

ANÁLISE DO COMPORTAMENTO DAS ESTRATÉGIAS DE ESCALONAMENTO CONSIDERANDO “N” MÁQUINAS DE PRODUÇÃO

ANALYSIS OF STRATEGIES SCHEDULING BEHAVIOR CONSIDERING “N” PRODUCTION MACHINES

Gilberto Irajá Muller¹ e Arthur Tórgo Gómez²

RESUMO

Neste trabalho, apresenta-se uma análise do comportamento das estratégias de escalonamento, através do M-EP (Modelo de Escalonamento de Produção), aplicado ao *Job-shop Scheduling Problem* que considera o tempo total de produção (*makespan*), o tempo total de atraso e o tempo total de paradas. O modelo, que utiliza o algoritmo “busca tabu” para a otimização do escalonamento, compõe-se por: (i) uma função objetivo que reflete as estratégias de otimização e (ii) uma arquitetura que está dividida em cinco fases. A arquitetura do modelo utiliza conceitos de tecnologia de grupo e regras de despacho para tratar os problemas de seleção de partes e do escalonamento. Os resultados e análises do comportamento das estratégias apresentam-se na conclusão do trabalho.

Palavras-chave: job-shop, escalonamento, tecnologia de grupo, regras de despacho, “busca tabu”.

ABSTRACT

This paper presents an analysis of scheduling strategies behavior through the M-EP (Production Scheduling Model) applied to Job-shop Scheduling Problem that considers production total time (makespan), total delaying time and total stopping time. The model, that uses the algorithm Tabu Search for the scheduling optimization, is composed by: (i) an objective function that reflects the strategies of optimization, and (ii) an architecture that is divided in five phases.

^{1,2}Universidade do Vale do Rio dos Sinos – UNISINOS,
PIPICA - Programa de Pós-Graduação em Computação Aplicada,
Av. Unisinos, 950 – Cristo Rei – CEP 93022-000 – São Leopoldo - RS.
irajamuller@terra.com.br, breno@unisinos.br

The architecture of the model uses group technology concepts and dispatching rules to deal with problems of part selection and scheduling. The results and analyses of the strategies behavior are presented in the conclusion of the paper.

Keywords: *job-shop, scheduling, group technology, dispatching rules, “tabu search”.*

INTRODUÇÃO

O *Job-shop Scheduling Problem* (JSP), com critério de *makespan*, tem sido amplamente pesquisado por sua complexidade e por sua aplicação em indústrias. Para ilustrar sua complexidade, uma instância com 10 partes e 10 máquinas, proposta por Muth e Thompson (1963), permaneceu por mais de 40 anos sem solução. O JSP possui a seguinte definição: n jobs, sendo que um job tem j operações a serem processadas em m máquinas e cada operação j possui um tempo de processamento e uma ordem no roteiro do job. A máquina processa uma operação por vez sem que ocorra interrupção. Objetiva-se, portanto, encontrar um escalonamento que minimize o *makespan* C_{max} (tempo total de produção), dado o processamento do último job.

Seja $J=\{0, 1, \dots, n, n+1\}$ o conjunto de operações e $M=\{1, \dots, m\}$ o conjunto de máquinas. As operações 0 e $n+1$ não possuem duração e representam o início e o fim da operação. Para as operações inter-relacionadas existem dois tipos de restrições: (i) a de precedência, em que cada operação j será escalonada depois que todas as operações predecessoras P_j finalizarem-se e (ii) a operação somente se nivelará se a máquina estiver ociosa. A seguir, será representado o tempo de processamento por d_j e o tempo final da operação j por F_j . Para o escalonamento, será utilizado um vetor dos tempos de finalização $(F_1, F_2, \dots, F_{n+1})$ e $A(t)$ para o conjunto de operações, as quais se processarão no tempo t ; $r_{j,m}=1$, se a operação j processar-se na máquina m e $r_{j,m}=0$, em caso contrário. O modelo do JSP pode ser definido por (GONÇALVES et al., 2005):

$$\text{Min } F_{n+1}(C_{max}) \quad (1)$$

$$F_k \leq F_j - d_j, \quad j=1, \dots, n+1; \quad k \in P_j, \quad (2)$$

$$\sum_{j \in A(t)} r_{j,m} \leq 1, \quad m \in M; \quad t \geq 0, \quad (3)$$

$$F_j \geq 0, \quad j=1, \dots, n+1. \quad (4)$$

Na função objetivo (1), busca-se minimizar o tempo total de produção (*makespan*). A restrição (2) assegura a precedência das operações e a (3) compromete-se com o processamento de uma operação por vez. Por fim, a restrição (4) garante a não negatividade. Dado o problema supracitado, objetiva-se analisar o comportamento das estratégias de escalonamento por meio de um modelo aplicado ao JSP. Na seção 2, apresentam-se os problemas abordados no JSP e o método utilizado; na seção 3, conceitua-se o M-EP e, na seção 4, definem-se os resultados da validação, o comportamento das estratégias de escalonamento e as conclusões finais.

PROBLEMAS ABORDADOS

Apresentam-se, nesse momento, dois problemas que são abordados na geração do M-EP, aplicado ao JSP. O primeiro relaciona-se à fase pré-operacional e o segundo à fase operacional, baseado no modelo conceitual para a manufatura flexível proposto por Stecke (1986):

- seleção de partes: existem diversos métodos para a geração das famílias de partes (FP), sendo destaques a inspeção visual, classificação por codificação e análise por fluxo de produção (formulação matricial, programação matemática e particionamento de grafos) (JHA, 1991). As FPs são formadas por meio de agrupamento das partes, baseadas nas similaridades, tais como: forma geométrica, processo, afinidade por um mesmo conjunto de ferramentas, entre outros, e é considerado um problema NP-Completo (JHA, 1991). O modelo proposto utilizou o método de análise por fluxo de produção, proposto por Kusiak e Chow (1987) denominado de *Cluster Identification*;
- escalonamento de partes: o escalonamento das partes num JSP é um problema de difícil solução, ou seja, NP-Hard (GAREY et al., 1976). Sendo assim, objetiva-se sequenciar os *jobs* em um conjunto de máquinas, através de rotas predefinidas em função do tempo (BLAZEWICZ et al., 1996). Inúmeros métodos de otimização foram propostos para a solução do JSP (ZOGHBY et al., 2004), sendo destacados os métodos de otimização e os métodos aproximativos. Para aqueles, cita-se a programação inteira, a relaxação lagrangeana, as técnicas de Surrogate e o *branch and bound* (BALAS et al., 1979), para estes, destacam-se os algoritmos iterativos “busca tabu”, as redes neurais, os algoritmos genéticos, a têmpera simulada e o GRASP (BLAZEWICZ et al., 1996), (GLOVER; LAGUNA, 1997)

e (GONÇALVES et al., 2005). Para resolver o problema do escalonamento de partes foi utilizado o algoritmo “busca tabu”, em virtude da complexidade do problema e dos diversos trabalhos relacionados.

MODELO DE ESCALONAMENTO DE PRODUÇÃO

A meta-heurística “busca tabu” (BT) desenvolveu-se para abordar problemas de programação inteira; posteriormente, foi dada uma descrição do método para uso geral em problemas de otimização combinatória (ADAMS et al., 1988). Ele possui três princípios fundamentais: (i) uso de uma estrutura de dados (lista) para guardar o histórico da evolução do processo de busca, (ii) uso de um mecanismo de controle para fazer um balanceamento entre a aceitação ou não de uma nova configuração, com base nas informações registradas na lista tabu referentes às restrições e aspirações desejadas e (iii) incorporação de procedimentos que alternam as estratégias de diversificação e intensificação (GLOVER; LAGUNA, 1997). Para a utilização da BT precisa-se definir o objetivo f do problema em questão e, após essa constatação, gera-se uma solução inicial viável, independente (Regra de Despacho). Entretanto, é importante salientar que, para a geração da solução inicial, ela necessita fazer parte do conjunto de soluções possíveis do espaço amostral.

O modelo apresenta a função objetivo para a otimização do escalonamento das partes e de uma arquitetura baseada no modelo de planejamento de produção para o sistema de manufatura flexível (SMF), proposto por Stecke (1986).

FORMULAÇÃO

A formulação utilizada, para a otimização do escalonamento das partes, mostra-se a seguir: m = número de máquinas; n = número de partes; i = índice para a parte; k = índice para a máquina; De_i = data de entrega da parte e Dsp_{ik} = data da saída de produção. De modo que:

$$\text{Min } f(e, p) = p_1 \cdot \text{makespan}(e, p) + p_2 \cdot \text{atraso}(e, p) + p_3 \cdot \text{parada}(e, p) \quad (5)$$

$$\text{makespan}(e, p) = \sum_{k=1}^m \sum_{i=1}^n \text{makespan}_{ik}, \text{ tal que } \text{makespan}_{ik} > 0 \quad (6)$$

$$\text{atraso}(e, p) = \sum_{k=1}^m \sum_{i=1}^n (Dsp_{ik} - De_i), \text{ tal que } (Dsp_{ik} - De_i) > 0 \quad (7)$$

$$\text{parada}(e, p) = \sum_{k=1}^m \sum_{i=1}^n \text{parada}_{ik}, \text{ tal que } \text{parada}_{ik} > 0 \quad (8)$$

$$p_1, p_2, p_3 \geq 0 \quad (9)$$

A função objetivo (5), que busca a minimização, é definida pela variável “e”, (escalonamento), que representa a dimensão temporal e pela variável “p”, (Famílias de Partes), que reproduz a dimensão física para o modelo proposto. Ela forma-se por três variáveis de decisão que refletem estratégias de otimização. A variável de decisão *makespan* (e,p) representa o tempo total de produção (6), ou seja, ela consiste no tempo inicial do primeiro *job* processado em produção até o tempo final do último *job* processado.

A variável de decisão atraso (e, p) avalia o tempo total de atraso (7). O espaço total de paradas de produção (8) obtém-se pela variável de decisão parada (e,p) e significa o tempo entre dois lotes, tempo esse definido como um conjunto de partes encadeadas em um mesmo turno e processadas por uma mesma máquina de produção (GÓMEZ, 1996). Por fim, a restrição (9) assegura a não negatividade das variáveis, conforme propõe a seção 2, sobre o problema do escalonamento. Nela, apresenta-se o algoritmo BT, o qual utiliza a função objetivo supracitada

```

while (niter - melhiter < nbmax) do
    f' ← fmelhor;
    niter ← melhiter + 1;
    p ← p*;
    Gerar  $V_e^*$  de soluções (e,p*)i em  $N_e(e, p^*) < A(f(e, p^*))$ ;
    Atualizar Lista Tabu L e A(z);
    if f(e', p*) < fmelhor then
        fmelhor ← f(e', p*);
    endif
    Gerar  $V_p^*$  de soluções (e, p)i em  $N_p(e, p) \text{ ou } f((e, p)_i) < A(f(e, p))$ ;
    Atualizar Lista Tabu L e A(z);
    if f(e', p') < fmelhor then
        fmelhor ← f(e', p');
    endif
    p ← p';
    e ← e';
    if fmelhor < f' then
        mlhiter ← niter
    endif
endw

```

Algoritmo 1 – Algoritmo “busca tabu”.

A cada iteração escolhe-se um ótimo local e, a partir dele, é gerada uma nova vizinhança. Para evitar ciclos e mínimos locais, é implementada uma lista de

movimentos reversos proibidos, chamada de lista tabu (L). Caso a iteração obtenha uma melhora na função f , utiliza-se a função critério de aspiração A , que admite o movimento até então proibido. Para as condições de parada do algoritmo, é utilizado o número máximo de iterações ($nbmax$) sem que ocorra melhora na função f .

ARQUITETURA

A arquitetura para o M-EP está dividida em cinco fases de aplicação, conforme apresenta a figura 1.



Figura 1 – Arquitetura do M-EP.

A primeira fase responsabiliza-se por obter a demanda de produção das informações contidas no banco de dados, sendo recuperadas as informações técnicas da parte e a data de entrega. Na segunda fase, ocorre a geração da matriz parte *versus* máquina, através da aplicação do algoritmo de identificação de agrupamentos. Na terceira fase, emprega-se a regra de despacho, a partir da utilização do *cluster identification* para a geração do escalonamento inicial. Após a geração, realiza-se a quarta fase, que consiste na aplicação do algoritmo BT para a otimização do escalonamento e gravação do escalonamento final em um plano de produção, possibilitando, assim, observar seus históricos e compará-los com o escalonamento efetivo.

RESULTADOS

A etapa de validação do M-EP realiza-se por meio da comparação com trabalhos clássicos que propõem a solução do JSP (BLAZEWICZ et al., 1996; JAIN; MEERAN, 1998; ZOGHBY et al., 2004). Para testar o desempenho da arquitetura utilizam-se diversas instâncias do JSP, a fim de realizar a validação em diversos ambientes (número de partes x número de máquinas). Para a validação aproveitam-se as instâncias propostas por Muth e Thompson (1963), denominada de FT06, FT10 e FT20, e as propostas por Lawrence (1984), intituladas de LA01, LA06, LA11, LA16, L21, LA29 e LA40.

TRABALHOS SELECIONADOS

Por meio dos trabalhos que propõem a resolução do JSP (JAIN; MEERAN, 1998; ZOGHBY et al., 2004), foram selecionados alguns autores em função do tipo de algoritmo utilizado e da sua importância na evolução do estado da arte, conforme demonstra a tabela 1.

CONFIGURAÇÃO

Implementou-se o modelo em *delphi 7* e seus experimentos executados em um *hardware Athlon XP 2500+ 512 RAM* com sistema operacional *windows XP Pro*. Para a configuração do algoritmo BT, atribui-se o valor de 100 para o número máximo de iterações sem obter melhora na função objetivo (*nbmax*) e, para o tamanho da lista tabu, conferiu-se 15. Em virtude dos autores selecionados utilizarem o *makespan* para os seus experimentos, os pesos (p_1 , p_2 e p_3) para a função objetivo foram fixados em 100, 1 e 1, respectivamente, fator que privilegiou a variável de decisão *makespan*.

É importante ressaltar que o M-EP, além de otimizar o *makespan*, possui uma arquitetura que objetiva o gerenciamento entre as datas de entrega (minimização do atraso) e a produtividade (minimização das paradas de produção). As demais abordagens supracitadas otimizam somente o *makespan*, desconsiderando as datas de entrega e a produtividade do sistema. Como as instâncias não possuem as datas de entrega para a parte e o modelo proposto as utiliza como estratégia de otimização, fixou-se todas as partes com a mesma data de entrega.

Tabela 1 – Autores selecionados para a validação do modelo.

Autor	Algoritmo	Ano
Dorndorf e Pesch, (1995)	Algoritmos genéticos	1995
Nowicki e Smutnicki, (1996)	“busca tabu”	1996
Wang e Zheng, (2001)	Genético híbrido e Têmpera simulada	2001
Aiex et al.(2003)	GRASP	2003
Gonçalves et al. (2005)	Algoritmos genéticos híbridos	2005

VALIDAÇÃO

A tabela 2 mostra os resultados da validação e, a partir da amostragem, apresenta-se a instância do problema, a dimensão do problema (número de partes x número de máquinas), a solução ótima conhecida (SOC), o recurso obtido pelo M-EP e o resultado dos autores selecionados.

Ela possibilita a comparação entre o modelo proposto e a solução ótima conhecida. Nota-se que o *makespan*, com a maior diferença em relação à SOC, diferencia-se em 7 unidades de tempo (LA40) e, desse modo, equivale a uma diferença de 0,57%. Sendo assim, a validação do modelo e dos resultados esperados atingiram o objetivo, ou seja, em muitos casos alcançou a SOC e, em outros, aproximou-se. As instâncias que obtiveram diferenças em relação à SOC sofreram influência da função objetivo do modelo proposto. Ao avaliar a função objetivo, considerou-se para esse problema do *makespan* um peso significativamente maior para a variável de decisão *makespan* do que às demais variáveis, direcionando, assim, a BT para a otimização do *makespan*.

A capacidade do modelo de gerenciar estratégias diferentes de escalonamento ocorre em virtude das variáveis de decisão que compõem a função objetivo e que abordam estratégias de escalonamento, tais como: (i) *makespan*, (ii) tempo total em atraso e (iii) tempo total parado em produção. A seguir, apresentam-se algumas estratégias de escalonamento, bem como seu comportamento.

Tabela 2 – Estratégia que privilegia a produtividade.

Instância	NxM	SOC	M-EP	Gonçalves	Aiex	Wang	Nowicki	Dorndorf
FT06	6x6	55	55	55	55	55	55	
FT10	10x10	930	930	930	930	930	930	960
FT20	20x5	1165	1165	1165	1165	1165	1165	1249
LA01	10x5	666	666	666	666	666	666	666
LA06	15x5	926	926	926	926	926	926	926
LA11	20x5	1222	1222	1222	1222	1222	1222	1222
LA16	10x10	945	945	945	945	945	945	1008
LA21	15x10	1046	1047	1046	1057	1058	1047	1139
LA29	20x10	1157	1160	1196	1203		1160	1336
LA40	15x15	1222	1229	1241	1244		1229	1321

ESTRATÉGIA QUE OTIMIZA AS PARADAS DE PRODUÇÃO

Dado o privilégio da variável de decisão parada (peso $p_3 = 100$), objetiva-se determinar a produtividade e, conseqüentemente, a velocidade de produção. Os experimentos, os quais se apresentarão na tabela 3, objetivam otimizar a produtividade, porém presencia-se um conflito em relação às datas de entrega (atraso) e às parcelas de produção, ou seja, quanto menor o tempo de paradas de produção (maior magnitude), menor a magnitude da variável de decisão atraso da BT.

Desse modo, constata-se a independência das variáveis, porque ao influenciar uma, a outra sofre efeito contrário. Esse comportamento ocorre, pois a variável de decisão atraso perde a influência na função objetivo, privilegiando, desse modo, a escolha do movimento na vizinhança (algoritmo BT) em função da variável de decisão parada.

A constatação da dependência de duas variáveis de decisão é outro aspecto importante neste experimento, pois ao reduzir o tempo parado, ocorre influência na redução do *makespan*. A estratégia que privilegia a redução do tempo parado consiste nos mesmos valores obtidos pelo *makespan*, calculado na etapa de validação do modelo.

Tabela 3 – Estratégia que privilegia a redução das paradas de produção.

Instância	SOC	M-EP	Diferença %	Partes em atraso
FT06	55	55	0,00	2
FT10	930	930	0,00	3
FT20	1165	1165	0,00	4
LA01	666	666	0,00	2
LA06	926	926	0,00	3
LA11	1122	1122	0,00	3
LA16	945	945	0,00	4
LA21	1046	1047	0,09	5
LA29	1157	1160	0,26	5
LA40	1222	1229	0,57	4

ESTRATÉGIA QUE OTIMIZA AS DATAS DE ENTREGA

Dado o privilégio da variável de decisão atraso (peso $p_2=100$), objetiva-se a entrega pontual das partes, desconsiderando, nesse momento, as demais estratégias de otimização.

Tabela 4 – Estratégia que privilegia a redução das paradas de produção.

Instância	SOC	M-EP	Diferença %	Partes em atraso
FT06	55	60	9,09	0
FT10	930	1003	7,84	0
FT20	1165	1243	6,69	0
LA01	666	722	8,40	0
LA06	926	1037	11,98	0
LA11	1222	1313	7,45	0
LA16	945	1020	7,94	0
LA21	1046	1145	9,46	0
LA29	1157	1259	8,82	0
LA40	1222	1356	10,96	0

A partir da tabela 4, observa-se que o modelo obteve a entrega pontual de todas partes. O *makespan*, porém, sofreu aumento significativo, consequência do aumento das paradas de produção. Dessa forma, constatou-se o conflito existente entre respeitar as datas de entrega e manter a produtividade.

ESTRATÉGIA NÃO TENDENCIOSA

A solução não tendenciosa consiste naquela em que o peso não privilegia nenhuma das variáveis de decisão da função objetivo. Dessa forma, ela se obtém por meio da proporção das variáveis de decisão, a partir da execução de 100 experimentos variando os pesos (p_1, p_2, p_3), seguindo uma distribuição normal com intervalo (0, 100). Com a estratégia não tendenciosa, ocorreu a redução do conflito entre as variáveis de decisão *makespan* e parada, ou seja, a redução entre a produtividade e as datas de entrega. Na tabela 5, ilustram-se os resultados obtidos.

Tabela 5 – Estratégia não tendenciosa.

Instância	SOC	M-EP	Diferença %	Partes em atraso
FT06	55	57	3,63	0
FT10	930	961	3,33	2
FT20	1165	1198	2,83	4
LA01	666	684	2,70	1
LA06	926	951	2,75	2
LA11	1222	1267	3,68	3
LA16	945	976	3,25	3
LA21	1046	1086	3,28	4
LA29	1157	1206	4,23	3
LA40	1222	1272	4,09	2

Essa estratégia pode ser utilizada como política de escalonamento, pois obteve-se uma relação consistente entre as variáveis de decisão *makespan* e atraso, observando-se que o *makespan* aproximou-se da SOC e poucas partes foram entregues com atraso. A utilização de uma estratégia não tendenciosa possibilitou a verificação da redução do *makespan* e não houve diferenças significativas em relação ao número de partes em atraso, ao ser comparado com a estratégia que privilegia a entrega pontual.

ANÁLISE DO COMPORTAMENTO

Na figura 2, apresenta-se o comportamento das variáveis de decisão em relação ao atraso. O eixo “y”, à esquerda, representa o valor do *makespan* e o eixo “y”, à direita, representa a quantidade de partes em atraso. Sob ponto de vista do atraso, a estratégia da variável atraso (p_2) e a estratégia não tendenciosa (NT) obtiveram boas respostas em relação às demais, sendo visível na magnitude dos eixos (maior compactação). Contudo, a estratégia das variáveis *makespan* (p_1) e parada (p_3) obtiveram qualidade de resposta superior em relação ao *makespan*, mostrando, assim, a capacidade do modelo de gerenciar diversos cenários e o conflito entre respeitar as datas de entrega e a produtividade.

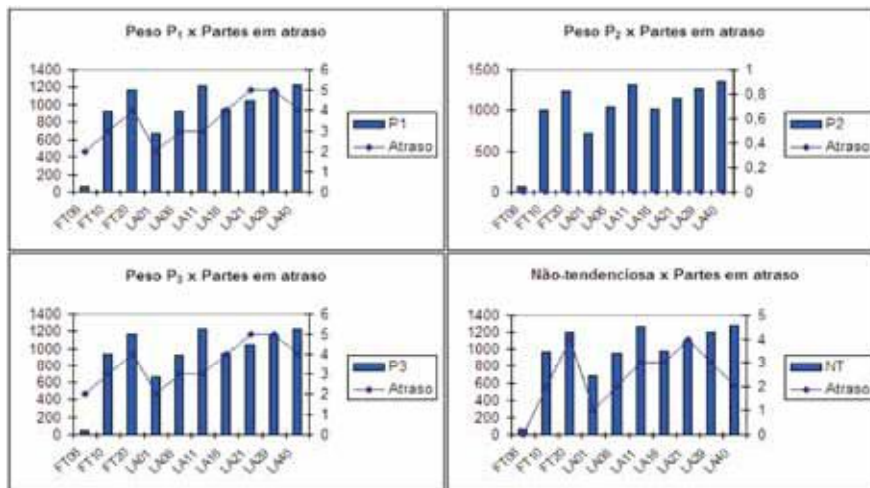


Figura 2 - Análise do comportamento das variáveis.

CONCLUSÃO

Neste trabalho, objetivou-se analisar o comportamento das estratégias de escalonamento, pelo M-EP, aplicado ao JSP. O modelo implementado baseia-se numa função objetivo que reflete, nas variáveis de decisão e em seus pesos respectivos, estratégias de escalonamento. A arquitetura do modelo responsabiliza-se por tratar os problemas de seleção e de escalonamento das partes. Por meio das estratégias, foi possível demonstrar a flexibilidade e a versatilidade do modelo.

Foram utilizadas, na validação, diversas instâncias para o problema do JSP e, desse modo, conclui-se que o modelo permite explorar estratégias de escalonamento e se adequar de acordo com o comportamento que a indústria necessita, ou seja, o objetivo que melhor rentabilize. Também apresentou-se uma análise do comportamento das variáveis e suas relações de dependência, através da variação dos pesos.

Como extensão futura, sugere-se aprimorar os métodos de geração e escolha de vizinhança e propor instâncias para o JSP que considerem as datas de entrega e paradas de produção, dado que as instâncias utilizadas adaptaram-se para a validação do modelo.

REFERÊNCIAS

ADAMS, J.; BALAS, E.; ZAWACK, D. **The shifting bottleneck procedure for job shop scheduling**. In: MANAGEMENT SCIENCE, v. 34, p. 57–73, 1988.

AIEX, R.; BINATO, S.; RESENDE, M. G. C. **Parallel GRASP with pathrelinking for job shop scheduling**. In: PARALLEL COMPUTING, v. 29, n. 4, p. 393–430, 2003.

BALAS, E.; JOHNSON, E. L.; KORTE, B. **Disjunctive programming**. In: HAMMER, P.L. **Discrete Optimisation II**, p. 49–55, 1979.

BLAZEWICZ, J.; DOMSCHKE, W.; PESCH, E. **The job shop scheduling problem: Conventional and new solution techniques**. In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, v. 93, p. 1–33, 1996.

DORNDORF, U.; PESCH, E. **Evolution based learning in a job shop environment**. In: COMPUTERS AND OPERATIONS RESEARCH, v. 22, p. 25–40, 1995.

GAREY, M. R.; JOHNSON, D. S.; SETHI, R. **The complexity of flowshop and jobshop scheduling**. In: MATHEMATICS OF OPERATIONS RESEARCH, v.1, p. 117–129, 1976.

GLOVER, F.; LAGUNA, M. **Tabu Search**. In: KLUWER ACADEMIC PUBLISHERS, p. 382, 1997.

GÓMEZ, A. T. **Modelo para o sequenciamento de partes e ferramentas em um sistema de manufatura flexível com restrições às datas de vencimento e à capacidade do magazine**. PhD (Thesis) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, São Paulo, Brasil, 1996.

GONÇALVES, J. F.; MENDES, M. J. J.; RESENDE, M. G. C. **A hybrid genetic algorithm for the job shop scheduling problem**. In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, v. 67, p. 77–95, 2005.

JAIN, A. S.; MEERAN, S. **Deterministic job-shop scheduling: Past, present and future**. In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, v. 113, n. 2, p. 390–434, 1998.

JHA, N. K. **Handbook of Flexible Manufacturing Systems**. ACADEMIC PRESS LIMITED, p. 328, 1991.

KUSIAK, A.; CHOW, W. **Efficient solving of the group technology problem**. In: JOURNAL OF MANUFACTURING SYSTEMS, v. 6, n. 2, p. 117–124, 1987.

LAWRENCE, S. **Resource constrained project scheduling**: an experimental investigation of heuristic scheduling techniques. Graduate school of industrial administration, Carnegie Mellon University: Pittsburgh, 1984.

MUTH, J. F.; THOMPSON, G. L. **Industrial Scheduling**. Englewood Cliffs, Prentice Hall, p. 225–251, 1963.

NOWICKI, E.; SMUTNICKI, C. **A Fast Taboo Search Algorithm for the Job Shop**. IN: MANAGEMENT SCIENCE, v. 42, p. 797–813, 1996.

SLACK, N.; JOHNSTON, R.; CHAMBERS, S. **Administração da Produção**. In: ATLAS, 1997.

STECKE, K. E. **A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems**. In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, v. 24, p. 369–375, 1986.

WANG, L.; ZHENG, D. **An effective hybrid optimization strategy for job-shop scheduling problems**. In: COMPUTERS AND OPERATIONS RESEARCH, v. 28, p. 585–596, 2001.

ZOGHBY, J.; BARNES, W. L.; HASENBEIN, J. J. **Modeling the Reentrant job shop problem with setups for metaheuristic searches**. In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH, v. 167, p. 336–348, 2004.