PRODUÇÃO ONLINE
REVISTA CIENTÍFICA ELETRÔNICA DE ENGENHARIA DE PRODUÇÃO
ISSN 1676-1901

# DESENVOLVIMENTO E ANÁLISE DE REGRAS DE SEQUENCIAMENTO DA PRODUÇÃO PARA MINIMIZAÇÃO DE MEDIDAS DE ATRASO EM *FLOW SHOP* COM TEMPOS DE *SETUP*

# DEVELOPMENT AND ANALYSIS OF PRODUCTION SEQUENCING RULES TO MINIMIZE TARDINESS MEASURES IN FLOW SHOP WITH SETUP TIMES

Tatiane Carvalho Silva* E-mail: tatiane_economiaufg@hotmail.com
Hélio Yochihiro Fuchigami** E-mail: heliofuchigami@ufg.br
*Universidade Federal de Goiás (UFG), Catalão, GO
**Universidade Federal de Goiás (UFG), Aparecida de Goiânia, GO

**Resumo:** As regras de sequenciamento são métodos de programação da produção que possuem como vantagens, dentre outras, sua fácil implementação e simples compreensão. Neste sentido, este trabalho tem como objetivo o desenvolvimento de novas regras de sequenciamento para a minimização de medidas de atraso em um ambiente *flow shop* com *setup* independente da sequência. Foram analisadas, como medidas de desempenho, o atraso total da programação, o atraso máximo e o número de tarefas atrasadas. O estudo considera um *flow shop* com *m* máquinas e *n* tarefas, e, portanto, traz resultados para diferentes ambientes (relacionados ao porte do problema). A pesquisa traz resultados também em relação aos diferentes cenários que se pode observar na produção, de acordo com um alto ou baixo fator de atraso e com a amplitude da faixa de datas de entrega. Os resultados foram gerados através da experimentação computacional. Dentre as novas regras de prioridade propostas, destacam-se os resultados obtidos pela regra SPT3, constituindo uma nova alternativa de sequenciamento da produção, capaz de apresentar melhor desempenho que a regra clássica EDD.

**Palavras-chave**: Programação da produção. *Flow shop*. Medidas de atraso. Regras de prioridade. *Setup* independente da sequência.

**Abstract:** Priority rules are production scheduling methods with many advantages, among others, the easy implementation and simple understanding. Therefore, this study aims to develop new priority rules for minimizing tardiness measures in a flow shop environment with sequence-independent setup times. The total tardiness of the schedule, the maximum tardiness and the percent of tardy jobs were analyzed as measures of performance. The study considers a flow shop with *m* machines and *n* jobs, and thus presents results for different options of problem-size. The research also evidence results for the different scenarios that can be seen in production, according to a high or low tardiness factor and the amplitude of the due dates. The results were generated by computational experiments. Among the new priority rules proposed, SPT3 rule constitutes a new alternative sequencing of production, able to present greater success than the traditional EDD.

**Keywords**: Scheduling. Flow shop. Tardiness. Priority rules. Sequence-independent setup times.

## 1 INTRODUÇÃO

Em um ambiente produtivo, o sequenciamento da produção é uma das importantes decisões que devem ser tomadas. Por isso, estudos que abordam o sequenciamento têm sido realizados ao longo do tempo, buscando as melhores

maneiras de programá-la. A programação da produção aborda a questão da alocação de recursos para as tarefas a serem realizadas em determinados períodos de tempo. Alocar adequadamente os recursos é aspecto fundamental na busca por um melhor desempenho produtivo.

Por conseguinte, este estudo é voltado à análise do sequenciamento da produção, com o objetivo de desenvolver e avaliar os efeitos das regras de sequenciamento no ambiente *flow shop* com *m* máquinas e *n* tarefas, tempos de *setup* independentes da sequência e restrição de data de entrega, observadas as medidas de desempenho relacionadas aos atrasos.

Para tanto, o método a ser aplicado constitui no desenvolvimento de regras de sequenciamento, que ordenem as tarefas de forma crescente de acordo as expressões de prioridade, as quais são desenvolvidas em função das restrições do problema. Para viabilizar esta análise, os dados são gerados através de programação computacional, e, aplicadas as regras desenvolvidas, será possível analisar os efeitos das decisões de sequenciamento propostas, bem como compará-los.

As regras aplicadas serão avaliadas em relação às medidas de desempenho de atraso. Serão analisados: o atraso total da programação, ou seja, a soma dos atrasos das tarefas que ultrapassaram o prazo de entrega; o número de tarefas atrasadas; e o atraso máximo, ou seja, o maior valor de atraso observado. Com a programação computacional aplicada ao problema proposto, será possível analisar os efeitos das regras de prioridade utilizadas sobre cada uma das medidas de desempenho.

O artigo está organizado na seguinte estrutura: a seção 2 apresenta uma revisão da literatura, com estudos que tratam sobre as regras de sequenciamento e ambiente *flow shop*; a seção 3 faz uma descrição do problema bem como das medidas de desempenho que serão analisadas e apresenta as regras de prioridade propostas e seus fundamentos; a seção 4 trata da experimentação computacional desenvolvida; os resultados da aplicação das regras constam na seção 5 e a seção 6 apresenta as conclusões deste trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

A programação da produção é um processo de tomada de decisão, usado

tanto na indústria como nas empresas de serviços. Esse processo lida com a alocação de recursos para as tarefas a serem realizadas em determinados períodos de tempo, de modo a otimizar o alcance dos objetivos da organização. Desta maneira, a programação, como um processo de tomada de decisão, desempenha um importante papel nas empresas (PINEDO, 2016).

O desenvolvimento de um ambiente detalhado de tarefas ajuda a manter a eficiência e o controle das operações. O sequenciamento da produção é uma das importantes decisões que devem ser tomadas dentro de um ambiente produtivo. Em virtude disso, estudos que abordam o sequenciamento têm sido realizados ao longo do tempo, buscando as melhores maneiras de programar a produção.

As regras de prioridade, também chamadas regras de sequenciamento, regras de ordenação ou regras de despacho, são regras de decisão lógica que selecionam a próxima tarefa a ser executada em um ambiente de produção por meio de algum critério. As regras de prioridade possuem vantagens na sua aplicação, sendo de simples compreensão e fácil implementação na esfera computacional. Além disso, servem de base para o desenvolvimento de métodos de solução mais complexos (FUCHIGAMI, 2016).

Jayamohan e Rajendran (2000) também salientam as vantagens das regras de sequenciamento, ao afirmar que tais regras são mais populares em muitos sistemas reais de manufatura do que as heurísticas, especialmente por serem de simples implementação e uso no ambiente produtivo.

Desde o final da década de 1950, tais regras vêm sendo intensamente investigadas (HAUPT, 1989). Muitas pesquisas sobre sequenciamento se voltam para a análise do desempenho dessas regras, suscitando algumas conclusões gerais sobre regras de prioridade simples. Comprovou-se, por exemplo, que a regra SPT – *Shortest Processing Time* - minimiza o tempo médio de fluxo e o número de tarefas atrasadas. Por outro lado, a regra EDD – *Earliest Due Date* - minimiza o atraso máximo das tarefas finalizadas (BARMAN, 1998).

Os trabalhos sobre regras de ordenação não se limitam às análises de regras consagradas. A busca por critérios de prioridade alternativos para o sequenciamento de tarefas em um processo produtivo tem sido objetivo de estudos de sequenciamento, revelando assim o interesse dos pesquisadores em ir além das regras de prioridades já conhecidas e consolidadas.

Barman (1998) realizou um estudo investigando o desempenho de regras criadas a partir da combinação de três regras de prioridade simples e já conhecidas. O problema abordado pelo pesquisador retratou um ambiente *flow shop* com três máquinas e sem tempos de *setup* explícitos, e as novas ordenações foram analisadas pelo seu desempenho com relação às medidas de atraso. Os resultados da pesquisa indicam que a combinação das regras é uma estratégia de ordenação melhor que a utilização de tais regras na sua forma pura quando várias medidas de desempenho são avaliadas em conjunto. Fuchigami e Moccellin (2015, 2016) analisaram os efeitos de regras de prioridade em problemas de *flow shop* híbridos com tempos de *setup* explícitos.

Alguns aspectos devem ser observados na pesquisa de sequenciamento. Um dos fatores a serem notados refere-se à chegada das tarefas no sistema. Com essa variável, os ambientes podem ser classificados em modelos estáticos ou dinâmicos. No modelo estático, as tarefas chegam ao sistema simultaneamente e estão liberadas no instante 0, ou seja, no início da programação. No modelo dinâmico, por outro lado, as tarefas chegam em diferentes momentos no sistema, sendo então integradas no processo de sequenciamento em curso (HAUPT, 1989).

Esta pesquisa propõe novas regras de sequenciamento e analisa seus desempenhos em um ambiente estático, e, portanto, a data de liberação das tarefas, $r_i$, são todas iguais a zero. As tarefas, já liberadas, são processadas em um ambiente de produção denominado *flow shop*, caracterizado por um fluxo linear de processamento, no qual as tarefas passam por várias máquinas, sendo cada máquina um estágio de produção distinto (FUCHIGAMI, 2016).

O problema de programação *flow shop* tem sido intensamente estudado na literatura. Contudo, verifica-se que a maioria das pesquisas realizadas considera os tempos de preparação das máquinas – *setup* – não significativos, ou são incluídos nos tempos de processamento das tarefas. Entretanto, estes artifícios afetam a qualidade da programação quando tais tempos tem uma variabilidade relevante em função da ordenação das tarefas nas máquinas, ou quando as máquinas podem ser preparadas antecipadamente para a execução das tarefas (MOCCELLIN; NAGANO, 2007).

Essencialmente, existem dois tipos de *setup* que podem ser tratados separadamente dos tempos de processamento das tarefas: o *setup* independente e

o *setup* dependente da sequência. O primeiro depende apenas da operação a ser realizada, independentemente da sequência das tarefas. Por outro lado, o segundo tipo depende tanto da operação a ser executada quanto da operação que foi processada imediatamente antes na mesma máquina e, portanto, é um *setup* dependente da ordenação das tarefas (MOCCELLIN; NAGANO, 2011).

Estudos sobre as regras de prioridade em ambiente *flow shop* com restrição de *setup* foram desenvolvidos por Fernandes *et al.* (2010). O trabalho analisa um problema *flow shop* com três máquinas, restrição de data de liberação e *setup* dependente da sequência. Buscando a minimização do *makespan*, ou seja, da duração total da programação, foram desenvolvidas novas regras de prioridade, baseadas nas regras SPT – *Shortest Processing Time* e LPT – *Longest Processing Time*. Os autores realizaram as análises por meio dos resultados obtidos em experimentação computacional.

Portanto, esta pesquisa baseia-se, dentre outros, nos estudos de Barman (1998), que analisa o desempenho de combinações de regras com medidas de atraso; na pesquisa de Fernandes *et al.* (2010), que trabalham com novas regras e restrição de *setup*; bem como nas vantagens do uso das regras de prioridade, como a sua simplicidade e a facilidade na implementação das regras computacionalmente. Desta maneira, as regras de prioridades constituem soluções aplicáveis ao problema proposto neste trabalho, descrito na seção a seguir.

## 3 NOTAÇÕES DOS PROBLEMAS E REGRAS DE PRIORIDADE PROPOSTAS

Este artigo tem como objetivo desenvolver novas regras de prioridade buscando soluções alternativas para as decisões de sequenciamento. O estudo está voltado ao sequenciamento em ambiente *flow shop* com *n* tarefas e *m* máquinas, com restrições de prazo de entrega e *setup* independente da sequência. O *setup* é antecipado, o que significa que a preparação de uma máquina para receber a tarefa pode iniciar enquanto ela ainda está sendo processada na máquina anterior.

Os indicadores de desempenho estão relacionados às medidas de atraso, sendo estas o atraso total, o atraso máximo e o número de tarefas atrasadas. Desta maneira, esta pesquisa considera três diferentes problemas que podem ser denotados das seguintes formas:

- $\boldsymbol{Fm|d_j,s_{jk}|T_{max}}$ – *flow shop* com *m* máquinas, restrição de *setup* indepen-

dente e data de entrega, buscando minimizar o atraso máximo, ou seja, o maior valor de atraso da sequência;

- $\boldsymbol{Fm|d_j,s_{jk}|T}$ – *flow shop* com *m* máquinas, restrições de prazo e *setup* independente, buscando minimizar o atraso total, ou seja, a soma dos atrasos de todas as tarefas;
- $\boldsymbol{Fm|d_j,s_{jk}|n_T}$ – *flow shop* com as mesmas restrições, buscando minimizar o número de tarefas atrasadas.

Para sequenciar os problemas propostos, regras de prioridade foram desenvolvidas baseando-se na lógica de conhecidas regras: EDD (*Earliest Due Date*), MST (*Minimum Slack Time*), CR (*Critical Ratio*) e SPT (*Shortest Processing Time*). Os critérios das regras propostas utilizam os prazos e os tempos de processamento das tarefas, e, portanto, consideram ordenação crescente. As decisões de sequenciamento são apresentadas da seguinte forma:

**- EDD**: ordena as tarefas pelo prazo de entrega, priorizando o processamento daquelas com menor prazo. É uma regra clássica e será utilizada para fins de comparação com os demais métodos desenvolvidos.

- **SPT1**: sequencia as tarefas pela soma do *setup* da tarefa na primeira máquina com os seus tempos de processamento em todas as máquinas. Este método considera o *setup* na primeira máquina uma vez que essa preparação inicial influenciará o início da programação.

- **SPT2**: ordena as tarefas pela soma dos tempos de *setup* e de processamento da tarefa em todas as máquinas. Analisa a carga total de trabalho das tarefas.

- **SPT3**: realiza a ordenação pela soma dos tempos de processamento e de *setup* da tarefa em todas as máquinas e o seu prazo de entrega. Este método soma a carga total de trabalho das ordens ao seu prazo, relacionando diretamente essas variáveis. Assim, quanto menor o prazo da tarefa, menor será seu valor de SPT3, e esta será alocada no início da sequência, reduzindo a possibilidade de atrasos. Seguindo este raciocínio, se $d_j$ e $p_j$ forem altos, a regra será alocada mais ao final. Caso uma dessas variáveis assuma um valor elevado, enquanto outra possui valor baixo, a soma de ambas poderá contrabalancear o resultado.

- **SPT4**: ordena as tarefas pela soma do tempo de *setup* e de processamento da tarefa na primeira máquina. A regra preza pela análise da influência da carga

inicial de trabalho da tarefa na programação, afetando o início do processamento das demais ordens.

**- CR**: ordena as tarefas pelo quociente entre seu prazo e tempo de processamento total. O fundamento desta regra clássica é a ordenação pela relação entre tempo disponível e tempo de processamento. Como não há restrição de data de liberação, o tempo disponível é representado apenas pelo prazo.

- **CR1**: ordena pelo quociente em que o numerador é formado pela diferença entre o prazo da tarefa e o seu tempo de *setup* na primeira máquina, que define o início efetivo do processamento. O denominador corresponde à soma dos tempos de processamento da tarefa em todas as etapas.

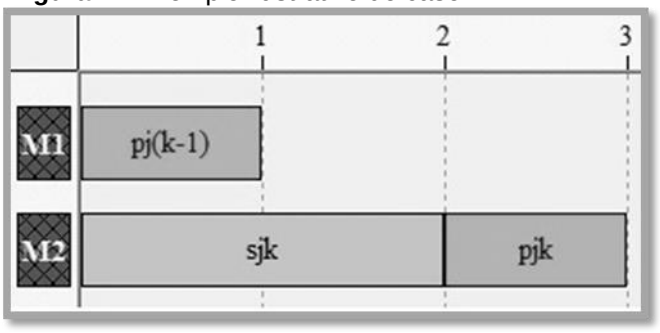
**- CR2**: realiza a ordenação pelo resultado do quociente em que o numerador é o prazo da tarefa, e o denominador é formado pela soma dos tempos de processamento e os de *setup* da tarefa em todas as etapas. Assim, considera-se a relação do prazo não apenas com os tempos de processamento, mas também com os tempos de preparação.

**- CR3**: ordena pelo quociente em que o numerador é o prazo da tarefa. O denominador é a soma entre: tempo de *setup* da tarefa na primeira máquina; a soma dos tempos de processamento da tarefa em todas as máquinas; e a soma do valor resultante da diferença entre o *setup* da tarefa na máquina *k* (considerando a partir da segunda máquina) menos o tempo de processamento da tarefa na máquina anterior, caso esta diferença seja positiva.

Para uma melhor compreensão dessa relação, a ilustração a seguir descreve o critério apresentado na regra CR3.

- Caso I: $max\{s_{jk} - p_{j(k-1)}, 0\} = s_{jk} - p_{j(k-1)}$

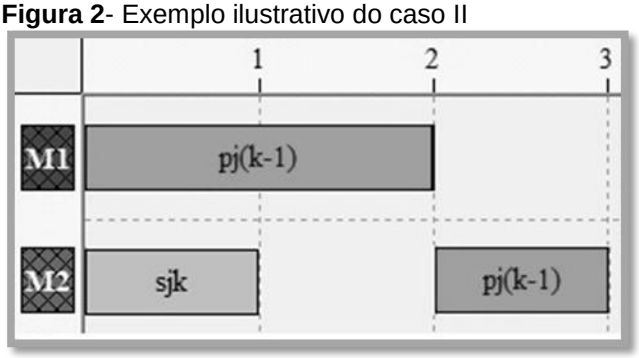
**Figura 1** - Exemplo ilustrativo do caso I



Neste caso, mesmo com a antecipação do *setup* da tarefa para a segunda máquina, o $s_{jk}$ é maior que o tempo de processamento na máquina anterior. Desta maneira, a diferença é maior que zero, indicando um tempo de espera da tarefa, e será considerada no cálculo do critério de ordenação da regra CR3.

- Caso II: $max\{s_{jk} - p_{j(k-1)}, 0\} = 0$

**Figura 2**- Exemplo ilustrativo do caso II




Nesta outra situação, o tempo de processamento da tarefa em determinada máquina é superior ao tempo de *setup* da tarefa para a próxima etapa. Portanto, este tempo de *setup*, quando antecipado, não afetará na programação. Logo, a diferença $s_{jk} - p_{j(k-1)}$ será negativa, e não será considerada no cálculo para a ordenação pela regra CR3.

**- MST1**: sequencia as ordens pelo resultado da diferença entre o prazo da tarefa e da soma do seu *setup* na primeira máquina, que afeta o efetivo início do processamento, e dos seus tempos de processamento em todas as máquinas.

- **MST2**: ordena de forma crescente a diferença entre o prazo de entrega da tarefa e a soma dos seus tempos de *setup* com os seus tempos de processamento em todas as máquinas. Desta forma, é possível observar a folga total que a tarefa possui, de forma isolada, com relação ao seu prazo de entrega.

**- MST3**: ordena pelo resultado da subtração em que o minuendo é o prazo da tarefa e o subtraendo é formado pela expressão apresentada no denominador da regra CR3. O fundamento de usar tal expressão é considerar o tempo de espera da tarefa entre uma etapa e outra, observando a folga em consideração aos tempos de *setup* antecipados.

A Tabela 1 resume as regras supracitadas e seus critérios de ordenação.

**Tabela 1** - Regras de sequenciamento e seus critérios de ordenação

| Regra | Critério de ordenação |
|---|---|
| EDD | $d_j$ |
| SPT1 | $s_{j1} + \sum_{k=1}^{m} p_{jk}$ |
| SPT2 | $\sum_{k=1}^{m} (s_{jk} + p_{jk})$ |
| SPT3 | $\sum_{k=1}^{m} (s_{jk} + p_{jk}) + d_j$ |
| SPT4 | $s_{j1} + p_{j1}$ |
| CR | $\frac{dj}{\sum_{k=1}^{m} p_{jk}}$ |
| CR1 | $\frac{d_j - s_{j1}}{\sum_{k=1}^{m} p_{jk}}$ |
| CR2 | $\frac{d_j}{\sum_{k=1}^{m} (s_{jk} + p_{jk})}$ |
| CR3 | $\frac{d_j}{s_{j1} + p_{j1} + \sum_{k=2}^{m} (p_{jk} + max\{s_{jk} - p_{j(k-1)}, 0\})}$ |
| MST1 | $dj - \left( s_{j1} + \sum_{k=1}^{m} p_{jk} \right)$ |
| MST2 | $d_j - \sum_{k=1}^{m} (s_{jk} + p_{jk})$ |
| MST3 | $d_j - \left( s_{j1} + p_{j1} + \sum_{k=2}^{m} (p_{jk} + max\{s_{jk} - p_{j(k-1)}, 0\}) \right)$ |

## 4 METODOLOGIA E EXPERIMENTAÇÃO COMPUTACIONAL

Segundo as definições de Martins (2010, p.45-49) e Nakano (2010, p.64), esta pesquisa possui abordagem quantitativa, pois se volta para os aspectos de mensurabilidade, causalidade, generalização e replicação. Pode também ser classificada como experimento, uma vez que testa o relacionamento entre as variáveis da pesquisa operacionalizada, manipulando variáveis independentes (neste caso, a programação do problema) para se observar o resultado na variável dependente (representada aqui pelas medidas de atraso).

Além disso, de acordo com Jung (2004), este trabalho se classifica como: pesquisa aplicada quanto à natureza, por gerar conhecimento com finalidades de aplicação prática; pesquisa exploratória quanto aos objetivos, pois visa à melhoria teórico-prática de sistemas, processos e produtos; inovação, pela proposição de novos modelos, além de ser feita a partir de impulsos criativos, simulações e experimentações, podendo originar novos modelos destinados a invenções, inovações e a otimização; e pesquisa experimental quanto aos procedimentos para a obtenção de novos conhecimentos e produtos tecnológicos, requerendo uma manipulação de variáveis detalhada e sistemática, e originando inovações a partir de ensaios e estudos dinâmicos em laboratório.

Conforme a prática comum na área de programação da produção, para a realização da experimentação computacional foram definidos oito diferentes valores para o número de tarefas, cinco para as máquinas e intervalos uniformemente distribuídos para os tempos de processamento e de *setup*. Em relação às datas de entrega, optou-se por seguir o método utilizado por Armentano e Ronconi (2000) e Ronconi e Birgin (2012), que utiliza a distribuição uniforme no intervalo [$P(1–T–R/2)$, $P(1–T+R/2)$], onde $T$ e $R$ são dois parâmetros denominados fator de atraso das tarefas e faixa de dispersão das datas de entrega, respectivamente, e $P$ é um limitante inferior para o *makespan*, definido mais adiante. Todos os valores utilizados são apresentados na Tabela 2.

**Tabela 2** - Parâmetros da experimentação computacional

| **Parâmetros** | **Símbolo** | **Valores** |
|---|---|---|
| Número de tarefas | $n$ | 5, 10, 15, 20, 30, 50, 80, 100 |
| Número de máquinas | $m$ | 2, 3, 5, 10, 20 |
| Tempos de processamentos | $p_{jk}$ | $U$[1,99] |
| Tempos de *setup* | $s_{jk}$ | $U$[1,49] |
| Datas de entrega (*due dates*) | $d_j$ | $U$[P(1–$T$–R/2),P(1–$T$+R/2)] |
| Fator de atraso | $T$ | 0,2; 0,4 |
| Faixa de dispersão | $R$ | 0,6; 1,2 |

A partir da variação de $T$ e $R$, foram obtidos os seguintes cenários:

- Cenário 1: Baixo fator de atraso ($T$=0,2) e pequena faixa de datas de entrega ($R$=0,6);

- Cenário 2: Baixo fator de atraso ($T$=0,2) e ampla faixa de datas de entrega ($R$=1,2);
- Cenário 3: Alto fator de atraso ($T$=0,4) e pequena faixa de datas de entrega ($R$=0,6);
- Cenário 4: Alto fator de atraso ($T$=0,4) e ampla faixa de datas de entrega ($R$=1,2).

O limitante inferior (*lower bound*) para o *makespan P*, utilizado neste trabalho, é uma adaptação daquele proposto por Taillard (1993), considerando o *setup* explícito.

O limitante de Taillard (1993) baseia-se em outros dois: o primeiro é o *machine-based bound*, ou seja, limitante baseado em máquinas, e considerado a existência de uma máquina gargalo no sistema; e o segundo é o *job-based bound* ou o limitante baseado em tarefas, que considera o pior caso, isto é, a tarefa com o maior tempo total de processamento em todas as máquinas.

Assim, considerando simultaneamente as duas situações numa mesma expressão, tem-se o limitante inferior geral para o *makespan* em problema com *m* máquinas:

$$LB = \max\left\{\max_{1\le k\le m}\left\{\sum_{j=1}^{n} p_{jk} + \min_{j}\sum_{q=1}^{k-1} p_{jq} + \min_{j}\sum_{q=k+1}^{m} p_{jq}\right\}, \max_{j}\sum_{k=1}^{m} p_{jk}\right\} \qquad (4.1)$$

Porém, esse *lower bound* não leva em consideração os tempos de *setup*, que é uma das restrições deste estudo. Desta forma, foram necessárias algumas adequações, de modo que o limitante superior P usado aqui seja igual ao $LB_{Final}$.

$$LB1 = \sum_{j=1}^{n}\left(p_{j1} + s_{j1}\right) + \min_{j}\sum_{k=2}^{m} p_{jk} \qquad (4.2)$$

$$LB2 = \max_{2\le k\le m}\left[\sum_{j=1}^{n}\left(p_{jk} + s_{jk}\right) + \min_{j}\sum_{q=1}^{k-1}\left(p_{jq} + s_{j1}\right) + \min_{j}\sum_{q=k+1}^{m}\left(p_{jq}\right) - \max_{j}\left(s_{jk}\right)\right] \qquad (4.3)$$

$$LB3 = \max_{j}\left[\sum_{k=1}^{m}\left(p_{jk} + s_{j1}\right)\right] \qquad (4.4)$$

$$LB_{Final} = \max\left\{LB1,\ LB2,\ LB3\right\} \qquad (4.5)$$

O primeiro termo considera que a carga da primeira máquina (tempos de processamento e de *setup*) somada ao tempo de processamento da menor tarefa

nas máquinas seguintes. O segundo é a adaptação da consideração de existência de uma máquina gargalo. E o terceiro considera a tarefa com a maior carga de trabalho (processamento e *setup*). O maior valor dentre os três definirá o limitante inferior.

Com estes parâmetros, obteve-se 160 classes de problemas: 8 opções para $n$ * 5 opções para $m$ * 1 intervalo de *setup* * 2 opções de $T$ * 2 opções de $R$ = 160. Para cada classe foram gerados aleatoriamente 100 problemas, totalizando assim 16 mil problemas analisados.

Os resultados do desempenho das regras propostas foram avaliados estatisticamente por meio da ANOVA (Análise de Variância) e em relação ao percentual de sucesso que as regras obtiveram em cada medida de desempenho considerada: soma dos atrasos, atraso máximo e percentual de tarefas atrasadas. O sucesso de uma regra é o número de vezes que ela forneceu o melhor resultado para os problemas resolvidos, empatando ou não. O percentual de sucesso também foi utilizado na análise detalhada separadamente por parâmetros do problema: o número de tarefas, o número de máquinas e o cenário.

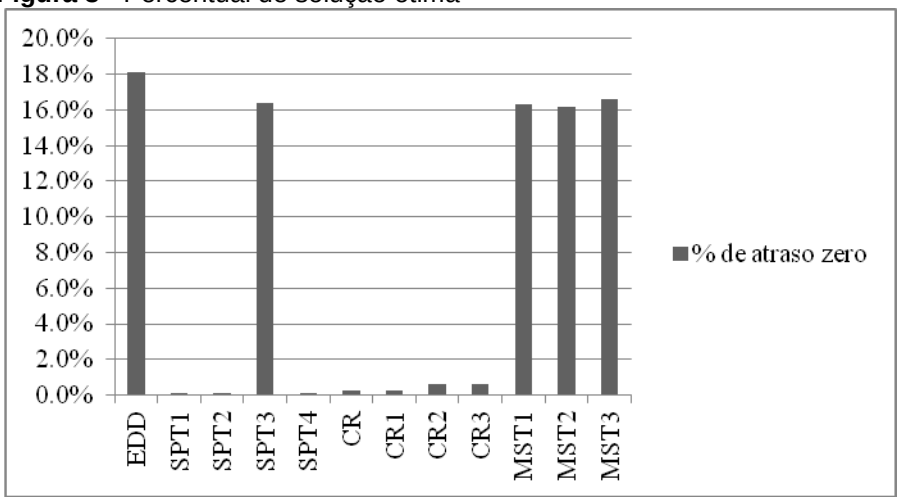
## 5 RESULTADOS E DISCUSSÕES

Os resultados são apresentados em relação ao percentual de sucesso que as regras de sequenciamento obtiveram para cada medida de desempenho. Os tópicos a seguir apresentam esses resultados e análises.

### 5.1 Análise geral do percentual de problemas com atraso zero

Com os resultados obtidos na experimentação computacional, pode-se inicialmente observar o índice de sucesso das regras de prioridade no fornecimento de soluções com atraso total zero. Neste caso, é possível afirmar que a regra forneceu a solução ótima para o problema. A Figura 3 apresenta, dentre todos os problemas gerados para cada regra, o percentual de resultados cujo atraso foi igual a zero.

**Figura 3** - Percentual de solução ótima




Percebe-se que a regra EDD foi a que obteve melhor desempenho nesse aspecto. A EDD é uma regra clássica que fornece a solução ótima em um ambiente de máquina única, com relação à minimização do atraso máximo. O resultado apresentado na figura acima confirma que essa regra foi eficiente também no ambiente *flow shop*, comparando-se com as demais regras. Entretanto, a SPT3 e as regras MST apresentaram resultados aproximados, fornecendo solução com atraso zero para pouco mais de 16% de todos os problemas gerados. Essa análise indica que tais regras podem apresentar bons resultados com relação às medidas de atraso. Os tópicos a seguir analisam cada problema separadamente.

## 5.2 Resultados para o problema $Fm|d_j,s_{jk}|T$

O problema $Fm|d_j,s_{jk}|T$ considera como medida de desempenho a soma dos atrasos das tarefas. A Tabela 3 apresenta os resultados do teste ANOVA com nível de significância 0,05.

**Tabela 3** – Resultados da tabela ANOVA para o fator soma dos atrasos (α=0,05)

| ***Fonte da variação*** | ***SQ*** | ***GL*** | ***MQ*** | ***F*** | ***valor-P*** | ***F crítico*** |
|---|---|---|---|---|---|---|
| Entre grupos | $8{,}97 \cdot 10^{12}$ | 10 | $8{,}16 \cdot 10^{11}$ | 1265,46 | 0,00 | 1,79 |
| Dentro dos grupos | $1{,}24 \cdot 10^{14}$ | 191988 | $6{,}45 \cdot 10^{8}$ | | | |
| Total | $1{,}33 \cdot 10^{14}$ | 191999 | | | | |

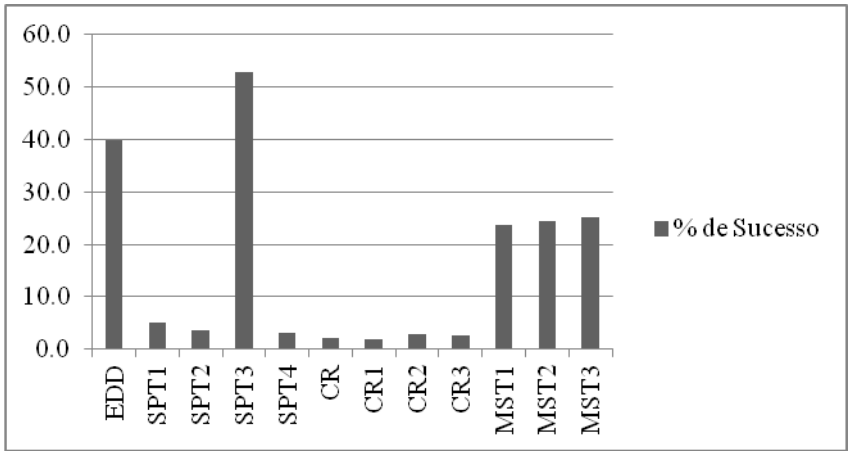
Para o problema da soma dos atrasos das tarefas, a estatística teste *F* foi 1265,46, que é bem maior que o valor *F* crítico (igual a 1,79) para o nível de

significância de 0,05. Daqui, infere-se que a hipótese nula restrita ao fator desta medida de desempenho é rejeitada, ou seja, existe diferença significa entre os resultados das regras para a soma dos atrasos. Esta informação é confirmada pelo valor *P*, que ficou igual a zero.

A Figura 4 representa graficamente o percentual de sucesso dos métodos de sequenciamento propostos neste estudo. A regra SPT3 obteve 52,9% de sucesso, sendo o melhor método para a redução da soma dos atrasos. O gráfico demonstra que essa regra obteve resultado melhor que a EDD, regra clássica aplicada quando o objetivo é reduzir os atrasos da programação. Portanto, a primeira conclusão deste estudo é a que a regra SPT3 obteve mais sucesso na redução do atraso total, podendo ser uma nova alternativa de programação.

A regra SPT3 ordena as tarefas pela soma de seus tempos de *setup*, processamento e seu prazo. Diferentemente do que se poderia imaginar, a ideia da soma consistiu numa boa alternativa, ficando a variável $d_j$ com sinal positivo na equação. Desta maneira, as variáveis tomam uma relação diretamente proporcional, em que, havendo diminuição do prazo e do processamento, diminui-se a medida SPT3, e vice-versa.

**Figura 4** - Sucesso (%) dos métodos propostos em relação à soma dos atrasos das tarefas




Analisando o desempenho das regras de forma mais detalhada, de acordo com os parâmetros *n*, *m* e cenários, é possível concluir que, com relação ao número de tarefas, a regra SPT3 continua sendo a que apresenta maior percentual de sucesso, seguida da clássica EDD, como pode ser visto na Figura 5. Percebe-se que, para essas regras, quanto maior o número de tarefas, maior o percentual de

sucesso da regra. As regras MST1, MST2 e MST3 também apresentaram essa tendência crescente.

As demais regras SPT (SPT1, SPT2 e SPT4) e as regras CR (CR, CR1, CR2 e CR3) apresentaram uma tendência de desempenho decrescente, ou seja, na medida em que aumentou o número de tarefas, menores foram os percentuais de sucesso. De forma geral, independente do número de máquinas, essas regras obtiveram baixos índices de sucesso, próximos a zero ou, em alguns casos, percentual de sucesso igual a 0.

Observando-se a Figura 6, que apresenta o sucesso em função do número de máquinas, percebe-se que, semelhante à análise do número de tarefas, a regra SPT3 supera o percentual de sucesso da regra EDD. De modo semelhante à análise anterior, as demais regras SPT e as regras CR apresentaram baixos níveis de sucesso. A diferença neste novo foco de análise é que, na medida em que aumenta o número de máquinas no problema, os percentuais de sucesso para as melhores regras declinam, enquanto que para as regras com níveis de sucesso mais baixo, o aumento do número de máquinas fez aumentar o percentual de sucesso, embora ainda permaneçam bem abaixo das regras SPT3, EDD e as regras baseadas na MST.

**Figura 5** - Sucesso (%) dos métodos propostos em relação ao número de tarefas da programação

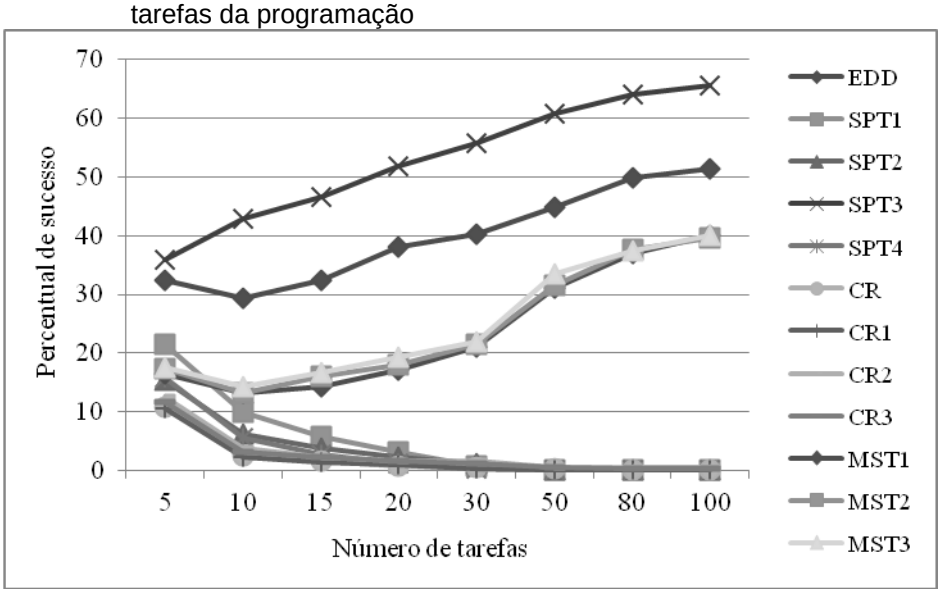
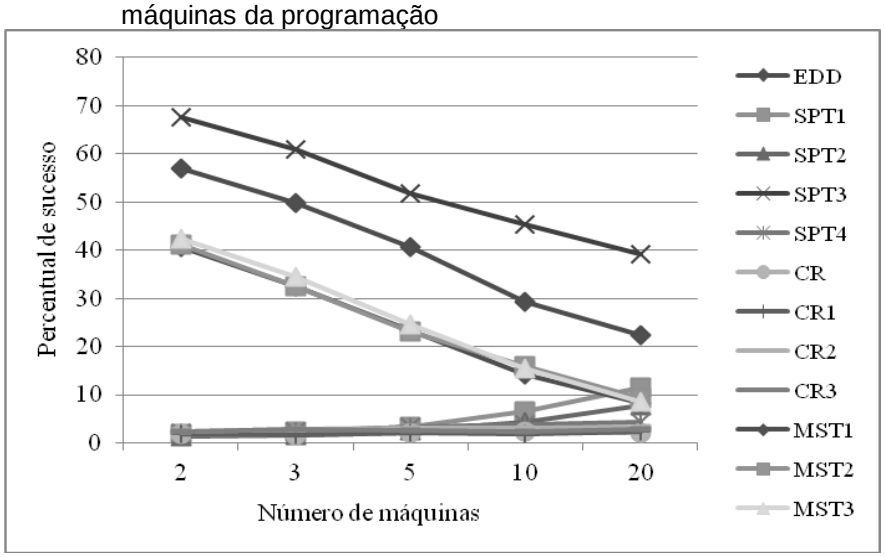
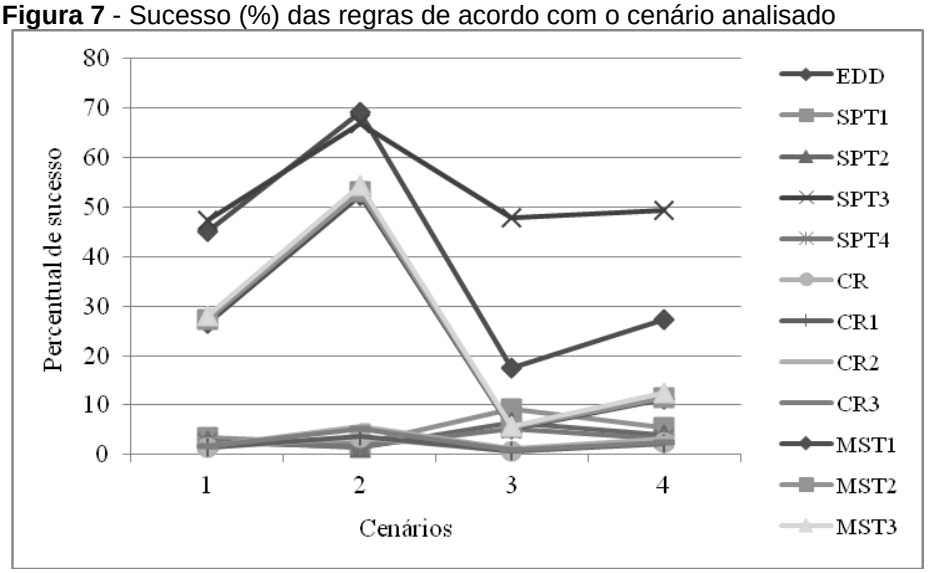


**Figura 6** - Sucesso (%) das regras de acordo com o número de máquinas da programação




Por fim, para o problema que busca minimizar a soma dos atrasos, conclui-se que a regra SPT3 também apresenta os mais altos índices de sucesso observando os quatro cenários separadamente, conforme a Figura 7. Com exceção do cenário 2, em que a regra EDD apresentou 69,3% de sucesso e a SPT3, 66,93%, em todos os outros cenários a regra SPT3 apresentou melhores resultados. Nos cenários 1 e 2, essas duas regras apresentaram índices muito próximos. Nos cenários 3 e 4, entretanto, a regra EDD obteve um desempenho bem mais baixo. Observa-se novamente que as regras com maiores percentuais de sucesso foram as SPT3, EDD e as MST. As demais, como nas análises anteriores, não foram igualmente satisfatórias, obtendo, em todos os cenários, percentuais de sucesso não superiores a 10%.

**Figura 7** - Sucesso (%) das regras de acordo com o cenário analisado



## 5.3 Resultados para o problema $Fm|d_j,s_{jk}|T_{max}$

Analisando-se agora os resultados das regras para a medida de atraso máximo, a Tabela 4 apresenta os resultados obtidos pelo teste ANOVA com nível de significância de 0,05.

**Tabela 4** – Resultados da tabela ANOVA para o fator atraso máximo (α=0,05)

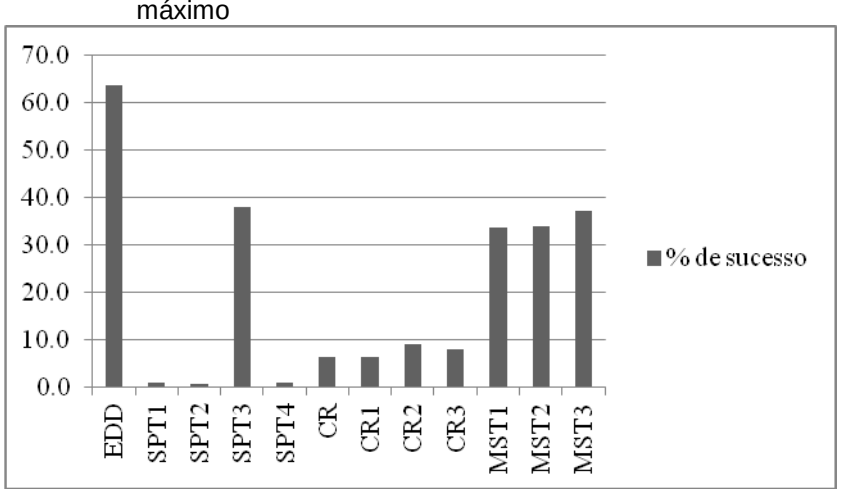
| *Fonte da variação* | *SQ* | *GL* | *MQ* | *F* | *valor-P* | *F crítico* |
|---|---|---|---|---|---|---|
| Entre grupos | $1{,}05 \cdot 10^{11}$ | 11 | $9{,}53 \cdot 10^{9}$ | 5870,20 | 0,00 | 1,78 |
| Dentro dos grupos | $3{,}12 \cdot 10^{11}$ | 191988 | $1{,}62 \cdot 10^{6}$ | | | |
| Total | $4{,}17 \cdot 10^{11}$ | 191999 | | | | |

No caso do atraso máximo, a estatística teste *F* foi 5870,20, que é bem maior que o valor *F* crítico (igual a 1,78) para o nível de significância de 0,05. Daqui, infere-se que a hipótese nula restrita ao fator desta medida de desempenho é rejeitada, ou seja, existe diferença significa entre os resultados das regras para o atraso máximo das tarefas. Esta informação é confirmada pelo valor *P*, que ficou igual a zero.

Em relação à porcentagem de sucesso, percebe-se que a EDD, regra clássica de ordenação pelo menor tempo de processamento, assumiu o posto de melhor solução, como mostra a Figura 8. Dentre todos os problemas analisados, com diferentes parâmetros e cenários, tal regra obteve 63,7% de sucesso, enquanto a SPT3, que ocupa o segundo lugar neste ranking, obteve 38% de sucesso. Conclui-se, portanto, que, de modo geral, a regra EDD é a melhor solução quando o problema com restrições de prazo e *setup* busca minimizar o atraso máximo da programação.

É importante salientar que já era conhecido que a regra EDD fornece a solução ótima para o problema de minimização do atraso máximo em máquina única. Porém, para o problema abordado neste trabalho, não foi encontrada uma análise comparativa tal como a realizada, o que representa a contribuição desta pesquisa.

**Figura 8** - Sucesso (%) dos métodos propostos em relação ao atraso máximo




Quanto às análises sobre o percentual de sucesso em função do número de tarefas, máquinas e cenário, este problema apresentou resultados semelhantes ao problema anterior, com tendências similares, conforme as Figuras 9, 10 e 11. A maior diferença é que, para a medida do atraso máximo, como já salientado, a regra EDD foi a melhor solução, superando a regra SPT3. Em todos os tipos de análise para este problema, a regra SPT3 obteve resultados semelhantes às regras MST.

**Figura 9 -** Sucesso (%) dos métodos propostos em função do número de tarefas

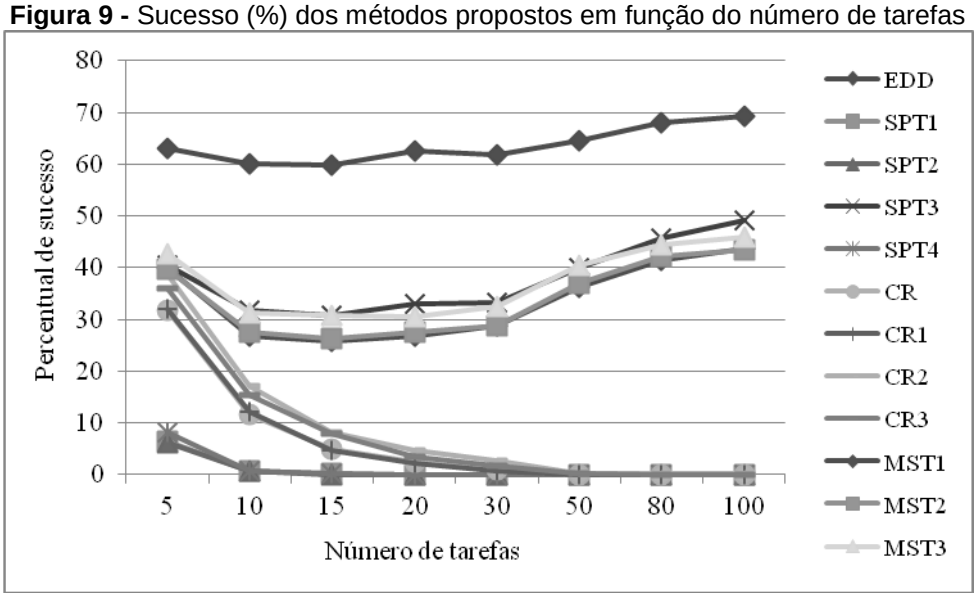


**Figura 10** - Sucesso (%) dos métodos propostos em função do número de máquinas

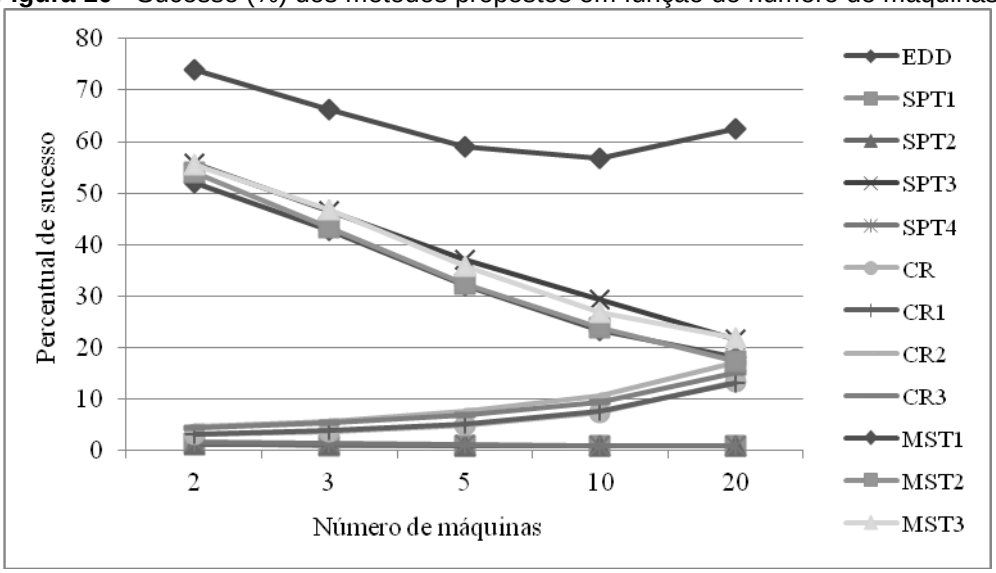
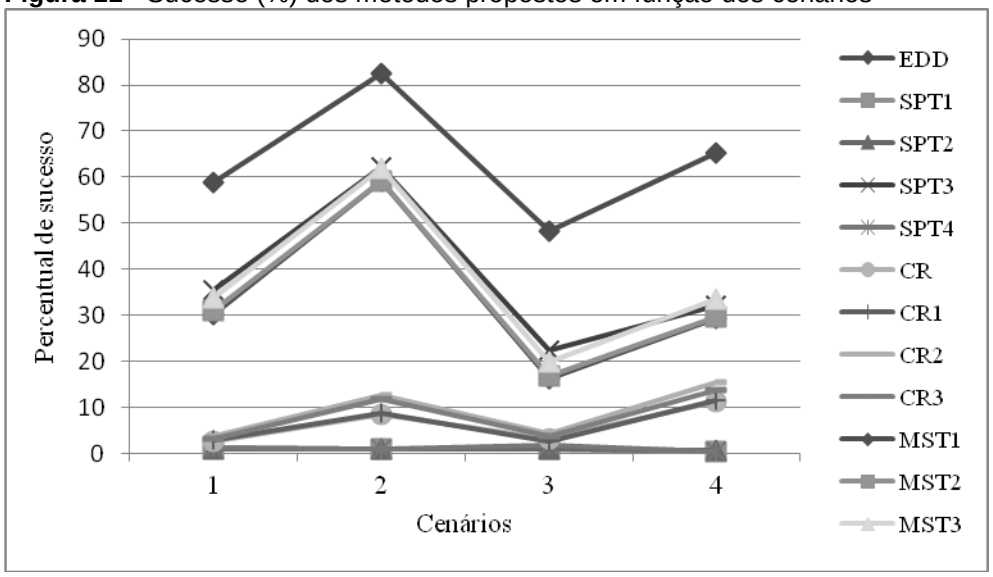



**Figura 11** - Sucesso (%) dos métodos propostos em função dos cenários




## 5.4 Resultados para o problema $Fm|d_j,s_{jk}|n_T$

Com relação ao número de tarefas atrasadas, a Tabela 5 apresenta os resultados do teste ANOVA com nível de significância de 0,05.

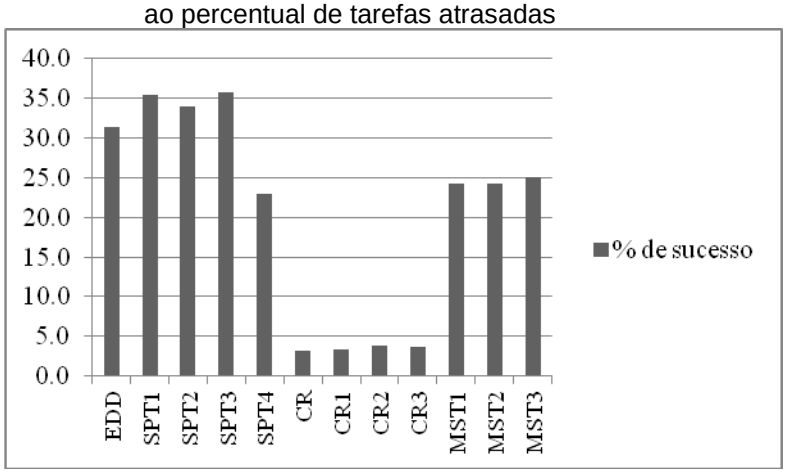
**Tabela 5** – Resultados da tabela ANOVA para o fator número de tarefas atrasadas (α=0,05)

| *Fonte da variação* | *SQ* | *GL* | *MQ* | *F* | *valor-P* | *F crítico* |
|---|---|---|---|---|---|---|
| Entre grupos | $3{,}80 \cdot 10^6$ | 11 | 345209.1 | 361,16 | 0,00 | 1,79 |
| Dentro dos grupos | $1{,}84 \cdot 10^8$ | 191988 | 955.8277 | | | |
| Total | $1{,}87 \cdot 10^8$ | 191999 | | | | |

Neste problema do número de tarefas atrasadas, a estatística teste *F* foi 361,16, que é bem maior que o valor *F* crítico (igual a 1,79) para o nível de significância de 0,05. Então, infere-se que a hipótese nula restrita ao fator desta medida de desempenho é rejeitada, ou seja, existe diferença significa entre os resultados das regras para número de tarefas atrasadas. Esta informação é confirmada pelo valor *P*, que ficou igual a zero.

Na análise do percentual de sucesso, é possível perceber no gráfico da Figura 12 que, com exceção das regras baseadas na clássica CR, as demais apresentaram desempenhos relativamente próximos. O maior percentual de sucesso foi apresentado pela regra SPT3, seguida pela regra SPT1, com respectivamente 35,8% e 35,5% de sucesso.

**Figura 12** - Sucesso (%) dos métodos propostos em relação ao percentual de tarefas atrasadas



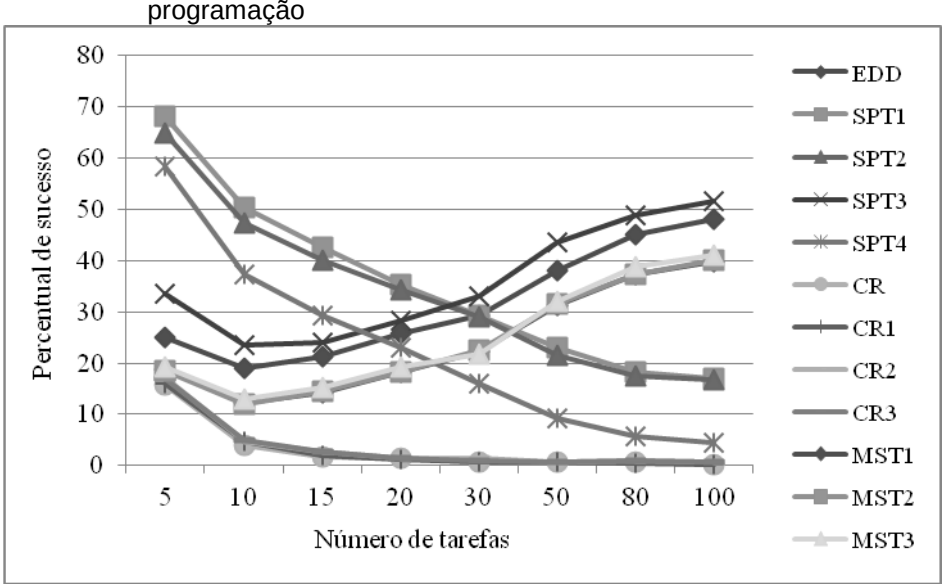

O gráfico demonstra que as regras SPT, de forma geral, apresentaram os melhores resultados em comparação com o conjunto de regras, ficando com percentual de sucesso entre 30% e 35%. A exceção é a regra SPT4, que ficou mais próxima das regras MST, as quais apresentaram resultados próximos a 25%. As regras CR foram as piores, e nem chegaram a atingir o nível de 5% de sucesso.

Entretanto, percebe-se que em comparação com os demais problemas estudados, quando se busca a minimização do percentual de tarefas atrasadas, os níveis de sucesso foram menores. Enquanto a melhor solução para o problema que trata da soma dos atrasos obteve 52,9% de sucesso e a melhor solução para o atraso máximo atingiu 63,7%, neste caso das tarefas atrasadas a melhor solução

obteve percentual de sucesso de 35,8%, abaixo das soluções para os dois primeiros problemas.

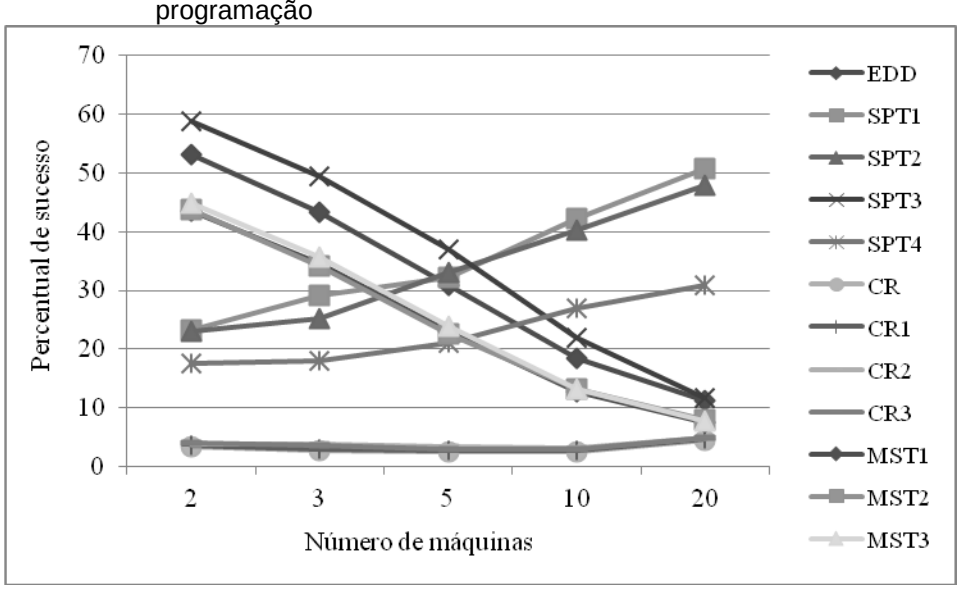
Da mesma forma que os problemas anteriores, existe um grupo de regras que melhoram o seu desempenho com o aumento do porte do problema (SPT3, EDD e as regras MST) e outro cujo desempenho piora (as demais). Por outro lado, ocorre uma inversão na superioridade dos métodos (cruzamento das linhas) com problemas com 30 tarefas. Em problemas com até 20 tarefas, a ordem de melhor desempenho foi: SPT1, SPT2, SPT4 e SPT3. Já em problemas acima de 30 tarefas, a superioridade foi: SPT3, EDD e as regras MST. A Figura 13 representa essa inversão.

**Figura 13** - Sucesso (%) das regras de acordo com o número de tarefas da programação



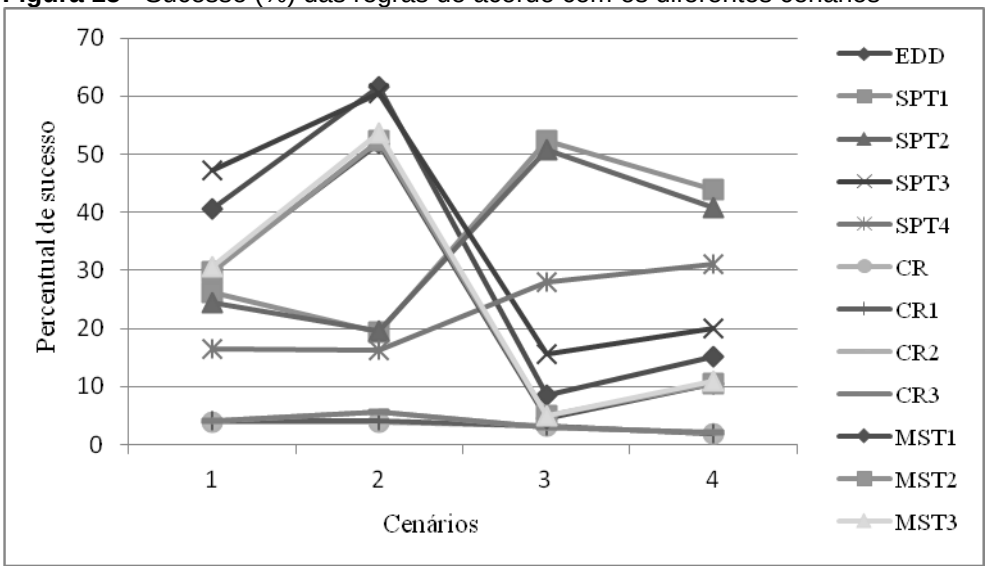

Inversões semelhantes no desempenho ocorrem com a análise do número de máquinas e dos cenários, como representado nas Figuras 14 e 15. Com relação ao número de máquinas, as regras MST, EDD e SPT3 têm maiores percentuais de sucesso quando há menos máquinas no problema. Na medida em que aumenta a quantidade de máquinas, o sucesso de tais regras declina e as demais regras SPT assumem os melhores resultados. A exceção são as regras CR, que estão sempre em níveis baixos de sucesso.

**Figura 14** - Sucesso (%) das regras de acordo com o número de máquinas da programação




Análise semelhante pode ser realizada quando se observa separadamente os cenários do problema. As regras EDD e SPT3 atingem os maiores percentuais de sucesso no cenário 2, e os menores no cenário 3; as regras MST seguem essa tendência dessas duas regras; e as demais regras SPT atingem níveis mais baixos de sucesso no cenário 2 e mais altos no cenário 3. As regras CR, como nas demais análises, permanecem com baixos níveis de sucesso em todos os cenários, não chegando sequer ao percentual de 6%.

**Figura 15** - Sucesso (%) das regras de acordo com os diferentes cenários



## 6 CONCLUSÕES

Conclui-se que a regra SPT3 apresentou os melhores resultados, estando sempre entre os métodos com maiores percentuais de sucesso, tanto para a análise global do problema, quanto para as análises em função de *n*, *m* e cenários. Seu desempenho foi superior à EDD para as análises do percentual de tarefas atrasadas e atraso total.

Portanto, a ordenação pela soma dos tempos de *setup*, processamento e prazo da tarefa consistiu numa boa alternativa, ficando a variável $d_j$ com sinal positivo na equação. Desta maneira, as variáveis tomam uma relação diretamente proporcional, em que, havendo diminuição do prazo e do processamento, diminui-se a medida SPT3, e vice-versa.

Com relação ao problema que busca minimizar o atraso máximo da programação, a regra EDD é a melhor solução. Salienta-se que já era conhecido que essa regra fornece a solução ótima para o problema de minimização do atraso máximo em máquina única. Porém, para o problema abordado neste trabalho, não foi encontrada uma análise comparativa tal como a realizada, o que representa uma contribuição desta pesquisa.

As regras SPT, de forma geral, comparando-se com os demais métodos, obtiveram altos percentuais de sucesso para a análise de tarefas atrasadas. Este resultado está de acordo com a ideia já consagrada de que a regra SPT minimiza o número de tarefas atrasadas.

As regras MST apresentaram resultados medianos, seguindo as tendências das regras EDD e SPT3, porém com desempenhos, de forma geral, inferiores. Tais regras, por outro lado, foram mais eficientes que o grupo das regras CR, que não apresentaram bons resultados.

Com relação aos cenários, de forma geral os métodos apresentaram seus mais altos níveis de sucesso no cenário 2, ou seja, na situação com baixo fator de atraso e ampla faixa de datas de entrega. Isso se explica uma vez que, se a faixa de dispersão das datas de entrega é maior, os prazos serão mais elevados, diminuindo a possibilidade de atrasos. Enquanto que o baixo fator de atraso das tarefas também constitui um aspecto favorável aos problemas analisados.

Por fim, salienta-se que a nova regra proposta, SPT3, se destacou e constitui uma nova alternativa de sequenciamento da produção, capaz de apresentar em certas situações sucesso maior que a regra clássica EDD. Para trabalhos futuros, sugere-se trabalhar com as regras CR novamente, ordenando as tarefas por ordem decrescente, alteração que, acredita-se, poderá revelar novas perspectivas de sucesso para tais regras.

**AGRADECIMENTOS**

A pesquisa relatada neste artigo teve o apoio do CNPq, da CAPES e da FAPEG.

## REFERÊNCIAS

ARMENTANO, V.A.; RONCONI, D.P. Minimização do tempo total de atraso no problema de *flow shop* com *buffer* zero através de busca tabu. **Gestão & Produção**, v. 7, n. 3, p.352-363, 2000. DOI: http://dx.doi.org/10.1590/S0104-530X2000000300011

BARMAN, S. The impact of priority rule combinations on lateness and tardiness. **IIE Transactions**, v. 30, n. 5, p. 495-504, 1998. DOI: http://doi.org/10.1023/A:1007503508080

FERNANDES, S.C.; PIRES, T.A.; CAMARGO, V.H.C.; FUCHIGAMI, H.Y. Análise de desempenho de regras de prioridade para programação em *flow shop* com três máquinas, datas de liberação e tempos de *setup* dependentes da sequência. In: Encontro Nacional de Engenharia de Produção, 30, 2010, São Carlos. **Anais**. São Carlos: ENEGEP, 2010.

FUCHIGAMI, H.Y. **Introdução ao Sequenciamento da Produção**. Catalão: UFG, 2016. Material didático. Versão 7.0.

FUCHIGAMI, H.Y.; MOCCELLIN, J.V. Desempenho relative de regras de prioridade para programação de *flow shop* híbrido com tempos de *setup*. **Produção Online**, v. 15, n. 4, p. 1174-1194, 2015.DOI: http://dx.doi.org/10.14488/1676-1901.v15i4.1791

FUCHIGAMI, H.Y.; MOCCELLIN, J.V. Efeitos de regras de prioridade para programação da produção em sistemas industriais complexos. **Produção Online**, v. 16, n. 1, p. 3-25, 2016. DOI: http://dx.doi.org/10.14488/1676-1901.v15i4.1791

HAUPT, R. A survey of priority rule-based scheduling. **OR Spektrum**, v. 11, p. 3-16, 1989. DOI: https://doi.org/10.1007/BF01721162

JAYAMOHAN, M.; RAJENDRAN, C. New dispatching rules for shop scheduling: a step forward. **International Journal of Production Research**, v. 38, n. 3, p. 563-586, 2000. DOI: http://dx.doi.org/10.1080/002075400189301

JUNG, C. F. **Metodologia para pesquisa & desenvolvimento**: aplicada a novas tecnologias, produtos e processos. Rio de Janeiro: Axcel Books, 2004.

MARTINS, R.A. Abordagens Quantitativa e Qualitativa. In: MIGUEL, P.A.C.(Org.). **Metodologia de pesquisa em Engenharia de Produção e Gestão de Operações**. Rio de Janeiro: Elsevier, 2010. p. 45-61, cap.3.

MOCCELLIN, J.; NAGANO, M. Heuristic for flow shop sequencing with separated and sequence independent setup times. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v. 33, n. 1, p. 74-78, 2011. DOI: http://dx.doi.org/10.1590/S1678-58782011000100011

MOCCELLIN, J.; NAGANO, M. Uma propriedade estrutural do problema de programação da produção *flow shop* permutacional com tempos de *setup*. **Pesquisa Operacional**, v. 27, n. 3, p. 487-515, 2007. DOI: http://dx.doi.org/10.1590/S0101-74382007000300005

NAKANO, D. Métodos de Pesquisa Adotados na Engenharia de Produção e Gestão de Operações. In: MIGUEL, P.A.C.(Org.). **Metodologia de pesquisa em Engenharia de Produção e Gestão de Operações**. Rio de Janeiro: Elsevier, 2010. p. 63-72, cap.4.

PINEDO, M. L. **Scheduling:** theory, algorithms and systems. New York: Springer, 2016, 5th ed.

RONCONI, D.P.; BIRGIN, E.G. Mixed-integer programming models for flow shop scheduling problems minimizing the total earliness and tardiness. In **Just-in-Time Systems**, Y.A. Ríos-Solís and R.Z. Ríos-Mercado (Eds.), Springer Series on Optimization and Its Applications, P.M. Pardalos and Ding-Zhu Du (Series eds.). 2012.

TAILLARD, E. Benchmarks for basic scheduling problems. **European Journal of Operational Research**, v. 64, p. 278-285, 1993. DOI: https://doi.org/10.1016/0377-2217(93)90182-M

Artigo recebido em 28/03/2017 e aceito para publicação em 11/08/2017
DOI: http://dx.doi.org/10.14488/1676-1901.v18i2.2819