Brazilian Journal of Development

Uma proposta de heurísticas e meta-heurísticas aplicadas ao problema de *flow shop scheduling*

A proposal of heuristics and metaheuristics for solving the *flow shop scheduling* problem

DOI:10.34117/bjdv6n6-390

Recebimento dos originais: 11/05/2020
Aceitação para publicação: 17/06/2020

Diego Moah Lobato Tavares
Mestre em Engenharia de Produção pela Pontifícia Universidade Católica
Instituição: Universidade do Estado do Pará - UEPA
Endereço: v. Dr. Eneas Pinheiro, 2626 - Marco, Belém - PA
E-mail: moah6@hotmail.com

Leonardo dos Santos Lourenço Bastos
Doutorando em Engenharia de Produção pela Pontifícia Universidade Católica
Instituição: Pontifícia Universidade Católica – PUC-RIO
Endereço: Rua Marques de São Vicente 225, Gávea, Rio de Janeiro- RJ
E-mail: lslbastos@aluno.puc-rio.br

Kamila Almeida dos Reis
Mestre em Engenharia de Produção pela Pontifícia Universidade Católica
Instituição: Pontifícia Universidade Católica – PUC-RIO
Endereço: Rua Marques de São Vicente 225, Gávea, Rio de Janeiro- RJ
E-mail: kamiladosreis13@gmail.com

RESUMO
Este trabalho trata do problema de *Flow Shop Scheduling Problem* (PFSP), onde um conjunto de tarefas devem ser sequenciadas numa quantidade de máquinas, objetivando a minimização do *Total Completion Time* (TCT) do processo. A proposta é, portanto, solucionar o problema através da aplicação das meta- heurísticas *Variable Neighborhood Descent* (VND) e *Iterated Local Search* (ILS). Para a construção da solução inicial, utilizou-se um algoritmo aleatorizado e o NEH, proposto por (Newaz Enscore e Ham 1983), e as buscas locais foram incorporadas nas meta-heurísticas para refinamento de resultados. Os métodos foram aplicados ao conjunto de instâncias proposto por (Taillard 1993) e tiveram seus respectivos desempenhos comparados às melhores soluções conhecidas na literatura a fim de conferir sua eficácia. Verificou-se, então, que o método NEH garante uma solução inicial de alta qualidade e, associado ao ILS, gera melhores resultados finais com tempo computacional razoável.

Palavras chave: Programação da produção, Sequenciamento de Tarefas, *Flow Shop,* Meta-heurísticas.

ABSTRACT
This research deals with the problem of Flow Shop Scheduling Problem (PFSP), where a set of tasks must be sequenced in a number of machines, focusing on minimizing the process Total Completion Time (TCT). Thus, the proposal is solving the problem through the application of Variable Neighborhood Descent (VND) and Iterated Local Search (ILS) meta-heuristics. To

Brazilian Journal of Development

construct the initial solution, it was used a randomized algorithm and NEH proposed by (Newaz Enscore and Ham 1983), and local searches were incorporated in the meta-heuristics to refine results. These methods were applied to instances proposed by (Taillard 1993), and their respective performances were compared to the best known solutions in literature in order to check their effectiveness. It was verified then, that the NEH method ensures a high quality initial solution and, associated with ILS, produces better final results with reasonable computing time.

Key-words: Production Scheduling, Job Scheduling, Flow Shop, Meta- heuristics**.**

1 INTRODUÇÃO

Os problemas de *scheduling* em *Flow Shop (Flow Shop Scheduling Problem* - FSP), são comuns em um sistema produtivo em que as tarefas seguem o mesmo fluxo por todas as máquinas na mesma sequência, este é um dos mais importantes e mais estudados problemas de planejamento da produção (Fuchigami e Rangel, 2015). Tais problemas tem o objetivo de obter uma sequência de tarefas que otimiza uma determinada medida de desempenho.

Nos modelos matemáticos utilizados para resolução do problema, as medidas mais utilizadas referem-se à minimização da duração total da programação (*makespan*), associada à utilização eficiente dos recursos produtivos disponíveis, e à minimização do tempo total de processamento (*total completion time*) (Rossi, Nagano e Tavares 2016). Sendo que esta última foi a medida trabalhada neste estudo.

Tal problema de programação da produção é NP-difícil, a dificuldade em propor e resolver os problemas deste nível de complexidade consistem num desafio que tem gerado diversas pesquisas no âmbito acadêmico (Garey *et al*. 1976, Rinnoy Kan 1976). Devido ao alto custo computacional dos modelos matemáticos de métodos exatos os algoritmos Heurísticos (H) e os Meta-Heurísticos (MH) são alternativas que alcançam resultados satisfatórios para tais problemas rapidamente, com um tempo computacional aceitável, mas não garantem que a solução obtida seja a solução ótima (Fernandez-Viagas e Framinan, 2015).

O objetivo deste trabalho é obter soluções de qualidade para o problema de Flow Shop *scheduling* com a redução do *total completion time* (TCT) das atividades, para oferecer apoio a tomada de decisão na programação da produção. Para isso, foram apresentadas duas meta-heurísticas baseadas nos algoritmos de *Variable Neighbordhood Search e Iterated Local Search*, as quais foram iniciadas com o método NEH, de forma a obter soluções de maior qualidade, as quais foram aplicadas a um conjunto de instancias da literatura.

Assim, este artigo foi estruturado com uma breve introdução do tema e sua importância, posteriormente no tópico referencial teórico abordou-se os assuntos relevantes ao estudo. Então apresentou-se o método e as heurísticas utilizadas de maneira mais ampla, no próximo tópico na

análise de resultados expõe-se os resultados encontrados e realiza-se uma comparação com os melhores tempos encontrados na literatura juntamente com os tempos computacionais de ambos.

2 REFERENCIAL TEÓRICO

2.1 O PROBLEMA DE FLOW SHOP

Em um problema de *flowshop,* um conjunto de *jobs* (tarefas) deve ser processado por meio de múltiplas estações na mesma ordem, desde a primeira até a última estação, e cada estação tem unicamente uma máquina (Lugo *et. al*. 2013).Para a modelagem deste problema são aceitas as premissas de que cada máquina está disponível continuamente, sem interrupções, de modo que o tempo de setup está incluso nos tempos de processamento independente das tarefas precedentes, sendo que cada tarefa possui apenas uma sucessora e uma precedente. Além disso, cada máquina processa apenas uma tarefa de cada vez (Fernandez-Viagas e Framinan 2015).

Alguns objetivos em realizar a alocação de tarefas corresponde à minimizar o *makespan* – tempo de término da última tarefa-, e o *Total Completion Time* – soma de todos os términos das tarefas, também denominado Cmáx (Yahyaoui *et al*. 2015). Quando o número de máquinas é maior que 3 o problema de *Flow Shop* é caracterizado como NP-difícil e por isso, faz-se o uso de heurísticas e meta-heurísticas para resolvê-los, de forma a reduzir os altos custos computacionais decorrentes de soluções advindas de métodos exatos.

2.2 HEURÍSTICAS E META-HEURÍSTICAS

Os métodos heurísticos podem ser classificados em construtivos e de refinamento, dependendo da forma de obtenção da solução. Os métodos construtivos se caracterizam pelo fato de gerarem apenas uma solução, que será a solução final do problema (Palmer 1965, Campbell *et al*. 1970, Gupta 1971, Dannenbring 1977, Nawaz *et al*. 1983, Koulamas 1998, Davoud Pour 2001, Nagano e Moccellin 2002, Kalczynski e Kamburowski 2007).

No caso dos métodos de refinamento, obtém-se uma solução inicial e, posteriormente, através de algum procedimento iterativo (que geralmente envolve trocas de posições dos jobs na sequência), como as buscas locais, busca-se obter uma sequência das tarefas melhor que a atual.

Após a construção de uma solução inicial aplica-se heurísticas de Busca Local (métodos de refinamento), que consiste em, caminhar pelo espaço de soluções, no intuito de refinar a solução corrente, essa nova solução é chamada de “solução vizinha”. Define-se como “solução vizinha” uma nova sequência das tarefas, obtida a partir da solução inicial, pela troca de posições entre duas tarefas adjacentes (Dannenbring 1977).

Brazilian Journal of Development

Neste trabalho foram utilizadas três tipos de buscas locais (Yahyaoui *et al*. 2015): *Swap:* troca aleatória de 2 *jobs* – também conhecido como 1-Opt; *Interchange*, troca entre *jobs* vizinhos; e *Insertion*, inserção de cada *job* em todas as possíveis posições até que todos os *jobs* tenham sido testados.

O termo Meta-heurísticas foi proposto por Glover (1986) para definir uma estratégia em um nível superior, que guia e modifica a operação de heurísticas subordinadas para explorar espaços de busca pelo uso de estratégias de aprendizado para produzir soluções de alta qualidade.

Arenales *et al*. (2015) afirma que uma meta-heurística pode manejar uma solução completa ou incompleta ou um conjunto de soluções. As heurísticas subordinadas ou coordenadas podem ser procedimentos de alto nível, ou mais simples como uma busca local, ou ainda apenas uma heurística construtiva.

O autor Reis (2013) listou algumas das meta-heurísticas mais conhecidas e amplamente utilizadas com sucesso, podem-se citar: algoritmo genético, busca *tabu*, *simulated annealing*, GRASP (*Greedy Randomized Adaptive Search Procedure*), ILS (*Iterated Local Search*), VND (*Variable Neighborhood Descended*), busca em vizinhança variável (*variable neighborhood search* – VNS), colônia de formigas, entre outras.

2.2.1 Newaz, Enscore e Ham (NEH)

Dentre os métodos construtivos mais conhecidos para a solução de problemas de FSP, salienta-se o NEH, proposto por (Nawaz *et al*. 1983), que tem sido considerado, até hoje, como o melhor método heurístico construtivo, para minimizar o *makespan* em problemas de *Flow Shop scheduling*, com alta qualidade de solução e eficiência computacional.

Este algoritmo consiste na geração de uma sequência inicial R pela ordenação decrescente do TCT das tarefas. A sequência final de *jobs* S é construída inserindo cada *job* de R em todas as posições possíveis de S, gerando sequências parciais e adotando a que minimiza TCT em cada iteração, até que todos os *jobs* tenham sido programados (Miyata 2015), conforme demonstrado no pseudocódigo ilustrado na Figura 1.

Brazilian Journal of Development

Figura 1 – Pseudocódigo da meta-heurística NEH Fonte: Pereira, 2011

NEH

1. Ordenar as tarefas de acordo com uma regra de despacho obtendo a lista $LC:=\{\pi(1),...,\pi(n)\}$;
2. Selecionar a tarefa $\pi(j)$ aleatoriamente de $LC=\{\pi(1),..., \pi(h)\}$, onde $h := max(1, \alpha\times|LC|)$;
3. $\pi := \pi(j)$ uma sequência parcial formada pela tarefa selecionada;
4. Remover a tarefa $\pi(j)$ em LC;
5. **Para** $i := 2$ até n **faça**
6. Selecionar a tarefa $\pi(j)$ aleatoriamente de LCR, onde LCR é formada pelas primeiras $h := max(1, \alpha\times|LC|)$ tarefas de LC;
7. Insira a tarefa $\pi(j)$ em todas as posições possíveis de π, gerando i sequências parciais com i tarefas;
8. $\pi' :=$ Selecione a melhor sequência gerada;
9. Remover a tarefa $\pi(j)$ de LC;
10. $\pi := \pi'$;
11. **Fim – Para**
12. Retorne a sequência π com n tarefas.

Um estudo realizado por (Park & Pegden, 1984) concluiu que o NEH supera de maneira significante outras 15 heurísticas estudadas. Em estudos posteriores, os autores (Turner e Booth, 1987), (Taillard, 1990), (Watson *et al*., 2002), (Framinan *et al*., 2004), (Ladhari e Haouari, 2005), (Ruiz & Maroto, 2005), (Gupta *et al*., 2006), também confirmaram a superioridade de desempenho do NEH em suas publicações.

Outras referências ao NEH podem ser encontradas em (Framinan *et al*., 2004), (Ruiz e Maroto 2006), e (Reza Hejazi e Saghafian 2005). Algumas meta-heurísticas que registraram os melhores resultados utilizaram sequência inicial determinada pelo NEH, conforme visto em (Reeves 1995), (Nowicki e Smutnicki 1996), (Stützle 1998), (Solimanpur *et al*. 2004), (Grabowski e Wodecki 2004), (Rajendran e Ziegler 2004), (Agarval *et al*. 2006), e (Ruiz *et al*. 2006).

2.2.2 Variable Neighborhood Descent (VND)

Proposto por (Mladenovic e Hansen 1997), o VND é uma meta-heurística que utiliza uma busca local para explorar o conjunto de soluções através de trocas de estruturas de vizinhança, aceitando apenas soluções que sejam melhores que a solução corrente e retornando à primeira estrutura quando uma solução melhor é encontrada. Caso não haja uma solução de melhor, a busca continuará na estrutura de vizinhança sequente. Não existindo uma estrutura de vizinhança sequente, o processo é interrompido e a solução corrente é retornada como a solução ótima local. Abaixo na Figura 2 o pseudocódigo do VND.

Brazilian Journal of Development

Figura 2 – Pseudocódigo da meta-heurística VND para Problema de Minimização Fonte: Reis, 2013.

Procedimento *Variable Neighborhood Descent*
1. Seja s_0 uma solução inicial e r o número de estruturas de vizinhança;
2. $s \leftarrow s_0$; {Solução corrente}
3. $k \leftarrow 1$; {Tipo de estrutura de vizinhança}
4. enquanto ($k \leq r$) faça
5. Encontre o melhor vizinho $s' \in N^{(k)}(s)$;
6. Se f(s') < f(s)
7. então $s \leftarrow s'$;
8. $k \leftarrow 1$;
9. senão $k \leftarrow k + 1$;
10. Fim-se;
11. Fim-enquanto;
12. Retorne s;

Fim

2.2.3 Iterated Local Search (ILS))

O ILS baseia-se na ideia de que um procedimento de busca local pode otimizado por meio de sucessivas perturbações na solução local (ótimo local), suscitando novas soluções iniciais para o método de busca (Glover & Kochenberg, 2003). O mesmo autor explica que o princípio da diversificação deve ser usado através de grandes perturbações na solução ótima local, com objetivo de explorar soluções de vizinhanças distantes da solução corrente.

A meta-heurística ILS baseia-se em quatro componentes ou procedimentos principais que afetam diretamente a sua performance (Glover & Kochenberg, 2003):

a) Procedimento gerador da solução inicial: gera uma solução inicial para o problema;

b) Procedimento de busca local: realiza uma busca na vizinhança da solução corrente até encontrar uma solução possivelmente melhorada;

c) Procedimento perturbatório: modifica a solução atual para uma solução intermediária s';

d) Procedimento do critério de aceitação: decide em qual solução a próxima perturbação será aplicada.

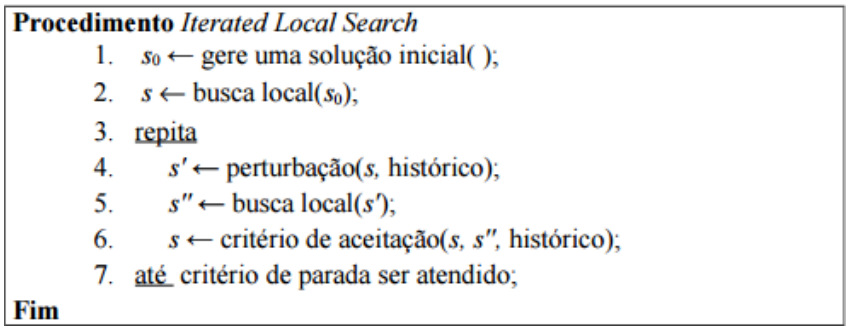
Por isso, segundo Reis (2013), a intensidade das perturbações devem ser calibradas até que possuam força suficiente para de fato produzir novas soluções iniciais diferentes das soluções locais já encontradas, porém sem demasiada intensidade a fim de não descaracterizar a solução local corrente. O critério de aceitação e perturbação devem seguir dois princípios básicos: o da diversificação e o da intensificação.

O princípio da diversificação consiste em aplicar grandes perturbações na solução ótima local, com o objetivo de explorar soluções de vizinhanças distantes da solução corrente. O princípio da intensificação consiste em aplicar leves perturbações na solução corrente, com intuito de

Brazilian Journal of Development

intensificar a busca local no entorno da melhor solução encontrada (Glover & Kochenberg, 2003). Abaixo na Figura 3 o pseudocódigo do ILS:

Figura 3 – Pseudocódigo da meta-heurística ILS para Problema de Minimização Fonte: Glover & Kochenberg, 2003



3 PROCEDIMENTOS METODOLÓGICOS

Para a realização do trabalho, foi feita uma comparação entre os modelos randômicos, modelos com base no *Total Completion Time* "Mais Cedo" (*Earliest Completion Time*), e o NEH, escolhendo para a fase construtiva das meta-heurísticas propostas o método que retornou melhores resultados (ver seção 4.1).

Após a escolha do método para a fase construtiva, foram propostos 4 métodos heurísticos que foram comparados segundo medidas comumente utilizadas na literatura (ver seção 3.1). O funcionamento de cada heurística se dá conforme apresentado a seguir:

3.1 META-HEURÍSTICA VND1BL

a)Passo 1: Calcule, para cada tarefa, a soma dos tempos de processamento em todas as máquinas

b) Passo 2: Construa uma solução inicial com o algoritmo NEH, de forma que os TCT fiquem organizados em ordem crescente (inverteu-se a ordem utilizada para melhores resultado)

c)Passo 3: Considere a vizinhança de insertion, swap e interchange salvando a melhor solução de cada método

d) Passo 4: Repita passo 3 até não haver melhoria

3.2 META-HEURÍSTICA VND2BL

a) Passo 1: Repita passos 1 e 2 do VND1BL

Passo 2: Considere a vizinhança de *insertion+swap, insertion+interchange, swap+insertion swap+interchange, interchange+insertion e interchange+swap* e salve a melhor solução de cada método

Brazilian Journal of Development

b) Passo 3: Repita passo 2 até não haver melhoria

3.3 META-HEURÍSTICA VND3BL

a) Passo 1: Repita passos 1 e 2 do VND1BL

b) Passo 2: Considere a vizinhança de *insertion+swap+interchange, insertion+interchange+swap, swap+insertion+interchange, swap+interchange+insertion, interchange+insertion+swap e interchange+swap+insertion* e salve a melhor solução de cada método

c) Passo 3: Repita passo 2 até não haver melhoria

3.4 META-HEURÍSTICA ILS1

a) Passo 1: Repita todos os passos do VND1BL

b) Passo 2: Gere uma perturbação

c) Passo 3: Considere a vizinhança de *insertion*, *swap* e *interchange* que foi utilizada no VND1BL salvando a melhor solução de cada método

d) Passo 4: Repita a sequência passo 2 e passo 3 até não haver melhoria

3.5 MEDIDA DE COMPARAÇÃO

A fim de definir o melhor método, foram utilizadas as seguintes medidas comparativas:

a) OBS: para cada instância, a melhor solução em cada método utilizado

b) Dev: para cada método, Dev é o desvio relativo percentual entre as melhores soluções encontradas na literatura (BKS) e a BS.

Dev = 100 * (BKS – BS) / BS

c) AvgDev: valor médio de Dev de um método para todas as instâncias

d) T: para cada método, o tempo médio de processamento (em segundos)

4 EXPERIMENTAÇÃO COMPUTACIONAL

Para verificar os resultados dos algoritmos propostos, foram realizadas experimentações de forma a obter as soluções e análise do comportamento dos componentes construtivos, buscas locais e as meta-heurísticas. Logo, os algoritmos foram aplicados às instâncias de (Taillard, 1993) pertencentes à três grupos: conjunto de 50 tarefas em 5 máquinas (50x5- ta031 a ta041), 50 tarefas em 10 máquinas (50x10- ta041 a ta050) e 50 tarefas em 20 máquinas (50x20- ta051 a ta060). Os resultados foram comparados às melhores soluções conhecidas (*Best Known Solutions* - BKS) em termos de resultados, e, aos melhores, fez-se uma análise dos tempos computacionais.

Para os métodos construtivos, compararam-se resultados entre os modelos randômicos, modelos com base no *Total Completion Time* "Mais Cedo" *(Earliest Completion Time*), e o NEH, foco deste estudo. Além disso, para a meta-heurística ILS, realizaram-se experimentos variando o fator de perturbação em passos de 10% do total de tarefas de cada instância, considerando um total de 120 iterações, para verificar qual a melhor opção em termos de resultados, considerando todas as combinações de buscas locais

4.1 MÉTODOS CONSTRUTIVOS

Em relação aos métodos construtivos, a comparação entre as três opções comprovou a eficácia do método NEH em termos de obtenção de resultados com menor TCT, conforme visto na Tabela 1, com um desvio relativo médio de 5,95% das BKS. Em termos de tempos computacionais, o NEH apresenta uma certa demora, em média 0,13 segundos, em detrimento de soluções de maior qualidade, devido a percorrer um conjunto maior de possibilidades.

Tabela 1 – Desvios Relativos de Resultados – Métodos Construtivos

AvgDev (%)	Aleatório			ECT	NEH
	Max	Med	Min		
ta031-ta040	40.80	36.76	32.71	17.24	**4.92**
ta041-ta050	34.11	32.38	30.65	20.82	**6.64**
ta051-ta060	26.82	25.79	24.77	17.42	**6.28**
Geral	33.91	31.64	29.37	18.49	**5.95**

Fonte: Autores (2016)

Além disso, também se observou que o uso de um algoritmo guloso, como a consideração de solução por ordem crescente de menor TCT, já realiza melhorias em termos de solução inicial. Ao se considerar uma rápida convergência do resultado para o TCT mínimo, o NEH ainda se apresenta como uma solução inicial de maior qualidade. Contudo, é possível que este comportamento possa oferecer certa dificuldade nas buscas locais seguintes.

5 ANÁLISE DE RESULTADOS

5.1 RESULTADOS EM VARIABLE NEIGHBORHOOD SEARCH – VND

Para as diferentes combinações utilizadas em VND, verificou-se que a medida que foram utilizadas mais buscas locais em sequência, os resultados apresentaram melhoras, em aproximadamente 1%, em relação aos algoritmos do tipo VND1BL. A melhor opção se apresentou ao utilizar um VND3BL, com a primeira busca local sendo *Swap* e as duas seguintes *Interchange* - *Insertion*, tendo resultados similares com a sequência *Insertion -Interchange*, conforme visto na Tabela 2

Brazilian Journal of Development

Tabela 2 – Desvios Relativos de Resultados – Métodos VND

AvgDv (%)					
Algoritmo	Combinação	ta031-ta040	ta041-ta050	ta051-ta060	Geral
VND1BL	*Insertion*	**2.63**	**3.58**	**2.83**	3.01
	Swap	3.45	4.57	4.35	4.12
	Interchange	4.72	6.47	6.14	5.78
VND2BL	*Insertion + Swap*	2.47	3.53	**2.79**	2.93
	Insertion + Interchange	2.63	3.58	2.83	3.01
	Swap + Insertion	**2.27**	**3.25**	3.04	2.85
	Swap + Interchange	3.45	4.57	4.35	4.12
	Interchange + Insertion	2.57	3.48	3.42	3.15
	Interchange + Swap	3.38	4.40	4.05	3.94
VND3BL	*Insertion + Swap + Interchange*	2.47	3.53	**2.79**	2.93
	Insertion + Interchange + Swap	2.47	3.53	**2.79**	2.93
	Swap + Insertion + Interchange	**2.27**	**3.25**	3.04	**2.85**
	Swap + Interchange + Insertion	**2.27**	**3.25**	3.04	**2.85**
	Interchange + Insertion + Swap	2.35	3.38	3.35	3.03
	Interchange + Swap + Insertion	2.32	3.40	3.36	3.03

Fonte: Autores (2016)

A busca local do tipo *Insertion* apresentou maiores convergências nas meta-heurísticas em que foram utilizados, visto que esta busca local consegue explorar um conjunto de vizinhanças maior que os métodos Swap e *Interchange*. Esta última, apesar de apresentar um tempo de execução menor, realiza poucas melhorias na solução final. Dessa forma, a melhor opção da meta-heurística se apresentou no tipo VND3BL, com cerca de 2,85% de desvio em relação às BKS.

5.2 RESULTADOS EM ITERATED LOCAL SEARCH – ILS

Em relação à meta-heurística ILS apresentada, notou-se uma grande melhoria dos resultados em relação aos algoritmos VNDs. Devido à presença do fator de perturbação, todos os métodos de busca local puderam alcançar melhores resultados. Entretanto, com o estilo de busca local e critérios de VND, e do número de iterações, o tempo computacional aumentou consideravelmente.

Para todos os algoritmos, o fator de perturbação conferiu melhores resultados ao considerar valores de 10% e 20%. Conforme demonstrado na Figura 4, tendo como exemplo a instância ta031, para o ILS1 com buscas locais *Interchange + Insertion*, verifica-se o comportamento das soluções para um conjunto de instâncias, e, na Figura 5, confirma-se a escolha do fator de perturbação para o algoritmo ao analisar as médias de cada conjunto. Com isso, pode-se verificar que fatores de perturbação muito altos acabam não resultado em resultados bons, em média, o que acontece devido à brusca mudança de vizinhança.

Brazilian Journal of Development

Figura 4 – Comportamento de Perturbação Interchange - Inserion

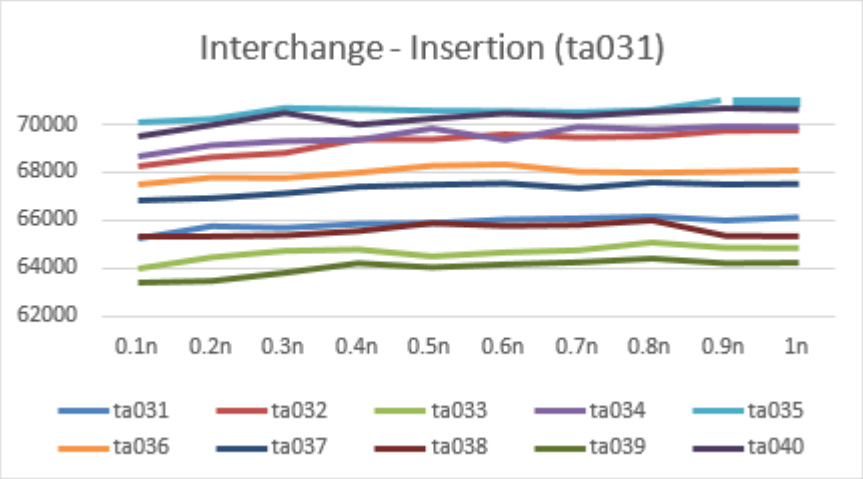
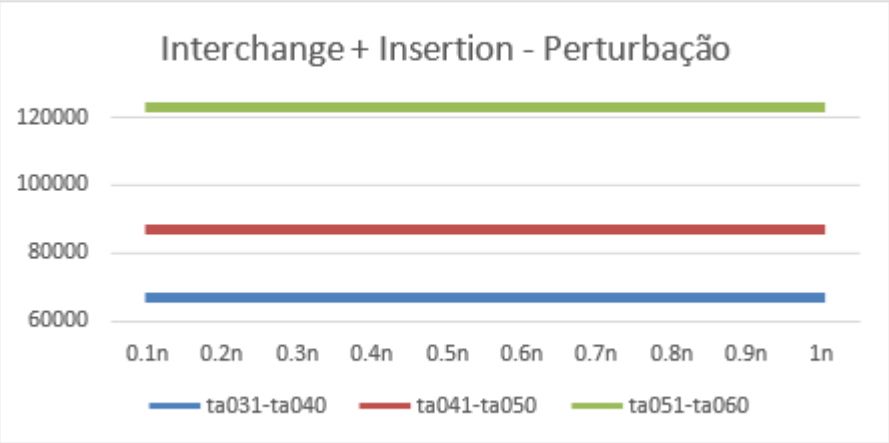


Figura 5 – Gráficos de Comportamentos da Perturbação entre Conjuntos de Instâncias



Conforme pode-se verificar na Tabela 3, o algoritmo com melhor resultado correspondeu ao tipo ILS1, e conferiu um desvio relativo de 1,09% em relação às BKS, com o uso da combinação *Interchange – Insertion*, com um fator de perturbação igual a 20% do total de tarefas. Notou- se que, para o conjunto de instâncias com mais máquinas (50x20), outras heurísticas apresentaram resultados mais satisfatórios. Porém, o tempo computacional do melhor algoritmo se apresentou, em média, menor, devido à velocidade do método *Interchange*,

Adita-se que o uso de métodos com maior abrangência de exploração de vizinhança, como o *Insertion e Swap*, também apresentação soluções com boa qualidade, mas ofereceram um tempo maior de execução. Verifica-se que a meta-heurística ILS1 com a combinação *Swap-Insertion*, assim como *Swap-Swap e Insertion-Insertion* resultaram em desvios relativos em quase 1%.

Brazilian Journal of Development

Tabela 3 – Desvios Relativos de Resultados – Métodos ILS

Instâncias	ILS - LS Interchange			ILS - LS Swap			ILS - LS Insertion		
	Insertion	Swap	Interchange	Insertion	Swap	Interchange	Insertion	Swap	Interchange
ta031	**0.61**	0.97	4.99	1.08	0.98	3.25	0.81	0.96	3.36
ta032	**0.34**	1.32	3.66	0.90	1.18	3.48	1.03	0.75	2.41
ta033	1.23	1.40	3.76	1.26	**1.05**	2.55	1.16	1.36	2.39
ta034	**0.66**	1.27	4.77	1.24	1.16	3.15	1.03	1.04	2.80
ta035	1.00	**0.87**	4.77	0.99	**0.87**	3.20	1.10	0.97	3.76
ta036	0.91	0.93	3.63	0.87	**0.85**	2.58	1.13	1.11	1.33
ta037	0.91	0.93	7.29	1.22	1.18	4.82	**0.67**	1.19	3.26
ta038	1.48	1.17	5.35	1.45	1.31	4.88	1.04	**0.91**	1.82
ta039	**0.66**	1.24	3.19	0.97	1.27	2.86	0.99	0.82	2.23
ta040	**1.03**	1.57	4.83	1.66	1.28	3.32	1.45	1.36	2.95
AvgDev	0.88	1.17	4.62	1.17	1.11	3.41	1.04	1.05	2.63
ta041	1.75	2.86	5.41	**0.84**	2.76	5.22	1.61	2.49	3.53
ta042	1.42	1.23	7.13	1.47	2.02	4.50	**1.06**	2.29	3.80
ta043	**0.84**	1.87	6.55	0.95	1.99	4.62	1.91	2.14	3.88
ta044	1.09	1.59	5.63	1.06	1.55	4.15	**0.80**	1.70	3.74
ta045	1.40	2.46	6.18	1.44	1.55	5.28	**0.84**	2.79	3.61
ta046	**0.68**	1.70	5.53	1.42	2.14	4.30	1.42	1.64	3.68
ta047	1.32	1.96	6.02	**1.08**	2.05	4.50	1.28	1.53	2.90
ta048	**1.01**	2.15	5.22	1.13	2.14	4.07	1.53	1.93	3.46
ta049	**1.08**	1.83	5.89	1.35	1.94	4.56	1.52	1.94	3.57
ta050	1.35	2.03	8.56	1.04	2.56	4.52	**0.82**	2.19	3.66
AvgDev	1.19	1.97	6.21	1.18	2.07	4.57	1.28	2.06	3.58
ta051	1.20	2.44	6.07	**0.78**	1.73	5.31	0.95	1.63	2.73
ta052	1.08	2.02	5.64	**0.90**	1.67	3.91	1.05	1.91	2.18
ta053	1.05	2.12	5.45	**0.74**	1.81	4.17	1.00	1.88	2.76
ta054	**1.44**	2.06	6.96	1.56	2.42	4.23	1.89	2.40	2.84
ta055	**0.94**	2.39	7.95	1.50	2.92	4.55	1.00	1.85	2.99
ta056	**0.89**	2.00	6.62	1.57	2.22	2.83	1.57	2.34	2.59
ta057	1.41	2.58	4.70	**1.19**	2.01	4.38	**1.19**	1.94	2.37
ta058	1.22	1.87	5.20	**1.10**	2.08	4.47	**1.10**	3.01	4.02
ta059	1.51	2.18	5.06	**0.84**	1.68	4.22	**0.84**	2.00	2.70
ta060	**1.11**	2.29	6.49	1.31	1.42	5.07	1.31	1.97	3.07
ta061	1.19	2.20	6.02	**1.15**	2.00	4.31	1.19	2.09	2.83
AvgDev	1.19	2.20	6.02	1.15	2.00	4.31	1.19	2.09	2.83
Geral	1.09	1.78	5.62	1.16	1.73	4.10	1.17	1.73	3.01

Fonte: Autores (2016)

Tabela 4 – Desvios Relativos de Resultados – Métodos ILS

AvgDev (%)	Aleatório			ECT	NEH
	Max	Med	Min		
ta031-ta040	40.80	36.76	32.71	17.24	**4.92**
ta041-ta050	34.11	32.38	30.65	20.82	**6.64**
ta051-ta060	26.82	25.79	24.77	17.42	**6.28**
Geral	33.91	31.64	29.37	18.49	**5.95**

Fonte: Autores (2016)

Brazilian Journal of Development

Conforme verificado na Tabela 4, é possível observar que, dentre as duas meta-heurísticas, há uma significativa diferença nos tempos computacionais. Em média, para cada conjunto de instâncias e no geral, o melhor método VND apresenta-se mais rápido em relação ao ILS, visto que o segundo realiza maior número de iterações a fim de obter melhores resultados

6 CONSIDERAÇÕES FINAIS

O uso de heurísticas e meta-heurísticas tem se mostrado eficaz para a resolução de problemas do tipo NP-Hard, tais como o *Flow Shop Scheduling Problem* apresentado neste estudo. Desta forma, os algoritmos propostos apresentaram resultados satisfatórios em termos de soluções e tempos de execução, em comparação a um possível resultado exato, apesar de não terem alcançado as melhores soluções presentes na literatura.

Notou-se a eficácia do método construtivo NEH, o qual já demonstra alta convergência da solução, com desvios de cerca de 6% da BKS, logo na solução inicial. Em seguida, as propostas baseadas em VND também apresentaram resultados satisfatórios, mas com desvio relativo mínimo de aproximada 2%, com o uso conjunto e sequente de três buscas locais. Logo, verifica- se que o ILS1 converge mais rapidamente, apesar do tempo de execução ser maior, principalmente devido à presença do fator de perturbação

Dentre as duas meta-heurísticas propostas, os melhores resultados em termos de qualidade de solução foram do algoritmo ILS1, utilizando a combinação de buscas locais *Interchange – Insertion*, com desvio de cerca de 1,09% em relação às BKS. Nota-se que a exploração mais ampla da vizinhança ocorre após a perturbação com o uso da segunda-busca local, a partir da ligeira melhora proporcionada pela primeira. Além disso, o algoritmo apresentou melhor tempo computacional em relação àqueles que utilizam combinações apenas de buscas locais mais exploratória (*Swap e Insertion*).

Adita-se que a realização de novos experimentos a partir de diferentes conjuntos de números aleatórios para calibragem do fator de perturbação do algoritmo ILS1 para possibilitar novas soluções. Uma maior precisão neste fator também pode resultar em soluções melhores, porém com tempo de execução maior. Com isso, o tempo computacional também pode ser reduzido ao adicionar critérios de paradas considerando um limite de tempo computacional e/ou um conjunto de soluções não melhoradas ao longo das iterações.

Brazilian Journal of Development

REFERÊNCIAS

AGARVAL A, COLAK S, ERYARSOY E. Improvement heuristic for the flow-shop scheduling problem: an adaptive-learning approach. European Journal of Operational Research, 169:801–15, 2006.

ARENALES M, ARMENTANO V. A, MORABITO, R. YANASSE H. H. Pesquisa Operacional para cursos de engenharia, 2° ed. Elsevier, 2015.

Dannenbring, D.G. An Evaluation of flow-shop sequencing heuristics. Management Science, Providence, v.23, n.11, p.1174-1182, 1977.

FERNANDEZ-VIAGAS, V., FRAMINAN J. M. A new set of high-performing heuristics to minimize flowtime in permutation Flow Shops. Copmuters & Operations Research, 53:68-90, 2015.

FRAMINAN, J.M., GUPTA J.N.D, LEISTEN R. A. Review and classification of heuristics for permutation flow-shop scheduling with makespan objective. Journal of the Operational Research Society, 55:1243–55, 2004.

FUCHIGAMI, H. Y. E RANGEL, **S**. Métodos heurísticos para maximização do número de tarefas just-in-time em Flow Shop permutacional, Em: Simpósio Brasileiro de Pesquisa Operacional, Porto de Galinas, Brasil, 25-28 de Agosto de 2015.

GAREY, M. R., JOHNSON, D. S., SETHI, R. The Complexity of Flow Shop and Jobshop Scheduling. Mathematics of Operations Research, 1, 117-129, 1976.

GLOVER, F.; KOCHENBERG, G. A. Handbook of Metaheuristics. International Series. Operations Research & Management Series, Kluwer's International Series, Stanford University, 2003.

GLOVER, F. Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research, Vol. 13, pp. 533-549, 1986.

GRABOWSKI J., WODECKI M. A very fast tabu search algorithm for the permutation Flow Shop problem with makespan criterion. Computers & Operations Research, 31:1891–1909, 2004.

GUPTA J. N. D., KOULAMAS, C., KYPARISIS G. J. Perfomance guarantees for Flow Shop heuristics to minimize makespan. European Journal of Operational Research, 169:865–72, 2006.

HAOUARI M., LADHARI T. A branch-and-bound-based local search method for the Flow Shop problem. Journal of the Operational Research Society, 54:1076–84, 2003.

KALCZYNSKI P. J., KAMBUROWSKI J. An improved NEH heuristic to minimize makespan in permutation Flow Shops. The International Journal of Management Science, 35(1):53–60, 2008.

LADHARI T, HAOUARI M. A. computational study of the permutation Flow Shop problem based on a tight lower bound. Computers & Operations Research, 32:1831–47, 2005.

LUGO, P. L. M, TEIXEIRA R. F., MARTÍNEZ K. P. Um modelo de programação inteira mista para a programação da Produção em Flow Shop híbrido com buffers limitados. Em: Simpósio Brasileiro de Pesquisa Operacional, Natal, Brasil, 16-19 de Setembro de 2013.

MIYATA, H. H. Métodos heurísticos para minimização da duração total da programação em ambiente no-wait Flow Shop com políticas de manutenção preventiva. 191 f. Dissertação (Mestrado)- Programa de Pós-Graduação em Engenharia de Produção Escola de Engenharia de São Carlos da Universidade de São Paulo, 2015.

MLADENOVIC, N., HANSEN, P. Variable neighborhood Descent. Computers and Operations Research, v. 24, n. 11, p. 1097–1100, 1977.

NAWAZ M., ENSCORE, J. R. E., HAM I. A heuristic algorithm for the m-machine n-job flow-shop sequencing problem. Omega. The International Journal of Management Science, 11:91–95, 1983.

NOWICKI E, SMUTNICKI C. A fast tabu search algorithm for the permutation flow-shop problem. European Journal of Operational Research, 91:160–75, 1996.

PARK Y. B., PEGDEN C. D. Enscore E.E.A survey and evaluation of static Flow Shop scheduling heuristics. International Journal of Production Research, 22:127–41, 1984.

PEREIRA, A. A. S. Metaheurísticas para o problema de Flow Shop flexível com penalidades de adiantamento e atraso 70f. Dissertação (mestrado) Programa de Pós-Graduação em Ciência da Computação Universidade Federal de Viçosa, 2011.

PINEDO, M. Scheduling: theory, algorithms, and systems. 2 ed. New Jersey: Prentic-Hall, 1995.

RAJENDRAN C., ZIEGLER H. Ant-colony algorithms for permutation Flow Shop scheduling to minimize makespan/total flowtime of jobs. European Journal of Operational Research, 155:426–38, 2004.

Reeves C. R. Genetic algorithm for Flow Shop sequencing. Computers & Operations Research, 15:5–23, 1995.

REIS J. V. A. Meta-heurísticas baseadas em busca em vizinhança variável aplicadas a problemas de operação de transportes. 221f. Tese (Doutorado) Escola Politécnica da Universidade de São Paulo em Engenharia de Transportes, 2013.

REZA HEJAZI S., SAGHAFIAN S. Flow Shop-scheduling with makespan criterion: a review. International Journal of Production Research, 43: 895–929, 2005.

RINNOOY KAN, A. H. G. Machine Scheduling Problems: Classification, Complexity, and Computations. The Hahue: Nijhoff., 1976.

ROSSI, F. L. NAGANO M. S., TAVARES R. F. N. Evaluation of high performance constructive heuristics for the Flow Shop with makespan minimization. The International Journal of Advanced Manufacturing Technology, 2016.

RUIZ R, MAROTO C. A. comprehensive review and evaluation of permutation Flow Shop heuristics. European Journal of Operational Research, 165:479–94, 2005.

RUIZ R., MAROTO C., ALCARAZ J. Two new robust genetic algorithms for the Flow Shop scheduling problem. Omega. The International Journal of Management Science, 34:461–76, 2006.

SOLIMANPOUR M., VRAT P., SHANKAR R. A neuro-tabu search heuristic for the Flow Shop scheduling problem. Computers & Operations Research, 31:2151–2164, 2004.

STÜTZLE T. Applying iterated local search to the Flow Shop problem. Applying iterated local search to the Flow Shop problem. Technical Report, AIDA-98-04, Computer Science Department, Intelictics Group, Darmstadt, Germany. Darmstadt University of Technology, 1988.

TAILLARD E. Some efficient heuristic methods for the Flow Shop sequencing problem. European Journal of Operational Research, 47:65–74, 1990.

TAILLARD E. Benchmark for Basic Scheduling Problems. European Journal of Operational Research, 64(2):278-285, 1993.

TURNER S, BOOTH D. Comparison of heuristics for Flow Shop sequencing. Omega. The International Journal of Management Science, 15:75–85, 1897.

WATSON J. P, BARBULESCU L, WHITLEY L.D, HOWE A. E. Contrasting structured and random permutation flow-shop scheduling problems: search-space topology and algorithm performance. INFORMS Journal on Computing. 14: 98–123, 2002.

YAHYAOUI, H., KRICHEN, S., DERBEL, B., TALBI, E-G. A hybrid ILS-VND based hyper-heuristic for permutation Flow Shop scheduling problem. Knowledge-Based and Intelligent Information & Engineering Systems. 60: 632-641, 2015