

## **Problemas de Programação da Produção: Uma Visão Geral dos Métodos de Solução**

**Márcia de Fátima Morais, Colegiado de EPA, UNESPAR/Campus de Campo Mourão**

**marciamorais.engenharia@gmail.com**

**Rony Peterson da Rocha, Colegiado de EPA, UNESPAR/Campus de Campo Mourão**

**ronypeterson\_eng@hotmail.com**

**Larissa de Carvalho, EPA, UNESPAR/Campus de Campo Mourão**

**larissadecarvalho9@gmail.com**

*Resumo: O campo dos métodos de solução aplicados aos problemas de programação da produção, uma das atividades executadas pelo Planejamento e Controle da Produção, vem crescendo rapidamente, devido à importância dos processos decisórios desta atividade para o mundo industrial. Verifica-se na literatura, a existência de diferentes caminhos para classificar e descrever os métodos de solução, dependendo das características selecionadas para diferenciá-los. Neste contexto, a pesquisa aqui apresentada, que se enquadra nas áreas de engenharia de operações e processos da produção e de pesquisa operacional, tem como objetivo fornecer uma visão geral dos principais métodos de solução aplicáveis aos problemas de programação da produção, por meio de uma classificação proposta. A pesquisa classifica-se quanto aos fins como descritiva e explicativa e, quanto aos meios, como bibliográfica. O método de abordagem adotado foi o qualitativo. Os métodos de soluções para problemas de programação da produção foram categorizados em exatos ou ótimos e em aproximados ou heurísticos. Os métodos de soluções aproximadas foram categorizados em regras de prioridades, heurísticas e metaheurísticas. Foram identificados e analisados sete tipos de métodos de soluções exatas e treze tipos de métodos de soluções aproximadas.*

*Palavras-chave: Programação da Produção; Métodos de Solução; Classificação; Panorama.*

### **1. Introdução**

A pesquisa aqui apresentada enquadra-se em duas das dez grandes áreas de conhecimento de Engenharia de Produção: i) A engenharia de operações e processos da Produção que trata de projetos, operações e melhorias dos sistemas que criam e entregam os produtos (bens ou serviços) primários da empresa; e ii) A pesquisa operacional que engloba a resolução de problemas reais envolvendo situações de tomada de decisão, através de modelos matemáticos habitualmente processados computacionalmente, aplicando conceitos e métodos de outras disciplinas científicas na concepção, no planejamento ou na operação de sistemas para atingir seus objetivos (ABEPRO, 2008).

A programação da produção é uma das atividades executadas pelo planejamento, programação e controle da produção (PPCP) que constitui uma parte central dos processos associados à produção. A necessidade de aperfeiçoamento nos processos produtivos e de decisões tem levado as empresas a buscarem os melhores mecanismos para auxiliar nas

análises de decisões e resultados da empresa. Neste contexto, a pesquisa operacional pode ser caracterizada como um dos mecanismos mais utilizados.

O campo dos métodos de solução aplicados aos problemas de programação da produção está crescendo rapidamente, devido à importância destes processos decisórios para o mundo industrial. Verifica-se na literatura, a existência de diferentes caminhos para classificar e descrever os métodos de solução, dependendo das características selecionadas para diferenciá-los (BLUM; ROLI, 2003).

Diante do exposto, o objetivo deste artigo é fornecer uma visão geral dos métodos de solução aplicáveis aos problemas de programação da produção, por meio de uma classificação proposta. Não se pretende neste estudo fornecer uma prescrição para a construção dos métodos de solução, nem julgar os méritos relativos aos diferentes métodos de solução.

Este artigo encontra-se estruturado em cinco seções. Após a contextualização e ambientalização da pesquisa, o referencial teórico-conceitual referente aos problemas de programação da produção é exposto. Em seguida, é apresentada a metodologia da pesquisa, que contempla uma proposta de classificação para os métodos de solução. Na quarta parte, de acordo com a classificação proposta, os principais métodos de solução para problema de programação da produção são explicitados. Por fim, têm-se as considerações finais.

## **2. Referencial Teórico: Problemas de Programação da Produção**

### **2.1 Caracterização dos Problemas de Programação da Produção**

A programação da produção determina o sequenciamento de tarefas em máquinas, especificando os tempos de início e fim de processamento de cada tarefa. Em outras palavras, “problemas de programação consistem em determinar a ordem ou sequência em que as máquinas irão processar as tarefas de modo a otimizar alguma medida de desempenho” (JOHNSON; MONTGOMERY, 1974). Neste contexto, a programação da produção “pode ser definida como a alocação de recursos ao longo do tempo para executar tarefas para melhor atender um conjunto de critérios pré-definidos” (MACCARTHY; LIU, 1993).

Cada tarefa compreende um conjunto de operações a ser executado. As operações requerem recursos e devem ser executadas de acordo com alguma sequência tecnológica viável, segundo Morton; Pentico (1993) e Pinedo (2008). Mais formalmente, um problema de programação da produção surge em situações em que se requer que um conjunto de  $n$  tarefas  $\{J_1, J_2, \dots, J_j, \dots, J_n\}$  sejam processadas em  $m$  máquinas  $\{M_1, M_2, \dots, M_k, \dots, M_m\}$  disponíveis (FRENCH, 1982).

De acordo com Pinedo (2008), se uma tarefa requer um número de etapas de processamento ou operações, então o par  $(i, j)$  refere-se à etapa de processamento ou operação da tarefa  $j$  na máquina  $i$ . Além disso, em cada operação  $op_{ij}$  existe um tempo de processamento  $p_{ij}$  associado. Segundo Pinedo (2008) cada tarefa  $j$  está associada a uma data a partir da qual a tarefa pode ser executada (*realise date* ou *ready date* -  $r_j$ ), assim como a uma data em que a tarefa deve estar concluída (*due date* -  $d_j$ ). Cada tarefa também pode estar associada a um peso ( $w_j$ ) que é basicamente um fator de prioridade, denotando a importância da tarefa  $j$  relativa a outras tarefas no sistema.

As hipóteses básicas de um problema de programação da produção são: i) Hipóteses sobre as tarefas; ii) Hipóteses sobre as máquinas; e iii) Hipóteses sobre políticas de operações (CONWAY; MAXWELL; MILLER (1967), GUPTA (1979), FRENCH (1982), HAX;

CANDEA (1984), MACCARTHY; LIU (1993) e GUPTA; STAFFORD JR. (2006). Segundo Hax; Candea (1984) uma vez que o problema básico é uma simplificação da maioria das situações reais, sua investigação pode trazer muitas informações úteis para o tratamento dos problemas de programação.

De acordo com Gupta; Stafford Jr. (2006) os pesquisadores no decorrer dos anos vêm relaxando ou modificando as hipóteses básicas de forma a obter problemas mais realistas, ou seja, vêm considerando que a programação da produção está sujeita a restrições que descrevem situações específicas.

## 2.2 Ambientes de Máquinas em Problemas de Programação

Os possíveis ambientes de máquinas em problemas de programação são sumarizados no Quadro 1.

Ambiente de Máquinas	Características
Máquina Única	Existe apenas uma máquina a ser utilizada.
Máquinas Paralelas	Existem duas ou mais máquinas em paralelo, sendo que, uma tarefa, pode ser processada em qualquer uma das máquinas. As máquinas em paralelo podem ser: Idênticas; Proporcionais ou Uniformes; ou Não-Relacionadas.
<i>Flow Shop/Flow Shop Tradicional</i>	Todas as tarefas têm o mesmo fluxo de processamento nas máquinas.
<i>Flow Shop Permutacional</i>	<i>Flow Shop</i> no qual a ordem de processamento das tarefas deve ser a mesma em todas as máquinas.
<i>Job Shop</i>	Cada tarefa tem sua própria ordem de processamento nas máquinas.
<i>Flow Shop</i> com Máquinas Múltiplas ou Híbrido	<i>Flow Shop</i> em que, em pelo menos um dos estágios de produção existe um conjunto de máquinas paralelas.
<i>Job Shop</i> com Máquinas Múltiplas ou Híbrido	<i>Job Shop</i> em que, em pelo menos um dos estágios de produção existe um conjunto de máquinas paralelas.
<i>Open Shop</i>	Não há fluxo definido (específico) para as tarefas serem processadas nas máquinas.

Quadro 1 – Ambientes de Máquinas em Problemas de Programação

Fonte: Elaborado pelos autores com base em MacCarthy; Liu (1993), Pinedo (2008) e Allahverdi; Cheng; Kovalyov (2008)

## 2.3 Características das Tarefas em Problemas de Programação da Produção

Na literatura pode ser verificado um grande número de características relativas às tarefas que podem ser verificadas nos diferentes ambientes de máquinas. As principais características verificadas nos diferentes tipos de problemas de programação são apresentadas no Quadro 2.

Características das Tarefas	Descrição
Datas de Liberação ( $r_j$ )	Datas a partir da qual a tarefa pode ser executada. Se uma data de liberação para uma tarefa não for definida, a mesma pode ter seu processamento iniciado em qualquer tempo.
Datas de Entrega ( $d_j$ )	Prazo de entrega final de uma tarefa. As datas de entrega das tarefas podem ser classificadas em: comuns; distintas; e com tolerância ou janelas de tempo.
Precedência ( $prec$ )	Indicam que uma ou mais tarefas devem ser concluídas antes que outra tarefa tenha seu processamento iniciado. Existem diversas formas de restrições de precedência, como segue: se cada tarefa tem no máximo um predecessor e um sucessor, as restrições são referidas como cadeias ( $prec\ chain$ ); se cada tarefa tem no máximo um sucessor ( $prec\ in\ tree$ ); e se cada tarefa tem no máximo um predecessor ( $prec\ out\ tree$ ).
Interrupções ( $prmp$ )	Implica que não é necessário manter uma tarefa na máquina, uma vez iniciada sua execução, até a sua conclusão. É permitido ao programador interromper o processamento de uma tarefa a qualquer momento e processar uma tarefa diferente na máquina. O processamento já realizado da tarefa na máquina não é perdido.
Bloqueio de Máquinas ( $block$ )	Uma tarefa após a conclusão de seu processamento numa máquina bloqueia a máquina se a máquina seguinte estiver ocupada processando outra tarefa.
No-Wait ( $nwt$ )	Situação em que não é permitido que tarefas esperem entre duas máquinas sucessivas. Isso implica que o tempo de início de uma tarefa no primeiro estágio tem que ser atrasado para garantir que a tarefa possa passar pelo sistema sem ter que esperar em qualquer máquina.
Quebras de Máquinas ( $brkdown$ )	Implica que uma máquina pode não estar disponível continuamente. Pode ocorrer quando as máquinas estão em operação, durante a realização do <i>setup</i> da máquina para uma tarefa ou quando uma tarefa está sendo removida.
Elegibilidade de Máquinas ( $M_j$ )	Este caso aparece em ambientes de máquinas paralelas e <i>flow shop</i> híbrido e ocorre quando uma tarefa deve ser processada em uma máquina ou tipo de máquina específico.
Presença de <i>Setup</i>	Neste caso a influência dos <i>setups</i> não permite que estes sejam incorporados aos tempos de processamento. Os <i>setups</i> , nestes casos podem ser independentes ou dependentes da sequência e podem também ser realizados antecipadamente ou não.
Processamento em Lotes ( $batch$ )	Caso uma máquina seja capaz de processar um número de tarefas simultaneamente e as tarefas são agrupadas para processamento em lotes.

Quadro 2 – Características das Tarefas em Problemas de Programação

Fonte: Elaborado pelos autores com base em Pinedo (2008), Allahverdi; Cheng; Kovalyov (2008) e Baker; Scudder (1990)

## 2.4 Critérios de Otimização em Problemas de Programação da Produção

A programação da produção é sempre realizada buscando atingir um critério (objetivo), ou conjunto de critérios, geralmente relacionados com os objetivos de desempenho da produção. De acordo com Almeida (1995) os critérios de otimização adotados nos

problemas de programação permitem avaliar o grau de sucesso da programação realizada. Os critérios de otimização mais comumente adotados em problemas de programação são sumarizados no Quadro 3.

<b>Crítérios de Otimização</b>	<b>Descrição</b>
Data de término da tarefa j ( <i>Completion Time</i> - $C_j$ )	Corresponde ao período desde o início da programação na data zero até ao momento em que a tarefa j é finalizada.
Duração total da programação ( <i>Makespan</i> - $C_{max}$ )	Corresponde ao máximo ( $C_1, C_2, \dots, C_j$ ), que é equivalente ao tempo de conclusão da última tarefa a deixar o sistema.
Tempo de fluxo da tarefa j ( <i>Flow Time</i> - $F_j$ )	Corresponde ao tempo entre o momento que a tarefa está disponível para ser processada e o momento em que ela é completada, correspondendo, então, ao tempo que a tarefa j permanece no sistema.
Atraso da tarefa j ( <i>Lateness</i> - $L_j$ )	Corresponde ao desvio entre o tempo de conclusão da tarefa e sua data de entrega. O <i>lateness</i> de uma tarefa j é definido como: $L_j = C_j - d_j$ .
Adiantamento da tarefa Jj ( <i>Earliness</i> - $E_j$ )	Corresponde ao desvio entre o tempo de conclusão da tarefa e sua data de entrega. O <i>earliness</i> de uma tarefa j é definido como: $E_j = \max(d_j - C_j, 0)$ .
Atraso da tarefa j ( <i>Tardiness</i> - $T_j$ )	Corresponde ao atraso, na execução da tarefa, em relação à sua data de entrega. O <i>tardiness</i> de uma tarefa j é definido como: $T_j = \max(C_j - d_j, 0) = \max(L_j, 0)$ .

Quadro 3 – Critérios de Otimização Comumente Adotados em Problemas de Programação da Produção.

Fonte: Elaborado pelos autores com base em French (1982), Bedworth; Bailey (1987), MacCarthy; Liu (1993), Morton; Pentico (1993) e Pinedo (2008)

De acordo com Allahverdi *et al.* (1999) e Allahverdi; Cheng; Kovalyov (2008) outros critérios de otimização utilizados para resolver problemas de programação são: Tempo Médio de Conclusão ( $\Sigma C_j/n$ ); Tempo Total de Conclusão ( $\Sigma C_j$ ); Tempo Total de Conclusão Ponderado ( $\Sigma w_j C_j$ ); Tempo Total de Fluxo ( $\Sigma F_j$ ); Tempo Médio de Fluxo ( $\Sigma F_j/n$ ); Tempo Total de Fluxo Ponderado ( $\Sigma w_j F_j$ ); *Lateness* Máximo ( $L_{max}$ ); *Lateness* Total ( $\Sigma L_j$ ); Total *Lateness* Total Ponderado ( $\Sigma w_j L_j$ ); *Tardiness* Máximo ( $T_{max}$ ); *Tardiness* Total ( $\Sigma T_j$ ); *Tardiness* Médio ( $\Sigma T_j/n$ ); *Tardiness* Total Ponderado ( $\Sigma w_j T_j$ ); *Earliness* Máximo ( $E_{max}$ ); *Earliness* Total ( $\Sigma E_j$ ); *Earliness* Médio ( $\Sigma E_j/n$ ); *Earliness* Total Ponderado ( $\Sigma w_j E_j$ ); Tempo de Espera da Tarefa ( $W_j$ ); Tempo Total de Espera ( $\Sigma W_j$ ), Tempo Médio de Espera ( $\Sigma W_j/n$ ); Tempo Total de Espera Ponderado ( $\Sigma w_j W_j$ ); Tempo Médio de Espera ( $\Sigma W_j/n$ ); Número de Tarefas em Atraso ( $\Sigma U_j$ ); e Número Médio de Tarefas em Atraso ( $\Sigma U_j/n$ ).

Além dos critérios de desempenho anteriormente mencionados, outros critérios tem sido reportados na literatura. Dentre esses, incluem-se: Custo Total de *Setup* (TSC); Tempo Total de *Setup* (TST ou  $\Sigma S_j$ ); Custos de Estoque (EC); Custos de Transporte (TC); Custo Total de Oportunidade (TCO); Custo Total de Utilidade (TCU); Tempo Total Ocioso (MIT); Estoque em Processo (WIP); e Tempo de Bloqueio das Máquinas (MBT). Em função da diversidade de critérios de desempenho utilizados em problemas de programação, verificados na literatura especializada, o conjunto de critérios pode ser expandido.

### 3. Metodologia

A pesquisa aqui relatada classifica-se, quanto aos fins, como descritiva e explicativa, e quanto aos meios, como bibliográfica. O método de abordagem adotado é o qualitativo, uma

vez que visou identificar e descrever os principais métodos de solução para o problema de programação da produção, disponíveis na literatura especializada.

O referencial teórico foi elaborado tendo como base livros, artigos científicos, teses e dissertações das áreas de Engenharia de Produção e Pesquisa Operacional. Para a identificação dos principais métodos de solução para o problema de programação da produção foram investigadas as mesmas bases utilizadas para a construção do referencial teórico.

Os principais métodos de solução identificados na literatura pesquisada e aplicáveis aos problemas de programação da produção foram agrupados em dois grupos ou categorias, conforme ilustrado na Figura 1.

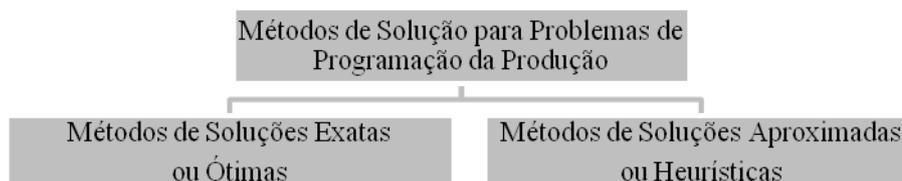


Figura 1 – Categoria dos Métodos de Solução para Problemas de Programação da Produção  
Fonte: Elaborado pelos autores

Posteriormente, os métodos enquadrados na categoria de soluções aproximadas ou heurísticas foram agrupados em três grupos ou categorias, conforme ilustrado na Figura 2.

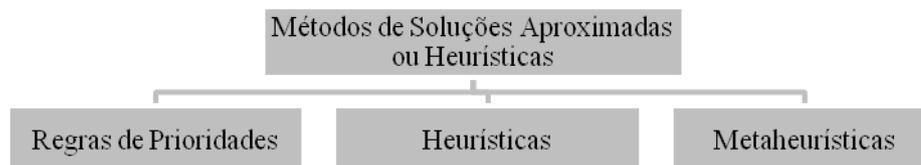


Figura 2 – Categoria dos Métodos de Soluções Aproximadas ou Heurísticas  
Fonte: Elaborado pelos autores

Após a categorização dos métodos de solução identificados, as características básicas de cada método, extraídas da literatura especializada por meio do método de análise de conteúdo foram sumarizadas.

## 4. Métodos de Solução para Problemas de Programação da Produção

Soluções para os problemas de programação da produção nos diferentes tipos de ambientes de máquinas, independentemente das características das tarefas incorporadas aos problemas e dos critérios de otimização adotados, podem ser obtidas por meio da aplicação de Métodos de Soluções Exatas ou Ótimas ou Métodos de Soluções Aproximadas ou Heurísticas, discutidos nas subseções seguintes.

### 4.1 Métodos de Soluções Exatas

Métodos de Soluções Exatas são métodos que garantem a obtenção de uma solução ótima de acordo com o critério de desempenho adotado (função objetivo). Apesar dessa garantia, em problemas com certo grau de complexidade (em particular, a maioria dos problemas combinatórios de interesse), a determinação de uma solução ótima em tempo computacional aceitável para a aplicação em questão pode ser difícil ou até mesmo impossível (MORABITO; PUREZA, 2010).

Dentre os vários Métodos Exatos disponíveis na literatura para resolução de problemas

de programação da produção, Nemhauser; Wolsey (1988), Arenales *et al.* (2007) e Stefanello (2011) destacam: *Branch-and-Bound*; Planos de Cortes; *Branch-and-Cut*; Geração de Colunas (*Dantzig-Wolf*); Relaxação Lagrangeana; Método de Benders; e Programação Dinâmica.

Algoritmos *Branch-and-Bound* (B&B) realizam enumerações sistemática das soluções candidatas, de tal maneira que soluções infrutíferas são eliminadas com base em limites superiores e inferiores da variável a ser otimizada. Neste tipo de método, subproblemas consistindo de relaxações lineares do problema inteiro são gerados e organizados em estrutura de árvore. A análise dos limitantes permite que ramos da árvore sejam descartados sem que as possíveis soluções factíveis que representam sejam explicitamente obtidas (QUADT, 2004 e MORABITO; PUREZA, 2010).

Algoritmos de Planos de Cortes “buscam obter uma aproximação da envoltória convexa da região factível de um problema de programação inteira que contenha um ponto extremo correspondente a uma solução ótima. Essa aproximação é obtida por meio de cortes ou desigualdades válidas” (ARENALES *et al.*, 2007, p.251).

O *Branch-and-Cut* (B&C) segue a mesma lógica do B&B, com a adição de vários cortes em cada um dos nós cuja relaxação é factível. Segundo Arenales *et al.* (2007) o método B&C “combina estratégias dos métodos B&B e planos de cortes, com o objetivo de reduzir o número de nós na árvore B&B” (Arenales *et al.*, 2007, p.256).

O Método de Geração de Colunas ou Método de *Dantzig-Wolf* segundo Arenales *et al.* (2007) “é uma aplicação do princípio de decomposição, que envolve a resolução de diversos subproblemas de tamanho menor, tipicamente estruturados, em lugar da resolução do problema original, cujo tamanho e complexidade impedem que seja resolvido em tempo razoável” (Arenales *et al.*, 2007, p.266).

A Relaxação Lagrangeana resolve problemas de programação inteira relaxando algumas restrições do problema e impondo uma penalidade proporcional as quantidades pelas quais as restrições são violadas (MORTON; PENTICO, 1993).

O Método de Benders de acordo com Arenales *et al.* (2007) “utiliza o conceito de projeção no espaço de variáveis complicadoras para então obter um problema, em geral, com uma estrutura mais fácil de resolução” (Arenales *et al.*, 2007, p.265).

A Programação Dinâmica segundo Arenales *et al.* (2007) “consiste na decomposição do problema original em uma sequência de subproblemas menores e mais simples de serem resolvidos” (Arenales *et al.*, 2007, p.375).

## 4.2 Métodos de Soluções Aproximadas

Métodos de Soluções Aproximadas, comumente denominados Métodos Heurísticos, produzem soluções com base geralmente em informações e a intuição do analista acerca do problema e de sua estrutura. Em geral, requerem apenas uma fração do esforço computacional de um método exato e quando bem projetados geram soluções de alta qualidade, porém sem a garantia da otimalidade (MORABITO; PUREZA, 2010). Os Métodos de Soluções Aproximadas disponíveis na literatura pesquisada podem ser categorizados em: Regras de Prioridades; Heurísticas; e Metaheurísticas.

## 4.2.1 Regras de Prioridade.

Métodos de solução baseados em Regras de Prioridade constituem os Métodos Aproximados ou Heurísticas, considerados os métodos mais elementares para a solução de problemas de programação da produção. Métodos baseados em Regras de Prioridade podem ser utilizadas para a obtenção de boas soluções ou para a geração de soluções iniciais. Segundo Pereira (2011) Regras de Prioridade são largamente utilizados em operações dentro dos ambientes de produção.

Uma Regra de Prioridade ou regra de despacho é uma regra que estabelece a prioridade com que as tarefas serão processadas em uma máquina (CARVALHO, 2000). De acordo com Davis; Aquilano; Chase (2001) as Regras de Prioridade são utilizadas para se obter uma sequência para o processamento das tarefas. As Regras de Prioridade fornecem orientações para o sequenciamento das tarefas nas máquinas, seguindo algumas informações sobre as tarefas, tais como o tempo de processamento, a data de entrega ou a ordem de chegada.

As Regras de Prioridade são geralmente classificadas em: i) Local – as prioridades são definidas com base apenas nas tarefas que estão esperando naquele centro de trabalho específico; e ii) Global – as prioridades são definidas de acordo com fatores como a carga de trabalho programada nas demais estações de trabalho pelas quais a tarefa deve ser processada (REID; SANDERS, 2005). As Regras de Prioridade também podem ser classificadas em: i) estáticas – são regras que alteram as prioridades quando ocorrem mudanças no sistema produtivo; e ii) dinâmicas – são regras que acompanham as mudanças e alteram as prioridades, em função das mudanças verificadas (TUBINO, 2007).

A literatura (CONWAY; MAXWELL; MILLER, 1967, ALMEIDA, 1995, PINEDO, 2008 e FUCHIGAMI, 2010) aponta um grande número de regras que podem ser aplicadas a problemas de programação, tais como: SPT (Menor Tempo de Processamento – *Shortest Processing Time*); LPT (Maior Tempo de Processamento – *Longest Processing Time*); EDD (Menor Data de Liberação – *Earliest Due Date*); ERD (Menor Data de Liberação - *Earliest Release Date*); MWKR (Maior Trabalho Remanescente – *Most Work Remaining*); LWKR (Menor Trabalho Remanescente – *Least Work Remaining*); MOPNR (Maior Número de Operações Remanescente – *Most Operations Remaining*); LOPNR (Menor Número de Operações Remanescente – *Least Operations Remaining*); MS (Menor Folga - *Minimum Slack*); SST (Menor Tempo de Setup - *Shortest Setup Time*); e CR (Razão Crítica – *Critical Ratio*).

É comum encontrarmos na literatura, variações das regras SPT e LPT que incorporam prioridades ou penalidades por atrasos ou antecipações, como por exemplo: WSPT (Menor Tempo de Processamento Ponderado – *Weighted Shortest Processing Time*); e WLPT (Maior Tempo de Processamento Ponderado – *Weighted Longest Processing Time*) (SILVER; PYKE; PERTERSON, 1998, PINEDO, 2008, FUCHIGAMI, 2010).

Algumas Regras de Prioridade mais elaboradas são basicamente uma combinação das regras elementares. Exemplos de regras mais elaboradas, denominadas regras de prioridade compostas, são: ATC (Custo Aparente de Atraso – *Apparent Tardiness Cost*); ATCS (Custo Aparente de Atraso com *Setup* – *Apparent Tardiness Cost with Setup*); e COVERT (Custo ao Longo do Tempo - *Cost Over Time*).

De acordo com Almeida (1995) a escolha de uma ou outra regra, depende de alguns

fatores relacionados com o problema de programação em questão, tais como o tamanho do sistema, o número de tarefas a ser processada e o critério de otimização adotado.

## 4.2.2 Heurísticas

As Heurísticas conseguem boas soluções empregando um esforço computacional relativamente baixo, mas sem a garantia da otimalidade (VOB, 2001). De acordo com Queiroz (2011) Heurísticas são procedimentos que procuram solucionar problemas de forma racional, explorando a estrutura do problema, com o objetivo de encontrar uma solução boa, e quando possível, a ótima.

Dentre os vários Métodos Heurísticos disponíveis para resolução de problemas combinatórios, Silver (2004) e Rodriguez (2006) destacam: métodos construtivos; métodos de decomposição ou partição; métodos indutivos ou de manipulação do modelo; métodos de redução do espaço de busca; e métodos de busca em vizinhança.

Métodos Construtivos são aqueles que gradualmente adicionam componentes individuais à solução até que se obtenha uma solução factível. Tipicamente nenhuma solução é obtida até que o procedimento seja concluído (SILVER, 2004).

Métodos de Decomposição ou partição são métodos que dividem o problema em subproblemas menores, presumivelmente mais simples de resolverem, sendo a saída de um subproblema a entrada do subproblema seguinte. A resolução de todos os subproblemas fornece uma solução para o problema global (SILVER, 2004).

Métodos Indutivos ou de Manipulação do Modelo são métodos que consideram versões menores ou mais simples de um problema, modificando a estrutura do modelo com a finalidade de torná-lo mais fácil de resolver, e deduzindo, a partir de sua resolução, a solução do problema original (SILVER, 2004).

Métodos de Redução do Espaço de Busca tem como a ideia básica a redução do espaço de busca, por meio da eliminação de soluções. Este tipo de método visa identificar alguma característica que presumivelmente deva aparecer na solução ótima para que desse modo seja possível a simplificação do problema. Uma forma de redução consiste em considerar somente soluções que satisfazem uma propriedade específica do problema (SILVER, 2004).

Métodos de Busca em Vizinhança, também conhecidos como métodos de Busca Local ou Heurísticas de Refinamento, são métodos que partem de uma solução inicial factível, gerada por uma heurística construtiva ou randomicamente. Soluções factíveis na vizinhança da solução corrente e que melhoram seu valor são então selecionadas de forma iterativa até que se satisfaça algum critério de parada (SILVER, 2004 e RIBEIRO, 2009).

De acordo com Silver (2004) estes tipos de heurísticas não são necessariamente mutuamente exclusivas, quando utilizadas na solução de problemas combinatórios.

## 4.2.3 Metaheurísticas

Metaheurísticas são procedimentos heurísticos sofisticados que coordenam estratégias de busca locais de nível mais elevado, criando um processo para superar ótimos locais, e efetuando uma busca da solução mais robusta para um problema (GLOVER; KOCHENBERGER, 2003). Dentre as metaheurísticas mais populares, destacam-se *Simulated Annealing*, *Tabu Search*, Algoritmos Genéticos, *Variable Neighborhood Search*, *Greedy*

*Randomized Adaptive Search Procedure, Guided Local Search e Iterated Local Search.*

O *Simulated Annealing* (SA) é um método desenvolvido pela analogia entre o processo de resfriamento dos sólidos e o processo de otimização de um sistema (SILVER, 2004), onde os estados do sistema correspondem às soluções do problema, a energia dos estados corresponde à função objetivo, o estado fundamental corresponde à solução ótima, os estados meta-estáveis correspondem aos ótimos locais e a temperatura corresponde a um parâmetro de controle.

A metaheurística *Tabu Search* (TS) sua forma mais simples é uma busca em vizinhança dotada de uma estrutura que armazena atributos das soluções geradas por um dado número de iterações. O termo tabu vem do fato de que soluções com estes atributos ativos são proibidas ou penalizadas pelo período especificado (MORTON; PENTICO, 1993).

Algoritmos Genéticos (*Genetic Algorithms* - GA) são métodos de otimização global, baseados nos mecanismos de seleção natural e genética. Em algoritmos genéticos a busca da solução se dá em um processo iterativo onde são realizadas operações de seleção, cruzamento e mutação. Os algoritmos genéticos trabalham sobre uma população de soluções, gerando uma nova população a cada iteração (LACERDA; CARVALHO; LUDERMIR, 2002).

O *Variable Neighborhood Search* (VNS) é um método que aplica explicitamente uma estratégia baseada na mudança dinâmica de estrutura de vizinhança (BLUM; ROLI, 2003). Segundo Silver (2004) o método VNS considera várias estruturas de vizinhança, em que dada uma solução corrente, uma solução é selecionada aleatoriamente da vizinhança. Um procedimento de melhoria local é aplicado resultando em um ótimo local.

O *Greedy Randomized Adaptive Search Procedure* (GRASP) é um método iterativo, composto de duas fases: construção probabilística de uma solução inicial e a melhoria da solução. A geração da solução inicial é feita por uma heurística construtiva gulosa com aleatoriedade controlada na escolha dos componentes da solução, ou seja, restrita aos componentes mais bem avaliados segundo a função objetivo. A solução inicial é então usada como ponto de partida para uma busca local tradicional ou metaheurística (ARROYO, 2002).

O *Guided Local Search* (GLS) é um método que aplica explicitamente uma estratégia baseada em estruturas de vizinhança dinâmica (BLUM; ROLI, 2003). A ideia básica é adicionar um conjunto de funções ponderadas à função objetivo original, em que características de soluções para o problema original são incorporadas nas funções ponderadas (SILVER, 2004).

*Iterated Local Search* (ILS) é um tipo de método que aplica um procedimento de busca local em uma solução inicial até a obtenção de um ótimo local. Essa solução é perturbada, obtendo-se nova solução. O procedimento de busca local é então aplicado a esta nova solução. O procedimento é encerrado de acordo com o critério de parada pré-estabelecido (BLUM; ROLI, 2003).

## 5. Considerações Finais

Este artigo apresentou os principais métodos de soluções exatas e de soluções aproximadas disponíveis na literatura especializada. Foram identificados sete tipos de métodos de soluções exatas e treze tipos métodos de soluções aproximadas. Os métodos de soluções aproximadas foram categorizados em regras de prioridades, heurísticas e metaheurísticas. Ressalta-se que os métodos de soluções, quer sejam métodos exatos, quer

sejam métodos aproximados, não se esgotam no referencial teórico-conceitual aqui apresentado.

Para os métodos aqui apresentados, sugere-se para trabalhos futuros a elaboração de descrições detalhadas que incorporem os procedimentos para construção dos métodos de solução aqui apresentados, as análises referentes aos méritos em termos de qualidade das soluções fornecidas e de desempenho computacional, bem como as principais variações e/ou adaptações desenvolvidas para os métodos.

Sugere-se também para trabalhos futuros que o referencial aqui explicitado seja expandido para incorporar outras categorias e métodos de solução, como por exemplo, a categoria de métodos bioinspirados baseados em populações, que incorporam os métodos evolutivos e os métodos de enxames.

## Referências

- ABEPRO. *Áreas e Subáreas de Engenharia de Produção*. 2008. Disponível em: <<http://www.abepro.org.br/interna.asp?p=399&m=424&s=1&c=362>>.
- ALLAHVERDI, A.; CHENG, T. C. E.; KOVALYOV, M. Y. A survey of scheduling problems with setup times or costs, *European Journal of Operational Research*, 187, 2008, p. 985–1032.
- ALLAHVERDI, A.; GUPTA, J.N.D.; ALDOWAISAN, T. A review of scheduling research involving setup considerations, *Omega - The International Journal of Management Science*, 27, 1999, p. 219-239.
- ALMEIDA, A. M. N. Escalonamento dinâmico de tarefas industriais sujeitas a prazos de entrega. Dissertação (Mestrado). Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, 1995.
- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa Operacional*. Rio de Janeiro: Elsevier, 2007.
- ARROYO, J. E. C. Heurística e Metaheurística para Otimização Combinatória Multi-Objetivo. Tese (Doutorado). Programa de Pós-Graduação em Engenharia Elétrica 2002. Universidade Estadual de Campinas. Campinas, 2002.
- BAKER, K. R.; SCUDDER, G. D. Sequencing with earliness and tardiness penalties: a review. *Operations Research*, 38, 1, 1990, p. 22-36.
- BEDWORTH, D. D.; BAILEY, J. E. *Integrated Production Control Systems: management, analysis, design*. 2 edição. New York: John Wiley & Sons, Inc., 1987.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys*, 35, 1, 2003, p.268-308.
- CARVALHO, J. D. A. *Programação da Produção. Apontamentos*. Departamento de Engenharia de Produção e Sistemas. Escola de Engenharia da Universidade do Minho. Minho, 2000. Disponível em [http://pessoais.dps.uminho.pt/jdac/apontamentos/Cap03\\_Program.pdf](http://pessoais.dps.uminho.pt/jdac/apontamentos/Cap03_Program.pdf)
- CONWAY, R. W.; MAXWELL, W. L.; MILLER, L. W. *Theory of scheduling*. Massachusetts: Addison-Wesley Publishing Company, 1967.
- DAVIS. M. M.; AQUILANO, N. J.; CHASE, R. B. *Fundamentos de Administração da Produção*. 3. ed. São Paulo: Bookman. Porto Alegre, 2001.
- FRENCH, S. *Sequencing and scheduling: an introduction to the mathematics of the job shop*. New York: Wiley, 1982
- FUCHIGAMI, H.Y. *Programação de operações em máquinas*. Apostila. Universidade Federal de Goiás, Catalão, 2010.
- GLOVER, F.; KOCHENBERGER, G. A. *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers, 2003.

GUPTA, J.N.D. A review of flowshop scheduling research. In: RITZMAN, L.P.; KRAJEWSKI, L.J.; BERRY, W.L.; GOODMAN, S.T.; HARDY, S.T.; VITT, L.D. Disaggregation Problems in Manufacturing and Service Organizations. MartinusNijhoff, The Hague, 1979, pp. 363–388.

GUPTA, J.N.D.; STAFFORD JR. Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169, 2006, p. 699-711.

HAX, A. C.; CANDEA, D. Production and inventory management. New Jersey: Prentice-Hall, 1984.

JOHNSON S. M.; MONTGOMERY D. C. Operations Research in Production, Planning, Scheduling and Inventory Control. New York: Wiley, 1974.

LACERDA, E. G. M.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. Um tutorial sobre algoritmos genéticos. *Revista de Informática Teórica e Aplicada*, 9, 3, 2002, 109-139.

MACCARTHY, B. L.; LIU, J.Y. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, 31, n.1, 1993, p. 59-79.

MORABITO, R.; PUREZA, V. Modelagem e Simulação. In: MIGUEL, Paulo Augusto Cauchick. Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações. Rio de Janeiro: Elsevier, 2010.

MORTON, T. E.; PENTICO, D. W. Heuristic Scheduling Systems. New York: Wiley, 1993.

NEMHAUSER, G.; WOLSEY, L. A. Integer and Combinatorial Optimization. New York: John Wiley and Sons, 1988.

PEREIRA, A. M. S. Metaheurísticas para o problema de flowshop flexível com penalidades de adiantamento e atraso. Dissertação (Mestrado). Departamento de Ciência da Computação, Universidade Federal de Viçosa. Viçosa, 2011.

PINEDO, M. Scheduling: theory, algorithms, and systems. 3a. edição. New Jersey: Prentice-Hall, 2008.

QUADT, D. Lot-sizing and scheduling for flexible flow lines. Lecture notes in economics and mathematical systems. Germany: Springer, 2004.

QUEIROZ, M. M. Métodos heurísticos aplicados ao problema de programação de frotas de navios PLVs. Dissertação (Mestrado). Departamento de Engenharia Naval e Oceânica. Escola Politécnica da Universidade de São Paulo. São Paulo, 2011.

REID, R. D.; SANDERS, N. R. Gestão de Operações. Rio de Janeiro: LTC, 2005.

RIBEIRO, F. F. Um algoritmo genético adaptativo para a resolução do problema de sequenciamento em uma máquina com penalização por antecipação e atraso da produção. Dissertação (Mestrado). Programa de Pós-Graduação em Modelagem Matemática e Computacional. Centro Federal de Educação Tecnológica de Minas Gerais. Belo Horizonte, 2009.

RODRIGUEZ, C. O. R. Um algoritmo Grasp con doble relajación para resolver el problema de flow shop scheduling. Monografia (Graduação). Facultad de Ciencias e Ingeniería. Pontificia Universidad Católica del Peru. Lima, 2006.

SILVER, E. A. An overview of heuristic solution methods. *Journal of the Operational Research Society*, 55, 2004, p. 936-956.

SILVER, E. A.; PYKE, D. F.; PETERSON, R. Inventory, Management and Production Planning and Scheduling, 3a. edition. USA: John Wiley & Sons, Inc., 1998.

STEFANELLO, F. Hibridização de métodos exatos e heurísticos para resolução de problemas de otimização combinatória. Dissertação (mestrado). Centro de Tecnologia, Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria. Porto Alegre, 2011.

TUBINO, D. F. Planejamento e Controle da Produção: Teoria e Prática. São Paulo: Atlas, 2007.

VOB, S. Metaheuristics: the state of the art. In: NAREYEK, A. Local search for planning and scheduling. Berlin: LNAI, 2001.