

# Análise de algoritmo construtivo para otimização de diferentes critérios em ambiente integrado *scheduling*-distribuição

*Analysis of a constructive algorithm to optimize different performance measures in an integrated scheduling-distribution environment*

Roberto Fernandes Tavares Neto<sup>1</sup> - Universidade Federal de São Carlos - Departamento de Engenharia de Produção  
Ronaldo Castro de Oliveira<sup>2</sup> - Universidade Federal de São Carlos - Departamento de Engenharia de Produção

**RESUMO** A literatura já demonstrou a necessidade de se considerar a integração entre diversas funções de sistemas produtivos. Nesse sentido, o presente artigo apresenta um algoritmo construtivo para o problema integrado *scheduling*-distribuição, onde o ambiente produtivo é uma única máquina, e a distribuição é composta de um único veículo capacitado, capaz de realizar múltiplas rotas. Um conjunto de 7 regras de inicialização é avaliado. As análises realizadas são baseadas no *gap* encontrado em soluções ótimas (no caso de problemas de pequeno porte) ou entre os algoritmos (para as demais instâncias). Para a análise, as instâncias são analisadas de acordo com os parâmetros que foram utilizados para sua geração. Fica evidenciado que a eficácia do operador de inserção depende muito do objetivo a ser minimizado. Além disso, percebeu-se que a ordenação baseada em tempos de *setup* permitiu ao operador de inserção a obtenção de melhores resultados finais.

**Palavras-chave:** *Scheduling*. Roteirização. Distribuição. Algoritmo Construtivo.

**ABSTRACT** *The literature has already shown the need to consider the integration of several functions of productive systems. Based on this, the current paper introduces a constructive algorithm for the integrated scheduling-distribution problem, where the productive environment is a single machine and the distribution is composed of a single vehicle capable of performing multiple routes. A series of seven initialization rules is evaluated. The analyses performed are based on the gap found in optimal solutions (in the case of small scale problems) or among the algorithms (for the remaining problems.) For the analyses, the problems are analyzed according to the parameters that were used for their generation. It is evident that the efficiency of the insertion operator depends substantially on the objective that is to be minimized. Further, that the set-up based ordering times enabled the insertion operator to obtain better results was evident.*

**Keywords:** *Scheduling*. Routing. Distribution. Constructive Algorithm.

1. Rodovia Washington Luís, s/n, CEP: 13565-905, São Carlos/SP, tavares@dep.ufscar.br; 2. ronaldo.co@gmail.com

NETO, R. F. T.; OLIVEIRA, R. C. Análise de algoritmo construtivo para otimização de diferentes critérios em ambiente integrado *scheduling*-distribuição. **GE-PROS. Gestão da Produção, Operações e Sistemas**, Bauru, Ano 12, nº 2, abr-jun/2017, p. 21-42.

DOI: 10.15675/gepros.v12i2.1631

## 1. INTRODUÇÃO

A integração entre a produção de bens e sua distribuição é um tema em crescente expansão, gerando tantos resultados muito relevantes do ponto de vista científico quanto do ponto de vista de aplicações.

Inspirado na literatura, que muito frequentemente trata o caso da integração produção-distribuição na indústria de bens com vida útil curta, ou itens perecíveis (GEISMAR et al., 2008; CHIANG et al., 2009), o objetivo do problema abordado por esse trabalho se foca em indicadores clássicos de sequenciamento e programação de operações (*scheduling*) relacionados ao tempo em que ordens de serviço ficam no sistema. Inspirado na necessidade de desenvolvimento teórico nessa área observado por várias pesquisas (ERENGÜÇ et al. 1999; CHEN; LEE 2004), esse trabalho se foca no estudo e desenvolvimento de algoritmos que integram, no nível operacional, decisões de *scheduling* e de roteamento de veículos considerando uma única máquina e um único veículo com múltiplas rotas de entrega - dois dos sub-sistemas básicos encontrados em sistemas produtivos, seja na literatura de *scheduling* (um ambiente de máquina única) ou roteirização (um único veículo capacitado capaz de múltiplas rotas). Seguindo o observado pela literatura - e fortemente inspirados por problemas clássicos de *scheduling* - são apresentados estudos que se focam na minimização de dois indicadores de desempenho: o somatório do momento de entrega das ordens de serviço (ou tempo de fluxo, uma extensão natural do indicador de desempenho tempo total de finalização, também adotada por Li et al. (2005) e o momento que o último veículo retorna da última entrega extensão do indicador *makespan*, adotado por Geismar et al. (2008).

Para resolver esses problemas, esse artigo se utiliza de um algoritmo construtivo baseado na técnica NEH (NAWAZ et al., 1983) e proposto inicialmente por Tavares Neto e Oliveira (2015), no qual são experimentados sete diferentes algoritmos de inicialização para cada função objetivo definida (tempo de fluxo e *makespan*). Esses algoritmos são aplicados em um conjunto de 2.430 instâncias de teste contendo 5 ordens de serviço e comparados com a solução ótima. Em instâncias maiores, os oito algoritmos de inicialização têm seu comportamento avaliado comparativamente com o uso de 12.150 instâncias de teste de ordem maior ou igual.

Para apresentar o algoritmo, em conjunto com suas análises, o presente trabalho está organizado da seguinte forma: a Seção 2 identifica as contribuições mais recentes no tema integração produção-distribuição, se focando na indústria de produtos perecíveis; a Seção 3 define o problema; a Seção 4 apresenta os algoritmos desenvolvidos para esse artigo; a Seção 5 apresenta os dados e os analisa; por fim, a Seção 6 apresenta as considerações finais.

## 2. REFERENCIAL TEÓRICO

Cada vez mais, percebe-se que o planejamento integrado de várias funções distintas de sistemas produtivos é fator chave para o sucesso das empresas (MIN; ZHOU 2002). Para tal, busca-se determinar como otimizar um conjunto de indicadores de desempenho com a especificação de um conjunto de decisões normalmente consideradas de forma isolada. A literatura traz um conjunto de trabalhos que se focam em diferentes formas de integração entre diferentes funções da produção, sendo classificadas por Chen e Lee (2004) como: estratégia de compras, gestão de fornecimento, integração de logística e coordenação de cadeia de suprimento.

Uma das funções da logística é, uma vez definido o conjunto de pontos de coleta de material (por exemplo, armazéns) e o conjunto de pontos de entrega (por exemplo, clientes), determinar como é realizada a distribuição dos produtos. Algumas estruturas de distribuição são citadas de forma comum na literatura, Li et al. (2005): distribuição por canais dedicados (onde um veículo atende um cliente exclusivamente), distribuição por veículos que se utilizam de rotas fixas (visitando clientes sendo necessário ou não) e distribuição para múltiplos clientes por múltiplas rotas (REIMANN et al., 2014). Algumas tentativas de se simplificar o problema de roteirização também são encontradas na literatura (FU et al., 2015) existe um custo fixo para cada rota, e cada entrega adiciona um valor constante à função objetivo).

O momento de liberação de rotas também é tratado de diferentes formas: por exemplo, Leung e Chen (2013) considera instantes de despacho de rotas constantes, enquanto também é encontrada na literatura trabalhos que consideram que o despacho de veículos pode ser realizada a qualquer momento (TAVARES NETO; OLIVEIRA, 2015; TAVARES NETO; ASANO, 2015; TAVARES NETO; OGAWA, 2015)

A produção, por sua vez, é tratada na literatura através de duas principais abordagens:

a primeira, normalmente classificada como uma decisão de nível tático, se foca no dimensionamento de lotes de produção e planejamento de custos de estoque; a segunda, se foca em decisões de curto prazo de sequenciamento e programação da produção (*scheduling* - nível operacional).

Em todas as variações do problema de produção-distribuição integrado mencionadas acima, percebe-se que existe um consenso na importância das medidas de performance baseadas em tempo da produção (CHEN; VAIRAKTARAKIS, 2005) indica a importância de se otimizar a rapidez de entrega e o *lead-*

-*time* do sistema produtivo). Essa necessidade é reforçada quando observamos ambientes produtivos de produtos perecíveis, como produção de jornais, gêneros alimentícios e outros. Em tais sistemas, existe uma necessidade inerente de se encurtar o tempo entre o recebimento da ordem e sua entrega ao cliente. Nesse sentido, na literatura científica, percebe-se que existe um interesse cada vez maior no estudo de técnicas que unam o planejamento da produção de bens como a distribuição com o fim de minimizar tempo de entrega, estoques de produtos acabados aguardando entrega, entre outros (CHEN, 2010; REIMANN et al., 2014).

Em nossa análise, nenhum trabalho foi encontrado tratando de múltiplas rotas de entrega. Além disso, não foi encontrado nenhum algoritmo construtivo para a resolução de problemas integrados *scheduling*/distribuição.

### 3. DEFINIÇÃO DO PROBLEMA ESTUDADO

#### 3.1. Parâmetros de entrada

Conforme mencionado anteriormente, o presente trabalho tem como alvo um problema integrado produção-distribuição composto de um ambiente de máquina única (produção) e um único veículo capacitado (distribuição). Esse problema requer os seguintes parâmetros para ser definido:

- $i, j$  : índice das ordens de serviço
- $k$  : índice da rota de entrega
- $n$  : número de ordens de serviço
- $\rho_i$  : tempo de processamento da ordem de serviço
- $\kappa_{ij}$  : tempo de *setup* entre ordens e
- $\delta_{ij}$  : distância entre os pontos de entrega das ordens e
- $\sigma_i$  : tamanho (volume) da ordem
- $\psi$  : capacidade do veículo

#### 3.2. Variáveis de decisão

A resposta é composta dos seguintes conjuntos ordenados:

$S_\pi$  : a sequência de ordens a serem produzidas no ambiente produtivo

$V_\pi^k$  : as sequências de  $k$  rotas

Por fim, define-se os seguintes indicadores:

$L_k$  : momento de partida da rota

$C_i$  : tempo de finalização da ordem no ambiente produtivo

$V_i$  : momento em que os produtos referentes à ordem chegam ao destino

$M$  : momento em que o último veículo retorna ao ponto de origem

Nossa função objetivo é, então, minimizar  $\sum_i V_i$  ou  $M$ .

### 3.3. Restrições do problema

Dadas as sequências  $S_\pi$  e  $V_\pi^k$ , os indicadores definidos acima são definidos através da programação das tarefas e rotas usando as seguintes diretrizes:

1. Existem dois limitantes para o valor de  $L_k$ :
  - a) A rota  $k$  só pode iniciar após o retorno da rota  $k - 1$  (com exceção da primeira rota, que pode iniciar a qualquer momento, desde que as ordens respectivas estejam produzidas);
  - b) A rota  $k$  só pode iniciar após todas as ordens pertencentes a  $V_\pi^k$  tenham sido produzidas.
2. A primeira ordem de serviço inicia seu processamento em  $t = 0$ .
3. Seja  $i \in V_\pi^k$  a ordem de serviço da rota  $k$  com maior índice no conjunto  $S_\pi$ . Seu tempo de finalização  $C_i$  deve ser menor que  $L_k$ .
4. Não existe tempo de inatividade do veículo enquanto o mesmo estiver entregando produtos acabados. A inatividade do mesmo ocorrerá apenas quando estiver aguardando a produção disponibilizar as ordens de produção.

## 4. ALGORITMOS CONSTRUTIVOS PARA A RESOLUÇÃO DO PROBLEMA

Provavelmente um dos algoritmos de maior sucesso e impacto na literatura de *scheduling* é o NEH, proposto por Nawaz et al. (1983) para minimizar o makepan em ambientes *flowshop* permutacional. Esse algoritmo, assim como um conjunto de extensões do mesmo (FRAMINAN et al., 2010), é composto de três fases distintas: a Fase de Ordenação onde o conjunto de trabalhos a serem

processados são colocados em uma lista L ordenada segundo uma regra específica; a Fase de Inserção: nessa fase, cada uma das tarefas é, uma a uma, removida de L e testada em cada posição da sequência final, sendo alocada na posição que gera o melhor valor para a função objetivo; e a Fase de Aprimoramento (opcional), quando a sequência obtida na fase anterior é submetida a um algoritmo de busca local.

Devido à relevância do algoritmo NEH e suas extensões na literatura de sequenciamento e programação da produção, esse artigo se baseia nesse algoritmo e propõe um algoritmo baseado em dois estágios para a resolução do problema apresentado na seção 3. Os estágios são:

### **I – Fase de ordenação**

Nessa fase, o conjunto de trabalhos a serem processados são colocados em uma lista L ordenada segundo uma regra específica. Foram utilizadas oito regras, gerando assim oito algoritmos distintos, a saber:

- FIFO – *First In, First Out* (primeiro a chegar é o primeiro a sair): essa regra não faz nenhum tipo de ordenação.
- SPT – *Shortest Processing Time* (menor tempo de processamento): ordena as tarefas conforme seu tempo de processamento, em ordem crescente;
- LPT – *Longest Processing Time* (maior tempo de processamento): ordena as tarefas conforme seu tempo de processamento, em ordem decrescente;
- SIZE: ordena as tarefas por tamanho, em ordem crescente;
- SIZEDEC: ordena as tarefas por tamanho, em ordem decrescente;
- NNSETUP: ordena as tarefas através de um algoritmo padrão de caminhos mínimos (onde as distâncias entre duas ordens são dadas pelos tempos de setup respectivos);
- NNDIST: ordena as tarefas através de um algoritmo padrão de caminhos mínimos

### **II – Fase de inserção**

A fase de inserção da heurística proposta é muito semelhante ao NEH: inicialmente, o primeiro trabalho da lista ordenada é inserido na sequência produtiva final, e inserida em uma rota vazia. Depois, cada tarefa é inserida na sequência de produção (como o NEH) e, para cada posição, testa-se cada posição de cada rota na sequência de distribuição. Além disso, também são testadas a criação de novas rotas no início, entre as rotas e após todas as rotas. Como o

NEH, cada tarefa é alocada nas posições que melhor satisfazem a função objetivo do problema.

Essas 7 variações das heurísticas foram implementadas em C++ e executadas em um computador i7 rodando Linux Mint. Os resultados obtidos são mostrados a seguir.

## 5. EXPERIMENTOS COMPUTACIONAIS, RESULTADOS OBTIDOS E ANÁLISES

### 5.1. Conjunto de testes

Para análise da resposta dos algoritmos, um conjunto de teste extensivo foi criado da seguinte forma: para cada instância de teste, um conjunto de  $n \in [5,10,20,40,80]$  ordens de serviço é gerado. Do ponto de vista da produção das ordens, os seguintes parâmetros são utilizados: o tempo de produção  $k_{ij}$  é amostrado no intervalo  $[1, 100]$ . O tempo de *setup*  $\rho_j$  é gerado da seguinte forma: um espaço  $\theta_s \times \theta_s$ ,  $\theta_s \in [10,50,100]$  é criado, e as ordens são indexadas em uma posição aleatória nesse espaço. O tempo de *setup* entre duas ordens é dado pela distância euclidiana entre as ordens. Do ponto de vista da distribuição, a distância  $\delta_{ij}$  entre os pontos de entrega de duas ordens é obtida de forma semelhante, sendo usado um espaço  $\theta_d \times \theta_d$ ,  $\theta_d \in [5,10,30]$ . A distância entre as ordens e a origem é multiplicada por um fator  $\theta_g \in [1,10,30]$ . O tamanho  $\sigma_i$  de cada ordem é amostrada em um intervalo  $[1, 10]$ . A capacidade de cada caminhão é dada por  $\Psi = \max\{\sigma_i\} \cdot [1, \theta_s]$ . Isso resultou em um total de 12.150 problemas-teste.

A análise dos algoritmos foi realizada em duas etapas: na primeira etapa, apresentada na seção 5.1, para as 2.430 instâncias de 5 tarefas, se obteve a resposta ótima através de um modelo de programação linear mista (tal modelo não está presente no presente trabalho por limitações de espaço, podendo ser disponibilizado pelos autores por solicitação). Porém, tal abordagem não foi capaz de obter a solução ótima para instâncias maiores. Assim, em uma segunda análise, apresentada na seção 5.2, os diferentes algoritmos são comparados entre si. Em ambas análises, para cada uma das instâncias, calculou-se o valor do desvio percentual entre a melhor resposta obtida e a resposta do algoritmo. Esse desvio

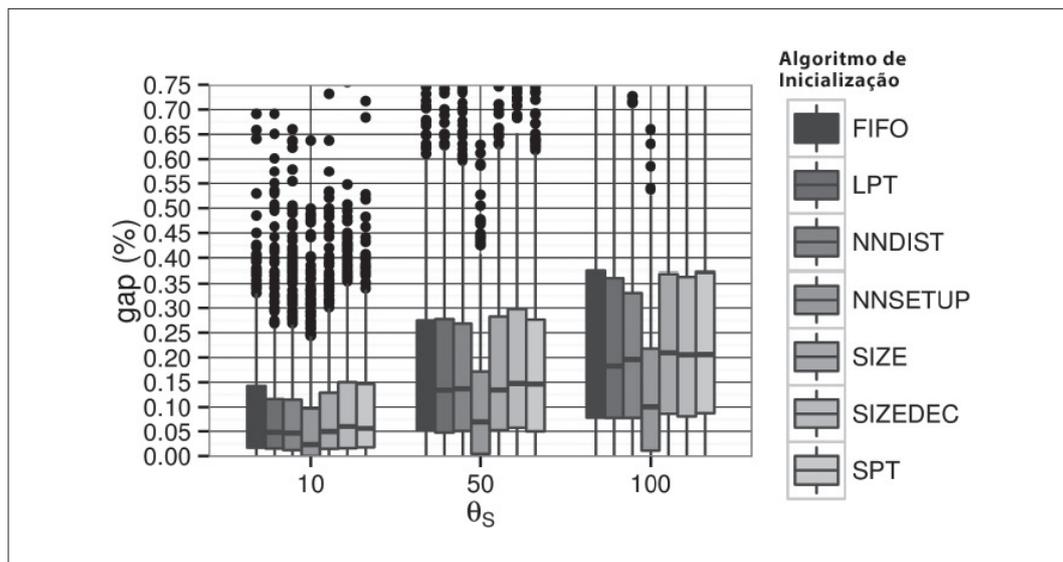
é indicado por  $\text{gap} = \frac{F - F^*}{F^*} \times 100$ , sendo  $F$  a solução obtida pelo algoritmo em questão e  $F^*$  a melhor solução encontrada para a instância analisada.

## 5.2. Análise das instâncias $n = 5$

### 5.2.1. Dados agrupados conforme valor de $\theta_s$

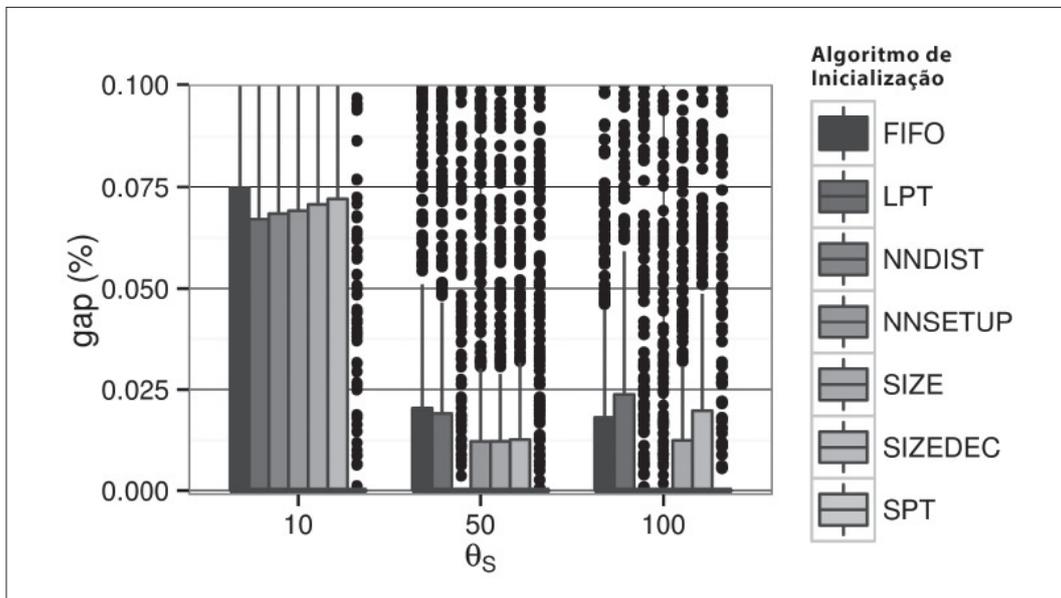
A Figura 1 traz os resultados referentes à função objetivo *Makespan*. A Figura 2 traz os resultados referentes à função objetivo  $\sum V_i$ . Pode-se perceber que a mudança da função objetivo alterou significativamente o comportamento do algoritmo: no caso do *Makespan*, a regra de ordenação NNSETUP mostrou os melhores resultados, enquanto para o  $\sum V_i$  as melhores regras foram NNDIST e o SPT. Ainda no caso do *makespan*, um aumento do valor de  $\theta_s$  produziu um maior valor de *gap*. No caso do  $\sum V_i$ , o efeito foi contrário.

Figura 1 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_s$ .



Fonte: Os autores (2016).

Figura 2 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_i$ , agrupados por  $\theta_s$ .

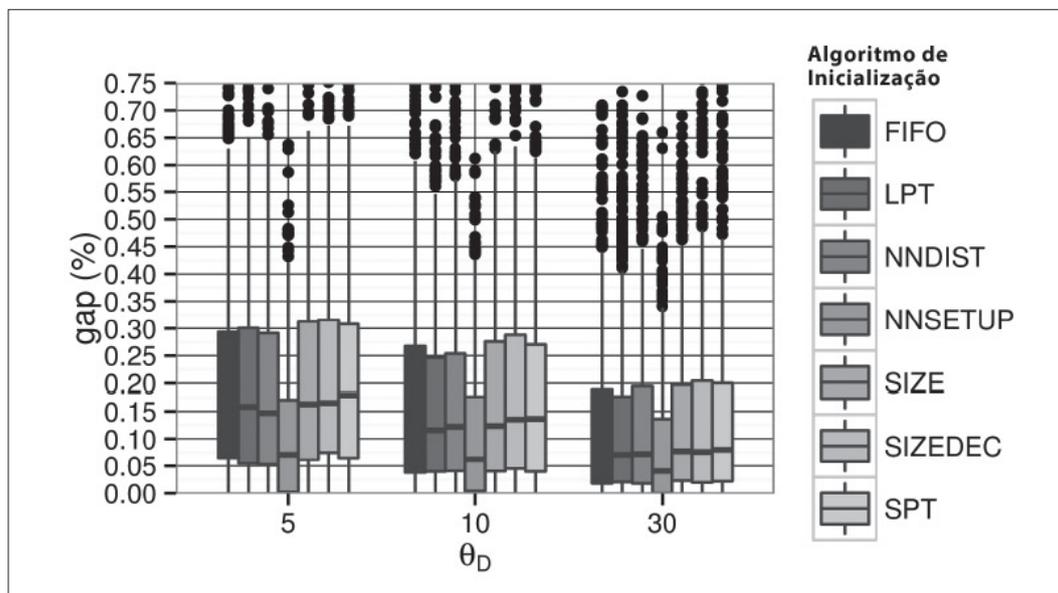


Fonte: Os autores (2016).

### 5.2.2. Dados agrupados conforme valor de $\theta_D$

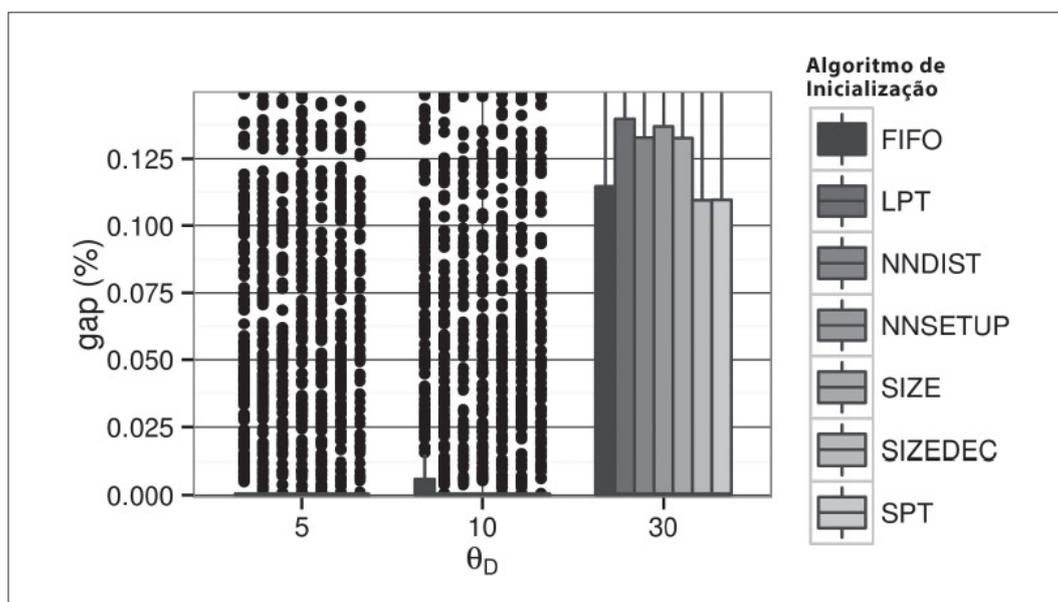
Como pode ser observado nas Figuras 3 (*Makespan*) e 4 ( $\sum V_i$ ), o algoritmo mostra comportamentos bem diferentes com as diferentes funções objetivo: enquanto no caso do *makespan* temos um resultado praticamente constante de *gaps* médios, onde sempre a regra NNSETUP gera resultados melhores, no caso do  $\sum V_i$  um valor maior de  $\theta_D$  produz resultados elevados de *gap*, onde a regra SIZEDEC e SPT produzem resultados minimamente melhores.

Figura 3 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_D$ .



Fonte: Os autores (2016).

Figura 4 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_p$ , agrupados por  $\theta_D$ .

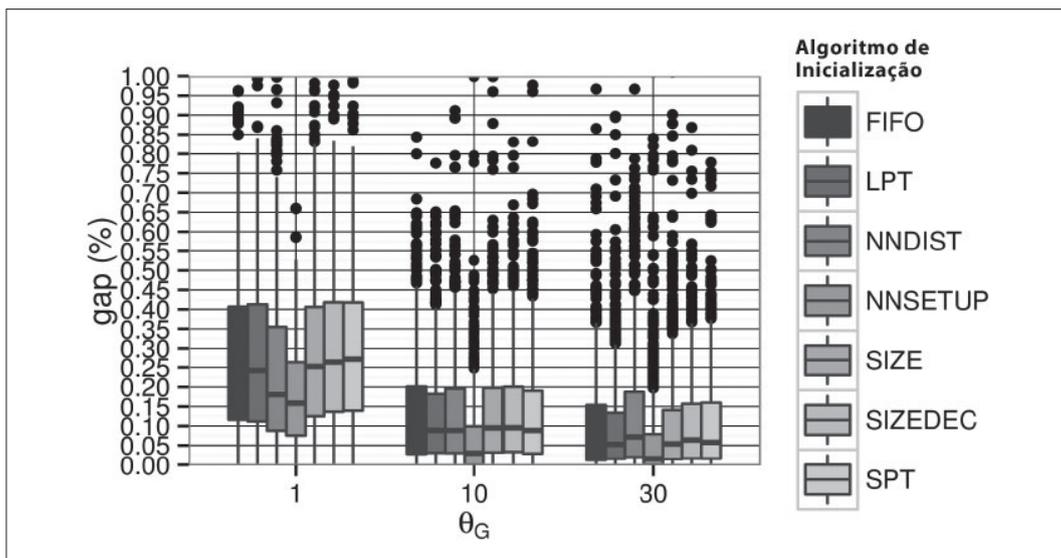


Fonte: Os autores (2016).

### 5.2.3. Dados agrupados conforme valor de $\theta_G$

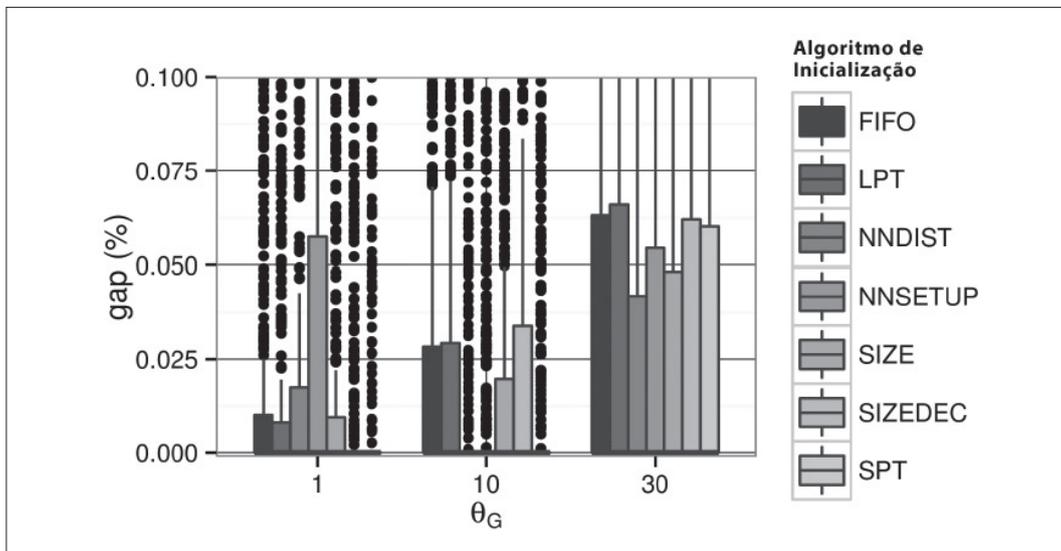
Quando a distância entre o centro produtivo e os pontos de entrega são alterados, percebe-se que, quando se considera o *Makespan* (Figura 5), o *gap* é significativamente mais alto para o valor  $\theta_G = 1$ , e relativamente estável para maiores valores de  $\theta_G$ . Em todos os três casos, a regra NNSETUP se mostrou mais adequada para a resolução do problema. Já a Figura 6 mostra que, para a função objetivo  $\sum V_p$ , o crescimento de  $\theta_G$  gera um aumento do *gap*. Mais que isso, quando  $\theta_G = 1$ , as regras SIZEDEC e SPT produzem um melhor efeito. Para o caso de  $\theta_G = 10$ , as regras NNDIST e NNSETUP trouxeram melhores resultados. Por fim, com  $\theta_G = 30$ , a regra NNDIST se mostrou superior às demais.

Figura 5 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_G$ .



Fonte: Os autores (2016).

Figura 6 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_i$ , agrupados por  $\theta_G$ .

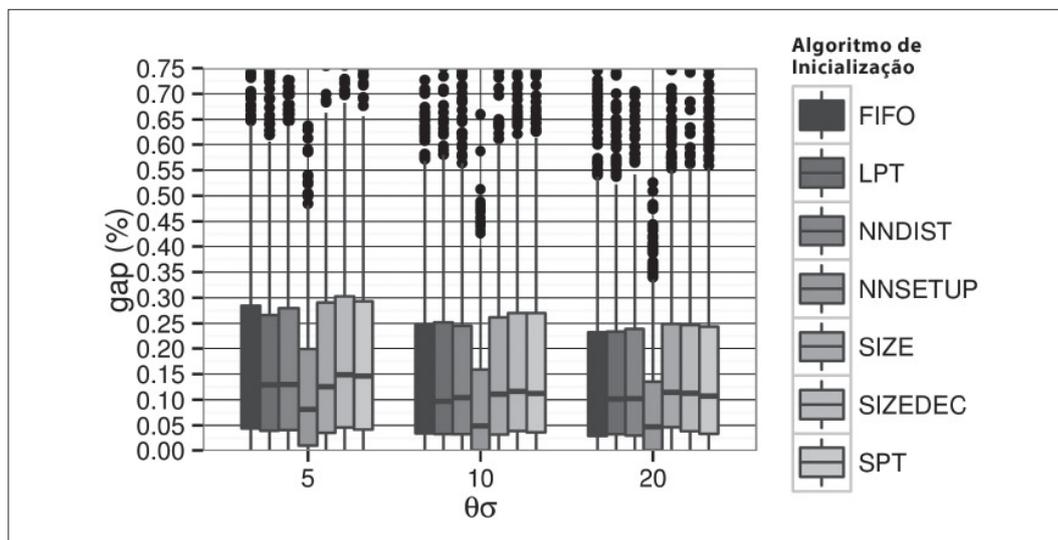


Fonte: Os autores (2016).

#### 5.2.4. Dados agrupados conforme valor de $\theta_\sigma$

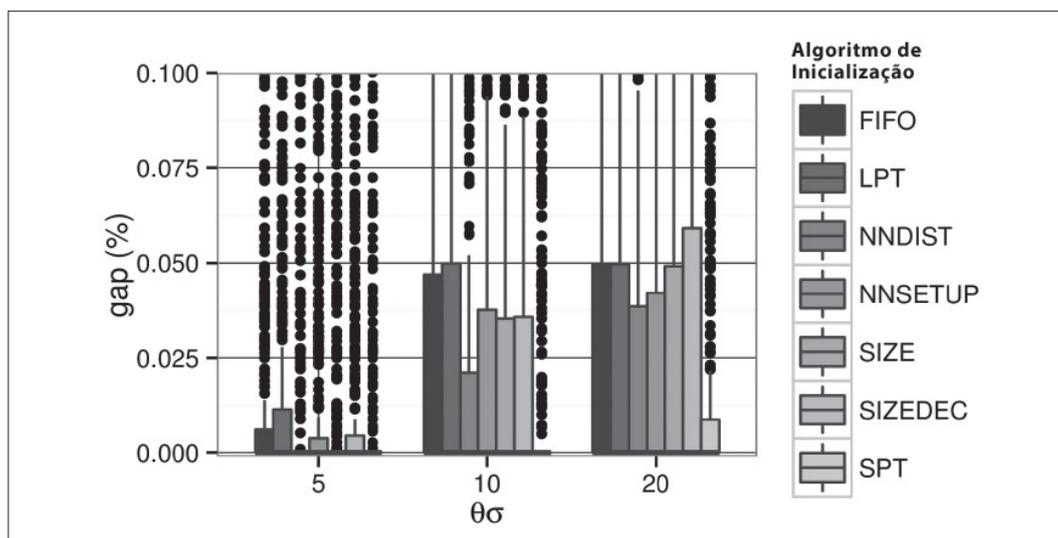
As Figuras 7 e 8 trazem, respectivamente, as análises referente aos algoritmos quando se adota o *Makespan* ou  $\sum V_i$  como função objetivo. No caso do *Makespan*, pode-se perceber que existe uma certa estabilidade dos resultados (com leve queda com o aumento de  $\theta_\sigma$ ). Em todos os casos, a melhor regra de inicialização foi novamente o NNSETUP. No caso do  $\sum V_i$ , percebe-se que os *gaps* encontrados quando  $\theta_\sigma = 5$  são muito baixos. Para maiores valores de  $\theta_\sigma$ , a regra SPT produziu melhores resultados.

Figura 7 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_\sigma$ .



Fonte: Os autores (2016).

Figura 8 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_i$ , agrupados por  $\theta_\sigma$ .



Fonte: Os autores (2016).

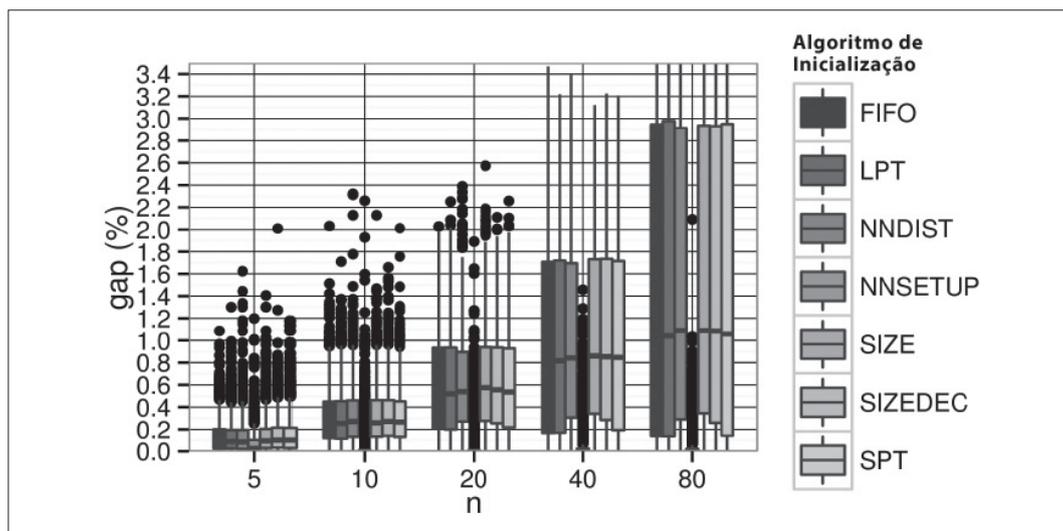
### 5.3. Análise de todas as instâncias de problemas

A seguir, são apresentadas as demais análises do problema. É importante notar que, nesse caso, não foi possível utilizar o modelo de programação matemática para se obter o valor ótimo. Dessa forma, a comparação entre os dados se deu apenas entre os valores encontrados pelo próprio algoritmo.

#### 5.3.1. Dados agrupados conforme tamanho das instâncias

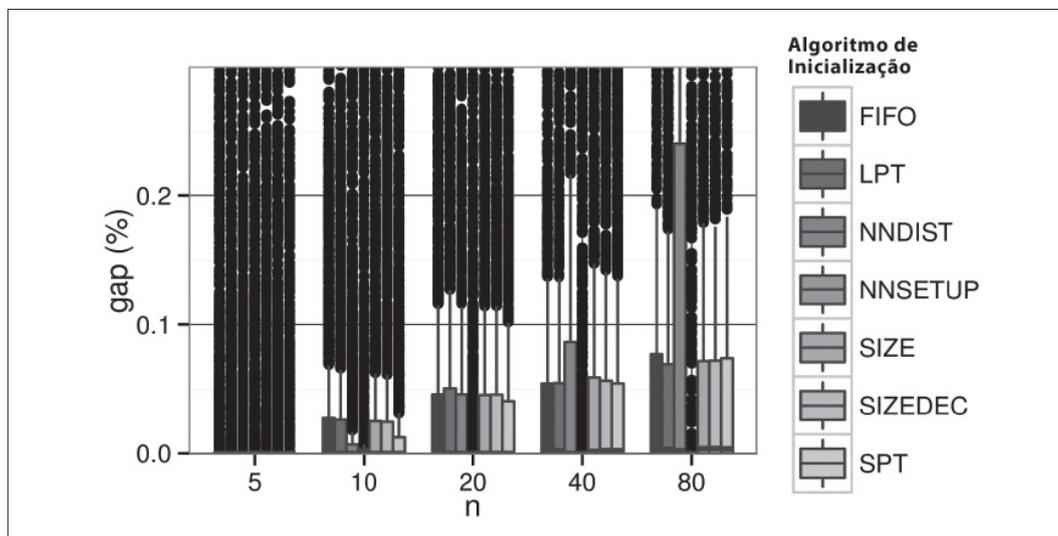
Quando se considera o total de instâncias geradas, percebe-se que, como esperado, o aumento do tamanho das instâncias gera um maior *gap* entre as respostas de ambos os algoritmos. Também em ambos os algoritmos, a melhor regra de inicialização para os problemas foi o NNSETUP.

Figura 9 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por *n*.



Fonte: Os autores (2016).

Figura 10 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_i$ , agrupados por n.

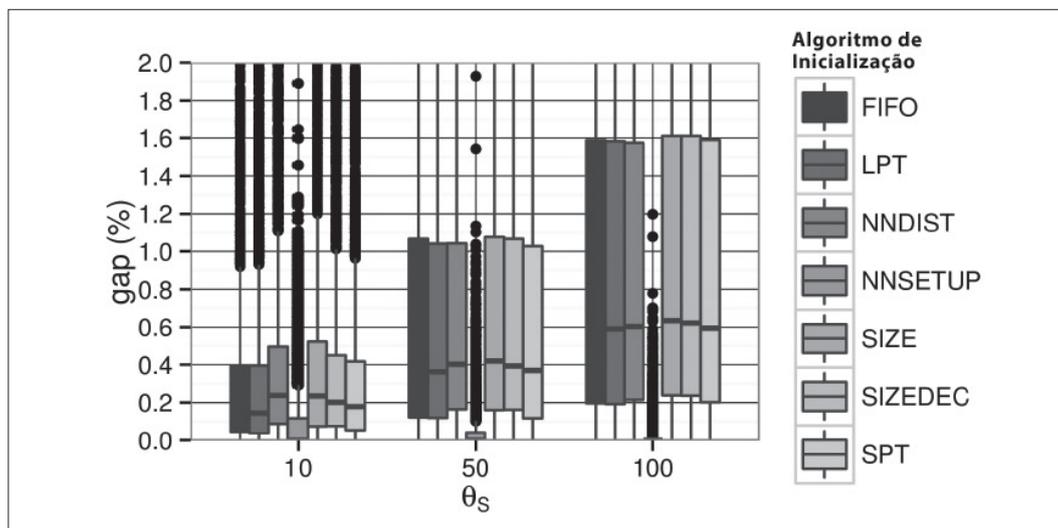


Fonte: Os autores (2016).

### 5.3.2. Dados agrupados conforme valor de $\theta_s$

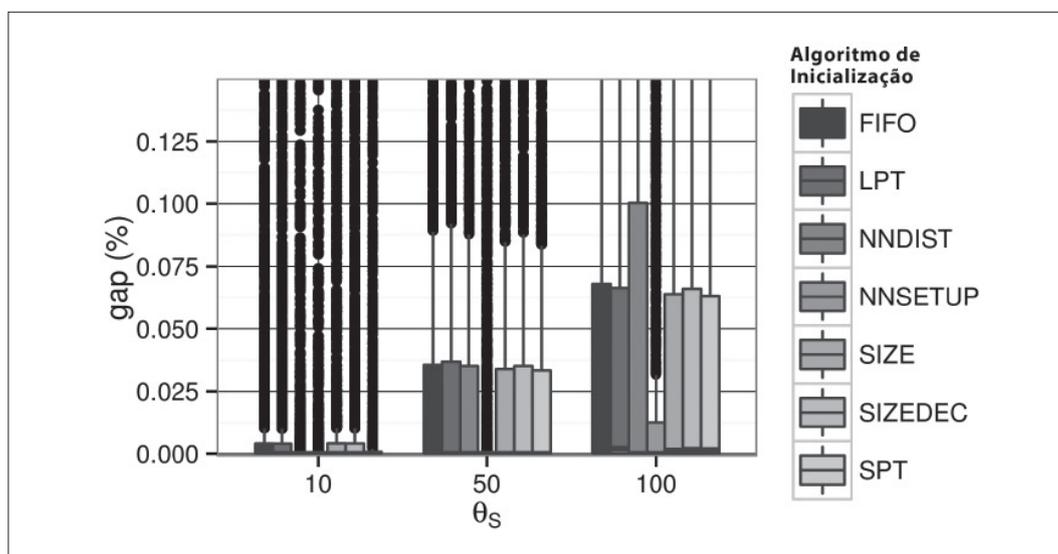
Em uma análise mais abrangente que a realizada na seção 5.2.1, percebe-se que o algoritmo, em ambos os casos, demonstrou dificuldade em tratar aumento do parâmetro  $\theta_s$ . Tanto no caso da aplicação do algoritmo para resolver o *Makespan* quanto  $\sum V_i$ , a regra NNSETUP se mostrou muito superior.

Figura 11 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_s$ .



Fonte: Os autores (2016).

Figura 12 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_p$ , agrupados por  $\theta_s$ .

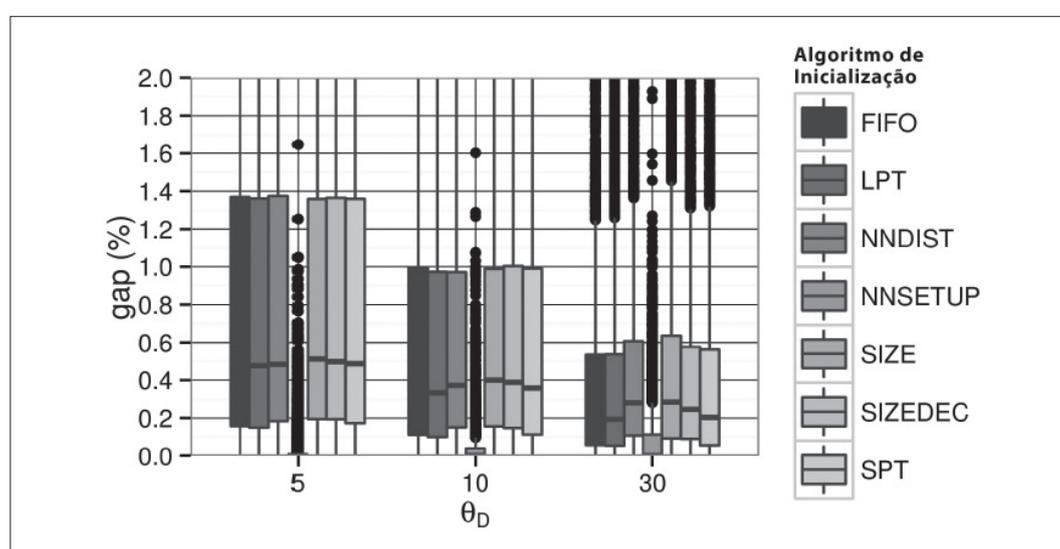


Fonte: Os autores (2016).

### 5.2.3. Dados agrupados conforme valor de $\theta_D$

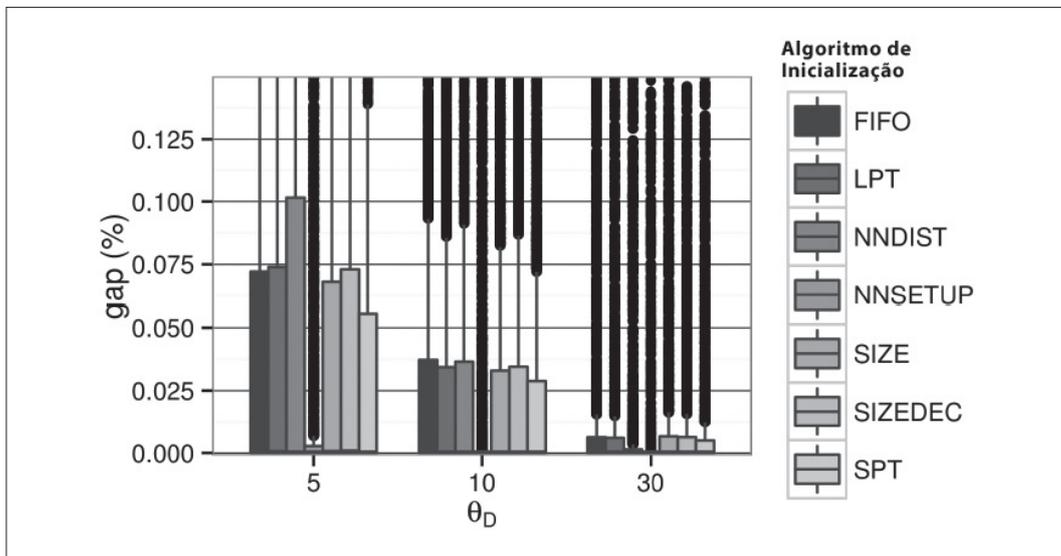
Analisando as Figuras 13 e 14, percebe-se que o aumento do valor de  $\theta_D$  em ambos os algoritmos, fez com o que as respostas sejam mais semelhantes. É importante observar que, como o valor mínimo utilizado para se calcular o *gap* nessa seção é o valor mínimo encontrado pelas execuções dos algoritmos, não se pode dizer que um aumento de  $\theta_D$  levou a produção de melhores resultados – pode-se, porém, dizer que os resultados tendem a ser mais semelhantes, ou seja, a regra de inicialização produziu um efeito menor na qualidade do algoritmo.

Figura 13 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_D$ .



Fonte: Os autores (2016).

Figura 14 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_i$ , agrupados por  $\theta_D$ .

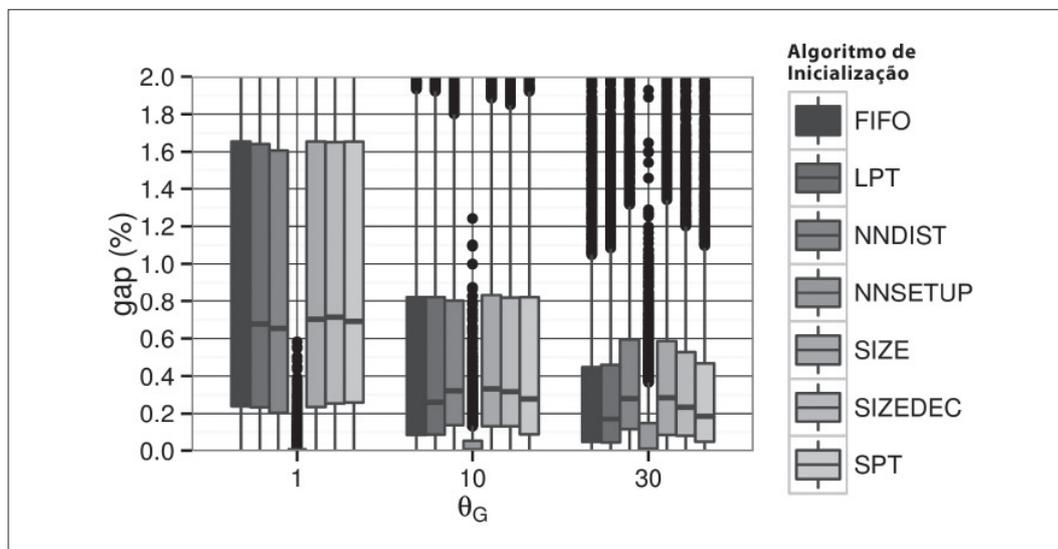


Fonte: Os autores (2016).

#### 5.2.4. Dados agrupados conforme valor de $\theta_G$

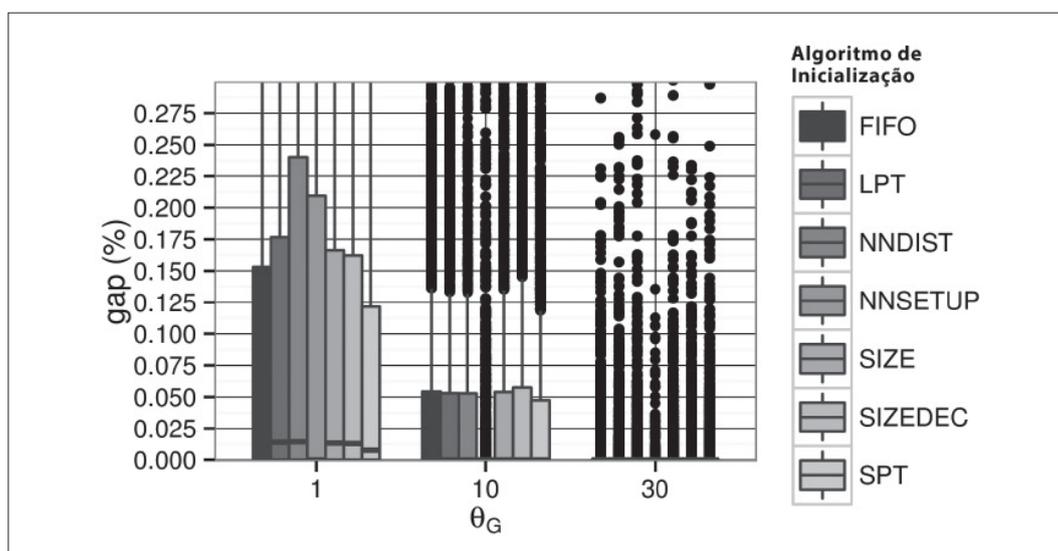
No caso do *Makespan* a variação do  $\theta_G$  gerou um comportamento muito semelhante à variação do  $\theta_D$ : *gap* decrescente, com a regra NNSETUP gerando melhores resultados. No caso da regra  $\sum V_i$ , tal comportamento de *gap* decrescente também é observado, chegando ao ponto onde, para  $\theta_G = 30$ , a variação das regras de inicialização se mostrou ineficiente como estratégia de melhorar a solução do algoritmo.

Figura 15 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_G$ .



Fonte: Os autores (2016).

Figura 16 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_p$ , agrupados por  $\theta_G$ .

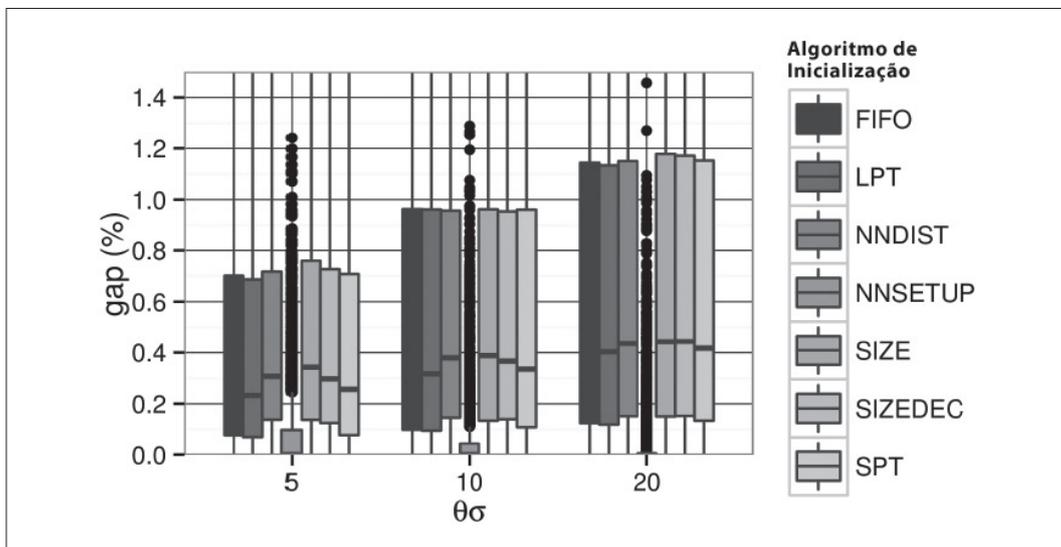


Fonte: Os autores (2016).

### 5.2.5. Dados agrupados conforme valor de $\theta_\sigma$

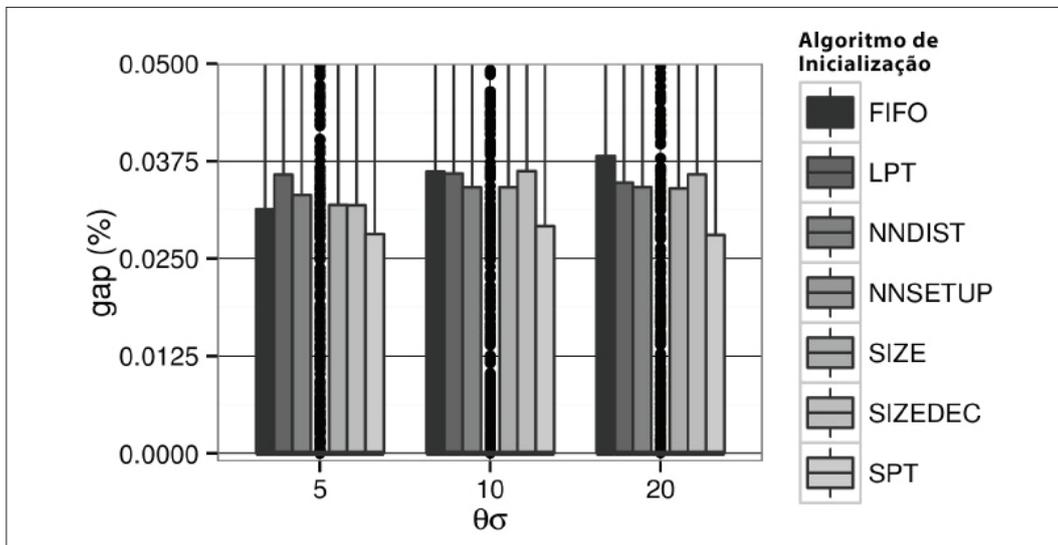
A Figura 17 mostra que o aumento do parâmetro  $\theta_\sigma$  para a função objetivo *Makespan* fez uma piora significativa de todas as regras, com exceção da regra NNSETUP. Essa regra se mostrou mais eficiente que as demais. Já a Figura 18 mostrou que a regra NNSETUP gerou melhores resultados para o problema de minimização de  $\sum V_i$ , não importando o valor de  $\theta_\sigma$ . Porém o *gap* encontrado nesse caso é muito pequeno (menos de 4%).

Figura 17 – Análise dos resultados obtidos quando se adota a função objetivo *makespan*, agrupados por  $\theta_\sigma$ .



Fonte: Os autores (2016).

Figura 18 – Análise dos resultados obtidos quando se adota a função objetivo  $\sum V_i$ , agrupados por  $\theta_\sigma$ .



Fonte: Os autores (2016).

## 6. CONCLUSÕES

O presente trabalho mostra uma análise comparativa no uso do algoritmo construtivo proposto em Tavares Neto e Oliveira (2015) para a resolução do problema de minimização de somatório de tempos de fluxo e *makespan*.

Com os dados obtidos pela aplicação dos algoritmos no mesmo conjunto de testes, percebeu-se que, em alguns subgrupos de problemas, a regra de inicialização não é relevante, o que pode nos dizer que o algoritmo de inserção consegue gerar resultados melhores do que a regra de ordenação isoladamente. Mais do que isso, ficou claro que a regra NNSETUP é a mais adequada entre as regras analisadas para a minimização do *makespan*. A regra NNSETUP também foi a mais adequada em várias análises da minimização do tempo de fluxo. Porém, para casos específicos, a regra SPT gerou resultados melhores do que a regra NNSETUP. Percebe-se adicionalmente que o algoritmo de inserção se comporta de forma diferente de acordo com a função objetiva adotada. Essas análises sinalizam que: (i) futuras melhorias da regra de inicialização devem contar com a informação do setup dos trabalhos; (ii) a regra de inserção pode ser melhorada consideravelmente no caso da minimização do *makespan*.

Entende-se que o presente trabalho possui diversas ramificações: além de expandir o ambiente produtivo (para máquinas paralelas, *flowshop*, etc) e de distribuição (para múltiplos veículos, janelas de tempo, etc), o algoritmo em si pode ser melhorado. Como principais caminhos de pesquisa nesse sentido,

elencam-se: (i) o aprimoramento da regra de inicialização; (ii) a adoção de outras regras que substituam os algoritmos de inserção; e (iii) o desenvolvimento de buscas locais. Além disso, percebe-se uma urgência na obtenção de exemplares de problemas provenientes de ambientes industriais reais.

## REFERÊNCIAS

CHIANG W.; RUSSELL R.; XU X.; ZEPEDA D. A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. **International Journal of Production Economics**, n. 121, p. 752–767. 2009.

FRAMINAN, J. M.; NAGANO, M. S.; MOCCELLIN, J. V. An efficient heuristic for total flowtime minimization in no-wait flowshops. **International Journal of Advanced Manufacturing Technology**, v. 46, p. 1049-1057, 2010.

GEISMAR H.N.; LAPORTE G.; LEI L.; SRISKANDARAJAH C. The integrated production and transportation scheduling problem for a product with a short life span and non-instantaneous transportation time. **INFORMS Journal on Computing**, n. 20, p. 21–33. 2008.

NAWAZ, M.; ENSCORE, E. E.; HAM, I. A heuristic algorithm for the m-machine, n-job flowshop sequencing problem. **Omega**, v. 11, p. 91-95, 1983.

ÖNAL, M.; ROMEIJ, H. E.; SAPRA, A.; HEUVEL, W. The economic lot-sizing problem with perishable items and consumption order preference. **European Journal of Operational Research**, n. 244, p. 881-891. 2015.

REIMANN, M.; TAVARES NETO, R. F.; BOGENDORFER, E. Joint Optimization of Production Planning and Vehicle Routing Problems: a Review of Existing Strategies. **Pesquisa Operacional**, n. 34, p.189-214. 2014.

SOYSAL, M.; BLOEMHOF-RUWAARD, J. M.; HAIJEMA, R.; VORST, J. G. A. J. Modeling an Inventory Routing Problem for Perishable Products with Environmental Considerations and Demand Uncertainty. **International Journal of Production Economics**, n. 164, p. 118-133. 2015.

TAVARES NETO, R. F.; OLIVEIRA, R. C. A Redução do Tempo Total de Fluxo em Sistemas Integrados Produção-Distribuição. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 22, 2015. **Anais...** UNESP, Bauru, São Paulo, 2015.

ZANONI, S.; ZAVANELLA, L. Single-vendor Single-buyer with Integrated Transport-Inventory System: Models and Heuristics in the Case of Perishable Goods. **Computers & Industrial Engineering**, n. 52, p. 107-123. 2007.