  $j+1$ .

Essa propriedade pode ser provada, pois os custos dos arcos  $(c_{\Phi^*} I(j,k))$  necessários para conectar os ciclos distintos do gráfico indireto são computados a partir do Mapeamento de Permutação ótimo (obtido se  $b_j \leq b_k \rightarrow a_{\emptyset^*(j)} \leq a_{\emptyset^*(k)}$ ), o qual não possui duas setas se cruzando. Como mostrado em 4.2, o custo para se trocar duas setas não adjacentes é no mínimo igual ao custo de se trocar todas as setas entre elas. Dessa forma, percebe-se que o custo é considerado como a interseção de dois intervalos dados por 4.2. Em particular, se  $k > j+1$ ,

$$c_{\Phi^*} I(j,k) \geq \sum_{i=j}^{k-1} \left\| [b_i, b_{i+1}] \cap [a_{\emptyset(i)}, a_{\emptyset(i+1)}] \right\| \quad (4.3)$$

$$c_{\Phi^*} I(j,k) = \sum_{i=j}^{k-1} c_{\Phi^*} I(i, i+1) \quad (4.4)$$

Então, o arco  $(j,k)$  pode ser substituído pela sequência de arcos  $(i, i+1)$ ,  $i = j, \dots, k-1$  para conectar as duas subviagens para as quais  $j$  e  $k$  pertencem como menor ou mínimo custo.

É importante notar que na construção do gráfico indireto, o custo designado para os arcos que conectam as subviagens foram computados sob a premissa de que as trocas são aplicadas em  $\Phi^*$  onde as setas não se cruzam. No entanto, como

---

<sup>1</sup> *Greedy Algorithm* ou Algoritmos Gulosos constroem a solução parte por parte, escolhendo sempre o próximo movimento que lhe ofereça o benefício mais óbvio e imediato.

as trocas são executadas para conectar as subviagens, esta condição não permanece mais válida. No entanto, se a ordem em que as trocas são executadas for determinada com atenção, o custo das trocas não será afetado pelas trocas anteriores, entretanto o exemplo seguinte mostra que a sequência em que as trocas são executadas podem ter um impacto no custo final.

Considere as situações das Figuras 12-a e Figuras 13-a, onde o custo da troca  $I(2,3)$ ,  $c_{\Phi} I(2,3)$ , é igual a 1 e o custo da troca  $I(1,2)$ ,  $c_{\Phi} I(1,2)$ , também é igual a 1. Como mostrado na Figura 12, se a troca  $I(2,3)$  é executada primeiro que a troca  $I(1,2)$ , a sobreposição dos intervalos que determina o custo de troca das mudanças de posição permanece inalterada. Entretanto, se a sequência das trocas é invertida, ou seja, a troca  $I(1,2)$  é executada antes da troca  $I(2,3)$ , então os custos mudam: o custo da primeira troca permanece inalterado, mas o custo da segunda troca,  $c_{\Phi} I(2,3)$  agora se torna 2 em vez de 1, como apresentado na Figura 13.

O ponto chave aqui é que as duas trocas consideradas têm uma seta em comum, isto é,  $b_2 \rightarrow a_{\emptyset(2)}$ . Esta seta aponta para “cima” e qualquer troca que permaneça apontando para cima não afetará o custo da troca abaixo dela, uma vez que a sobreposição dos intervalos não muda.

Esse exemplo sugere que se uma sequência de trocas precisa ser executada, as trocas cujos arcos menores apontam para cima podem ser executados começando pelo topo em direção para baixo sem mudar os custos das trocas abaixo deles. Uma analogia similar pode ser deduzida com respeito às trocas cujas setas mais baixas apontam para “baixo”.

Para tornar essa noção de “cima” e “baixo” mais rigorosa, classifica-se os nós em dois tipos. Um nó é dito do tipo I se  $a_j \leq b_{\emptyset(j)}$ , isto é, as setas apontam para cima. E um nó é do tipo II se  $a_j > b_{\emptyset(j)}$ . Uma troca é do tipo I se seu menor nó é do tipo I e é do tipo II se seu menor nó é do tipo II. Pela razão anterior, é fácil deduzir que, se as trocas  $I(j,j+1)$  do tipo I são executadas em ordem decrescentes dos índices dos nós, seguidas pelas trocas do tipo II em ordem crescentes dos nós, uma única viagem é obtida sem nenhuma mudança  $c_{\Phi^*} I(j,j+1)$  envolvidas nas trocas.

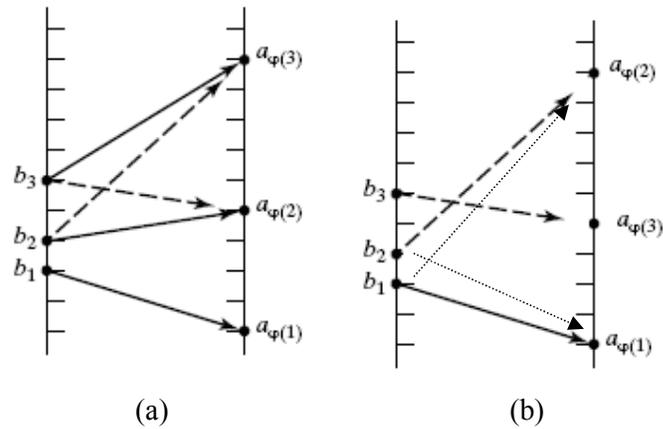


Figura 12 – Troca  $I(2,3)$  antes da troca  $I(1,2)$   
 Fonte: Adaptado de Pinedo (2008)

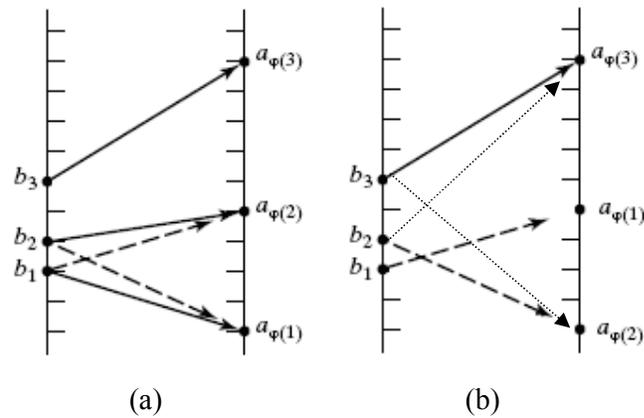


Figura 13 – Troca  $I(1,2)$  antes da troca  $I(2,3)$   
 Fonte: Adaptado de Pinedo (2008)

#### 4.2. Algoritmo para um TSP

O algoritmo para se achar a viagem ótima num TSP pode ser descrito em sete passos, conforme Quadro 1:

Quadro 1 – Algoritmo para um TSP

**Passo 1:**

Organize os parâmetros  $b_j$  em ordem de tamanho e renumere os trabalhos tais que:

$$b_0 \leq b_1 \leq \dots \leq b_n$$

O mapa de permutação  $\Phi^*$  é definido por  $\varnothing^*(j) = k$ ,

Onde  $k$  é tal que  $a_k$  é o menor  $(j+1)$ -ésimo elemento entre os parâmetros  $a_j$

**Passo 2:**

A partir de um gráfico indireto com  $n+1$  nós e arcos indiretos  $A_{j, \varnothing^*(j)}$  conectando o  $j$ -ésimo nó e o  $\varnothing^*(j)$ -ésimo nó.

Se o gráfico possuir somente um elemento, então PARE.

Caso contrário, siga para o passo 3

**Passo 3:**

Associe os custos de troca  $c_\Phi * I(j, j+1)$  para  $j = 0, \dots, n-1$ :

$$c_\Phi * I(j, j+1) = 2\max(\min(b_{j+1}, a_{\varnothing^*(j+1)}) - \max(b_j, a_{\varnothing^*(j)}), 0)$$

**Passo 4:**

Selecione o menor valor de  $c_\Phi * I(j, j+1)$  tal que  $j$  é um componente e  $j+1$  é outro (desfaça as conexões arbitrariamente)

Insira o arco indireto  $A_{j, j+1}$  no gráfico e repita este passo até que todos os componentes do gráfico indireto sejam conectados.

**Passo 5:**

Divida os arcos selecionados no passo 4 em dois grupos:

grupo 1: composto pelos arcos  $A_{j, j+1}$  para os quais  $a_{\varnothing^*(j)} \geq b_j$

grupo 2: composto pelos arcos restantes.

**Passo 6:**

Ache o maior índice  $j_1$  tal que  $A_{j_1, j_1+1}$  esteja no grupo 1;

Ache o segundo maior índice  $j_2$  e assim por diante;

Ache o menor índice  $k_1$  tal que  $A_{k_1, k_1+1}$  esteja no grupo 2;

Ache o segundo menor índice  $k_2$  e assim por diante;

**Passo 7:**

O percurso ótimo  $\Phi^{**}$  é formado aplicando-se a seguinte sequência de trocas à permutação  $\Phi^*$ :

$$\Phi^{**} = \Phi^* I(j_1, j_1+1) I(j_2, j_2+1) \dots I(j_p, j_p+1) I(k_1, k_1+1) I(k_2, k_2+1) \dots I(k_m, k_m+1).$$

Fonte: Adaptado de Pinedo (2008)

O custo total do percurso resultante pode ser visto como a composição de duas componentes. Uma das componentes é o custo do mapeamento de permutação irrestrito  $\Phi^*$  antes da realização das trocas. O outro componente é o custo adicional causado pelas trocas.

Este algoritmo realmente leva a um percurso ideal e pode ser mostrado em duas etapas. Primeiramente, um limite inferior é estabelecido para o custo total de um mapeamento de permutação arbitrário. Em segundo lugar, é mostrado que este limite inferior, no caso do mapeamento de permutação, representa uma viagem

real, é maior ou igual ao custo total da viagem construída no algoritmo. Estas duas etapas, então, provam a otimalidade do percurso do algoritmo.

Como exemplo, considere sete cidades com os seguintes parâmetros, conforme mostrado na Tabela 1:

Tabela 1 – Parâmetros iniciais para o TSP

Cidades	0	1	2	3	4	5	6
$b_j$	1	15	26	40	3	19	31
$a_j$	7	16	22	18	4	45	34

Fonte: Adaptado de Pinedo (2008)

### Passo 1.

Reordenando as cidades tais que um percurso  $b_j \leq b_{j+1}$  (ordem crescente de  $b_j$ ) resulte na ordem e no mapeamento  $\varnothing^*(j) = k$ , com  $a_k \leq a_{k+1}$  (ordem crescente de  $a_k$ ), como apresentado na Tabela 2, tem-se:

Tabela 2 – Parâmetros reordenados para o mapeamento  $\varnothing^*(j)$

Cidades	0	1	2	3	4	5	6
$b_j$	1	3	15	19	26	31	40
$a_j$	7	4	16	45	22	34	18
$a_k = a_{\varnothing^*(j)}$	4	7	16	18	22	34	45
$k = \varnothing^*(j)$	1	0	2	6	4	5	3

Fonte: Adaptado de Pinedo (2008)

### Passo 2.

Construir o gráfico indireto com  $j$  conectados a  $\varnothing^*(j)$ . Os nós 0 e 1 devem ser ligados um ao outro; os nós 3 e 6 têm de ser ligados um com o outro; os nós 2, 4 e 5 são independentes (cada um destes três nós está ligado a si próprio). Ou seja, nesse primeiro Mapeamento de Permutação,  $\Phi^*$ , as cidades estão divididas em duas subviagens:  $0 \rightarrow 1 \rightarrow 0$  e  $3 \rightarrow 6 \rightarrow 3$ , além das três subviagens redundantes:  $2 \rightarrow 2$ ,  $4 \rightarrow 4$  e  $5 \rightarrow 5$ , ou seja,  $\Phi^* = \{\varnothing(0) = 1, \varnothing(1) = 0, \varnothing(2) = 2, \varnothing(3) = 6, \varnothing(4) = 4, \varnothing(5) = 5, \varnothing(6) = 3\}$

O custo de  $\Phi^*$  é:  $\sum_{j=1}^n |b_j - a_k| = 3 + 4 + 1 + 1 + 4 + 3 + 5 = 21$ . No entanto, esse

Mapeamento de Permutação não representa uma trajetória viável. Uma trajetória viável aleatória seria, por exemplo,  $0 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 0$ , cujo custo é  $3 + 19 + 8 + 6 + 15 + 30 + 12 = 93$ .

**Passo 3.** Cálculo dos custos de  $c_{\Phi^*I}(j, j + 1)$ :

$$c_{\Phi^*I}(j, j + 1) = 2\max[\min(b_{j+1}, a_{\emptyset^*(j+1)}) - \max(b_j, a_{\emptyset^*(j)}), 0]$$

$$c_{\Phi^*I}(0, 1) = 2\max[\min(3, 7) - \max(1, 4), 0] = 2\max(3 - 4, 0) = 2\max(-1, 0) = 0$$

$$c_{\Phi^*I}(1, 2) = 2(15 - 7) = 16$$

$$c_{\Phi^*I}(2, 3) = 2(18 - 16) = 4$$

$$c_{\Phi^*I}(3, 4) = 2(22 - 19) = 6$$

$$c_{\Phi^*I}(4, 5) = 2(31 - 26) = 10$$

$$c_{\Phi^*I}(5, 6) = 2(40 - 34) = 12$$

**Passo 4.**

Os arcos indiretos  $A_{1,2}$ ,  $A_{2,3}$ ,  $A_{3,4}$  e  $A_{4,5}$  são inseridos no gráfico.

**Passo 5.**

Os quatro arcos devem ser divididos em dois grupos. Para determinar os grupos, cada  $b_j$  deve ser comparado ao seu  $a_{\emptyset^*(j)}$  correspondente, como indicado na Tabela 3:

Tabela 3 – Divisão dos grupos

Arcos	$b_j$	$a_{\emptyset^*(j)}$	Grupo
$A_{1,2}$	$b_1 = 3$	$a_{\emptyset^*(1)} = a_0 = 7$	1
$A_{2,3}$	$b_2 = 15$	$a_{\emptyset^*(2)} = a_2 = 16$	1
$A_{3,4}$	$b_3 = 19$	$a_{\emptyset^*(3)} = a_6 = 18$	2
$A_{4,5}$	$b_4 = 26$	$a_{\emptyset^*(4)} = a_4 = 22$	2

Fonte: Adaptado de Pinedo (2008)

**Passo 6.**

$$j_1 = 2, j_2 = 1, k_1 = 3 \text{ e } k_2 = 4.$$

**Passo 7**

O percurso ótimo é obtido após as seguintes trocas.  $\Phi^{**} = \Phi^* I(j_1, j_1+1) I(j_2, j_2+1) \dots I(j_p, j_p+1) I(k_1, k_1+1) I(k_2, k_2+1) \dots I(k_m, k_m+1)$ . Dessa forma,  $\Phi^{**} = \Phi^* \cdot I(2, 3) I(1, 2) I(3, 4) I(4, 5)$ .

Como foi visto,  $\Phi^* = \{\emptyset(0) = 1, \emptyset(1) = 0, \emptyset(2) = 2, \emptyset(3) = 6, \emptyset(4) = 4, \emptyset(5) = 5, \emptyset(6) = 3\}$ , ou seja,  $0 \rightarrow 1 \rightarrow 0, 3 \rightarrow 6 \rightarrow 3, 2 \rightarrow 2, 4 \rightarrow 4 \text{ e } 5 \rightarrow 5$ .

Aplicando inicialmente a troca  $I(2, 3)$  à  $\Phi^*$ , teremos,  $\Phi^{*1} = \{\emptyset'(0) = 1, \emptyset'(1) = 0, \emptyset'(2) = \emptyset(3) = 6, \emptyset'(3) = \emptyset(2) = 2, \emptyset'(4) = 4, \emptyset'(5) = 5, \emptyset'(6) = 3\}$ , ou seja

$$0 \rightarrow 1 \rightarrow 0, 2 \rightarrow 6 \rightarrow 3 \rightarrow 2, 4 \rightarrow 4, 5 \rightarrow 5$$

Aplicando agora  $I(1,2)$  à  $\Phi^{*1}$ , teremos  $\Phi^{*2} = \{\emptyset''(0) = 1, \emptyset''(1) = \emptyset'(2) = 6, \emptyset''(2) = \emptyset'(1) = 0, \emptyset''(3) = 2, \emptyset''(4) = 4, \emptyset''(5) = 5, \emptyset''(6) = 3\}$ , ou seja

$$0 \rightarrow 1 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 0, 4 \rightarrow 4, 5 \rightarrow 5$$

Analogamente, aplicando agora  $I(3,4)$  à  $\Phi^{*2}$ , teremos  $\Phi^{*3}$ , onde  $\emptyset'''(3) = \emptyset''(4) = 4$  e  $\emptyset'''(4) = \emptyset''(3) = 2$  e todos dos outros  $\emptyset'''(l) = \emptyset''(l)$ , ou seja

$$0 \rightarrow 1 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 0, 5 \rightarrow 5$$

Por último, aplicando  $I(4,5)$  à  $\Phi^{*3}$ , teremos finalmente  $\Phi^{**}$ , onde  $\emptyset^{**}(4) = \emptyset'''(5) = 5$ ,  $\emptyset^{**}(5) = \emptyset'''(4) = 4$  e todos os outros  $\emptyset^{**}(l) = \emptyset'''(l)$ , ou seja

$$0 \rightarrow 1 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 0$$

O custo de  $\Phi^{**}$  é  $3 + 15 + 5 + 3 + 8 + 15 + 8 = 57$

Na prática, quando os tempos de preparação de máquina têm uma estrutura arbitrária, a regra do Menor Tempo de Preparação de Máquina (*Shortest Setup Time* – SST), como visto em 3.3.1, é frequentemente usada. Essa regra implica que quando um trabalho terminar, o trabalho com menor tempo de preparação de máquina é selecionado para ser o próximo. A regra SST é equivalente à regra do Vizinho Mais Próximo para o TSP. Aplicando a regra SST ao exemplo acima tem-se que:

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 0$$

E o custo dessa viagem é:  $3 + 13 + 3 + 5 + 3 + 8 + 24 = 59$ , superior ao encontrado pelo algoritmo, porém é uma boa aproximação da viagem ótima.