

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

**AVALIAÇÃO DE MÉTODOS HEURÍSTICOS PARA
MINIMIZAÇÃO DO MAKESPAN EM AMBIENTE FLOWSHOP
COM BLOQUEIO**

ADRIANO SEIKO KOMESU

Orientador: Prof. Dr. Marcelo Seido Nagano

São Carlos

2012

ADRIANO SEIKO KOMESU

**AVALIAÇÃO DE MÉTODOS
HEURÍSTICOS PARA MINIMIZAÇÃO DO
MAKESPAN EM AMBIENTE FLOWSHOP
COM BLOQUEIO**

Trabalho de Conclusão de Curso
Apresentado à Escola de Engenharia de São Carlos
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em Eletrônica

ORIENTADOR: Marcelo Seido Nagano

São Carlos
2012

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

Komesu, Adriano Seiko

K81h Avaliação de Métodos Heurísticos para minimização do makespan em ambiente flowshop com bloqueio. / Adriano Seiko Komesu ; orientador Marcelo Seido Nagano -- São Carlos, 2012.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2012.

1. *Flowshop* com boqueio. 2. Métodos heurísticos. 3. Programação da produção. Título.

FOLHA DE APROVAÇÃO

Nome: Adriano Seiko Komesu

Título: "Avaliação de Métodos Heurísticos para Minimização do Makespan em Ambiente Flowshop com Bloqueio"

Trabalho de Conclusão de Curso defendido e aprovado
em 20 / 6 / 2012,

com NOTA 10,0 (Dez, zero), pela comissão julgadora:

Marcelo Seido Nagano
Prof. Dr. Marcelo Seido Nagano (Orientador) - EESC/USP

Rogério Andrade
Prof. Dr. Rogério Andrade Flauzino - EESC/USP

Walther Azzolini Júnior
Prof. Dr. Walther Azzolini Júnior - EESC/USP

Homero Schiabel
Prof. Associado Homero Schiabel
Coordenador da CoC-Engenharia Elétrica
EESC/USP

Dedicatória

Aos meus pais, Alice e Seiko, dedico este trabalho por todo o apoio e esforços o que tornou possível a sua realização.

Dedico também este trabalho ao meu amigo Luis que de colega de sala tornou-se um grande irmão. Agradeço também aos meus amigos de sala pela companhia nos estudos e lazer, que tornaram meus anos em São Carlos inesquecíveis.

Agradecimentos

Agradeço aos meus irmãos, Andrea e Acácio, em especial minha irmã Andrea que dividiu um lar em São Carlos e me auxiliou nas horas difíceis de estudo e saúde.

Agradeço ao Prof. Marcelo S. Nagano pela paciência e dedicação em orientar ao longo de quase três anos e por toda experiência e conhecimento passado.

Agradeço também ao CNPq pelos dois anos de auxílio financeiro durante o período de iniciação científica.

Sumário

Lista de Tabelas	i
Lista de Gráfico.....	iii
Lista de Figuras	v
Lista de Símbolos	vii
Resumo	ix
Abstract	xi
1. Introdução.....	1
1.1. Objetivos da Pesquisa.....	2
1.2. Estrutura da Monografia	3
2. O Problema da Programação <i>Flowshop</i>	4
3. Revisão e Descrição dos Métodos Heurísticos	9
3.1. Heurísticas Construtivas Simples	13
• <i>McCormick – PF</i> (<i>McCormick, et al., 1989</i>)	13
• <i>Ronconi – MM</i> (<i>Ronconi, 2004</i>)	14
• <i>Pan QK – wPF</i> (<i>Liu & Reeves (2005) adaptado por Pan QK, 2011</i>)	15
3.2. Heurísticas Construtivas Melhoradas	17
• <i>Nawaz – NEH</i> (<i>Nawaz-Enscore-Ham, 1983</i>)	17
• <i>Ronconi – MME</i> (<i>Ronconi, 2004</i>).....	17
• <i>Ronconi – PFE</i> (<i>Ronconi, 2004</i>)	18
• <i>Pan QK – wPFE</i> (<i>Ronconi (2005) adaptado por Pan QK, 2011</i>)	18
• <i>Pan QK – PF-NEH</i> (<i>Pan QK, 2011</i>)	19
• <i>Pan QK – wPF-NEH</i> (<i>Pan QK, 2011</i>)	19
• <i>Pan QK – PF-NEH(x)</i> (<i>Pan QK, 2011</i>)	20
• <i>Pan QK – wPF-NEH(x)</i> (<i>Pan QK, 2011</i>)	20
3.3. Heurísticas Construtivas Compostas.....	22
• <i>Pan QK – RLS</i> (<i>Pan QK, 2007</i>)	22
3.4. Exemplos numéricos dos Métodos	25
4. Experimentação Computacional	34
5. Análise dos Resultados.....	35
Heurísticas Simples.....	35

Heurísticas Melhoradas	40
Heurísticas Compostas	45
Todas as heurísticas	50
6. Considerações Finais.....	58
Referências Bibliográficas.....	59

Lista de Tabelas

Tabela 1 - Exemplo numérico para obtenção do valor de <i>makespan</i>	7
Tabela 2: Fases das Heurísticas.....	24
Tabela 3: Dados para os exemplos numéricos	25
Tabela 4: Porcentagem de Sucesso para as Heurísticas Simples	35
Tabela 5: Desvio Relativo Médio para as Heurísticas Melhoradas.....	35
Tabela 6: Tempo Computacional para as Heurísticas Simples	36
Tabela 7: Porcentagem de Sucesso para as Heurísticas Melhoradas	40
Tabela 8: Desvio relativo Médio para as Heurísticas Melhoradas	40
Tabela 9: Tempo Computacional para as Heurísticas Melhoradas	41
Tabela 10: Porcentagem de Sucesso para as Heurísticas Compostas	45
Tabela 11: Desvio Médio Relativo para as Heurísticas Compostas	45
Tabela 12: Tempo Computacional para as Heurísticas Melhoradas	46
Tabela 13: Porcentagem de Sucesso para todas as Heurísticas	51
Tabela 14: Desvio Relativo Médio para todas as Heurísticas.....	51
Tabela 15: Tempo de Processamento para todas as Heurísticas	52

Lista de Gráfico

Gráfico 1 - Sequenciamento de cinco tarefas em cinco máquinas e uma situação genérica
(Retirado de Pan QK, 2012)..... 7

Lista de Figuras

Figura 1 - Esquema de relação entre as diversas classes de problemas de programação de operações em máquinas (Adaptado de MacCarthy & Liu, 1993).....	5
Figura 2: Procentagem de Sucesso para as Heurísticas Simples	36
Figura 3: Média da PS para Heurísticas Simples	37
Figura 4: Desvio Relativo Médio para as Heurísticas Simples	37
Figura 5: Média da DRM para Heurísticas Simples.....	38
Figura 6: Tempo de Processamento Relativo para as Heurísticas Simples	38
Figura 7: Porcentagem de Sucesso para as Heurísticas Melhoradas	41
Figura 8: Desvio Relativo Médio para as Heurísticas Melhoradas.....	42
Figura 9: Média da PS para Heurísticas Melhoradas	42
Figura 10: Média da DRM para Heurísticas Melhoradas.....	43
Figura 11: Tempo de Processamento Relativo para as Heurísticas Melhoradas.....	43
Figura 12: Porcentagem de Sucesso para as Heurísticas Compostas.....	46
Figura 13: Desvio Relativo Médio para as Heurísticas Compostas	47
Figura 14: Tempo Computacional Relativo para as Heurísticas Compostas	47
Figura 15: PS Média para as Heurísticas Compostas	48
Figura 16: DRM Média para as Heurísticas Compostas.....	48
Figura 17: Porcentagem de Sucesso para todas as Heurísticas	53
Figura 18: Desvio Relativo Médio para todas as Heurísticas	54
Figura 19: Tempo Computacional Relativo para todas as heurísticas.....	55
Figura 20: Tempo Computacional Absoluto para todas as Heurísticas	56
Figura 21: Média da PS para todas as Heurísticas	56
Figura 22: Média da DRM para todas as Heurísticas	57

Lista de Símbolos

n	Número de tarefas
m	Número de máquinas
$d_{i,m}$	Tempo de partida da tarefa i da máquina m
C_{max}^*	Valor ótimo de <i>Makespan</i>
C_{max}	Duração total da programação – <i>Makespan</i>
T_i	Tempo total de Processamento da tarefa i
β, y, π, π'	Sequencia de tarefas
β_i, π_i	Tarefa i
O	Complexidade Computacional
$p_{i,m}$	Tempo de Processamento da tarefa i na máquina m
U	Tarefas não sequenciadas (agendadas)
J	Conjunto de todas as tarefas
π^{ref}	Sequencia de Tarefas usadas como referencia
$\delta_{j,k}$	Soma dos tempos de <i>idle</i> e <i>blocking</i> para a tarefa j na k -ésima posição
α	Ordenação não-descrescente das tarefas
$f(\pi)$	Cálculo do <i>makespan</i> para a sequencia π

Resumo

O presente trabalho trata o problema de Programação de Operações em máquinas no ambiente *flowshop* com bloqueio com o objetivo de minimizar a duração total da programação. O problema *flowshop* com bloqueio possui diversas aplicações em indústrias, por exemplo, em indústrias químicas e de blocos de concreto. Recentemente, diversos métodos heurísticos foram propostos na literatura para solução do problema e muitos deles não foram avaliados de uma forma eficiente para identificar o melhor método heurístico. Dessa maneira, esta monografia tem como objetivo principal realizar um estudo da literatura e através de uma extensa experimentação computacional identificar o melhor método heurístico entre os reportados na literatura atual. Os resultados são analisados em termos da Porcentagem de Sucesso, Desvio Relativo Médio e o Tempo de Programação.

Palavras-chave: *Flowshop* com bloqueio, Métodos Heurísticos, Programação da Produção.

Abstract

The present work deals with the problem of scheduling operations on machines in the flowshop environment with blocking in order to minimize the total duration of the scheduling. The problem with blocking flowshop has many applications in industries, for example, in chemical and concrete blocks. Recently, several heuristic methods have been proposed in the literature for solving the problem and many of them have not been evaluated in an efficient way to identify the best heuristic. Thus, this work aims at providing a survey of the literature and through an extensive computational experiment to identify the best among the heuristic method reported in the literature. The results are analyzed in terms of percentage of success, relative mean deviation and Time Programming.

Keywords: Flowshop with Blocking, Heuristic Method, Scheduling Operations.

1. Introdução

Desde o surgimento da indústria de manufatura moderna, ao longo dos anos a competitividade do mercado passou a exigir uma produção mais eficiente com um controle mais fino, o que resultou na busca por uma melhoria no desempenho dos sistemas produtivos e de todos os seus indicadores e, com isso, reduzir o tempo total de programação. Com esta importância, o problema de programação da produção tem sido estudado intensamente na literatura.

Com o decorrer do tempo, vários trabalhos surgiram na busca da resolução deste problema. Um dos primeiros trabalhos exibidos foi reportado por Johnson (1954) que aborda soluções ótimas para problemas com duas máquinas. Com a extensão do problema para m máquinas, a obtenção de soluções ótimas tornou-se impraticável devido ao grande esforço computacional. Desde então, métodos que vão de heurísticas construtivas até metaheurísticas. Esta diversidade de métodos é classificada como sendo NP-Hard, assim, a maioria dos métodos fornecem soluções boas, porém, não ótimas.

Focando o estudo em heurísticas construtivas, neste trabalho foi notado que na literatura de Flowshop com Bloqueio, os métodos propostos foram avaliados de forma independente sem que haja uma comparação entre eles. Com isso, este trabalho propõe a seleção dos melhores métodos atuais presentes na literatura para um estudo efetivo.

1.1. Objetivos da Pesquisa

O objetivo deste trabalho é realizar uma revisão da literatura no assunto *flowshop* com bloqueio realizando uma extensa experimentação computacional dos melhores métodos heurísticos estudados da literatura. Com os dados computacionais obtidos, será realizada uma comparação efetiva entre os métodos por meio das estatísticas: Porcentagem de Sucesso e Desvio Relativo Médio indicando os melhores métodos heurísticos existentes. O resultado desta análise será utilizado para trabalhos futuros na comparação de novos métodos heurísticos.

1.2. Estrutura da Monografia

Esta monografia está dividida em seis seções com o objetivo de organizar de forma a facilitar a compreensão do assunto estudado. A primeira seção apresenta os objetivos fundamentais da pesquisa. A terceira seção realiza uma revisão literária do problema e as descrições dos métodos heurísticos estudados em conjunto com exemplos numéricos para ilustrar a implementação dos métodos. A quarta seção apresenta os resultados computacionais obtidos com a extensa experimentação computacional realizada. A quinta seção apresenta a análise dos resultados obtidos através da programação dos métodos heurísticos com uma série de tabelas e gráficos ilustrativos. A sexta seção, apresenta as considerações finais e as conclusões obtidas através da análise dos resultados.

2. O Problema da Programação *Flowshop*

Com a alta mecanização das linhas de produção e com a exigência de uma produtividade maior, buscaram-se maneiras de realizar a programação nas máquinas disponíveis de uma maneira a otimizar o processo nos recursos. A maioria dos problemas de programação da produção é complexa e de difícil solução devida a restrições tecnológicas e de linha que devem ser especificadas. De acordo com MacCarthy & Liu (1993), neste contexto, os problemas de programação podem ser classificados da seguinte maneira:

- a) *Jobshop*: cada tarefa possui sua própria ordem de processamento nas máquinas.
- b) *Flowshop*: as tarefas tem a mesma sequencia de processamento nas máquinas.
- c) *Openshop*: não há uma sequencia específica ou preestabelecida para o processamento nas máquinas.
- d) *Flowshop Permutacional*: um *Flowshop* no qual a ordem de processamento nas máquinas é a mesma.
- e) *Máquina única*: existe somente um estágio de produção com uma única maquina.
- f) *Máquinas Paralelas*: há mais de uma máquina disponível em um único estágio de produção.
- g) *Jobshop com múltiplas máquinas*: um *Jobshop* com a existência de um conjunto de máquinas paralelas em cada estágio da produção.
- h) *Flowshop com múltiplas máquinas*: *Flowshop* com a existência de um conjunto de máquinas paralelas em cada estágio de produção

A Figura 1 ilustra a relação entre as classes dos problemas da programação nas máquinas:

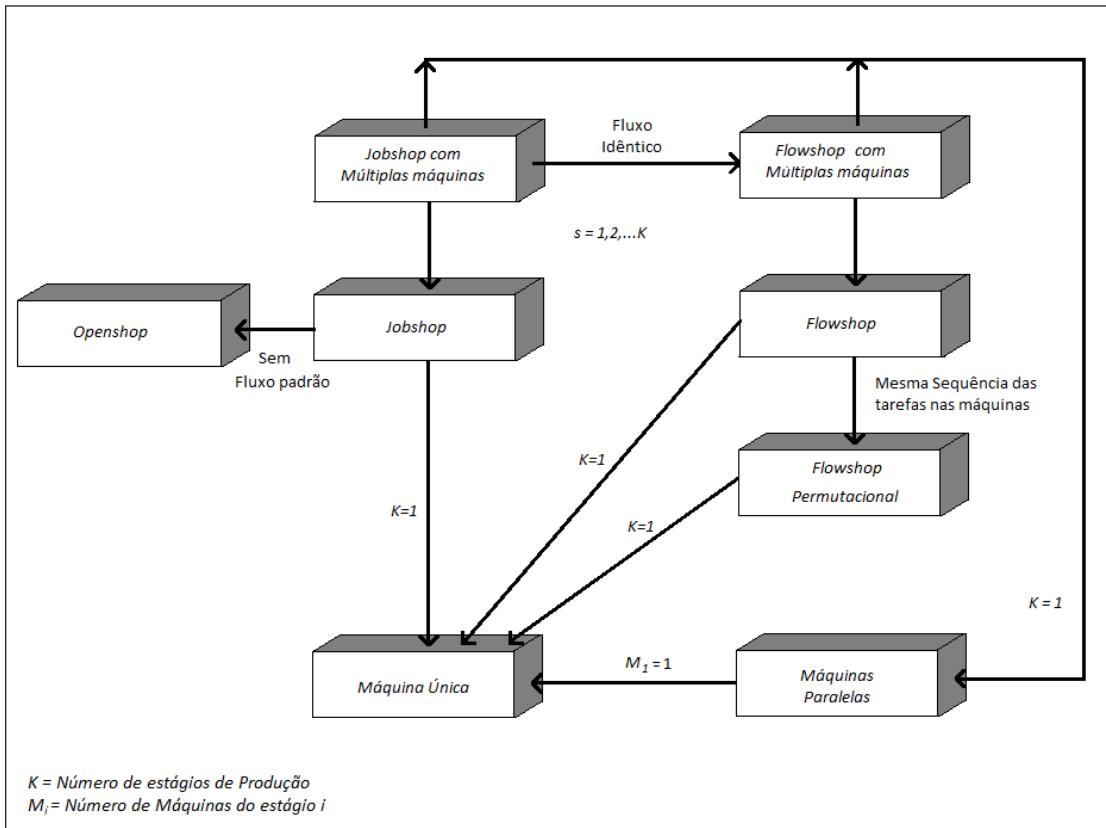


Figura 1 - Esquema de relação entre as diversas classes de problemas de programação de operações em máquinas (Adaptado de MacCarthy & Liu, 1993).

Neste trabalho será focado o problema de *Flowshop com bloqueio* (*Blocking Flowshop Scheduling Problem, BFSP*) e este é denotado como $Fm/block/C_{max}$. Garey, et al. (1979) utilizou o padrão de três campos $\alpha/\beta/\gamma$ para representá-lo no qual α representa o ambiente das máquinas, β os detalhes das características de processamento e γ a função objetivo. Em linhas gerais, segundo Taillard, (1993), o problema de programação da produção é definido como o conjunto de n tarefas $\{j_1, j_2, \dots, j_n\}$ que devem ser processadas em m máquinas $\{m_1, m_2, \dots, m_m\}$ que estão disponíveis, onde cada tarefa é sequenciada na primeira máquina em seguida na segunda máquina e assim sucessivamente até a última máquina. Para este caso, são consideradas as seguintes exigências:

- Não possua *buffers* intermediários entre as máquinas;
- Todas as tarefas são independentes;
- Disponibilidade das tarefas para o processamento no tempo Zero;
- As máquinas estão continuamente disponíveis;
- Cada máquina pode processar no máximo uma tarefa por vez;
- Cada tarefa pode ser processada no máximo em uma máquina a qualquer momento.

A solução do Sistema *Flowshop* tem como objetivo determinar entre as $n!$ possíveis sequências aquela que minimiza um critério estabelecido. O critério que será utilizado neste

trabalho para a comparação e análise de resultados será a duração total da programação (*Maximum Completion Time* ou *makespan*). O *makespan* é relacionado ao uso eficiente dos recursos de produção e ele é determinado quando se obtém o tempo de processamento de todas as tarefas em todas as máquinas.

Cada sequenciamento das tarefas pode ser representado por uma permutação $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ onde $\pi_i \in J$. Seja Π como sendo o conjunto de todas as permutações possíveis. O objetivo para o problema é obter uma permutação $\pi^* \in \Pi$ tal que:

$$C_{max}(\pi^*) \leq C_{max}(\pi) \quad \forall \in \Pi \quad (1)$$

onde C_{max} representa o valor de *makespan* para a permutação de π .

Seja o tempo de partida $d_{i,k}$ da operação $O_{i,k}$, na qual i representa a i ésima tarefa da permutação π . De acordo com Pan QK, et al. (2011), $d_{i,k}$ pode ser calculado como a seguir:

$$d_{1,0} = 0 \quad (2)$$

$$d_{1,k} = \sum_{q=1}^k p_{1,q}, \quad k = 1, 2, \dots, m - 1, \quad (3)$$

$$d_{i,0} = d_{i-1,1}, \quad i = 2, \dots, n, \quad (4)$$

$$d_{i,k} = \max(d_{i,k-1} + p_{i,k}, d_{i-1,k+1}), \quad i = 2, \dots, n, k = 1, \dots, m - 1 \quad (5)$$

$$d_{i,m} = d_{i,m-1} + p_{i,m}, \quad i = 1, \dots, n, \quad (6)$$

Onde $d_{i,0}, i = 1, 2, \dots, n$, (2) significa o começo do tempo da tarefa π_i na primeira máquina e $d_{i,m}$ representa o tempo completo da tarefa π_i no processo, nas equações (3), (5) e (6) o termo $p_{i,m}$ é o tempo de processo da tarefa i na máquina m . Recursivamente, os tempos de partida da primeira tarefa são calculados em todas as máquinas primeiramente, posteriormente os da segunda tarefa e assim sucessivamente até a última tarefa. Com isso, o *makespan* da programação $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ é definido por $C_{max}(\pi) = \max_{i=1,2,\dots,n} d_{i,m} = d_{n,m}$. O grau de complexidade para obter o valor de $C_{max}(\pi)$ é definido como $O(mn)$.

Para ilustrar a obtenção deste valor, a Tabela 1 apresenta um exemplo que representa um processo de cinco tarefas em cinco máquinas e, cada elemento desta, representa os valores de processo de cada tarefa em cada máquina, $p_{i,m}$.

Tabela 1 - Exemplo numéricico para obtenção do valor de makespan.

	J1	J2	J3	J4	J5
Máquina 1	2	5	6	1	9
Máquina 2	3	8	7	10	4
Máquina 3	4	9	3	12	6
Máquina 4	7	1	5	9	1
Máquina 5	2	2	6	8	4

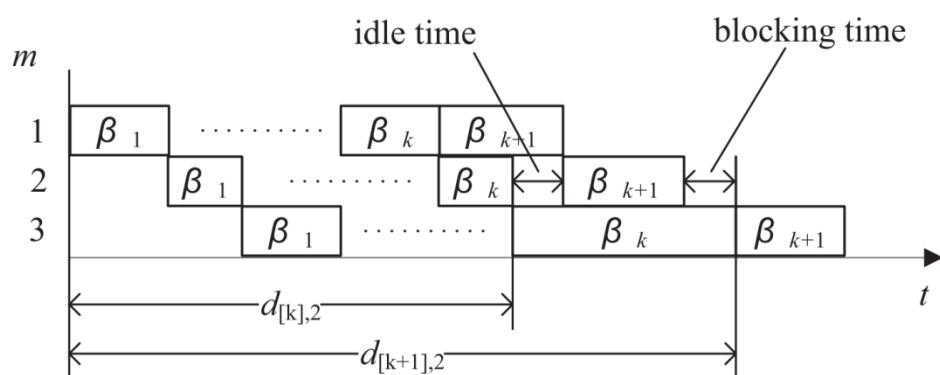
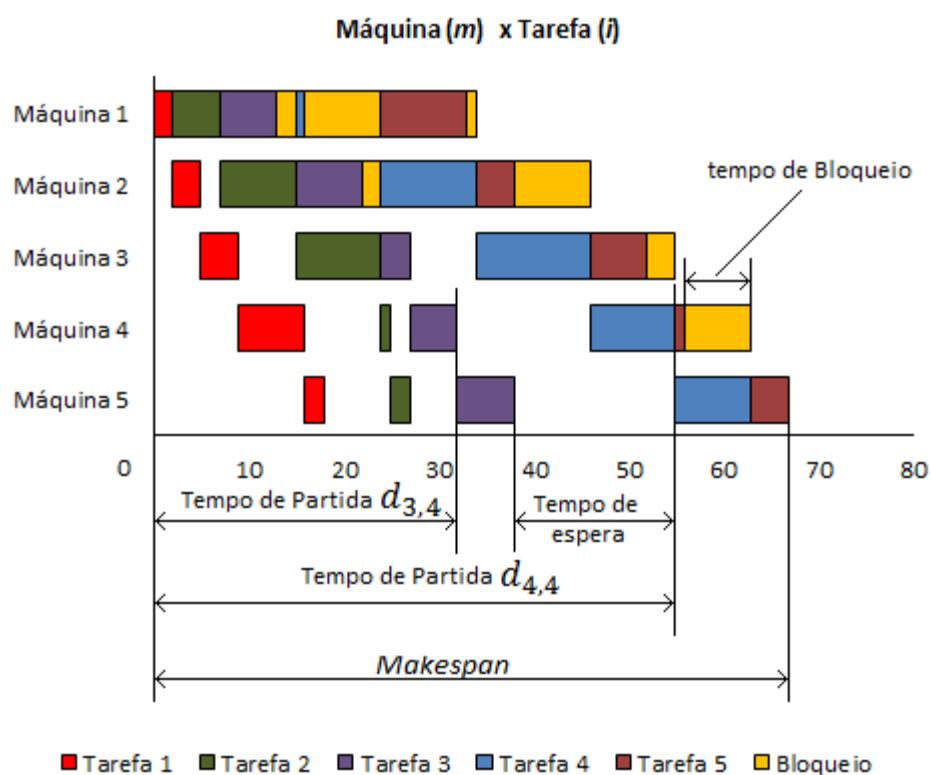


Gráfico 1 - Sequenciamento de cinco tarefas em cinco máquinas e uma situação genérica
(Retirado de Pan QK, 2011).

O Gráfico 1 representa o sequenciamento das cinco tarefas nas cinco máquinas da Tabela 1. Graficamente, obtemos o valor do *makespan* de $C_{max}(\pi) = 68$. São ilustrados também, os tempos de partida de duas tarefas na quarta máquina, $d_{3,4} = 32$ e $d_{4,4} = 38$. O tempo de espera ocorre quando a máquina m , após a partida da tarefa β_k , aguarda a tarefa β_{k+1} para um novo processo. O tempo de bloqueio ocorre pela falta de um armazenamento intermediário, isto é, a tarefa β_{k+1} deve aguardar na máquina m até que a tarefa β_k parta da máquina $m+1$.

3. Revisão e Descrição dos Métodos Heurísticos

Ao longo do tempo, o estudo para o problema da programação para o ambiente *flowshop* com bloqueio foi tema de diversos trabalhos que propuseram heurísticas e meta-heurísticas para a solução da minimização do *makespan* para este tipo de problema. Com uma simples explicação, as soluções heurísticas exigem um menor esforço computacional devido a um menor grau de complexidade com relação às soluções obtidas através de meta-heurísticas, porém, estes métodos resultam em soluções melhores. Nesta seção, ela será subdividida em duas etapas: primeiramente serão revistas heurísticas e posteriormente meta-heurísticas.

Levner (1969) apresentou o algoritmo *Branch-and-Bound* (*B&B*) específico para o ambiente *flowshop* com bloqueio e este mostrou que calculando o *makespan* de um ambiente *flowshop* é equivalente a encontrar o maior caminho utilizando um gráfico direto. Mais tarde, Suhami & Man (1981) propuseram também um algoritmo *Branch-and-Bound* que usa um parâmetro empírico no qual ajusta o valor do limitante inferior, superando o algoritmo anterior para problemas de pequeno porte, encontrando soluções próximas a 1% dos valores ótimos. Ronconi e Armentano (2001) apresentaram, utilizando o mesmo algoritmo, novos valores de limitante inferior.

McCormick, et al. (1989) desenvolveu o algoritmo *Profile Fitting* (PF) que realiza o sequenciamento das tarefas de acordo com a soma dos tempos de *idle* e *blocking*. A ordenação por este método tem como objetivo reduzir a soma destes tempos consequentemente na busca da redução do *makespan*.

Nawaz et al. (1983) criaram o algoritmo NEH que a princípio, foi desenvolvida para o problema *flowshop* clássico. Posteriormente foi amplamente estudada em diversos tipos de problema *flowshop* devido a sua fácil aplicação que obtêm soluções de qualidade com um esforço computacional baixo.

Leisten (1990) apresentou uma abordagem mais abrangente, propôs duas heurísticas que lidam com Permutação *flowshop* e com a Não-Permutação *flowshop* com uma quantidade finita e ilimitada de *buffer*. A principal proposta das heurísticas de Leisten é a maximização do uso de *buffers* e minimizar o bloqueio na máquina. Em seu trabalho, comparou suas heurísticas e com outras e concluiu que o algoritmo NEH inicialmente proposta para *buffer* ilimitado, tem um desempenho superior.

Como base de um trabalho anterior de Ronconi & Armentano (2001) apresentaram novas propriedades para o critério de *makespan*, Ronconi (2004) criou uma nova heurística denominada *MinMax* (MM). A heurística MM foi comparada com a heurística PF em seu trabalho, mas ela foi superada pela heurística PF. No mesmo trabalho, Ronconi apresentou

outras duas novas heurísticas, MME e PFE, que seriam respectivamente as heurísticas MM e PF combinadas com o procedimento de enumeração de NEH. Estas novas heurísticas apresentadas foram comparadas com o método NEH, considerado até o momento como sendo o melhor algoritmo para a solução do problema de *flowshop* com bloqueio. Os novos métodos apresentaram um desempenho superior com relação ao número de melhores soluções com relação ao NEH. O algoritmo PFE teve uma superação significativa para problemas com 500 tarefas.

Abadi et al. (2000) propôs uma heurística para a diminuição do tempo de ciclo no problema e recentemente, Ronconi & Henriques (2009) apresentaram algumas heurísticas construtivas para a minimização do atraso total no problema e os métodos tiveram resultados promissores.

Recentemente, Pan QK (2011) apresentou uma série de heurísticas que ele as classificou como: heurísticas construtivas simples, heurísticas construtivas melhoradas e heurísticas construtivas compostas. A heurística simples apresentada em seu trabalho foi wPF. Esta heurística é uma variação da heurística PF. Quanto as heurísticas construtivas melhoradas, apresentou a wPFE, PF-NEH(x), wPF-NEH(x), estas tres heurísticas, são variações de PFE apresentada por Ronconi, sendo o índice x o número de sequencias que é gerada. Nas heurísticas construtivas compostas foi utilizada uma heurística de busca local que foram já referenciadas em outros trabalhos de Smutnicki (1998), Liu & Reeves (2001), Framinan (2004) entre outros. A heurística utilizada foi a *Referenced Local Search* (RLS) que previamente Pan QK (2007) havia apresentado. Nesta classe, foi aplicada esta busca local em todas as heurísticas anteriormente citadas (heurísticas construtivas simples e heurísticas construtivas melhoradas). Em sua análise, os resultados foram comparados com as soluções de Ribas (2011) e os resultados obtidos apresentaram boa qualidade. Entre as heurísticas construtivas simples, a heurística wPF obteve um resultado superior às apresentadas por McCormick e Ronconi. O método wPF-NEH(5) obteve um resultado superior entre os métodos desta classe. O método PF – NEH_{LS(5)} obteve um resultado superior a todas as anteriores.

Para as meta-heurísticas, Nowicki (1999) generalizou a noção de bloqueio com tarefas, sugerido por Grabowski, et al. (1983) para o ambiente *flowshop*. Nowicki utilizou a meta-heurística de busca *tabu* (*Tabu search, TS*), propriedade que reduz a vizinhança de uma busca local. Em seus testes, os seus resultados apresentaram uma qualidade superior a outros métodos conhecidos na literatura.

Caraffa,et al. (2001), desenvolveu um algoritmo genético para problemas de grande porte com uma desaceleração restrita no qual o problema de *flowshop* com bloqueio foi abordado como um caso especial. Grabowski & Pempera (2007) desenvolveu uma TS que

de acordo com seus testes computacionais, apresentou uma superioridade sobre os métodos de algoritmo genéticos de Caraffa e *Branch-and-Bound* de Ronconi.

Wang, et al. (2010) apresentou o método *Hybrid discrete differential evolution* (HDDE) que trata uma versão discreta de uma evolução diferencial e uma busca local baseada em vizinhança de inserção. Foi mostrado que o desempenho das soluções de HDDE foi muito superior a apresentada por TS.

Em um trabalho mais recente, Ribas, et al. (2011) em seu trabalho, propôs o algoritmo *Iterated Greedy* e Ribas demonstrou que seu método obtinha soluções de qualidade superiores ao método HDDE, fornecendo as melhores soluções conhecidas.

De acordo com a revisão literária realizada neste trabalho, com foco nas heurísticas, percebe-se que vários métodos de solução para o problema de minimização do *makespan* foram elaborados de forma independente, não sendo realizada uma comparação efetivamente entre eles. O único trabalho que realiza uma comparação entre métodos, seria o trabalho de Ronconi (2005) que provou a superioridade das heurísticas propostas, MME e PFE, sobre as heurísticas PF e NEH, consideradas as melhores para este tipo de problema. O trabalho de Pan QK (2011) apresentou diversas heurísticas construtivas, mas seus resultados foram comparados com as soluções de Ribas (2011), ele mostrou que seus resultados foram próximos aos resultados de Ribas, porém, não houve uma confrontação entre os métodos propostos e os existentes na literatura.

Com os aspectos apresentados acima, a comparação entre os melhores métodos será realizada de acordo com a classificação apresentada por Pan QK (2011), sendo esta classificação dividida em três categorias: Heurística Construtiva Simples, Heurística Construtiva Melhorada e Heurística Construtiva Composta.

Como visto, há um grande número de métodos heurísticos propostos para o problema de minimização do *makespan* e é conveniente classificá-los em categorias distintas, nas quais se considera a estrutura da heurística proposta, se há a existência ou não de uma fase de melhoramento da solução. Para uma melhor estruturação, este trabalho será baseado na proposta de Framinan et al. (2005) mas será adaptado de acordo com as características do trabalho de Pan QK et al. (2011). A estruturação de Framinan sugere que os métodos sejam classificados em três fases:

- a) *Index Development*: Esta fase tem como objetivo obter uma ordenação simples das tarefas de acordo com uma regra de prioridade, baseada na soma dos tempos de processamento das tarefas nas máquinas. Há diversas maneiras de se obter essa regra.
- b) *Solution Construction*: Nessa fase é onde se caracteriza o método, mas apesar de prover uma solução, ela pode não garantir que a sequencia gerada seja melhor que a sequencia gerada pela primeira fase.

- c) *Solution Improvement*: Como base a solução da segunda fase, tenta-se obter uma nova solução com a inserção da k -ésima tarefa em todas as posições possíveis das soluções parciais que são obtidas.

A maioria dos trabalhos presentes na literatura possui a classificação acima como característica de estruturação. No trabalho de Pan QK (2011), ele utilizou as três fases anteriores descritas, porém, após a terceira fase, ele aplicou um algoritmo de busca local e caracterizou as Heurísticas como Compostas (*Composite heuristics*). Devido ao grande desempenho apresentado por essa composição realizada por Pan QK, podemos classificá-la como uma nova etapa.

- d) *Solution Composite*: esta fase também busca um melhoramento, mas são utilizados comumente métodos de busca local para obter uma nova solução que seja melhor ou igual a fornecida pela fase anterior.

Esta última fase agrega ao algoritmo um maior custo computacional. Contudo, a qualidade obtida pelas soluções desta fase, o custo computacional torna-se insignificante quando equiparadas.

Com base na revisão realizada pela seção anterior, serão descritas as heurísticas consideradas como as mais promissoras da literatura. Assim, os métodos descritos serão implementados para fins de comparação.

3.1. Heurísticas Construtivas Simples

Nas subseções abaixo, serão descritas as heurísticas Construtivas Simples.

- **McCormick – PF (McCormick, et al., 1989)**

Fase de Desenvolvimento

Passo 1: Para cada tarefa i , calcule o tempo total de processamento T_i que é dada pela equação (7).

$$T_i = \sum_{k=1}^m p_{i,k}, \quad \text{Para } i = 1, 2, 3, \dots, n. \quad (7)$$

Passo 2: Ordene as tarefas em ordem não-decrescente de acordo com os valores de T_i para todas as máquinas.

Passo 3: Selecione a primeira tarefa de acordo com a sequencia obtida anteriormente e faça a sequencia parcial como $\beta = (\beta_1)$ e seja $U = J - \{\beta_1\}$.

Fase de Construção da Sequencia

Passo 4: Considere $k = 1$.

Passo 5: Calcule o tempo de partida $d_{k,i}, i = 1, 2, 3, \dots, m$, para a última tarefa β_k na sequencia parcial $\beta = (\beta_1, \beta_2, \dots, \beta_k)$.

Passo 6: Para cada tarefa $j \in U$, teste a tarefa na posição $(k + 1)^{\text{ésima}}$ de β , calcule os tempos de partida de $d_{k+1,i}$ para $i = 1, 2, \dots, m$, e também calcule a soma dos tempos de espera e de bloqueio com a equação (8).

$$\delta_{j,k} = \sum_{i=1}^m (d_{k+1,i} - d_{k,i} - p_{j,i}) \quad (8)$$

Passo 7: Selecione a tarefa que possuir o menor valor de $\delta_{j,k}$ na posição $(k + 1)^{\text{ésima}}$ de β e remova a tarefa selecionada de U .

Passo 8: Faça $k = k + 1$, se $k = n - 1$ Faça Passo 9, caso contrário faça Passo 5.

Passo 9: A única tarefa restante em U , inserir na posição n^{th} de β .

No método PF, a heurística possui $n - 2$ iterações e cada iteração pode ser calculada com complexidade $O(nm)$. Então, o método de McCormick possui complexidade $(n - 2)(nm)$, isto é, $O(n^2m)$.

- **Ronconi – MM (Ronconi, 2004)**

Fase de Desenvolvimento

Passo 1: Para cada tarefa i , calcule o tempo total de processamento T_i que é dada pela equação (7).

Passo 2: Seja $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_n)$ a sequencia em ordem não-decrescente dos valores obtidos da equação (7). Insira a tarefa γ_1 na primeira posição da sequencia parcial $\beta = (\beta_1)$ e faça: $U = J - \{\gamma_1\}$.

Passo 3: Insira γ_2 na última posição da sequencia parcial $\beta = (\beta_1, \dots, \beta_n)$ e faça: $U = J - \{\gamma_2\}$.

Fase de Construção da Sequencia

Passo 4: Faça $k = 2$.

Passo 5: Para cada tarefa $j \in U$, obtenha um valor pela equação (9)

$$\alpha \sum_{i=1}^{m-1} |p_{j,i} - p_{k,i+1}| + (1 - \alpha) \sum_{l=1}^m p_{j,l} \quad (9)$$

Passo 6: A tarefa que obter o menor valor com a equação do Passo 5, insira-a na posição k de β e remova a tarefa selecionada de U .

Passo 7: Faça $k = k+1$. Se $k = n$ faça Passo 8, caso contrário, Passo 5.

Passo 8: A sequencia final será β .

A heurística apresentada por Ronconi (2004) para o cálculo de cada parâmetro da equação (9), possui complexidade de $O((m-1))(m)$ e o método possui $n-2$ iterações. Com isso, o grau de complexidade de sua heurística é de $O(m^2n)$. O parâmetro α é usado para ponderar cada termo da equação (9). Esta heurística é baseada nas propriedades do trabalho de Ronconi & Armentano (2001) na obtenção do *Makespan* e com a afirmação de que sejam duas sequencias de números inteiros $A = a_1, a_2, \dots, a_t$ e $B = b_1, b_2, \dots, b_t$ e definida pela equação (10):

$$S(A, B) = \sum_{l=1}^t \max(a_l, b_l) \quad (10)$$

assim, $S(A, B)$ é minimizado se as sequencias A e B forem ordenadas de tal maneira que o caminho de cada par (a_l, b_l) há um $k \in (1, 2, \dots, t)$, de modo que a_l e b_l correspondem ao k -ésimo menor elemento da sequencia A e B respectivamente. Ronconi explica que não é possível ordenar os tempos de processamento das tarefas envolvendo a condição de Ronconi &Armamento, então, a heurística MM é uma estratégia que visa construir uma solução onde a diferença entre eles é reduzida. O primeiro termo da expressão (9) é o módulo da diferença entre os tempos de processamento de tarefas consecutivas em máquinas consecutivas. O segundo termo prioriza as tarefas com a menor média de processamento.

- **Pan QK – wPF (Liu & Reeves (2005) adaptado por Pan QK, 2011)**

Fase de Desenvolvimento

Passo 1: Esta etapa segue os mesmos passos da Fase de Desenvolvimento do método PF.

Fase de Construção da Sequencia

Passo 4: Considere $k = 1$.

Passo 5: Calcule o tempo de partida $d_{k,i}, i = 1, 2, 3, \dots, m$, para a última tarefa β_k na sequencia parcial $\beta = (\beta_1, \beta_2, \dots, \beta_k)$.

Passo 6: Para cada tarefa $j \in U$, teste a tarefa na posição $(k + 1)$ -ésima de β , calcule os tempos de partida de $d_{k+1,i}$ para $i = 1, 2, \dots, m$, e também calcule a soma dos tempos de espera e de bloqueio com a equação (11).

$$\delta_{j,k} = \sum_{i=1}^m w_i (d_{k+1,i} - d_{k,i} - p_{j,i}) \quad (11)$$

Passo 7: Selecione a tarefa que possuir o menor valor de $\delta_{j,k}$ na posição $(k + 1)$ -ésima de β e remova a tarefa selecionada de U .

Passo 8: Faça $k = k + 1$, se $k = n - 1$ Faça Passo 9, caso contrário faça Passo 5.

Passo 9: A única tarefa restante em U , inserir na posição n -ésima de β .

Na equação (11), o termo $\delta_{j,k}$ representa a soma dos tempos de espera e bloqueio da tarefa j na posição k . O método apresentado por Pan QK (2011) é uma adaptação do método PF apresentado por McCormick. A diferença está presente na obtenção do valor no Passo 6 da descrição anterior que seria a inserção do peso w_i .

$$w_i = \frac{m}{i + \frac{k(m-i)}{n-2}} \quad (12)$$

A equação (12) representa o peso da soma dos tempos de bloqueio e espera. Este peso foi usado por Liu & Reeves (2001) e adaptado por Pan QK e o objetivo dele é diferenciar o efeito das máquinas em diferentes estágios e tarefas em diferentes posições. Este método busca reduzir os tempos de ociosidade, causados por tarefas previamente sequenciadas. O método apresentado possui a mesma complexidade do método PF, $O(n^2m)$.

3.2. Heurísticas Construtivas Melhoradas

As heurísticas que serão apresentadas nesta subseção, basicamente, são as heurísticas simples apresentadas na subseção anterior com a utilização de uma das etapas do método de Nawaz (1983), o NEH.

Abaixo segue a descrição do método.

- **Nawaz – NEH (Nawaz-Enscore-Ham, 1983)**

Fase de Desenvolvimento e Construção

Passo 1: Para cada tarefa i , calcule o tempo total de processamento T_i que é dada pela equação (7).

Passo 2: Ordene as tarefas em ordem não-crescente LPT (*Longest Processing Time*) de acordo com os valores de T_i para todas as máquinas $\beta = (\beta_1, \beta_2, \dots, \beta_k)$.

Passo 3: Selecione a primeira tarefa de acordo com a sequencia obtida anteriormente e faça a sequencia parcial como $\pi = (\beta_1)$.

Fase de Melhoramento da Sequencia

Passo 4: Considere $k = 2$.

Passo 5: Pegue a tarefa β_k de β e teste em todas as k posições possíveis de π .

Passo 6: Insira a tarefa β_k em π na sequencia parcial que resultar o menor valor de *makespan*.

Passo 7: Faça $k = k + 1$, se $k = n$ Faça Passo 8, caso contrário faça Passo 5.

Passo 8: A sequencia final π será o resultado do método.

A complexidade da heurística NEH é dada por $((n(n+1))/2)-1$. Esta equação representa o número de inserções que a heurística realiza. Como cada interação é realizada para as m máquinas, a complexidade é de $O(n^2m)$.

Como podemos observar, o método NEH é dividido em duas fases, fase de desenvolvimento e fase de construção. Nas heurísticas construtivas melhoradas, utilizam em sua grande maioria a fase de construção para o melhoramento das sequencias.

- **Ronconi – MME (Ronconi, 2004)**

Fase de Desenvolvimento e Construção

Passo 1: Obtenha a primeira ordenação através do método MM.

Fase de Melhoramento da Sequencia

Passo 2: Obtenha o melhoramento aplicando a segunda fase do método NEH (Fase de Desenvolvimento e construção).

Neste método, ao invés de utilizar a ordenação LPT para obter a solução inicial de NEH, utiliza-se o método MM.

A complexidade do método é dada pelo pior caso, o método MM possui complexidade de $O(m^2n)$ e o método NEH de complexidade $O(n^2m)$, com isso, a complexidade é de $O(n^2m)$.

- **Ronconi – PFE (Ronconi, 2004)**

Fase de Desenvolvimento e Construção

Passo 1: Obtenha a primeira ordenação através do método PF

Fase de Melhoramento da Sequencia

Passo 2: Obtenha o melhoramento aplicando a segunda fase do método NEH (Fase de Desenvolvimento e Construção).

Como ambas as heurísticas apresentam a mesma complexidade então o método PFE apresenta complexidade de $O(n^2m)$.

- **Pan QK – wPFE (Ronconi (2005) adaptado por Pan QK, 2011)**

Fase de Desenvolvimento e Construção

Passo 1: Obtenha a primeira ordenação através do método wPF.

Fase de Melhoramento da Sequencia

Passo 2: Obtenha o melhoramento aplicando a segunda fase do método NEH (Fase de Desenvolvimento e Construção).

Esta heurística segue os mesmos procedimentos da heurística PFE demonstrada anteriormente, porém, no Passo 1, na Fase de Desenvolvimento e Construção, a obtenção da sequencia é realizada com o método wPF da seção 4.1.3 e sua complexidade tem a mesma complexidade da heurística que a originou, PFE, $O(n^2m)$.

- **Pan QK – PF-NEH (Pan QK, 2011)**

Fase de Desenvolvimento e Construção

Passo 1: Obtenha a primeira ordenação através do método PF, $\beta = (\beta_1, \beta_2, \dots, \beta_n)$.

Seja a sequencia, $\pi = (\beta_1, \beta_2, \dots, \beta_{n-\lambda})$.

Fase de Melhoramento da Sequencia

Passo 2: Seja $k = n - \lambda + 1$.

Passo 3: Pegue β_k de β e teste em todas as posições k possíveis em π .

Passo 4: Insira a tarefa β_k em π na posição que resultou no menor valor de *makespan*.

Passo 5: Faça $k = k + 1$. Se $k = n$ faça Passo 6, caso contrário faça Passo 3.

Passo 6: A sequencia final π será o resultado do método.

A nova heurística de Pan QK é bem similar à heurística proposta por Ronconi. Neste caso, o procedimento de inserção proposto por Nawaz na Fase de Construção de seu método, começa de uma tarefa a partir da posição $k = n - \lambda + 1$. O número de inserções realizadas por este novo método é menor que a do método PFE porque considera somente $((2n-\lambda+1)\lambda/2)$ sequencias parciais. A complexidade computacional deste método é da ordem de $O(n^2m)$.

- **Pan QK – wPF-NEH (Pan QK, 2011)**

Fase de Desenvolvimento e Construção

Passo 1: Obtenha a primeira ordenação através do método wPF, $\beta = (\beta_1, \beta_2, \dots, \beta_n)$.

Fase de Melhoramento da Sequencia

Passo 2: Este passo segue exatamente ao procedimento de Fase de Desenvolvimento e Construção do método PF-NEH da seção 4.2.5.

A complexidade é a mesma que a complexidade que originou este método, o PF-NEH, com isso seu grau de complexidade computacional é $O(n^2m)$.

As heurísticas wPF-NEH (x) e PF-NEH (x) são variações simples das heurísticas wPF-NEH e PF-NEH respectivamente. A seguir, será realizado o passo-a-passo destas heurísticas.

- **Pan QK – PF-NEH(x) (Pan QK, 2011)**

Fase de Desenvolvimento e Construção

Passo 1: Para cada tarefa i , calcule o tempo total de processamento T_i que é dada pela equação (7).

Passo 2: Ordene as tarefas em ordem não decrescente de acordo com os valores de T_i para todas as máquinas.

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k).$$

Fase de Melhoramento da Sequencia

Passo 1: Faça $l = 1$.

Passo 2: Seja α_l a primeira tarefa a gerar a sequencia $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ usando o método PF.

Seja a sequencia, $\pi = (\beta_1, \beta_2, \dots, \beta_{n-\lambda})$.

Passo 3: Este passo segue o mesmo procedimento da Fase de Construção do método PF-NEH previamente demonstrado na seção 4.2.4.

Passo 4: Faça $l = l + 1$. Se $l = x$ faça Passo 5, caso contrário, faça Passo 2.

Passo 5: Retorne como resultado a sequencia $\pi \in (\pi^1, \pi^2, \dots, \pi^x)$ com o menor valor de *makespan*.

- **Pan QK – wPF-NEH(x) (Pan QK, 2011)**

Fase de Desenvolvimento e Construção

Passo 1: Para cada tarefa i , calcule o tempo total de processamento T_i que é dada pela equação (7).

Passo 2: Ordene as tarefas em ordem não decrescente de acordo com os valores de T_i para todas as máquinas.

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k).$$

Fase de Melhoramento da Sequencia

Passo 1: Faça $l = 1$.

Passo 2: Seja α_l a primeira tarefa a gerar a sequencia $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ usando o método PF.

Seja a sequencia, $\pi = (\beta_1, \beta_2, \dots, \beta_{n-\lambda})$.

Passo 3: Este passo segue o mesmo procedimento da Fase de Construção do método PF-NEH previamente demonstrado na seção 4.2.4.

Passo 4: Faça $l = l + 1$. Se $l = x$ faça Passo 5, caso contrário, faça Passo 2.

Passo 5: Retorne como resultado a sequencia $\pi \in (\pi^1, \pi^2, \dots, \pi^x)$ com o menor valor de *makespan*.

Este método segue os mesmos procedimentos da heurística PF-NEH(x) demonstrado anteriormente, porém, no Passo 2 na Fase de Desenvolvimento e Construção, a obtenção da sequencia β é realizada com o método wPF da seção 4.1.3.

Para ambos os métodos, wPF-NEH(x) e PF-NEH(x), a complexidade é dada pelo método que as originou, PF-NEH, porém, com a multiplicação do número de sequencias que é gerada por estas novas heurísticas. Então a complexidade computacional é na ordem de $O(xn^2m)$.

3.3. Heurísticas Construtivas Compostas

As Heurísticas Construtivas Compostas são caracterizadas pela junção de métodos heurísticos para a obtenção de resultados melhores em comparadas com as heurísticas simples e heurísticas melhoradas. As heurísticas compostas, na literatura, são as que possuem melhor qualidade de resultados relacionada ao tempo computacional necessário para obtenção dos resultados.

As heurísticas que são utilizadas comumente para este melhoramento, são as heurísticas de busca local. Métodos como Vizinhança de Inserção e Vizinhança de Permutação apresentadas por Nagano & Moccellin, et al. (2008) apresentaram bons resultados na área de *flowshop permutacional*, mas no trabalho de Pan QK (2007) é apresentado o método *Referenced Local Search (RLS)* que em composição com as heurísticas para *flowshop* com bloqueio (Pan QK, 2011), apresentou resultados significativos comparados aos valores de Ribas (2011) e nesta seção, será apresentada a heurística RLS.

- **Pan QK – RLS (Pan QK, 2007)**

Seja a sequencia $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ que deve ser melhorada e $\pi^{ref} = (\pi_1^{ref}, \pi_2^{ref}, \dots, \pi_n^{ref})$ uma sequencia referencia.

Procedimento RLS (π, π^{ref}).

Passo 1: Seja $i = 1$

Passo 2: Faça $\pi' = \pi$

Passo 3: Procure a tarefa π_i^{ref} em π' e remova-a.

Passo 4: Teste π_i^{ref} em todas as posições possíveis de π' .

Passo 5: Insira π_i^{ref} em π' na posição que resulta o menor valor de *Makespan*.

Passo 6: Se $f(\pi') < f(\pi)$ então $\pi = \pi'$

Passo 7: Faça $i = i + 1$. Se $i > n$ faça Passo 8, caso contrário, faça Passo 2.

Passo 8: Retorne π como solução do problema.

O padrão que segue para a utilização das heurísticas compostas nas heurísticas simples e melhoradas segue abaixo.

Fase de Desenvolvimento

Passo 1: Utilizam-se comumente as fases de desenvolvimento citadas nas seções acima (LPT ou ordenação não-decrescente).

Fase de Construção

Passo 2: Nesta etapa, utilizam-se métodos simples (PF, MM, wPF).

Fase de Melhoramento

Passo 3: Utilizam-se métodos de Inserção de tarefas (Segunda fase do Algoritmo NEH).

Fase Compositiva

Passo 4: Aplicam-se métodos de busca local (Neste trabalho, a heurística RLS).

Seguindo estes passos, obtêm-se as Heurísticas Construtivas Compostas: MM_{LS} , PF_{LS} , wPF_{LS} , MME_{LS} , PFE_{LS} , $wPFE_{LS}$, $PF - NEH_{LS}$, $wPF - NEH_{LS}$, $PF - NEH(x)_{LS}$, $wPF - NEH(x)_{LS}$.

Tabela 2: Fases das Heurísticas

	Fases				
	Desenvolvimento	Construção	Melhoramento	Composição	
Heurísticas Simples	PF MM wPF	Ordenação Ordenação Ordenação	PF MM wPF		
Heurísticas Melhoradas	NEH PFE MME wPFE PF-NEH wPF-NEH PF-NEH(x) wPF-NEH(x)	Ordenação Ordenação Ordenação Ordenação Ordenação Ordenação Ordenação	LPT PF MM wPF PF wPF PF wPF	NEH NEH-2 NEH-2 NEH-2 NEH-2 NEH-2 NEH-2 NEH-2	
Heurísticas Compostas	PF-NEH wPF-NEH PF-NEH(x) wPF-NEH(x)	Ordenação Ordenação Ordenação Ordenação	PF wPF PF wPF	NEH-2 NEH-2 NEH-2 NEH-2	RLS RLS RLS RLS

A Tabela 2 apresenta de forma exemplificada a divisão das fases de cada método.

O índice NEH-2 representa a utilização da segunda fase do método NEH (fase de inserção).

Os métodos presentes na tabela são os que apresentaram resultados com qualidade superiores na literatura para a solução de problemas no ambiente *Flowshop com bloqueio* através de Heurísticas Construtivas.

3.4. Exemplos numéricos dos Métodos

Nesta seção, serão apresentados exemplos numéricos dos métodos estudados para uma demonstração de seus passos e cálculos para a obtenção de resultados. A Tabela 3 apresenta os valores de processamento de cinco tarefas em cinco máquinas que serão utilizados a seguir.

Tabela 3: Dados para os exemplos numéricos

	J_1	J_2	J_3	J_4	J_5
m_1	6	7	11	4	1
m_2	3	1	7	8	9
m_3	11	2	4	2	1
m_4	2	13	8	9	4
m_5	2	2	3	4	5

McCormick – PF (McCormick, et al., 1989)

Passo 1: Utilizando a equação (7), as tarefas serão ordenadas em ordem não-decrescente de acordo com os valores obtidos através da equação.

Soma:

$$J_1 = 24.$$

$$J_2 = 25.$$

$$J_3 = 33.$$

$$J_4 = 27.$$

$$J_5 = 20.$$

Passo 2: Sequencia não-descrescente: $J_5 - J_1 - J_2 - J_4 - J_3$.

Passo 3: Selecione e insira a primeira tarefa J_5 na primeira posição da nova sequencia e retire-a da sequencia do Passo 2.

Passo 4: Teste todas as tarefas restantes da sequencia do Passo 2 na ultima posição da nova sequencia do Passo 3 e obtenha um valor com a equação (8).

$$J_5 - J_1 = 20.$$

$$J_5 - J_2 = 12.$$

$$J_5 - J_3 = 29.$$

$$J_5 - J_4 = 26.$$

$$\text{Menor: } J_5 - J_2 = 12.$$

Selecione como nova sequencia aquela que obtiver o menor valor entre as possibilidades testadas e remova a tarefa correspondente, no caso J_2 , do conjunto de tarefas do Passo 2. As tarefas restantes são: $J_1 - J_4 - J_3$.

Teste todas as tarefas restantes do Passo 2 na ultima posição da nova sequencia ($J_5 - J_2$) e para cada variação, calcule o valor através da equação (8).

$$J_5 - J_2 - J_1 = 13.$$

$$J_5 - J_2 - J_3 = 37.$$

$$J_5 - J_2 - J_4 = 21.$$

$$\text{Menor: } J_5 - J_2 - J_1 = 13.$$

Remova a tarefa J_1 do conjunto do Passo 2 seguindo o mesmo critério de seleção da segunda tarefa da nova sequencia e faça $J_5 - J_2 - J_1$ a nova sequencia. Na nova sequencia, teste na ultima posição todas as tarefas restantes do conjunto do Passo 2 novamente e calcule os valores pela equação (8).

$$J_5 - J_2 - J_1 - J_3 = 30.$$

$$J_5 - J_2 - J_1 - J_4 = 10.$$

$$\text{Menor: } J_5 - J_2 - J_1 - J_4 = 10.$$

Continuando o mesmo procedimento, remova a tarefa J_4 do conjunto restante e coloque-a na ultima posição da nova sequencia, gerando: $J_5 - J_2 - J_1 - J_4$. A última tarefa restante do Passo 2 é inserida na última posição da nova sequencia, obtendo a solução final:

Sequencia Final: $J_5 - J_2 - J_1 - J_4 - J_3$.

Makespan = 53.

Ronconi – MM (Ronconi, 2004)

Utilizando os dados da Tabela 3, segue o exemplo numérico do método MM.

Passo 1: Calcule a soma dos valores pela equação (7).

$$J_1 = 24.$$

$$J_2 = 25.$$

$$J_3 = 33.$$

$$J_4 = 27.$$

$$J_5 = 20.$$

Passo 2: Selecione a tarefa com o menor valor adquirido através da equação (7) e insira na primeira posição da nova sequência, no caso a tarefa J_5 e remova-a do conjunto do conjunto de tarefas do Passo 1. Insira na ultima posição da nova sequência entre as tarefas restantes do conjunto, a tarefa que possuir o menor valor adquirido pela equação (7), no caso, a tarefa J_1 .

Passo 3: Teste todas as tarefas restantes na nova sequência através da equação (9)

$$J_5 - J_2 = 15,25.$$

$$J_5 - J_3 = 16,5.$$

$$J_5 - J_4 = 20,25.$$

Nova Sequencia Parcial: $J_5 - J_2 = 15,25$.

Insira a tarefa que obteve o menor valor na segunda posição da nova sequência, J_2 e retire ela do conjunto de tarefas do Passo 1. A nova sequência: $J_5 - J_2 - \dots - J_1$ e as tarefas restantes a serem inseridas são: J_3 e J_4 .

Novamente, teste as tarefas restantes na terceira posição da nova sequencia seguindo o critério da equação (9).

$$J_5 - J_2 - J_3 = 30,75.$$

$$J_5 - J_2 - J_4 = 27.$$

Nova Sequencia Parcial: $J_5 - J_2 - J_4 = 27$.

Insira a tarefa J_4 na terceira posição da nova sequencia e remova-a do conjunto de tarefas do Passo 1 por resultar no menor valor da equação (9) e na quarta posição (única disponível), insira a tarefa restante do Passo 1 na nova sequencia, gerando:

Sequencia Final: $J_5 - J_2 - J_4 - J_3 - J_1$.

Makespan = 52.

Pan QK – wPF (Pan QK, 2011)

Utilizando os dados da Tabela 3, segue o exemplo numérico do método wPF.

Passo 1: Calcule a soma dos valores pela equação (7).

$$J_1 = 24.$$

$$J_2 = 25.$$

$$J_3 = 33.$$

$$J_4 = 27.$$

$$J_5 = 20.$$

Passo 2: Sequencia não-descrescente: $J_5 - J_1 - J_2 - J_4 - J_3$.

Passo 3: Selecione e insira a primeira tarefa J_5 na primeira posição da nova sequencia e retire-a da sequencia do Passo 2.

Passo 4: Teste todas as tarefas restantes da sequencia do Passo 2 na ultima posição da nova sequencia do Passo 3 e obtenha um valor com a equação (11).

$$J_5 - J_1 = 25,54.$$

$$J_5 - J_2 = 15,01.$$

$$J_5 - J_3 = 34,47.$$

$$J_5 - J_4 = 35,02.$$

Nova Sequencia Parcial: $J_5 - J_2 = 15,01$.

Selecione como nova sequencia aquela que obtiver o menor valor entre as possibilidades testadas e remova a tarefa correspondente, no caso J_2 , do conjunto de tarefas do Passo 2. As tarefas restantes são: $J_1 - J_4 - J_3$.

Teste todas as tarefas restantes do Passo 2 na ultima posição da nova sequencia ($J_5 - J_2$) e para cada variação, calcule o valor através da equação (11).

$$J_5 - J_2 - J_1 = 15.$$

$$J_5 - J_2 - J_3 = 41,78.$$

$$J_5 - J_2 - J_4 = 23,44.$$

Nova Sequencia Parcial: $J_5 - J_2 - J_1 = 15$.

Remova a tarefa J_1 do conjunto do Passo 2 seguindo o mesmo critério de seleção da segunda tarefa da nova sequencia e faça $J_5 - J_2 - J_1$ a nova sequencia. Na nova sequencia, teste na ultima posição todas as tarefas restantes do conjunto do Passo 2 novamente e calcule os valores pela equação (11).

$$J_5 - J_2 - J_1 - J_3 = 30.$$

$$J_5 - J_2 - J_1 - J_4 = 10.$$

Nova Sequencia Parcial: $J_5 - J_2 - J_1 - J_4 = 10$.

Continuando o mesmo procedimento, remova a tarefa J_4 do conjunto restante e coloque-a na ultima posição da nova sequencia, gerando: $J_5 - J_2 - J_1 - J_4$. A ultima tarefa restante do Passo 2, insira na última posição da nova sequencia, obtendo a solução final:

Sequencia Final: $J_5 - J_2 - J_1 - J_4 - J_3$.

Makespan = 53.

Nawaz – NEH (Nawaz-Enscore-Ham, 1983)

Utilizando os dados da Tabela 3, segue o exemplo numérico do método NEH.

Passo 1: Calcule a soma dos valores pela equação (7).

$$J_1 = 24.$$

$$J_2 = 25.$$

$$J_3 = 33.$$

$$J_4 = 27.$$

$$J_5 = 20.$$

Passo 2: Ordene as tarefas em ordem não-crescente de acordo com a soma dos processamentos das tarefas obtidos através da equação (7).

$$J_3 - J_4 - J_2 - J_1 - J_5.$$

Passo 3: Selecione as duas primeiras tarefas e encontre a melhor disposição delas, isto é, a melhor sequencia parcial será a que possuir o menor valor de *makespan* obtido através das equações (2) a (6).

$$J_3 - J_4 = 43 .$$

$$J_4 - J_3 = 37.$$

Nova Sequencia Parcial: $J_4 - J_3 = 37$.

Selecione como nova sequência aquele que apresentar o menor valor de *makespan* parcial, no caso a sequência $J_4 - J_3$. Selecione a terceira tarefa da sequencia do Passo 2 e insira em todas as posições possíveis da nova sequencia.

$$J_2 - J_4 - J_3 = 44.$$

$$J_4 - J_2 - J_3 = 47.$$

$$J_4 - J_3 - J_2 = 49.$$

Nova Sequencia Parcial: $J_2 - J_4 - J_3 = 44$.

Com o mesmo critério de seleção realizado para a segunda tarefa, selecione a sequencia que obteve o menor valor de *makespan* como a nova sequencia, no caso: $J_2 - J_4 - J_3$. Para a seleção da quarta tarefa, realiza-se o mesmo procedimento seqüencial, insira a quarta tarefa da sequencia do Passo 2 em todas as possibilidades de posições da nova sequencia com três tarefas.

$$J_1 - J_2 - J_4 - J_3 = 55.$$

$$J_2 - J_1 - J_4 - J_3 = 50.$$

$$J_2 - J_4 - J_1 - J_3 = 52.$$

$$J_2 - J_4 - J_3 - J_1 = 48.$$

Nova Sequencia Parcial: $J_2 - J_4 - J_3 - J_1 = 48$.

Realizando os mesmos procedimentos para a obtenção da terceira tarefa da nova sequencia, é inserida a quarta tarefa do Passo 2 em todas as possibilidades de posições na nova sequencia. Com isso, obtemos a sequencia nova $J_2 - J_4 - J_3 - J_1$ e ela será utilizada para a inserção da última tarefa restante na sequencia do Passo 2.

$$J_5 - J_2 - J_4 - J_3 - J_1 = 52.$$

$$J_2 - J_5 - J_4 - J_3 - J_1 = 54.$$

$$J_2 - J_4 - J_5 - J_3 - J_1 = 56.$$

$$J_2 - J_4 - J_3 - J_5 - J_1 = 56.$$

$$J_2 - J_4 - J_3 - J_1 - J_5 = 55.$$

Nova Sequencia Final: $J_5 - J_2 - J_4 - J_3 - J_1 = 52$.

Com a inserção em todas as possibilidades da ultima tarefa do Passo 2 em todas as posições da nova sequencia, obtém-se a sequencia final do método com

$$Makespan = 52$$

Os métodos PFE, MME, wPFE, são similares ao método NEH. Ao invés de utilizar a solução inicial *LPT*, utiliza-se a PF, MM e wPF respectivamente para a obtenção dos resultados dos métodos, ou seja, utiliza-se somente o Passo de inserção apresentado pelo método NEH.

Pan QK – PF-NEH (Pan QK, 2011)

Passo 1: Utilizando a solução obtida pelo método PF, será usado o exemplo numérico anteriormente demonstrado como ponto de partida.

$$J_5 - J_2 - J_1 - J_4 - J_3.$$

Passo 2: como há um menor numero de tarefas, o valor de λ que será utilizado para fins demonstrativos, terá o valor 3.

Com o valor de $\lambda = 3$, a inserção das tarefas será realizada a partir da terceira tarefa da sequencia gerada pelo método PF, idêntico ao processo de inserção do método NEH a partir da terceira tarefa, como segue abaixo:

$$J_1 - J_5 - J_2 = 41.$$

$$J_5 - J_1 - J_2 = 41.$$

$$J_5 - J_2 - J_1 = 34.$$

Nova Sequencia Parcial: $J_5 - J_2 - J_1$.

Inserção da quarta tarefa da sequencia do Passo 1 na Nova Sequencia Parcial:

$$J_4 - J_5 - J_2 - J_1 = 45.$$

$$J_5 - J_4 - J_2 - J_1 = 46.$$

$$J_5 - J_2 - J_4 - J_1 = 43.$$

$$J_5 - J_2 - J_1 - J_4 = 45.$$

Nova Sequencia Parcial: $J_5 - J_2 - J_4 - J_1$.

Inserção da Quinta tarefa da sequencia do Passo 1 na Nova Sequencia Parcial:

$$J_3 - J_5 - J_2 - J_4 - J_1 = 62.$$

$$J_5 - J_3 - J_2 - J_4 - J_1 = 59.$$

$$J_5 - J_2 - J_3 - J_4 - J_1 = 55.$$

$$J_5 - J_2 - J_4 - J_3 - J_1 = 52.$$

$$J_5 - J_2 - J_4 - J_1 - J_3 = 55.$$

Sequencia Final: $J_5 - J_2 - J_4 - J_3 - J_1$.

Makespan = 52.

O método PF-NEH (x) e o método wPF-NEH (x) e suas variações, a diferença está no número de sequencias que são geradas, ou seja, são x repetições do método anteriormente exemplificado. A mudança está na construção do método PF e do wPF respectivamente. O índice x representa a posição da tarefa do Passo 2 que deve ser inserida na primeira posição da nova sequencia parcial do Passo 3 e o restante da construção da nova sequencia segue o mesmo padrão de inserção e cálculo dos valores que são utilizados como critério.

Pan QK – RLS (Pan QK, 2007)

Passo 1: Para a ilustração do procedimento desta heurística, será utilizada a sequencia aleatória referente à Tabela 3 abaixo:

$$J_5 - J_2 - J_4 - J_1 - J_3, \text{ makespan} = 55.$$

Passo 2: Basicamente, este método funciona semelhantemente ao método de inserção do NEH, porém, a inserção não é realizada em uma sequencia parcial e sim, em uma sequencia com todas as tarefas.

Primeiramente realiza-se a busca da primeira tarefa na sequencia do Passo 1 na sequencia nova, no caso do exemplo representada pela tarefa J_5 e em seguida, tarefa é inserida em todas as possibilidades de posições na nova sequencia.

$$J_2 - J_5 - J_4 - J_1 - J_3 = 58.$$

$$J_2 - J_4 - J_5 - J_1 - J_3 = 61.$$

$$J_2 - J_4 - J_1 - J_5 - J_3 = 56.$$

$$J_2 - J_4 - J_1 - J_3 - J_5 = 58.$$

Os valores são comparados com a sequencia original (Passo 1). Como os valores de *makespan* obtidos com a inserção foram maiores a sequencia anterior, a sequencia nova parcial continuará sendo a sequencia do Passo 1.

Sequencia nova parcial: $J_5 - J_2 - J_4 - J_1 - J_3$.

Em seguida, busca-se na sequencia nova parcial a segunda tarefa correspondente a sequencia do Passo 1 e ela é inserida em todas as posições possíveis na sequencia nova parcial.

$$J_2 - J_5 - J_4 - J_1 - J_3 = 58.$$

$$J_5 - J_4 - J_2 - J_1 - J_3 = 57.$$

$$J_5 - J_4 - J_1 - J_2 - J_3 = 58.$$

$$J_5 - J_4 - J_1 - J_3 - J_2 = 63.$$

Os valores obtidos com a inserção da segunda tarefa, não foram inferiores a sequência nova parcial anterior, com isso a sequência nova parcial continua sendo a mesma.

Sequencia nova parcial: $J_5 - J_2 - J_4 - J_1 - J_3$.

Para a terceira tarefa, realiza-se o mesmo procedimento que as tarefas anteriores.

Segue as variações das inserções na sequencia nova parcial.

$$J_4 - J_5 - J_2 - J_1 - J_3 = 60.$$

$$J_5 - J_4 - J_2 - J_1 - J_3 = 57.$$

$$J_5 - J_2 - J_1 - J_4 - J_3 = 53.$$

$$J_5 - J_2 - J_1 - J_3 - J_4 = 59.$$

Neste caso, a sequência que apresentou o menor valor de *makespan* foi a sequencia com o valor de *makespan* de 53 e a sequencia nova parcial é atualizada pela sequencia que obteve o menor valor de *makespan* entre as inserções da terceira tarefa.

Sequencia nova parcial: $J_5 - J_2 - J_1 - J_4 - J_3$.

A quarta tarefa, J_1 , do Passo 1, é procurada na sequencia nova parcial e é testada em todas as posições na sequencia nova parcial. Como segue abaixo:

$$J_1 - J_5 - J_2 - J_4 - J_3 = 59.$$

$$J_5 - J_1 - J_2 - J_4 - J_3 = 59.$$

$$J_5 - J_2 - J_4 - J_1 - J_3 = 55.$$

$$J_5 - J_2 - J_4 - J_3 - J_1 = 52.$$

Com a inserção da quarta tarefa, é obtido um valor de *makespan* inferior ao anterior, com o valor de *makespan* de 52 e, com isso, a sequencia nova parcial é atualizada.

Sequencia nova parcial: $J_5 - J_2 - J_4 - J_3 - J_1$.

A quinta tarefa da sequencia do Passo 1 é buscada na Sequencia nova parcial e testada em todas as posições nesta sequencia.

$$J_3 - J_5 - J_2 - J_4 - J_1 = 62.$$

$$J_5 - J_3 - J_2 - J_4 - J_1 = 59.$$

$$J_5 - J_2 - J_3 - J_4 - J_1 = 55.$$

$$J_5 - J_2 - J_4 - J_1 - J_3 = 55.$$

Com a inserção, os valores obtidos de *Makespan* foram superiores a sequência obtida com a inserção da quarta tarefa então, a sequência Final será a sequência nova parcial obtida com a quarta tarefa.

Sequencia Final: $J_5 - J_2 - J_4 - J_3 - J_1$.

Makespan = 52.

4. Experimentação Computacional

Os métodos Heurísticos descritos foram programados em linguagem de programação C no sistema operacional Linux utilizando uma IDE, NetBeans, em um microprocessador Intel Core 2 Duo (2.2 GHz) e 4 GB de RAM. Para a experimentação, utilizou o banco de dados de Taillard que totaliza em 120 problemas com o número de máquinas $m \in \{5, 10, 20\}$ e o número de tarefas $n \in \{20, 50, 100, 200, 500\}$ e cada combinação $n \times m$ corresponde a 10 problemas. Os valores de tempos de processamentos utilizados são números inteiros gerados aleatoriamente que estão na faixa de [1,99].

As estatísticas usadas para a avaliação do desempenho dos métodos foram o Desvio Relativo Médio e Porcentagem de Sucesso. O Desvio Relativo Médio (DR_h), obtido através da equação (12) quantifica o desvio que o método h obtém com relação ao melhor tempo total da programação obtido para o mesmo problema. O segundo é definido como o quociente entre o número de problemas no qual o método obteve o melhor tempo de processamento total e o número total de problemas resolvidos, quando os métodos possuem o mesmo valor (melhor), os índices são incrementados para ambos.

$$DR_h = \frac{\bar{F}_h - \bar{F}_0}{\bar{F}_0} \quad (12)$$

Onde \bar{F}_h é o tempo total da programação obtido pelo método h , \bar{F}_0 o melhor tempo total da programação obtido entre os métodos.

Para a Porcentagem de Sucesso, é evidente que o método que apresentar o maior valor possui o melhor desempenho neste índice e para o Desvio Relativo Médio, o método que possuir o menor valor, será o método que apresenta o melhor desempenho nesta estatística.

Os resultados serão analisados de acordo com a classe estabelecida neste trabalho: primeiro serão consideradas as Heurísticas Construtivas Simples, em seguida as Heurísticas Construtivas Melhoradas e as Heurísticas Construtivas Compostas. Com isso, será realizada a análise estatística para todos os métodos descritos para cada determinada categoria. Após esta análise, será realizado o estudo para todos os métodos em conjunto, sem levar em consideração a categoria pertencente.

Os dados das tabelas que serão apresentadas, as estatísticas Porcentagem de Sucesso e Desvio Relativo Médio são dadas em porcentagem (%) e o tempo computacional, em milissegundos (ms).

5. Análise dos Resultados

Heurísticas Simples

Para a literatura *Flowshop* com bloqueio, neste trabalho, foram estudadas as heurísticas simples MM, PF e wPF. Os resultados da experimentação computacional são ilustrados nas tabelas Tabela 4 à Tabela 6 e Figura 2 à Figura 5.

Tabela 4: Porcentagem de Sucesso para as Heurísticas Simples

Tarefas (n)	Máquinas (m)	MM	PF	wPF
20	5	10,00	50,00	60,00
20	10	30,00	10,00	70,00
20	20	10,00	10,00	90,00
50	5	10,00	60,00	30,00
50	10	0,00	30,00	70,00
50	20	0,00	0,00	100,00
100	5	0,00	40,00	60,00
100	10	0,00	60,00	40,00
100	20	0,00	30,00	70,00
200	10	0,00	60,00	40,00
200	20	0,00	30,00	70,00
500	20	0,00	80,00	20,00
Média		5,00	38,33	60,00

Tabela 5: Desvio Relativo Médio para as Heurísticas Melhoradas

Tarefas (n)	Máquinas (m)	MM	PF	wPF
20	5	4,16	1,17	1,23
20	10	3,34	2,57	0,24
20	20	2,74	3,09	0,23
50	5	4,89	0,72	0,74
50	10	6,44	1,47	0,38
50	20	4,42	2,69	0,00
100	5	4,57	0,46	0,35
100	10	6,39	0,33	0,66
100	20	6,47	1,48	0,21
200	10	7,75	0,27	0,58
200	20	7,86	0,93	0,22
500	20	8,47	0,08	0,40
Média		5,62	1,27	0,44

Tabela 6: Tempo Computacional para as Heurísticas Simples

Tarefas (n)	Máquinas (m)	MM	PF	wPF
20	5	0	0	0
20	10	0	0	0
20	20	0	0	10
50	5	0	50	20
50	10	0	100	70
50	20	0	200	140
100	5	0	370	210
100	10	30	690	510
100	20	40	1380	1050
200	10	160	5010	3780
200	20	230	9990	7600
500	20	2580	149670	109160
Soma		3040	167460	122550

A Figura 2 ilustra a Tabela 4 para Porcentagem de Sucesso para cada classe de problema.

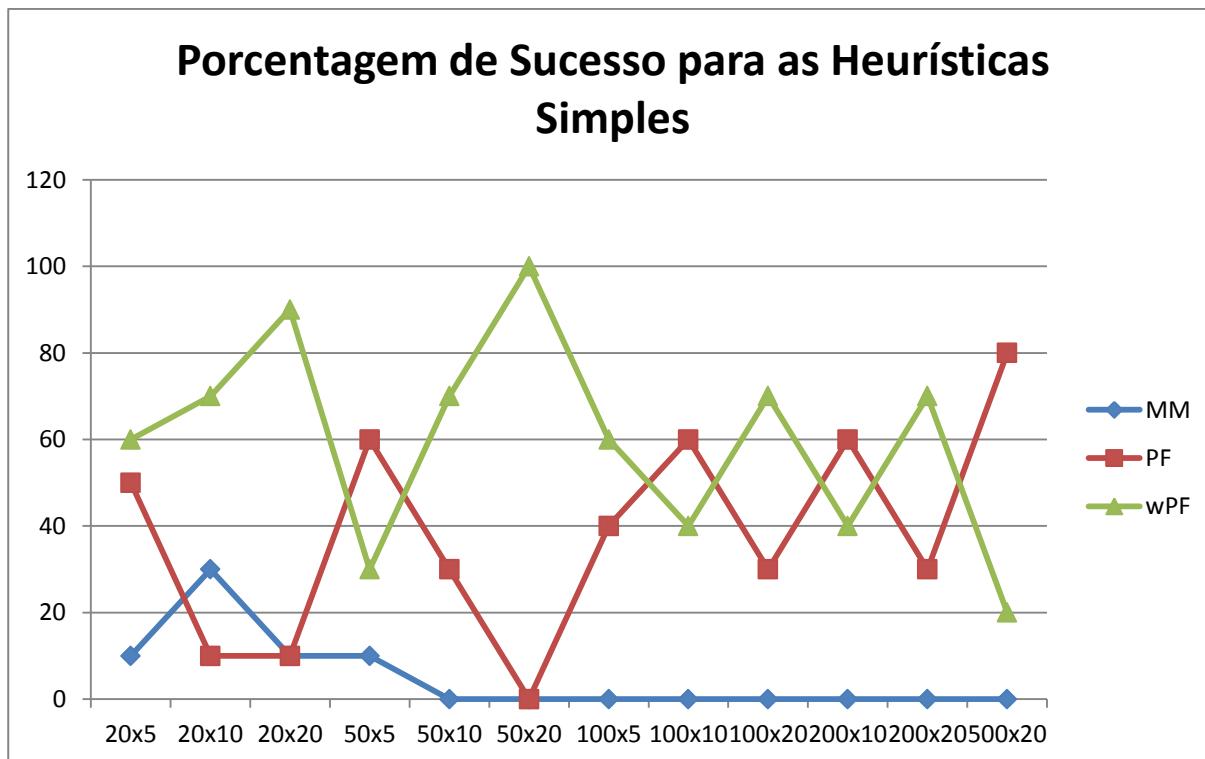


Figura 2: Porcentagem de Sucesso para as Heurísticas Simples

A Figura 3 ilustra a Tabela 4 para a média da Porcentagem de Sucesso.

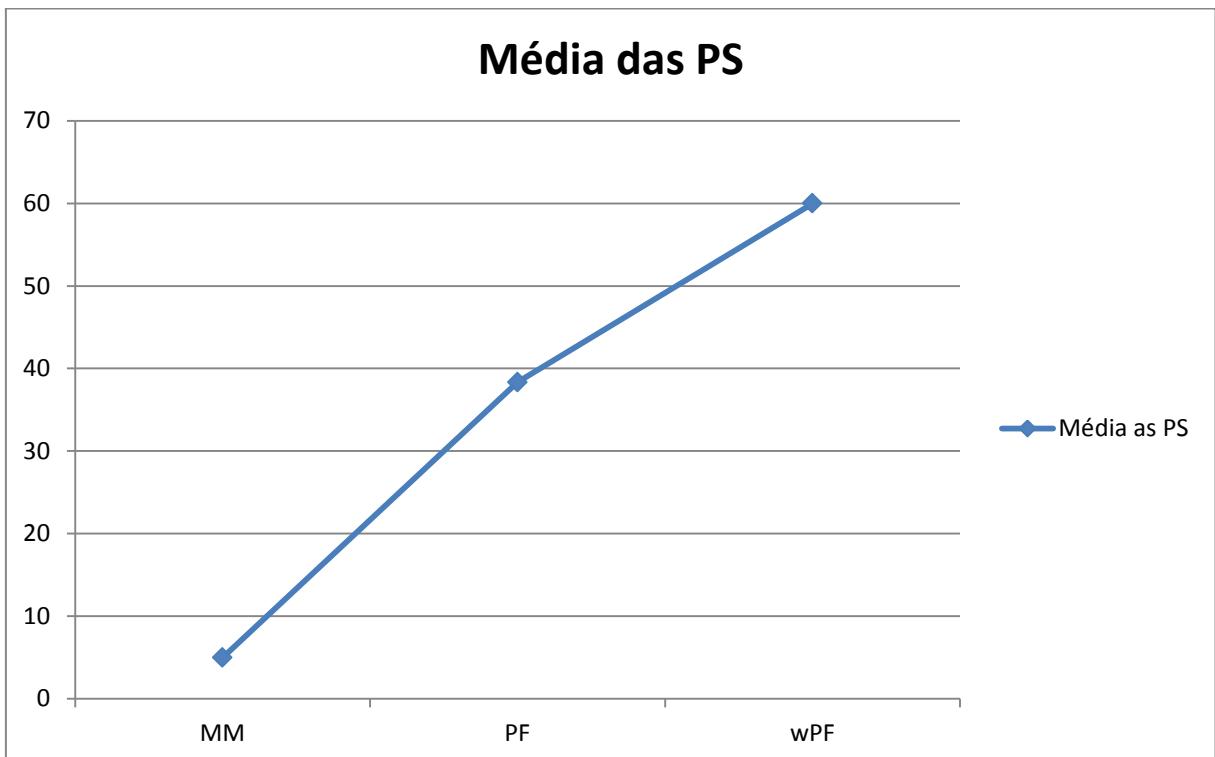


Figura 3: Média da PS para Heurísticas Simples

A Figura 4 ilustra a Tabela 5 para o Desvio Médio Relativo para cada porte de problema.

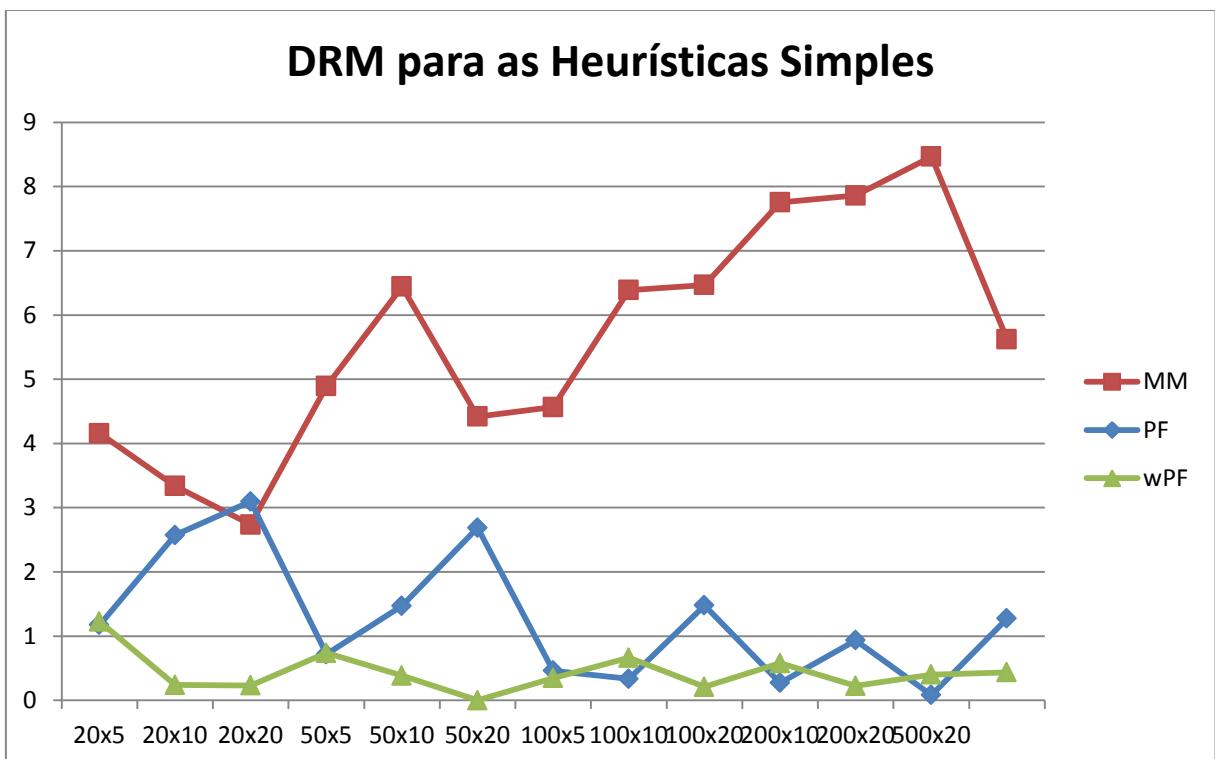


Figura 4: Desvio Relativo Médio para as Heurísticas Simples

A Figura 5 ilustra a Tabela 5 para o a média do Desvio Médio Relativo.

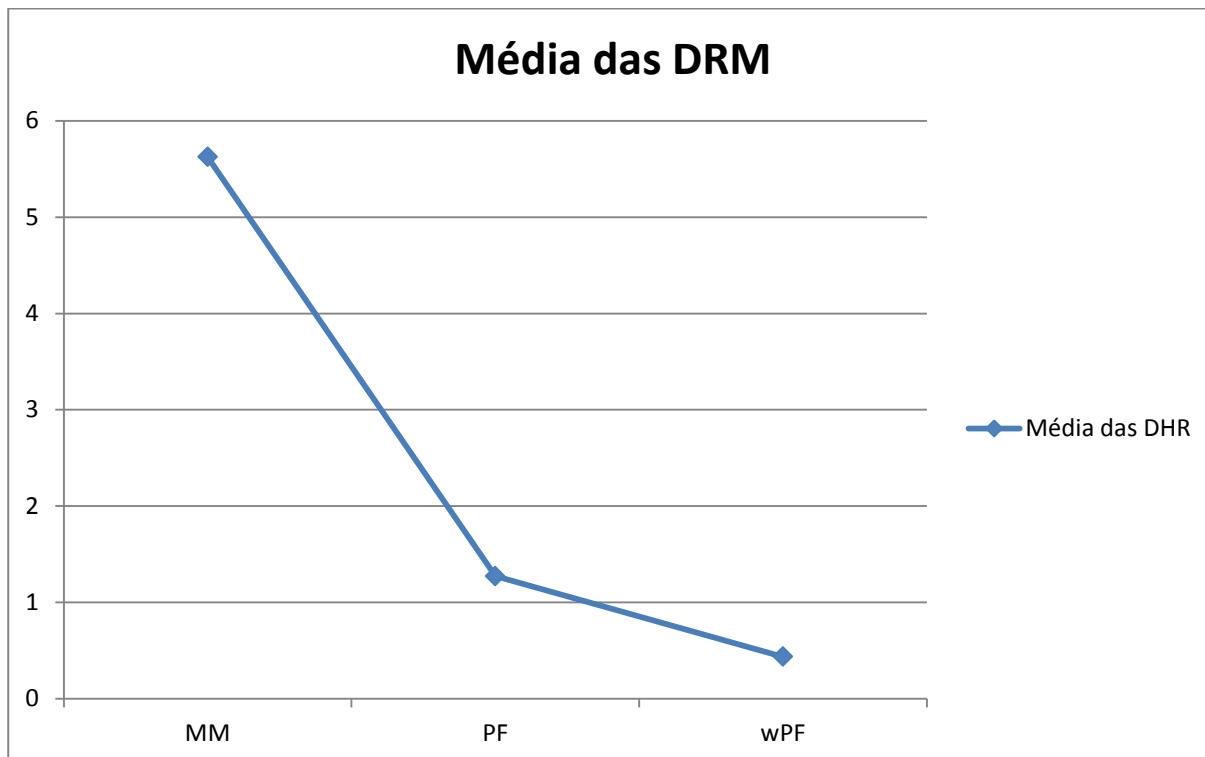


Figura 5: Média da DRM para Heurísticas Simples

A Figura 6 ilustra a Tabela 6 para o tempo computacional para as Heurísticas Simples

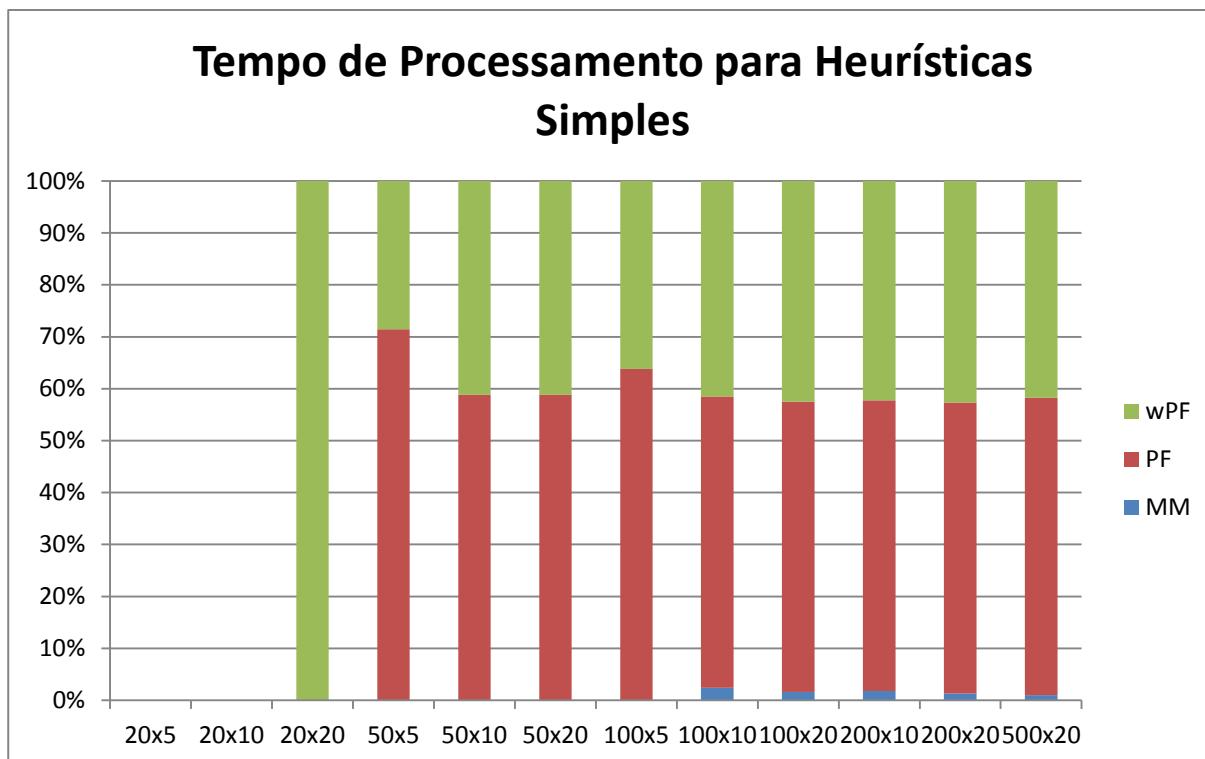


Figura 6: Tempo de Processamento Relativo para as Heurísticas Simples

Com o auxílio da Figura 2, podemos analisar a Porcentagem de Sucesso para as Heurísticas Simples. É evidente que o método wPF apresentou um desempenho superior a heurística PF em problemas de pequeno porte. Para os problemas de grande porte, houve uma variação de superioridade entre os métodos. A média da estatística, ilustrada pela Figura 3, é possível ver com clareza que o método wPF é superior sobre MM e PF.

O Desvio Relativo Médio, com a ilustração da Tabela 5 na Figura 4, o método wPF possui o menor valor na maioria dos portes de problema e, na Figura 5, é possível analisar que a média do Desvio Relativo Médio é inferior entre os três métodos, porém, bem próxima a do método PF. A heurística MM apresentou o maior valor em quase todos os portes de problema, apenas inferior com relação ao método PF para problemas de 20 tarefas e 20 máquinas.

De acordo com a Figura 6, podemos avaliar o tempo computacional relativo gasto. Os tempos dos métodos wPF e PF são bem próximos. Com isso, o wPF é superior que o método PF por fornecer melhores soluções sem um aumento no tempo computacional.

Heurísticas Melhoradas

As Heurísticas Melhoradas da literatura que foram implementadas são: NEH, PFE, MME, wPFE, PF-NEH, wPF-NEH, PF-NEH(2), PF-NEH(5), wPF-NEH(2), wPF-NEH(5). As heurísticas PF-NEH e wPF-NEH, segundo Pan QK, o valor do termo λ que maximiza o desempenho dos métodos, é o valor correspondente a 25. Para o restante das heurísticas, o valor $\lambda = 20$ mostrou-se superior em desempenho quando comparado com outros valores de λ . Para o termo x , os resultados foram gerados para $x = 1, 2$ e 5 de acordo com os dados apresentados por Pan QK.

As tabelas Tabela 7 à Tabela 9 representam os resultados computacionais obtidos para as Heurísticas Melhoradas. As figuras Figura 7 à Figura 11 ilustram os dados das tabelas.

Tabela 7: Porcentagem de Sucesso para as Heurísticas Melhoradas

Tarefas (n)	Máquinas (m)	NEH	PFE	MME	wPFE	PF-NEH	wPF- NEH	PF- NEH(2)	PF- NEH(5)	wPF- NEH(2)	wPF- NEH(5)
20	5	0	20	10	10	20	10	20	50	10	50
20	10	10	20	20	20	20	20	20	30	20	50
20	20	30	0	50	0	0	0	0	0	0	20
50	5	0	10	0	0	10	10	10	40	30	50
50	10	0	10	0	0	0	10	10	40	20	50
50	20	10	10	30	0	0	10	0	0	10	50
100	5	0	0	0	0	30	20	30	50	20	50
100	10	0	0	0	0	10	20	10	60	30	40
100	20	0	0	10	0	0	20	0	10	40	80
200	10	0	0	0	0	20	0	40	90	10	10
200	20	0	0	0	0	10	40	10	20	50	80
500	20	0	0	0	0	20	0	30	70	30	40
Média		4,17	5,83	10,00	2,50	11,67	13,33	15,00	38,33	22,50	47,50

Tabela 8: Desvio relativo Médio para as Heurísticas Melhoradas

Tarefas (n)	Máquinas (m)	NEH	PFE	MME	wPFE	PF-NEH	wPF- NEH	PF- NEH(2)	PF- NEH(5)	wPF- NEH(2)	wPF- NEH(5)
20	5	2,32	2,41	2,39	2,37	2,41	2,37	1,59	0,68	1,87	0,29
20	10	1,69	2,57	1,84	2,14	2,57	2,14	1,04	0,45	0,96	0,42
20	20	0,63	2,66	0,52	2,19	2,66	2,19	1,93	1,10	1,65	0,76
50	5	4,43	2,81	2,04	2,75	1,16	1,91	0,95	0,33	1,19	0,21
50	10	2,95	1,93	1,93	2,26	2,02	1,64	1,35	0,74	0,73	0,17
50	20	2,65	2,10	0,77	1,36	3,36	1,66	2,62	1,78	1,34	0,73
100	5	6,15	4,72	4,58	4,17	0,78	0,92	0,29	0,15	0,88	0,43
100	10	5,63	3,67	3,50	4,06	0,95	1,31	0,75	0,16	1,11	0,59
100	20	2,26	1,80	1,18	1,70	1,56	0,96	1,29	0,97	0,22	0,06
200	10	6,74	5,20	5,00	6,30	0,57	0,97	0,39	0,04	0,76	0,51
200	20	4,55	3,17	3,02	3,24	0,92	0,49	0,67	0,37	0,34	0,18
500	20	5,82	4,49	4,47	5,09	0,30	0,54	0,15	0,04	0,33	0,29
Média		3,82	3,13	2,60	3,14	1,61	1,43	1,09	0,57	0,95	0,39

Tabela 9: Tempo Computacional para as Heurísticas Melhoradas

Tarefas (n)	Máquinas (m)	NEH	PFE	MME	wPFE	PF-NEH	wPF- NEH	PF- NEH(2)	PF- NEH(5)	wPF- NEH(2)	wPF- NEH(5)
20	5	10	0	10	0	0	30	20	40	20	50
20	10	0	0	20	0	0	0	40	90	30	90
20	20	10	10	40	40	20	50	90	220	90	200
50	5	70	110	140	110	110	80	200	540	180	480
50	10	160	230	280	200	220	170	420	1100	400	1030
50	20	270	540	540	480	450	410	900	2270	810	2090
100	5	570	910	1010	870	620	530	1220	3140	1120	2740
100	10	1150	1830	1980	1680	1220	1100	2470	6200	2150	5470
100	20	2330	3740	3940	3460	2500	2210	4970	12530	4370	11070
200	10	9100	14150	15240	12960	7480	6250	14820	37280	12390	31080
200	20	18040	28300	29520	25720	14900	12430	29820	74480	24820	62090
500	20	275880	428530	453100	389600	181880	141110	362660	908220	280780	702480
Soma		307590	478350	505820	435120	209400	164370	417630	1046110	327160	818870

A Figura 7 é uma ilustração dos dados para porcentagem de sucesso da Tabela 7.

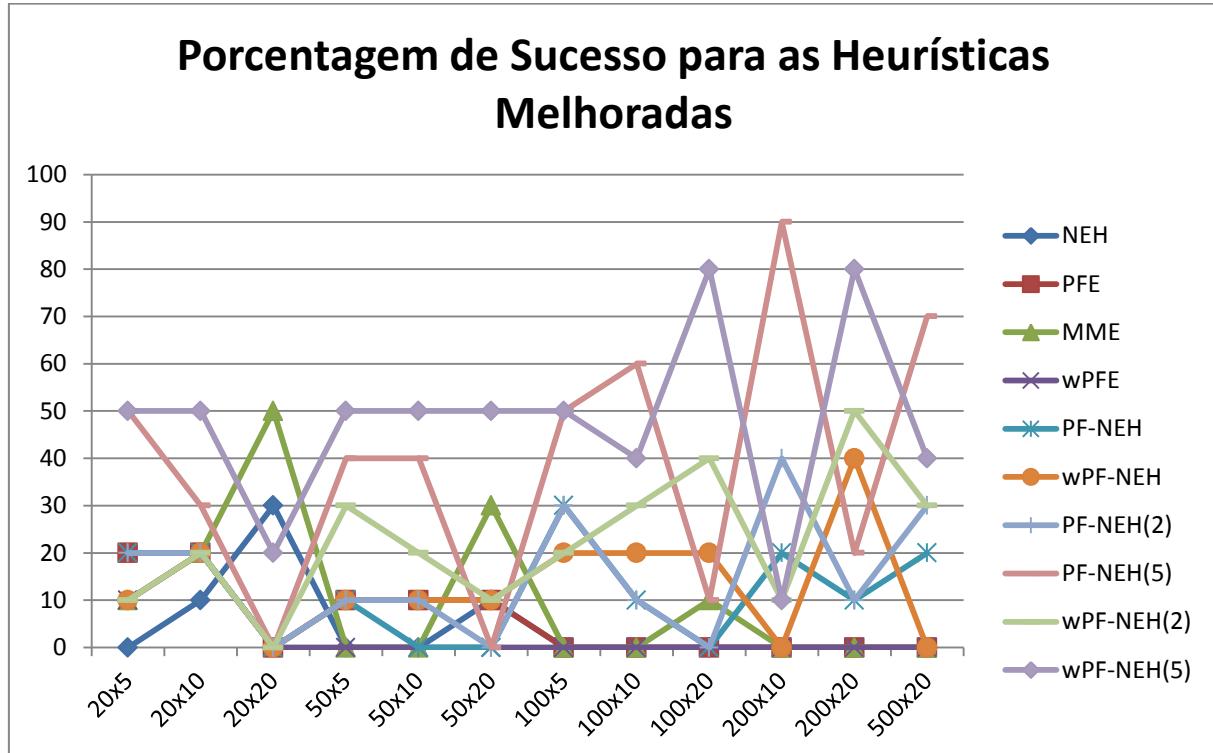


Figura 7: Porcentagem de Sucesso para as Heurísticas Melhoradas

A Figura 8 é uma ilustração dos dados para desvio relativo médio da Tabela 8.

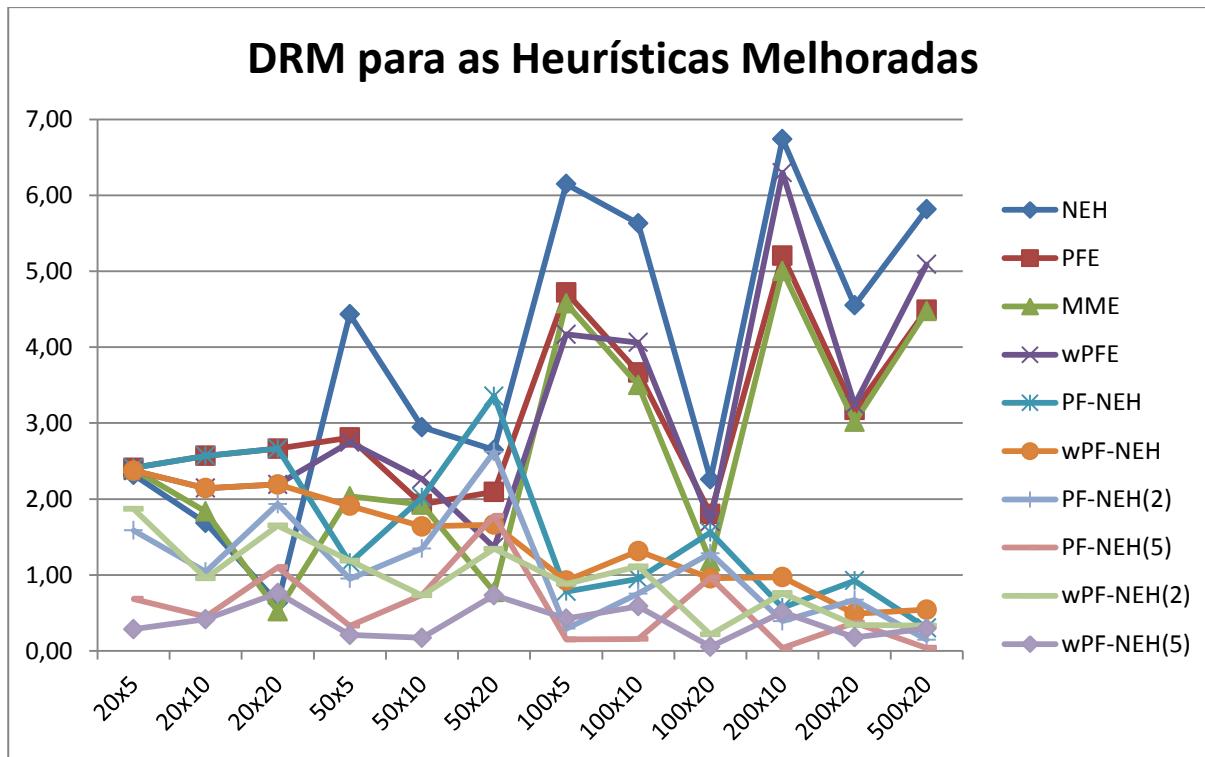


Figura 8: Desvio Relativo Médio para as Heurísticas Melhoradas

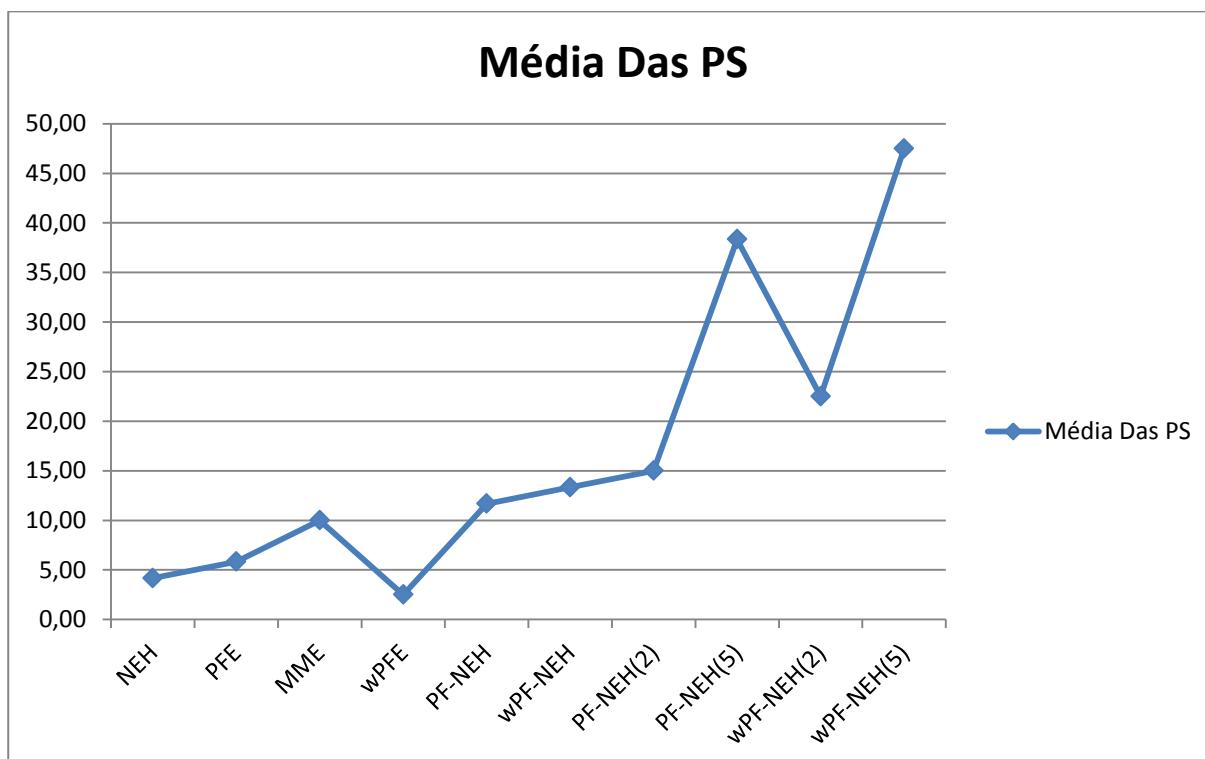


Figura 9: Média da PS para Heurísticas Melhoradas

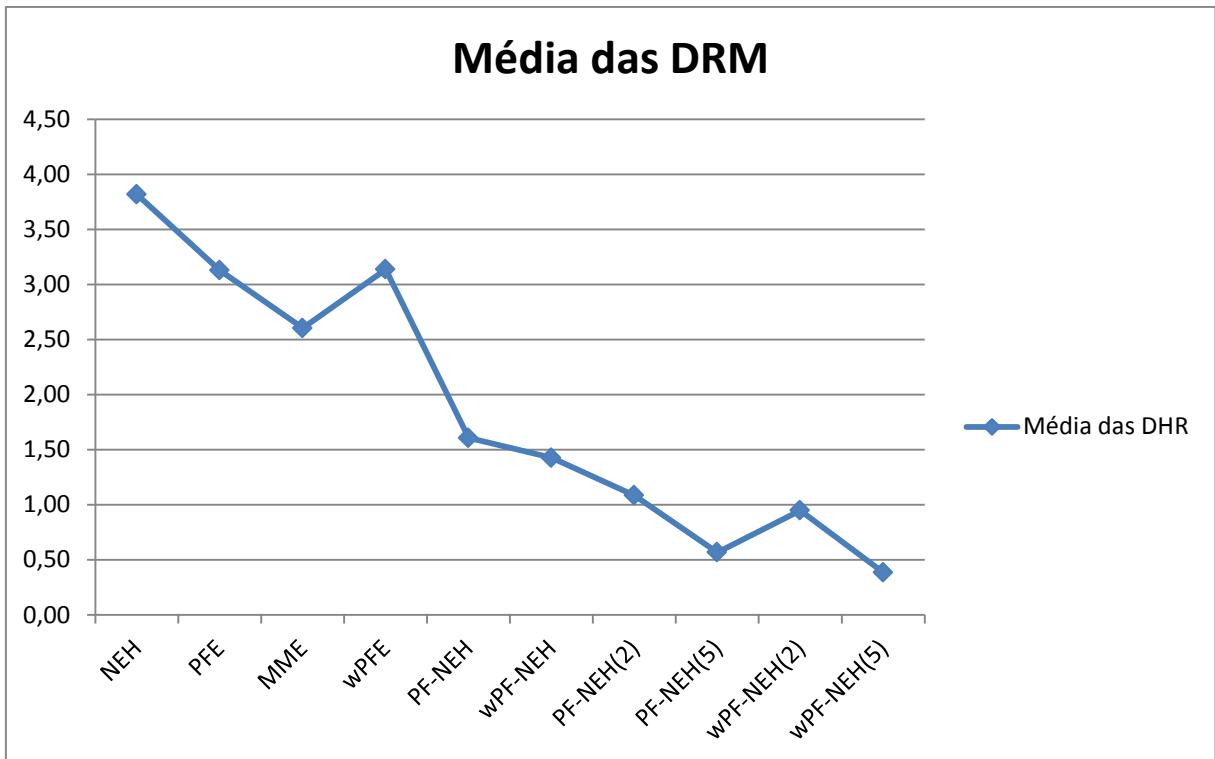


Figura 10: Média da DRM para Heurísticas Melhoradas

A Figura 11 é uma ilustração dos dados de tempos de processamento da Tabela 9.

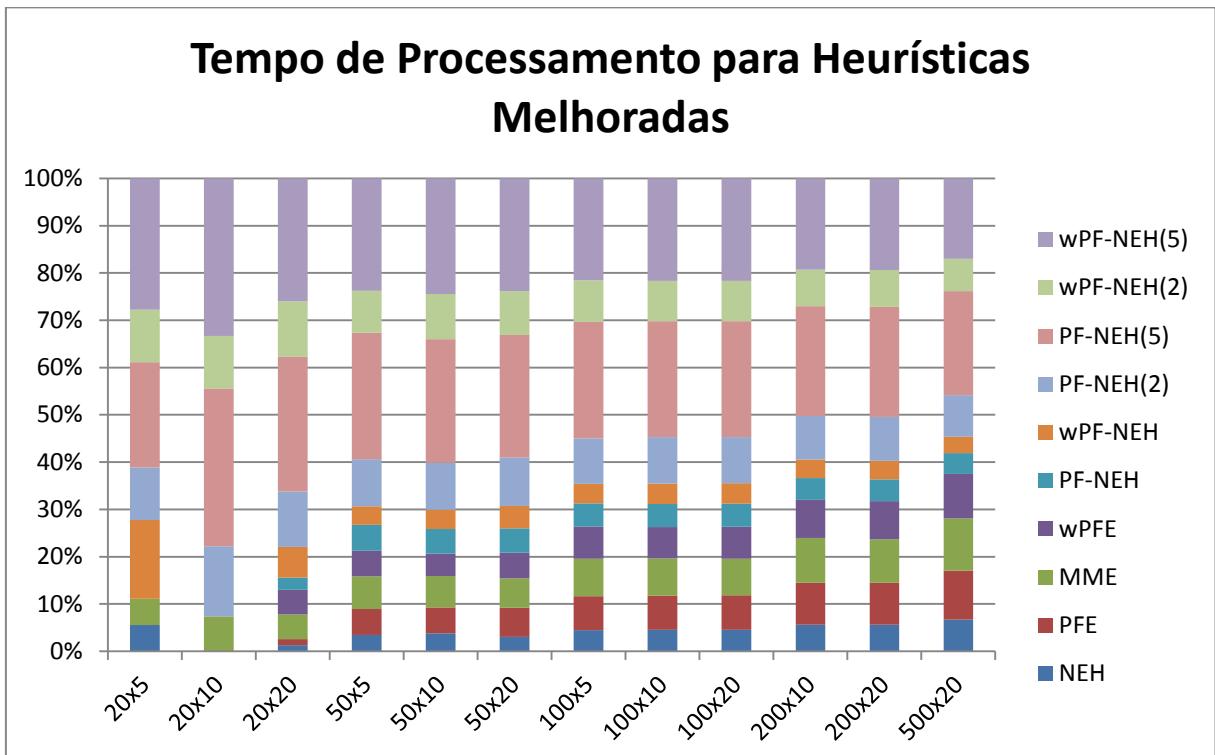


Figura 11: Tempo de Processamento Relativo para as Heurísticas Melhoradas

De acordo com os dados da Tabela 7 e Tabela 8 e pelas Figura 7 e Figura 8, o método que obteve o melhor índice de Porcentagem de Sucesso e de Desvio Relativo Médio na maioria dos portes de problema, foi o método wPF-NEH(5).

O método que se aproximou da heurística wPF-NEH(5), foi o método PF-NEH(5) que obteve resultados satisfatórios, porém, não superiores. Os métodos propostos por Ronconi (MME e PFE) e Nawaz (NEH) foram superados por estes métodos quando confrontados diretamente. Pelas figuras Figura 9 e Figura 10, são ilustrados os valores médios da Porcentagem de Sucesso e Desvio Relativo Médio, a superioridade dos métodos sugeridos por Pan QK torna-se mais evidente sobre os métodos de Ronconi e Nawaz.

Com relação aos tempos computacionais gastos, os métodos PF-NEH(5) e wPF-NEH(5) apresentaram tempos semelhantes e dominantes no tempo total utilizado. É evidente o aumento de tempo gasto, porém, com os desempenhos adquiridos nas estatísticas avaliadas, tornam-se insignificantes quando equiparados.

Heurísticas Compostas

Com a adição da etapa de uma nova etapa de melhoramento, a composição, foram programadas as heurísticas $PF - EH_{LS}$, $wPF - NEH_{LS}$, $PF - NEH(2)_{LS}$, $PF - NEH(5)_{LS}$, $wPF - NEH(2)_{LS}$, $wPF - NEH(5)_{LS}$.

As tabelas de Tabela 10 à Tabela 12 apresentam os resultados das heurísticas compostas.

Tabela 10: Porcentagem de Sucesso para as Heurísticas Compostas

Tarefas (n)	Máquinas (m)	PF-NEH-LS	wPF-NEH- LS	PF-NEH- LS(2)	PF-NEH- LS(5)	wPF-NEH- LS(2)	wPF-NEH- LS(5)
20	5	10	20	20	40	20	80
20	10	0	20	10	40	60	70
20	20	0	40	10	30	50	90
50	5	20	10	20	40	30	60
50	10	0	0	20	50	0	50
50	20	20	10	50	50	10	50
100	5	20	10	30	60	10	40
100	10	20	20	40	70	20	40
100	20	0	10	0	40	20	60
200	10	20	0	30	80	10	20
200	20	10	10	20	50	10	50
500	20	10	0	30	70	30	30
Média		10,83	12,50	23,33	51,67	22,50	53,33

Tabela 11: Desvio Médio Relativo para as Heurísticas Compostas

Tarefas (n)	Máquinas (m)	PF-NEH-LS	wPF-NEH- LS	PF-NEH- LS(2)	PF-NEH- LS(5)	wPF-NEH- LS(2)	wPF-NEH- LS(5)
20	5	1,97	1,55	1,42	0,42	1,23	0,22
20	10	2,17	1,35	1,29	0,49	0,59	0,33
20	20	1,74	0,68	1,20	0,38	0,39	0,14
50	5	0,97	1,55	0,61	0,18	1,06	0,30
50	10	1,36	1,39	0,73	0,18	0,97	0,44
50	20	1,06	1,00	0,50	0,35	0,85	0,23
100	5	0,68	0,85	0,41	0,20	0,81	0,46
100	10	0,94	1,27	0,59	0,15	0,98	0,50
100	20	0,94	0,55	0,85	0,36	0,42	0,16
200	10	0,46	0,87	0,31	0,03	0,65	0,50
200	20	0,52	0,41	0,25	0,12	0,37	0,24
500	20	0,21	0,64	0,08	0,03	0,40	0,37
Média		1,09	1,01	0,69	0,24	0,73	0,32

Tabela 12: Tempo Computacional para as Heurísticas Melhoradas

Tarefas (n)	Máquinas (m)	PF-NEH-LS	wPF-NEH- LS	PF-NEH- LS(2)	PF-NEH- LS(5)	wPF-NEH- LS(2)	wPF-NEH- LS(5)
20	5	30	20	50	140	40	130
20	10	60	60	110	290	100	280
20	20	100	90	200	560	210	540
50	5	480	500	960	2550	990	2490
50	10	830	810	1630	4240	1670	4120
50	20	1540	1470	3000	7580	3010	7490
100	5	5960	5880	11860	30110	11920	30270
100	10	8230	8270	16850	41860	16260	41420
100	20	12850	12660	25880	65780	25220	63600
200	10	96430	96220	195960	468590	197070	493020
200	20	131410	129290	264520	657310	260470	656520
500	20	1085390	1033330	2163720	5456840	2062330	5161790
Soma		1343310	1288600	2684740	6735850	2579290	6461670

A Figura 12 ilustra os valores presentes na Tabela 10

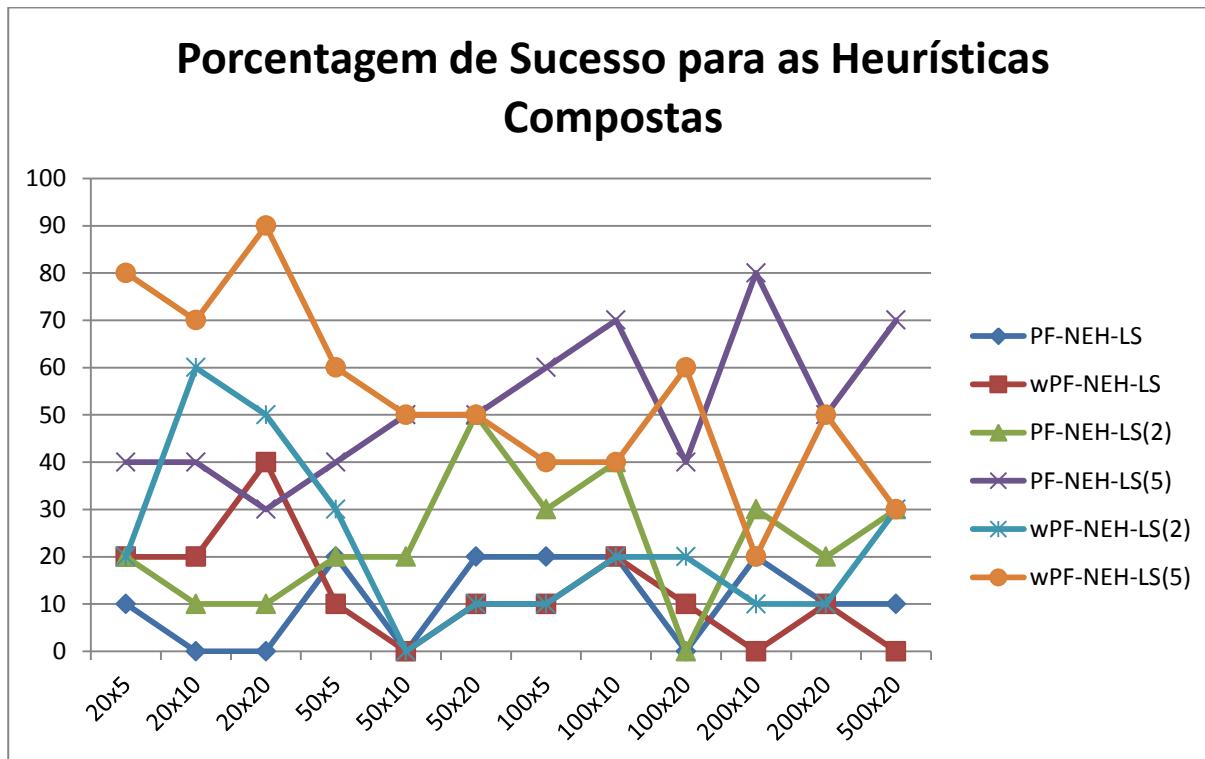


Figura 12: Porcentagem de Sucesso para as Heurísticas Compostas

A Figura 13 ilustra os valores presentes na Tabela 11

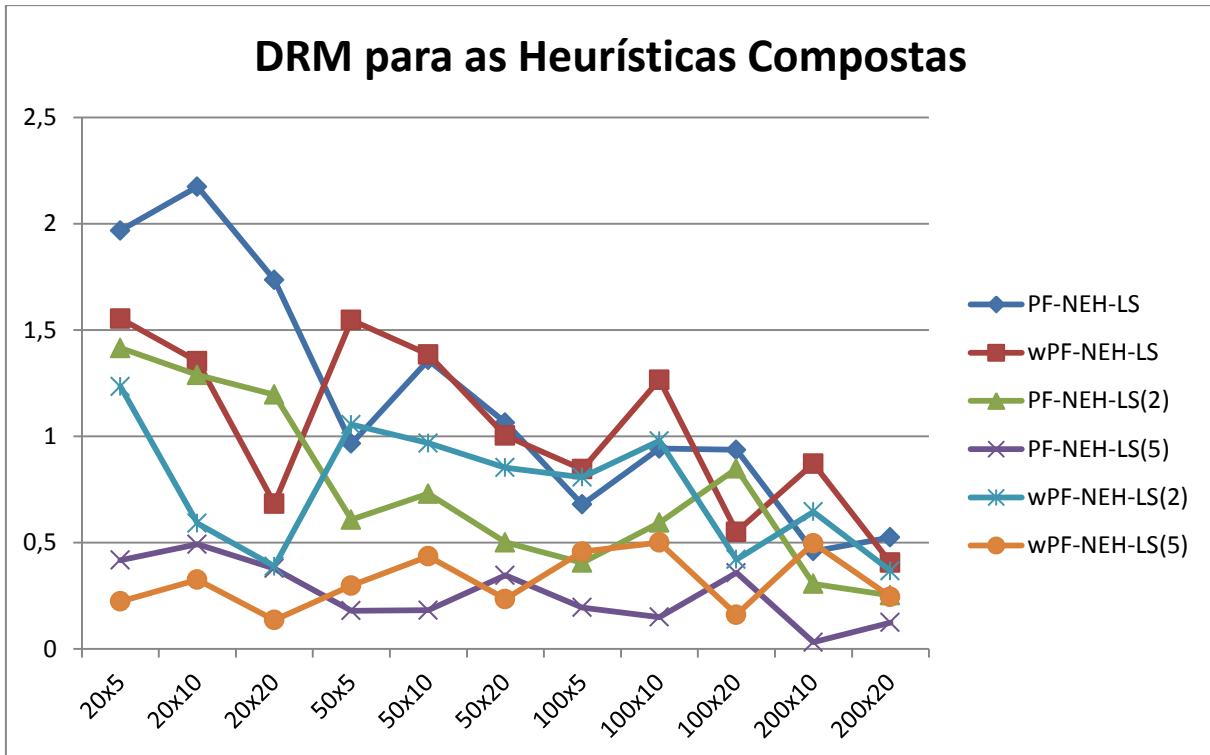


Figura 13: Desvio Relativo Médio para as Heurísticas Compostas

A Figura 14 ilustra os valores da Tabela 12.

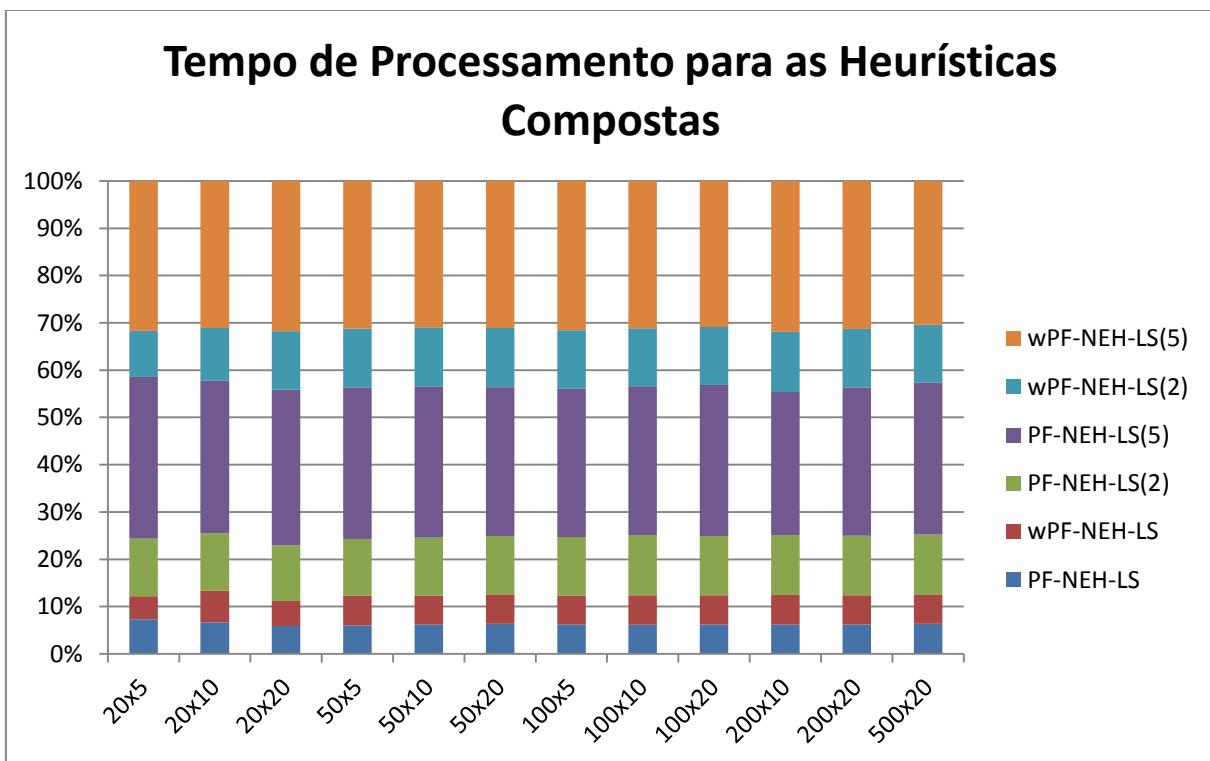


Figura 14: Tempo Computacional Relativo para as Heurísticas Compostas

As figuras Figura 15 e Figura 16 ilustram as médias dos valores presentes nas tabelas Tabela 10 e Tabela 11.

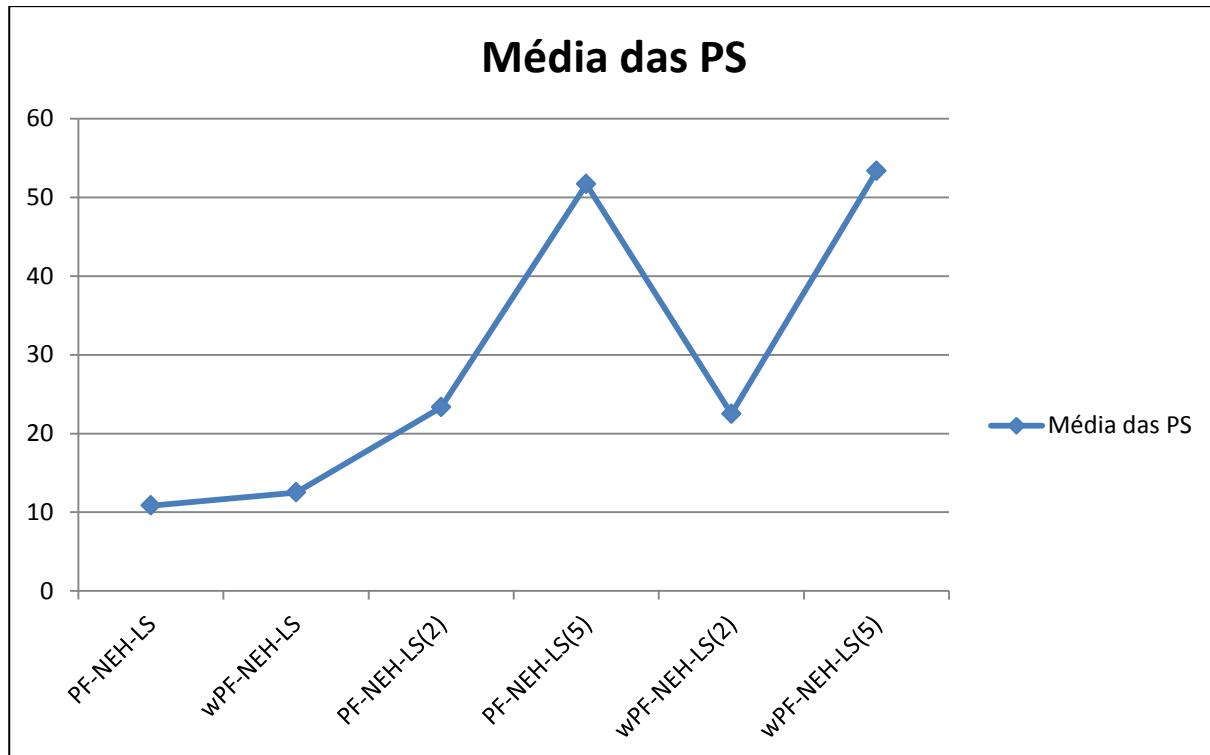


Figura 15: PS Média para as Heurísticas Compostas

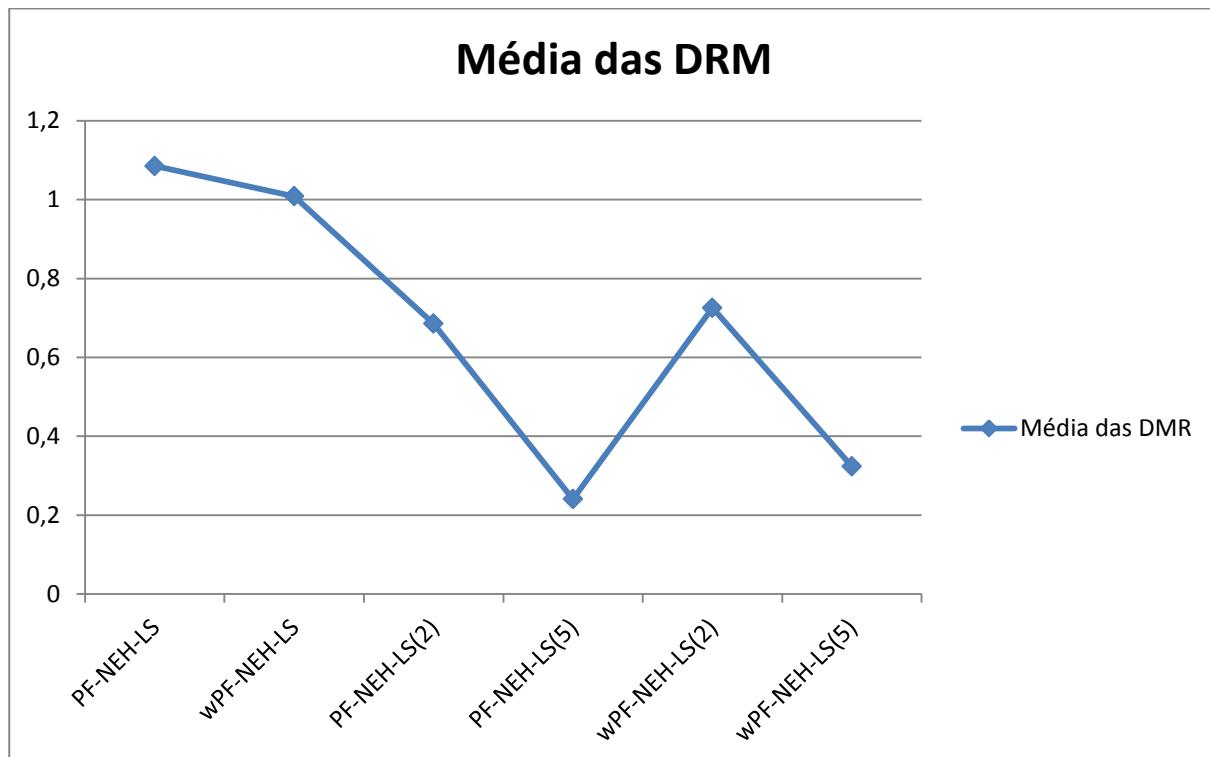


Figura 16: DRM Média para as Heurísticas Compostas

A Tabela 10 e a Figura 12 representam os dados de Porcentagem de Sucesso para as Heurísticas Compostas. É evidente que as heurísticas PF – NEH_{LS}(5) e wPF – NEH_{LS}(5) foram superiores as outras apresentadas. O método wPF – NEH_{LS}(5) mostrou –se superior em problemas de pequeno e médio porte (problemas de até 50 tarefas), porém, para problemas de grande porte (a partir de 100 tarefas), o método PF – NEH_{LS}(5) apresentou uma eficiência superior. A Figura 15 ilustra a média dos valores da Porcentagem de Sucesso e o método wPF – NEH_{LS}(5) é superior a todas as outras heurísticas desta classe quando comparadas. A Tabela 11 e a Figura 13 apresentam os valores de Desvio Relativo Médio e de acordo com a Figura 16, que ilustra a média, a heurística PF – NEH_{LS}(5) apresentou superioridade sobre todas. Com esta análise ressalta a superioridade dos métodos PF-NEH(5) e wPF-NEH(5) e suas composições.

O custo computacional relativo, apresentado na Figura 14, foi maior para os métodos que apresentaram os melhores resultados e, pela Tabela 12, os tempos totais gastos por elas foram bem superiores aos outros métodos.

Todas as heurísticas

Nesta seção, serão comparados todos os métodos diretamente sem distinção de categoria para as mesmas estatísticas utilizada anteriormente, Porcentagem de Sucesso e Desvio Relativo Médio. As tabelas Tabela 13 e Tabela 14 apresentam estes dados.

A Figura 17 e a Figura 18 são ilustrações dos dados apresentados nas tabelas Tabela 13 e Tabela 14, respectivamente. A Figura 19 e a Figura 20 ilustram os tempos computacionais relativos e absolutos para todas as heurísticas dos valores presentes na Tabela 15. Para uma análise de desempenho geral dos problemas, as figuras Figura 21 e Figura 22 ilustram as médias dos valores apresentados nas tabelas Tabela 13 e Tabela 14.

Tabela 13: Porcentagem de Sucesso para todas as Heurísticas

N x M	MM	PF	wPF	NEH	PFE	MME	wPFE	PF-NEH	wPF-NEH	PF-NEH(2)	PF-NEH(5)	wPF-NEH(2)	wPF-NEH(5)	PF-NEH-LS	wPF-NEH-LS	PF-NEH-LS(2)	PF-NEH-LS(5)	wPF-NEH-LS(2)	wPF-NEH-LS(5)
20x5	0	0	0	0	0	0	10	0	10	0	0	10	20	10	20	20	40	20	80
20x10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	10	40	60	70
20x20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	10	30	50	90
50x5	0	0	0	0	0	0	0	0	0	0	0	0	0	20	10	20	40	30	60
50x10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	50	0	50
50x20	0	0	0	0	0	0	0	0	0	0	0	0	0	20	10	50	50	10	50
100x5	0	0	0	0	0	0	0	0	0	0	0	0	0	20	10	30	60	10	40
100x10	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	40	70	20	40
100x20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	40	20	60
200x10	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	30	80	10	20
200x20	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	20	50	10	50
500x20	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	30	70	30	30
Média	0,00	0,00	0,00	0,00	0,00	0,00	0,83	0,00	0,83	0,00	0,00	0,83	1,67	10,83	12,50	23,33	51,67	22,50	53,33

Tabela 14: Desvio Relativo Médio para todas as Heurísticas

N x M	MM	PF	wPF	NEH	PFE	MME	wPFE	PF-NEH	wPF-NEH	PF-NEH(2)	PF-NEH(5)	wPF-NEH(2)	wPF-NEH(5)	PF-NEH-LS	wPF-NEH-LS	PF-NEH-LS(2)	PF-NEH-LS(5)	wPF-NEH-LS(2)	wPF-NEH-LS(5)
20x5	11,80	8,61	8,65	3,40	3,49	3,48	3,46	3,49	3,46	2,67	1,75	2,95	1,35	1,97	1,55	1,42	0,42	1,23	0,22
20x10	14,60	13,81	11,25	3,57	4,47	3,73	4,05	4,47	4,05	2,92	2,32	2,83	2,28	2,17	1,35	1,29	0,49	0,59	0,33
20x20	13,42	13,83	10,69	2,00	4,06	1,89	3,58	4,06	3,58	3,32	2,48	3,03	2,13	1,74	0,68	1,20	0,38	0,39	0,14
50x5	9,51	5,17	5,20	5,55	3,91	3,13	3,85	2,24	3,00	2,03	1,40	2,27	1,28	0,97	1,55	0,61	0,18	1,06	0,30
50x10	12,64	7,41	6,27	4,58	3,54	3,53	3,88	3,64	3,25	2,95	2,33	2,33	1,76	1,36	1,39	0,73	0,18	0,97	0,44
50x20	13,37	11,51	8,59	3,76	3,20	1,87	2,46	4,48	2,77	3,74	2,89	2,45	1,83	1,06	1,00	0,50	0,35	0,85	0,23
100x5	7,68	3,46	3,34	6,98	5,54	5,39	4,98	1,57	1,71	1,07	0,93	1,67	1,21	0,68	0,85	0,41	0,20	0,81	0,46
100x10	10,46	4,19	4,52	6,56	4,58	4,42	4,98	1,85	2,21	1,65	1,05	2,01	1,48	0,94	1,27	0,59	0,15	0,98	0,50
100x20	11,41	6,19	4,87	3,38	2,92	2,29	2,82	2,67	2,06	2,40	2,08	1,32	1,15	0,94	0,55	0,85	0,36	0,42	0,16
200x10	10,32	2,66	2,97	7,31	5,77	5,57	6,87	1,11	1,51	0,93	0,58	1,30	1,05	0,46	0,87	0,31	0,03	0,65	0,50
200x20	10,77	3,66	2,93	5,23	3,84	3,68	3,91	1,58	1,14	1,32	1,02	0,99	0,83	0,52	0,41	0,25	0,12	0,37	0,24
500x20	10,28	1,76	2,08	6,40	5,07	5,05	5,67	0,86	1,10	0,70	0,59	0,89	0,85	0,21	0,64	0,08	0,03	0,40	0,37
Média	11,36	6,85	5,95	4,89	4,20	3,67	4,21	2,67	2,49	2,14	1,62	2,00	1,43	1,09	1,01	0,69	0,24	0,73	0,32

Tabela 15: Tempo de Processamento para todas as Heurísticas

N x M	MM	PF	wPF	NEH	PFE	MME	wPFE	PF-NEH	wPF-NEH	PF-NEH(2)	PF-NEH(5)	wPF-NEH(2)	wPF-NEH(5)	PF-NEH-LS	wPF-NEH-LS	PF-NEH-LS(2)	PF-NEH-LS(5)	wPF-NEH-LS(2)	wPF-NEH-LS(5)
20x5	0	0	0	10	0	10	0	0	30	20	40	20	50	30	20	50	140	40	130
20x10	0	0	0	0	0	20	0	0	0	40	90	30	90	60	60	110	290	100	280
20x20	0	0	10	10	10	40	40	20	50	90	220	90	200	100	90	200	560	210	540
50x5	0	50	20	70	110	140	110	110	80	200	540	180	480	480	500	960	2550	990	2490
50x10	0	100	70	160	230	280	200	220	170	420	1100	400	1030	830	810	1630	4240	1670	4120
50x20	0	200	140	270	540	540	480	450	410	900	2270	810	2090	1540	1470	3000	7580	3010	7490
100x5	0	370	210	570	910	1010	870	620	530	1220	3140	1120	2740	5960	5880	11860	30110	11920	30270
100x10	30	690	510	1150	1830	1980	1680	1220	1100	2470	6200	2150	5470	8230	8270	16850	41860	16260	41420
100x20	40	1380	1050	2330	3740	3940	3460	2500	2210	4970	12530	4370	11070	12850	12660	25880	65780	25220	63600
200x10	160	5010	3780	9100	14150	15240	12960	7480	6250	14820	37280	12390	31080	96430	96220	195960	468590	197070	493020
200x20	230	9990	7600	18040	28300	29520	25720	14900	12430	29820	74480	24820	62090	131410	129290	264520	657310	260470	656520
500x20	2580	149670	109160	275880	428530	453100	389600	181880	141110	362660	908220	280780	702480	1085390	1033330	2163720	5456840	2062330	5161790
Soma	3040	167460	122550	307590	478350	505820	435120	209400	164370	417630	1046110	327160	818870	1343310	1288600	2684740	6735850	2579290	6461670

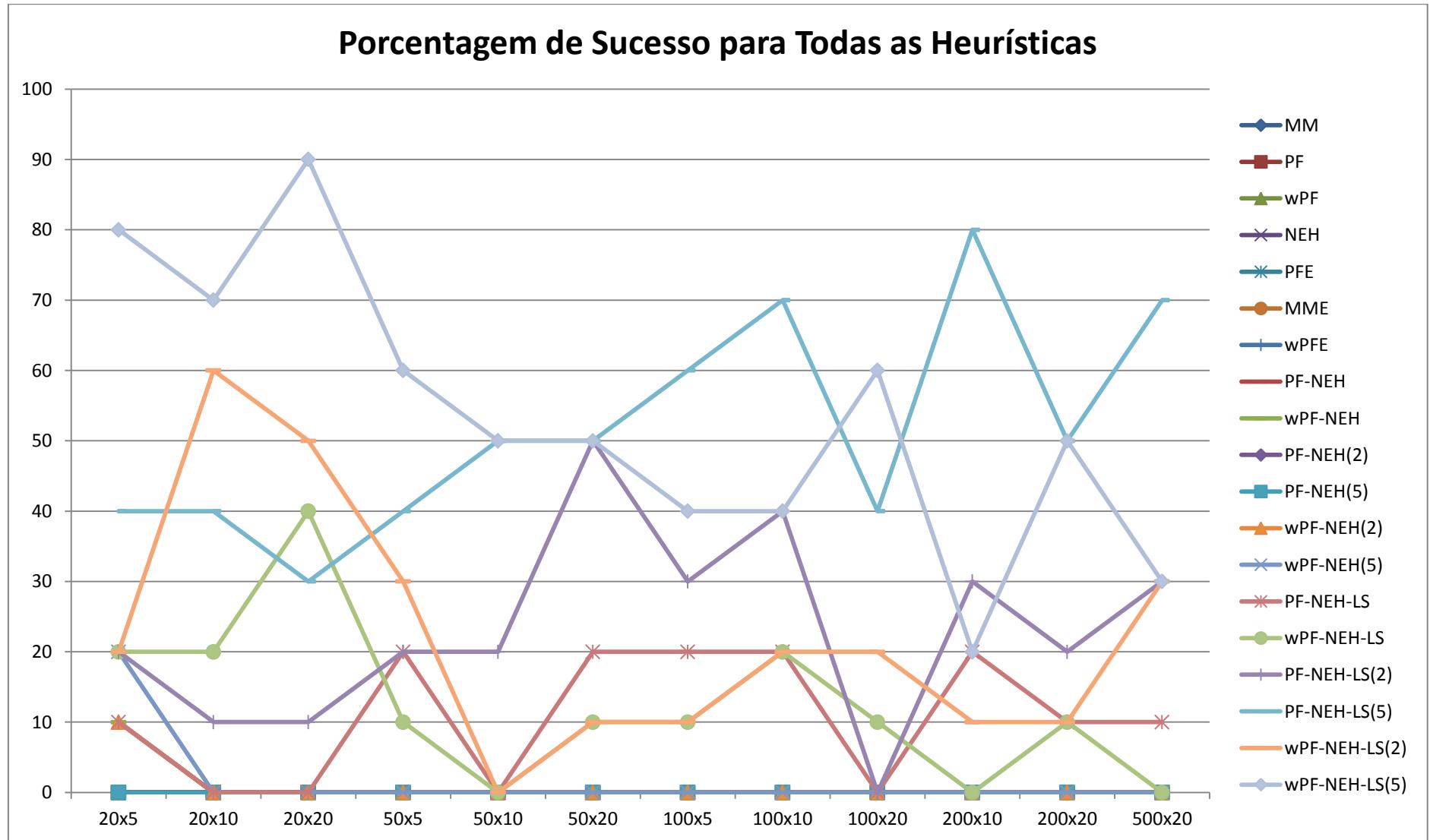


Figura 17: Porcentagem de Sucesso para todas as Heurísticas

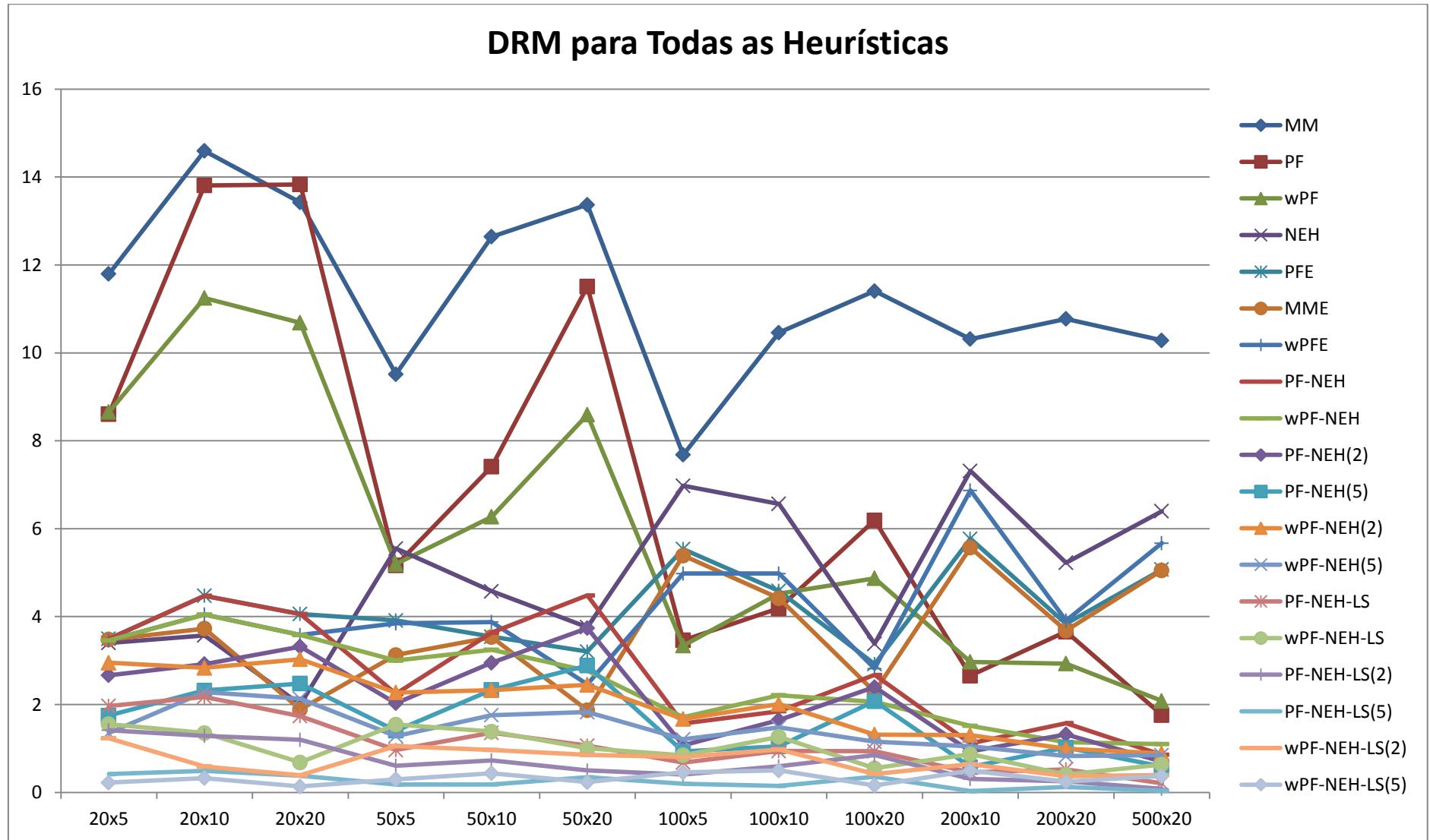


Figura 18: Desvio Relativo Médio para todas as Heurísticas

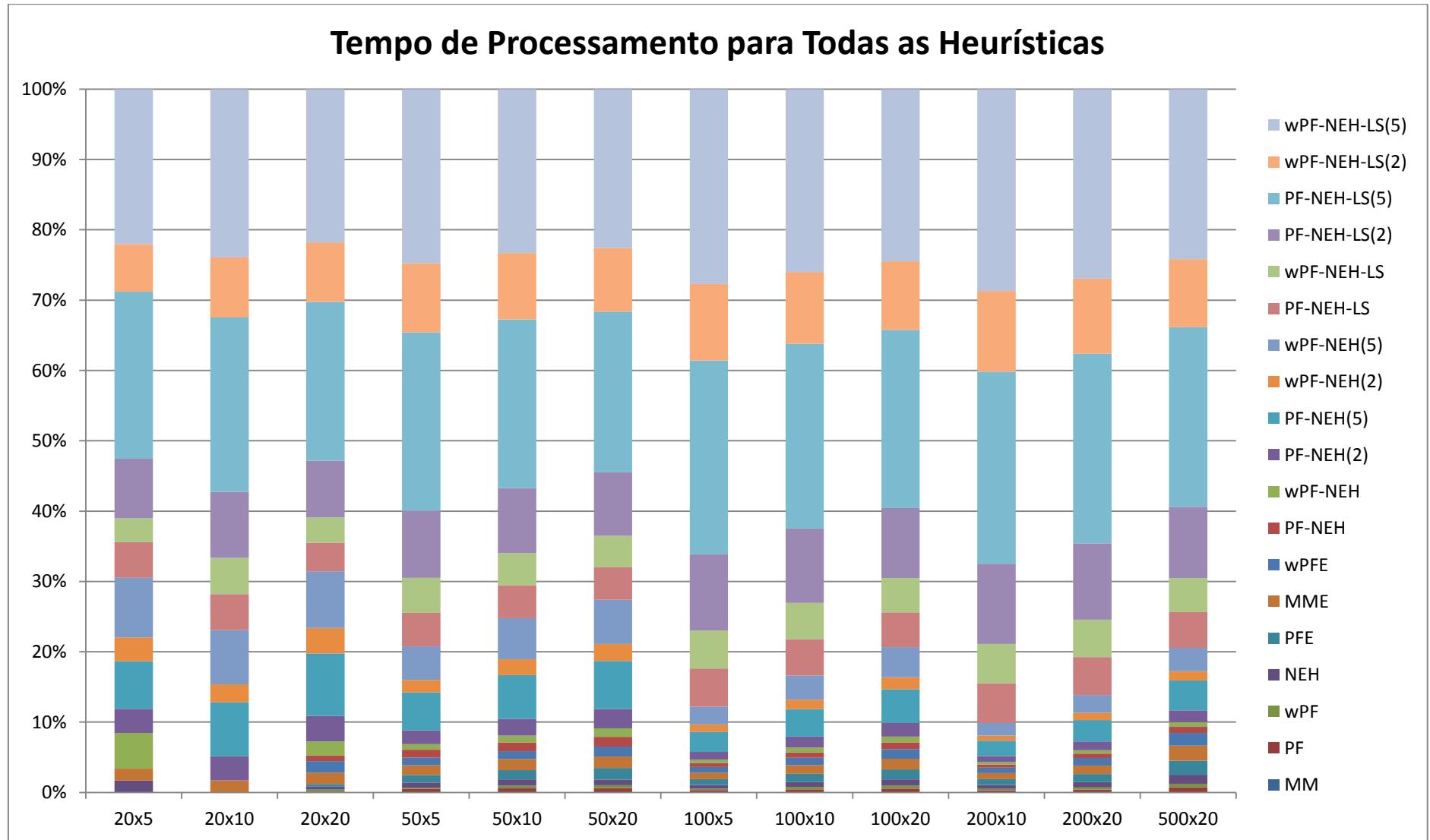


Figura 19: Tempo Computacional Relativo para todas as heurísticas

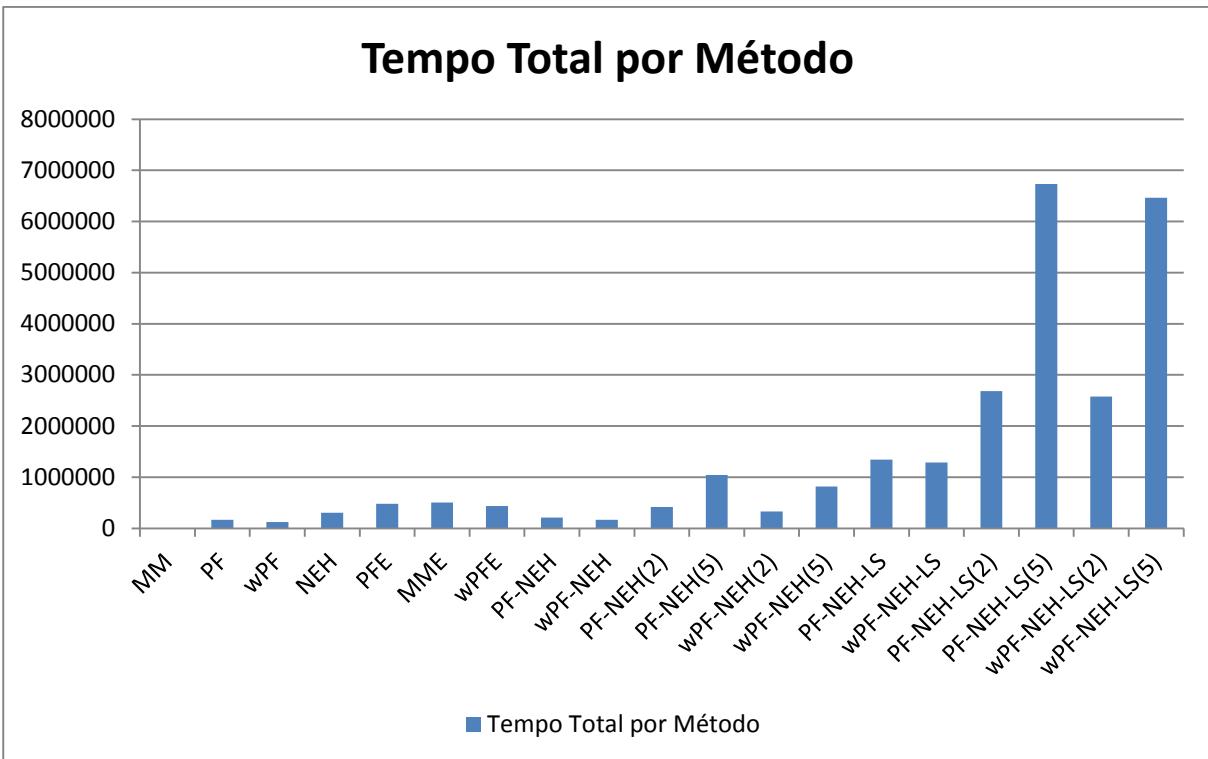


Figura 20: Tempo Computacional Absoluto para todas as Heurísticas

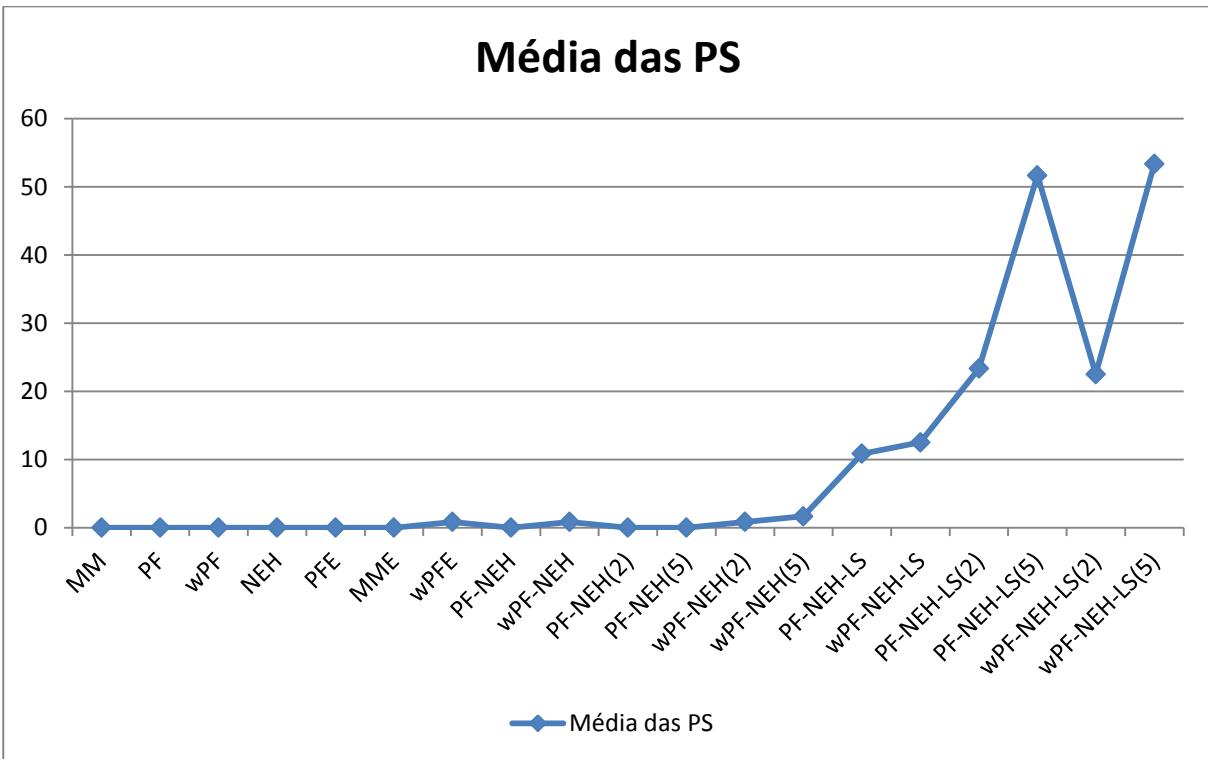


Figura 21: Média da PS para todas as Heurísticas

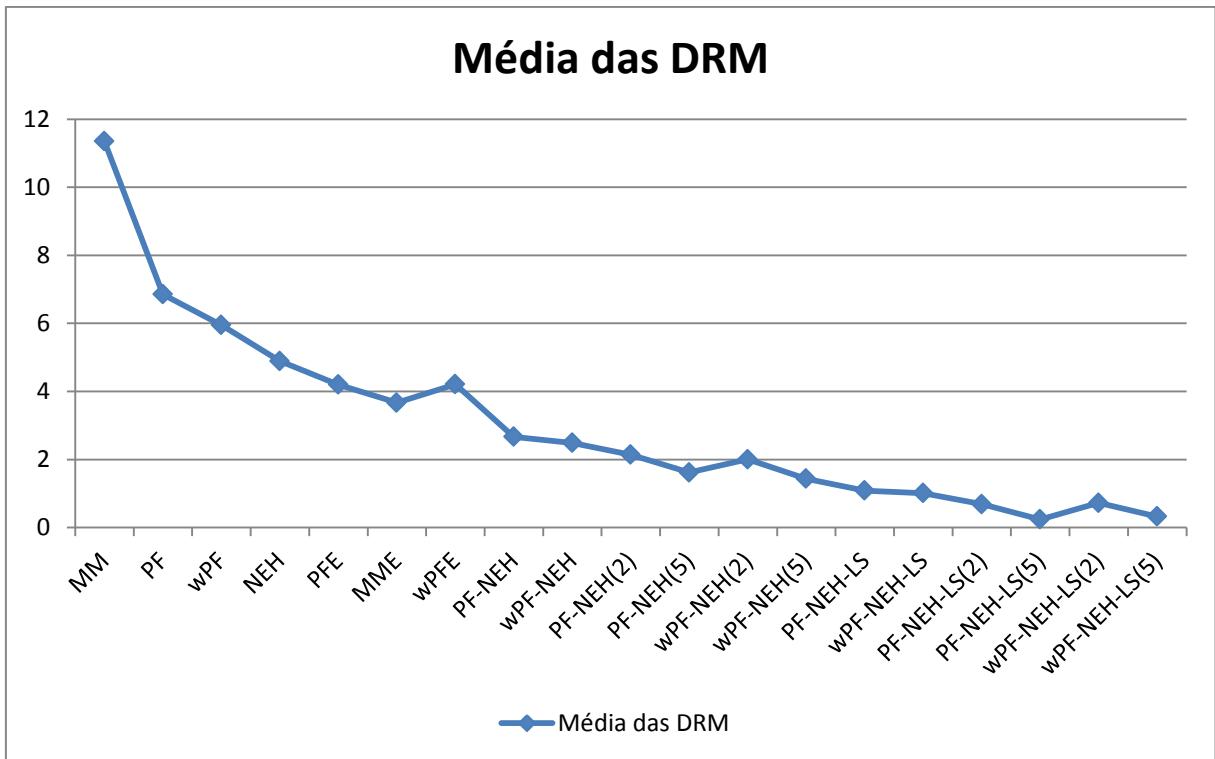


Figura 22: Média da DRM para todas as Heurísticas

A Porcentagem de Sucesso apresentada na Tabela 13 e na Figura 17, ressaltam o desempenho superior das heurísticas compostas sobre as heurísticas simples e melhoradas e, a Figura 21, ilustra a média que confirma esta superioridade no geral. Entre as heurísticas compostas, os métodos PF – NEH_{LS}(5) e wPF – NEH_{LS}(5) destacaram-se ao obterem em torno de 50% (cada método) de Porcentagem de Sucesso na média. Para as outras classes, as heurísticas simples apresentaram nenhum sucesso e as heurísticas melhoradas uma porcentagem quase nula.

Para o Desvio Relativo Médio, apresentado na Tabela 14 e na Figura 18, as heurísticas compostas apresentaram superioridade sobre as outras classes, porém, com uma diferença menor que a estatística anteriormente citada. De acordo com a Figura 22, o método PF – NEH_{LS}(5) apresentou o menor valor de Desvio Relativo Médio na média entre todas as heurísticas.

Os tempos computacionais presentes na Tabela 15 e ilustrados pelas figuras Figura 19 e Figura 20, os métodos compostos tiveram tempos superiores às outras classes para a obtenção da solução dos problemas. Os métodos que apresentaram os melhores resultados, o tempo computacional total destes chegou a ser oito vezes maior aos tempos de processamento das heurísticas melhoradas.

6. Considerações Finais

A análise das Heurísticas Simples permitiu concluir que para a maioria dos portes de problema, o método wPF possuiu os melhores resultados nas estatísticas de Porcentagem de Sucesso e Desvio Relativo Médio para as heurísticas simples e o esforço computacional gasto é o mesmo que o método PF o que ressalta mais a qualidade do método wPF. Já para as heurísticas melhoradas, podemos destacar dois métodos, o PF-NEH (5) e o wPF-NEH(5). Na análise realizada, os resultados de ambos em conjunto, apresentaram superioridade em todos os portes de problema para as estatísticas avaliadas entre as heurísticas melhoradas, mas na média geral de todos os portes de problema, o método wPF-NEH(5) apresentou um desempenho superior ao método PF-NEH(5) em ambos os quesitos avaliados. Para as heurísticas compostas, as heurísticas wPF – NEH(5)_{LS} e PF – NEH(5)_{LS} apresentaram os melhores resultados em conjunto para Porcentagem de Sucesso e Desvio Relativo Médio em todos os portes de problema entre as heurísticas compostas. Os resultados de ambas as heurísticas foram bem semelhantes na média das estatísticas, com uma pequena superioridade do método wPF – NEH(5)_{LS} na estatística de Porcentagem de Sucesso e do método PF – NEH(5)_{LS} na estatística de Desvio Relativo Médio.

No comparativo de todas as heurísticas, os métodos wPF – NEH(5)_{LS} e PF – NEH(5)_{LS} ainda se mantêm como os melhores para as estatísticas analisadas, mas em linhas gerais, o esforço computacional para a obtenção da soluções é elevado o que reflete na qualidade das soluções. Com esta analise do tempo computacional de cada método heurístico, é evidente que a complexidade do método está relacionada ao tempo computacional gasto.

A análise comparativa realizada neste trabalho é de grande importância literária para a aplicação em trabalhos futuros, podendo ser utilizada em comparações com outros métodos a serem desenvolvidos e também na extensão para outros ambientes de controle de produção, como, por exemplo, no ambiente *flowshop permutacional* e *no-idle flowshop*.

Referências Bibliográficas

- Abadi INK, Hall NG, Sriskandarajh C.** Minimizing cycle time in a blocking flowshop // Operations Research. 2000; 48 – pp. 77-80.
- Baker K. R.** Indroduction to sequencing and scheduling. – New York: Wiley, 1974.
- Framinan Jose M., Leisten Rainer e Usano Rafael Ruiz** Comprasion of heuristics for flowtime minimisation in permutation flowshops // Computers & Operatios Research. – 2005. – pp. 1237-1254.
- Garey M. R., Johnson D. S. e Sethi R.** The complexity of flowshop and jobshop Scheduling // Mathematics of Operations Research. 1976 – Vol. 1 – pp. 117-129.
- Gigante Rodrigo Luiz** Heurística construtiva para a programação de operações flow shop permutacional. São Carlos. 2010.
- Grabowski J., Pempera J.** Sequencing of jobs in some production system // European Journal of Operational Research. 2000;125 – pp. 535-550.
- Grabowski J., Skubalska E., Smutnick C.** On flow shop scheduling with release and due dates to minimize maximum lateness // Journal of the Operational Research Society. 1983; 34 – pp. 615-620.
- Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG.** Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 1979;5 – pp. 287–362
- Gupta J. N. D.** Heuristic Algorithms for multistage flowshop scheduling problem // AIIE Transactions. 1972 – Vol. 4 – pp; 11-18.
- Januário J.S.S.** Redução do estoque em processamento em sistemas de produção flowshop permutacional: uma avaliação dos métodos heurísticos. São Carlos - Trabalho de Conclusão de Curso. 2011.
- Johnson S. M.** Optimal two- and three-stage production schedules with setup times included. // Naval Research Logistics Quarterly. 1954 – Vol. 1 – pp. 61-68.
- Leisten R.** Flowshop sequencing problems with limited buffer storage. // International Journal of Production Research, 1990;28 – pp. 2085-2100.
- Levner E.M.** Optimal planning of parts machining on a number of machines // Automation and Remote Control. 1969; 12 – pp. 1972-1978.
- MacCarthy, B. Liu, j.** Addressing the gap in scheduling research: a reviem of optimization and heuristic methods in production scheduling. International Journal of Production Research, London. 1993 – p.59-79.

- McCormick ST, Pinedo ML, Shenker S, Wolf B.** Sequencing in an assembly line with blocking to minimize cycle time // Operations Research 1989; 28 – pp. 25-36.
- Nagano M. e Moccelin J.V.** Reducing mean flow time in permutation flow shop // Journal of the Operational Research Society. 2008 – pp. 939-945.
- Nawaz M, Enscore EEJ, Ham I.** A heuristic algorithm for the m -machine, n -job flow shop sequencing problem // Omega – International Journal of management Science 1983;11 – pp. 91-95.
- Nowicki E.** the permutation flow shop with buffers: A tabu search approach // European Journal of Operational Research. 1999; 116 – pp. 205-219.
- Pan QK e Wang L.** Effective heuristics for the blocking flowshop scheduling problem with makespan minimization // Omega. 2011; 40 – pp. 218-229.
- Quan-Ke Pan, M. Fatih Tasgetiren, Yun-Chia Liang** A discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem – London: GECCO, 2007.
- Ribas I, Companys R, Tort-Martorell X.** An iterated greedy algorithm for the flowshop scheduling with blocking // Omega – The International Journal of Management Science, 2011; 39 – pp. 293-301.
- Ribas I. Companys R, Tort-Martorell X.** Na iterated greedy algorithm for the flowshop scheduling with blocking // Omega – The International Journal of Management Science. 2009;39 – pp. 293-301.
- Ronconi D.P.** A branch-and-bound algorithm to minimize the makespan in a flowshop problem with blocking // Annals of Operations Research. 2005; 138 – pp. 53-65.
- Ronconi D.P. e Armentano V.A.** Lower bounding schemes for flowshops with blocking in-process // Journal of Operational Research Society. 2001; 52 – pp. 1289-1297.
- Ronconi D.P. e Henriques L.R.S.** Uma heurística construtiva para o problema de minimização do atraso total no ambiente flowshop com buffer zero. XXXVI – SBPO. 2004.
- Ronconi DP** A note on constructive heuristics for the flowshop problem with blocking // International Journal of Production Economics 2004; 87 – pp. 39-48.
- Suhami I., Mah R.S.H.** An implicit enumeration scheme for the flowshop problem with no intermediate storage // Computers and Chemical Engineering. 1981;5 – pp. 83-91.
- Suhami S.M.A.** A two-phase heuristic approach to the permutation flow-shop scheduling problem // International Journal of Production Economics. 2000; 64 – pp. 143-152.
- Taillard E.** Benchmarks for basic scheduling problems // European Journal of Operational Research. 1993; 64 – pp. 278-285.
- Wang L, Pan QK, Suganthan PN, Wang WH, Wang YM** A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems // Computers and Operations Research. 2010; 37 – pp. 09-20.