

# Algoritmo genético de agrupamento para formação de módulos de arranjo físico

Ana Rita Tiradentes Terra Argoud  
Eduardo Vila Gonçalves Filho  
Alexandre José Tiberti



## Resumo

*O projeto de arranjo físico modular tem como base o agrupamento de máquinas em módulos a partir de subsequências de operações comuns a um conjunto de peças. No método proposto por Huang (2003), os módulos de arranjo físico são gerados por análise de agrupamentos. O presente trabalho apresenta uma forma alternativa à geração de módulos de arranjo físico, através de algoritmo genético de agrupamento (AGA). O AGA permite ao usuário especificar a priori o número desejado de módulos (problema de K-agrupamentos), e também trabalhar de forma que o número e a formação dos módulos sejam variáveis de decisão do problema (problema de agrupamento automático). Uma característica importante do AGA é sua flexibilidade, pois fornece a possibilidade ao usuário de interagir com o método de solução através da escolha de diferentes codificações do cromossomo, de medidas de similaridade para comparação de seqüências de operações e de métodos de geração da nova população. Para tal foram desenvolvidos dois operadores de cruzamento e três operadores de mutação. No exemplo apresentado, os módulos de arranjo físico gerados pelo AGA e o arranjo físico final da fábrica foram comparados aos de Huang (2003) e houve importante redução da distância total percorrida pelo conjunto de peças, demonstrando a eficácia do algoritmo genético de agrupamento.*

*Dessa forma, os resultados apontaram o AGA como uma alternativa à geração de módulos de arranjo físico em projeto de arranjo físico modular.*

**Palavras-chave:** Arranjo físico modular. Algoritmos genéticos. Agrupamento. Manufatura.

## 1 Introdução

O projeto de arranjo físico em manufatura trata da disposição física das máquinas, equipamentos, áreas de suporte à produção e áreas para pessoal (TOMPKINS et al., 2003). O correto projeto do arranjo físico é decisivo para a sobrevivência e sucesso de uma empresa, pois permite a racionalização do espaço, minimização da movimentação de materiais e de pessoas, levando à redução de custos e aumento da eficiência do sistema produtivo.

Novos tipos de arranjo físico como o ágil, fractal, modular e distribuído têm despertado grande interesse entre os pesquisadores, seja em metodologia de projeto, em formas de avaliação de desempenho ou em comparações com arranjos físicos tradicionais (BENJAAFAR et al., 2002).

A prática tradicional da indústria é usar uma amostra de produtos para projetar o tipo de arranjo físico da fábrica. Por exemplo, para produtos padronizados e com

grande volume de demanda usa-se arranjo físico em linha, para produtos com características similares de projeto e processo constrói-se o arranjo físico celular, e produtos com baixo volume de demanda e sem similaridade aparente são processados em arranjo físico funcional.

O arranjo físico modular, tema deste trabalho, não assume a priori padrões de fluxo e critérios para agrupamento de máquinas, como nos arranjos celular, funcional e em linha. O arranjo físico modular identifica os padrões de fluxo existentes na fábrica e gera um arranjo, chamado módulo, para cada fluxo de materiais. Uma das vantagens do arranjo físico modular é possibilitar o encadeamento de operações e com isso diminuir a movimentação de peças na fábrica.

O primeiro trabalho encontrado sobre arranjo físico modular foi o de Shahrukh A. Irani e Heng Huang apresentado em 1998 (IRANI; HUANG, 1998). Em continuidade ao trabalho de 1998, Huang (2003) propôs

um procedimento para projeto de arranjo físico modular. Os dados de entrada necessários para o projeto de arranjo físico modular são o conjunto de peças com a seqüência de operações, a quantidade produzida de cada peça e os tempos de processamento de cada operação. Neste procedimento, os módulos de arranjo físico são gerados a partir de agrupamento de subseqüências de operações através de análise de agrupamentos. Para tal, Huang (2003) usou um método de agrupamento hierárquico aglomerativo com método da média das distâncias para regra de ligação.

Segundo Cole (1998), métodos de agrupamento hierárquico possuem algumas desvantagens:

- a) eles são indicados para pequenos conjuntos de dados, devido à necessidade de armazenar e atualizar constantemente matrizes de similaridade;
- b) se um elemento for alocado incorretamente a um grupo, em um estágio inicial, ele permanecerá no mesmo grupo até o término do processo. Não há chance para que o elemento procure pelo grupo correto; e
- c) os resultados refletem o grau com que os dados se ajustam à estrutura formal do método de agrupamento.

O objetivo deste trabalho foi desenvolver uma alternativa à geração dos módulos de arranjo físico de Huang (2003). No presente trabalho, os módulos são gerados a partir do agrupamento de subseqüências de operações através de algoritmo genético de agrupamento (AGA).

Para aplicação do procedimento proposto por Huang (2003), o usuário deve especificar o número desejado de módulos (grupos de máquinas), ou seja, o número de grupos é conhecido e é usado como parâmetro para se encontrar a solução. Entretanto, quando se define o número de grupos, podem ser gerados agrupamentos artificiais. Geralmente, o algoritmo é rodado várias vezes, para depois escolher a solução que pareça melhor ou que tenha a interpretação mais significativa. Para isso deve-se ter conhecimento do domínio do problema. Dessa forma, esses algoritmos realizam busca pela solução em um espaço relativamente pequeno (HRUSCHKA, 2001).

O algoritmo genético de agrupamento (AGA) desenvolvido neste trabalho permite ao usuário tanto especificar a priori o número desejado de módulos (problema de K-agrupamentos), como trabalhar de forma que o número e a formação dos módulos sejam variáveis de decisão do problema (problema de agrupamento automático). Além disso, o AGA é altamente flexível, pois possibilita ao usuário interagir com o método de solução através da escolha de diferentes codificações do cromossomo, de medidas de similaridade e de métodos de geração da nova população.

## 2 Metodologia de pesquisa

Segundo a classificação dos tipos de pesquisa apresentada por Marconi e Lakatos (1985), esta pesquisa é de natureza aplicada com caráter experimental. Para seu desenvolvimento faz-se uso de pesquisa bibliográfica.

A Figura 1 a seguir apresenta um fluxograma com as etapas do desenvolvimento desta pesquisa.

## 3 Algoritmos genéticos

Um algoritmo genético (AG) é um método de busca e otimização inspirado no princípio de seleção natural

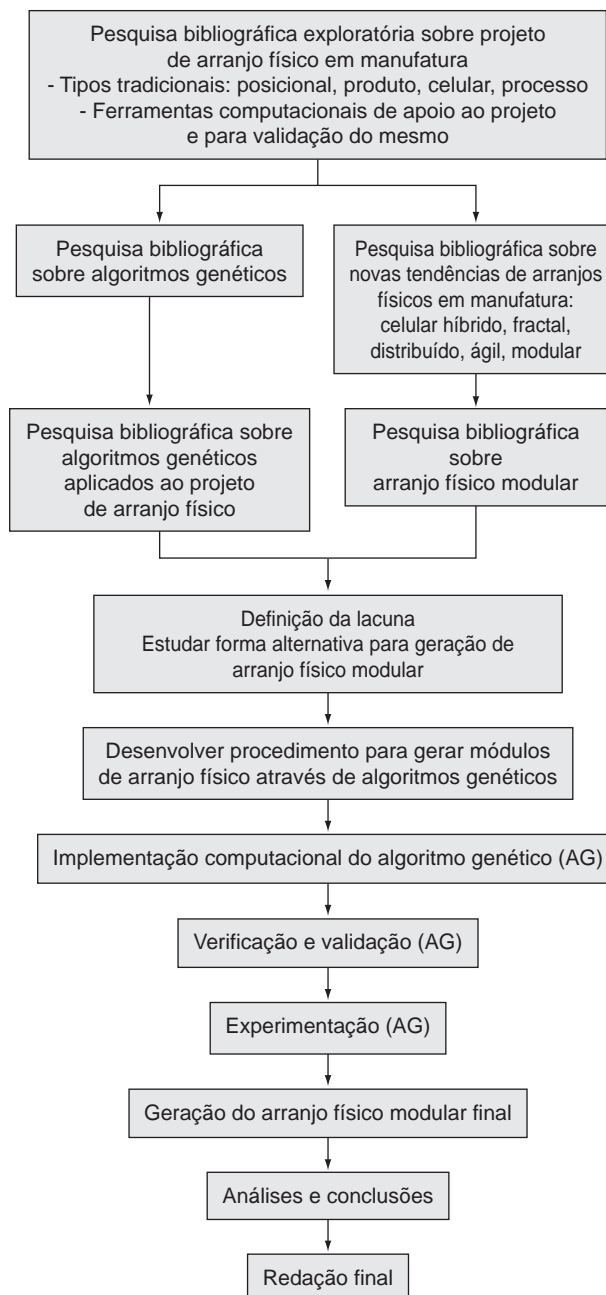


Figura 1. Fluxograma de desenvolvimento da pesquisa.

e sobrevivência dos indivíduos mais aptos, descoberto pelo fisiologista Charles Darwin e descrito em seu livro “A Origem das Espécies”, de 1859. Algoritmos genéticos foram introduzidos por John Holland, em 1975, e popularizados por David Goldberg, em 1989. Segundo Goldberg (1989), as principais características de um algoritmo genético (AG) são:

- trabalha com a codificação do conjunto de parâmetros em uma *string*;
- realiza busca a partir de uma população de pontos (possíveis soluções) e não de um ponto único;
- usa informação da função objetivo (função de aptidão) para efetuar busca por melhores soluções;
- usa regras de transição probabilísticas (escolha aleatória) para guiar a busca em regiões do espaço com maior chance de melhoria.

Muitos termos usados em algoritmos genéticos são originados da biologia. O Quadro 1, a seguir, sumariza os principais termos segundo Lacerda, Carvalho e Ludermir (2002).

Um AG inicia com uma população de indivíduos (cromossomos) que representam possíveis soluções do problema a ser resolvido. A população é avaliada e cada cromossomo recebe uma nota, denominada aptidão. Os melhores cromossomos (mais aptos) são selecionados e os piores descartados. Os selecionados podem sofrer cruzamento e mutação, gerando descendentes para a próxima geração. O processo continua até que uma solução satisfatória seja encontrada (LACERDA; CARVALHO; LUDERMIR, 2002). Este processo é ilustrado na Figura 2.

**Quadro 1.** Analogia de termos entre algoritmos genéticos e a biologia. Adaptado de Lacerda et al. (2002).

| Termo                   | Biologia  | Algoritmos genéticos (AG)   |
|-------------------------|---|---|
| Cromossomo ou indivíduo | Conjunto de genes de um organismo.  | <i>String</i> que representa uma solução para o problema.                   |
| Gene                    | Unidade de hereditabilidade transmitida pelo cromossomo e que controla as características do organismo. | Parâmetro codificado no cromossomo, ou seja, um elemento da <i>string</i> . |
| Genótipo                | Composição genética contida no cromossomo.  | Informação contida no cromossomo.   |
| Fenótipo                | Manifestação física de um gene no organismo, por exemplo, pele lisa ou rugosa.                          | Estrutura construída a partir do genótipo.                                  |
| Alelo                   | Uma das formas alternativas de um gene.   | Valores que o gene pode assumir.  |

Os principais elementos de um algoritmo genético são: codificação, população inicial, função de aptidão, seleção, operadores genéticos, geração da nova população, critério de parada e parâmetros. A seguir é apresentado o algoritmo genético de agrupamento (AGA) desenvolvido e é detalhado cada um daqueles elementos.

## 4 O algoritmo genético de agrupamento

O algoritmo genético de agrupamento (AGA) realiza o agrupamento de subsequências de operações e gera os módulos básicos de arranjo físico. O AGA segue os passos do Quadro 2, a seguir.

A determinação das subsequências comuns, descrita no passo 2 do Quadro 2 é feita de acordo com o algoritmo apresentado em Huang (2003). A matriz de similaridade entre subsequências de operações comuns descrita no passo 3 é calculada a partir da escolha pelo usuário de um dos coeficientes de similaridade: coeficiente baseado na distância de Levenshtein (TAM, 1990), coeficiente LCS – mais longa subsequência comum (ASKIN; ZHOU, 1998) ou coeficiente *Merger* (HUANG, 2003).

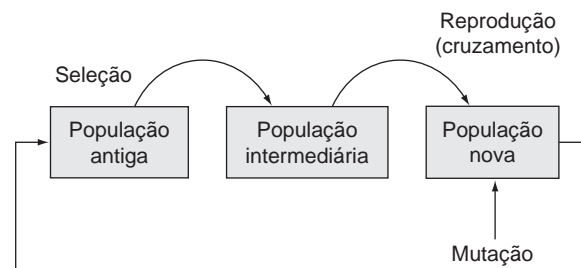
### 4.1 Codificação do cromossomo

A codificação refere-se à forma pela qual o cromossomo representa uma solução. O AGA permite a seleção de uma entre três formas de codificação do cromossomo, R1, R2 ou R3. As codificações R1 e R2 foram inspiradas nos trabalhos de Tseng e Yang (2001) e de Garai e Chaudhuri (2004), e a codificação R3 no de Hruschka (2001). Nas três codificações, o gene representa uma subsequência comum (conjunto de máquinas). O valor do gene indica o módulo ao qual a subsequência pertence.

R1: Inteiros e zeros. Esta codificação é usada quando o usuário especifica o número desejado de módulos ou grupos (K). O tamanho do cromossomo equivale ao número de subsequências comuns. Seja o exemplo com 8 subsequências e 4 grupos.

(2 0 0 3 1 0 4 0)  
S1 S2 S3 S4 S5 S6 S7 S8

As subsequências S1, S4, S5 e S7 foram sorteadas e receberam aleatoriamente valores de 1 a 4. Essas subse-



**Figura 2.** Princípio de trabalho de um algoritmo genético. (GONÇALVES FILHO et al., 2004).

**Quadro 2.** Algoritmo genético de agrupamento (AGA). (ARGOUD, 2007).

|              |   |
|--------------|---|
| Passo 1:     | Entrada de dados: peças e seqüência de operações.   |
| Passo 2:     | Encontre as subseqüências de operações (máquinas) consecutivas comuns entre todos os pares de seqüência.  |
| Passo 3:     | Gere a matriz de similaridade entre subseqüências comuns.   |
| Passo 4:     | Escolha a forma de codificação do cromossomo (R1, R2 ou R3) e o método de aplicação dos operadores genéticos (probabilístico ou faixa percentual).                            |
| Passo 5:     | Gere a população inicial.   |
| Passo 5.1:   | Defina intervalo para o valor de K (número de grupos).  |
| Passo 5.2:   | Inicie o indivíduo.   |
| Passo 5.2.1: | Plante as sementes nos grupos.  |
| Passo 5.2.2: | Agrupe as demais subseqüências de acordo com a codificação do cromossomo usada: se codificação R1 ou R2 faça itens a) e b); se R3, faça o item c).                            |
| a)           | para cada subseqüência não agrupada ( $D_i$ ), procure pelo grupo ( $G_j$ ) com o qual $D_i$ possua a maior similaridade média ( $\overline{Sim}$ ) e insira $D_i$ em $G_j$ ; |
| b)           | repita o item a) até que todas as subseqüências sejam agrupadas. Mostre os agrupamentos formados. Vá para o passo 6; e  |
| c)           | cada subseqüência não agrupada ( $D_i$ ) é inserida em um grupo escolhido aleatoriamente. Mostre os agrupamentos formados. Vá para o passo 6.                                 |
| Passo 6:     | Calcule o valor da função de aptidão para cada cromossomo.  |
| Passo 7:     | Teste o critério de parada. Em caso afirmativo, mostre os cromossomos em ordem decrescente de aptidão e termine o algoritmo. Em caso negativo vá para o passo 8.              |
| Passo 8:     | Determine as melhores soluções (regra do elitismo) da população atual e leve para a nova população. Faça os passos 9 a 15 até que se complete a nova população.               |
| Passo 9:     | Selecione os cromossomos para cruzamento usando o procedimento da roleta.   |
| Passo 10:    | Se o método de aplicação dos operadores for probabilístico vá para o passo 11, se for faixa percentual vá para o passo 13.  |
| Passo 11:    | Aplique o operador de cruzamento caso o cruzamento ocorra. Corrija os cromossomos descendentes se necessário.   |
| Passo 12:    | Aplique os operadores de mutação nos descendentes se a probabilidade de mutação ocorrer. Vá para o passo 15.  |
| Passo 13:    | Aplique o operador de cruzamento aos x% (taxa de cruzamento) primeiros cromossomos sorteados e corrija os descendentes se necessário.   |
| Passo 14:    | Aplique os operadores de mutação ao restante dos cromossomos sorteados, respeitando as taxas de cada mutação e a ordem de seleção.  |
| Passo 15:    | Adicione os descendentes à nova população.  |
| Passo 16:    | Substitua a população atual pela nova e retorne ao passo 6.   |

qüências funcionarão como sementes de agrupamentos iniciais. Os grupos serão formados pelas seguintes subseqüências sementes: grupo 1: S5; grupo 2: S1; grupo 3: S4 e grupo 4: S7.

R2: Esta codificação é usada quando o número de módulos não é especificado pelo usuário, ou seja, quando o algoritmo resolve o problema de agrupamento automático. O usuário apenas especifica um valor mínimo e máximo para K (número de grupos). A diferença em relação à R1 está na adição de um gene na última posição do cromossomo, indicando o número de grupos do cromossomo específico. Seja o exemplo com 8 subseqüências. O número de grupos (K) é sorteado aleatoriamente no intervalo  $1 \leq K \leq 8$ . No exemplo,  $K = 4$ :

(2 0 0 4 1 0 3 0 4)  
S1 S2 S3 S4 S5 S6 S7 S8 K

os grupos serão formados pelas seguintes subseqüências de sementes, sorteadas aleatoriamente: grupo 1: S5; grupo 2: S1; grupo 3: S4 e grupo 4: S7. A designação dos

genes zerados aos grupos, nas codificações R1 e R2, é feita de acordo com procedimento descrito na seção 3.2.

R3: Grupo-rótulo. Esta codificação também é usada em problemas de agrupamento automático. A diferença em relação à R2 está no fato de que em R3 todas as subseqüências são designadas aos grupos aleatoriamente. Seja o exemplo com 8 subseqüências e número de grupos (K) variando no intervalo  $1 \leq K \leq 8$ . Suponha que o valor sorteado de K seja 3.

(2 1 2 1 3 3 2 1 3)  
S1 S2 S3 S4 S5 S6 S7 S8 K

Os grupos serão formados pelas subseqüências: grupo 1: S2, S4, S8; grupo 2: S1, S3, S7 e grupo 3: S5, S6.

#### 4.2 População inicial

A população inicial é formada por um conjunto de cromossomos, ou indivíduos, que representam possíveis soluções do problema. No AGA, a geração da população inicial é feita conforme os passos a seguir.

Passo 1: Definição do valor de K (número de grupos)



O número de grupos ( $K$ ) deve variar entre um valor mínimo e máximo definido pelo usuário, ou seja,  $K_{\min} \leq K \leq K_{\max}$ . Conforme a codificação do cromossomo, R1, R2 ou R3, algumas restrições devem ser respeitadas.

- restrição 1:  $K_{\min} = K_{\max}$  para a codificação R1. Como  $K$  é fixo, todos os indivíduos terão o mesmo número de grupos.
- restrição 2:  $K_{\min} \neq K_{\max}$  para codificações R2 ou R3. Sorteie o número de grupos ( $K$ ) para cada indivíduo aleatoriamente conforme distribuição uniforme entre  $K_{\min}$  e  $K_{\max}$ . Para a função de aptidão largura da silhueta média, detalhada adiante,  $K_{\min} \geq 2$  e  $K_{\max} \leq N_{\text{Subseq}}$ .

Passo 2: Iniciação de um indivíduo

Neste passo são formados os grupos em cada cromossomo, do seguinte modo:

- plante as sementes nos grupos. Selecionam-se  $K$  subseqüências aleatoriamente e coloca-se uma em cada grupo, para garantir que haja ao menos uma subseqüência por grupo; e
- agrupe as demais subseqüências de acordo com a codificação do cromossomo. Para codificações R1 ou R2: para cada subseqüência não agrupada ( $D_i$ ) procure pelo grupo ( $G_j$ ) com o qual  $D_i$  possua a maior similaridade média ( $\overline{\text{Sim}}$ ), e insira  $D_i$  em  $G_j$ . A similaridade média é calculada pela Equação 1:

$$\overline{\text{Sim}}(G_j, D_i) = \frac{1}{|G_j|} \sum_{x \in G_j} M[x, D_i] \quad (1)$$

em que:

$D_i$  = subseqüência não agrupada;  $x$  = subseqüência do grupo  $G_j$ ;  $|G_j|$  = número de elementos do grupo  $G_j$ ;  $M$  = valor da similaridade na matriz de subseqüências de operações.

Este procedimento de agrupamento de subseqüências foi inspirado nos trabalhos de Tseng e Yang (2001) e Garai e Chaudhuri (2004), que realizaram um pré-agrupamento do conjunto de dados com o objetivo de reduzir o tempo computacional do algoritmo. Entretanto, esses autores trabalharam com distância do centróide e implementaram o método do vizinho mais próximo. O procedimento aqui proposto, para R1 ou R2, visa formar agrupamentos iniciais com o intuito de guiar o algoritmo genético a partir de boas soluções iniciais.

Dois casos especiais devem ser considerados para as codificações R1 e R2.

Quando a subseqüência  $D_i$  possui similaridade igual a zero para com todas as outras subseqüências e o número de grupos ( $K$ ) do cromossomo é menor que o máximo, ou seja,  $\text{Sim}(D_i) = 0$  e  $K < K_{\max}$ . Neste caso deve ser criado um novo grupo para  $D_i$ .

Quando  $\text{Sim}(D_i) = 0$  e  $K = K_{\max}$ , ou seja, não se pode criar novo grupo, pois  $K$  está no máximo. Neste caso,  $D_i$  é inserida em um grupo existente aleatoriamente.

Para a codificação R3, cada subseqüência não agrupada ( $D_i$ ) é inserida em um grupo escolhido aleatoriamente.

Passo 3: Classificação dos indivíduos

Os indivíduos são classificados do modo a seguir, para posteriormente serem selecionados para cruzamento e/ou mutação.

- calcule o valor de aptidão para cada indivíduo;
- ordene os indivíduos em ordem decrescente dos valores de aptidão e guarde os valores de melhor e pior aptidão; e
- calcule a média dos valores de aptidão.

### 4.3 Função de aptidão

O papel da função de aptidão em um algoritmo genético é avaliar a qualidade da solução. A função de aptidão largura da silhueta média, empregada no AGA, foi proposta por Kaufman e Rousseeuw (1990). Ela trabalha com dissimilaridades entre elementos de um conjunto de dados. A largura da silhueta média maximiza a homogeneidade interna de cada grupo e a heterogeneidade entre grupos distintos.

Aplicada ao projeto de arranjo físico modular, a homogeneidade dentro de cada módulo pode ser obtida pela minimização das dissimilaridades entre suas subseqüências. A heterogeneidade entre diferentes módulos pode ser medida pela maximização da dissimilaridade média entre as subseqüências.

Considere uma subseqüência  $i$  qualquer do conjunto de dados. Para definir a silhueta  $s(i)$ , primeiro definimos  $A$ , o grupo ao qual a subseqüência pertence e calculamos:

- $a(i)$  = dissimilaridade média de  $i$  em relação a todas as outras subseqüências de  $A$ ; e
- agora considere qualquer grupo  $C$  diferente de  $A$  e definimos:  $d(i, C)$  = dissimilaridade média da subseqüência  $i$  em relação a todas as subseqüências de  $C$ . Depois de calcular  $d(i, C)$  para todos os grupos com  $C \neq A$ , selecionamos a menor delas (Equação 2):

$$b(i) = \min d(i, C), C \neq A \quad (2)$$

O grupo  $B$  para o qual esta dissimilaridade é mínima [ $d(i, B) = b(i)$ ] é chamado de vizinho da subseqüência  $i$ . É a segunda melhor escolha da subseqüência  $i$ . Note que  $b(i)$  depende da capacidade dos grupos em diferir de  $A$ , portanto, as silhuetas não são definidas para  $K=1$  (quando houver apenas um grupo).

A largura da silhueta de cada subseqüência  $i$  é definida pela Equação 3:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3)$$

Propriedades de  $s(i)$ :

- $-1 \leq s(i) \leq 1$ ;
- $s(i) = 0$  quando  $A = \{i\}$  (o grupo  $A$  possui apenas uma subseqüência); e
- $2 \leq K \leq N_{\text{Subseq}}$  (número de subseqüências comuns).

Quando  $s(i)$  é muito próxima do valor 1 significa que a dissimilaridade interna,  $a(i)$ , é muito menor que a menor dissimilaridade externa,  $b(i)$ . Portanto podemos concluir que a subsequência  $i$  está bem classificada em seu grupo. Quando  $s(i)$  se aproxima de zero,  $a(i)$  e  $b(i)$  são aproximadamente iguais, portanto não está claro se a subsequência  $i$  deveria ficar em A ou em seu vizinho B. Já quando a silhueta  $s(i)$  está próxima do valor  $-1$ , então  $a(i)$  é muito maior que  $b(i)$ , o que significa que a subsequência  $i$  está mal classificada no grupo A. Ela está muito mais próxima de B do que de A.

A média de  $s(i)$  para  $i = 1, 2, \dots, n$ , ou seja, para todas as subsequências, é chamada largura da silhueta média para o conjunto de dados e é a função de aptidão usada no algoritmo genético de agrupamento (Equação 4).

$$\text{Função de aptidão (FA)} = \overline{s(i)} \quad (4)$$

Este valor,  $\overline{s(i)}$ , é usado para seleção do melhor valor de  $K$  (número de grupos), que acontece quando  $\overline{s(i)}$  for máximo (KAUFMAN; ROUSSEUW, 1990).

#### 4.4 Operadores genéticos

Neste trabalho foram desenvolvidos dois operadores de cruzamento e três de mutação. A aplicação dos operadores irá depender da codificação do cromossomo. Para a codificação R1, na qual o usuário determina o número desejado de módulos, somente podem ser aplicados um dos operadores de cruzamento e o operador de mutação 3. Para as codificações R2 ou R3, nas quais o número de módulos formado é um resultado do algoritmo, podem ser aplicados um dos operadores de cruzamento e os operadores de mutação 1, 2 e 3.

##### 4.4.1 Operador de cruzamento Recombinação com Correção Aleatória

O operador de cruzamento denominado Recombinação com Correção Aleatória foi adaptado de Hruschka (2001), que desenvolveu um algoritmo genético de agrupamento e o aplicou a várias bases de dados, entre elas *Iris Plants Database* (classes de plantas), *Australian Credit Approval Database* (cartão de crédito), *Wisconsin Breast Cancer Database* (dados sobre câncer) e dados meteorológicos. De um modo simplificado, alguns grupos do pai A são copiados no pai B, e vice-versa, formando dois descendentes, C e D.

As principais contribuições deste operador, que o fazem diferir do operador de Hruschka (2001) são:

- Hruschka (2001) trabalhou com centróide como medida de proximidade. No presente trabalho, foi usada como medida de proximidade a menor das dissimilaridades médias da subsequência  $i$  para com cada grupo contido no descendente, ou seja,  $b(i) = \min d(i, G_j)$ . O uso desta medida é detalhado adiante.

- a introdução do mecanismo de correção dos descendentes para evitar cromossomos inválidos. A seguir é apresentado um exemplo que ilustra a ação do operador.

Sejam dois cromossomos pais, A e B, com codificação R3, constituídos de 8 subsequências comuns:

A - 1 1 2 3 1 4 3 3    n° de grupos,     $K_A = 4$

B - 1 2 3 1 1 4 4 3    n° de grupos,     $K_B = 4$

O número de grupos não está representado no cromossomo. Para o cromossomo A o grupo 1 é composto pelas subsequências S1, S2 e S5, o grupo 2 pela subsequência S3, o grupo 3 pelas subsequências S4, S7 e S8 e o grupo 4 pela subsequência S6.

Escolha aleatoriamente  $t$  ( $1 \leq t \leq K_A$ ) grupos de A para copiar em C (descendente). Por exemplo, selecione os grupos 1 e 2 de A (em negrito), ou seja,  $t = \{1, 2\}$ :

A - **1 1 2 3** 1 4 3 3

B - 1 2 3 1 1 4 4 3

Estes grupos modificam os grupos 1, 2 e 3 de B. O grupo 4 de B não é modificado.

A - **1 1 2 3** 1 4 3 3

B - **1 2 3** 1 1 4 4 3

Os genes de B sublinhados também serão modificados no descendente C. Copie os genes selecionados de A, ou seja,  $\{1, 2\}$ , em C:

C - **1 1 2 0** 1 0 0 0

Os genes completados com zero em C indicam posições com potencial para receber genes não alterados de B, ou representam genes que serão reposicionados no grupo mais próximo. Por exemplo, o grupo 4 de B é copiado em C, pois não foi mudado. Os grupos 1 e 3 de B (sublinhados) serão reposicionados no grupo mais próximo. Assim:

C - **1 1 2 0** 1 4 4 0

Para reposicionamento dos genes zerados em C primeiro calcula-se a dissimilaridade média da subsequência  $i$  (gene zerado) para com cada grupo ( $G_j$ ) contido no descendente C. O gene zerado será reposicionado no grupo com o qual possuir a menor das dissimilaridades médias, ou seja,  $b(i) = \min d(i, G_j)$ , com  $1 \leq j \leq k$ .  $k$  é o número de grupos do cromossomo.

Para gerar o descendente D é feito o processo inverso. Copie em D todos os genes de B que foram alterados. Assim:

D - 1 2 3 1 1 0 0 3

Os genes completados com zero em D indicam posições com potencial para receber genes não alterados de A, ou representam genes que serão reposicionados no grupo mais próximo. Por exemplo, o grupo 4 de A é copiado em D, pois não foi mudado. O grupo 3 de A não é copiado (foi mudado), então o gene zerado de D será reposicionado no grupo mais próximo. Assim:

D - 1 2 3 1 1 4 0 3

#### 4.4.1.1 Mecanismo de correção dos descendentes

Após o cruzamento, algumas correções devem ser feitas nos descendentes para evitar inconsistência. São elas:

- reidentificação dos grupos. Ao criar os descendentes devem-se reidentificar os grupos, pois o grupo 1 do pai A é diferente do grupo 1 do pai B; e
- verificação do número mínimo e máximo de grupos.

O número de grupos do descendente deve estar entre o número de grupos mínimo e máximo estabelecido pelo usuário, ou seja,  $K_{\min} \leq K \leq K_{\max}$ . Duas situações podem ocorrer.

Situação 1: Quando  $K < K_{\min}$ . Nesta situação é preciso criar grupos. O seguinte procedimento deve ser adotado:

- escolha aleatoriamente um grupo que possua no mínimo duas subseqüências;
- selecione um ponto de partição aleatoriamente e divida o grupo, gerando dois novos grupos;
- repita o procedimento até se atingir o número de grupos mínimo permitido ( $K_{\min}$ ).

Situação 2: Quando  $K > K_{\max}$ . Nesta situação é preciso extinguir grupos. O seguinte procedimento deve ser adotado:

- selecione um grupo aleatoriamente;
- distribua cada subseqüência do grupo selecionado para o grupo que possuir a menor dissimilaridade média:  $b(i) = \min d(i, G_j)$ , ou seja, a menor das dissimilaridades médias da subseqüência  $i$  para com cada grupo  $G_j$  do cromossomo; e
- repita o procedimento até se atingir o número de grupos máximo permitido ( $K_{\max}$ ).

#### 4.4.2 Operador de cruzamento ponto único de grupo

O operador de cruzamento denominado ponto único de grupo age nos grupos de subseqüências comuns. Este operador foi inspirado no trabalho de Gonçalves Filho e Tiberti (2006). As principais contribuições referentes ao operador Ponto Único de Grupo para este trabalho, que o fazem diferir do operador de Gonçalves Filho e Tiberti (2006) são:

- no operador ponto único de grupo houve a preocupação em preservar nos descendentes o mesmo número de grupos dos cromossomos pais. Para tal foi definido um procedimento para verificação e preenchimento de grupos vazios;
- em Gonçalves Filho e Tiberti (2006) os genes representam máquinas, enquanto no presente trabalho, subseqüências comuns;
- neste trabalho as correções nos descendentes são realizadas na porção do cromossomo que foi trocada,

enquanto que em Gonçalves Filho e Tiberti (2006) foram realizadas na parte “fixa” do cromossomo;

- para Gonçalves Filho e Tiberti (2006), em um grupo não pode haver menos que duas máquinas. Neste trabalho não há esta restrição quanto às subseqüências;
- introdução de critério para designação de elementos (subseqüências) faltantes (passo 6 do exemplo a seguir); e
- na implementação deste operador não foi necessário usar as formas 2 e 3 de codificação do cromossomo, propostas por Gonçalves Filho e Tiberti (2006). Os grupos e os elementos dos grupos são identificados automaticamente a partir da codificação original do cromossomo, definida na seção 3.1. Entretanto, no passo 1 do exemplo a seguir, a forma 2 de Gonçalves Filho e Tiberti (2006) foi usada para facilitar a visualização da ação do operador.

Sejam dois cromossomos, A e B, com 5 e 3 grupos, respectivamente, ou seja,  $K_A = 5$  e  $K_B = 3$ .

A - 1 2 3 1 3 4 5 4

B - 1 1 2 3 1 2 3 3

Para o cromossomo B, o grupo 1 é composto pelas subseqüências S1, S2 e S5, o grupo 2 pela subseqüência S3 e S6, e o grupo 3 pelas subseqüências S4, S7 e S8.

Passo 1: Expresse a codificação original em termos de grupos. Os cromossomos A e B são re-escritos da forma:

A - 1 4 / 2 / 3 5 / 6 8 / 7

B - 1 2 5 / 3 6 / 4 7 8

O primeiro grupo do cromossomo A é formado pelas subseqüências S1 e S4, o segundo grupo pela subseqüência S2, e assim por diante. As barras representam a separação entre os grupos.

Passo 2: Gere um número aleatório inteiro entre 1 e  $k-1$ , sendo  $k$  o número de grupos do cromossomo pai com menos grupos. Este valor define o ponto do cruzamento.

No exemplo, como B possui o menor número de grupos, o ponto de cruzamento é escolhido aleatoriamente entre 1 e 2. Suponha que o valor 2 seja escolhido.

Passo 3: Os descendentes são gerados trocando os grupos com rótulo superior ao ponto de cruzamento. Portanto, os grupos trocados serão: grupos 3, 4 e 5 de A e grupo 3 de B.

A - 1 4 / 2 / 3 5 / 6 8 / 7    C - 1 4 / 2 / 4 7 8  
 B - 1 2 5 / 3 6 / 4 7 8    D - 1 2 5 / 3 6 / 3 5 / 6 8 / 7

Passo 4: Exclusão de subseqüências repetidas. Esta troca de grupos pode violar a restrição de que cada subseqüência esteja em apenas um grupo. Para corrigir este problema, as subseqüências que foram trocadas e que já estejam presentes na parte “fixa” do cromossomo descendente são excluídas (subseqüências sublinhadas).

C - 1 4 / 2 / **4 7 8**

D - 1 2 5 / 3 6 / **3 5 / 6 8 / 7**

Portanto, a subseqüência 4 no descendente C e as subseqüências 3, 5 e 6 no descendente D são excluídas. Assim:

C - 1 4 / 2 / **7 8**

D - 1 2 5 / 3 6 // **8 / 7**

Passo 5: Preenchimento de grupos vazios. Após a exclusão das subseqüências, caso haja grupo vazio no descendente, o que não é permitido, escolha aleatoriamente uma subseqüência sem grupo (não presente no cromossomo) e a designe ao grupo vazio. Repita este procedimento até que os grupos vazios sejam preenchidos.

No descendente D, há um grupo vazio. Então, a subseqüência 4 ainda sem grupo, será designada ao grupo vazio. Assim, os descendentes são reescritos na forma:

C - 1 4 / 2 / 7 8

D - 1 2 5 / 3 6 / 4 / 8 / 7

Passo 6: Designação de subseqüências faltantes. Se ainda houver subseqüência sem grupo, insira-a no grupo com o qual possua menor dissimilaridade média.

No descendente C, as subseqüências 3, 5 e 6, ainda sem grupo, serão designadas aos grupos conforme a menor dissimilaridade média. No descendente D não há mais subseqüências a serem designadas.

Passo 7: Verificação de grupos vazios. Caso ainda haja grupos vazios (o número de subseqüências sem grupo não foi suficiente), escolha aleatoriamente um grupo com mais de duas subseqüências e retire aleatoriamente uma subseqüência para ser colocada no novo grupo.

#### 4.4.3 Operadores de mutação

Aplicada ao problema de agrupamento, a mutação corresponde a mover um objeto de um grupo para outro grupo. Como realizar a mutação depende da codificação do cromossomo (COLE, 1998). No presente trabalho foram desenvolvidos três operadores de mutação. Os operadores de mutação 1 e 2 foram inspirados no trabalho de Hruschka (2001) e os operadores de mutação 3 no trabalho de Cole (1998). As principais contribuições referentes aos operadores de mutação desenvolvidos, que os fazem diferir dos operadores dos autores *supra* citados, são:

- a) a forma de inserção dos objetos (subseqüências) nos grupos pelo operador de mutação 1. Hruschka (2001) os insere no grupo com centróide mais próximo. No presente trabalho as subseqüências são inseridas no grupo com a menor dissimilaridade média;
- b) a forma de escolha das sementes no operador de mutação 2. Em Hruschka (2001) as sementes são os objetos mais próximos e mais distantes do centróide. No presente trabalho as sementes são as subseqüências com menor e maior dissimilaridade média para com as subseqüências do mesmo grupo; e

- c) o operador *Move* aleatório é semelhante ao de Cole (1998), com a diferença de que neste trabalho o grupo selecionado deve possuir no mínimo dois objetos (subseqüências).

Operador mutação 1 (*merge*): opera em cromossomos que satisfazem à restrição  $K > K_{\min}$ . Este operador elimina aleatoriamente um grupo e insere suas subseqüências nos demais grupos. Cada subseqüência é inserida no grupo com o qual possuir a menor dissimilaridade média:  $b(i) = \min d(i, G_j)$ , ou seja, a menor das dissimilaridades médias da subseqüência  $i$  para com cada grupo  $G_j$  do cromossomo. A mutação 1 faz diminuir o número de grupos.

Operador mutação 2 (*break*): opera em cromossomos que satisfazem a restrição  $K < K_{\max}$ . Este operador divide um grupo selecionado aleatoriamente (com no mínimo dois elementos) em dois novos grupos. A semente do primeiro subgrupo será a subseqüência com menor  $a(i)$ , ou seja, menor dissimilaridade média da subseqüência  $i$  em relação às outras subseqüências do grupo selecionado. A semente do segundo subgrupo será a subseqüência com maior  $a(i)$ . Feito isso, as subseqüências restantes do grupo selecionado são deslocadas para os subgrupos mais próximos. O subgrupo mais próximo será aquele com menor dissimilaridade média:  $b(i) = \min d(i, G_j)$ , ou seja, a menor das dissimilaridades médias da subseqüência  $i$  para com cada grupo  $G_j$  do cromossomo. Os demais grupos ficam intactos. A mutação 2 faz aumentar o número de grupos.

Operador mutação 3 (*move*): A mutação 3 preserva o número de grupos ( $K$ ), mas muda o tamanho de dois grupos. Selecione aleatoriamente uma subseqüência de um grupo com no mínimo dois elementos. Mova esta subseqüência para qualquer outro grupo aleatoriamente.

#### 4.5 Processo de seleção e geração da nova população

A seleção é o processo de escolha dos cromossomos mais aptos para sofrerem a ação dos operadores genéticos (cruzamento e mutação). A geração da nova população refere-se ao modo de inserção dos descendentes na população. A cada geração, os  $N$  pais ( $N$  é o tamanho da população) são substituídos por  $(N - m)$  filhos, conforme os passos a seguir.

Passo 1: Regra do elitismo.

Para garantir que as boas soluções permaneçam de uma geração para outra e não sejam perdidas no processo de seleção, os  $m$  melhores cromossomos podem ser copiados integralmente para a nova geração. O usuário pode selecionar a porcentagem de cromossomos copiados, com  $0 \leq m < N$ , em que  $N$  é o tamanho da população. O restante da população é submetida ao processo de seleção pela roleta para, em seguida, sofrer cruzamento e/ou mutação.

Passo 2: Inicia a roleta para selecionar os indivíduos



O processo detalhado de seleção pela roleta é descrito a seguir.

- a) faça a normalização do valor de aptidão de cada indivíduo.
- segundo Lacerda, Carvalho e Ludermir (2002), os indivíduos podem apresentar valores de aptidão muito próximos. Quando isto ocorre, o processo de seleção pode se tornar praticamente aleatório, como se cada indivíduo possuísse uma estreita faixa na roleta. Para evitar que isso ocorra, é feita uma normalização dos valores da aptidão. No presente trabalho foi definido um fator para normalização (*Fat*), calculado pela diferença entre os valores de aptidão do melhor e do pior indivíduo, dividida pelo número de indivíduos da população (Equação 5):

$$Fat = \frac{(melhor\ aptidão - pior\ aptidão)}{N} \quad (5)$$

- b) calcule a probabilidade de seleção para cada indivíduo.
- a probabilidade de seleção ( $P_i$ ) de cada indivíduo equivale a uma determinada faixa da roleta, e é dada pela Equação 6:

$$P_i = (N - i) \cdot Fat + Fat, \text{ com } 1 \leq i \leq N \quad (6)$$

- $i$  é a posição de cada indivíduo na população ordenada (do melhor para o pior). Isso fará com que os indivíduos com melhor aptidão tenham maior chance de seleção, pois possuirão maior faixa da roleta. Caso  $Fat = 0$  (zero) assume-se  $Fat = 1/N$ , em que  $N$  é o tamanho da população. A constante  $Fat$  foi adicionada ao segundo termo da Equação 6 para permitir que mesmo os indivíduos com piores classificações tenham uma pequena chance de seleção.
- c) selecione os indivíduos aleatoriamente pela roleta.
- a roleta é montada na forma de um vetor ( $r$ ) com  $(N+1)$  elementos, sendo  $N$  o número de indivíduos da população. Cada elemento  $r[i]$  corresponde a uma faixa (intervalo) da roleta equivalente à probabilidade de seleção do  $i$ -ésimo indivíduo da população ( $P_i$ ), sendo o valor extremo do intervalo calculado pela Equação 7:

$$r[i] = r[1 + 1] + P_i, \text{ para } 1 \leq i \leq N \quad (7)$$

E, ainda,  $r[N+1] = 0$ .

O procedimento adotado para seleção de um indivíduo é:

- a) gera-se um número aleatório ( $x$ ) entre 0 e  $r[1]$  (comprimento da roleta), isto é,  $0 \leq x < r[1]$ ; e
- b) o vetor  $r$  é varrido em busca de um índice  $i$  cujo valor  $x$  esteja entre  $r[i]$  e  $r[i + 1]$ , isto é,  $r[i + 1] \leq x < r[i]$ , em que  $i$  é a posição do indivíduo selecionado.

Passo 3: Gere os  $(N-m)$  indivíduos restantes da nova população

Os indivíduos restantes totalizam  $(N - m)$ , sendo  $N$  o número de indivíduos da população e  $m$  o número de indivíduos copiados pela Regra do elitismo. Os indivíduos restantes da nova população são gerados com base em um dos métodos descritos a seguir, selecionados pelo usuário.

#### 4.5.1 Método probabilístico

Este é o método tradicional (GOLDBERG, 1989) de aplicação dos operadores de cruzamento e mutação, conforme suas taxas de probabilidade. O método obedece às seguintes etapas:

- a) selecione dois indivíduos (pais) aleatoriamente pela roleta (item c) do passo 2;
- b) aplique o operador de cruzamento para gerar dois descendentes, caso a probabilidade de cruzamento ocorra, caso contrário, os dois descendentes serão apenas cópias de seus pais; e
- c) aplique os operadores mutação 1, mutação 2 e mutação 3 em cada descendente, se a probabilidade de mutação ocorrer.

O operador é aplicado quando sua taxa de probabilidade for maior ou igual a um valor ( $x$ ) gerado aleatoriamente entre 0 e 1 (limites das taxas de probabilidade).

Seja o exemplo, com as seguintes taxas de probabilidade definidas pelo usuário:  $P_{\text{cruzamento}} = 0,80$ ;  $P_{\text{mutação 1}} = 0,05$ ;  $P_{\text{mutação 2}} = 0,05$ ; e  $P_{\text{mutação 3}} = 0,10$ . Suponha que para o cruzamento,  $x$  seja igual a 0,673 (gerado aleatoriamente). Portanto, o cruzamento irá ocorrer, pois  $0,80 > 0,673$ .

#### 4.5.2 Método faixa percentual

O método faixa percentual foi proposto por Hruschka (2001). Os operadores de cruzamento e mutação são aplicados em um determinado número de indivíduos sorteados aleatoriamente, conforme as etapas:

- a) sorteie  $(N - m)$  indivíduos aleatoriamente pela roleta (item c) do passo 2;
- b) nos primeiros  $x\%$  (taxa de cruzamento) indivíduos sorteados, aplique o operador de cruzamento entre o 1° e o último indivíduo, seguido do 2° com o penúltimo, e assim por diante; caso  $x\%$  seja um número ímpar, apenas copie o indivíduo central (sem par) para a nova população; e
- c) aplique os operadores de mutação nos indivíduos restantes:

Para codificação R1: aplica-se o operador de mutação 3 em todos os indivíduos restantes. As mutações 1 e 2 não são aplicadas para essa codificação.

Para codificações R2 ou R3: aplicam-se todas as mutações (1, 2 e 3), respeitando a ordem de seleção e as taxas de cada mutação.

Seja o exemplo com as seguintes taxas de probabilidade definidas pelo usuário:  $P_{\text{cruzamento}} = 0,50$ ;  $P_{\text{mutação1}} = 0,25$ ;  $P_{\text{mutação2}} = 0,25$  e  $P_{\text{mutação3}} = 0,00$ . Neste exemplo, a primeira metade dos cromossomos sorteados pela roleta sofre cruzamento, pois  $P_c = 0,50$ . O primeiro cromossomo sorteado é cruzado com o  $n/2$ , em que  $n$  representa o número de cromossomos sorteados, o segundo cromossomo com  $[(n/2)-1]$  e assim por diante. No exemplo, a mutação 1 é aplicada aos cromossomos que pertencem ao intervalo  $[(n/2)+1, 3n/4]$  e a mutação 2 é aplicada aos cromossomos que pertencem ao intervalo  $[(3n/4)+1, n]$ . A mutação 3 não é aplicada, pois  $P_{\text{mutação3}} = 0,00$ .

Independente do método usado, por probabilidade ou por faixa percentual, se for gerado menos que  $(N - m)$  indivíduos, a população será completada com indivíduos gerados aleatoriamente pela roleta (item c) do passo 2.

Passo 4: Classificação dos indivíduos

Classifique os indivíduos da nova população por ordem decrescente da aptidão e obtenha os valores: médio, melhor e pior da aptidão.

#### 4.6 Critério de parada

O usuário deve especificar o número de gerações desejado para término do processo de evolução e parada do algoritmo. Não há um número pré-definido para este valor, ele é determinado experimentalmente e deve ser suficiente para permitir a convergência da população. Após o término, o algoritmo ordena os indivíduos em ordem decrescente pelo valor da aptidão.

### 5 Exemplo de aplicação

O conjunto de dados usado neste exemplo é o mesmo apresentado em Huang (2003) para gerar módulos de arranjo físico através de análise de agrupamento (*cluster analysis*). Para que se pudesse fazer a comparação entre as duas soluções, neste exemplo é tratado o problema de K-agrupamentos, no qual o usuário especifica o número desejado de grupos, neste caso, 4 módulos de arranjo físico. Os dados de entrada são apresentados na Tabela 1.

Na Tabela 1, a peça 1 é feita nas máquinas 1, 4, 8 e 9, e assim por diante. A partir da seqüência de operações, conforme procedimento descrito em Huang (2003), são encontradas as subseqüências comuns únicas (Tabela 2).

A partir das subseqüências comuns únicas da Tabela 2, o usuário deve definir os parâmetros do algoritmo genético de agrupamento (AGA). Os parâmetros definidos no AGA foram: coeficiente de similaridade: Merger (HUANG, 2003), codificação do cromossomo: R3, número de grupos (módulos): 4, tamanho da população: 200 cromossomos, critério de parada: 500 gerações, taxa de elitismo: 0,005, método de geração da nova população:

probabilístico, operador de cruzamento: ponto único de grupo; probabilidade de cruzamento: 0,50, operador de mutação: *move*; probabilidade de mutação: 0,50.

O tempo de execução do algoritmo genético de agrupamento (AGA) foi de quatro segundos em computador *Pentium 4* com 1,5 GB de RAM. O melhor cromossomo

**Tabela 1.** Seqüência de operações e tempos de processamento. (VAKHARIA; WEMMERLÖV apud HUANG, 2003).

| Peça (P) | Seqüência de operações | Tempo total de processamento (minuto/lotte) |
|----------|------------------------|---|
| 1        | 1, 4, 8, 9             | 96-36-36-72                                 |
| 2        | 1, 4, 7, 4, 8, 7       | 36-120-20-120-24-20                         |
| 3        | 1, 2, 4, 7, 8, 9       | 96-48-36-120-36-72                          |
| 4        | 1, 4, 7, 9             | 96-36-120-72                                |
| 5        | 1, 6, 10, 7, 9         | 96-72-200-120-72                            |
| 6        | 6, 10, 7, 8, 9         | 36-120-60-24-36                             |
| 7        | 6, 4, 8, 9             | 72-36-48-48                                 |
| 8        | 3, 5, 2, 6, 4, 8, 9    | 144-120-48-72-36-48-48                      |
| 9        | 3, 5, 6, 4, 8, 9       | 144-120-72-36-48-48                         |
| 10       | 4, 7, 4, 8             | 120-20-120-24                               |
| 11       | 6                      | 72  |
| 12       | 11, 7, 12              | 192-150-80                                  |
| 13       | 11, 12                 | 192-60                                      |
| 14       | 11, 7, 10              | 288-180-360                                 |
| 15       | 1, 7, 11, 10, 11, 12   | 15-70-54-45-54-30                           |
| 16       | 1, 7, 11, 10, 11, 12   | 15-70-54-45-54-30                           |
| 17       | 11, 7, 12              | 192-150-80                                  |
| 18       | 6, 7, 10               | 108-180-360                                 |
| 19       | 12                     | 60  |

**Tabela 2.** Subseqüências comuns únicas. (HUANG, 2003).

| Id  | Subseqüência              |
|-----|---------------------------|
| S1  | 1 → 4                     |
| S2  | 1 → 4 → 7                 |
| S3  | 1 → 7 → 11 → 10 → 11 → 12 |
| S4  | 11 → 12                   |
| S5  | 11 → 7                    |
| S6  | 11 → 7 → 12               |
| S7  | 3 → 5                     |
| S8  | 4 → 7                     |
| S9  | 4 → 7 → 4 → 8             |
| S10 | 4 → 8                     |
| S11 | 4 → 8 → 9                 |
| S12 | 6 → 10 → 7                |
| S13 | 6 → 4 → 8 → 9             |
| S14 | 7 → 10                    |
| S15 | 7 → 8 → 9                 |
| S16 | 7 → 9                     |
| S17 | 8 → 9                     |

apresentou os grupos formados pelas subseqüências: [S1, S2, S8, S9], [S3, S4, S5, S6], [S7, S12, S14, S16], [S10, S11, S13, S15, S17]. A Tabela 3 apresenta os módulos gerados pelo AGA.

A terceira coluna da Tabela 3 mostra que o módulo M1 é formado pelos tipos de máquinas 1, 4, 7 e 8, e assim por diante. As setas indicam o fluxo de peças. As peças (Tabela 1) são designadas aos módulos de arranjo físico (Tabela 3) de acordo com o procedimento para projeto de arranjo físico modular (HUANG, 2003) e, assim, é elaborado o arranjo físico final da fábrica. O arranjo físico elaborado a partir dos módulos gerados pelo AGA (Tabela 3) foi comparado ao arranjo físico apresentado por Huang (2003). Como a função de aptidão usada no AGA (largura da silhueta média) é diferente da função usada por Huang (2003) (soma das similaridades), foi estabelecida uma base para comparação do arranjo físico final. A comparação foi feita através da distância Manhattan percorrida pelo conjunto de peças (KAUFMAN; ROUSSEUW, 1990). Se no AGA fosse usada a mesma função de Huang (2003), ou seja, a soma da similaridade, tenderia a se formar um agrupamento único, com todas as subseqüências, visto que o objetivo é a maximização da função. Ou seja, seria formado apenas um módulo, com todas as máquinas. Portanto, a função soma da similaridade não é indicada para projeto de arranjo físico modular quando se usa um processo de busca por melhores soluções.

Para o problema de K-agrupamentos, em que o número de módulos foi pré-especificado em 4, os módulos de arranjo físico gerados pelo AGA totalizaram 27 máquinas com uma distância percorrida de 76 unidades. Na solução apresentada por Huang (2003) são necessárias 24 máquinas (1 duplicada), mas a distância percorrida é de 93 unidades. Dessa forma, a solução do AGA apresenta uma redução de 18,28% da distância percorrida.

**Tabela 3.** Módulos gerados pelo AGA para K-agrupamentos. (ARGOUD, 2007).

| Id. módulo | Grupo de subseqüência comum | Módulo de arranjo físico   |
|------------|-----------------------------|----------------------------|
| M1         | S1, S2, S8, S9              | 1 → 4 → 8<br>↓<br>7        |
| M2         | S3, S4, S5, S6              | 1 → 7 ↔ 11 ↔ 10<br>↓<br>12 |
| M3         | S7, S12, S14, S16           | 6 → 10 ↔ 7 → 9<br>3 → 5    |
| M4         | S10, S11, S13, S15, S17     | 6 → 4 → 8 → 9<br>↑<br>7    |

A solução gerada pelo AGA apresenta uma situação ideal, onde é possível a duplicação de quatro máquinas. Esta duplicação simplifica o fluxo de peças pela fábrica. Entretanto, quando a duplicação possui um alto custo, há necessidade de intervenção do projetista com o objetivo de minimizá-la. Assim foi analisado um arranjo físico alternativo, no qual não foi permitida a duplicação de máquinas. Para isso, máquinas do mesmo tipo em módulos diferentes foram posicionadas de modo a formar seções funcionais. No caso de quebra de máquina ou mudança na demanda, a peça pode ser transferida entre as máquinas do mesmo tipo. Este arranjo físico possui implicações em termos de designação das seqüências aos módulos, de distância percorrida e de carga de máquina (ARGOUD, 2007). No arranjo sem a duplicação, a distância percorrida pelo conjunto de peças foi de 79 unidades. Ou seja, houve um aumento de 3,8% em relação à solução gerada pelo AGA, mas por outro lado, não foi necessário duplicar máquinas. Ainda assim, o arranjo físico alternativo, comparado à solução de Huang (2003), apresentou uma redução de 15% na distância total percorrida.

Para o problema de agrupamento automático, no qual não foi especificado o número de módulos, o AGA gerou 7 módulos (Tabela 4) de arranjo físico, totalizando 30 máquinas e uma distância percorrida de 74 unidades.

Essa solução (Tabela 4) fez com que o fluxo de peças se tornasse mais simples e facilitou a programação da fábrica, estando assim em sintonia com o projeto de arranjo físico modular. Por outro lado, em termos de duplicação de máquinas, este ganho pode não se justificar. Com isso, a partir da solução gerada pelo AGA, também foram analisados caminhos para reduzir a duplicação de máquinas. Na solução com alternativas à duplicação de máquinas houve aumento do fluxo intermodular e os

**Tabela 4.** Módulos gerados pelo AGA para agrupamento automático. (ARGOUD, 2007).

| Id. módulo  | Grupo de subseqüência comum | Módulo de layout agregado  |
|-------------|-----------------------------|----------------------------|
| M1          | S1, S2                      | 1 → 4 → 7                  |
| M9= M2 + M3 | S3, S4, S5, S6              | 1 → 7 ↔ 11 ↔ 10<br>↓<br>12 |
| M4          | S7                          | 3 → 5                      |
| M5          | S8, S9                      | 4 ↔ 7<br>↓<br>8            |
| M6          | S10, S11, S13               | 6 → 4 → 8 → 9              |
| M7          | S12, S14, S16               | 6 → 10 ↔ 7 → 9             |
| M8          | S15, S17                    | 7 → 8 → 9                  |

módulos foram descaracterizados. O número de módulos diminuiu de 7 para 6, a necessidade de máquinas de 30 para 25, e a distância total percorrida pelo conjunto de peças de 74 para 73 unidades. Algumas peças saíram do seu fluxo ideal e provocaram complexidade na programação dos módulos.

## 6 Conclusões

Este trabalho apresentou uma nova forma de se gerar módulos de arranjo físico em projeto de arranjo físico modular. Os módulos são gerados a partir do agrupamento de subsequências de operações de peças utilizando um algoritmo genético de agrupamento (AGA). Para isso foram implementadas três medidas de similaridade para comparação de seqüências de operações, foram desenvolvidas três formas de codificação do cromossomo, um procedimento para gerar a população inicial, dois operadores de cruzamento, três operadores de mutação, e aplicados dois processos de geração da nova população. Essas funcionalidades demonstram a flexibilidade do AGA, que permite ao usuário interagir com o método de solução. Além disso, o AGA é capaz de resolver tanto problemas de  $k$ -agrupamentos como de agrupamento automático. No primeiro, o usuário define o número desejado de módulos e, no segundo, o número de módulos é uma variável de decisão do problema. O exemplo apresentado mostra que, para o problema em que o usuário define o

número de módulos, houve um ganho importante, em termos de distância percorrida pelas peças, comparado à solução de Huang (2003). Para o problema de agrupamento automático, o fluxo de peças foi simplificado e a distância reduzida, mas para isso houve necessidade de duplicação de máquinas. Portanto, o projetista deve analisar o *trade-off* existente entre o custo da duplicação e o ganho pela redução da distância percorrida, e ainda, as implicações quanto ao fluxo de materiais e a programação da fábrica.

Em relação a trabalhos futuros, uma pesquisa importante é sobre um estudo de caso prático com dados obtidos diretamente da indústria, de forma que se possa comparar um arranjo físico já existente com aquele resultante da aplicação do AGA na geração dos módulos de arranjo físico. Neste estudo prático é interessante que o conjunto de dados seja relativamente grande com, por exemplo, cinquenta subsequências comuns únicas.

Outra pesquisa relevante é a avaliação do arranjo físico final por medidas de desempenho dinâmicas como, por exemplo, a quantidade de estoque em processo, ou ainda, o tempo de passagem das peças pelo sistema. Neste caso, a simulação de eventos discretos com uso de medidas dinâmicas é uma ferramenta importante e indicada para análise do desempenho do arranjo físico.

## A clustering genetic algorithm for the formation of facility layout modules

### Abstract

*The modular facility layout consists of grouping machines into modules according to subsequences of operations common to a set of parts. In the method proposed by Huang (2003), facility layout modules are generated by cluster analysis. This paper introduces an alternative way to generate layout modules through a clustering genetic algorithm (CGA). The CGA allows the user to specify the number of modules he wants ( $K$ -grouping problem) and to optimize the number of modules (automatic clustering problem). An important feature of the genetic algorithm is its flexibility, which enables the user to interact with the problem-solving method by presenting him with a choice of chromosome codes, measures of similarity to compare sequences of operations, and methods for generating new populations. To this end, two crossover operators and three mutation operators were developed. In the example presented here, the layout modules generated by the CGA and the plant's final physical layout were compared with those of Huang (2003). This comparison revealed a significant reduction in the total distance covered by the set of parts, demonstrating the effectiveness of the clustering genetic algorithm. Thus, the results indicated that the clustering genetic algorithm is an alternative for generating layout modules in the design of modular facility layouts.*

**Keywords:** Modular facility layout. Genetic algorithms. Clustering. Manufacturing.



## Referências bibliográficas

- ARGOUD, A. R. T. T. **Procedimento para projeto de arranjo físico modular em manufatura através de algoritmo genético de agrupamento**. São Carlos, 2007. Tese - (Doutorado), Escola de Engenharia de São Carlos, Universidade de São Paulo.
- ASKIN, R. G.; ZHOU, M. Formation of independent flow-line cells based on operation requirements and machine capabilities. **IIE Transactions**, v.30, n.4, p.319-329, 1998.
- BENJAAFAR, S.; HERAGU, S. S.; IRANI, S. A. Next generation factory layout: research challenges and recent progress. **Interfaces**, v.32, n.6, p.58-76, 2002.
- COLE, R. M. **Clustering with genetic algorithms**. Australia, 1998. Thesis - (Master of Science), Department of Computer Science, University of Western Australia.
- GARAI, G.; CHAUDHURI, B. A novel genetic algorithm for automatic clustering. **Pattern Recognition Letters**, v.25, n.2, p.173-187, 2004.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. 2 ed. USA: Addison-Wesley, 1989.
- GONÇALVES-FILHO, E. V.; GORGULHO-JÚNIOR, J. H. C.; ARGOUD, A. R. T. T. Algoritmo genético para formação de células de fabricação. In: Encontro Nacional de Engenharia de Produção, 24, 2004. **Anais...** Florianópolis: UFSC, 2004. (CD-ROM)
- GONÇALVES-FILHO, E. V.; TIBERTI, A. J. A group genetic algorithm for the machine cell formation problem. **International Journal of Production Economics**, v.102, n.1, p.1-21, 2006.
- HRUSCHKA, E. R. **Algoritmos genéticos de agrupamento para extração de regras de redes neurais**. Rio de Janeiro, 2001. 139f. Tese - (Doutorado em Ciências em Engenharia Civil) - COPPE, Universidade Federal do Rio de Janeiro.
- HRUSCHKA, E. R.; EBECKEN, N. F. F. A genetic algorithm for cluster analysis. **Intelligent Data Analysis**, v.7, n.1, p. 15-25, 2003.
- HUANG, H. **Facility layout using layout modules**. Columbus, 2003. Thesis - (PhD in Industrial and Systems Engineering), The Ohio State University.
- IRANI, S. A.; HUANG, H. Layout modules: a novel extension of hybrid cellular layouts. In: **ASME INTERNATIONAL MECHANICAL ENGINEERING CONGRESS & EXPOSITION, WINTER ANNUAL MEETING OF THE ASME**, Anaheim, CA, November 15-20, p.1-7, 1998.
- IRANI, S. A.; HUANG, H. Cascading flowlines and layout modules: practical strategies for machine duplication in facility layouts. **International Journal of Flexible Manufacturing Systems**, v.17, n.2, p.119-149, 2006.
- KAUFMAN, L.; ROUSSEUW, P. J. **Finding groups in data: an introduction to cluster analysis**. New York: John Wiley & Sons, 1990. Wiley series in probability and mathematical statistics.
- LACERDA, E. G. M.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. Um tutorial sobre algoritmos genéticos. **RITA – Revista de Informática Teórica e Aplicada**, Porto Alegre, v. 9, n. 3, p.7-39, 2002.
- MARCONI, M. A.; LAKATOS, E. M. **Técnicas de pesquisa**. São Paulo: Editora Atlas, 1985.
- TAM, K. Y. An operation sequence based similarity coefficient for part families formations. **Journal of Manufacturing Systems**, v.9, n.1, p.55-68, 1990.
- TOMPKINS, J. A. et al. **Facilities planning**. 3 ed. USA: John Wiley & Sons, Inc., 2003. 750p.
- TSENG, L.; YANG, S. B. A genetic approach to the automatic clustering problem. **Pattern Recognition**, v.34, n.2, p.415-424, 2001.

## Sobre os autores

**Ana Rita Tiradentes Terra Argoud**

**Eduardo Vila Gonçalves Filho**

**Alexandre José Tiberti**

Departamento de Engenharia Mecânica, Escola de Engenharia de São Carlos, Universidade de São Paulo - USP, Av. Trabalhador São-carlense, 400, CEP 13560-970, São Carlos, SP, Brasil, e-mails: ar.terra@terra.com.br; evila@sc.usp.br; atiberti@sc.usp.br

**Agradecimentos:** Agradecemos ao CNPq pela concessão de bolsa de estudo para realização desta pesquisa.

Recebido em 05/9/2007  
Aceito em 28/6/2008