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

PROPOSIÇÃO DE ALGORITMO *SIMULATED ANNEALING* PARA PROGRAMAÇÃO EM *FLOW SHOPS* PARALELOS PROPORCIONAIS COM TEMPOS DE *SETUP* EXPLÍCITOS

A PROPOSAL SIMULATED ANNEALING ALGORITHM FOR PROPORTIONAL PARALLEL FLOW SHOPS WITH SEPARATED SETUP TIMES

Hélio Yochihiro Fuchigami* E-mail: helio@catalao.ufg.br
*Universidade Federal de Goiás (UFG), Catalão, GO

Resumo: Este trabalho aborda o problema de otimização da duração total da programação (*makespan*) em dois *flow shops* paralelos proporcionais com tempos de *setup* explícitos e independentes da sequência de tarefas. O problema de programação em *flow shops* paralelos é um caso específico do conhecido *flow shop* híbrido, caracterizado por sistemas produtivos multiestágio com mais de uma máquina operando em paralelo em cada estágio, e muito frequente em vários tipos de indústrias como química, eletrônica, automotiva, farmacêutica e alimentícia. Este trabalho objetivou propor seis algoritmos *Simulated Annealing* e seus respectivos esquemas de perturbação, além de um algoritmo para geração da sequência inicial, para solução do problema descrito. Este estudo pode ser classificado como "pesquisa aplicada" quanto à natureza, "pesquisa exploratória" quanto aos objetivos e "pesquisa experimental" quanto aos procedimentos, além da abordagem "quantitativa". Os algoritmos propostos mostraram-se eficazes quanto a solução e eficientes computacionalmente. Os resultados da Análise de Variância (ANOVA) revelaram que não existe diferença significativa entre os esquemas de perturbação em termos do *makespan*. Sugere-se a utilização do esquema PS4, que faz o deslocamento de uma subsequência de tarefas, por ter fornecido a melhor porcentagem de sucesso. Verificou-se também que é significativa a diferença entre os resultados dos algoritmos para cada valor do fator de proporcionalidade dos tempos de processamento e de *setup* dos *flow shops*.

Palavras-chave: Programação da produção. *Flow shops* paralelos. *Simulated annealing*. Tempos de *setup*.

Abstract: This article addresses the problem of minimizing makespan on two parallel flow shops with proportional processing and setup times. The setup times are separated and sequence-independent. The parallel flow shop scheduling problem is a specific case of well-known hybrid flow shop, characterized by a multistage production system with more than one machine working in parallel at each stage. This situation is very common in various kinds of companies like chemical, electronics, automotive, pharmaceutical and food industries. This work aimed to propose six Simulated Annealing algorithms, their perturbation schemes and an algorithm for initial sequence generation. This study can be classified as "applied research" regarding the nature, "exploratory" about the objectives and "experimental" as to procedures, besides the "quantitative" approach. The proposed algorithms were effective regarding the solution and computationally efficient. Results of Analysis of Variance (ANOVA) revealed no significant difference between the schemes in terms of makespan. It's suggested the use of PS4 scheme, which moves a subsequence of jobs, for providing the best percentage of success. It was also found that there is a significant difference between the results of the algorithms for each value of the proportionality factor of the processing and setup times of flow shops.

Keywords: Scheduling. Parallel flow shops. Simulated annealing. Setup times.

1 INTRODUÇÃO

A programação da produção pode ser definida genericamente como a alocação de recursos disponíveis para a execução de tarefas em um horizonte de tempo. Esta é uma importante decisão a ser tomada em sistemas de controle da produção. Para tanto, muitas técnicas de programação podem ser utilizadas, incluindo Programação Linear, métodos de solução exata como *Branch-and-bound*, procedimentos heurísticos e meta-heurísticas, a exemplo do Algoritmo Genético, *Simulated Annealing*, Busca Tabu, Colônia de Formigas e Algoritmo Imunológico.

O problema de programação em *flow shops* híbridos, caracterizados por ambientes multiestágio com mais de uma máquina operando em paralelo em cada estágio, tem recebido uma considerável atenção dos pesquisadores. Mais recentemente alguns trabalhos têm tratado dos ambientes com tempos de *setup* explícitos. O sistema de *flow shops* paralelos, um caso específico de *flow shop* híbrido, quase não tem sido abordado.

O objetivo deste trabalho é propor uma meta-heurística *Simulated Annealing* para resolver o problema de programação em dois *flow shops* paralelos proporcionais com tempos de *setup* explícitos, antecipados e independentes da sequência de execução de tarefas. O estudo visa ainda apresentar esquemas de perturbação a serem usados na meta-heurística e também um algoritmo de geração da solução inicial.

Segundo a revisão de Ruiz, Vásquez-Rodrígues (2010), as três meta-heurísticas mais utilizadas para o problema de *flow shop* híbrido e para programação da produção em geral são: Algoritmo Genético, em 8% dos trabalhos publicados abordando *flow shop* híbrido, Busca Tabu em 6% e *Simulated Annealing* em 5%. Ou seja, das três meta-heurísticas mais frequentes pelos resultados bem-sucedidos, a *Simulated Annealing* foi a menos utilizada. Esta é a justificativa pela sua escolha nesta pesquisa.

As seções seguintes deste artigo estão estruturadas da seguinte forma: a seção 2 contém a revisão da literatura, abordando os conceitos e trabalhos relacionados ao problema tratado. A seção 3 apresenta o algoritmo *Simulated Annealing* proposto e os seis esquemas de perturbação utilizados, além do algoritmo de geração de sequência inicial. A seção 4 traz definições do método da pesquisa, o

planejamento da experimentação computacional e a análise dos resultados. As considerações finais da pesquisa são descritas na seção 5.

2 REVISÃO BIBLIOGRÁFICA

2.1 *Flow shops* híbridos

Muitas publicações já abordaram o problema da programação em *flow shops* híbridos. Linn e Zhang (1999) propuseram uma classificação das pesquisas neste ambiente em três categorias: problemas com dois estágios, três estágios e mais de três estágios. Vignier, Billaut e Proust (1999) apresentaram para aquele momento o estado da arte de *flow shops* híbridos. Em geral as pesquisas se restringiram a situações ou configurações específicas de máquinas nos estágios, ou sem a presença de tempos de *setup* separados dos tempos de processamento.

Kis e Pesch (2005) atualizaram o estado da arte com trabalhos posteriores a 1999, enfocando métodos de solução exata para minimização do *makespan* e tempo médio de fluxo. Wang (2005) fez uma revisão da literatura, classificando-a em métodos de solução ótima, heurística e de inteligência artificial. Quadt e Khun (2007) publicaram uma taxonomia para *flow shop* híbridos, porém denominando-o de *flexible flow line*.

Ruiz e Vázquez-Rodríguez (2010) apresentaram uma revisão da literatura de métodos exatos, heurísticos e meta-heurísticos, discutindo as variações do problema, suas diferentes hipóteses, restrições e funções objetivo, além de apresentar as oportunidades de pesquisa na área. E uma extensa revisão dos trabalhos publicados recentemente (desde 1995) foi elaborada por Ribas, Leisten e Framiñan (2010), com um novo método de classificação dos trabalhos, do ponto de vista da produção, de acordo com as características das máquinas e das tarefas.

Como observou Quadt e Kuhn (2007), o sistema *flow shop* híbrido pode ser encontrado em um vasto número de indústrias, como química, eletrônica, de empacotamento, farmacêutica, automotiva, fabricação de embalagens de vidro, madeireira, têxtil, de herbicidas, alimentícia, de cosméticos e de semicondutores.

Estudos envolvendo tempos de *setup* dependentes da sequência de tarefas foram realizados por Kurz e Askin (2003), Lin e Liao (2003), Kurz e Askin (2004), Ruiz e Maroto (2006) e Ruiz et al. (2006).

Os trabalhos a seguir consideraram os tempos de *setup* independentes da sequência de execução das tarefas, que é o foco do presente trabalho.

O ambiente *flow shop* híbrido com dois estágios, sendo uma única máquina no primeiro e várias máquinas paralelas idênticas no segundo, com o critério de minimização do *makespan*, foi estudado por Gupta e Tunc (1994), Li (1997) e Huang e Li (1998).

Botta-Genoulaz (2000) estudou o problema de minimização do atraso máximo das tarefas sujeito ao mínimo tempo de transporte. Há também a presença de tempos de remoção das tarefas. O autor propôs e avaliou o desempenho de seis heurísticas. Allaoui e Artiba (2004) analisaram o problema *flexible flow shop* com várias medidas de desempenho, entre elas a minimização do *makespan* e do atraso máximo, considerando tempos de transporte.

O problema de programação em indústria de móveis foi estudado por Wilson, King e Hodgson (2004). Em cada estágio há várias máquinas paralelas idênticas. As heurísticas desenvolvidas baseiam-se em um algoritmo genético e demonstraram eficiência na minimização do tempo de *setup* independente da sequência e do *makespan*.

Low (2005) enfocou a minimização do tempo total de fluxo em problemas em que os estágios possuíam várias máquinas paralelas não relacionadas e tempos de remoção. Heurísticas construtivas para minimização do *makespan* foram apresentadas por Logendran *et al*. (2005). As tarefas foram agrupadas em famílias. Foram considerados os tempos de *setup* dependentes das máquinas e independentes das famílias de tarefas.

Fuchigami, Moccellin e Ruiz (2007) analisaram o desempenho de regras de prioridade em sistemas *flexible flow line*, um tipo de *flow shop* híbrido que permite que as tarefas saltem estágios. Os tempos de *setup* independentes da sequência podem ser antecipados ou não com determinada probabilidade.

Quatro heurísticas construtivas foram propostas por Morais e Moccellin (2010) para o problema de programação em *flow shop* híbrido com tempos de *setup* dependentes da sequência e minimização do tempo médio de fluxo. O melhor desempenho foi obtido pelo algoritmo que faz a ordenação inicial pela ordem não decrescente da soma dos tempos de processamento e *setup* de todos os estágios e a alocação sequencial de cada tarefa na máquina cuja data de término é a menor.

Pradenas et al. (2011) abordaram o problema de sequenciamento em uma fundição de cobre, modelada como um *flow shop* híbrido. O método solução primeiramente sequencia os lotes de produção e em seguida faz uma melhoria por meio do algoritmo *Simulated Annealing*.

Foram encontrados apenas dois trabalhos que tratam do problema de programação em *flow shops* paralelos proporcionais, porém ambos com apenas dois estágios, minimização do *makepan* e sem tempos de *setup* explícitos. Sundararaghavan, Kunnathur e Viswanathan (1997) propuseram duas heurísticas e métodos de simulação. E Al-Salem (2004) desenvolveu uma heurística multifase e procedimentos de simulação, comparando-os com o algoritmo de Sundararaghavan, Kunnathur e Viswanathan (1997).

2.2 *Flow shops* paralelos com tempos de *setup*

Como já salientado, o problema de programação em *flow shops* paralelos consiste em um caso específico de *flow shop* híbrido.

Sundararaghavan, Kunnathur e Viswanathan (1997) citaram como aplicação prática deste problema uma empresa de consultoria em engenharia que atende projetos de manufatura. No exemplo, foram considerados 10 projetos disponíveis no início de um mês, envolvendo sete estágios: levantamento dos dados, análise, desenvolvimento das especificações, estimação dos custos, elaboração da solução, preparação da documentação e apresentação do relatório final.

A execução de cada projeto requer a realização das sete atividades sequencialmente. Dois consultores estão disponíveis em cada estágio. Os tempos necessários para os consultores completarem o trabalho em cada etapa são diferentes. Somente um consultor é designado a cada estágio. O projeto termina quando todas as etapas são executadas. O problema consiste em determinar qual a qual das duas equipes de consultores cada um dos 10 projetos será designado e em que ordem. O objetivo é minimizar o tempo de execução dos 10 projetos. Analogamente, os consultores são modelados como “máquinas” e os projetos como “tarefas”.

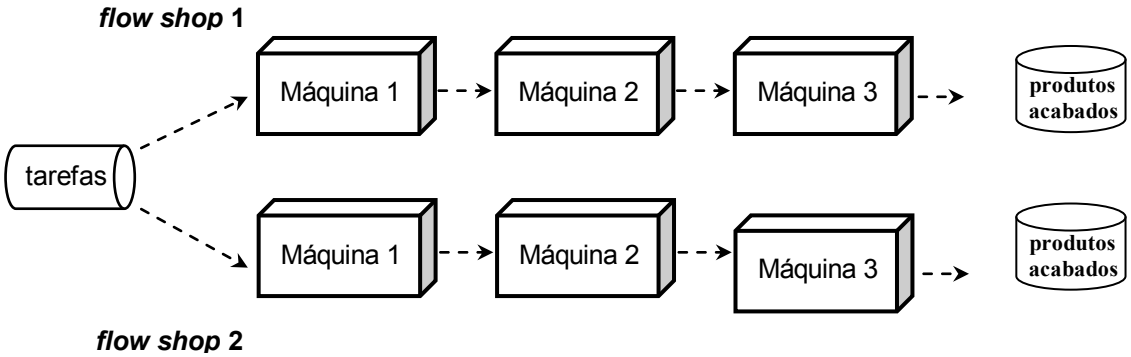
Como pode ser visto neste exemplo, os problemas de programação e sequenciamento da produção podem ser aplicados também a empresas de serviços,

não se restringindo aos problemas de fabricação de bens tangíveis, como indústrias de manufatura.

No mesmo trabalho, Sundararaghavan, Kunnathur e Viswanathan (1997) também citam que a ocorrência de *flow shops* paralelos é comum em indústrias onde novos equipamentos e tecnologias operam em paralelos com os já existentes. Esta situação é muito frequente na prática, pois na maioria dos casos a planta não é renovada por completo, mas com acréscimos periódicos de novas máquinas e processos.

A Figura 1 exemplifica o funcionamento de dois *flow shops* paralelos com três máquinas em cada estágio.

Figura 1 – Esquema ilustrativo de dois *flow shops* paralelos com três máquinas



As características do problema especificado nesta pesquisa são as seguintes: existe um conjunto de tarefas disponíveis para serem processadas e dois *flow shops* operando em paralelo, cada um consistindo de duas ou mais máquinas em série; ambos os *flow shops* possuem o mesmo número de máquinas; cada tarefa deve ser processada em exatamente uma máquina em cada estágio, seguindo um fluxo linear unidirecional; as tarefas podem ser processadas qualquer um dos dois *flow shops*, sendo o primeiro (*flow shop* 1) mais rápido e o segundo (*flow shop* 2), mais lento; em cada máquina, os tempos de *setup* são separados dos tempos de processamento das tarefas; os *flow shops* são proporcionais, ou seja, relativamente em cada estágio, os tempos de processamento e de *setup* no *flow shop* 2 são múltiplos dos do *flow shop* 1 por meio de um fator de proporcionalidade (α); uma vez programadas, as tarefas não podem trocar de *flow shop*. O objetivo é a minimização da duração total da programação ou *makespan*.

Mesmo para os tradicionais critérios de otimização e sem considerar os tempos de *setup*, o problema de programação em *flow* shop híbrido é classificado como *NP-hard* (KIS; PESCH, 2005; QUADT; KUHN, 2007). Assim, o tempo de computação de procedimentos de solução ótima para problemas de médio e grande porte constitui um fator limitador.

Segundo a conhecida notação de três campos e as adaptações de Vignier, Billaut e Proust (1999) para *flow shops* híbridos, o problema estudado pode ser representado por $\mathbf{2}\boldsymbol{Fm}|\boldsymbol{s_{jk}}|\boldsymbol{C}_{\max}$.

2.3 Tempos de *setup*

Em geral, problemas reais exigem um tempo de preparação ou *setup* entre a execução das tarefas. Muitas pesquisas em programação da produção desconsideram estes tempos ou então os incluem no tempo de processamento de cada tarefa. Isto simplifica a análise das aplicações, porém afeta diretamente a qualidade da solução para muitas situações que requerem o tratamento explícito do *setup* (ALLAHVERDI et al., 1999).

Os trabalhos que consideram os tempos de *setup* explícitos, ou seja, separados do tempo de processamento das tarefas, são divididos em dois grupos: com ***setup* independente** da sequência e com ***setup* dependente** da sequencia. Nos primeiros, o tempo de *setup* depende apenas da tarefa a ser processada. Já no segundo grupo de problemas, a duração do *setup* depende tanto da tarefa a ser processada quanto daquela que foi processada imediatamente antes na mesma máquina.

Alguns exemplos de aplicações com presença de tempos de *setup* dependentes da sequência são as indústrias de tinta e farmacêutica, em que os processos de limpeza e esterilização devem ser diferenciados dependendo da tarefa que foi feita e daquela que será processada em seguida. Processos que requerem ajuste de temperatura também requerem tempos de preparação diferentes dependendo da sequência de tarefas. Exceto estas situações peculiares, os demais problemas tratam o *setup* explícito como independente da sequência, como é o caso da indústria eletrônica, automotiva, madeireira, têxtil e de papel.

Outra classificação dos tempos de *setup* se refere à possibilidade da sua antecipação em relação à liberação da tarefa. O ***setup* antecipado** pode ser

realizado antes da liberação da tarefa, ou seja, antes do término da operação no estágio anterior. Uma importante implicação dos tempos de *setup* antecipados é que eles podem ser iniciados na máquina ou estágio subsequente enquanto a tarefa ainda está sendo executada (ALDOWAISAN, 2001). E como é comum haver tempo ocioso na segunda máquina em diante, antecipar o *setup* é uma vantagem para medidas de desempenho regulares, como é caso do *makespan* e do tempo de fluxo (ALLAHVERDI, 2000).

Por outro lado, o ***setup* não antecipado** pode ser realizado somente após a liberação da tarefa na máquina. São os casos que requerem que a peça ou o produto que será processado esteja presente na máquina para que as operações de preparação sejam realizadas como, por exemplo, ajustes e posicionamento.

Em relação aos trabalhos que consideram explicitamente os tempos de *setup*, Liaee e Emmons (1997) apresentaram uma classificação por critério de desempenho dos problemas de processamento de famílias de tarefas com tempos de *setup*. Allahverdi, Gupta e Aldowaisan (1999) fizeram uma revisão da literatura de problemas de programação da produção envolvendo tempos de *setup*. Os problemas foram classificados em *batch* e *non-batch*, e *setup* dependente e independente da sequência.

Zhu e Wilhelm (2006) publicaram uma revisão da literatura de diversas configurações de ambientes com tempos e custos de *setup* dependentes da sequência de execução das tarefas. E Allahverdi *et al.* (2008) atualizaram a revisão da literatura de problemas com tempos e custos de *setup*, classificando mais de 300 trabalhos publicados após o levantamento de Allahverdi, Gupta e Aldowaisan (1999).

Como já salientado, neste trabalho foram considerados os tempos de *setup* antecipados e independentes da sequência de execução das tarefas.

É importante ressaltar que não foi encontrado na literatura nenhum trabalho abordando *flow shops* paralelos com tempos de *setup* explíticos.

2.4 *Simulated Annealing*

O algoritmo *Simulated Annealing* genérico faz uma analogia com o fenômeno físico de recristalização de metais, particularmente do aço. O método *Annealing* consiste em aquecer o sistema a ser estudado a uma temperatura efetivamente alta

e, a partir deste ponto, resfriá-lo em estágios progressivos até que a energia interna seja tão baixa que não permita nenhuma alteração nas características do metal.

Proposto inicialmente por Kirkpatrick, Gelatt e Vecchi (1983), o algoritmo *Simulated Annealing* baseia-se no procedimento de Metrópolis, aplicado no estudo da mecânica dos sólidos por um modelo computacional.

Nesta técnica, um átomo do sistema sofre um pequeno deslocamento, escolhido aleatoriamente, representado por uma transição de energia. Se a variação energética for menor ou igual a zero, a nova configuração cristalina é aceita. Caso contrário, o movimento só será aceito mediante uma condição probabilística, ou seja, se a energia do sistema for maior que um valor aleatório, ou então o movimento é rejeitado e o método segue para a próxima iteração.

Segundo a analogia de Kirkpatrick, Gelatt e Vecchi (1983), substituindo a condição energética por uma função de custo, o procedimento de Metrópolis é perfeitamente capaz de gerar um conjunto de soluções de um problema de otimização em que a condição inicial de temperatura seria um parâmetro de controle.

O algoritmo *Simulated Annealing* genérico parte de uma solução inicial *S*, escolhe uma solução vizinha *S'* aplicando um operador apropriado na solução *S* e compara os custos das duas soluções. Se a nova solução *S'* tiver um custo menor, então o algoritmo aceita-a, substituindo *S* por *S'*. Caso contrário, aceita *S'* com a probabilidade $e^{-\Delta/T}$, onde Δ é a diferença entre os custos de *S* e *S'* e *T* é um parâmetro referido como *temperatura*. O papel da temperatura *T* é bastante relevante na execução do algoritmo. Inicialmente, *T* assume um valor pré-definido e decresce a cada iteração de acordo com a função denominada *resfriamento*. O algoritmo termina quando a temperatura atinge o valor zero ou próximo de zero.

O desempenho do *Simulated Annealing* também é influenciado por fatores como critério de parada, a escolha do espaço de soluções factíveis, a função objetivo e a estrutura da busca na vizinhança, chamada de esquema de perturbação.

Por ter proporcionado resultados bem sucedidos em uma ampla variedade de problemas, recentemente diversos pesquisadores têm utilizado o algoritmo *Simulated Annealing* como método de solução para programação da produção, desde o trabalho pioneiro de Osman e Potts (1989).

Recentemente, Hooda e Dhingra (2011) publicaram uma revisão da literatura sobre *Simulated Annealing* para *flow shop* com uma classificação baseada nos

vários critérios como parâmetro de seleção, utilização de recursos computacionais, hibridização e evolução dos métodos ao longo do tempo.

Alguns exemplos de pesquisas com o ambiente de máquinas paralelas aplicando *Simulated Annealing* são descritos a seguir. Behnamian, Zandieh e Fatemi Ghomi (2009) propuseram uma meta-heurística híbrida com três componentes: um método de geração de soluções iniciais baseado no algoritmo Colônia de Formigas, um *Simulated Annealing* para evolução da solução e um procedimento de busca em vizinhança variável (*variable neighborhood search – VNS*). O problema analisado possui *setup* dependente da sequência e o objetivo é a minimização do *makespan*.

Kolahan e Kayvanfar (2009) abordaram o problema multi-objetivo de programação em máquinas paralelas não relacionadas. A função de custo minimizada é composta pelo custo da máquina, penalidades por falta de pontualidade (atraso e adiantamento) e *makespan*. Chang e Chen (2011) estudaram o problema com *setup* dependente da sequência, propondo uma meta-heurística baseada no Algoritmo Genético e no *Simulated Annealing*.

Três variações do algorimo *Simulated Annealing* para o problema de máquinas paralelas proporcionais com minimização do *makespan* foram apresentadas por Senthilkumar e Narayanan (2011). Os autores utilizaram a Análise de Variância (ANOVA) para comparar a qualidade das soluções.

O algoritmo *Simulated Anealing* também já foi extensivamente utilizado na programação de *flow shops* híbridos. Janiak, Kozan, Lichtenstein e Oğuz (2007) analisaram o problema com datas de liberação e prazos de entrega com o objetivo de minimizar a soma ponderada do adiantamento, atraso e tempo de espera. Foram desenvolvidos três heurísticas construtivas e três meta-heurísticas baseadas nos algoritmos Busca Tabu e *Simulated Annealing*.

Syam e Al-Harkan (2010) compararam três meta-heurísticas (Algoritmo Genético, *Simulated Annealing* e Busca Tabu) para o problema de minimização do *makespan* no *flow shop* híbrido com duas máquinas paralelas em cada estágio. Wang, Chou e Wu (2011) propuseram um algoritmo *Simulated Annealing* para minimização do *makespan* em *flow shop* híbrido baseado em três métodos: programação em lista, programação permutacional e método *first-fit*.

Em relação aos problemas com tempos de *setup* dependentes da sequência, Jungwattanakit *et al.* (2007) estudaram o *flow shop* híbrido com máquinas paralelas

não relacionadas visando a minimização da soma ponderada do *makespan* e do número de tarefas atrasadas.

Dois trabalhos relativamente recentes abordaram a programação em *flow shop* híbrido com máquinas paralelas idênticas, *setup* dependente e minimização do *makespan*: Tabriz, Zandieh e Vaziri (2009) e Mirsanei *et al.* (2011). Ambos comparam os resultados do *Simulated Annealing* com o método *Random Key Genetic Algorithm* (*RKGA*), mas o segundo confronta também com o Algoritmo Imunológico.

As publicações mais recentes encontradas abordando *flow shop* híbrido de *Simulatad Annealing* são as descritas a seguir, porém ambas não consideram explicitamente os tempos de *setup*. Jolai *et al.* (2013) abordaram o problema *no-wait* (restrição de não haver espera entre as operações de cada tarefa) com apenas dois estágios e bi-objetivo de minimização do *makespan* e do atraso máximo. E Dai *et al.* (2013) estudaram o *trade-off* existente entre o *makespan* e o consumo de energia e propuseram um método híbrido genético-*simulated annealing*.

Não foi encontrado nenhum trabalho que aplica o algoritmo *Simulated Annealing* ao problema de programação em dois *flow shops* paralelos.

3 ALGORITMO *SIMULATED ANNEALING* PROPOSTO

Esta seção apresenta o algoritmo *Simulated Anneling* proposto para minimização do *makespan* em dois *flow shops* paralelos proporcionais com tempos de *setup*, desenvolvido com base nas ideias de Senthilkumar e Narayanan (2011).

O método proposto utiliza um Algoritmo para Geração de Sequência Inicial, elaborado a partir de ideias de Low (2005) e descrito a seguir.

Seja o conjunto $J=\{1, ..., n\}$ de n tarefas, cada uma a ser programada em apenas um dos dois *flow shops*, compostos pelo conjunto $M=\{1, ..., m\}$ de m máquinas em série. Considere os tempos de processamento do *flow shop* 1, p_{jk}, e de *setup*, s_{jk}, de cada tarefa j em cada máquina k, com $j \in J$ e $k \in M$, e analogamente os tempos de processamento, αp_{jk}, e de *setup*, αs_{jk}, do *flow shop* 2.

ALGORITMO PARA GERAÇÃO DE SEQUÊNCIA INICIAL

PASSO 1

Determine os tempos de processamento modificados $P_j^{'}$ para cada tarefa *j*:

$P_j^{'} = \sum_{k \in M} P_{jk}^{'}$, com $j \in J$,

onde $P_{jk}^{'} = S_{jk} + P_{jk} - S_{j(k+1)}$, para $j \in J, k \in M \backslash \{m\}$,

$P_{jm}^{'} = P_{jm}$, com $j \in J$,

$P_{jk} = (p_{jk} + \alpha\, p_{jk})/2$, para $j \in J, k \in M$,

$S_{jk} = (s_{jk} + \alpha\, s_{jk})/2$, para $j \in J, k \in M$.

PASSO 2

Ordene as tarefas de acordo com a regra *SPT (Shortest Processing Time)*, ou seja, em ordem não-decrescente dos tempos $P_j^{'}$.

PASSO 3

Para cada uma das tarefas da sequência obtida:

3.1. Associe a tarefa a cada um dos dois *flow shops*.

3.2. Calcule o *makespan* parcial da subsequência de cada *flow shop*, considerando as tarefas já programadas e a tarefa que está sendo associada (em todas as máquinas do *flow shop*).

3.3. Programe a tarefa associada no *flow shop* com o menor *makespan* parcial.

ESQUEMAS DE PERTURBAÇÃO

O algoritmo *Simulated Annealing* também requer a definição de esquemas de perturbação ou busca na vizinhança, que foram adaptados de Nearchou (2004) para o problema tratado e são apresentados a seguir. Em todos os esquemas, a escolha do *flow shop* em que haverá a perturbação é feita aleatoriamente.

- **PS1:** troca de duas tarefas adjacentes
  Seleciona aleatoriamente uma posição da sequência e em seguida troca a tarefa da posição selecionada com a tarefa da posição imediatamente posterior.
- **PS2:** Troca de duas tarefas aleatórias

Seleciona aleatoriamente duas posições da sequência e em seguida troca as tarefas das posições selecionadas.

- **PS3:** Deslocamento de uma tarefa
  Seleciona aleatoriamente duas posições da sequência e em seguida remove a tarefa da primeira posição e insere-a na segunda, deslocando as tarefas intermediárias uma posição (à esquerda ou à direita, dependendo da localização da segunda posição selecionada).

- **PS4:** Deslocamento de uma subsequência de tarefas
  Seleciona aleatoriamente uma subsequência com tamanho também aleatório, de 2 a n_f–1, onde n_f é o número de tarefas alocadas no *flow shop* selecionado aleatoriamente para a perturbação. Em seguida, seleciona uma nova posição onde iniciará a subsequência selecionada.

- **PS5:** Inversão de uma subsequência de tarefas
  Seleciona aleatoriamente uma subsequência com tamanho também aleatório, de 2 a n_f, onde n_f é o número de tarefas alocadas no *flow shop* selecionado aleatoriamente para a perturbação. Em seguida, inverte a ordem das tarefas da subsequência.

- **PS6:** Deslocamento ou inversão de uma subsequência de tarefas
  Seleciona aleatoriamente a opção “deslocamento” ou “inversão”. Se o “deslocamento” for selecionado, executa o esquema PS4 ou, se a “inversão” for selecionada, procede-se o esquema PS5.

Com base nestas definições, foi desenvolvida a seguinte meta-heurística que, aplicando cada um dos esquemas de perturbação descritos, origina seis algoritmos diferentes.

ALGORITMO SIMULATED ANNEALING PARA O PROBLEMA $2Fm|s_{jk}|C_{max}$

PASSO 1

Inicialize os seguintes dados:

- Número de tarefas (n)
- Número de máquinas em cada *flow shop* (m)

- Tempos de processamento de cada tarefa *j* em cada máquina *k* do *flow shop* 1 (p_{jk})
- Tempos de processamento de cada tarefa *j* em cada máquina *k* do *flow shop* 2 (αp_{jk})
- Tempos de *setup* para cada tarefa *j* em cada máquina *k* do *flow shop* 1 (s_{jk})
- Tempos de *setup* para cada tarefa *j* em cada máquina *k* do *flow shop* 2 (αs_{jk})
- Fator de proporcionalidade dos tempos de processamento e de *setup* no *flow shop* 2 (α)
- Temperatura inicial (T_i)
- Temperatura final (T_f)
- Fator de resfriamento (r)

PASSO 2

Obtenha uma solução inicial (S) por meio do Algoritmo para Geração de Solução Inicial.

Calcule o *makespan* desta sequência [$C_{\max}(S)$].

Faça $S^* = S$ (onde S^* é a melhor sequência encontrada).

Faça $T = T_i$.

PASSO 3

Utilizando um esquema de perturbação, selecione uma solução S' na vizinhança de S.

Calcule $\Delta = C_{max}(S') - C_{max}(S)$

Se $\Delta \leq 0$ (função objetivo melhora), então faça $S = S'$, e se $C_{max}(S) < C_{max}(S^*)$, faça $S^* = S$.

Senão ($\Delta > 0$), gere um número aleatório u. Se $u \leq e^{-\Delta/T}$, então faça $S = S'$.

PASSO 4

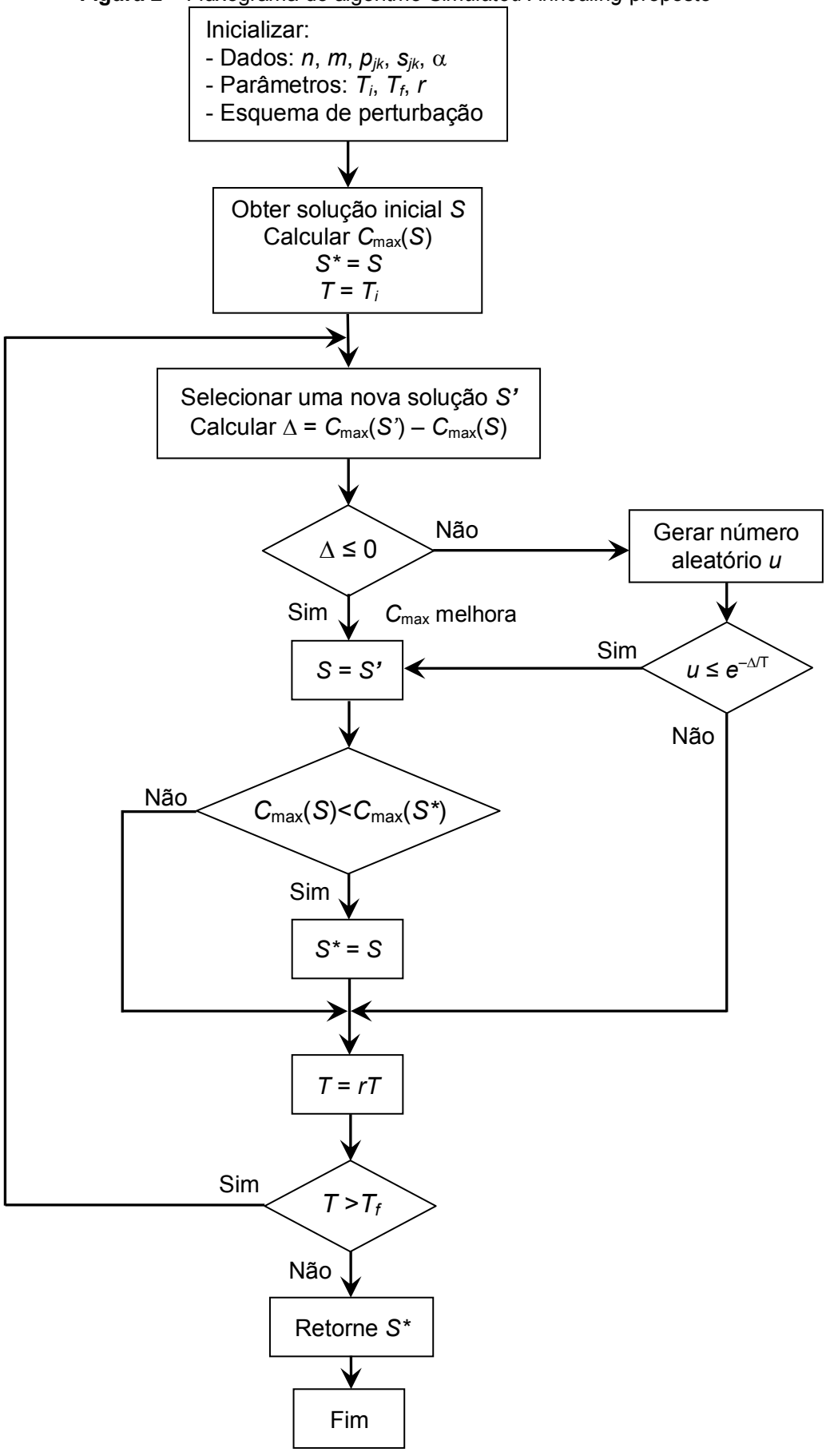
Faça $T = rT$ (resfriamento).

Se $T > T_f$, então vá para o *PASSO 3*.

Senão, retorne S^*. *FIM*.

O fluxogramada da Figura 2 auxilia na melhor compreensão do funcionamento da meta-heurística proposta.

Figura 2 – Fluxograma do algoritmo *Simulated Annealing* proposto



4 EXPERIMENTAÇÃO COMPUTACIONAL E RESULTADOS

4.1 Método da pesquisa

Segundo as definições de Martins (2010, p.45-49) e Nakano (2010, p.64), esta pesquisa possui *abordagem quantitativa*, pois preocupa-se com 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 pela medida de desempenho *makespan*).

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 melhoria teórico-prática de sistemas, processos e produtos, e 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.

4.2 Planejamento do experimento

Os parâmetros da experimentação computacional foram definidos com base em trabalhos publicados na literatura e organizados conforme a Tabela 1 a seguir. Tal como Al-Salem (2004), foram gerados problemas com 5, 10, 15, 20, 40, 60, 80 e 100 tarefas. O número de máquinas em cada *flow shop* foi 2, 3, 5 ou 8. Os tempos de processamento no *flow shop* 1 seguiram a distribuição uniforme no intervalo $U[1,99]$ e os tempos de *setup* em $U[0,99]$.

Tabela 1 – Parâmetros da experimentação computacional

Parâmetro	*Valores*
número de tarefas	5, 10, 15, 20, 40, 60, 80, 100
número de máquinas em cada *flow shop*	2, 3, 5, 8
fator de proporcionalidade α	1, 1.15, 1.5, 1.75, 2
intervalo dos tempos de processamento no *flow shop* 1	*U*[1,99]
intervalo dos tempos de *setup* no *flow shop* 1	*U*[0,99]

A combinação do número de tarefas e de máquinas totaliza 32 problemas analisados. Cada um deles foi resolvido utilizando cada uma das cinco opções do fator α: 1, 1.15, 1.5, 1.75 e 2. Esses valores também foram baseados em Al-Salem (2004). Assim como Balakrishnan, Kanet e Sridharan (1999), optou-se por utilizar o mesmo fator de proporcionalidade para os tempos de processamento e de *setup*.

Os parâmetros do *Simulated Annealing* utilizados foram baseados em Senthilkumar e Narayanan (2011): temperatura inicial T_i = 60, fator de resfriamento r = 0.85 e temperatura final T_f = 0.01.

Foi utilizado o sistema operacional Windows e o ambiente de programação Delphi. As configurações da máquina são as seguintes: processador AMD Turion com 1.8 GHz de frequência e 512 MB de memória RAM.

4.3 Análise dos resultados

Na análise dos resultados foi utilizada a mesma metodologia de Senthilkumar e Narayanan (2011), a Análise de Variância (ANOVA) com nível de significância de 0.05. Foram considerados os seguintes fatores e seus respectivos testes de hipóteses:

Fator: esquema de perturbação

H_0: Não há diferença significativa entre os esquemas de perturbação (PS1, PS2, PS3, PS4, PS5 e PS6) em termos do *makespan*.

H_a: Existe diferença significativa entre os esquemas de perturbação (PS1, PS2, PS3, PS4, PS5 e PS6), para pelo menos um par de algoritmos, em termos do *makespan*.

Fator: alfa (fator de proporcionalidade dos tempos de processamento e de setup)

H_0: Não há diferença significativa entre as opções do fator alfa (1, 1.15, 1.5, 1.75 e 2) em termos do *makespan*.

H_a: Existe diferença significativa entre opções do fator alfa (1, 1.15, 1.5, 1.75 e 2), para pelo menos um par de fatores alfa, em termos do *makespan*.

Os resultados de cada tabela ANOVA são apresentados nas Tabelas 2 e 3 a seguir.

Tabela 2 – Resultados da tabela ANOVA para o fator esquema de perturbação

Fonte de variação	***SQ***	***GL***	***MQ***	***F (calculado)***	***F crítico (α=0.05)***
Entre perturbações	169644,0	5	33928,8	0,007355	2,223485
Erro	4400562824,6	954	4612749,3		
Total	4400732468,7	959			

Tabela 3 – Resultados da tabela ANOVA para o fator alfa

Fonte de variação	***SQ***	***GL***	***MQ***	***F (calculado)***	***F crítico (α=0.05)***
Entre valores alfa	64808419,3	4	16202104,8	3,568561	2,381251
Erro	4335924049,3	955	4540234,6		
Total	4400732468,7	959			

Para o fator esquema de perturbação, a estatística teste *F* foi 0.007355, que é menor que o valor *F* para o nível de significância de 0.05. Daqui, infere-se que a hipótese nula restrita a este fator é aceita, ou seja, não há diferença significativa entre os esquemas de perturbação em termos do *makespan*.

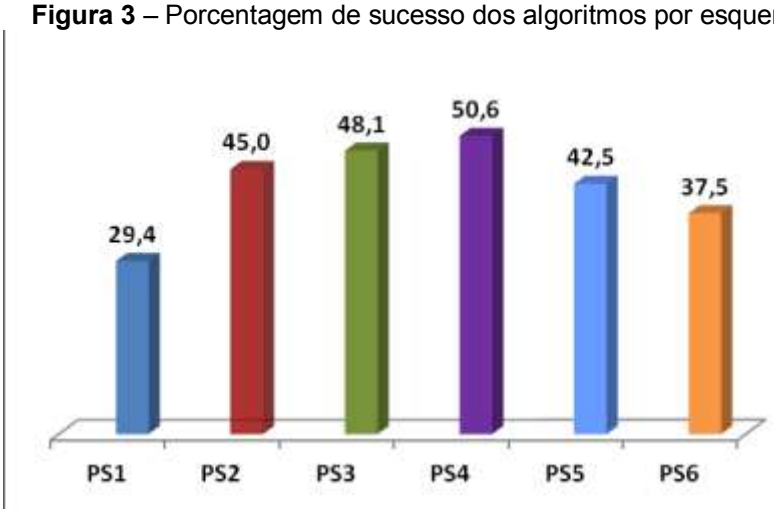
Já para o fator alfa de proporcionalidade dos tempos de processamento e de *setup*, a estatística teste *F* foi 3.568561, maior que o valor *F* para o nível de significância de 0.05. Assim, infere-se que a hipótese nula restrita a este fator é rejeitada, indicando que existe diferença significativa entre os resultados para cada valor de alfa em termos do *makespan*. É importante observar que o fator de proporcionalidade depende da característica do ambiente de produção, especificamente das máquinas de cada *flow shop*, e não dos algoritmos.

Para uma análise mais detalhada dos resultados dos algoritmos, foi considerada também a porcentagem de sucesso, ou seja, o número de vezes que o

algoritmo forneceu a melhor solução, empatando ou não, dividido pelo número de problemas analisados, em porcentagem.

O gráfico da Figura 3 apresenta a comparação global da porcentagem de sucesso para cada esquema de perturbação, agregando-se os resultados dos outros parâmetros (número de tarefas e de máquinas).

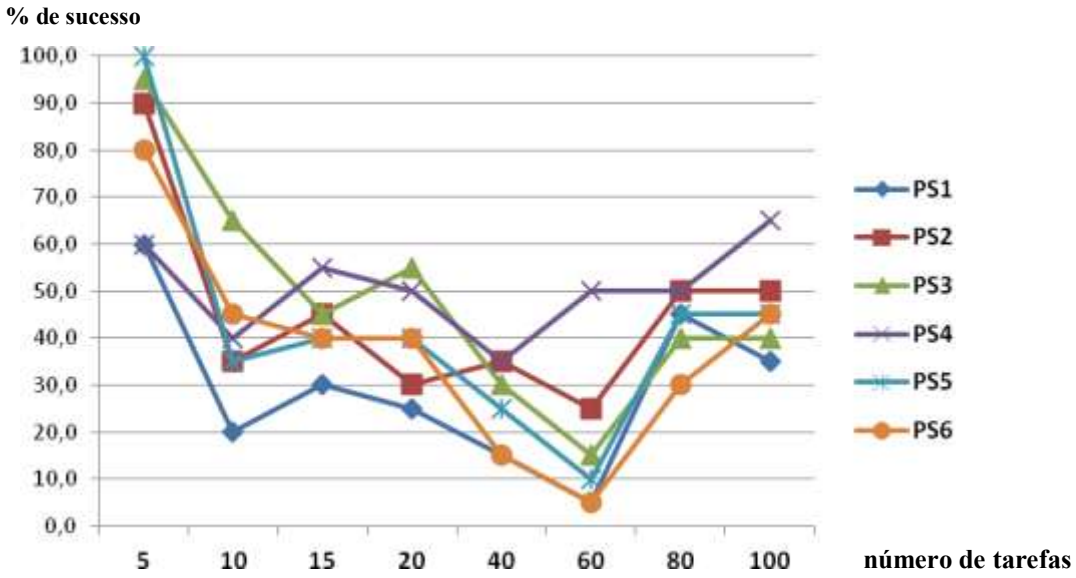
Figura 3 – Porcentagem de sucesso dos algoritmos por esquema de perturbação



Conforme pode ser visto, o esquema PS4, que faz o deslocamento de uma subsequência de tarefas, forneceu o melhor resultado, com 50,6% de sucesso. O desempenho dos demais esquemas de perturbação ficou na faixa de 29,4 a 48,1% de sucesso. Entretanto, pelo resultado da tabela ANOVA, essa diferença de desempenho entre os esquemas não chega a ser relevante. Sugere-se portanto a resolução do problema tratado utilizando-se o esquema PS4.

Em um nível de detalhamento ainda maior, o gráfico da Figura 4 mostra o desempenho de cada perturbação para cada opção do porte de problema (em número de tarefas). Nota-se a grande instabilidade dos resultados com a variação do número de tarefas.

Figura 4 – Porcentagem de sucesso dos algoritmos por número de tarefas



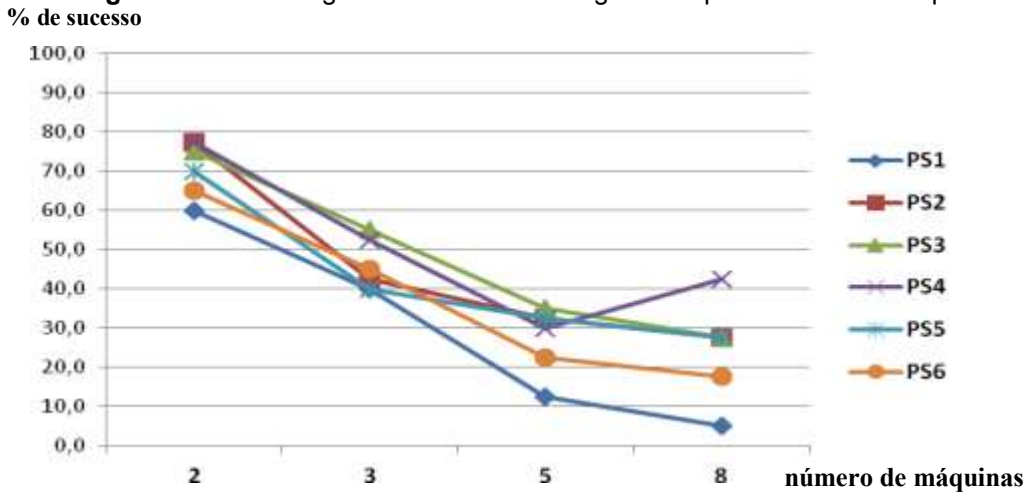
Para problemas de pequeno porte, com 5 tarefas, todos os esquemas obtiveram melhores resultados. Em muitos casos, os valores do *makespan* foram iguais para mais de um esquema de perturbação. Isto pode indicar que os esquemas são relativamente eficazes para problemas com 5 tarefas, porém requerem procedimentos mais robustos para problemas de maior porte, que demonstraram maior variabilidade do *makespan*.

Em problemas com 60 tarefas, os métodos obtiveram resultados inferiores, exceto o PS4. Para as outras opções de número de tarefas, os resultados ficaram relativamente constantes, na faixa de 20% a 60% de sucesso.

Nesta análise por número de tarefas, nem sempre o desempenho do melhor esquema de perturbação na análise global prevaleceu. Há casos em que o PS3 e o PS5 obtiveram os melhores resultados, respectivamente com 10 e 20 tarefas e com 5 tarefas. Além disso, em problemas com 40 e 80 tarefas, houve um empate entre os esquemas PS2 e PS4. Isto comprova a importância da análise detalhada por parâmetro.

A Figura 5 apresenta a comparação dos resultados dos esquemas de perturbação para cada opção do número de máquinas nos *flow shops*.

Figura 5 – Porcentagem de sucesso dos algoritmos por número de máquinas

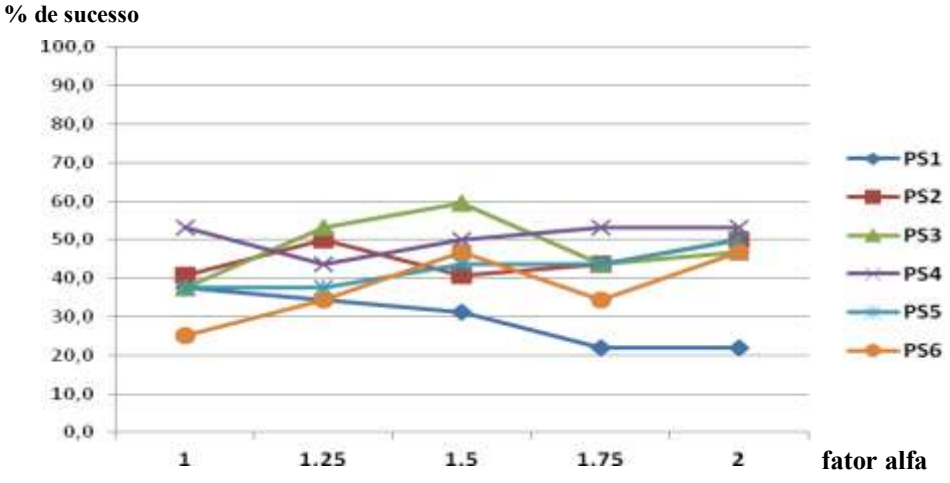


Existe um claro decrescimento do desempenho dos métodos com o aumento do número de máquinas, indo de cerca de 70% de sucesso, com duas máquinas, para em torno de 30%, em problemas com 8 máquinas. É interessante observar o comportamento uniforme de todos os esquemas de perturbação, com exceção do PS4, que melhora o seu desempenho quando passa de 5 para 8 máquinas.

A maior amplitude na diferença de desempenho dos métodos foi verificada em problemas com 8 máquinas. Novamente, houve certo revezamento do melhor esquema de perturbação.

O gráfico da Figura 6 mostra os resultados de cada esquema de perturbação para cada opção do fator alfa de proporcionalidade dos tempos de processamento e de *setup* entre os *flow shops*.

Figura 6 – Porcentagem de sucesso dos algoritmos por fator de proporcionalidade alfa



O desempenho de cada método para os diferentes valores do fator alfa ficou relativamente constante, na faixa próxima de 25% a 55% de sucesso. O esquema PS4 forneceu os melhores resultados para alfa igual a 1, 1.75 e 2, enquanto o PS3 mostrou-se superior nos casos em que alfa é igual a 1.25 e 1.5.

O tempo médio de CPU despendido na resolução dos problemas foi de 76 ms, o que não compromete a eficiência computacional dos algoritmos.

Em resumo, os resultados da análise de variância da experimentação computacional mostraram que a diferença de desempenho dos seis esquemas de perturbação não é relevante em relação ao *makespan*. Já em relação à porcentagem de sucesso, o esquema de perturbação PS4, que desloca uma subsequência de tarefas na busca por uma solução vizinha, obteve o melhor resultado, com mais de 50% de sucesso.

Como esperado, verificou-se que é significativa a diferença entre os resultados dos algoritmos para cada valor do fator de proporcionalidade dos tempos de processamento e de *setup* dos *flow shops*. Além disso, foi demonstrada a importância da análise detalhada por parâmetro do problema (número de tarefas, de máquinas e fator alfa) por meio da porcentagem de sucesso, revelando os melhores esquemas de perturbação para cada opção específica dos valores desses parâmetros.

5 CONSIDERAÇÕES FINAIS

O objetivo desta pesquisa foi atingido com a proposição e implementação computacional de seis variações do algoritmo *Simulated Annealing* com seus respectivos esquemas de perturbação, ou busca na vizinhança, para o problema tratado, incluindo um algoritmo de geração de sequência inicial.

Os algoritmos se mostraram eficazes quanto a solução e eficientes computacionalmente.

Os resultados desta pesquisa possuem relevância tanto teórica como prática. Sua importância teórica reside na proposição de novos métodos de solução para o problema tratado que, como mostrou o exame da literatura, tem sido pouco estudado. Podem ainda servir como ponto de partida para novas pesquisas e análises em problemas correlacionados.

Já os resultados práticos podem contribuir para melhorias nos processos produtivos em ambientes industriais compatíveis com o problema tratado, que realisticamente são bastante complexos.

Consistem limitações desta pesquisa a indisponibilidade dados reais para análise de desempenho e a necessidade de novos métodos de comparação dos resultados, como por exemplo, a definição de limitantes inferiores do *makespan* (*lower bounds*).

Visando o desenvolvimento de futuros trabalhos, sugere-se a criação de novos esquemas de perturbação e algoritmos de busca local, além da utilização de outras medidas de desempenho, como o tempo médio de fluxo e o desvio de pontualidade (*lateness*). Pode-se também implementar outras meta-heurísticas, como o algoritmo genético e a busca tabu, para comparar seus resultados com os desta pesquisa.

Nota:
Os resultados parciais desta pesquisa foram apresentados durante o XLIII Simpósio Brasileiro de Pesquisa Operacional.

REFERÊNCIAS

ALDOWAISAN, T. A new heuristic and dominance relations for no-wait flowshop with setups. **Computer and Operations Research**, v. 28, p. 563-584, 2001. http://dx.doi.org/10.1016/S0305-0548(99)00136-7

ALLAHVERDI, A. Minimizing mean flowtime in a two-machine flowshop with sequence-independent setup times. **Computers and Operations Research**, v. 27, p. 111-127, 2000. http://dx.doi.org/10.1016/S0305-0548(99)00010-6

ALLAHVERDI, A.; GUPTA; J.N.D.; ALDOWAISAN, T. A review of scheduling research involving Setup considerations. **Omega – The International Journal of Management Science**, v. 27, p. 219-239, 1999.

ALLAHVERDI, A.; NG, C.T.; CHENG, T.C.E.; KOVALYOV, M.Y. A survey of scheduling problems with setup times or costs. **European Journal of Operational Research**, v. 187, p. 985-1032, 2008. http://dx.doi.org/10.1016/j.ejor.2006.06.060

ALLAOUI, H., ARTIBA, A. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. **Computers and Industrial Engineering**, v. 47, p. 431-450, 2004. http://dx.doi.org/10.1016/j.cie.2004.09.002

AL-SALEM, A. A heuristic to Minimize Makespan in Proportional Parallel Flow Shops. **International Journal of Computing & Information Sciences**, v. 2, p. 98-107, 2004.

BALAKRISHNAN, N.; KANET, J.J.; SRIDHARAN, S.V. Early/Tardy scheduling with sequence dependent setups on uniform parallel machines. **Computers & Operations Research**, v. 26, p. 127-141, 1999. http://dx.doi.org/10.1016/S0305-0548(98)00051-3

BEHNAMIAN, J.; ZANDIEH, M.; FATEMI GHOMI, S.M.T. Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. **Expert Systems with Applications**, v. 36, p. 9637-9644, 2009. http://dx.doi.org/10.1016/j.eswa.2008.10.007

BOTTA-GENOULAZ, V. Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. **International Journal of Production Economics**, v.64, n.1-3, p.101-111, 2000. http://dx.doi.org/10.1016/S0925-5273(99)00048-1

CHANG, P.-C.; CHEN, S.-H. Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times. **Applied Soft Computing**, v. 11, p. 1263-1274, 2011. http://dx.doi.org/10.1016/j.asoc.2010.03.003

DAI, M.; TANG, D.; GIRET, A.; SALIDO, M.A.; LI, W.D. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. **Robotic and Computer-Integrated Manufacturing**, v. 29, p. 418-429, 2013. http://dx.doi.org/10.1016/j.rcim.2013.04.001

FUCHIGAMI, H.Y.; MOCCELLIN, J.V.; RUIZ, R. Análise comparativa do desempenho de regras de prioridade em sistemas *flexible flow line* com múltiplas máquinas e tempos de *setup* independentes da sequência. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 39, 2007, Fortaleza. **Anais...** Rio de Janeiro: SOBRAPO, 2007.

GUPTA, J.N.D., TUNC, E.A. Scheduling a two-stage hybrid flowshop with separable setup and removal times. **European Journal of Operational Research**, v. 77, p. 415-428, 1994. http://dx.doi.org/10.1016/0377-2217(94)90407-3

HOODA, N.; DHINGRA, A.K. Flow shop scheduling using Simulated Annealing: a review. **International Journal of Applied Engineering Research**, v. 2, n. 1, p. 234-249, 2011.

HUANG, W.; LI, S. A two-stage hybrid flowshop with uniform machines and setup times. **Mathematical and Computer Modeling**, v. 27, n. 2, p. 27-45, 1998. http://dx.doi.org/10.1016/S0895-7177(97)00258-6

JANIAK, A.; KOZAN, E.; LICHTENSTEIN, M.; OĞUZ, C. Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. **International Journal of Production Economics**, v. 105, p. 407-424, 2007. http://dx.doi.org/10.1016/j.ijpe.2004.05.027

JOLAI, F.; ASEFI, H.; RABIEE, M.; RAMEZANI, P. Bi-objective simulated annealing approaches for no-wait two-stage flexible flow shop scheduling problem. **Sciencia Iranica**, v. 20, p. 861-872, 2013.

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

JUNGWATTANAKIT, J.; REODECHA, M.; CHAOVALITWONGSE, P.; WERNER, F. Constructive and simulated annealing algorithms for hybrid flow shop problems with unrelated parallel machines. **Tammasat International Journal of Science and Technology**, v. 12, n. 1, p. 31-41, 2007.

KIRKPATRICK, S.; GELATT, C.D.; VECCHI, M.P. Optimization by Simulated Annealing. **Science**, v. 220, p. 671-680, 1983. http://dx.doi.org/10.1126/science.220.4598.671

KIS, T.; PESCH, E. A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. **European Journal of Operational Research**, v. 164, p. 592-608, 2005. http://dx.doi.org/10.1016/j.ejor.2003.12.026

KOLAHAN, F.; KAYVANFAN, V. A heuristic Algorithm Approach for Scheduling of Multi-criteria Unrelated Parallel Machines. **World Academy of Science, Engineering and Technology**, v. 59, p. 253-256, 2009.

KURZ, M.E.; ASKIN, R.G. Comparing scheduling rules for flexible flow lines. **International Journal of Production Economics**, v. 85, p. 371-388, 2003. http://dx.doi.org/10.1016/S0925-5273(03)00123-3

KURZ, M.E.; ASKIN, R.G. Scheduling flexible flow lines with sequence-dependent setup times. **International Journal of Operational Research**, v. 159, p. 66-82. 2004. http://dx.doi.org/10.1016/S0377-2217(03)00401-6

LI, S. A hybrid two-stage flowshop with part family, batch production, and major and minor set-ups. **European Journal of Operational Research**, v. 102, p. 142-156, 1997. http://dx.doi.org/10.1016/S0377-2217(96)00213-5

LIAEE, M.M.; EMMONS, H. Scheduling families of jobs with setup times. **International Journal of Production Economics**, v. 51, n. 3, p. 165-176, 1997. http://dx.doi.org/10.1016/S0925-5273(96)00105-3

LIN, H.T.; LIAO, C.J. A case study in a two-stage hybrid flow shop with setup time and dedicated machines. **International Journal of Production Economics**, v. 86, n. 2, p. 133-143, 2003. http://dx.doi.org/10.1016/S0925-5273(03)00011-2

LINN, R.; ZHANG, W. Hybrid flow shop scheduling: a survey. **Computers & Industrial Engineering**, v. 37, n. 1-2, p. 57-61, 1999. http://dx.doi.org/10.1016/S0360-8352(99)00023-6

LOGENDRAN, R.; CARSON, S.; HANSON, E. Group scheduling in flexible flow shops. **International Journal of Production Economics**, v. 96, p. 143-155, 2005. http://dx.doi.org/10.1016/j.ijpe.2004.03.011

LOW, C. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. **Computers & Operations Research**, v. 32, p. 2013-2025, 2005. http://dx.doi.org/10.1016/j.cor.2004.01.003

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.

MIRSANEI, H.S.; ZANDIEH, M.; MOAYED, M.J.; KHABBAZI, M.R. A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. **Journal of Intelligent Manufacturing**, v. 22, p. 965-978, 2011. http://dx.doi.org/10.1007/s10845-009-0373-8

MORAIS, M.F.; MOCCELLIN, J.V. Métodos heurísticos construtivos para redução do estoque em processo em ambientes de produção *flow shop* híbridos com tempos de *setup* dependentes da sequência. **Gestão & Produção**, v. 17, n. 2, p. 367-375, 2010. http://dx.doi.org/10.1590/S0104-530X2010000200011

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.

NEARCHOU, A.C. A novel metaheuristic approach for the flow shop scheduling problem. **Engineering Applications of Artificial Intelligence**, v. 17, p. 289-300, 2004. http://dx.doi.org/10.1016/j.engappai.2004.02.008

OSMAN, I.H.; POTTS, C.N. Simulated Annealing for Permutation Flow-Shop Scheduling. **Omega**, v. 17, p. 551-557, 1989. http://dx.doi.org/10.1016/0305-0483(89)90059-5

PRADENAS, L.; CAMPOS, A.; SALDAÑA, J.; PARADA, V. Scheduling copper refining and casting operations by means of heuristics for the flexible flow shop problem. **Pesquisa Operacional**, v. 31, n. 3, p. 443-457, 2011. http://dx.doi.org/10.1590/S0101-74382011000300002

QUADT, D.; KUHN, H. A taxonomy of flexible flow line scheduling procedures, **European Journal of Operational Research**, v. 178, p. 686-698, 2007. http://dx.doi.org/10.1016/j.ejor.2006.01.042

RIBAS, I.; LEISTEN, R.; FRAMIÑAN, J.M. Review and classifications of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. **Computers & Operations Research**, v. 37, p. 1439-1454, 2010.http://dx.doi.org/10.1016/j.cor.2009.11.001

RUIZ, R.; MAROTO, C. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. **European Journal of Operational Research**, v. 169, p. 781-800, 2006. http://dx.doi.org/10.1016/j.ejor.2004.06.038

RUIZ, R.; ŞERIFOĞLU, F.S.; URLINGS, T. Modeling realistic hybrid flexible flowshop scheduling problems. **Computers & Operations Research**, v. 35, p. 1151-1175, 2008. http://dx.doi.org/10.1016/j.cor.2006.07.014

RUIZ, R.; VÁZQUEZ-RODRÍGUEZ, J.A. The hybrid flow shop scheduling problem. **European Journal of Operational Research**, v. 205, p. 1-18, 2010. http://dx.doi.org/10.1016/j.ejor.2009.09.024

SENTHILKUMAR, P.; NARAYANAN, S. Simulated annealing algorithm to minimize makespan in single machine problem with uniform parallel machines. **Intelligent Information Management**, v. 3, p. 22-31, 2011. http://dx.doi.org/10.4236/iim.2011.31003

SUNDARARAGHAVAN, P.S.; KUNNATHUR, A.S.; VISWANATHAN, I. Minimizing makespan in parallel flowshops. **Journal of the Operational Research Society**, v. 48, p. 834-842, 1997. http://dx.doi.org/10.2307/3010711

SYAM, W.P.; AL-HARKAN, I.M. Comparison of three meta heuristics to optimize hybrid flow shop scheduling problem with parallel machines. **World Academy of Science, Engineering and Technology**, v. 62, p. 271-278, 2010.

TABRIZ, A.A.; ZANDIEH, M.; VAZIRI, Z. A novel simulated annealing algorithm to hybrid flow shops scheduling with sequence-dependent setup times. **Journal of Applied Sciences**, v. 9, n. 10, p. 1943-1949, 2009. http://dx.doi.org/10.3923/jas.2009.1943.1949

VIGNIER, A.; BILLAUT, J.C.; PROUST, C. Les problèmes d'ordonnancement de type flow-shop hybride: état de l'art. **RAIRO – Recherche Opérationnelle**, v. 33, n. 2, p. 117-183, 1999.

WANG, H. Flexible flow shop scheduling: optimum, heuristic and artificial intelligence solutions. **Expert Systems**, v. 22, n. 2, p. 78-85, 2005. http://dx.doi.org/10.1111/j.1468-0394.2005.00297.x

WANG, H.-M.; CHOU, F.-D.; WU, F.-C. A simulated annealing for hybrid flow shop scheduling with multiprocessor tasks to minimize makespan. **International Journal of Advanced Manufacturing Technology**, v. 53, n. 5-8, p. 761-776, 2011.http://dx.doi.org/10.1007/s00170-010-2868-z

WILSON, A.D.; KING, R.E.; HODGSON, T.J. Scheduling non-similar groups on a flow line: multiple group setups. **Robotics and Computer-Integrated Manufacturing**, v. 20, p. 505-515, 2004. http://dx.doi.org/10.1016/j.rcim.2004.07.002

ZHU, X.; WILHELM, W.E. Scheduling and lot sizing with sequence-dependent setup: a literature review. **IIE Transactions**, v. 38, p. 987-1007, 2006. http://dx.doi.org/10.1080/07408170600559706

Artigo recebido em 01/07/2013 e aceito para publicação em 18/06/2014
DOI: http://dx.doi.org/ 10.14488/1676-1901.v14i3.1631