

Algoritmo Genético Construtivo aplicado ao projeto de células de manufatura

Geraldo Ribeiro Filho
Universidade de Mogi das Cruzes
Av. Cândido Xavier Almeida Souza, 200
08780-911 - Mogi das Cruzes - SP

Luiz Antonio N. Lorena
LAC/INPE- Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas 1758 - São José dos Campos - SP

Resumo

Uma aplicação importante da tecnologia de grupo é o desenvolvimento de um sistema de manufatura celular em que partes similares são agrupadas em famílias e máquinas são agrupadas em células. A célula ideal é independente, isto é, famílias de partes são completamente produzidas dentro da célula. Tais sistemas proporcionam benefícios como simplificação de controle, de implementação e de automação, redução dos tempos de preparação, do tempo de entrada de material e saída de produto (throughput), manejo de material, bem como contribui para o aumento da qualidade do produto final. Apresenta-se neste trabalho a aplicação do Algoritmo Genético Construtivo (AGC) ao problema de formação de células de manufatura. A representação considera simultaneamente o arranjo de partes e ferramentas, como um problema de p-medianas, usando como distância o coeficiente de Jaccard. O AGC foi testado em problemas gerados aleatoriamente e em duas instâncias disponíveis na literatura. Os resultados são promissores.

Palavras chave: Algoritmo Genético Construtivo, Projeto de células de manufatura, Problema das p-medianas.

1. Introdução

A competitividade internacional e conseqüente necessidade de respostas rápidas à demanda do mercado têm levado muitas empresas a considerar abordagens não tradicionais para o projeto e controle de sistemas de manufatura. Uma abordagem é a aplicação de “tecnologia de grupo”, caracterizada pela exploração de similaridades nas atividades ligadas à produção. Em essência, a tecnologia de grupo tenta decompor os sistemas de manufatura em vários subsistemas, ou grupos, controláveis.

Uma aplicação importante da tecnologia de grupo é o desenvolvimento de um sistema de manufatura celular em que partes similares são agrupadas em famílias e máquinas são agrupadas em células. A célula ideal é independente, isto é, famílias de partes são completamente produzidas dentro da célula. Tais sistemas proporcionam benefícios como simplificação de controle, de implementação e de automação, redução dos tempos de preparação, do tempo de entrada de material e saída de produto (throughput), manejo de material, bem como contribui para o aumento da qualidade do produto final.

Os métodos para formação células de manufatura podem ser classificados como orientados pelo projeto ou pela produção. Enquanto os métodos orientados por

projeto agrupam partes baseando-se em características de seu projeto, os métodos orientados por produção o fazem baseando-se nos processos requeridos para sua produção.

Este trabalho tem foco em técnicas de formação de células orientadas pela produção. Os primeiros trabalhos nesta área foram feitos por Mitrofanov [23,24] e Burbidge[2,3]. A análise do fluxo de produção, de Burbidge [3] é uma das primeiras e mais reconhecidas metodologias associadas com tecnologia de grupo. Análise de fluxo de componentes por El-Essawy e Torrance [8] é considerada por muitos autores como equivalente aos trabalhos de Burbidge. O objetivo destas técnicas é obter células de máquinas independentes, minimizando o movimento entre as células.

Há muitos métodos para tratar o problema, e uma boa parte deles operam sobre uma matriz partes/máquinas cujos elemento são zeros e uns, indicando quais máquinas são usadas na produção de cada parte. Sendo A a matriz 0-1 assim descrita, as linhas correspondem às partes e as colunas às máquinas (ou vice-versa) e se $a_{ij} = 1$ então a parte i necessita da máquina j para sua produção (*figura 1*). Os algoritmos manipulam linhas e colunas da matriz tentando produzir pequenos blocos de uns agrupados, o que os torna basicamente algoritmos para problemas de agrupamento, ou *clustering* (*figura 2*).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0
1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0
2	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
3	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0
4	0	1	0	1	1	0	1	0	1	0	0	1	0	0	0
5	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0
6	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
8	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
9	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0

Figura 1: Matriz A de partes e máquinas (original)

	1	3	6	8	11	0	2	4	5	9	7	10	12	13	14
1						1	1	1		1					1
3							1	1	1	1					
9							1	1		1					
2	1				1	1									
4	1	1	1	1	1			1							
5	1	1			1										
6	1	1	1	1	1										
0											1	1	1	1	
7												1		1	1
8											1		1	1	1

Figura 2: Matriz A de partes e máquinas (em células)

Chandrasekharan e Rajagopalan [6] e Venugopal e Narendran [29] apresentaram análises na matriz 0-1 de modo a extrair propriedades para criação de algoritmos de formação de células. Algoritmos como os descritos acima podem ser vistos nos trabalhos de McCormick [21], King [13,14,15], Chan e Milner [4]. Chu [7] apresenta um trabalho comparativo sobre três técnicas deste mesmo tipo.

No entanto, uma grande variedade de abordagens têm sido tentadas no tratamento deste problema. Técnicas de agrupamento hierárquico foram estudadas, sejam do tipo divisivo como em Stanfel [27], ou aglomerativo como em McAuley [20]. Agrupamento não hierárquico também foi estudado por Chandrasekharan e Rajagopalan [5]. Técnicas de agrupamento baseadas em grafos foram estudadas e aplicadas desde os primeiros trabalhos de Rajagopalan e Batra [25]. Mesmo técnicas de Inteligência Artificial como redes neurais artificiais, nos trabalhos de Malave e Ramchandran [19], e lógica fuzzy, nos trabalhos de Xu e Wang [30]. Heurísticas mais genéricas como Simulated Annealing têm sido empregadas, como se vê nos trabalhos de Boctor [1] e Venugopal et al. [28,28]. Joines [12] desenvolveu uma abordagem de algoritmo genético para solucionar o problema de projeto de células de manufatura. O algoritmo foi testado sobre conjuntos de dados da literatura e foi capaz de encontrar soluções ao menos tão boas quanto aquelas já existentes.

Neste trabalho apresenta-se uma abordagem que aplica o Algoritmo Genético Construtivo (AGC) ao problema de formação de células de manufatura.

2. Algoritmo Genético Construtivo - resumo

Recentemente, fizemos algumas experiências com aplicação do Algoritmo Genético Construtivo (AGC) aplicado ao problema de geração de células de manufatura, na sua forma mais simples, baseada em uma representação matricial de partes e máquinas.

Ao contrário dos Algoritmos Genéticos (AG) tradicionais, o AGC trabalha com uma população inicial formada apenas por *esquemas*, isto é, partes da representação completa de uma solução do problema. A teoria do esquema foi por algum tempo o ponto central nos AG clássicos [10, 11, 17, 18, 22] e hoje tem sido menos explorada. Como um esquema não representa uma solução completa do problema, o AGC usa uma medida de adaptação proporcional do esquema à população. A cada nova iteração do processo a população aumenta com a inclusão de novos indivíduos gerados com a combinação e mutação de esquemas da população atual. Os bons esquemas são preservados dentro dos novos esquemas gerados, que geralmente se aproximam de representações completas da solução do problema. Também a cada iteração do processo, um parâmetro evolutivo é usado na eliminação de indivíduos que não satisfazem um critério de permanência da população. Uma cópia da melhor solução encontrada até o momento é mantida salva, e o processo termina quando um limite de iterações previamente estipulado é atingido ou eventualmente a população fique vazia.

O AGC, mesmo antes de ter esta denominação, foi inicialmente estudado por Lorena e Lopes [16], foi aplicado a alguns problemas de otimização combinatória e,

mais recentemente, foi apresentado como uma heurística geral para problemas de otimização na tese de doutorado de J.C. Furtado (veja tese em <http://www.lac.inpe.br/~lorena/teseJC/CGA-tese.ps>). Com base na representação utilizada por Furtado e Lorena [9] em sua aplicação ao problema das p -medianas (veja o trabalho em <http://www.lac.inpe.br/~lorena/sbpo98/AGC-clust.ps>), que pode ser generalizada para outros problemas de agrupamento, foi feita uma adaptação ao problema de formação de células de manufatura. Com a especificação a priori do número de células desejado, a idéia é formar agrupamentos de máquinas e partes simultaneamente, alterando a posição das linhas e das colunas na matriz na matriz 0-1 que representa o problema.

A analogia com o problema das p -medianas ocorre pelo fato de que o algoritmo tenta encontrar p partes (ou linhas da matriz), de modo que as demais partes (linhas) possam ser associadas a uma delas de acordo com uma medida de "distância" entre as partes (ou linhas), formando agrupamentos de partes com a menor soma de "distâncias" possível. O mesmo pode ser aplicado às máquinas (ou colunas da matriz), simultaneamente.

2.1. Representação

Como *esquema* e *estrutura* foi utilizada uma *cadeia* de $n+m$ símbolos, onde n é o número de partes (ou linhas da matriz) e m é o número de máquinas (ou colunas da matriz). Cada uma das duas porções do esquema evolui independente da outra. Se forem considerados os símbolos: 1 para indicar uma máquina ou parte mediana, 0 para indicar uma parte ou máquina não mediana e associada a uma mediana, e # para indicar partes ou máquinas ainda não associadas a nenhuma mediana (temporariamente fora do problema), ambas as porções do esquema (partes/máquinas) possuem exatamente o mesmo número de uns, equivalente ao número de grupos que se deseja formar, e o restante das posições no esquema terá ou o símbolo 0, ou o símbolo #. Uma estrutura completa para o problema terá apenas 1's e 0's.

Considerando que desejamos formar três células de manufatura, um esquema para um problema de 11 partes e 14 máquinas poderia ser representado por

$$S_k = (0,1,\#,1,0,\#,1,0,\#,0,0 / 0,0,1,\#,0,\#,1,\#,0,0,0,\#).$$

Onde os 11 primeiros símbolos se referem a partes e os 14 símbolos restantes se referem a máquinas. Seja $V_1^p(S_k) = \{2,4,7\}$, o conjunto de índices das *partes medianas* (representadas por 1 nas cadeias) de cada grupo que será formado, e $V_2^p(S_k) = \{1,5,8,10,11\}$ o conjunto de índices das *partes não medianas* (representadas por 0 nas cadeias). Analogamente para as máquinas, $V_1^m(S_k) = \{3,7,9\}$, e $V_2^m(S_k) = \{1,2,6,11,12,13\}$. De uma forma geral sejam $V_1^p(S_k) = \{\zeta_1^p, \dots, \zeta_p^p\}$ e $V_1^m(S_k) = \{\zeta_1^m, \dots, \zeta_p^m\}$.

Os grupos de partes e máquinas serão então formados associando os elementos cujos índices estão nos conjuntos $V_2^p(S_k)$ aos elementos cujos índices estão nos

conjuntos $V_1^p(S_k)$ (o mesmo para $V_2^m(S_k)$ e $V_1^m(S_k)$). Para a associação de partes e máquinas com sua mediana mais “próxima”, é necessária uma medida de similaridade, ou "distância", entre as linhas e entre as colunas da matriz. Foi utilizada uma medida conhecida como *coeficiente de Jaccard*. O coeficiente de similaridade de Jaccard entre seqüências binárias é definido como o número de posições com valor 1 em ambas as seqüências, dividido pelo número de posições com valor 1 em ambas ou apenas em uma das seqüências. Esse coeficiente pode ser convertido em medida de distância subtraindo-o de um.

Após as associações são identificados os grupos $C_1^p(S_k), C_2^p(S_k)$ e $C_3^p(S_k)$ para as partes, e $C_1^m(S_k), C_2^m(S_k)$ e $C_3^m(S_k)$.

2.2. População inicial

Uma população inicial foi gerada aleatoriamente com apenas 20% das linhas e das colunas compondo os esquemas com símbolos 0 e exatamente p (número de células) símbolos 1.

2.3. Seleção e recombinação

A população foi mantida ordenada de acordo com um critério que privilegia aqueles esquemas mais próximos de uma solução completa e com melhor qualidade, isto é, menos "distância" entre os componentes de cada agrupamento de máquinas ou partes.

O método de seleção para recombinação utiliza uma estratégia em que são selecionados um esquema entre os melhores da população (*base*) e outro em toda a população (*guia*). Se o primeiro selecionado já for uma solução completa, sem zeros, aplica-se apenas uma busca local que pode ser vista como um tipo de mutação. Neste caso, escolhe-se aleatoriamente uma das medianas e a trocamos de lugar, deixando-a na posição em que houver maior melhoria na solução.

Caso a estrutura base for um esquema, a recombinação entre ambos é usada como uma espécie de cruzamento, gerando um novo indivíduo na população, procurando preservar as informações do esquema base e complementando-as com as informações do guia. O número de uns (medianas) de ser sempre mantido em cada uma das duas porções dos esquema (partes/máquinas).

Recombinação

$S_{base}(j) = \#$ e $S_{guia}(j) = \#$ então $S_{novo}(j) \leftarrow \#$
 $S_{base}(j) = 1$ e $S_{guia}(j) = 1$ então $S_{novo}(j) \leftarrow 1$
 $S_{base}(j) = 0$ e $S_{guia}(j) = 0$ então $S_{novo}(j) \leftarrow 0$
 $S_{base}(j) = 1$ e $S_{guia}(j) = \#$ então $S_{novo}(j) \leftarrow 1$
 $S_{base}(j) = 0$ e $S_{guia}(j) = \#$ então $S_{novo}(j) \leftarrow 0$
 $S_{base}(j) = \#$ e $S_{guia}(j) = 0$ então $S_{novo}(j) \leftarrow 0$
 $S_{base}(j) = \#$ ou 0 e $S_{guia}(j) = 1$ então

$S_{novo}(j) \leftarrow 1$ e $S_{novo}(i) \leftarrow 0$ para algum $S_{novo}(i)=1$
 $S_{base}(j)=1$ e $S_{guia}(j)=0$ então
 $S_{novo}(j) \leftarrow 0$ e $S_{novo}(i) \leftarrow 1$ para algum $S_{novo}(i)=0$

A cada iteração do processo várias novas soluções foram inseridas na população e a eliminação dos piores indivíduos foi feita com base nos princípios do algoritmo.

2.4. Funções de avaliação

Cada estrutura ou esquema S_k da população é mapeado no \mathcal{R}_+ de acordo com duas funções, $f(S_k)$ e $g(S_k)$, $f(S_k) \leq g(S_k)$, além de um limitante superior g_{max} da função g calculado no início do processo como sendo o valor de g para uma solução completa do problema gerada aleatoriamente (veja figura 3).

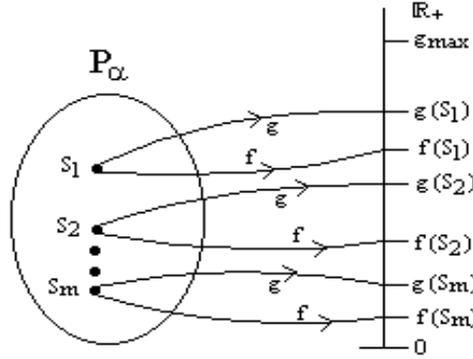


Figura 3: Representação gráfica das funções de avaliação.

Após a formação dos grupos $C_1^p(S_k), C_2^p(S_k), \dots, C_p^p(S_k)$ e $C_1^m(S_k), C_2^m(S_k), \dots, C_p^m(S_k)$, as funções foram particularizadas para o problema como:

$$g(s_k) = \sum_{i=1}^p \sum_{j \in C_i^p(S_k)} \mu_{\zeta_{ij}} + \sum_{i=1}^p \sum_{j \in C_i^m(S_k)} \mu_{\zeta_{ij}},$$

$$f(s_k) = \sum_{i=1}^p \lambda_i^p \cdot [C_i^p(S_k) - 1] + \sum_{i=1}^p \lambda_i^m \cdot [C_i^m(S_k) - 1],$$

onde $\lambda_i^p = \min_{j \in C_i^p(S_k)} \{\mu_{\zeta_{ij}}\}$ e $\lambda_i^m = \min_{j \in C_i^m(S_k)} \{\mu_{\zeta_{ij}}\}$ representam a menor distância que foi

atribuída ao grupo i , e $|C_i(S_k)|$ é a cardinalidade do conjunto $C_i(S_k)$.

2.5. Evolução

A eliminação de um indivíduo da população depende do valor do parâmetro de evolução α , que inicialmente tem valor zero e a cada iteração é lentamente acrescido. Um indivíduo S_k é removido da população quando

$\alpha \geq \frac{dg_{\max} - d_i g(S_k)}{d(g_{\max} - g(S_k))} = \delta(S_k)$, onde $d_k = \frac{g(S_k) - f(S_k)}{g(S_k)}$, e o desvio percentual esperado em relação a g_{\max} é dado por d .

O parâmetro α é considerado um elemento de controle da evolução da população inicial. Considera-se variar α a partir de zero, em pequenos intervalos de evolução, onde novas gerações são criadas. Cada estrutura s_k , quando criada recebe um valor $\delta(S_k)$ que indica o seu tempo de permanência na população. Quanto menor o seu valor d_k , melhor será sua adaptação, e poderá permanecer mais tempo na população, e com maior probabilidade de participar dos operadores recombinação e mutação. Quanto maior a diferença $g_{\max} - g(s_k)$, menor seu tempo de permanência na população.

Nossa condição de parada será quando a população for completamente eliminada ou para um determinado número de gerações. No processo a melhor estrutura será preservada.

3. Testes computacionais

Infelizmente, tem sido grande a dificuldade encontrar instâncias disponíveis na literatura, e sendo assim foram aleatoriamente geradas algumas instâncias de tamanhos variados. Da literatura foram usadas nos testes apenas duas instâncias de dimensões 20×35 (Burbidge [3]) e 40×100 (Chandrasekharan Rajagopalan [5]).

Como *medida de performance* considerou-se um coeficiente que considera a quantidade de zeros dentro dos agrupamentos e a quantidade de uns fora deles, em relação ao total de uns da matriz que representa o problema:

$$Coef = \frac{e - e_0}{e + e_1}$$

onde:

e	=	número de 1's na matriz
e_0	=	número de 0's dentro das células
e_1	=	número de 1's fora das células

O AGC procurou a maximizar tal coeficiente nas matrizes processadas.

Para as instancias geradas aleatoriamente foram especificados o numero de partes, número de máquinas, *densidade intra-celular* (WCD) e *densidade inter-celular* (ICD). Como WCD entende-se a razão entre o numero de 1's dentro da célula e o tamanho da célula. Como ICD entende-se a razão entre o numero de 1's fora de qualquer célula e o numero de posições fora de qualquer célula. Inicialmente a matriz é gerada com as células definidas da forma descrita, para em seguida ser perturbada com várias trocas de linhas e colunas de forma aleatória. Esta última matriz é então usada nos testes.

Para testes foi escrito um programa em C, sem grandes preocupações com otimização de código, e usada uma estação SUN Ultra30. Alguns resultados são

ilustrados nas tabelas abaixo, tanto para as instâncias geradas aleatoriamente quanto para as duas tiradas da literatura. Os parâmetros do AGC nos testes foram: *incremento de $\alpha = 0.01$ e $d = 0.1$.*

Instância	Part/maq	Cel	WCD	ICD	Coef _{ori}	Iter.	Tempo (s)	Coef _{AGC}	Gap (%)
t-10-15-3	10/15	3	0.8	0.02	0.727273	150	1	0.727273	-
						150	2	0.727273	-
						150	6	0.727273	-
t-15-40-4	14/40	4	0.8	0.02	0.717391	200	19	0.717391	-
						200	23	0.717391	-
						200	20	0.717391	-
t-20-25-5	20/25	5	0.8	0.02	0.708333	200	14	0.708333	-
						200	34	0.708333	-
						200	31	0.708333	-
t-20-35-4	20/35	4	0.8	0.02	0.716049	150	26	0.716049	-
						150	20	0.716049	-
						150	17	0.716049	-

Tabela 1: Testes com instâncias geradas aleatoriamente

Instancia	Part/maq	Cel.	Coef _{ori}	Iter.	Tempo (s)	Coef _{AGC}	Gap (%)
Burbidge	20/35	4	0.7571	150	25	0.688406	9.1
				150	27	0.688406	9.1
				150	25	0.688406	9.1
Chandra	40/100	10	0.8403	150	1413	0.556468	33.8
				150	2482	0.658487	21.6
				150	986	0.292871	65.1

Tabela 2: Testes com instâncias da literatura

Os resultados da *tabela 1* mostram que o AGC se comportou bem para os problemas gerados aleatoriamente, conseguindo, em poucos segundos, reproduzir a forma original das matrizes geradas. Os resultados da *tabela 2* foram promissores para o problema pequeno (*Burbidge*) e não tanto para o grande (*Chandra*) significando que mais trabalho de refinamento do AGC deve ser feito nesse caso. A comparação foi feita com o melhor Coef conhecido para essas instâncias.

Novos os teste foram feitos com a instância maior (*Chandra*) usando agora os seguintes parâmetros no AGC: *incremento de $\alpha = 0.003$, $d = 0.1$ número máximo de gerações = 300* (veja *tabela 3*). Os tempos aumentaram e mais esquemas e/ou estruturas foram examinadas, conseqüentemente os resultados melhoraram, mas ainda um trabalho adicional é requerido para alcançar o melhor resultado conhecido para essa instância.

Instancia	Part/maq.	Cel.	Coef _{ori}	Iter.	Tempo (s)	Coef _{AGC}	Gap (%)
Chandra	40/100	10	0.8403	300	5960	0.772340	8.1
				300	2218	0.612774	27.1
				300	6398	0.685535	18.4

Tabela 3: Novos testes com Chandra 40/100

Foram feitos em seguida alguns testes com as instancias geradas aleatoriamente para tentar detectar alguma influência das densidades ICD (*tabela 4*) e WCD (*tabela 5*)

sobre os resultados. Pode-se notar que uma maior densidade inter-celular ICD dificultou a aplicação do AGC.

Instancia	Part/maq	Cel.	WCD	ICD	Coef _{ori}	Coef _{AGC}	Gap (%)
Ti-20-35-5-2	20/35	5	0.8	0.02	0.686567	0.686507	-
						0.686507	-
						0.686507	-
Ti-20-35-5-3	20/35	5	0.8	0.03	0.673611	0.673611	-
						0.673611	-
						0.673611	-
Ti-20-35-5-5	20/35	5	0.8	0.05	0.650602	0.650602	-
						0.438596	32.6
						0.650602	-
Ti-20-35-5-10	20/35	5	0.8	0.10	0.612613	0.591111	3.5
						0.504274	17.7
						0.591111	3.5

Tabela 4: Testes de sensibilidade ao ICD

Instancia	Part/maq	Cel.	WCD	ICD	Coef _{ori}	Coef _{AGC}	Gap (%)
Tw-20-35-5-7	20/35	5	0.7	0.02	0.521008	0.521008	-
						0.521008	-
						0.521008	-
Tw-20-35-5-8	20/35	5	0.8	0.02	0.686567	0.686567	-
						0.686567	-
						0.661765	3.6
Tw-20-35-5-9	20/35	5	0.9	0.02	0.818792	0.818792	-
						0.818792	-
						0.818792	-

Tabela 5: Testes de sensibilidade ao WCD

A figura 4 mostra o resultado da aplicação do método na uma matriz de 20x35 (Burbidge [3]).

	0	1	3	20	26	27	32	33	4	7	11	15	17	24	25	34	2	8	9	10	14	16	22	23	5	6	12	13	18	19	21	28	29	30	31			
0										1	1	1	1	1	1	1																						
2										1	1	1	1	1																								
3										1	1	1	1	1																								
9								1		1	1	1	1	1																								
13									1			1	1																									
1																	1	1	1	1	1	1	1	1	1													
4																	1	1	1	1	1	1	1	1	1													
6																	1	1	1	1	1	1	1	1	1													
11																	1	1	1	1	1	1	1	1	1													
14																	1	1	1	1	1	1	1	1	1													
5																										1												
7																										1												
8																										1												
10																										1												
15																										1												
12	1																																					
16		1																																				
17		1																																				
18		1																																				
19		1																																				

Figura 4: Matriz Processada com AGC

4. Considerações Finais

O AGC apresentou bons resultados quando aplicado ao projeto de células de manufatura. O problema foi considerado como um problema de *clustering*, mais especificamente como um problema das *p-medianas*, em que as partes e máquinas são arranjadas ao mesmo tempo para a formação das células.

O problema de formação de células de manufatura foi formulado e resolvido de várias maneiras na literatura. Essa liberdade nos deixa margem para outras aplicações do AGC, formulando o problema de maneira diferente, por exemplo, como um problema de k-coloração [26]. Isso será tentado futuramente na seqüência deste trabalho.

Agradecimentos: O trabalho do segundo autor recebeu apoio parcial do CNPq através dos projetos de números 350034/91-5, 520844/96-3, 680082/95-6 e da FAPESP nos projetos números 95/9522-0 e 96/04585-6.

5. Referencias

- [1] F. Boctor. A linear formulation of the machine-part cell formation problem. *IJPR*, 29(2):343-356, 1991.
- [2] J.L.Burbidge. Production flow analysis. *Production Engineer*, 42:742-752, 1963.
- [3] J.L.Burbidge. An introduction of group technology. In, seminar on group Technology, Turin, 1969.
- [4] H.M. Chan and D.A. Milner. Direct clustering algorithm for group formation in cellular manufacture. *JMS*, 1(1):65-74, 1982.
- [5] M.P. Chandrasekharan and R. Rajagopalan. An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *IJPR*, 24(2):451-464, 1986.
- [6] M.P. Chandrasekharan and R. Rajagopalan. Groupability: Analysis of the properties of binary data matrices for group technology. *IJPR*, 27(6):1035-1052, 1989.
- [7] C.H. Chu and M. Tsai. A comparison of three array-based clustering techniques for manufacturing cell formation. *IJPR*, 28(8):1417-1433, 1990.
- [8] I. El-Essay and J. Torrance. Component flow analysis-an effective approach to productions systems design. *Production Engineer*, 51(5):165-170, 1972.
- [9] Furtado, J. C. e Lorena, L. A N. Algoritmo Genético Construtivo na orimização de problemas combinatoriais de agrupamentos. IIIa. Oficina Nacional de Cortes e Empacotamento e XXX SBPO, Curitiba, novembro, 1998.
- [10] Goldberg, D.E., Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, MA, p. 11-172, 1989.
- [11] Holland, J.H., Adaptation in natural and artificial systems. MIT Press, p. 11-147, 1975.
- [12] J.A. Joines. Manufacturing cell design using genetic algorithms. Ms thesis, North Carolina State University, Raleigh, NC, 1993.
- [13] J.R.King. Machine-component grouping formation in group

- technology. *International Journal of management Science*, 8(2):193-199, 1980.
- [14] J.R.King. Machine-component grouping in production flow analysis: An approach using rank order clustering algorithm. *IJPR*, 18(2):213-232, 1980.
- [15] J.R.King and V. Nakornchai. Machine-component group formation in group technology: Review and extension. *IJPR*, 20(2):117-133, 1982.
- [16] L.A.N. Lorena and F.B. Lopes. A dynamic list heuristic for 2D-cutting. In *System Modelling and Organization*, ed. Chapman-Hill, 1996, p.481-488.
- [17] Lorena, L.A.N.; Lopes, L.S. *Computational experiments with genetic algorithms applied to set covering problems*. *Pesquisa Operacional*, 16, p. 41-53, 1996b.
- [18] Lorena, L.A.N.; Lopes, L.S. *Genetic algorithms applied to computationally difficult set covering problems*. *Journal of the Operational Research Society*, 48, p. 440-445, 1997.
- [19] C. O. Malave and S. Ramachandran. A neural network based design of cellular manufacturing system. *Journal of Intelligent Manufacturing*, 2:305-314, 1991.
- [20] J. McAuley. Machine grouping for efficient production. *Production Engineer*, 51(2):53-57, 1972.
- [21] W.T.McCormick, Jr., P.J. Schweitzer, and T.W.White. Problem decomposition and data reorganization by a cluster technique. *Operations Research*, 20(5):993-1009, 1972.
- [22] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, 1996.
- [23] S. P. Mitrofanov. *Scientific principles of group technology*. Technical report, Leningrad, 1959.
- [24] S. P. Mitrofanov *Scientific principles of group technology*. Technical report, London: National Lending Library, 1966.
- [25] R. Rajagopalan and J.L.Batra. Design of cellular production systems: A graph theoretic approach. *IJPR*, 13(6):567-579, 1975.
- [26] Ribeiro Filho, G. *Uma heurística construtiva para coloração de grafos*. Master thesis, INPE, 1997.
- [27] L. E. Stanfel. Machine clustering for economic production. *Engineering Costs and Production Economics*, 9:73-81, 1985.
- [28] V. Venugopal and T. T. Narendran. Cell formation in manufacturing systems through simulated annealing: an experimental evaluation. *EJOR*, 63(3):409-422, 1992.
- [29] V. Venugopal and T. T. Narendran. Design of cellular manufacturing systems based on asymptotic forms of a boolean matrix. *EJOR*, 67:405-417, 1993.
- [30] H.Xu and H. P. Wang. Part family formation for gt applications based on fuzzy mathematics. *IJPR*, 27(9):1637-1651, 1989.