



PROPOSTA DE CLASSIFICAÇÃO HIERARQUIZADA DOS MODELOS DE SOLUÇÃO PARA O PROBLEMA DE *JOB SHOP SCHEDULING*

Ricardo Ferrari Pacheco
Miguel Cezar Santoro

Departamento de Engenharia de Produção
Escola Politécnica da Universidade de São Paulo
E-mails: rpacheco@usp.br, santoro@usp.br

Resumo

Este artigo propõe uma classificação hierarquizada dos modelos utilizados na solução do problema de programação da produção intermitente do tipo job shop, incluindo tanto os que fornecem solução ótima, quanto os modelos heurísticos mais recentes baseados em métodos de busca estendida. Por meio dessa classificação obteve-se um painel amplo dos modelos existentes, evidenciando as diferentes abordagens do problema e suas soluções, com o objetivo de proporcionar uma orientação preliminar na escolha do modelo de job shop scheduling mais adequado.

***Palavras-chave:* job shop scheduling, modelos heurísticos, modelos de solução ótima.**

1. Introdução

Classificar significa organizar coisas em grupos tomando por base características comuns relevantes. O entendimento das características consideradas na classificação de determinada área do conhecimento permite compreender com maior clareza as semelhanças e diferenças entre os diversos grupos e ter uma

visão conjunta dos mesmos. De outra forma, pode-se dizer que as classificações utilizam critérios para efetuar uma partição numa determinada população, de forma que todo e qualquer elemento da população deve pertencer a uma e apenas uma parte. Classificações hierarquizadas constituem-se de partições, segundo diferentes critérios, encadeadas por ordem de importância, ou seja, um grupo de uma classificação (segundo

		PARA					Saída
		Maq 1	Maq 2	Maq 3	Maq 4	Maq 5	
DE	Entrada	.3	-	.2	.1	.4	
	Maq 1		-	.3	-	.5	.2
	Maq 2	.1		-	.6	-	.3
	Maq 3	.5	-		.1	.1	.3
	Maq 4	-	.2	.2		-	.6
	Maq 5	.1	.1	.1	.6		.1

Figura 1 – Matriz de transferência de um *job shop* geral composto de 5 máquinas.
(Fonte: HAX & CANDEA; 1984, p. 269)

determinado critério) é particionado em grupo segundo critério classificatório de importância menor, e esse procedimento continua até terminarem os critérios relevantes. Assim, para uma dada hierarquia de classificação, grupos de indivíduos possuem maior similaridade quanto maior for a quantidade de classes (“pais”) comuns. A principal preocupação dos autores neste trabalho foi ressaltar as diferentes formas de abordagem do problema de programação da produção intermitente do tipo *job shop*, classificando-as de forma hierarquizada. Os autores basearam-se em textos consagrados, que tratam direta ou indiretamente de classificação e caracterização de modelos da área.

De acordo com MORTON & PENTICO (1993), a programação da produção, ou *scheduling*, envolve a consideração de uma série de elementos que disputam vários recursos por um período de tempo, recursos esses que possuem capacidade limitada. Os elementos a serem processados são chamados de ordens de produção ou *jobs* e são compostos de partes elementares chamadas atividades ou operações.

Os objetivos geralmente tratados no problema de *scheduling* podem ser resumidos no atendimento de prazos (ou datas de entrega), na minimização de tempo de fluxo ou dos estoques intermediários e na maximização da utilização da capacidade disponível, ou mesmo na combinação destes (função objetivo multicritério).

Uma grande parte dos problemas de programação estudados aplica-se ao ambiente conhecido como *job shop*. O *job shop* dito geral é

caracterizado por permitir diferentes fluxos das ordens entre as máquinas e diferentes números de operações por ordem, que são processadas apenas uma vez em cada máquina. Um ambiente *job shop* geral pode ser caracterizado por uma matriz de transferência, conforme o exemplo da Figura 1. Os valores das células representam a probabilidade de uma ordem, estando na máquina da linha, ter sua próxima operação executada na máquina da coluna correspondente na tabela.

Pode-se considerar o *flow-shop scheduling* como um caso particular de *job shop scheduling*. ” O *flow shop* contém m máquinas, e as ordens compõem-se de seqüências de operações estritamente ordenadas. Todos os movimentos devem ter uma direção uniforme. Um exemplo é uma linha de montagem com estações de trabalho compostas de uma máquina. Deve-se notar, entretanto, que no *flow shop* geral, algumas ordens possuem menos do que m operações. ” (HAX & CANDEA, 1984, p. 268). Um *flow shop* pode ser descrito por uma matriz de transferência de trabalho na qual as máquinas que compõem o fluxo do produto podem ser arranjadas de forma que, para qualquer célula da matriz de linha j e coluna k , seu valor será nulo se $j \geq k$. (HAX & CANDEA, 1984, p. 268).

Há na literatura diversas classificações hierarquizadas dos problemas de *job shop*. (HAX & CANDEA, 1984; MORTON & PENTICO, 1993; PINEDO, 1995). Uma infinidade de modelos tem sido proposto para a solução de problemas de *job shop scheduling* considerando diversas configurações do problema, objetivos,

restrições consideradas na modelagem (LOPEZ *et al.*, 1995). Alguns exemplos podem ser encontrados em GRAVES (1981), GUPTA & KYPARISIS, (1987), RODAMMER & WHITE (1988), RAMASESH (1990), SURESH & CHAUDHURI (1993). No entanto, as classificações encontradas parecem ser simplificadas, considerando dois ou três aspectos para diferenciá-los, deixando ainda no mesmo grupo modelos bastante distintos, ou parciais, não abordando modelos mais recentes, tais como os que utilizam algoritmos heurísticos de busca.

Por outro lado, tem-se observado uma grande distância entre a quantidade de modelos propostos e a quantidade de aplicações práticas relatadas dos mesmos na programação da produção nas empresas (KING, 1976; GRAVES, 1981; MCKAY *et al.*, 1988; RODAMMER & WHITE, 1988; BUXEY, 1989; MCKAY *et al.*, 1989; WIERS & VAN DER SCHAAF, 1997).

Acredita-se que uma classificação hierarquizada e mais abrangente dos modelos de *scheduling* seja uma contribuição importante no sentido de tornar mais claras as diferenças e semelhanças de abordagens entre eles, sendo um ponto de partida para a complexa discussão dos aspectos que devem envolver a escolha do modelo mais adequado à solução de problemas reais.

Este artigo está dividido em 6 seções. Na seção 2 discute-se as principais classificações dos problemas de *job shop scheduling*. Um breve resumo dos principais modelos utilizados na solução do problema é apresentado na seção 3. Na seção 4 apresenta-se as principais classificações dos modelos encontrados na bibliografia. Uma nova classificação dos modelos existentes é proposta na seção 5. Na seção 6 são feitas as considerações finais e sugestões de trabalhos futuros.

2. Classificações dos Problemas de *Job Shop Scheduling*

Os problemas de *job shop scheduling* são classificados na literatura de diversas formas. Tais classificações podem considerar o

número de máquinas envolvidas, roteiros das ordens, regime de chegadas das ordens, variabilidade dos tempos de processamento, entre outras características. As classificações consideradas importantes para este trabalho são apresentadas a seguir.

- Quanto ao número de máquinas envolvidas (HAX & CANDEA, 1984): de acordo com essa classificação, tem-se problemas de programação onde o ambiente é composto por uma, duas, três ou mais (m) máquinas. Para uma ou duas máquinas, existem algoritmos simples que garantem a solução ótima para determinados objetivos. No caso de três máquinas em série (*flow shop*), o objetivo de minimização do horizonte de programação ou duração total da programação (*makespan*) possui modelo com solução ótima em caso bastante particular (JOHNSON, 1954). Nos demais casos, apenas a enumeração explícita ou implícita de todas as alternativas possíveis garante a solução ótima (problema NP-completo).

HAX & CANDEA (1984) tratam o problema de máquinas paralelas como uma extensão do problema de uma máquina, sendo que neste caso, a operação poderia ser executada por um conjunto de máquinas paralelas, com tempo de processamento distinto para cada uma delas. Já MORTON & PENTICO (1993) utilizam o conceito de centro (ou *cluster*). Desta forma, um conjunto de máquinas paralelas pode ser visto como um centro, ou uma única macromáquina.

- Quanto a forma de atendimento da demanda, segundo classificação dada por HAX & CANDEA (1984, p. 261), o *job shop* pode ser classificado em aberto, fechado ou misto. No caso de *job shop* fechado, os produtos ou serviços são conhecidos e suas demandas podem, em princípio, ser previstas. A produção pode ser feita em lotes maiores do que os pedidos sendo o excesso estocado (MTS, ou *make-to-stock*). Neste caso, a preocupação do administrador é, em geral, a rapidez de atendimento.

No caso do *job shop* aberto, os produtos são fabricados sob encomenda, ou seja, fabricados de acordo com pedidos de clientes (MTO, ou

make-to-order). Como observa HAX & CANDEA (1984, p. 262), *job shops* abertos puros são raros, uma vez que os administradores tentam encontrar formas de aliviar as desvantagens de se produzir estritamente sob pedido.

Job shop mistos combinam as características dos *job shop* abertos e fechados.

- Quanto ao processo de chegada das ordens, classificam-se em estáticos, quando as ordens a serem programadas estão todas disponíveis no início da programação, ou dinâmicos, quando as ordens vão chegando ao longo do tempo. Chegadas dinâmicas podem ser divididas em determinísticas, quando os instantes de chegada são previamente conhecidos, ou estocásticas, quando os instantes seguem uma distribuição de probabilidade.
- Quanto ao tempo de processamento das ordens, classificam-se em determinísticos, caso em que os tempos de processamento das operações são fixos e conhecidos, ou estocásticos, caso em que os tempos de processamento das operações são representados por uma distribuição de probabilidade.
- Quanto ao tempo de *setup* ou tempos de preparação, classificam-se em dependentes ou independentes. Problemas com tempos de *setup* dependentes são aqueles nos quais o tempo de preparação da máquina para executar uma operação depende de qual operação tenha sido executada anteriormente. Para maiores detalhes ver MORTON & PENTICO (1993), KONDILI *et al.* (1993), BAKER (1995), PINEDO (1995).
- Quanto à existência de restrições de dependência entre operações, classificam-se em tecnologicamente dependentes (devido à espera ou devido à seqüência) ou independentes. Operações tecnologicamente dependentes por seqüência são aquelas nas quais sua execução numa determinada seqüência torna-se desejável ou obrigatória por critérios tecnológicos. Operações dependentes por espera são aquelas nas quais, efetuada determinada operação numa ordem, deve-se obrigatoriamente executar a próxima operação dentro de uma janela de tempo. Ambientes sem filas intermediárias são um caso

particular de restrição de dependência por espera no qual o tempo da mesma deve ser nulo.

3. Descrição dos Principais Modelos Matemáticos de Solução do Problema de *Job Shop Scheduling*

Modelos são representações da realidade. O objetivo da construção de modelos é facilitar a compreensão da realidade ou explicar seu comportamento, possibilitando que melhores decisões sejam tomadas. A grande maioria dos modelos apresentam a propriedade de homomorfismo, significando que, embora se pareçam com a realidade sendo retratada, possuem características de simplificação e abstração (SHANNON, 1975). Simplificação pode significar a desconsideração de aspectos ou informações irrelevantes ou pouco importantes tendo em vista os objetivos do estudo, ou a adoção de hipóteses que facilitem o estudo do problema. Abstração significa concentrar-se nas qualidades essenciais ou no comportamento de algo, representando-o não necessariamente da mesma forma. Assim, um mesmo problema pode ser modelado de diversas maneiras, conforme as simplificações e abstrações assumidas, resultando em representações distintas de uma mesma realidade e podendo levar a diferentes conclusões.

Os principais modelos matemáticos de solução dos problemas de *job shop scheduling* são:

- Modelos de Enumeração Explícita (ou Total): todas as alternativas possíveis são examinadas na busca da solução ótima.
- Modelos de Programação Linear Inteira Mista: caracterizam-se por utilizarem variáveis inteiras e contínuas na função objetivo e/ou em algumas restrições. As duas abordagens encontradas na literatura divergem quanto à forma de consideração do tempo. A abordagem com restrições disjuntivas trabalha com tempo contínuo e restrições de não simultaneidade de duas ou mais operações nas máquinas. A abordagem com unidades discretas de tempo considera explicitamente o problema combinatório de utilização das unidades de tempo do

programa. Exemplos das duas modelagens podem ser encontradas em MORTON & PENTICO (1993) e PINEDO (1995).

- Modelos *Branch and Bound*: utilizam uma estrutura de busca em árvore para a construção da solução, analisando, a cada ramificação, quais os nós que são qualificados, isto é, apresentam um limite inferior da solução (*lower-bound* ou LB) inferior ao limite superior (*upper-bound* ou UB). O *upper bound* representa a solução inicial ou a melhor solução completa já encontrada. Nós que não atendem a este requisito são descartados, eliminando-se todas as ramificações deles decorrentes. Os cálculos dos LB costumam ser simplificados. Os nós qualificados são desdobrados em novas ramificações, de modo a se buscar soluções completas melhores que o UB. Caso sejam encontradas, o valor do UB é atualizado e a busca prossegue, verificando-se implicitamente todas as alternativas, até que nenhum nó possua valor de LB inferior ao UB atual. Maiores detalhes podem ser vistos em MORTON & PENTICO (1993).

- Modelos de Programação Dinâmica: envolvem a otimização de um processo de decisão multiestágio, ou seja:

- um problema é dividido em estágios, ou subproblemas;
- subproblemas são resolvidos seqüencialmente até que o problema inicial seja totalmente resolvido.

A base da programação dinâmica é o princípio da otimalidade (BELLMAN & DREYFUS *apud* GILLET, 1976). Uma estratégia ótima tem esta propriedade se, dado um estado inicial e uma decisão inicial ótima, as decisões restantes também constituem uma estratégia ótima, considerando o estado resultante da decisão anterior.

- Algoritmos Eficientes: caracterizam-se por fornecer solução ótima em casos especiais dos problemas de *scheduling* denominados P-completos. Um algoritmo é denominado eficiente quando o tempo decorrido para se encontrar a solução ótima pode ser expresso por uma função polinomial do tamanho do problema (KARP, 1975).

- Seqüências Predefinidas: modelo no qual a programação das máquinas obedece a seqüências obrigatórias, definidas no instante inicial da programação e obtidas utilizando-se quaisquer regras ou procedimentos formais. Determinadas seqüências podem ocasionar bloqueio do programa ou *gridlock*, ou seja, uma determinada máquina ficará aguardando uma ordem indefinidamente e paralisará o programa, uma vez que a operação anterior desta ordem não pode também ser executada noutra máquina, que também encontra-se paralisada.

- Regras de Liberação (também chamadas de regras de seqüenciação, seqüenciamento, ou despacho): modelo no qual a seqüência a ser obedecida pelo programa é construída simultaneamente ao mesmo tempo, priorizando-se as ordens presentes na fila por meio de regras. Para tal, utiliza-se um procedimento de liberação ou *dispatch procedure* no qual, tão logo uma máquina torna-se disponível, inicia-se a operação da ordem em primeiro lugar na fila. A prioridade dos *jobs* pode ser dada estática ou dinamicamente (dependendo do momento de sua atribuição), local ou globalmente (regras que analisam outras filas para a definição da prioridade) ou de forma míope. Regras míopes são aquelas nas quais a prioridade é atribuída “otimizando-se” o problema em cada máquina independentemente das outras, sem preocupação com otimização global. A mesma regra de liberação pode ser utilizada em todas as máquinas, ou regras diferentes podem ser utilizadas em máquinas distintas. Existem inúmeras regras de liberação citadas na literatura (CONWAY, 1965 (a) e (b); PANWALKAR & ISKANDER, 1977).

- Seqüências Predefinidas com Procedimento de Liberação: a seqüência do problema é predefinida, utilizando-se ainda procedimentos de liberação de forma que, caso a seqüência original não possa ser executada (*gridlock*), seja introduzida na máquina a ordem de maior prioridade presente na fila. É importante observar que, neste caso, se a seqüência original for gerada por uma regra estática no instante

inicial, obtém-se o mesmo programa que o gerado pela regra de liberação correspondente.

- Regras com Inserção de Esperas: admitem a possibilidade de não alocarem imediatamente uma ordem a uma máquina desocupada mesmo que exista alguma na fila. Tal tipo de regra costuma ser utilizada em ambientes nos quais não se admite ou é antieconômica a interrupção (*preemption*) de uma ordem, e em problemas com *setups* dependentes.
- Redes Neurais: são heurísticas que utilizam uma malha de uma ou mais camadas de nós (ou neurônios) para tomada de decisões. Cada neurônio fornece um valor de resposta de acordo com sua função de ativação (cujos parâmetros controlam sua sensibilidade), e com o estímulo recebido (valor de entrada, que é obtido pelo somatório das respostas dos neurônios antecedentes). Seu valor de resposta, por sua vez, influencia o comportamento dos neurônios sucessores. O comportamento de cada neurônio, explicitado pelo valor dos parâmetros que controlam a sensibilidade da função de ativação é “aprendido” pela malha, utilizando-se um conjunto exemplo de problema-resposta. Uma vez treinada a malha (ajustados os parâmetros), o processo de tomada de decisões é simples, em termos de cálculos. Maiores detalhes podem ser encontrados em ZAHEDI (1991).
- Métodos de Gargalo: heurísticas que se concentram nos gargalos (*bottlenecks*) do sistema (recursos limitantes de capacidade) para a determinação das seqüências das ordens nas máquinas. É um método bastante utilizado em problemas cujo objetivo é a minimização de tempo total de processamento (*makespan*). Métodos de gargalo único programam as operações de modo a obter a melhor seqüência para a máquina definida como gargalo, fazendo com que as seqüências nas outras máquinas respeitem a determinada para a máquina gargalo. O método do gargalo agregado, chamado por MORTON & PENTICO (1993, p. 376 e 383) de método do corte gargalo, de forma semelhante ao gargalo único, procura resolver o problema de seqüência para um conjunto gargalo de

máquinas, dividindo o problema em três problemas de seqüência: antes do gargalo, no gargalo conjunto de máquinas e depois do gargalo. Já o método do gargalo móvel procura determinar a seqüência gargalo a gargalo, sendo que a seqüência determinada no primeiro gargalo (mais importante) serve de restrição ao segundo e assim sucessivamente. Maiores detalhes podem ser encontrados em MORTON & PENTICO (1993, p. 383).

- Gargalo Dinâmico: heurística dinâmica com possibilidade de inserção de esperas que procura estimar o custo econômico de atraso de uma ordem de uma unidade de tempo, bem como o custo econômico de parada (ou de uso) de cada recurso, também por uma unidade de tempo. Genericamente falando, a relação entre o benefício de se executar uma operação e o custo de utilizar o recurso dá a prioridade de cada operação nas máquinas. Essas relações benefício/custo são calculadas dinamicamente e admitem procedimentos diversos de cálculo, com maiores ou menores aproximações. Maiores detalhes podem ser encontrados em MORTON & PENTICO (1993, p. 164).
- *Beam Search*: heurística derivada do *branch and bound* que procura avaliar a relevância de se explorar cada alternativa de cada nó da árvore de busca utilizando uma função de avaliação simplificada (por exemplo, o *lower-bound* de um subproblema). Entretanto, no caso do *beam search*, tomando-se por base os resultados da função de avaliação das alternativas, continua-se a explorar apenas as M alternativas que pareçam ser mais promissoras em cada nó (onde M é chamado de tamanho ou amplitude do *beam*).

O *beam search* pode ser classificado em guiado global ou guiado localmente. No caso do *beam search* guiado global, comparam-se alternativas de níveis diferentes da árvore. Desta forma, mesmo com o corte dado pelo *beam* de tamanho M, centenas ou milhares de avaliações podem ser necessárias, pois se houver N níveis na árvore, pode-se ter até M^N alternativas completas exploradas. Já o *beam search* guiado local busca reduzir a exploração dos ramos da

árvore de busca. Neste caso, todas as alternativas em cada nível da árvore são ordenadas por uma função de avaliação simplificada, que considera apenas a parte da solução já construída. A avaliação é, na verdade, apenas uma estimativa ou “palpite” de quão promissora parece ser uma alternativa num determinado nível. As M melhores alternativas são expandidas a cada nível. Ao final, tem-se M alternativas completas consideradas, escolhidas dentre as mais promissoras, conforme a função de avaliação utilizada. Maiores detalhes são encontrados em OW & MORTON (1988).

- **Sistemas Especialistas:** método heurístico e interativo que visa decidir qual a melhor seqüência a ser executada, dependendo das condições atuais da produção. Ele mantém, armazenadas em bases de dados de conhecimento, denominadas *knowledge bases*, a configuração, as restrições do problema, bem como um conjunto de regras lógicas a serem utilizadas na decisão. A configuração e as restrições do problema dizem respeito, por exemplo, aos tempos das operações, número e características das máquinas, restrições tecnológicas de produção, etc. Regras lógicas traduzem a forma de raciocínio do especialista (tamanhos aceitáveis de lotes, preferências de seqüências de operações, etc.) que são, muitas vezes, de difícil compreensão e formalização.

Bases de conhecimentos são estruturas complexas, geralmente compactas, modulares e hierarquizadas. Os módulos de dados armazenados nas bases são denominados *frames*.

Desta forma, é possível “imitar” as regras utilizadas por especialistas humanos. O algoritmo computacional, que seleciona hierarquicamente quais regras devem ser utilizadas numa determinada condição é chamado motor de inferência. *Shells* são ambientes que facilitam a construção das bases de conhecimento e sua varredura (por motores de inferência). Segundo MORTON & PENTICO, algumas das formas de se obter regras (etapa que é denominada aquisição de conhecimento) são:

- ⇒ estudar a forma de trabalho de um tomador de decisões humano. Seu raciocínio é formalizado em regras que são incorporadas ao sistema especialista após serem validadas pelo tomador de decisões humano.
- ⇒ uma série de regras são propostas para o problema por especialistas em pesquisa operacional, implementadas pelo construtor do *software* e validadas pelo tomador de decisões humano.
- ⇒ o construtor do sistema simplifica heurísticas existentes, criando regras simples, tal como o tomador de decisão humano provavelmente faria. Tais regras são validadas pelo mesmo.

Visto sob o aspecto combinatorial do problema de *scheduling*, os sistemas especialistas efetuam uma busca na árvore de alternativas, auxiliados por regras que descartam seqüências e priorizam outras, conforme o estado do sistema. Maiores detalhes sobre sistemas especialistas são encontrados em HAYES-ROTH *et al.* (1983).

- **Métodos de Busca em Vizinhança:** são assim denominados porque procuram obter soluções cada vez melhores do problema, caminhando para pontos vizinhos do espaço de alternativas ou de busca. Podem utilizar estratégias de intensificação e/ou diversificação. Denomina-se intensificação o processo em que somente se caminha de uma solução para outra melhorada (mais próxima da ótima) ou subótima. Se, de uma determinada solução, não se obtém qualquer melhoria pela exploração de soluções vizinhas, pode-se optar pela técnica de diversificação, ou seja, aceitar a exploração de algumas mudanças que forneçam soluções piores como forma para fugir de ótimos locais. Pode-se classificar os métodos de busca em vizinhança em probabilísticos, de busca direta ou de busca estendida.

- **Métodos de Busca em Vizinhança Probabilísticos:** consideram um grupo de regras de liberação, sendo que a cada seleção de uma ordem a ser executada escolhe-se, por meio de um sorteio, qual regra será utilizada, repetindo-se este sorteio de regras a cada ingresso

em máquina, até o final do programa. O processo é repetido diversas vezes e escolhe-se o melhor programa obtido. Maiores detalhes em MORTON & PENTICO (1993).

- Métodos de Busca em Vizinhança Direta: heurísticas nas quais, dada uma solução completa inicial, procura-se caminhar para uma solução melhorada (intensificação) sem o recurso da diversificação. Maiores detalhes sobre os conceitos de busca em vizinhança são encontrados em STORER *et al.* (1992).
- Métodos de Busca em Vizinhança Estendida: assim como o método anterior, procuram explorar a troca da ordem de execução entre ordens, buscando soluções melhoradas. Esta nova solução é tomada como atual, e novas vizinhanças são novamente exploradas (intensificação). Entretanto, diferentemente do método de busca direta, soluções inferiores às já obtidas podem ser aceitas (diversificação), como forma de escapar de ótimos locais. Exemplos de métodos de busca estendida em vizinhança são *tabu search*, *simulated annealing* e algoritmos genéticos, descritos em GLOVER (1990); DELLAMICO & TRUBIAN (1993); VAN LAARHOVEN *et al.* (1992) e BEAN (1994).

4. Classificações dos Principais Modelos

Poucos são os trabalhos que classificam lógica e consistentemente os modelos aplicáveis ao problema de *job shop scheduling* e que, com isso, permitem a formação de um quadro conceitual amplo que facilite o entendimento dos modelos e a escolha destes para aplicações específicas.

A maioria das classificações agrupa os modelos de acordo com o número de máquinas que consideram. Outras classificações enfatizam o processo de chegada das ordens (estático, dinâmico determinístico ou dinâmico probabilístico), a forma de consideração dos tempos de operação (determinístico ou estocástico) ou os objetivos da programação.

Segundo CONWAY *et al.* (1967), as abordagens para solução do problema de programação

dividem-se em aplicáveis a problemas estáticos (programação inteira, *branch and bound*, heurísticas de um passo ou com ajuste dos parâmetros, com ou sem inserção de ociosidade) e a problemas dinâmicos (modelos com base na teoria de processos estocásticos ou simulação). A classificação de CONWAY *et al.* retrata o panorama de pesquisa dos métodos de solução de problemas combinatoriais da década de 60, já que não considera a existência dos atuais algoritmos heurísticos de busca.

HAX & CANDEA (1984) classificam hierarquicamente as abordagens de solução conforme o ambiente da modelagem do problema, separando os modelos em aplicáveis a problemas estáticos determinísticos e naqueles aplicáveis a problemas dinâmicos probabilísticos.

- Estáticos-Determinísticos
 - Programação Linear Inteira;
 - Algoritmos Construtivos;
 - De enumeração exaustiva;
 - Branch and Bound;
 - Procedimentos Heurísticos;
- Dinâmicos-Probabilísticos
 - Abordagem Analítica;
 - Abordagem Experimental;
 - Em ambientes reais;
 - Simulações baseadas em ambientes reais com objetivos operacionais;
 - Simulações conduzidas com objetivos de pesquisa.

Denominam-se algoritmos construtivos aqueles nos quais a solução inicial é vazia e, a cada passo do algoritmo, uma ou mais operações são associadas às máquinas, tendo-se, ao final dos passos, a seqüenciação e programação completas.

MORTON & PENTICO fazem diversas classificações. Uma delas divide os métodos em dois grupos: métodos básicos e avançados (ou recentes). Os principais métodos considerados por eles como sendo básicos são: manuais de intervalos, *dispatch* manual, simulação de *dispatch*, matemáticos-exatos, matemáticos-heurísticos, sistemas especialistas puros e sistemas híbridos, que misturam técnicas de

inteligência artificial (IA), pesquisa operacional (PO) e sistemas de apoio à decisão (SAD's).

Os métodos mais recentes, utilizando abordagens heurísticas, são classificados por MORTON & PENTICO (1993) em três categorias: métodos de Intensificação e Diversificação (*tabu search*, *simulated annealing*, algoritmos genéticos e *beam search*), métodos de gargalos e sistemas especialistas (ou híbridos).

A classificação de MORTON & PENTICO não hierarquiza as abordagens de solução quanto à estrutura de busca executada para sua obtenção, sendo basicamente uma classificação histórica.

Segundo outra classificação, também feita por MORTON & PENTICO, tem-se:

- Métodos Exatos, com formulação por restrições disjuntivas ou com unidade de tempo discreto.
- Métodos Heurísticos de Um Passo, divididos em multipermutação, regra de liberação simples, heurísticas de um passo míopes e gargalo simples.
- Métodos Heurísticos de Busca, divididos em *dispatch* aleatório guiado, *beam search* guiado (total ou local) e busca em vizinhança.
- Métodos Heurísticos Iterativos, divididos em métodos de *dispatch* iterativo, métodos de gargalo (agregado ou móvel) ou gargalo dinâmico.

Por esta classificação, dividiu-se os métodos inicialmente em exatos e heurísticos. Os heurísticos foram subdivididos em métodos de passo único, métodos de busca e métodos iterativos. Métodos de passo único fornecem uma solução com apenas uma execução do algoritmo. Métodos de busca procuram gerar diversas soluções, escolhendo a melhor delas. Métodos iterativos procuram buscar o melhor valor do conjunto de parâmetros, de modo a obter uma boa solução.

Os modelos costumam ainda ser classificados quanto à inserção ou não de espera ou tempo ocioso entre operações, e quanto à possibilidade de interromper operações (HAX & CANDEA). Segundo estas classificações, o problema de

programação pode admitir abordagens como:

- 1) Modelos que permitem ou não a inserção de espera, ou seja, uma máquina pode ficar esperando desocupada por uma determinada ordem, mesmo com outras ordens na fila.
- 2) Modelos sem a possibilidade de interrupção da ordem sendo processada, ou com a permissão de interrupção de uma ordem, seja com o recomeço da ordem interrompida a partir do início (*repeat*) ou com o recomeço do ponto em que houve a interrupção (*resume*).

Outras formas de classificações dos modelos de solução são:

- Modelos de construção de programa (tais como programação dinâmica e métodos de gargalo) ou modelos de alteração de programa (tais como os de busca em vizinhança).
- Modelagem centrada no gargalo ou na ordem.

5. A Classificação Proposta

A classificação hierarquizada proposta neste artigo para os modelos de *job shop scheduling* está resumida na Figura 2. Os retângulos sombreados com cantos arredondados contém os modelos. Retângulos sombreados com bordas pontilhadas representam detalhes relevantes dos modelos. Retângulos sem sombreamento representam as classes de modelos. Os números indicados na legenda representam os critérios utilizados para partição de uma classe em subclasses.

Como foi dito anteriormente, o problema de *job shop scheduling* é um problema combinatório, na maioria das vezes NP-completo. Desta forma, algoritmos otimizantes são computacionalmente viáveis quando aplicados a problemas reais pequenos, com objetivos limitados. Para problemas de porte similar aos encontrados no ambiente real, costuma-se sacrificar a obtenção de uma solução otimizada por uma solução subótima obtida por métodos heurísticos, com tempo computacional aceitável.

Por esta razão, os modelos estão classificados inicialmente em duas classes: modelos de solução ótima e modelos heurísticos.

5.1 Modelos de Solução Ótima: modelos que geram soluções ótimas, com exceção dos que possuem algoritmos eficientes, caracterizam-se pelo uso de técnicas genéricas para solução de problemas combinatoriais. Essas técnicas genéricas são procedimentos enumerativos que não utilizam informação ou conhecimento de características específicas do problema para direcionar a busca. Tais modelos podem ser utilizados nos problemas de *scheduling* ou em qualquer outro problema combinatorial, diferindo pela forma como são definidas a função objetivo, as restrições e os parâmetros.

Dentre os modelos de solução ótima enumerativos, a enumeração pode se dar explicitamente (todas as alternativas tem o resultado calculado) ou implicitamente (algumas combinações não são exploradas por possuírem resultado parcial deduzido como inferior a outro já obtido).

Modelos de solução ótima de enumeração implícita diferenciam-se pelas particularidades de sua implementação e eficiência computacional. Os modelos desta categoria são os de programação inteira, *branch and bound* e programação dinâmica.

Já os algoritmos eficientes particularizam-se como uma subclasse dos modelos otimizantes por gerarem soluções ótimas sem a necessidade de enumeração de todas as alternativas. Infelizmente, poucos são os problemas cuja solução pode ser encontrada por algoritmos eficientes e, mesmo os existentes, possuem hipóteses pouco aderentes aos problemas normalmente encontrados na realidade.

Portanto, os modelos de solução ótima, subclassificados pela forma de enumeração das alternativas, são apresentados na Tabela 1.

Modelos de programação inteira podem ainda ser classificados quanto à forma de consideração do tempo por restrições disjuntivas ou unidade de tempo discreto.

5.2 Modelos Heurísticos: propiciam que uma boa solução, não obrigatoriamente ótima, seja encontrada em tempo adequado, podendo utilizar duas estratégias distintas:

- construir a seqüência e o programa solução de uma só vez (heurística de passo único) e sem consideração simultânea de alternativas, procedimento para o qual existe uma expectativa ou crença de bom desempenho.
- explorar um subconjunto do espaço de soluções possíveis para o problema (heurísticas de busca).

5.2.1 Modelos Heurísticos de Passo Único: tanto os modelos heurísticos de passo único quanto as de busca podem focalizar a seqüência das operações ou os gargalos existentes.

5.2.1.1 Modelos Heurísticos de Passo Único Orientados à Seqüência: para esses modelos, a capacidade de se produzir boas soluções pode ser conseguida de duas formas: pode ser aprendida pelo modelo ou informada. Na primeira categoria (aprendida) encontram-se os modelos de redes neurais, nos quais o modelo “aprende” a melhor forma de resolver o problema de *scheduling* por meio de um conjunto-exemplo de problemas que “calibra” a malha de neurônios. Na segunda categoria (informada), o procedimento de resolução do problema (ou lógica) é incorporado na modelagem. Isto significa que o problema é resolvido gerando-se uma única solução, presumivelmente boa, considerando as particularidades do problema, de forma simplificada. Assim, troca-se a garantia de uma solução ótima (como se fossem avaliadas todas as alternativas), pela simplicidade de se obter uma boa solução com apenas uma tentativa.

5.2.1.1.1 Modelos Heurísticos de Passo Único Orientados à Seqüência com Lógica Informada: esta abordagem pode se dar de três formas: por seqüências predefinidas, por seqüências predefinidas associadas a procedimentos de liberação ou por regras de seqüenciação.

As regras de seqüenciação (não confundir com regras de liberação) são divididas em duas categorias conforme o tipo de informação que utilizam para definir a seqüência: regras que consideram informações das ordens e das filas e regras que, além disso, consideram custos econômicos das ordens e das máquinas.

Tabela 1 – Subclasses de modelos de solução ótima.

Subclasses	Modelos
de enumeração explícita	enumeração total
de enumeração implícita	programação inteira, <i>branch-and-bound</i> , programação dinâmica
sem enumeração	eficientes

As regras que consideram ordens e filas, podem ainda ser classificadas quanto a possibilidade ou não de ociosidade, sendo denominadas respectivamente regras com inserção de espera e regras de liberação (regras de seqüenciação associada a procedimento de liberação).

As regras que consideram, além das ordens e filas, os custos econômicos de ordens e máquinas constituem os chamados modelos de gargalo dinâmico, podendo diferenciar-se pela possibilidade de se inserir ou não esperas na programação.

5.2.1.2 Modelos Heurísticos de Passo Único Orientados a Gargalo e depois à Seqüência: se caracterizam por procurar resolver o problema de seqüência inicialmente na(s) máquina(s) mais crítica(s) com respeito à limitação de capacidade. Os modelos existentes se diferenciam pela quantidade e pela forma de determinação dos gargalos. Os três modelos desta categoria são os modelos de gargalo único, gargalo agregado e gargalo móvel.

5.2.2 Modelos Heurísticos de Busca: procuram analisar um subconjunto das alternativas do espaço de soluções viáveis, retornando como resultado final a melhor solução encontrada. Desta forma, utilizam uma carga computacional intermediária entre heurísticas de passo único e métodos otimizantes. Heurísticas de busca podem ser divididas em duas categorias quanto a existência ou não de mecanismos de intensificação: a busca pode se dar aleatoriamente (permutar aleatoriamente ordens, avaliando os resultados obtidos), ou com intensificação, com ou sem diversificação. As heurísticas de busca com intensificação podem se diferenciar quanto ao foco de atenção que possuem, podendo ser orientadas exclusivamente à seqüência ou aos

gargalos e à seqüência. Modelos de busca orientados a gargalos caracterizam-se como sendo aqueles nos quais a escolha das vizinhanças leva em consideração a importância relativa das máquinas com restrição de capacidade (gargalo), aumentando a probabilidade da vizinhança escolhida na busca corresponder a uma seqüência de máquina limitante de capacidade e não a uma máquina com ociosidade. Não foram localizadas propostas de modelos de busca orientada a gargalos, embora os autores considerem uma abordagem viável e eventualmente inexplorada de modelagem do problema.

5.2.2.1 Modelos Heurísticos de Busca com Intensificação: podem ser divididos quanto à estrutura de dados utilizada, em duas categorias: buscas em árvore e buscas em vizinhança.

5.2.2.1.1 Modelos Heurísticos de Busca com Intensificação por meio de Estrutura em Árvore: exploram uma parte de todas as ramificações de seqüência possíveis. Nesta categoria encontram-se os modelos de *beam search* global e local e os sistemas especialistas.

5.2.2.1.2 Modelos Heurísticos de Busca com Intensificação em Vizinhança: exploram as trocas de seqüência entre as ordens, trabalhando portanto com a avaliação de soluções completas e se diferenciando pela existência ou não de mecanismos de diversificação.

5.2.2.1.2.1 Modelos Heurísticos de Busca com Intensificação em Vizinhança Direta: buscas em vizinhança direta caracterizam-se inexistência de mecanismos de diversificação, isto é, somente mudam de uma solução para outra melhor. Desta forma, a exploração pode ficar restrita a uma região e obter um ótimo local.

5.2.2.1.2.2 Modelos Heurísticos de Busca Estendida: métodos de busca em vizinhança

que utilizam mecanismos de diversificação são chamados de métodos de busca estendida, por permitirem que a busca se estenda explorando alternativas vizinhas piores do que a solução atual, mecanismo este que permite fugir de ótimos locais. Nesta última categoria encontram-se os modelos denominados *tabu search*, *simulated annealing* e algoritmos genéticos.

6. Conclusões

Este trabalho enfocou as principais formas de classificação tanto do problema de *job shop scheduling* quanto dos modelos de solução existentes na literatura, propondo uma nova classificação hierarquizada dos modelos.

A principal diretriz da classificação hierarquizada foi quanto a otimalidade ou não do modelo. Abaixo de cada uma dessas duas grandes classes, os critérios de classificação foram escolhidos seqüencialmente, de forma a agrupar e diferenciar os modelos, segundo características cada vez menos importantes dos procedimentos

de obtenção de solução inerentes aos mesmos. Logicamente, esses critérios são diferentes para modelos de solução ótima e heurísticos.

A classificação dos modelos existentes de forma hierarquizada pretende ser uma contribuição para compreensão das abordagens de cada um, fornecendo um painel amplo e comparativo do estado da arte na modelagem do problema de *scheduling*.

Espera-se que este artigo contribua também para trabalhos que tratem, por exemplo, da adequação dos modelos aos problemas reais, onde seriam analisadas as qualidades das soluções, as eficiências computacionais dos modelos e classes de modelos, essas últimas considerando aderência à realidade, facilidade de entendimento do modelo ou transparência, praticidade de parametrização e robustez.

Foi ainda detectada uma forma alternativa de modelagem do problema, baseada em buscas com consideração de gargalos, cujos desdobramentos podem resultar em trabalhos futuros.

Referências Bibliográficas

- ASHOUR, S.:** *Sequencing Theory*, Springer-Verlag, 1972.
- BAKER, K.R.:** *Introduction to Sequencing and Scheduling*, Wiley, New York, 1974.
- BAKER, K.R.:** *Elements of Sequencing and Scheduling*, Dartmouth College, Hanover, 1995.
- BUFFA, E.S. & MILLER, J.G.:** *Production-Inventory Systems: Planning and Control*, Illinois, Richard D. Irwing Inc., 1979.
- BUXEY, G.:** "Production Scheduling: practice and theory", *European Journal of Operational Research*, 39, 1989.
- COFFMAN, E.G., JR (editor):** *Computer and Job-Shop Scheduling Theory*, Wiley, New York, 1976.
- CONWAY, R.W.:** "Priority Dispatching and Work-In-Process Inventory in a Job Shop", *Journal of Industrial Engineering*, n.16, p.123-130, 1965 a.
- CONWAY, R.W.:** "Priority Dispatching and Job Lateness in a Job Shop", *Journal of Industrial Engineering*, n.16, p.228-237, 1965 b.
- CONWAY, R.W.; MAXWELL, W.L. & MILLER, L.W.:** *Theory of Scheduling*, Addison Wesley, Reading, Mass., 1967.
- DELL'AMICO, M. & TRUBIAN, M.:** "Applying Tabu-Search to the Job Shop Scheduling Problem", *Annals of Operations Research*, n.41, p.231-252, 1993.
- GILLET, B.E.:** *Introduction to Operations Research: A Computer Oriented Algorithmic Approach*, McGraw-Hill, 1976.
- EGLESE, R.W.:** "Simulated Annealing: a tool for operational research", *European Journal of Operational Research*, 1990.
- GLOVER, F.:** "Tabu Search: a tutorial", *Interfaces*, n.20, p.74-94, 1990.
- GRAVES, S.C.:** "A Review of Production Scheduling", *Operations Research*, 29, 1981.
- GUPTA, S.K. & KYPARISIS, J.:** "Single Machine Scheduling Research", *Omega*, 15, 1987.

- HAYES-ROTH, F.; WATERMAN, D.A. & LENAT, D.B.:** *Building Expert Systems*; Addison Wesley, 1983.
- HAX, A & CANDEA, D.:** *Production and Inventory Management*, Englewood Cliffs, N.J., Prentice-Hall, 1984.
- HILLIER, F. & LIEBERMAN, G.:** *Introdução à Pesquisa Operacional*, R.J., ed. Campus, 1988.
- JOHNSON, S.M.:** “Optimal Two and Three Stage Production Schedules with Set-up Times Included”, *Naval Research Logistics Quarterly*, vol.1; pp.61-68, 1954.
- KARP, R.M.:** “On The Computational Complexity of Combinatorial Problems”, *Networks*, v.5, p.45-68, 1975.
- KING, J.R.:** “The Theory-Practice Gap in Job-Shop Scheduling”, *The Production Engineer*, March, 1976.
- KLAHR, P. & WATERMAN, D.A.:** *Expert Systems: Techniques, Tools and Applications*, Addison Wesley, 1986.
- KONDILI, E.; PANTELIDES, C.C. & SARGENT, R.W.H.:** “A General Algorithm for Short-Term Scheduling of Batch Operations-I. MILP Formulation”, *Computers Chem. Engineering*, vol.17; nro.2, 1993.
- LIPPMANN, R.P.:** “An Introduction to Computing with Neural Nets”, *IEEE ASSP*, 1987.
- LOPEZ, O.C.; BARCIA, R.M. & EYADA, O.:** “Problema de Programação da Produção: um Esquema de Classificação”, *Produção*, v.5; n.2; 1995.
- MCKAY, K.N.; SAFEYENI, F.R. & BUZACOTT, J.A.:** “Job shop Scheduling Theory: What is Relevant?”, *Interfaces*, 18, 1988.
- MORTON, T.E. & PENTICO, D.W.:** *Heuristic Scheduling Systems*, John Wiley & Sons, N.Y., 1993.
- OW, P.S. & MORTON, T.E.:** “Filtered Beam Search in Scheduling”, *International Journal of Production Research*, n.26, p.297-307, 1988.
- PANWALKAR, S.S. & ISKANDER, W.:** “A Survey of Scheduling Rules”, *Operations Research*, 25, p.45-61, 1977.
- PINEDO, M.:** *Scheduling : Theory, Algorithms and Systems*, Englewood Cliffs, New Jersey, 1995.
- RODAMMER, F.A. & WHITE, K.P.:** “A Recent Survey of Production Scheduling”, *IEEE Transactions on Systems, Man and Cybernetics*, 18, 1988.
- RAMASESH, R.:** “Dynamic Job-Shop Scheduling: A Survey of Simulation Research”, *Omega*, vol.18, n.1, p.43-57, 1990.
- SHANNON, R.E.:** *System Simulation: the art and science*, Prentice-Hall, 1975.
- STORER, R.H.; WU, S.D. & VACCARI, R.:** “New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling”, *Management Science*, n.38, p.1495-1509, 1992.
- SURESH, V. & CHAUDIHURI, D.:** “Dynamic Scheduling- a Survey of Research”, *International Journal of Production Economics*, 32 (August), 1993.
- VAN LAARHOVEN, P.J.M.; AARTS, E.H.L. & LENSTRA, J.K.:** “Job Shop Scheduling by Simulated Annealing”, *Operations Research*, n.40, p.113-125, 1992.
- WIERS, V.C.S. & VAN DER SCHAAF, T.W.:** “A Framework for Decision Support in Production Scheduling Tasks”, *Production Planning and Control*, vol.8, n.6, 1997.
- ZAHEDI, F.:** “An Introduction to Neural Networks and a Comparison with Artificial Intelligence and Expert Systems”, *Interfaces*, v.21, n.2, p.25-38, 1991.

A PROPOSITION OF HIERARCHICAL CLASSIFICATION FOR SOLUTION MODELS IN THE JOB SHOP SCHEDULING PROBLEM

Abstract

This paper proposes a hierarchical model classification used in the job shop scheduling problem, including those that provide an optimal solution and the more recent ones based on heuristics, called extended search methods. A panel with the existing models is obtained by this classification, and solutions and approach differences are highlighted with the aim of a preliminary orientation on the choice of a more adequate job shop scheduling model.

Key words: scheduling, job shop, classification, models.