



FACULDADE DE TECNOLOGIA, CIÊNCIAS E EDUCAÇÃO

Graduação

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uso de algoritmo genético aplicado ao problema de programação da produção

Jiezes Guides de Araujo Neto
Adinovam Henriques de Macedo Pimenta (Orientador)

RESUMO

Muitas pesquisas são realizadas com o objetivo de minimizar o tempo total da produção de produtos em uma programação da produção. O problema da programação da produção é considerado um problema combinatório pertencente à classe NP-Hard dos problemas computacionais. Este trabalho propõe um método simples que utiliza Algoritmo Genético a fim de gerar a programação da produção, obtendo uma programação reativa, visando minimizar o valor de makespan. Resultados experimentais mostraram que algoritmo proposto foi capaz de trazer resultados de makespan melhores que o algoritmo comparado devido ao direcionamento na busca vinda a partir da representação cromossômica da solução.

Palavras-chave: Programação da Produção. Algoritmos Genéticos. Otimização.

ABSTRACT

Many researches are carried out with the objective of minimizing the total time of production of products in a production schedule. The problem of production scheduling is considered a combinatorial problem belonging to the NP-Hard class of computational problems. This work proposes a simple method that uses Genetic Algorithm in order to generate the production schedule, obtaining a reactive programming, aiming to minimize the value of makespan. Experimental results showed that the proposed algorithm was able to bring better makespan results than the algorithm compared due to the search direction from the chromosomal representation of the solution.

Keywords: Production Scheduling. Genetic Algorithms. Optimization.

Introdução

O atual cenário industrial, cujas características exigem alta produtividade com elevado padrão de qualidade e baixo custo, e os avanços das técnicas e tecnologias de processos utilizadas por elas, levam a um aumento da quantidade de informações coletadas a respeito dos processos produtivos e, para aumentar a competitividade das indústrias, gerou-se a necessidade de tratar essas informações.

Para isso, surgiram os Sistemas de Execução da Manufatura (MES, do inglês *Manufacturing Execution Systems*), que integram os Sistemas de Chão de Fábrica com os sistemas corporativos. Os Sistemas de Execução da Manufatura são Sistemas de Informação integrados, ou seja, são sistemas que tornam as informações sobre o sistema de manufaturas disponíveis em tempo real. Eles acompanham o processo produtivo, desde a emissão da ordem de serviço até que o produto saia da linha de produção e esteja pronto para ser entregue ou armazenado (QUI; ZHOU, 2004).

Segundo Reddy e Rao (2006), as indústrias de manufatura têm sofrido enormes transformações nos últimos anos devido à utilização crescente de tecnologias de processos, tais como: automação de máquinas, ferramentas de controle numérico computadorizado, robôs industriais, sistemas de manipulação e transporte de materiais.

Esse contexto evidencia uma mudança no enfoque que coloca a manufatura não mais como uma simples área de suporte, mas como integrante do grupo de fatores determinantes para a competitividade. Nesse novo ambiente, a manufatura deve ser avaliada em suas dimensões estratégicas, como: custo, qualidade, serviço e flexibilidade (MONTEVECHI; MORANDIN; MIYAGI, 2007).

Segundo Slack et al. (2002), a flexibilidade significa a capacidade de mudar a operação, ou seja, a possibilidade de alteração do que a operação faz, como faz ou quando faz. Assim sendo, um sistema de manufatura que possua certa flexibilidade deve proporcionar pelo uma das seguintes características: possibilitar a realização da produção em roteiros distintos; possibilitar que uma mesma máquina possa realizar diferentes operações sobre qualquer produto, etc.

Para fazer um melhor proveito da flexibilidade e utilizar de forma eficiente

os recursos do sistema produtivo, é necessário realizar uma boa programação da produção, que corresponde à definição da ordem e momento da alocação de recursos na sequência correta, além da alocação das máquinas e a alocação do sistema de transporte, cujo resultado é a conclusão do conjunto de tarefas, ou seja, a finalização da produção.

De acordo com Zhou et al. (2001) e Chan et al. (2006), a programação da produção é um problema de otimização combinatória, classificado como NP-Difícil. Portanto, em uma tarefa como a programação da produção, testar todas as suas possíveis combinações em um tempo hábil é uma tarefa que acarreta alto custo computacional, que cresce exponencialmente com o aumento do tamanho do problema.

Uma das dificuldades encontradas na programação da produção é o uso simultâneo de máquinas e sistemas de transporte. O sistema de transporte em um Sistema de Manufatura Flexível (FMS, do inglês, *Flexible Manufacturing System*) geralmente é constituído de Veículos Auto-Guiados (AGV – do inglês *Automated Guided Vehicle*). Segundo Slack et al. (2002), AGVs são veículos pequenos e autônomos e que movem materiais entre vários pontos do chão de fábrica.

Em Sistemas de Manufatura Flexíveis com recursos compartilhados, geralmente um produto possui vários roteiros e é manufaturado em máquinas distintas, não sequenciais, separadas geograficamente umas das outras, necessitando, assim, de um meio de transporte para movimentá-los. Nesse caso, o sistema de transporte pode ser constituído por AGVs, carregadeiras, ou outros veículos não automatizados que necessitem de um operador, etc.

Mesmo que a programação da produção seja realizada da forma mais otimizada possível, não se pode garantir que essa programação seja cumprida, pois o chão de fábrica está sujeito a variáveis que podem interferir no processo produtivo, atrasando a produção, ou até mesmo paralisando o sistema produtivo. Essas variáveis são conhecidas como ocorrências (eventos) não programadas, que podem ocorrer devido à quebra de máquinas, quebra do AGV, falta de matéria prima, falta de operadores, etc.

Na tentativa de minimizar os efeitos dessas ocorrências, é necessário que a fábrica seja monitorada e que as ocorrências dos eventos inesperados sejam tratadas e contornadas sem que o sistema produtivo seja prejudicado, mantendo

a produção em alta. Para isso, é realizada a reprogramação da produção, também conhecida como programação reativa da produção.

Segundo Li e Ierapetritou (2008), programação reativa da produção é a atividade responsável por monitorar a execução da programação previamente planejada, lidando com eventuais problemas em sua execução.

Os eventos não esperados trazem como consequência a parada do sistema produtivo, maior tempo de ociosidade das máquinas e até mesmo o não cumprimento da entrega dos produtos nas datas devidas, atrapalhando, assim, o relacionamento com os clientes (FUCHIGAMI; MOCCELLIN, 2015).

A programação reativa da produção, se bem executada, pode assegurar uma boa qualidade em um curto período de tempo, melhorando a competitividade, caso encontre uma programação que gere um melhor desempenho do sistema produtivo (SUN; XUE, 2001; TANG; WANG, 2008).

Com base nessas características, a ideia deste trabalho é propor um Algoritmo Genético aplicado ao Problema de Programação Reativa da Produção.

Objetivos

O objetivo geral deste trabalho é desenvolver um algoritmo genético que encontre uma programação da produção minimizando o tempo total de processamento de todos os produtos considerando a reprogramação da produção para os casos em que algum recurso material (máquina ou veículo de transporte) pode falhar.

2 Algoritmos Genéticos

2.1 Funcionamento dos algoritmos genéticos

O Algoritmo Genético (AG) faz parte de uma divisão particular da Computação Evolutiva que tem como base as estratégias evolutivas formuladas por Darwin em sua Teoria da Evolução Natural.

Os AGs são frequentemente usados em problemas de busca, tendo encontrado ampla aplicação em diversas áreas. Estes algoritmos utilizam os princípios de seleção natural análogo ao modelo biológico, e têm provado ser uma abordagem de sucesso para problemas de otimização (GOLDBERG, 1989). Eles utilizam uma estratégia de busca probabilística visando encontrar a melhor

solução dentro de um espaço de soluções.

Dentro do contexto dos AGs, uma possível solução para um determinado problema é chamada de indivíduo, ou simplesmente cromossomo.

O funcionamento de um AG dá-se a partir do processo de inicialização, em que é criada uma população de indivíduos representando soluções iniciais que são submetidas ao processo de evolução que acontece pelas seguintes etapas:

- **Avaliação:** a partir de uma função de aptidão (*fitness*) é atribuído um valor ao indivíduo que representa quão bem este indivíduo resolve o problema em questão. Um indivíduo é a unidade fundamental dos AG, pois ele representa uma possível solução para o problema;
- **Seleção:** nesta etapa são escolhidos os indivíduos que irão se reproduzir para gerar uma nova população. Quanto maior a aptidão do indivíduo, maior é a sua probabilidade de ser escolhido;
- **Cruzamento:** depois que os indivíduos são selecionados, eles passam com uma probabilidade pré-estabelecida pelo processo de cruzamento. Essa probabilidade é chamada de taxa de cruzamento. Neste processo são selecionados dois indivíduos para o cruzamento, gerando dois novos indivíduos que irão compor a próxima geração;
- **Mutação:** a mutação é aplicada sobre os indivíduos resultantes do cruzamento, com uma taxa de probabilidade pré-definida chamada taxa de mutação. A mutação altera algumas características dos indivíduos resultantes do cruzamento. Isso é feito para manter a variedade da população e evitar os máximos e os mínimos locais.
- **Atualização:** neste ponto todos os indivíduos criados são inseridos na nova população;
- **Encerramento:** caso a condição de parada do AG seja satisfeita, a execução do AG é encerrada. Caso contrário, o fluxo é retornado para a etapa de avaliação. A condição de parada pode ser o número de gerações (de ciclos) ou a convergência do algoritmo.

Apesar da aleatoriedade ser uma característica intrínseca aos AGs, eles fazem uma busca direcionada. Os novos indivíduos gerados possuem

características da população anterior, e essas características são usadas como histórico que direcionam a busca para um resultado melhor que o resultado gerado pela população anterior.

O Algoritmo 1 representa o fluxo básico de um AG, em que $P(t)$ significa “população da geração t ”.

Algoritmo 1 – Estrutura de um AG

Início

$t = 0$

inicializa $P(t)$

avalia $P(t)$

Enquanto (condição_de_parada = falso) $t = t +$

|

seleciona $P(t)$ de $P(t - 1)$

aplica_cruzamento $P(t)$

aplica_mutação $P(t)$

avalia $P(t)$

Fim-enquanto

Fim

2.2 Representação

O cromossomo pode ser representado de diversas formas, sendo que as mais comuns são: codificação binária, inteira ou real. A escolha da representação do cromossomo é uma das partes mais importantes durante o projeto de um algoritmo genético, visto que sua representação é um dos principais fatores responsáveis pelo seu desempenho. Exemplos de cromossomos são mostrados na Figura 1.

1	0	0	1	1	0	0	1
(a)							
1	2	2	3	1	3	5	1
(b)							
1.8	0.5	3.2	1.3	1.2	4.5	0.5	1.0
(c)							

Figura 1. Cromossomo binário (a), cromossomo inteiro (b) e cromossomo real (c).

Fonte: Elaborado pelo autor

A representação de cada parâmetro, de acordo com a representação utilizada, é chamada de gene. No exemplo da Figura 1 cada posição do vetor é um gene.

3 Proposta

A Programação da Produção em Sistema de Manufatura com Recursos Compartilhados e com uso Simultâneo de Máquinas e Sistema de Transporte envolve a tomada de decisão que implica na alocação dos recursos de produção e do sistema de transporte, assim como a escolha dos roteiros de manufatura a serem seguidos no processo produtivo. A quantidade total de tempo gasto na produção, conhecido como *makespan* é um valor comumente utilizado para definir o desempenho da produção em relação a essas decisões (BAI; TANG, 2013).

Dado um sistema de manufatura qualquer, supõe-se que sua meta seja fabricar certo número de produtos em um determinado tempo de produção. Para que se alcance essa meta, necessita-se determinar a alocação dos recursos, a ordem de fabricação dos produtos e a ordem das operações sobre cada produto. As dificuldades no cumprimento dessa meta estão em determinar, previamente, quais operações devem ser executadas e quando devem ser processadas, de forma que todos os produtos sejam finalizados no menor tempo possível (PAN; TASGETIREN; LIANG, 2008).

Neste trabalho, um cromossomo representará quais produtos devem ser produzidos, seguido de seus respectivos roteiros de fabricação.

Morandin et. al. (2008) e Deriz (2007) apresentam um exemplo de codificação do cromossomo, em que consideram uma fábrica com seis máquinas M1, M2, M3, M4, M5, M6 e três tipos de produto P1, P2, P3. A Tabela 1 exemplifica a relação dos roteiros de fabricação possíveis para os produtos.

Tabela 1. Produtos e roteiros de fabricação

Produto	Roteiros de Fabricação
P1	R1 (M1, M2, M6) R2 (M4, M5, M6)
P2	R1 (M1, M2, M5, M6) R2 (M3, M4, M5, M6)
P3	R1 (M4, M3, M2) R2 (M4, M1, M5)

Fonte: adaptado de Morandin et al. (2008)

A Figura 2 exemplifica a representação cromossômica de uma possível programação da produção tomando como base os possíveis roteiros de cada produto mostrados na Tabela 1.

Nesta representação, o produto é seguido das máquinas por onde ele deve passar seguindo a ordem da esquerda para a direita. O número 0 é usado para preencher espaços desnecessários no cromossomo. Por exemplo, o produto P2 precisa passar por 4 máquinas para ser produzido. Já os produtos P1 e P3 precisam passar por 3 máquinas. Como esta representação é estática, reserva-se no cromossomo, para cada produto, a quantidade de espaços que o produto com maior roteiro necessita. Uma vez que os produtos P1 e P3 não precisam de uma 4ª máquina, a 4ª posição reservadas respectivamente para eles é representada com o zero.

P1	M1	M2	M6	0	P3	M4	M1	M5	0	P2	M1	M2	M5	M6
----	----	----	----	---	----	----	----	----	---	----	----	----	----	----

Figura 2. Representação cromossômica de programação de solução

Fonte: Elaborado pelo autor

Morandin et al. (2008) simplificam a representação anterior usando não a sequência de máquinas, mas sim um dos possíveis roteiros para cada produto. A Figura 3 ilustra esta simplificação para o mesmo exemplo da Tabela 1.

A desvantagem desta representação é a possibilidade do algoritmo sugerir um roteiro em que um ou mais recursos estão indisponíveis.

P1	R1	P3	R2	P2	R1
----	----	----	----	----	----

Figura 3. Representação cromossômica simplificada

Fonte: Elaborado pelo autor

O trabalho aqui proposto visa utilizar a representação da Figura 2 como uma forma de aumentar a capacidade de encontrar de maneira mais eficiente a reprogramação da produção nos casos em que recursos da linha de produção apresentam falhas.

3 Experimentos e resultados

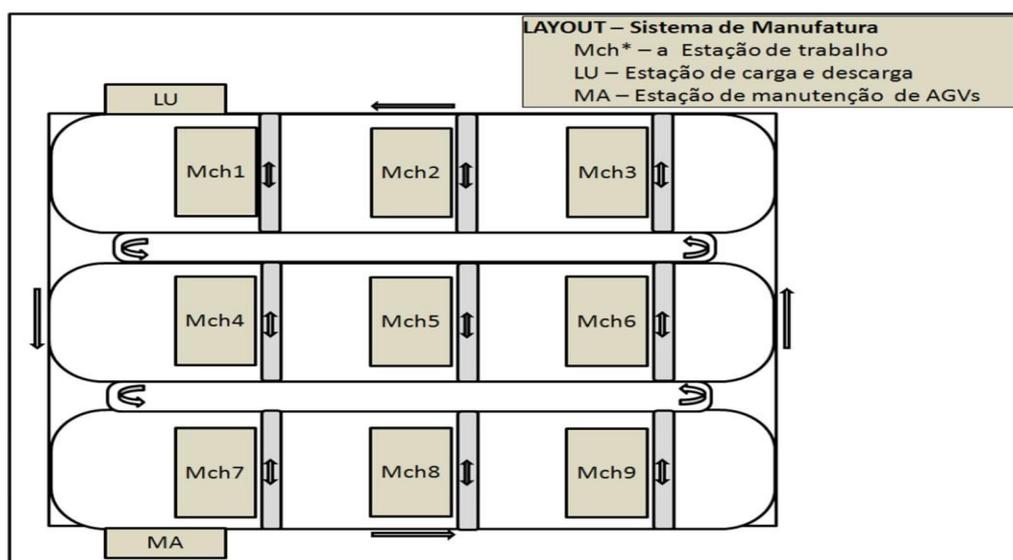
Os critérios adotados para definir o desempenho da proposta são

baseados na obtenção do mínimo *makespan* e no tempo computacional que o algoritmo leva para convergir em uma boa solução. A programação da produção é realizada no nível de controle e não no nível de planejamento. Portanto, não é necessário fazer o tratamento das ocorrências dos eventos inesperados, pois estas são tratadas no momento em que ocorrem para isto solicita-se a reprogramação da produção.

O arranjo físico do sistema de manufatura considerado neste trabalho é apresentado na Figura 4, que é o mesmo cenário utilizado por Morandin (2008). Desta forma, replicaremos o mesmo cenário utilizado por Morandin (2008) a fim de utilizá-lo como parâmetro de comparação para a validação do algoritmo aqui proposto.

O sistema é composto por nove estações de trabalho, uma estação de carga e descarga, uma estação de manutenção de AGVs e três AGVs. Os produtos inicialmente são introduzidos em seu processo de produção por meio da estação de carga (C) e retirados do processo através da estação de descarga (D). São manufacturados nove produtos distintos (P1, P2, P3, P4, P5, P6, P7, P8 e P9).

Essas escolhas foram feitas para que o trabalho proposto possa ser comparado ao de Morandin (2008) sob as mesmas condições de cenário. As informações referentes a cada um desses produtos, em relação aos roteiros de produção, aos tempos de produção e aos tempos de transporte, são apresentadas a seguir.



.Figura 4 - Layout do Sistema Produtivo

Fonte: Morandin (2008)

Os roteiros de produção dos nove produtos pertencentes ao sistema produtivo são demonstrados na Figura 5.

Os roteiros de fabricação dos produtos e os tempos de produção foram gerados aleatoriamente. Esses últimos, que variam de 400 a 500 unidades de tempo (u.t.), encontram-se ilustrados na Figura 6. Já os tempos de transporte, encontram-se na Figura 7. O tempo de transporte corresponde ao tempo que um veículo leva para movimentar um produto de um ponto a outro do sistema produtivo, ou seja, o tempo em que o veículo leva para alcançar uma máquina ou estação de carga/descarga à partir da sua posição atual.

Produtos	Roteiros		Máquinas
		R1	R2
Rot. P1	R1	R2	1 2 4 5 7 9 0
			3 4 5 6 8 9 0
Rot. P2	R1	R2	1 2 3 4 5 6 7
			2 3 5 7 8 9 0
Rot. P3	R1	R2	4 5 6 7 8 0 0
			2 3 7 8 9 0 0
Rot. P4	R1	R2	2 3 4 6 7 0 0
			1 5 6 8 9 0 0
Rot. P5	R1	R2	4 5 7 8 9 0 0
			1 2 3 5 6 0 0
Rot. P6	R1	R2	2 4 5 6 7 8 9
			1 3 6 7 8 9 0
Rot. P7	R1	R2	1 2 4 5 6 9 0
			1 2 3 7 8 9 0
Rot. P8	R1	R2	4 5 6 7 8 9 0
			3 4 5 7 8 9 0
Rot. P9	R1	R2	3 5 6 7 8 9 0
			2 4 6 7 8 9 0

Figura 5 - Roteiros de Produção
Fonte: Morandin (2008)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	
M1	428	439	453	403	481	446	414	491	458	M1
M2	423	433	474	436	440	495	457	419	486	M2
M3	459	487	417	410	477	474	452	435	416	M3
M4	433	405	447	410	442	448	426	491	454	M4
M5	467	447	486	400	450	469	493	495	452	M5
M6	461	497	496	468	468	408	408	452	438	M6
M7	464	495	459	489	436	454	457	477	484	M7
M8	455	469	489	439	486	424	497	452	435	M8
M9	418	439	480	457	435	482	445	408	416	M9
	P1	P2	P3	P4	P5	P6	P7	P8	P9	

Figura 6 - Tempo de Produção
Fonte: Morandin (2008)

	C	M1	M2	M3	M4	M5	M6	M7	M8	M9	D	
C	0	14	5	10	12	14	10	13	11	9	13	C
M1	11	0	11	6	5	6	5	9	11	6	11	M1
M2	6	9	0	7	5	6	10	6	9	11	13	M2
M3	10	7	5	0	11	5	10	12	14	8	6	M3
M4	7	13	8	9	0	10	11	6	14	7	9	M4
M5	9	10	8	11	10	0	7	13	9	12	9	M5
M6	14	11	13	7	14	6	0	12	13	14	8	M6
M7	10	7	9	13	5	8	10	0	8	10	13	M7
M8	11	6	9	6	8	10	12	13	0	9	7	M8
M9	6	5	6	8	9	10	9	5	4	0	8	M9
D	6	5	11	8	10	11	8	5	14	13	0	D
	C	M1	M2	M3	M4	M5	M6	M7	M8	M9	D	

Figura 7 - Tempo de Transporte

Fonte: Morandin (2008)

A população de um algoritmo genético é um conjunto de indivíduos que são soluções em potencial para o problema específico. O tamanho da população influencia diretamente o desempenho do AG. Se a população for pequena, pode-se não manter a diversidade entre os indivíduos, o que influencia a convergência a uma boa solução por cobrir uma pequena parte do espaço de busca do problema. Por outro lado, se a população for muito grande, a eficiência do AG ficará comprometida, a vista que terá um número maior de indivíduos para se calcular a função de aptidão (GOLDBERG, 1989; MICHALEWICZ, 1996). A Tabela 2 mostra as configurações do AG utilizado nos experimentos.

Tabela 2. Produtos e roteiros de fabricação

Variável	Valor
Número de gerações	60
Taxa de cruzamento	80%
Taxa de mutação	5%
Tamanho da população	300

Fonte: adaptado de Morandin et. al. (2008)

Considerando este cenário, o AG proposto aqui foi executado 50 vezes, da mesma forma que em Morandin (2008). Estas repetições visam obter amostragens do desempenho dos algoritmos desenvolvidos. Estes são avaliados quanto ao tempo necessário para obtenção da solução e valor de *makepan* fornecido em cada teste. Isto é necessário devido às características não determinísticas dos AGs. Assim, os resultados variam para cada teste. Os resultados podem ser vistos na Tabela 3.

Tabela 3. Resultado das 50 execuções

Execução	MORANDIN (2008)	Proposta
1	5561	5322
2	5900	5461
3	6205	6044
4	5900	5154
5	5893	5868
6	5679	5158
7	5948	5431
8	5460	5371
9	5459	5433
10	5926	5869
11	5852	5354
12	5951	5601
13	6341	6124
14	5954	5846
15	5900	5269
16	6386	6132
17	6010	5330
18	5900	5805
19	5481	5517
20	6294	5553
21	5525	5604
22	5172	4544
23	5998	5258
24	5515	5120
25	5490	5011
26	6266	6199
27	5413	4859
28	6302	5829
29	6307	5994
30	5462	5119
31	5893	5428
32	5679	5208
33	5948	5926
34	5460	4903
35	5459	5337
36	5926	5223
37	5900	5217
38	5951	5158
39	6341	5550
40	5954	5526
41	5900	5587
42	6266	6049
43	5413	4926
44	6302	6385
45	6307	6173
46	5462	4804
47	5893	5719
48	5679	5469
49	5900	5904
50	6266	5889
Média	5866,98	5511,2

Fonte: Elaborado pelo autor

A Tabela 3 mostra que em média o algoritmo proposto obtém uma programação melhor (em termos de *makespan*) que o método aplicado por Morandin (2008).

Considerações finais

A programação de produtos com o uso simultâneo de máquinas e sistemas de transporte é caracterizada pela grande quantidade de soluções possíveis. Em especial, para problemas de grande escala que possuem um número considerável de máquinas e produtos para a produção. Por isso, as características dos algoritmos genéticos se tornam apropriadas para tal tarefa, uma vez que esses possuem a capacidade de percorrer de forma global os espaços da busca com as soluções possíveis para o problema proposto.

Este trabalho propôs um Algoritmo Genético que pudesse tratar o problema da produção da programação. Os resultados obtidos demonstram a capacidade do algoritmo em encontrar uma programação com baixo *makespan*.

Referências

BAI, D.; TANG, L. Open shop scheduling problem to minimize makespan with release dates. **Applied Mathematical Modelling**, v. 37, n. 4, p. 2008-2015, 2013.

CHAN, F. T. S.; CHUNG, S. H.; CHAN, P. L. Y; FINKE, G; TIWARI, M. K., Solving distributed FMS scheduling problems subject to maintenance: Genetic algorithms approach. **Robotics and Computer-Integrated Manufacturing**, v. 22, p. 493-504, 2006.

DERIZ, A. C. **Um método de busca usando algoritmo genético para programação reativa da produção de sistemas de manufatura com recursos compartilhados**. 2007. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de São Carlos, São Carlos, 2007.

FUCHIGAMI, H. Y.; MOCCELLIN, J. V. Desempenho relativo de regras de prioridade para programação de flow shop híbrido com tempos de setup. **Revista Produção Online**, v. 15, n. 4, p. 1174-1194, 2015.

FUCHIGAMI, H. Y.; MOCCELLIN, J. V. Efeitos de regras de prioridade para programação da produção em sistemas industriais complexos. **Revista Produção Online**, v. 16, n. 1, p. 3-25, 2016.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley Longman Publishing Co., Inc. Boston, MA,

USA, 1989.

LI, Z.; IERAPETRITOU, M. Process scheduling under uncertainty: Review and challenges. **Computers and Chemical Engineering**, Oxford, v. 32, n. 4-5, p. 715- 727, April 2008.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. 3. ed. Springer-Verlag, 1994.

MONTEVECHI, J. A. B.; MORANDIN JR., O.; MIYAGI, P. E. Sistemas de manufatura In: AGUIRRE, L. A. **Enciclopédia de Automática, Controle & Automação**. São Paulo: Editora Blucher, 2007. p. 247-287.

MORANDIN O. et al. A Fuzzy Classifier Model for the Production Sequencing. International Conference on Computational Intelligence for Modelling Control & Automation, p. 597-602, 2008.

PAN, Q.-K.; TASGETIREN, M. F.; LIANG, Y.-C. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. **Computers & Operations Research**, v. 35, n. 9, p. 2807-2839, 2008.

QIU, R.; ZHOU, M. Mighty MESSs: state-of-the-art and future manufacturing execution systems. **Robotics & Automation Magazine**, IEEE, v. 11, n. 1, p. 19-25, 2004.

REDDY, B. S. P.; RAO, C. S. P. A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. **International Journal Advanced Manufacturing Technology**, v. 31, p. 601-613, 2006.

SLACK, N.; CHAMBERS S.; JOHNSTON, R. **Administração da Produção**. São Paulo: Atlas, 2002.

SUN, J.; XUE, D. A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. **Computers in Industry**, v. 46, n. 2, p. 189- 207, sep. 2001.

TANG, L.; WANG X. A predictive reactive scheduling method for color-coating production in steel industry. **International Journal Advanced Manufacturing Technology**, Guildford, v. 35, n. 7-8, p. 633-645, jan. 2008.

ZHOU, H.; FENG, Y.; HAN, L. The hybrid heuristic genetic algorithm for job shop scheduling. **Computers & Industrial Engineering**, v. 40, n. 3, p. 191-200, 2001.