

RODRIGO LUIZ GIGANTE

**HEURÍSTICA CONSTRUTIVA PARA A PROGRAMAÇÃO DE OPERAÇÕES *FLOW*  
*SHOP* PERMUTACIONAL**

Dissertação apresentada à Escola de Engenharia de São Carlos – Universidade de São Paulo como parte dos requisitos para obtenção do título de Mestre em Engenharia de Produção.

Área de concentração: Processos e Gestão de Operações.  
Linha de pesquisa: Pesquisa Operacional Aplicada aos Sistemas de Produção.

Orientador: Prof. Dr. Marcelo Seido Nagano.

São Carlos

2010

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

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

Gigante, Rodrigo Luiz

G459h      Heurística construtiva para a programação de operações  
*flow shop* permutacional / Rodrigo Luiz Gigante; orientador  
Marcelo Seido Nagano. -- São Carlos, 2010.

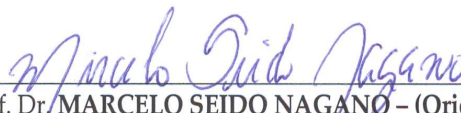
Dissertação (Mestrado-Programa de Pós-Graduação em  
Engenharia de Produção e Área de Concentração em  
Processos e Gestão de Operações) -- Escola de Engenharia  
de São Carlos da Universidade de São Paulo, 2010.

1. Programação da produção. 2. *Flow shop* permutacional. 3. Métodos  
heurísticos construtivos. 4. *Makespan*. I. Título.

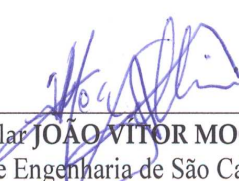
**FOLHA DE JULGAMENTO**

Candidato(a): Bacharel **RODRIGO LUIZ GIGANTE**.

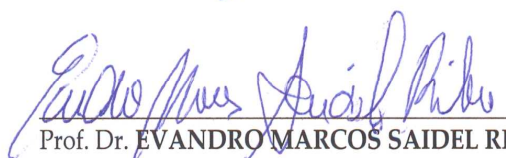
Dissertação defendida e julgada em 21.09.2010 perante a Comissão Julgadora:

  
\_\_\_\_\_  
Prof. Dr. **MARCELO SEIDO NAGANO** – (Orientador)  
(Escola de Engenharia de São Carlos/USP)

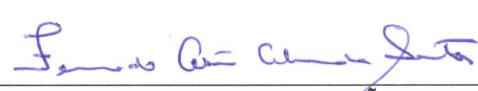
Aprovado

  
\_\_\_\_\_  
Prof. Titular **JOÃO VITOR MOCCELLIN**  
(Escola de Engenharia de São Carlos/USP)

APROVADO

  
\_\_\_\_\_  
Prof. Dr. **EVANDRO MARCOS SAIDEL RIBEIRO**  
(Faculdade de Economia, Administração e Contabilidade de Ribeirão Preto/USP)

Aprovado

  
\_\_\_\_\_  
Prof. Associado **AQUILES ELIE GUIMARÃES KALATZIS**  
Coordenador do Programa de Pós-Graduação em  
Engenharia de Produção

  
\_\_\_\_\_  
Prof. Titular **GERALDO ROBERTO MARTINS DA COSTA**  
Presidente da Comissão da Pós-Graduação da EESC

Dedico este trabalho à minha mãe Syrlei Mendes Gigante.

## **AGRADECIMENTOS**

Agradeço em especial ao Prof. Dr. Marcelo Seido Nagano, pelo empenho e dedicação em me  
indicar o melhor caminho para o desenvolvimento de um bom trabalho.

Agradeço meu pai, irmão e noiva pelo apoio na minha busca pessoal pelo conhecimento e  
crescimento profissional.

Agradeço a todos os que me auxiliaram de alguma forma no desenvolvimento deste trabalho.

“Antes de entrar numa batalha, é preciso  
acreditar no motivo da luta”

Sun Tzu

## RESUMO

---

GIGANTE, R. L. (2010) **Heurística construtiva para a programação de operações *flow shop* permutacional**. 88p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Os processos industriais de produção exigem uma programação da produção efetiva. Essa atividade consiste da alocação dos recursos produtivos, a fim de executar tarefas determinadas por um período de tempo definido. Programar a produção é uma das atividades mais complexas do Planejamento da Produção, pois existem diferentes tipos de recursos a serem administrados simultaneamente. E também a quantidade de possíveis soluções aumenta exponencialmente com o aumento da quantidade de tarefas e máquinas presentes no sistema. A proposta deste trabalho é apresentar um método heurístico construtivo para a solução de problemas *flow shop* permutacional. A função-objetivo utilizada é a minimização do tempo total da programação (*makespan*). O algoritmo foi desenvolvido com base no melhor algoritmo construtivo presente na literatura, e os resultados obtidos são discutidos e analisados com base na porcentagem de sucesso, desvio relativo médio e tempo médio de computação.

Palavras – chave: Programação da produção. *Flow shop* permutacional. Métodos heurísticos construtivos, *Makespan*.



## ABSTRACT

---

GIGANTE, R. L. (2010) *A constructive heuristic for scheduling operations flow shop sequencing problem*. 88p. M.Sc. Dissertation - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

*Industrial productive processes demand an effective Production Scheduling. These activities consist in allocating the productive resources in order to execute determined jobs for a established period of time. Scheduling the production is one of the most complex activities involved in Planning the Production because there are different kinds of resources to be managed simultaneously. Furthermore, the amounts of feasible solutions increase exponentially as the number of jobs and machines in large systems. This dissertation presents a constructive heuristic method to solve the permutational flow shop problem. The evaluation criterion is the total production elapsed time (makespan). The developed algorithm was based on the best algorithm found in the literature, the results are analysed based on the success rate, mean relative deviation and computing time.*

**Keywords:** *Production scheduling. Flow shop scheduling. Constructive heuristics. Makespan.*

## LISTA DE FIGURAS

---

Figura 1.1 – Programação da produção em função dos prazos solicitados .....	16
Figura 1.2 – Programação da produção em função da data de recebimento dos pedidos .....	17
Figura 1.3 – Relação entre as diversas classes de problemas de programação de operações em máquinas.....	21
Figura 3.1 – Resultado da aplicação do FRB3, <i>makespan</i> 39 .....	49
Figura 3.2 – Sequência resultante do FRB3+ENG, <i>makespan</i> 38.....	50

## LISTA DE TABELAS

---

Tabela 3.1 – Dados referentes às tarefas e máquinas para exemplificar a execução do método proposto .....	47
Tabela 4.1 – Porcentagem de sucesso, para as classes de problemas de Taillard.....	55
Tabela 4.2 – Porcentagem de sucesso com máquinas agrupadas para os problemas de Taillard.....	55
Tabela 4.3 – Porcentagem de sucesso para as classes de problemas de Ruben Ruiz .....	55
Tabela 4.4 – Porcentagem de sucesso com máquinas agrupadas para os problemas de Ruben Ruiz.....	57
Tabela 4.5 – Desvio Relativo médio da melhor solução em relação ao número de máquinas e tarefas no problema.....	58
Tabela 4.6 – Desvio Relativo em relação ao número de tarefas do problema.....	58
Tabela 4.7 – Desvio Relativo médio para as classes de problemas de Ruben Ruiz .....	58
Tabela 4.8 – Desvio Relativo dos Métodos sobre o banco de dados de Ruben Ruiz .....	60
Tabela 4.9 – Tempo médio de computação para os problemas de Taillard.....	60
Tabela 4.10 – Tempo médio de computação com máquinas agrupadas para os problemas de Taillard .....	61
Tabela 4.11 – Tempo médio de computação para os problemas de Ruben Ruiz .....	61
Tabela 4.12 – Tempo médio de computação para os problemas de Ruben Ruiz com máquinas agrupadas.....	62
Tabela 4.13– Porcentagem de sucesso para as classes de problema de Taillard com o método FRB3+ENG .....	63

Tabela 4.14 – Porcentagem de sucesso com máquinas agrupadas para as classes de problema de Taillard com o método FRB3+ENG .....	64
Tabela 4.15 – Porcentagem de sucesso para os problemas de Ruben Ruiz com o método FRB3+ENG .....	64
Tabela 4.16 – Porcentagem de sucesso com máquinas agrupadas para os problemas de Ruben Ruiz.....	65
Tabela 4.17 – Desvio relativo médio de todas as classes com FRB3+ENG, Taillard .....	66
Tabela 4.18 – Desvio relativo médio com máquinas agrupadas para os problemas de Taillard, com o método FRB3+ENG.....	67
Tabela 4.19 – Desvio relativo médio para os problemas de Ruben Ruiz com o método FRB3+ENG .....	67
Tabela 4.20 – Desvio relativo médio com máquinas agrupadas para os problemas de Ruben Ruiz com o método FB3+ENG.....	68
Tabela 4.21 – Tempo de execução para os problemas de Taillard com o método FRB3+ENG .....	69
Tabela 4.22 – Tempo de execução para os problemas de Ruben Ruiz com o método FRB3+ENG .....	69
Tabela A.1 – Resultados dos problemas de Taillard .....	77
Tabela A.2 – Resultados dos problemas de Ruben Ruiz.....	80

## LISTA DE SÍMBOLOS

---

$\sum TP_i$	Soma dos tempos de processamento da tarefa $J_i$ sobre todas as máquinas.
$AVG_j$	Média dos tempos de processamento da tarefa $j$ sobre todas as máquinas.
$C_{max}^*$	Valor ótimo do <i>makespan</i> .
$C_{max}$	Duração total de programação <i>makespan</i> .
$d_{ij}^k$	Cálculo de distância entre as tarefas em analogia ao problema do caixeiro viajante.
$f(i)$	Função de indexação da tarefa $J_i$ .
$J$	Conjunto de $n$ tarefas ( $J_1, J_2, \dots, J_n$ )
$k_i$	Número de máquinas no estágio $i$ .
$m$	Número total de máquinas.
$M(S)$	Duração total da programação da sequência $S$ .
$M_i$	Máquina $i$ do conjunto de máquinas.
$n$	Número de tarefas a serem seqüenciadas
$p'_{1i}$	Tempo de processamento da tarefa $J_i$ sobre a máquina artificial 1.
$P'_{2i}$	Tempo de processamento da tarefa $J_i$ sobre a máquina artificial 2.
$p_{ij}$	Tempo de processamento da tarefa $J_i$ sobre a máquina $M_j$ .
$s_i$	<i>slope index</i> para a tarefa $J_i$ .
$STD_j$	Desvio padrão dos tempos de processamento da tarefa $j$ .
$UBM(S)$	Limitante superior do <i>makespan</i> da sequência $S$ .
$\pi$	Sequência de tarefas.
$\pi'$	Sequência de tarefas.

# SUMÁRIO

---

## RESUMO

## ABSTRACT

<b>1 PLANEJAMENTO E PROGRAMAÇÃO DA PRODUÇÃO</b> .....	13
1.1 Problemas de Programação de Operações em Máquinas .....	20
1.2 Objetivos da Pesquisa.....	22
1.3 Estrutura dos Capítulos.....	22
<b>2 FLOW SHOP PERMUTACIONAL</b> .....	25
2.1 Principais Métodos Heurísticos .....	28
<b>3 MÉTODO HEURÍSTICO PROPOSTO</b> .....	45
<b>4 EXPERIMENTAÇÃO COMPUTACIONAL</b> .....	51
4.1 Determinação das Amostras do Experimento .....	51
4.2 Método de Análise dos Resultados.....	52
4.3 Resultados da Experimentação Computacional .....	53
4.3.1 Comparação dos Métodos da Literatura Atual .....	53
4.3.2 Experimentação com o novo método heurístico proposto.....	62
<b>5 CONCLUSÕES</b> .....	71
<b>REFERÊNCIAS</b> .....	73
<b>APÊNDICE A – TABELAS DE RESULTADOS DOS MÉTODOS</b> .....	77
<b>APÊNDICE B – CÓDIGO FONTE DO MÉTODO PROPOSTO</b> .....	87



# CAPÍTULO 1

---

## 1 PLANEJAMENTO E PROGRAMAÇÃO DA PRODUÇÃO

A produção de qualquer produto ou serviço pode ser vista como um processo, ou um conjunto de processos, que tem como objetivo transformar insumos em produtos ou serviços. A execução, controle e planejamento das atividades de produção são de responsabilidade do Planejamento e Controle da Produção (PCP), o qual deve garantir que sejam produzidos os produtos certos, na quantidade certa e no tempo certo (SLACK et al., 2002). O propósito do planejamento e controle da produção é garantir que a produção ocorra de maneira eficaz e produza produtos e serviços da forma planejada, com a qualidade que os clientes esperam (SLACK, 2002).

As atividades que envolvem o PCP são, de alguma forma, direcionadas para a gestão das capacidades de execução de determinada operação com as demandas exigidas sobre ela (SLACK, 2002). Para entender melhor esse gerenciamento, é necessário conhecer a diferença entre planejamento e controle.

Um planejamento é o detalhamento do que se pretende fazer em um período de tempo definido, já controle, é o processo de fazer os ajustes necessários para que as operações e atividades sejam realizadas da forma como foi estabelecido.

Se planejamento e controle é o processo de conciliar demanda e fornecimento, então a natureza das decisões tomadas para planejar e controlar uma operação produtiva dependerá tanto da natureza da demanda como da natureza do fornecimento nessa operação [...] (SLACK, 2002, p. 233)



O PCP deve gerenciar os processos produtivos, sujeito às restrições tecnológicas do ambiente, de modo a suprir as necessidades de demanda, ajustando a produção aos volumes de demanda e tempo de execução das atividades.

Dessa forma, para conciliar volume e tempo são necessárias três atividades distintas, mas integradas: o carregamento, a sequência e a programação.

O carregamento pode ser visto como a quantidade de trabalho alocada em um determinado centro de trabalho, quantidade esta que pode ser considerada fixa para um determinado período de tempo, ou pode ser contínuo de forma que a quantidade de trabalho varia com a saída das atividades acabadas.

O sequenciamento é a tomada de decisões sobre qual a ordem em que as operações devem ser executadas, esta atividade possui alta complexidade e pode ser explicada da seguinte forma: considere o sequenciamento de 5 atividades em uma máquina, todas as atividades são independentes uma da outra, e desta forma existem 5! diferentes sequências para a execução das atividades, resultando em 120 possíveis soluções para este problema. Em um ambiente com 10 atividades o número de possíveis soluções aumenta para 3.268.800. Este é um problema pequeno em comparação aos encontrados em ambientes reais.

A programação da produção envolve as atividades que procuram definir as quantidades de cada produto, as datas de início e de término e os equipamentos que serão utilizados na fabricação desses produtos.

Graves (1981) definiu a programação da produção como sendo a alocação, no tempo, dos recursos disponíveis de produção de tal forma que satisfaça, da melhor maneira possível, um conjunto de critérios. Para ele, o problema de programação da produção envolve um conjunto de tarefas a serem realizadas e os critérios, que podem envolver decisões entre o término mais cedo e/ou mais tarde de cada uma delas.

O objetivo da programação é encontrar uma maneira adequada de atribuir e sequenciar o uso de recursos compartilhados de tal forma que as restrições de produção sejam satisfeitas e os custos de produção sejam minimizados (RODAMMER; WHITE JR., 1981).

Pode-se afirmar, em nível geral, que a programação da produção e os estoques acomodam a demanda pelos produtos. No entanto, existem variáveis controláveis (níveis de estoque, carga-máquina, etc.), que são externas a ela, mas que afetam o desempenho do sistema produtivo como um todo.

A programação da produção envolve, portanto, a consideração dos fatores externos e internos, cada um deles conduzindo a estratégias diferentes. Em termos da orientação externa, é importante notar que a programação procura acomodar a influência da demanda, representada pelas solicitações dos clientes, em quantidade e prazo. Por outro lado, os aspectos internos da programação da produção estão ligados à utilização eficiente dos recursos (produtividade).

Orientada para os aspectos externos, a programação da produção, envolve basicamente a consideração do prazo. Estabelecer prazos consiste em subtrair de uma data de término (de um prazo de entrega), os tempos de execução (ou duração) mais as tolerâncias de fabricação. Por meio deste procedimento, são obtidas as datas de início de fabricação dos produtos ou de execução das atividades.

Dependendo do tipo do processo produtivo, os prazos são estabelecidos de formas diferentes, pois dependem de fatores específicos, exemplificado na Figura 1.1.

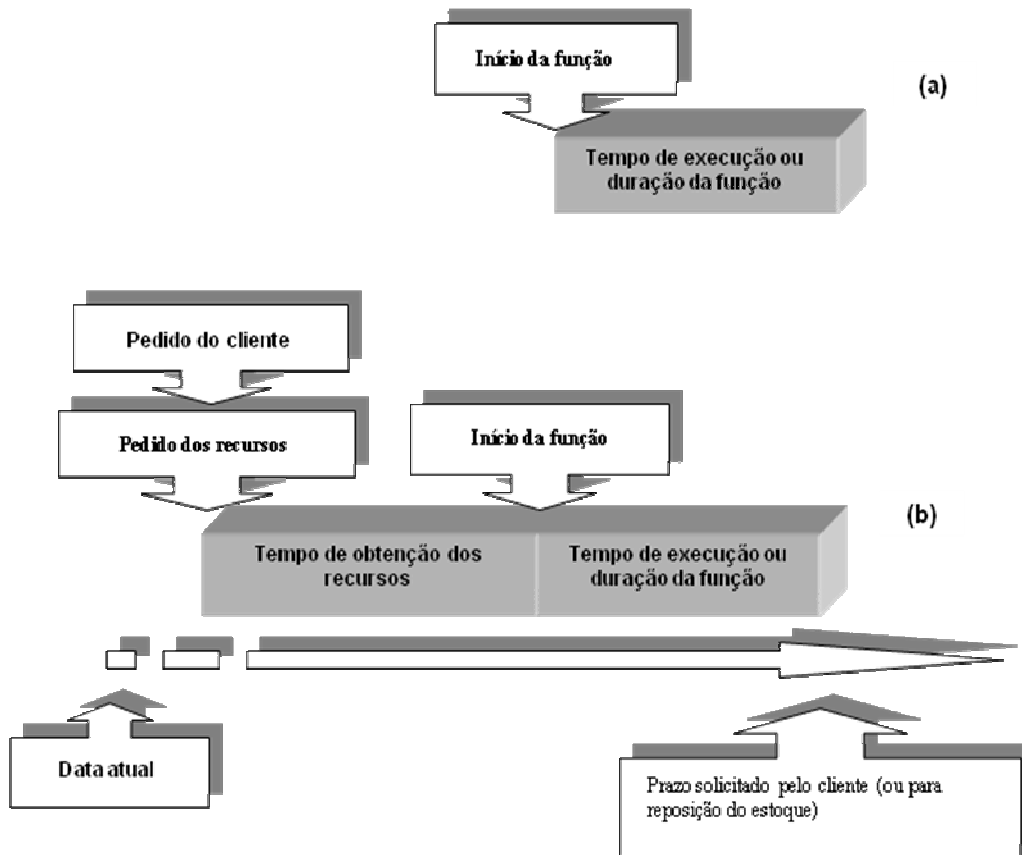


Figura 1.1 – Programação da produção em função dos prazos solicitados

Verifica-se acima que, a situação do tipo (a) aplica-se nos casos onde já estão disponíveis as matérias-primas que serão utilizadas na fabricação. O prazo para início da função é obtido subtraindo-se do prazo solicitado pelo cliente (ou prazo para a reposição do estoque) o tempo de execução da função, que pode complementar tempos de espera devido à indisponibilidade de equipamentos (outros produtos podem estar em fabricação).

Na situação do tipo (b), pela não disponibilidade de matérias-primas (ou outro recurso), o pedido do cliente só começa a ser efetivamente processado após a obtenção dos recursos. Assim, é preciso coordenar, além da programação em si, a obtenção dos recursos necessários.

Além dessas duas situações, pode ocorrer uma terceira, na qual não há um procedimento reverso de programação. Nesta situação, o pedido do cliente marca o início da função, conforme se visualiza na Figura 1.2.

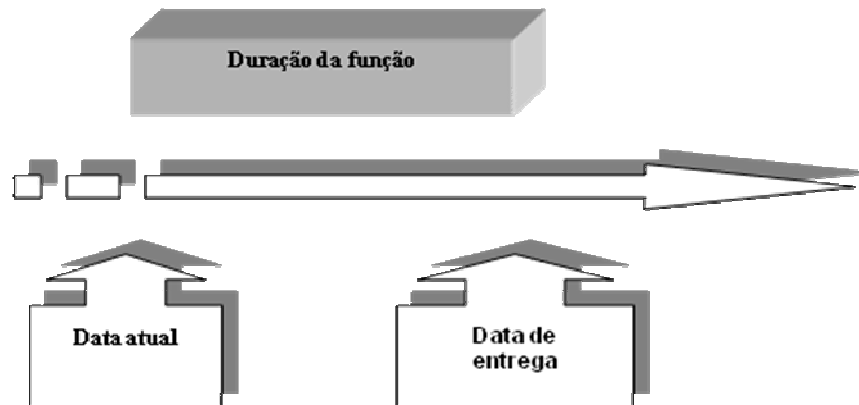


Figura 1.2 – Programação da produção em função da data de recebimento dos pedidos

Pode-se perceber que o prazo de entrega (ou término) é obtido a partir da duração da função (ou da atividade). O início imediato da função, uma vez recebido o pedido do cliente, só depende da disponibilidade de recursos.

A programação da produção orientada para aspectos internos tem como estratégia a utilização eficiente da capacidade, através da coordenação das atividades simultâneas que ocorrem internamente. Essas atividades referem-se à ordenação das tarefas, ou seja, quais tarefas devem ser realizadas (viáveis e disponíveis) e em qual ordem. Este tipo de programação busca obter a melhor eficiência dos recursos, como por exemplo, a máxima utilização das máquinas.

Sintetizando, a programação da produção envolve a especificação, em antecipação, dos prazos de ocorrência dos eventos dentro da função de fabricação.

Os problemas de programação da produção, face aos aspectos externos e internos, estão intimamente relacionados aos de capacidade e de estoques.

Nas situações (funções) em que não há estoque de produtos acabados, as decisões de programação serão influenciadas diretamente pela demanda imposta pelos clientes. Nesses casos, o grau de repetitividade das tarefas é pequeno e, conseqüentemente, a necessidade de controle passa a ser grande, tornando a programação mais complexa.

Em contraste, nas situações em que há estoque, a programação deve examinar três estágios do sistema, sendo eles: o prazo para o produto final entrar no estoque, a determinação do início da fabricação do produto; a disponibilidade de matérias-primas para a fabricação. Apesar de envolver mais estágios e ser mais complexa tecnicamente, o fato de o cliente não estar envolvido diretamente simplifica a determinação dos prazos (a orientação é para a reposição de estoques).

Em geral, um nível alto de demanda está associado à repetitividade da função e à existência de recursos com finalidades específicas, bem como à especialização (padronização) de produtos e de serviços. Já um nível baixo de demanda pode estar associado a uma grande variedade de produtos e serviços não padronizados, que levam a uma baixa repetitividade das tarefas, exigindo máquinas e equipamentos de uso geral (universais).

As medidas ou critérios para avaliar a eficiência da programação da produção podem incluir:

- Nível de produtos acabados ou trabalhos em andamento/progresso (orientação interna);
- Porcentagem de utilização dos recursos (externa e interna);
- Porcentagem das ordens entregues no prazo, ou antes dele (orientação externa);
- Porcentagem de faltas nos estoques (orientação interna);

- Quantidade de clientes perdidos (externa e interna);
- Tempo de preparação das máquinas (externa e interna);
- Tempo parado por outros motivos: quebras, falta de ocupação, etc. (orientação interna);
- Tempo que o cliente espera (orientação externa).

Sendo assim, a programação da produção orientada externamente procura satisfazer objetivos ligados ao nível de serviços ao cliente, enquanto a orientada internamente procura atingir a produtividade dos recursos. A estratégia a ser utilizada na programação da produção pode variar de empresa para empresa, mas alguns objetivos básicos são preservados.

Os principais objetivos são:

- Entregar os produtos fabricados nas datas compromissadas ou estabelecidas;
- Distribuir a carga de trabalho de forma a obter máxima utilização dos recursos;
- Garantir que toda a matéria-prima e componentes comprados estejam disponíveis quando forem solicitados pela fabricação;
- Prever e evitar grande concentração de trabalho em poucas máquinas (gargalos de produção);
- Prever a ociosidade da capacidade produtiva;
- Estabelecer seqüências de produção que minimizem o tempo de equipamento sem trabalho.

Para executar a programação da produção atingindo seus objetivos, existem técnicas e métodos que foram desenvolvidos com fins específicos, seja para atender aos aspectos externos, seja para satisfazer as orientações internas. Entre os métodos e técnicas mais difundidos estão os Gráficos de GANTT, o Diagrama de Montagem, as técnicas de Redes

(CPM e PERT) e diversos métodos heurísticos que determinam o melhor seqüenciamento da produção.

## 1.1 PROBLEMAS DE PROGRAMAÇÃO DE OPERAÇÕES EM MÁQUINAS

Os problemas de programação de operações em máquinas vêm sendo caracterizados por diversos autores em diferentes formas, dentre eles Baker, 1974; Blazewicz et al., 1996; Conway et al., 1967; French, 1982; Graves, 1981 e Pinedo, 2008.

Em situações de programar as operações nas máquinas disponíveis surgem problemas complexos. Pois, as restrições tecnológicas e a medida de desempenho da programação devem ser especificadas. As restrições tecnológicas são determinadas principalmente pelo fluxo das tarefas nas máquinas.

Neste contexto, Maccarthy e Liu (1993) classificam os problemas de programação de operações da seguinte forma:

- **Flow shop** - em que todas as tarefas possuem o mesmo fluxo de processamento em todas as máquinas;
- **Job shop** - em que todas as tarefas possuem um roteiro específico de processamento, determinado para cada tarefa;
- **Open shop** - em que não existem roteiros de processamento preestabelecidos para as tarefas;
- **Flow shop permutacional** - *flow shop* onde a ordem de processamento das tarefas é exatamente a mesma para todas as máquinas;
- **Máquina única** - existe somente uma única máquina disponível para a execução das tarefas;

- **Máquinas paralelas** - são disponíveis duas ou mais máquinas, que podem executar qualquer tarefa, onde a tarefa só é executada em uma máquina;
- **Job shop com máquinas múltiplas** - é um *job shop* no qual existem  $k_i$  máquinas idênticas em cada estágio  $i$  ( $i = 1, 2, \dots, m$ ), sendo que, em cada estágio, cada tarefa é processada por somente uma máquina.
- **Flow shop com máquinas múltiplas** - é um *flow shop* onde as tarefas são processadas em múltiplos estágios seguindo a mesma ordem em cada um deles, sendo possível a variação da quantidade por estágio e as tarefas são processadas apenas por uma máquina em cada estágio.

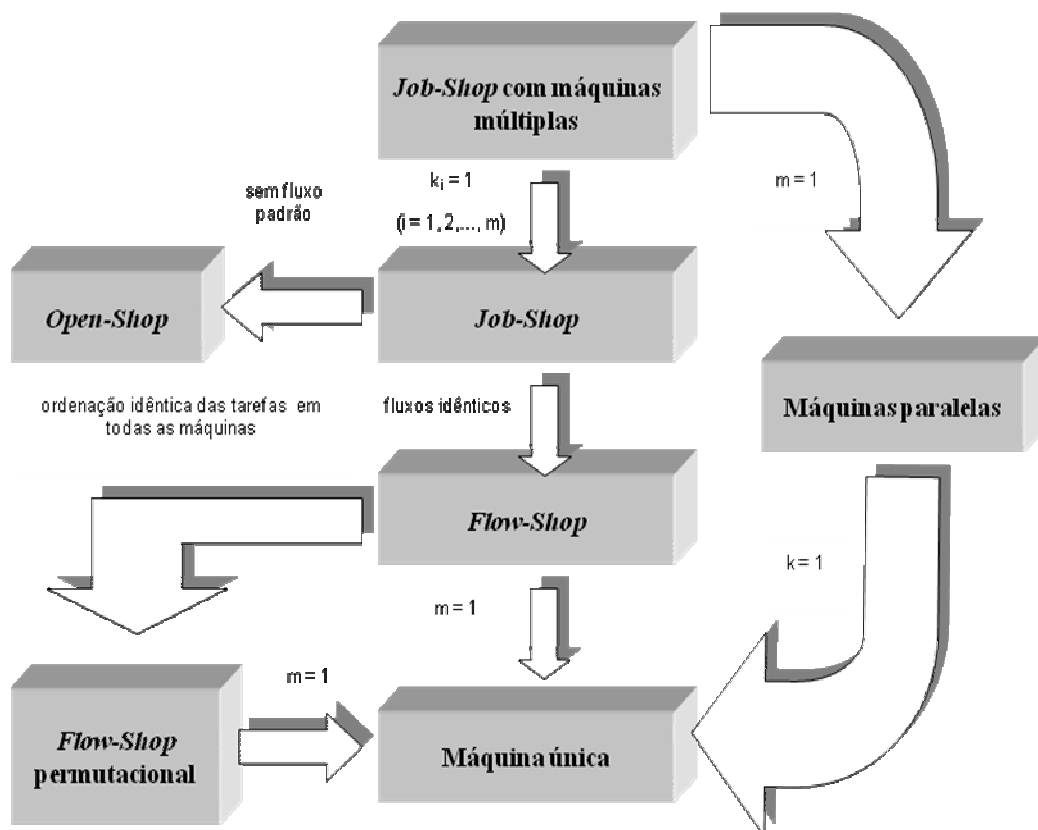


Figura 1.3 – Relação entre as diversas classes de problemas de programação de operações em máquinas  
 Fonte – Adaptado de Maccarthy e Liu (1993)



O estudo apresentado nesta pesquisa é direcionado ao problema de Programação de Operações em Ambiente *Flow Shop* Permutacional. Neste problema, independentemente do número de máquinas presentes no sistema, o número de soluções viáveis é  $(n!)$ , em que  $n$  representa o número de tarefas a serem programadas. Existe uma grande diversidade de métodos heurísticos, meta-heurísticos e métodos exatos para a solução deste tipo de problema de programação da produção. No capítulo 2 serão apresentados os métodos heurísticos construtivos para a solução do problema.

## **1.2 OBJETIVOS DA PESQUISA**

Os principais objetivos desta pesquisa são:

- 1) Comparar os melhores métodos heurísticos construtivos presentes na literatura, que não foram comparados diretamente em uma mesma experimentação computacional, para determinar qual método apresenta resultados na busca por melhores soluções para o problema;
- 2) Desenvolver, testar computacionalmente e propor um novo método heurístico construtivo de alto desempenho que resulte em melhores soluções.

## **1.3 ESTRUTURA DOS CAPÍTULOS**

Os próximos capítulos seguem a seguinte estrutura. No capítulo 2, é descrito o problema de programação de operações *flow shop* permutacional, seus conceitos e suposições e uma revisão da literatura com os melhores métodos heurísticos para a solução do problema. No

capítulo 3, é apresentado o método proposto nesta pesquisa. Em seguida, no capítulo 4, são expostos os resultados da experimentação computacional dos métodos heurísticos construtivos para a identificação do melhor método juntamente com o método aqui proposto. E, no capítulo 5, por fim, são expostas as conclusões da pesquisa e sugestões para trabalhos futuros.



# CAPÍTULO 2

---

## 2 O PROBLEMA DE PROGRAMAÇÃO *FLOW SHOP* PERMUTACIONAL

O problema de programação de operações *flow shop* permutacional é um problema onde se mantêm a mesma ordem de programação das  $n$  tarefas em todas as máquinas. Basicamente com o objetivo de determinar entre as  $n!$  possíveis sequências aquela que minimize alguma função objetivo estabelecida, tais como tempo total da programação das tarefas (*makespan*), tempo ocioso das máquinas (*idletime*), tempo de fluxo das tarefas (*flowtime*), entre outros.

Para a modelagem deste tipo de ambiente, faz-se necessário assumir algumas hipóteses:

- Cada máquina está disponível continuamente, sem interrupções;
- Cada operação tem, no máximo, uma sucessora e uma precedente (fluxo unidirecional);
- Cada operação pode ser executada por apenas uma máquina;
- Sempre que uma operação é iniciada, a mesma é finalizada sem que haja interrupção no seu processamento;
- Os tempos de preparação das máquinas (*setup*) estão inclusos nos tempos de processamento e estes independem das tarefas precedentes;
- Cada máquina processa apenas uma tarefa de cada vez e cada tarefa é processada por apenas uma máquina.

Os métodos heurísticos para solução deste tipo de problemas podem ser classificados de diferentes maneiras. Uma delas classifica-os em Construtivos ou Melhorativos, dependendo

da forma de obtenção da solução. Os métodos construtivos se caracterizam pelo fato de gerarem uma solução parcial que será utilizada na construção do resultado final do problema (CAMPBELL et al., 1970; DANNENBRING, 1977; DAVOUD POUR, 2001; DONG; HUANG e CHEN, 2008; GUPTA, 1971; KALCZYNSKI e KAMBUROWSKI, 2007; KALCZYNSKI e KAMBUROWSKI, 2008; KOULAMAS, 1998; NAWAZ et al., 1983; PALMER, 1965; NAGANO e MOCCELLIN, 2002; RAD; RUIZ e BOROOJERDIAN, 2009). Esta solução pode ser gerada:

- Diretamente a partir da ordenação das tarefas segundo índices de prioridade calculados em função dos tempos de processamento das tarefas, como por exemplo, Palmer (1965);
- Escolhendo-se a melhor seqüência das tarefas a partir de um conjunto de seqüências também obtidas utilizando-se índices de prioridade associados a cada tarefa, como por exemplo, Campbell, Dudek e Smith (1970) e Hundal e Rajgopal (1988);
- Ou ainda, a partir da geração sucessiva de seqüências parciais das tarefas (subseqüências) até a obtenção de uma seqüência completa através de algum critério de inserção de tarefas, como por exemplo, Dong; Huang e Chen (2008); Kalczyński e Kamburowski (2007); Nagano e Moccellin (2002); Nawaz; Enscore e Ham (1983); Rad, Ruiz e Boroojerdian (2009).

No caso dos métodos melhorativos, obtém-se uma solução inicial e, posteriormente, por meio de algum procedimento iterativo (geralmente envolvendo trocas de posições das tarefas na seqüência), busca-se obter uma seqüência das tarefas melhor que a atual quanto à medida de desempenho adotada. Por exemplo: Grabowski e Wodecki (2004); Nowicki e Smutnicki (1996); Ogbu e Smith (1990); Osman e Potts (1989); Reeves (1993); Taillard (1990) e Widmer e Hertz (1989). Outros algoritmos são: o *path-based method* de Werner (1993) ou o

*iterated local search* (ILS) de Stützle (1998). Rajendran e Ziegler (2004) apresentaram dois novos métodos baseados no *ant-colony optimization* (ACO).

Framinan, Gupta e Leisten (2004) propuseram uma estrutura geral para os métodos heurísticos, onde um método pode ser composto por uma ou mais fases, sendo elas as seguintes:

Fase 1 - *Index Development*: O objetivo dessa fase é obter uma ordem das tarefas de acordo com alguma regra de prioridade, baseada nos tempos de processamento das tarefas do problema, esta regra pode ser obtida de diversas formas;

Fase 2 - *Solution Construction*: Nessa fase, a solução é construída inserindo-se uma ou mais tarefas de cada vez em uma seqüência parcial, até que todas as tarefas sejam inseridas. Essa inserção é feita avaliando a  $k$ -ésima tarefa da ordenação da Fase 1 nas  $k$  posições possíveis na seqüência parcial e escolhendo a seqüência que minimiza um determinado critério de avaliação previamente determinado. Com isso, uma tarefa é retirada do conjunto de tarefas não seqüenciadas e colocada no conjunto de tarefas seqüenciadas. A inserção é feita até que o conjunto de tarefas não seqüenciadas fique vazio;

Fase 3 - *Solution Improvement*: Nessa fase tenta-se, por meio da utilização da solução obtida na Fase 2, obter uma nova solução que seja melhor que a da fase anterior. O procedimento dessa fase é a aplicação de mudanças na seqüência de tarefas de forma que melhore a solução final.

O problema de seqüenciamento de tarefas em um ambiente de produção *flow shop* é foco de várias pesquisas nos últimos 50 anos e, para fins práticos, tal problema pode ser considerado já resolvido. No entanto, tendo em vista sua complexidade, a busca de métodos heurísticos simples e mais eficazes quanto à qualidade da solução ainda permanece como uma direção de pesquisa.

## 2.1 PRINCIPAIS MÉTODOS HEURÍSTICOS.

A seguir, são descritos os principais métodos heurísticos com o critério de minimização da duração total da programação (*makespan*). Para a revisão apresentada adotam-se as seguintes notações:

Seja  $J = \{J_1, J_2, \dots, J_j, \dots, J_n\}$  um conjunto de  $n$  tarefas que devem ser processadas, na mesma seqüência, por um conjunto de  $m$  máquinas distintas. O tempo de processamento da tarefa  $J_j$  na máquina  $i$  é  $p_{ij}$  ( $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ ). Se uma tarefa não tiver operação em uma determinada máquina, seu correspondente tempo de processamento é igual a zero.

Desde que Johnson (1954) propôs uma solução ótima (exata) para o problema de  $n$  tarefas sendo processadas em 2 máquinas, vários outros métodos foram desenvolvidos para resolver o problema de seqüenciamento em sistemas *flow shop* de  $n$  tarefas e  $m$  máquinas, minimizando o critério do *makespan*. Em todos eles, muitas restrições foram feitas, as quais, embora distantes de situações reais, simplificaram o problema.

Palmer (1965) propôs um índice denominado *slope index*, a partir do qual se estabelece a seqüência de processamento das tarefas nas máquinas. Tal índice é calculado de forma que as tarefas em que os tempos de processamento tendem a crescer na seqüência das máquinas devem receber maior prioridade na programação, ou seja, devem ocupar as primeiras posições na ordem de execução. Tal índice, chamado *slope index*, para uma tarefa  $J_j$  é dado por:

$$s_j = \sum_{i=1}^m (2i - m - 1) p_{ij} \quad \text{para } j = 1, 2, \dots, n.$$

A partir dos valores de  $s_j$ , estabelece-se a seqüência de programação das tarefas, de acordo com a ordenação não-crescente dos índices.

Campbell, Dudek e Smith (1970) propuseram um procedimento conhecido por CDS, que é uma generalização do algoritmo de Johnson para a solução exata do problema com  $m$  igual a 2. Sua eficiência é atribuída a duas razões básicas: (i) origina  $m-1$  subproblemas artificiais de 2 máquinas a partir do problema original de  $m$  máquinas, (ii) utiliza para resolvê-los o método de Johnson no estilo heurístico, ou seja, corresponde à utilização da Regra de Johnson em  $(m-1)$  estágios, em cada um dos quais é obtido um problema com apenas 2 máquinas, com tempos de processamento “artificiais”  $p'_{1j}$  e  $p'_{2j}$  ( $j=1, 2, \dots, n$ ). No estágio 1,  $p'_{1j} = p_{1j}$  e  $p'_{2j} = p_{mj}$ , ou seja, a Regra de Johnson é aplicada somente considerando a primeira e a última máquinas, sendo as demais desprezadas. No estágio 2,  $p'_{1j} = p_{1j} + p_{2j}$  e  $p'_{2j} = p_{(m-1)j} + p_{mj}$ , ou seja, aplica-se a Regra de Johnson à soma dos tempos de processamento da primeira com a segunda máquina e da penúltima com a última. No estágio  $t$ , os tempos de processamento das duas máquinas “artificiais” serão:

$$p'_{1j} = \sum_{i=1}^t p_{ij} \text{ e } p'_{2j} = \sum_{i=1}^t p_{(m-i+1)j}$$

Em cada um dos  $(m-1)$  estágios, a seqüência de tarefas obtida pela Regra de Johnson é utilizada para calcular a duração total da programação (*makespan*) do problema original. A seqüência que fornecer a menor duração é escolhida como solução para o problema.

Gupta (1971) sugeriu outro algoritmo similar ao de Palmer (1965), exceto pela forma como define o índice. Gupta reconheceu que o algoritmo de Johnson (1954) para o problema com 2 ou 3 máquinas é, na verdade, um método de ordenação a partir da designação de um índice para cada tarefa, seqüenciando-as de acordo com a ordem crescente de tais índices.



Gupta generalizou a função de indexação, para o caso de  $m \geq 4$  máquinas, definindo, para cada tarefa  $J_j$ , o seguinte índice:

$$f(j) = \frac{A}{\min_{1 \leq i \leq m-1} (p_{ij} + p_{i+1,j})} \text{ para } j=1, 2, \dots, n,$$

onde:

$$A = \begin{cases} 1, & \text{se } p_{ij} \leq p_{1j} \\ -1, & \text{caso contrario.} \end{cases}$$

Dannenbring (1977) sugeriu uma variação para o algoritmo CDS. O método é chamado Procedimento *Rapid Access* (RA), o qual procura combinar as vantagens do *slope index* de Palmer com as do método CDS, obtendo uma boa solução, de maneira simples e rápida. Ao invés de resolver  $(m-1)$  problemas “artificiais” com 2 máquinas, o método RA resolve um único problema, no qual os tempos de processamento de cada tarefa em cada máquina são determinados por um esquema de peso, como exemplificado a seguir:

$$p_{1j} = \sum_{i=1}^m (m-i+1)p_{ij} \text{ e } p_{2j} = \sum_{i=1}^m (i)p_{ij} \text{ para } j=1, 2, \dots, n.$$

A partir da solução fornecida pelo procedimento RA, Dannenbring propôs ainda dois métodos, o RAES e o RACS, para garantir uma melhor solução. Ambos utilizam, como solução inicial, a seqüência gerada pelo método RA. Tais processos procuram encontrar, entre as seqüências “vizinhas” daquela fornecida pela solução inicial, uma seqüência de programação que fornece uma menor duração total. Dannenbring define como “vizinha” uma nova seqüência das tarefas, obtida a partir da solução inicial, pela troca de posições entre duas tarefas adjacentes. Por meio do processo RACS, são examinadas  $(n-1)$  novas seqüências (vizinhas), adotando-se como solução a seqüência associada ao menor *makespan*, desde que

seja menor que o da solução inicial. Por meio do processo RAES, ao invés de terminar o procedimento de melhoria após examinar as  $(n-1)$  seqüências vizinhas da solução inicial, ele usa a melhor seqüência para gerar as suas respectivas vizinhas, e assim sucessivamente, até encontrar uma seqüência cujas vizinhas não apresentam uma menor duração total da programação.

Nawaz, Enscore Jr. e Ham (1983) desenvolveram um algoritmo (conhecido por NEH) baseado na hipótese de que às tarefas devem ser designadas prioridades de programação diretamente proporcionais às somas dos seus tempos de processamento nas  $m$  máquinas. É interessante, neste ponto, ressaltar que o algoritmo NEH não transforma o problema original de  $m$  máquinas em um problema artificial de 2 máquinas, à semelhança dos algoritmos CDS e RA. O algoritmo NEH é definido pelos seguintes passos:

Passo 1 – Calcula-se, para cada tarefa, a soma dos tempos de processamento em todas as máquinas:

$$\sum TP_v = \sum_{k=1}^m p_{kv} \quad \text{para } v = 1, 2, \dots, n.$$

Ordena-se as  $n$  tarefas de acordo com os valores não crescentes das somas dos tempos de processamento.

Passo 2 – Seleciona-se as duas primeiras tarefas da ordenação, seqüenciando-as de maneira a minimizar o *makespan*.

Passo 3 – Para  $j = 3, 4, \dots, n$ , faça:

Selecione a  $j$ -ésima tarefa da ordenação obtida no passo 2 e examine as  $j$  possibilidades de inserção dessa tarefa na sub-seqüência já feita quanto à

minimização do *makespan*, adotando a que apresenta menor duração total da programação parcial.

Hundal e Rajgopal (1988) desenvolveram uma extensão do algoritmo de Palmer, a partir do fato de que este algoritmo ignora a máquina  $(m+1)/2$  quando  $m$  é ímpar, o que pode afetar a qualidade da solução, especialmente quando o número de tarefas é grande. A extensão do algoritmo de Palmer é considerada a partir de dois novos conjuntos de *slope index*, dados por:

$$s_j = \sum_{i=1}^m (2i-m)p_{ij} \quad \text{e} \quad s_j = \sum_{i=1}^m (2i-m-2)p_{ij}, \quad \text{para } j = 1, 2, \dots, n.$$

Desta forma, duas outras seqüências são obtidas, sendo selecionada a que apresentar menor *makespan*.

Ho e Chang (1991) apresentaram um método composto, no qual a solução inicial é obtida utilizando métodos já existentes, tais como Palmer, CDS, Gupta, Dannenbring (RA) e Hundal. Na segunda fase, propuseram um procedimento de melhoria da solução inicial baseado em relações entre os tempos de processamento das tarefas, consideradas em pares, ou seja, mais especificamente, utilizando-se as diferenças:

$$d_{ij}^k = p_{k+1,i} - p_{kj}$$

para

$$i, j = 1, 2, \dots, n,$$

$$k = 1, 2, \dots, (m-1)$$

e  $i \neq j$ .

onde:

$p_{k+1,i}$  é o tempo de processamento da tarefa  $J_i$  na máquina  $(k + 1)$  e

$p_{kj}$  o tempo de processamento da tarefa  $J_j$  na máquina  $k$ .

Moccellin (1995) apresentou um método melhorativo para a solução do problema de Programação de Operações *Flow shop* Permutacional denominado FSHOPH que utiliza uma analogia com o problema do Caixeiro-viajante, porém a expressão referente à “distância” entre duas tarefas é totalmente diferente daquela formulada por Widmer e Hertz (1989).

No problema de Programação de Operações *Flow shop* Permutacional, considerando-se uma seqüência qualquer  $S$  das  $n$  tarefas nas  $m$  máquinas, a Duração Total da Programação  $M(S)$  é dada por:

$$M(S) = \sum_{j=1}^n p_{mj} + \sum_{j=0}^{n-1} X_{j+1}^m$$

onde:

$p_{mj}$  = tempo de processamento, na última máquina, da tarefa que ocupa a  $j$ -ésima posição na seqüência  $S$ ;

$X_{j+1}^m$  = intervalo de tempo entre o término da operação da tarefa que ocupa a  $j$ -ésima posição na seqüência  $S$  e o início da operação da tarefa que ocupa a posição  $(j+1)$  de  $S$ , na última máquina; e

$j = 0$ , corresponde a uma tarefa fictícia com tempos de processamento nulos que ocupa sempre a 1ª posição em qualquer seqüência das  $n$  tarefas.

Se  $X_{j+1}^m$  for considerado como a “distância” entre as tarefas da seqüência  $S$  que ocupam as posições  $j$  e  $(j+1)$ , então o problema de Programação de Operações *Flow shop* Permutacional torna-se análogo ao problema Direcionado do Caixeiro-viajante. Assim, a seqüência  $S^*$  que minimiza a Duração Total da Programação é dada pela “rota mínima que interliga” as  $n$  tarefas. Entretanto, deve-se observar que os valores  $X_{j+1}^m$  não são previamente conhecidos, uma vez que dependem da seqüência  $S$  que for considerada. Desta forma, as distâncias entre as tarefas devem ser estimadas a partir dos dados do problema, ou seja, dos tempos de processamento das tarefas, de maneira que a solução do problema do Caixeiro-viajante corresponda a uma solução heurística do problema original de Programação de Operações.

No método FSHOPH, a distância entre duas tarefas quaisquer é fundamentada em uma propriedade do problema de Programação de Operações *Flow shop* Permutacional identificada a partir de investigações que foram realizadas sobre as características do problema e sua solução (MOCCELLIN, 1992). Tal propriedade é descrita por:

$$UBX_{j+1}^{k+1} = \max(0, UBX_{j+1}^k + (p_{k,j+1} - p_{k+1,j}))$$

onde

$$UBX_{j+1}^1 = 0 \text{ e}$$

$UBX_{j+1}^{k+1}$  = um Limitante Superior de  $X_{j+1}^{k+1}$  (intervalo de tempo entre o término da operação da tarefa que ocupa a  $j$ -ésima posição na seqüência  $S$  e o início da operação da tarefa que ocupa a posição  $(j+1)$  de  $S$ , na máquina  $(k+1)$ );

$UBX_{j+1}^k$  = um limitante Superior de  $X_{j+1}^k$ ;

$p_{k,j+1}$  = tempo de processamento na máquina  $k$  da tarefa que ocupa a posição  $(j+1)$ ;

$p_{k+1,j}$  = tempo de processamento na máquina  $(k+1)$  da tarefa que ocupa a posição  $j$ .

A partir dessa propriedade, tem-se que:

- a) Para uma seqüência qualquer  $S$  das  $n$  tarefas, a Duração Total da Programação  $M(S)$  é limitada superiormente por:

$$UBM(S) = \sum_{j=1}^n p_{mj} + \sum_{j=0}^{n-1} UBX_{j+1}^m$$

- b) Se as tarefas  $u$  e  $v$  são respectivamente programadas nas posições  $j$  e  $(j+1)$ , então para qualquer  $j = 0, 1, 2, \dots, n-1$ , tem-se que:

$$UBG_{uv} = \max(0, UBX_{uv}^{m-1} + (p_{m-1,v} - p_{mu}))$$

Onde  $UBG_{uv}$  é um Limitante Superior do tempo de espera, na última máquina, entre as operações das tarefas adjacentes  $u$  e  $v$ .

No primeiro passo básico do método FSHOPH, a “distância” de uma tarefa  $u$  a uma outra tarefa  $v$  é dada por  $UBG_{uv}$  e busca-se obter a “rota” (seqüência  $S$  das  $n$  tarefas) que minimiza o Limitante Superior da Duração Total da Programação  $UBM(S)$ . No segundo passo básico, melhora-se a solução inicial, utilizando-se a técnica de Busca Tabu.

Sevast’janov (1995) propôs um algoritmo para o problema de *flow shop* permutacional, reduzindo para um problema de adição de vetores. Nesse caso, o algoritmo de Sevast’janov obtém uma seqüência com um valor  $C_{max}$ , tal que:

$$C_{max} - C_{max}^* \leq \left( m^2 - 3m + 3 + \frac{1}{m-2} \right) \max_{ij} (p_{ij})$$

onde:

$m$  é o número de máquinas e  $C_{max}^*$  é o valor ótimo do *makespan*.

Lourenço (1996) implementou o algoritmo de Sevast'janov usando Programação Linear. Na experimentação efetuada, se o tempo de computação não for considerado um fator crítico, o algoritmo NEH apresenta um melhor desempenho do que o de Sevast'janov, quanto à qualidade da solução. No entanto, se o tempo de computação for considerado relevante, o algoritmo de Sevast'janov se apresenta como uma boa alternativa, especialmente em problemas com um grande número de tarefas e um número relativamente pequeno de máquinas. Deve ser salientado que o algoritmo de Sevast'janov é altamente dependente do programa computacional utilizado para resolver o problema de Programação Linear.

Koulamas (1998) apresentou um novo método heurístico construtivo denominado HFC para o problema de *flow shop* permutacional e também não-permutacional. O algoritmo HFC determina a posição de duas tarefas quaisquer na seqüência final baseado no problema de duas máquinas. A inspiração para o algoritmo HFC foi o algoritmo de Johnson, onde para duas máquinas este último fornece uma solução ótima. A principal idéia proposta no caso de 3 máquinas do algoritmo HFC é que se a tarefa  $i$  precede  $j$  na seqüência ótima, então para todos os casos M1-M2, M2-M3 e M1-M3 a tarefa  $i$  precede  $j$  em todas as fases. Cada tarefa  $i$  tem um índice de prioridade  $I_i$  inicialmente zero, e após determinada a relação de precedência das tarefas, os índices de prioridade  $I_i$  e  $I_j$  são quantificados, subtraindo 1 para a prioridade  $I_i$  da tarefa  $i$  e adicionando 1 na prioridade  $I_j$  da tarefa  $j$ . O processo é repetido para todos os pares de tarefas e, em seguida, os índices de prioridade são ordenados de forma não decrescente e a

seqüência das tarefas é obtida. O algoritmo HFC foi comparado com o NEH, para a verificação do seu desempenho foram realizadas duas experimentações: na primeira, os tempos de processamento de todas as tarefas foram gerados aleatoriamente num intervalo de variação discreta de [1,100] uniformemente distribuídos. Na segunda experimentação, os tempos de processamento das tarefas para cada problema foram gerados nos intervalos de [1,10] e [10,50]. Esta forma de geração aleatória dos tempos de processamento das tarefas permite, com maior chance, a formulação de problemas nos quais a solução ótima pode ser não-permutacional e com um subconjunto significativo de soluções não-permutacionais sub-ótimas, ou seja, com qualidade de solução superior às permutacionais. Na primeira experimentação computacional, os algoritmos HFC e NEH tiveram praticamente o mesmo desempenho. Na segunda, porém, o HFC apresentou melhor desempenho, o que seria esperado, uma vez que o NEH considera somente o espaço de soluções permutacionais.

Nagano e Moccasin (2002) propuseram uma heurística construtiva denominada N&M onde há uma mudança apenas na fase de ordenação inicial em relação ao NEH (1983). A ordenação inicial é feita em dois passos:

Passo 1 – Para cada tarefa calcular:

$$I_v = \sum TP_v - \max \sum LBY_{Jv}$$

onde,

$$\max \sum LBY_{Jv} = \max_{\substack{u=1 \\ u \neq v}}^n \left\{ \sum_{k=1}^{m-1} LBY_{uv}^k \right\}$$

$$\sum TP_v = \sum_{k=1}^m p_{kv}$$



$p_{ki}$  = tempo de processamento da tarefa  $i$  na máquina  $k$ ;

$n$  = número de tarefas

$m$  = número de máquinas

$$LBY_{uv}^k = \max\left(0, (p_{k+1,u} - p_{k,v}) - UBX_{uv}^k\right)$$

$$UBX_{uv}^k = \max\left(0, UBX_{uv}^{k-1} + (p_{k-1,v} - p_{k,u})\right)$$

sendo,

$$UBX_{uv}^1 = 0$$

A tarefa  $u$  é imediatamente precedente da tarefa  $v$ .

Passo 2 – Ordenar todos os  $I_i$  de forma decrescente e utilizar os próximos passos idênticos ao NEH. Este método apresentou resultados iguais ou superiores com relação ao NEH (NAWAZ; ENSCORE JR.; HAM, 1983).

Framinan et al., (2003) realizaram uma extensa experimentação computacional verificando mudanças na primeira fase (fase de indexação) do NEH, a qual avaliou as funções objetivo, tempo total da programação das tarefas (*makespan*), tempo ocioso das máquinas (*idletime*) e tempo total de fluxo das tarefas (*flowtime*). Nessa experimentação foram utilizados 177 critérios para a ordenação das tarefas. Na minimização do *makespan*, *idletime* e *flowtime* foram selecionados os cinco melhores métodos para a obtenção da seqüência inicial denominados B5MS (*best five makespan*), B5IT (*best five idletime*) e B5FS (*best five flowshop*) respectivamente. Para cada uma das 177 ordenações iniciais da seqüência com sua respectiva função objetivo foi calculado a porcentagem de desvio relativo (ARPD - *Average*

*Relative Percentage Deviation*). O resultados obtidos mostraram que os métodos propostos não superam o NEH com o critério de minimização do *makespan* e *idletime*. Entretanto, para a minimização do *flowtime*, o método proposto (B5FS) apresentou melhores resultados que o NEH.

Kalezynski e Kamburowski (2007) propuseram um novo método heurístico construtivo denominado NEHKK. O método proposto apresentou uma modificação do NEH no processo de construção da seqüência solução, onde as seqüências parciais obtidas no caso de empates nas soluções parciais são avaliadas levando em consideração a regra de Johnson. Os resultados obtidos através da experimentação computacional verificaram o melhor desempenho comparados com o N&M (NAGANO; MOCCELLIN, 2002) e o NEH (NAWAZ; ENSCORE JR.; HAM, 1983). O método NEHKK é constituído pelos seguintes passos:

Passo 1 – Calcule  $TP_j = \sum_{i=1}^m p_{ij}$ ,  $\forall n \in N$ , e coloque em ordem não-crescente;

Passo 2 – Selecciona-se as duas primeiras tarefas da ordenação, seqüenciando-as de maneira a minimizar o *makespan*, e faça  $\pi = (J_{[1]}, J_{[2]})$ .

Passo 3 – Insira próxima tarefa  $r$ , da ordenação  $TP_j$ , na primeira posição da seqüência parcial e assuma que a seqüência atual seja  $\pi^* = (r, J_{[2]}, J_{[3]}, \dots, J_{[k]})$ , e faça  $l^* \leftarrow 1$ .

Para  $l$  de 2 até  $n$  faça:

Teste  $r$  na posição  $l$  de  $\pi$  e avalie segundo o seguinte critério:

**SE** (o *makespan* da seqüência avaliada  $\pi$  é menor que o *makespan* da seqüência  $\pi^*$ ).

**OU SE** (o *makespan* da sequência avaliada  $\pi$  é igual ao *makespan* da sequência  $\pi^*$ ) **E** (o *makespan* da sequência  $(J_{l^*}, \dots, J_{l-1}, r)$  é menor que o *makespan* da sequência  $(r, J_{l^*}, \dots, J_{l-1})$ ).

Então faça  $l^* \leftarrow l$  e  $\pi^* \leftarrow \pi$ .

Dong, Huang e Chen (2007) propuseram um método heurístico construtivo baseado no método NEH, de Nawaz, Enscore Jr. e Ham (1983), chamado de NEH – D. O algoritmo altera a Fase I do NEH, criando uma regra de prioridade que é uma combinação linear da média ( $AVG_j$ ) e do desvio padrão ( $DEV_j$ ) dos tempos de processamento das tarefas, dadas da seguinte forma:

$$I_j = \alpha AVG_j + (1 - \alpha) DEV_j, \text{ com } \alpha \in [0, 1].$$

$$AVG_j = \frac{1}{m} \sum_{i=1}^m p_{ij},$$

$$DEV_j = \left[ \sum_{i=1}^m (p_{ij} - AVG_j)^2 \right]^{1/2},$$

$$STD_j = \left[ \frac{1}{m-1} \sum_{i=1}^m (p_{ij} - AVG_j)^2 \right]^{1/2}.$$

A combinação linear  $I_j = \alpha AVG_j + (1 - \alpha) DEV_j$  é utilizada para atribuir um valor para cada tarefa. Após atribuir o valor, as tarefas são arranjadas em ordem não-crescente da regra de prioridade, sendo esta a Fase I modificada do algoritmo NEH. Os autores propuseram ainda uma estratégia para os casos em que houver empates entre as tarefas. O desempate é calculado da seguinte forma:

Considere  $\pi(x)$  a tarefa na posição  $x$  de uma permutação  $\pi$ . Então  $p_{i,\pi(x)}$  é o tempo de processamento de  $\pi(x)$  sobre a máquina  $i$ ,  $C_{i,\pi(x)}$  é a primeira data de término da tarefa  $\pi(x)$  sobre a máquina  $i$ ,  $S_{i,\pi(x)}$  denota a última data de início da tarefa  $\pi(x)$  sobre a máquina  $i$ , então as duas seguintes medidas podem ser computadas para a tarefa  $\pi(x)$ .

$$E_{\pi(x)} = \frac{1}{m} \sum_{i=1}^m \frac{P_{i,\pi(x)}}{S_{i,\pi(x+1)} - C_{i,\pi(x-1)}}, \quad x = 1, \dots, n.$$

$$D_{\pi(x)} = \sum_{i=1}^m \left( \frac{P_{i,\pi(x)}}{S_{i,\pi(x+1)} - C_{i,\pi(x-1)}} - E_{\pi(x)} \right)^2, \quad x = 1, \dots, n.$$

Para o desempate, a tarefa é inserida na posição  $x$  com o menor  $D_{\pi(x)}$ . A seguir é apresentado o método NEH-D de forma detalhada:

Passo 1 – Calcule a média  $AVG_j$  e o desvio padrão  $STD_j$  dos tempos de processamento para cada tarefa  $j$ , e classifique as tarefas segundo a ordem não-crescente de  $I_j = AVG_j + STD_j$ ;

Passo 2 – Coloque a primeira tarefa, da ordenação do Passo 1, como seqüência parcial, como se a solução fosse somente essa tarefa;

Passo 3 – A partir da segunda tarefa faça, para  $k = 2$  até  $n$ , avalie o *makespan* da tarefa  $k$  em todas as possíveis posições da seqüência parcial e insira-a na posição que minimize o *makespan*; caso exista empate, a posição  $x$  com  $D_{\pi(x)}$  é escolhida, se ainda persistirem os empates, qualquer das posições é aceitável.

A complexidade do Passo 1 é a mesma que em NEH, a diferença está no Passo 3, visto que, pelo NEH – D, os empates são considerados utilizando  $D_{\pi(x)}$ .

Rad, Ruiz e Boroojerdian (2009) apresentaram novos métodos heurísticos construtivos para o problema, chamados de FRB1, FRB2, FRB3 e FRB4 e também uma meta-heurística chamada de FRB5. Dentre os algoritmos construtivos, o FRB3 e FRB4 foram os que obtiveram os melhores resultados. O método FRB3 é detalhado a seguir:

Passo 1 – Calcule  $TP_j = \sum_{i=1}^m p_{ij}$ ,  $\forall j \in n$ , e coloque em ordem não-crescente de  $TP_j$ , e  
 $\pi \leftarrow \emptyset$ ;

Passo 2 – Para  $l = 1$  até  $n$ , testar a tarefa  $h$ , na posição  $l$  de  $TP_j$ , em todas as possíveis posições de  $\pi$  e inserir a tarefa na posição com menor *makespan*;

Passo 2.1 – Para  $i = 1$  até  $l$ , retire a tarefa  $h$  na posição  $i$  de  $\pi$  e teste  $h$  em todas as possíveis posições de  $\pi$ , inserir a tarefa na posição com menor *makespan*;

O algoritmo FRB4 é uma variação do seu anterior FRB3, com uma alteração no processo de busca do passo 2.1. O FRB3 realiza a busca em  $l$  possíveis posições, já o FRB4 faz a busca em um intervalo definido entre o máximo de  $l$  e  $p - k$  e o mínimo de  $l$  e  $p + k$ , onde  $p$  é a posição em que a tarefa foi inserida no passo 2 e  $k$  é um valor pré-definido para o método e varia entre  $\{2, 4, 6, 8, 10, 12\}$ . O método FRB4 é apresentado a seguir:

Passo 1 – Calcule  $TP_j = \sum_{i=1}^m p_{ij}$ ,  $\forall j \in n$ , e coloque em ordem não-crescente de  $TP_j$ , e  
 $\pi \leftarrow \emptyset$ ;

Passo 2 – Para  $l = 1$  até  $n$ , testar a tarefa  $h$ , na posição  $l$  de  $TP_j$ , em todas as possíveis posições de  $\pi$  e inserir a tarefa na posição com menor *makespan*;

Passo 2.1 – Para  $i = \text{Max}(1, p - k)$  até  $\text{Mín}(l, p + k)$ , retire a tarefa  $h$  na posição  $i$  de  $\pi$  e teste-a em todas as possíveis posições de  $\pi$ . Insira a tarefa  $h$  na posição com menor *makespan*;

É válido ressaltar que dentre os métodos NEHKK, NEH-D, FRB3 e FRB4, ambos foram analisados separadamente por seus autores e comparados ao *benchmark* de Taillard. Neste trabalho, tais algoritmos serão analisados em conjunto com a heurística proposta, para validar os resultados de todos e verificar qual o melhor método heurístico construtivo presente na literatura atual. Será realizada uma experimentação computacional com duas bases de dados, a de Taillard e a de Ruiz para problemas *no idle*, com isso os métodos são comparados entre si e em seguida comparados ao método proposto nesse projeto de mestrado.



# CAPÍTULO 3

---

## 3 MÉTODO HEURÍSTICO PROPOSTO

Entre os métodos presentes na literatura atual, quatro devem ser citados como os melhores: NEHKK de Kalezynski e Kamburowski (2007), NEH-D de Dong, Huang e Chen (2009) e FRB3 e FRB4 de Rad, Ruiz e Boroojerdian (2009), tais métodos são apresentados por seus autores como sendo os melhores métodos heurísticos da atualidade para a solução do problema em questão. No entanto, não são comparados entre si, devido à proximidade de suas publicações.

O objetivo deste capítulo é apresentar um novo método heurístico modificado referente ao método FRB3, com aproximada eficiência computacional, mas como resultados superiores em comparação aos métodos NEHKK, NEH-D, FRB3 e FRB4. A idéia principal do novo método é a realização de uma avaliação na sua fase de construção da sequência solução, pela permutação de tarefas vizinhas a uma tarefa alocada na sequência. Isto é, sempre que uma tarefa for inserida na sequência, duas tarefas devem ser permutadas levando a uma nova sequência, se o *makespan* dessa nova sequência for menor que o da sequência atual, a sequência atual é então atualizada. Este procedimento é repetido de forma recursiva até que todas as tarefas estejam sequenciadas.

Desta forma o procedimento pode ser definido da seguinte forma:

Seja  $\pi = (J_{[1]}, J_{[2]}, \dots, J_{[h-1]}, J_{[h]}, J_{[h+1]}, \dots, J_{[k]})$  a sequência parcial de um conjunto de tarefas, onde  $k$  é o número de tarefas da sequência parcial, e  $J_{[h]}$  uma determinada tarefa inserida na posição  $h$  na sequência parcial  $\pi$ .



Se  $h > 1$  e  $h < k$  então faça  $\pi^* = (J_{[1]}, J_{[2]}, \dots, J_{[h+1]}, J_{[h]}, J_{[h-1]}, \dots, J_{[k]})$ , ou seja, permute de posição as tarefas  $J_{[h-1]}$  e  $J_{[h+1]}$  da seqüência  $\pi$ . Se  $C_{\max}(\pi^*) < C_{\max}(\pi)$ , então  $\pi \leftarrow \pi^*$ .

Se  $h = 1$  então faça  $\pi^* = (J_{[h]}, J_{[h+2]}, J_{[h+1]}, \dots, J_{[k]})$ , ou seja, permute de posição as tarefas  $J_{[h+1]}$  e  $J_{[h+2]}$  da seqüência  $\pi$ . Se  $C_{\max}(\pi^*) < C_{\max}(\pi)$ , então  $\pi \leftarrow \pi^*$ .

Se  $h = k$  então faça  $\pi^* = (J_{[1]}, J_{[2]}, \dots, J_{[h-1]}, J_{[h-2]}, J_{[h]})$ , ou seja, permute de posição as tarefas  $J_{[h-2]}$  e  $J_{[h-1]}$  da seqüência  $\pi$ . Se  $C_{\max}(\pi^*) < C_{\max}(\pi)$ , então  $\pi \leftarrow \pi^*$ .

O procedimento acima será chamado de ENG (*Exchange Nagano Gigante*) e será incorporado ao melhor método heurístico construtivo apresentado na literatura atual (FRB3). A adição deste procedimento busca proporcionar ao algoritmo uma melhoria significativa dos resultados, com um pequeno acréscimo do tempo computacional em relação ao método original.

A seguir são apresentados os passos do método FRB3+ENG:

Passo 1 – Calcule  $TP_j = \sum_{i=1}^m p_{ij}$ ,  $\forall j \in n$ , e coloque em ordem não-crescente de  $TP_j$ , e

$$\pi \leftarrow \emptyset;$$

Passo 2 – Para  $t = 1$  até  $n$ , testar a tarefa na posição  $t$  de  $TP_j$ , em todas as possíveis posições de  $\pi$  e inserir a tarefa na posição com menor *makespan*;

Passo 2.1 – Se  $t \geq 3$ , e  $t$  a posição de uma tarefa inserida na seqüência parcial  $\pi$ , então aplique o procedimento ENG;

Passo 2.2 – Para  $i = 1$  até  $t$ , retire a tarefa  $h$  da posição de  $\pi$  e teste  $h$  em todas as possíveis posições de  $\pi$ , inserir a tarefa na posição com menor *makespan*;

Para explicar de forma simples a aplicação do método ENG, segue exemplo:

Tabela 3.1 – Dados para exemplo de aplicação do método FRG3+ENG

	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$
$M_1$	8	4	9	2	4
$M_2$	3	2	3	1	3
$M_3$	4	8	4	7	10
$M_4$	5	4	2	7	3

Resolvendo pelo método FRB3+ENG, resulta em:

Passo 1 – Soma dos tempos de processamento e ordenação seguindo a ordem não-crescente, que resulta na sequência:  $J_1 J_5 J_2 J_3 J_4$ .

Passo 2 – Então a primeira tarefa é tomada como solução logo:  $\pi = (J_1)$ .

Inserindo a próxima tarefa  $J_5$  em todas as possíveis posições:

Resulta nas seqüências:

$J_1 J_5$ ,  $C_{\max} = 28$ ;

$J_5 J_1$ ,  $C_{\max} = 26$ ;

Portando  $\pi = (J_5, J_1)$ .

$t = 2$ , não aplica o ENG.

No passo 2.2 o algoritmo permanece inalterado.

Próxima iteração:

Inserindo a próxima tarefa  $J_2$  em todas as possíveis posições:

Resultando nas seqüências:

$$J_2 J_5 J_1, C_{\max} = 33;$$

$$J_5 J_2 J_1, C_{\max} = 34;$$

$$J_5 J_1 J_2, C_{\max} = 33;$$

Portanto  $\pi = (J_2 J_5 J_1)$ .

$t = 3$ , primeira aplicação do procedimento ENG.

Como  $J_2$  foi inserida na posição 1, o algoritmo ENG irá testar a sequência:

$$J_2 J_1 J_5, C_{\max} = 32.$$

Portanto  $\pi = (J_2 J_1 J_5)$ .

No passo 2.2, a sequência permanece inalterada;

Próxima iteração:

Inserindo a tarefa  $J_3$ , testando todas as possíveis posições:

$$J_3 J_2 J_1 J_5, C_{\max} = 41;$$

$$J_2 J_3 J_1 J_5, C_{\max} = 41;$$

$$J_2 J_1 J_3 J_5, C_{\max} = 41;$$

$$J_2 J_1 J_5 J_3, C_{\max} = 35;$$

Portanto  $\pi = (J_2 J_1 J_5 J_3)$ .

Passo 2.1, ENG, como  $J_3$  foi inserida na posição 4, sequência testada será:

$$J_2 J_5 J_1 J_3, C_{\max} = 35, \text{ não é menor, solução permanece inalterada.}$$

No passo 2.2, a solução permanece inalterada.

Próxima iteração:

Inserindo a tarefa  $J_4$  e testando em todas as possíveis posições:

$$J_4 J_2 J_1 J_5 J_3, C_{\max} = 38;$$

$$J_2 J_4 J_1 J_5 J_3, C_{\max} = 41;$$

$$J_2 J_1 J_4 J_5 J_3, C_{\max} = 42;$$

$$J_2 J_1 J_5 J_4 J_3, C_{\max} = 45;$$

$$J_2 J_1 J_5 J_3 J_4, C_{\max} = 47;$$

Passo 2.1, como  $J_4$  foi inserida na posição 1, sequência testada será:

$$J_4 J_1 J_2 J_5 J_3, C_{\max} = 41, \text{ não é menor, solução permanece inalterada.}$$

No passo 2.2 o algoritmo permanece inalterado.

Portanto o resultado final é:  $J_4 J_2 J_1 J_5 J_3$  com *makespan* 38, o mesmo problema aplicando o algoritmo FRB3 resultou na sequência  $J_4 J_2 J_5 J_1 J_3$  com o *makespan* de 39. Os resultados podem observados nas figuras a seguir:

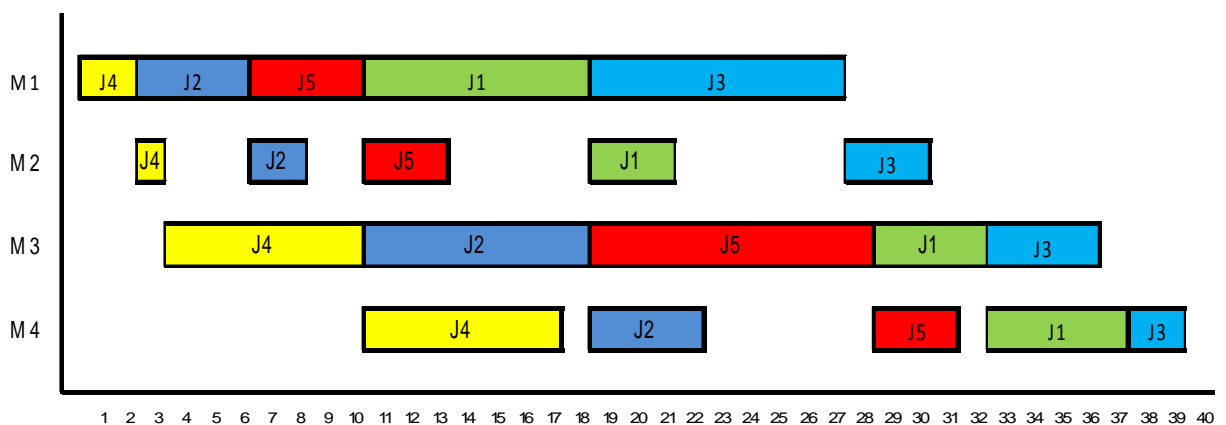


Figura 3.1 – Resultado da aplicação do FRB3,  $C_{\max} = 39$

Comparando as figuras 3.1 e 3.2, verifica-se que a troca de posição das tarefas  $J_5$  e  $J_1$  gerada pelo procedimento ENG no FRB3 (FRB3+ENG) resultou em uma redução do tempo total da programação.

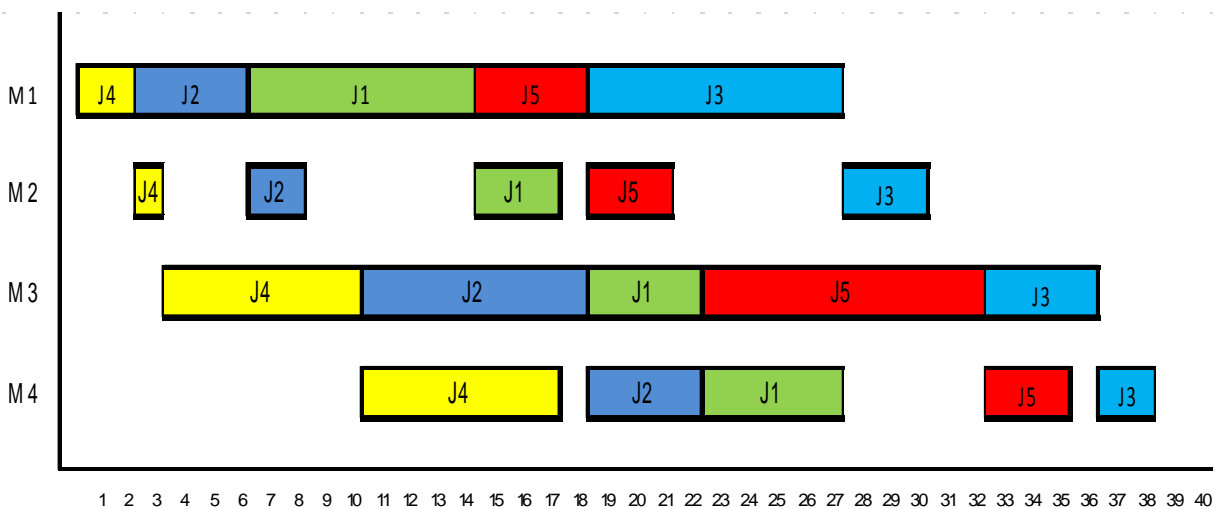


Figura 3.2 – Sequência resultante do FRB3+ENG, *makespan* 38

A inserção do método ENG no algoritmo FRB3 não apresenta aumento significativo de esforço computacional, deve-se atentar ao fato que é adicionada apenas uma troca de posição entre duas tarefas ao final de cada iteração do passo 2.

Para o desenvolvimento do método algumas alternativas diferentes foram testadas sem sucesso em relação aos principais métodos heurísticos. Dentre as alternativas, uma ordenação não crescente em relação à média dos tempos de processamento, combinação dos métodos NEHKK + FRB3 ou FRB4 e ainda a ordenação inicial do NEH-D com o critério de desempate do NEHKK.

O sucesso obtido com o método FRB3 + ENG deve-se ao fato de que o impacto no *makespan* de uma determinada sequência é maior se houver uma perturbação nas extremidades, primeira ou última posição. Devido ao fato de que a base do cálculo do *makespan* é a primeira ou a última tarefa.

A idéia principal para o desenvolvimento do método foi gerar uma perturbação, na duração total da sequência em construção, que seja reduzido o *gap* entre as tarefas. O tempo de espera entre as máquinas é modificado a cada inserção de tarefa na solução parcial, favorecendo assim que o procedimento ENG encontre melhores soluções para qualquer tamanho de problema.

No capítulo seguinte serão apresentados os resultados da experimentação computacional resultante da incorporação do procedimento ENG ao melhor método heurístico presente na literatura atual. Os resultados que serão apresentados mostrarão que o método aqui proposto melhora significativamente a qualidade das soluções apresentadas pelo método original e com um aumento não significativo do tempo computacional.

# CAPÍTULO 4

---

## 4 EXPERIMENTAÇÃO COMPUTACIONAL

Neste capítulo são apresentados os resultados de uma experimentação computacional realizada com o objetivo de determinar qual o melhor método heurístico construtivo para problemas de seqüenciamento de tarefas em ambiente *flow shop* permutacional. O experimento foi realizado em um *workstation* com 2 processadores modelo Intel Xeon X5450 3.00 GHz e 32 GB de memória RAM. Os programas foram desenvolvidos em linguagem C usando o compilador Dev-C++ versão 4.9.9.2.

### 4.1 DETERMINAÇÃO DAS AMOSTRAS DO EXPERIMENTO

Na experimentação computacional foram utilizadas duas bases de dados aplicadas para os problemas de *flow shop*, a base de Taillard com 120 problemas divididos em 12 classes com 10 problemas de cada classe variando máquinas entre 5, 10, 20 e as tarefas entre 20, 50, 100, 200, 500, divididas nas seguintes classes (tarefas, máquinas): (20, 5), (20, 10), (20, 20), (50, 5), (50, 10), (50, 20), (100, 5), (100, 10), (100,20), (200, 10), (200,20), (500,20). Para uma experimentação mais abrangente foi utilizado uma base de dados proposta por Ruben Ruiz disponível no endereço eletrônico: <http://soa.iti.es/problem-instances>. Os problemas utilizados neste trabalho são encontrados no link para download nomeado “*Instances for no-idle permutation flowshops*”. Esta base de dados é composta de 250 problemas variando máquinas entre 10, 20, 30, 40, 50 e as tarefas entre 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 com 5 problemas de cada classe e gerando 50 classes na aplicação de todas as possíveis

combinações entre as máquinas e tarefas disponíveis: (50, 10), (50, 20),..., (500, 40), (500, 50).

## 4.2 MÉTODO DE ANÁLISE DOS RESULTADOS

Os resultados obtidos na experimentação computacional foram submetidos à análise avaliando a porcentagem de sucesso, desvio relativo médio e o tempo médio de computação dos métodos apresentados.

Para o cálculo da porcentagem de sucesso é considerado sucesso o método que obtiver o menor *makespan* para o problema em questão. Se houver empate entre dois ou mais algoritmos, ambos são considerados sucesso, sendo a porcentagem de sucesso calculada dividindo a quantidade total de sucesso pelo número total de problemas resolvidos por um determinado método.

O desvio relativo mede a distância da solução obtida por um determinado método em relação a melhor solução obtida para um determinado problema. O método que apresentar o menor desvio relativo médio é o método que apresenta a melhor qualidade das soluções. O desvio relativo de um método para um determinado problema pode ser calculado da seguinte forma:

$$DR(x) = \frac{C_{\max}(x) - C_{\max}^*}{C_{\max}^*}$$

onde  $C_{\max}(x)$  é o *makespan* obtido por um determinado método para um determinado problema e  $C_{\max}^*$  é o menor *makespan* conhecido para o mesmo problema em avaliação. E o



desvio relativo médio para um determinado método é a soma dos desvios relativos  $DR(x)$  dividido pelo número de problemas avaliados.

O tempo médio de computação é obtido realizando a soma do tempo de computação de todos os problemas resolvidos por um determinado método, dividido pelo total de problemas. A utilização deste critério de comparação fornece informações sobre qual dos métodos é mais rápido na busca por soluções, o método que apresentar a menor média é considerado o mais rápido, apresentando maior eficiência computacional.

### **4.3 RESULTADOS DA EXPERIMENTAÇÃO COMPUTACIONAL**

Todos os métodos foram avaliados utilizando as bases de dados de Taillard e Ruben Ruiz, apresentadas na seção 4.1., e analisados segundo os critérios apresentados na seção 4.2.

#### **4.3.1 COMPARAÇÃO DOS MÉTODOS DA LITERATURA**

Os resultados da experimentação são expostos nas tabelas a seguir, primeiramente os resultados da porcentagem de sucesso dos métodos.

As tabelas 4.1, 4.2, 4.3 e 4.4 mostram superioridade do método FRB3 em relação aos demais métodos avaliados, a diferença da porcentagem de sucesso do FRB3 comparado aos resultados apresentados pelos métodos NEHKK e NEH-D é muito significativa. Nas tabelas 4.1 e 4.2 são apresentados os resultados dos métodos quando submetidos à avaliação para os problemas de Taillard. O FRB3 apresenta 73,33% e 71% de sucesso em relação aos demais métodos, com máquinas não agrupadas e agrupadas respectivamente. As tabelas 4.3 e 4.4 os resultados referentes aos problemas de Ruben Ruiz, para estes problemas o FRB3 apresentou

94% de sucesso sobre os demais métodos com classes agrupadas em relação ao número de máquinas.

Tabela 4.1. – Porcentagem de sucesso, para as classes de problemas de Taillard.

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>20</b>	<b>5</b>	20,00	20,00	70,00	70,00
<b>20</b>	<b>10</b>	20,00	30,00	70,00	50,00
<b>20</b>	<b>20</b>	10,00	10,00	80,00	50,00
<b>50</b>	<b>5</b>	0,00	10,00	70,00	70,00
<b>50</b>	<b>10</b>	0,00	0,00	50,00	60,00
<b>50</b>	<b>20</b>	0,00	0,00	60,00	50,00
<b>100</b>	<b>5</b>	30,00	20,00	100,00	60,00
<b>100</b>	<b>10</b>	0,00	0,00	90,00	20,00
<b>100</b>	<b>20</b>	0,00	0,00	70,00	30,00
<b>200</b>	<b>10</b>	10,00	0,00	80,00	40,00
<b>200</b>	<b>20</b>	0,00	0,00	90,00	10,00
<b>500</b>	<b>20</b>	20,00	40,00	50,00	0,00
<b>Todos</b>		9,17	10,83	73,33	45,50

Tabela 4.2. – Porcentagem de sucesso com classes agrupadas para os problemas de Taillard

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>20</b>	16,67	20,00	73,33	56,67
<b>50</b>	0,00	3,33	60,00	60,00
<b>100</b>	10,00	6,67	86,67	36,67
<b>200</b>	5,00	0,00	85,00	25,00
<b>500</b>	20,00	40,00	50,00	0,00
<b>Todos</b>	10,33	14,00	<b>71,00</b>	35,67

Tabela 4.3 – Porcentagem de sucesso para as classes de problemas de Ruben Ruiz

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>50</b>	<b>10</b>	0,00	0,00	80,00	60,00
<b>50</b>	<b>20</b>	0,00	0,00	80,00	20,00
<b>50</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>50</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>50</b>	<b>50</b>	0,00	0,00	60,00	4,00

continua

continuação

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>100</b>	<b>10</b>	0,00	0,00	40,00	60,00
<b>100</b>	<b>20</b>	0,00	0,00	80,00	20,00
<b>100</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>100</b>	<b>40</b>	0,00	0,00	80,00	20,00
<b>100</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>150</b>	<b>10</b>	0,00	0,00	80,00	20,00
<b>150</b>	<b>20</b>	0,00	0,00	100,00	0,00
<b>150</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>150</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>150</b>	<b>50</b>	0,00	0,00	60,00	40,00
<b>200</b>	<b>10</b>	0,00	20,00	100,00	60,00
<b>200</b>	<b>20</b>	0,00	0,00	80,00	20,00
<b>200</b>	<b>30</b>	0,00	0,00	80,00	20,00
<b>200</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>200</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>250</b>	<b>10</b>	0,00	0,00	100,00	20,00
<b>250</b>	<b>20</b>	0,00	0,00	100,00	0,00
<b>250</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>250</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>250</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>300</b>	<b>10</b>	0,00	20,00	100,00	40,00
<b>300</b>	<b>20</b>	0,00	0,00	100,00	0,00
<b>300</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>300</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>300</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>350</b>	<b>10</b>	0,00	0,00	100,00	20,00
<b>350</b>	<b>20</b>	0,00	0,00	100,00	0,00
<b>350</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>350</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>350</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>400</b>	<b>10</b>	0,00	40,00	100,00	60,00
<b>400</b>	<b>20</b>	0,00	0,00	100,00	0,00
<b>400</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>400</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>400</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>450</b>	<b>10</b>	0,00	20,00	80,00	40,00
<b>450</b>	<b>20</b>	0,00	0,00	100,00	0,00
<b>450</b>	<b>30</b>	0,00	0,00	100,00	0,00

continua

conclusão

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>450</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>450</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>500</b>	<b>10</b>	0,00	0,00	100,00	20,00
<b>500</b>	<b>20</b>	0,00	0,00	100,00	0,00
<b>500</b>	<b>30</b>	0,00	0,00	100,00	0,00
<b>500</b>	<b>40</b>	0,00	0,00	100,00	0,00
<b>500</b>	<b>50</b>	0,00	0,00	100,00	0,00
<b>Média</b>		0,00	2,00	94,00	11,60

Tabela 4.4 – Porcentagem de sucesso com classes agrupadas para os problemas de Ruben Ruiz.

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>50</b>	0,00	0,00	84,00	24,00
<b>100</b>	0,00	0,00	80,00	20,00
<b>150</b>	0,00	0,00	88,00	12,00
<b>200</b>	0,00	4,00	92,00	20,00
<b>250</b>	0,00	0,00	100,00	4,00
<b>300</b>	0,00	4,00	100,00	80,00
<b>350</b>	0,00	0,00	100,00	4,00
<b>400</b>	0,00	8,00	80,00	40,00
<b>450</b>	0,00	4,00	100,00	20,00
<b>500</b>	0,00	0,00	100,00	4,00
<b>Média</b>	0,00	2,00	94,00	11,60

As tabelas 4.5, 4.6, 4.7 e 4.8 apresentam os resultados referentes ao desvio relativo médio encontrado para os problemas avaliados. As tabelas 4.5 e 4.6 mostram os resultados para os problemas de Taillard e as tabelas 4.7 e 4.8, os resultados para os problemas de Ruben Ruiz. Para os problemas de Taillard o FRB3 apresenta o menor desvio relativo médio de 0,05612 e 0,04987, para máquinas não agrupadas e agrupadas respectivamente. Para os problemas propostos por Ruben Ruiz, o método FRB3 resultou em menor desvio relativo médio de 0,01430 e 0,01430, para máquinas não agrupadas e agrupadas respectivamente.

Portanto, em ambos os casos o algoritmo FRB3 apresentou menor desvio relativo médio em comparação aos métodos NEHKK, NEH-D e FRB4.

Tabela 4.5 – Desvio relativo médio da melhor solução em relação ao número de máquinas e tarefas no problema

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
20	5	0,15364	0,15733	0,13963	0,13982
20	10	0,12877	0,11613	0,10350	0,10675
20	20	0,08510	0,08543	0,07287	0,07427
50	5	0,07434	0,07508	0,06989	0,07001
50	10	0,06832	0,06494	0,05034	0,04536
50	20	0,05842	0,05236	0,03554	0,03769
100	5	0,04856	0,04804	0,04561	0,04635
100	10	0,07782	0,07508	0,06724	0,06974
100	20	0,04133	0,03941	0,02231	0,02570
200	10	0,03939	0,03744	0,03253	0,03384
200	20	0,02485	0,02049	0,00636	0,00962
500	20	0,03414	0,03069	0,02761	0,03000
<b>Média</b>		0,06956	0,06687	0,05612	0,05741

Tabela 4.6 – Desvio relativo médio em relação ao número de tarefas do problema

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
20	0,1225	0,1196	0,1053	0,1069
50	0,06702	0,6413	0,05192	0,05102
100	0,05590	0,05418	0,04505	0,04726
200	0,03212	0,02896	0,01945	0,02173
500	0,03414	0,03069	0,02761	0,03000
<b>Média</b>	0,06234	0,05952	0,04987	0,05138

Tabela 4.7 – Desvio Relativo médio para as classes de problemas de Ruben Ruiz

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
50	10	0,09488	0,07874	0,06066	0,06266
50	20	0,05190	0,03439	0,00662	0,01253
50	30	0,05271	0,03643	0,01488	0,02234
50	40	0,03987	0,03357	0,01359	0,01892
50	50	0,06248	0,05304	0,03474	0,03728
100	10	0,07148	0,04251	0,02828	0,02751
100	20	0,06610	0,03778	0,01579	0,02208
100	30	0,06147	0,03973	0,01654	0,02295

continua

Conclusão					
<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
100	40	0,04321	0,02989	0,00740	0,01192
100	50	0,03867	0,02412	0,00466	0,01096
150	10	0,05889	0,02573	0,01896	0,02020
150	20	0,07033	0,04216	0,02052	0,02524
150	30	0,05304	0,03391	0,00972	0,01443
150	40	0,05084	0,03391	0,01072	0,01975
150	50	0,03480	0,02532	0,00483	0,00776
200	10	0,04745	0,02771	0,02499	0,02579
200	20	0,05980	0,03150	0,01556	0,01961
200	30	0,06096	0,03344	0,01172	0,01458
200	40	0,05719	0,03337	0,00997	0,02090
200	50	0,04080	0,02641	0,00237	0,01040
250	10	0,04781	0,03403	0,01331	0,01999
250	20	0,06998	0,03403	0,01331	0,01999
250	30	0,05887	0,03353	0,00962	0,01681
250	40	0,05564	0,03022	0,01128	0,01609
250	50	0,05317	0,03265	0,00907	0,01730
300	10	0,03050	0,01368	0,01116	0,01126
300	20	0,06302	0,02397	0,00790	0,01372
300	30	0,05913	0,03190	0,00983	0,01762
300	40	0,05616	0,02867	0,00635	0,01417
300	50	0,04958	0,02752	0,00478	0,01261
350	10	0,06703	0,03925	0,03855	0,03982
350	20	0,06660	0,02919	0,01752	0,02129
350	30	0,05817	0,03127	0,00909	0,01885
350	40	0,06229	0,03441	0,01058	0,01929
350	50	0,05100	0,02475	0,00461	0,01120
400	10	0,05023	0,01932	0,01801	0,01826
400	20	0,06161	0,02473	0,01360	0,01750
400	30	0,05990	0,02325	0,00502	0,01284
400	40	0,06240	0,03424	0,00885	0,01733
400	50	0,05693	0,02973	0,00500	0,01461
450	10	0,06679	0,03253	0,03045	0,03060
450	20	0,06489	0,02071	0,00810	0,01233
450	30	0,06847	0,03008	0,01233	0,01672
450	40	0,05879	0,03121	0,00894	0,01886
450	50	0,05608	0,03276	0,01158	0,01828
500	10	0,05546	0,03219	0,03156	0,03173
500	20	0,07870	0,03564	0,02252	0,02570
500	30	0,06132	0,02484	0,00839	0,01512
500	40	0,06063	0,03058	0,01156	0,02006
500	50	0,06148	0,03211	0,00999	0,01817
<b>Média</b>		0,05779	0,03179	0,01430	0,01950

Tabela 4.8 – Desvio Relativo dos Métodos sobre o banco de dados de Ruben Ruiz

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>50</b>	0,060368	0,047234	0,026099	0,030744
<b>100</b>	0,056188	0,034807	0,014535	0,019085
<b>150</b>	0,053578	0,032206	0,012949	0,017474
<b>200</b>	0,05324	0,030486	0,012923	0,018257
<b>250</b>	0,057094	0,029466	0,011288	0,016776
<b>300</b>	0,051678	0,025147	0,008004	0,013876
<b>350</b>	0,06102	0,031775	0,016068	0,022091
<b>400</b>	0,058213	0,026525	0,010094	0,016108
<b>450</b>	0,063002	0,02946	0,014282	0,01936
<b>500</b>	0,063518	0,031074	0,016803	0,022156
<b>Média</b>	0,05779	0,031791	0,014305	0,019593

Para verificar a eficiência dos métodos foi analisado o tempo médio de computação dos algoritmos, os resultados são apresentados nas tabelas 4.9, 4.10, 4.11 e 4.12:

Tabela 4.9 – Tempo médio de computação para os problemas de Taillard

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>20</b>	<b>5</b>	0,00060	0,00070	0,00320	0,00250
<b>20</b>	<b>10</b>	0,00110	0,00130	0,00710	0,00580
<b>20</b>	<b>20</b>	0,00200	0,00110	0,01340	0,01160
<b>50</b>	<b>5</b>	0,00880	0,01190	0,11250	0,04330
<b>50</b>	<b>10</b>	0,01690	0,01600	0,23390	0,10410
<b>50</b>	<b>20</b>	0,03250	0,02450	0,04870	0,22900
<b>100</b>	<b>5</b>	0,07470	0,09440	1,63449	0,36020
<b>100</b>	<b>10</b>	0,15040	0,12740	3,35950	0,83200
<b>100</b>	<b>20</b>	0,29880	0,19270	7,54820	2,11030
<b>200</b>	<b>10</b>	1,15600	0,98640	49,0508	6,20460
<b>200</b>	<b>20</b>	2,27450	1,52850	112,26330	16,88730
<b>500</b>	<b>20</b>	33,92190	22,59820	4220,7768	260,79230
<b>Média</b>		3,16152	2,13193	366,29090	23,96525

Tabela 4.10 – Tempo médio de computação com classes agrupadas para os problemas de Taillard

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>20</b>	0,00123	0,00103	0,00790	0,00663
<b>50</b>	0,01940	0,01747	0,27787	0,12547
<b>100</b>	0,17463	0,13817	4,18087	1,10083
<b>200</b>	1,71525	1,25745	80,65705	11,54595
<b>500</b>	33,92190	22,59820	4220,77680	260,79230
<b>Média</b>	7,16648	4,80246	861,18010	54,71424

Tabela 4.11 – Tempo médio de computação para os problemas de Ruben Ruiz

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
<b>50</b>	<b>10</b>	0,01620	0,00620	0,23280	0,23240
<b>50</b>	<b>20</b>	0,03320	0,01340	0,48920	0,24120
<b>50</b>	<b>30</b>	0,04840	0,02000	0,72280	0,36400
<b>50</b>	<b>40</b>	0,06340	0,02600	0,96320	0,51380
<b>50</b>	<b>50</b>	0,07760	0,03200	1,19120	0,62740
<b>100</b>	<b>10</b>	0,12320	0,04940	3,37420	0,83920
<b>100</b>	<b>20</b>	0,27260	0,10960	7,65400	2,15580
<b>100</b>	<b>30</b>	0,39900	0,15920	11,35120	3,32980
<b>100</b>	<b>40</b>	0,52180	0,21020	15,05800	4,43960
<b>100</b>	<b>50</b>	0,64840	0,26380	18,84680	5,53480
<b>150</b>	<b>10</b>	0,40940	0,16000	16,29640	2,74320
<b>150</b>	<b>20</b>	0,91900	0,35680	36,81020	7,18140
<b>150</b>	<b>30</b>	1,35980	0,52860	56,32200	11,46720
<b>150</b>	<b>40</b>	1,77240	0,69280	74,50040	15,41440
<b>150</b>	<b>50</b>	2,21160	0,86180	93,32860	19,50320
<b>200</b>	<b>10</b>	0,95860	0,36180	49,31860	6,11060
<b>200</b>	<b>20</b>	2,16960	0,82140	113,39660	16,85280
<b>200</b>	<b>30</b>	3,31220	1,26480	177,59940	28,07100
<b>200</b>	<b>40</b>	4,30560	1,64020	234,64480	37,35360
<b>200</b>	<b>50</b>	5,30160	2,03520	293,23900	46,50080
<b>250</b>	<b>10</b>	1,82120	0,68840	117,47980	12,73020
<b>250</b>	<b>20</b>	4,17120	1,57000	269,07700	33,74200
<b>250</b>	<b>30</b>	6,46440	2,44360	423,81800	53,72040
<b>250</b>	<b>40</b>	8,46520	3,20420	566,36960	73,38860
<b>250</b>	<b>50</b>	10,36580	3,96920	706,93600	93,30020
<b>300</b>	<b>10</b>	3,14140	1,17420	240,68020	21,23040
<b>300</b>	<b>20</b>	7,08400	2,63700	547,41520	56,63940
<b>300</b>	<b>30</b>	11,04000	4,19580	864,39060	92,62640
<b>300</b>	<b>40</b>	14,83900	5,60800	1181,37520	129,81300

continua



conclusão

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
300	50	18,08480	6,86840	1461,23920	162,41960
350	10	4,87000	1,83660	435,36260	33,08860
350	20	11,11800	4,15980	1001,40460	86,87120
350	30	17,23000	6,54560	1579,34620	147,31280
350	40	23,52680	8,81420	2160,60000	205,95200
350	50	28,90340	10,91340	2697,59500	257,55220
400	10	7,30220	2,74220	746,44160	48,01900
400	20	16,41380	6,11080	1676,57900	128,53840
400	30	25,60440	9,49420	2647,84140	216,02600
400	40	34,84640	13,00880	3643,29220	302,29280
400	50	43,70460	16,43480	4619,14100	388,56720
450	10	10,66340	3,95300	1224,58340	73,05680
450	20	23,44480	8,68420	2694,32700	190,45200
450	30	36,55380	13,50020	4232,71680	311,27060
450	40	49,48840	18,52900	5823,44220	436,74940
450	50	62,23320	23,36340	7381,84360	553,59560
500	10	16,22360	5,92920	2027,79920	112,18760
500	20	33,72260	12,35020	4246,34380	267,56320
500	30	51,13340	18,90820	6537,19780	435,61600
500	40	68,85000	25,57820	8923,62000	598,21720
500	50	86,42020	32,28540	11276,73680	766,09980
<b>Média</b>		16,50	6,17	1810,84	141,01

Tabela 4.12 – Tempo de médio de computação para os problemas de Ruben Ruiz com máquinas agrupadas

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB3</b>	<b>FRB4</b>
50	0,05	0,02	0,72	0,37
100	0,39	0,16	11,26	3,26
150	1,33	0,52	55,45	11,26
200	3,21	1,22	173,64	26,98
250	6,26	2,38	416,74	53,38
300	10,84	4,10	859,02	92,55
350	17,13	6,45	1574,86	146,16
400	25,57	9,56	2666,66	216,69
450	36,48	13,61	4271,38	312,82
500	63,72	23,68	8078,71	546,66
<b>Média</b>	16,50	6,17	1810,84	141,01

Pelo que se pode verificar nas tabelas 4.9, 4.10, 4.11 e 4.12, o tempo de computação do método FRB3 é muito superior quando comparado aos demais métodos. É possível notar também que os tempos apresentados nesta dissertação são superiores aos encontrados no artigo original, deve-se atribuir essa diferença dos tempos a alguns fatores como de funções utilizadas na programação. Entretanto verifica-se a manutenção da proporção relativa entre os tempos de execução dos métodos.

#### 4.3.2 EXPERIMENTAÇÃO COM O NOVO MÉTODO HEURÍSTICO PROPOSTO

Os resultados expostos na subseção anterior apresentaram que o método FRB3 obtém melhores soluções em comparação aos métodos apresentados na literatura para o problema flow shop permutacional com critério de minimização do *makespan*. Com isso o objetivo da dissertação é aplicar a um determinado procedimento heurístico de melhoramento conforme apresentado no capítulo anterior, realizando-se uma integração dos métodos FRB3 e ENG, criando o algoritmo chamado FRB3+ENG que possibilite obter soluções de melhor qualidade com diferenças significativas comparadas aos demais métodos avaliados.

Tabela 4.13– Porcentagem de sucesso para as classes de problema de Taillard com o método FRB3+ENG

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>20</b>	<b>5</b>	0,00	0,00	10,00	10,00	90,00
<b>20</b>	<b>10</b>	10,00	10,00	50,00	70,00	30,00
<b>20</b>	<b>20</b>	0,00	10,00	30,00	40,00	80,00
<b>50</b>	<b>5</b>	0,00	10,00	10,00	10,00	80,00
<b>50</b>	<b>10</b>	0,00	0,00	50,00	30,00	40,00
<b>50</b>	<b>20</b>	0,00	0,00	20,00	50,00	60,00
<b>100</b>	<b>5</b>	0,00	0,00	10,00	20,00	80,00
<b>100</b>	<b>10</b>	0,00	0,00	10,00	30,00	70,00
<b>100</b>	<b>20</b>	10,00	0,00	0,00	30,00	80,00

continua

conclusão

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>200</b>	<b>10</b>	0,00	0,00	10,00	20,00	70,00
<b>200</b>	<b>20</b>	0,00	0,00	10,00	50,00	40,00
<b>500</b>	<b>20</b>	10,00	40,00	0,00	20,00	30,00
<b>Média</b>		2,50	5,83	17,50	31,67	62,50

Tabela 4.14 – Porcentagem de sucesso com classes agrupadas para os problemas de Taillard com o método FRB3+GN

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>20</b>	3,33	6,67	30,00	40,00	66,67
<b>50</b>	0,00	3,33	26,67	30,00	60,00
<b>100</b>	3,33	0,00	6,67	26,67	76,67
<b>200</b>	0,00	0,00	10,00	35,00	55,00
<b>500</b>	10,00	40,00	0,00	20,00	30,00
<b>Média</b>	3,33	10,00	14,67	30,33	57,67

Tabela 4.15 – Porcentagem de sucesso para os problemas de Ruben Ruiz com o método FRB3+ENG

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>50</b>	<b>10</b>	0,00	0,00	0,00	60,00	40,00
<b>50</b>	<b>20</b>	0,00	0,00	20,00	40,00	40,00
<b>50</b>	<b>30</b>	0,00	0,00	0,00	60,00	80,00
<b>50</b>	<b>40</b>	0,00	0,00	0,00	100,00	60,00
<b>50</b>	<b>50</b>	0,00	0,00	0,00	60,00	80,00
<b>100</b>	<b>10</b>	0,00	0,00	0,00	20,00	80,00
<b>100</b>	<b>20</b>	0,00	0,00	0,00	60,00	60,00
<b>100</b>	<b>30</b>	0,00	0,00	0,00	80,00	80,00
<b>100</b>	<b>40</b>	0,00	0,00	0,00	40,00	80,00
<b>100</b>	<b>50</b>	0,00	0,00	0,00	20,00	80,00
<b>150</b>	<b>10</b>	0,00	20,00	0,00	20,00	60,00
<b>150</b>	<b>20</b>	0,00	0,00	0,00	60,00	40,00
<b>150</b>	<b>30</b>	0,00	0,00	0,00	20,00	100,00
<b>150</b>	<b>40</b>	0,00	0,00	0,00	0,00	100,00
<b>150</b>	<b>50</b>	0,00	0,00	20,00	60,00	60,00
<b>200</b>	<b>10</b>	0,00	0,00	0,00	20,00	80,00
<b>200</b>	<b>20</b>	0,00	0,00	20,00	40,00	60,00
<b>200</b>	<b>30</b>	0,00	0,00	0,00	60,00	40,00

continua

conclusão							
Tarefas	Máquinas	NEHKK	NEH-D	FRB4	FRB3	FRB3+ENG	
200	40	0,00	0,00	0,00	40,00	60,00	
200	50	0,00	0,00	0,00	20,00	100,00	
250	10	0,00	0,00	0,00	40,00	60,00	
250	20	0,00	0,00	0,00	40,00	60,00	
250	30	0,00	0,00	0,00	60,00	40,00	
250	40	0,00	0,00	0,00	20,00	80,00	
250	50	0,00	0,00	0,00	60,00	40,00	
300	10	0,00	20,00	20,00	20,00	80,00	
300	20	0,00	0,00	0,00	20,00	80,00	
300	30	0,00	0,00	0,00	40,00	60,00	
300	40	0,00	0,00	0,00	60,00	40,00	
300	50	0,00	0,00	0,00	20,00	80,00	
350	10	0,00	0,00	20,00	40,00	60,00	
350	20	0,00	0,00	0,00	60,00	40,00	
350	30	0,00	0,00	0,00	60,00	40,00	
350	40	0,00	0,00	0,00	20,00	80,00	
350	50	0,00	0,00	0,00	40,00	60,00	
400	10	0,00	0,00	0,00	20,00	80,00	
400	20	0,00	0,00	0,00	40,00	80,00	
400	30	0,00	0,00	0,00	40,00	60,00	
400	40	0,00	0,00	0,00	40,00	60,00	
400	50	0,00	0,00	0,00	40,00	60,00	
450	10	0,00	0,00	0,00	40,00	80,00	
450	20	0,00	0,00	0,00	80,00	40,00	
450	30	0,00	0,00	0,00	60,00	40,00	
450	40	0,00	0,00	0,00	80,00	20,00	
450	50	0,00	0,00	0,00	60,00	40,00	
500	10	0,00	0,00	0,00	20,00	80,00	
500	20	0,00	0,00	0,00	80,00	40,00	
500	30	0,00	0,00	0,00	60,00	40,00	
500	40	0,00	0,00	0,00	40,00	80,00	
500	50	0,00	0,00	0,00	60,00	60,00	
<b>Média</b>		0,00	0,80	2,00	44,80	62,80	

Tabela 4.16 – Porcentagem de sucesso com classes agrupadas para os problemas de Ruben Ruiz

Tarefas	NEHKK	NEH-D	FRB4	FRB3	FRB3+ENG
50	0,00	0,00	4,00	64,00	60,00
100	0,00	0,00	0,00	44,00	76,00
150	0,00	4,00	4,00	32,00	72,00

Continua

conclusão

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>200</b>	0,00	0,00	0,00	44,00	76,00
<b>250</b>	0,00	4,00	4,00	32,00	72,00
<b>300</b>	0,00	0,00	4,00	36,00	68,00
<b>350</b>	0,00	0,00	0,00	44,00	56,00
<b>400</b>	0,00	4,00	4,00	32,00	68,00
<b>450</b>	0,00	0,00	0,00	64,00	44,00
<b>500</b>	0,00	0,00	0,00	52,00	60,00
<b>Média</b>	0,00	0,80	2,00	44,80	62,80

As tabelas 4.13, 4.14, 4.15 e 4.16 mostram a porcentagem de sucesso dos métodos, para os problemas de Taillard. Nas tabelas 4.13 e 4.14, verifica-se uma diferença significativa para o algoritmo FRB3+ENG com 62,50% de sucesso para todos os problemas e 57,67% com máquinas agrupadas. Para os problemas de Ruben Ruiz, tabelas 4.15 e 4.16, o método que foi incorporado ao método FRB3 apresenta uma porcentagem de sucesso superior aos demais métodos com 62,80% de sucesso, para máquinas não agrupadas e agrupadas, contra 44,80% de sucesso do método FRB3. A seguir são apresentados os dados da comparação do desvio relativo em relação ao melhor resultado.

Tabela 4.17 – Desvio relativo médio de todas as classes com FRB3+ENG, Taillard

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+GN</b>
<b>20</b>	<b>5</b>	0,15364	0,15733	0,13982	0,13963	0,11235
<b>20</b>	<b>10</b>	0,128765	0,116131	0,106567	0,103498	0,102712
<b>20</b>	<b>20</b>	0,089182	0,089511	0,078316	0,076905	0,067592
<b>50</b>	<b>5</b>	0,084915	0,085664	0,080544	0,080425	0,065104
<b>50</b>	<b>10</b>	0,071605	0,068218	0,048580	0,053575	0,048478
<b>50</b>	<b>20</b>	0,058423	0,052361	0,037688	0,035542	0,034335
<b>100</b>	<b>5</b>	0,050858	0,050339	0,048643	0,047905	0,040982
<b>100</b>	<b>10</b>	0,084745	0,081985	0,076616	0,074102	0,071380
<b>100</b>	<b>20</b>	0,041330	0,039412	0,025704	0,022308	0,018346
<b>200</b>	<b>10</b>	0,046140	0,044173	0,040550	0,039233	0,036259
<b>200</b>	<b>20</b>	0,025478	0,021111	0,010237	0,006979	0,006481
<b>500</b>	<b>20</b>	0,03412	0,030691	0,030003	0,027612	0,027320
<b>Média</b>		0,072435	0,069744	0,058976	0,060272	0,052612

Tabela 4.18 – Desvio relativo médio com máquinas agrupadas para os problemas de Taillard, com o método FRB3+ENG

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>20</b>	0,123862	0,120990	0,108233	0,106678	0,094218
<b>50</b>	0,071648	0,068747	0,055604	0,056514	0,049306
<b>100</b>	0,058978	0,057245	0,050321	0,048105	0,043569
<b>200</b>	0,035809	0,032642	0,025393	0,023106	0,021370
<b>500</b>	0,034142	0,030691	0,030003	0,027612	0,027320
<b>Média</b>	0,064888	0,062063	0,053911	0,052403	0,047157

Tabela 4.19 – Desvio relativo médio para os problemas de Ruben Ruiz com o método FRB3+ENG

<b>Tarefas</b>	<b>Máquinas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>50</b>	<b>10</b>	0,094884	0,078738	0,062658	0,060664	0,061196
<b>50</b>	<b>20</b>	0,051899	0,034394	0,012526	0,006622	0,009292
<b>50</b>	<b>30</b>	0,052705	0,036431	0,022339	0,014878	0,014529
<b>50</b>	<b>40</b>	0,039872	0,033566	0,018916	0,013593	0,014046
<b>50</b>	<b>50</b>	0,062479	0,053039	0,037284	0,034737	0,032801
<b>100</b>	<b>10</b>	0,082344	0,053078	0,037928	0,038703	0,029525
<b>100</b>	<b>20</b>	0,066096	0,037782	0,022084	0,015792	0,017038
<b>100</b>	<b>30</b>	0,061473	0,039732	0,022946	0,016539	0,015690
<b>100</b>	<b>40</b>	0,044373	0,031032	0,013045	0,008524	0,008474
<b>100</b>	<b>50</b>	0,039488	0,024927	0,011757	0,005452	0,004801
<b>150</b>	<b>10</b>	0,064855	0,031513	0,025952	0,024699	0,022821
<b>150</b>	<b>20</b>	0,070328	0,042165	0,025235	0,020515	0,021741
<b>150</b>	<b>30</b>	0,056658	0,037456	0,017910	0,013185	0,009340
<b>150</b>	<b>40</b>	0,055248	0,038250	0,024032	0,014964	0,013061
<b>150</b>	<b>50</b>	0,034797	0,025316	0,007755	0,004834	0,004476
<b>200</b>	<b>10</b>	0,047453	0,027712	0,025794	0,024986	0,018481
<b>200</b>	<b>20</b>	0,061101	0,032765	0,020863	0,016802	0,014807
<b>200</b>	<b>30</b>	0,06096	0,033435	0,014576	0,011725	0,012015
<b>200</b>	<b>40</b>	0,057186	0,033371	0,020903	0,009974	0,010123
<b>200</b>	<b>50</b>	0,041232	0,026836	0,010818	0,002788	0,002053
<b>250</b>	<b>10</b>	0,052805	0,021761	0,018520	0,017990	0,016840
<b>250</b>	<b>20</b>	0,070604	0,034634	0,020585	0,013903	0,014353
<b>250</b>	<b>30</b>	0,058869	0,033526	0,016810	0,009621	0,011260
<b>250</b>	<b>40</b>	0,055637	0,030215	0,016086	0,011281	0,009060
<b>250</b>	<b>50</b>	0,053173	0,032648	0,017304	0,009068	0,010910
<b>300</b>	<b>10</b>	0,032390	0,015538	0,013113	0,013012	0,010877
<b>300</b>	<b>20</b>	0,064377	0,025276	0,015019	0,099184	0,007491
<b>300</b>	<b>30</b>	0,059127	0,031901	0,017625	0,009830	0,010344
<b>300</b>	<b>40</b>	0,056163	0,028670	0,014168	0,006350	0,007267
<b>300</b>	<b>50</b>	0,055392	0,033209	0,018214	0,010348	0,008779

continua

conclusão						
Tarefas	Máquinas	NEHKK	NEH-D	FRB4	FRB3	FRB3+ENG
350	10	0,067033	0,039252	0,039825	0,038545	0,037333
350	20	0,068356	0,030881	0,022973	0,019188	0,019347
350	30	0,058968	0,032048	0,019622	0,009852	0,009670
350	40	0,062294	0,034412	0,019288	0,010580	0,009252
350	50	0,051004	0,024751	0,011196	0,004612	0,003807
400	10	0,053075	0,022078	0,021016	0,020764	0,019209
400	20	0,061608	0,024729	0,017501	0,013597	0,013289
400	30	0,059897	0,023247	0,012844	0,005016	0,005193
400	40	0,062395	0,034237	0,017327	0,008847	0,008148
400	50	0,057016	0,029811	0,014691	0,005083	0,006038
450	10	0,068069	0,033776	0,031842	0,031691	0,029819
450	20	0,064892	0,020707	0,012333	0,008104	0,008205
450	30	0,070309	0,031861	0,018478	0,014073	0,014209
450	40	0,058789	0,031212	0,018862	0,008944	0,008967
450	50	0,058102	0,034743	0,020227	0,013521	0,013308
500	10	0,055458	0,032192	0,031726	0,031550	0,029154
500	20	0,078703	0,035644	0,025704	0,022518	0,023134
500	30	0,061316	0,024840	0,015117	0,008394	0,007789
500	40	0,060632	0,030582	0,020062	0,011556	0,011283
500	50	0,061483	0,032114	0,018169	0,009989	0,011395
<b>Média</b>		0,058867	0,032841	0,01534	0,020631	0,014653

Tabela 4.20 – Desvio relativo médio com máquinas agrupadas para os problemas de Ruben Ruiz com o método FB3+ENG

Tarefas	NEHKK	NEH-D	FRB4	FRB3	FRB3+ENG
50	0,060368	0,047234	0,030744	0,026099	0,026373
100	0,058755	0,03731	0,021552	0,017002	0,015106
150	0,056377	0,03494	0,020177	0,01564	0,014406
200	0,053586	0,030824	0,018591	0,013255	0,011496
250	0,058218	0,030557	0,017861	0,012373	0,012485
300	0,05349	0,026919	0,015628	0,009745	0,008852
350	0,061531	0,032269	0,022581	0,016555	0,015882
400	0,058798	0,026821	0,016676	0,010661	0,010375
450	0,064032	0,03046	0,020349	0,015267	0,014902
500	0,063518	0,031074	0,022156	0,016803	0,016551
<b>Média</b>	0,058867	0,032841	0,020631	0,01534	0,014653

Nas tabelas 4.17, 4.18, 4.19 e 4.20, são apresentados os resultados referentes ao critério de desvio relativo em relação ao melhor resultado obtido pelo método. As tabelas 4.12 e 4.13 apresentam os resultados dos métodos para a base de dados de Taillard, o qual se verifica que o menor desvio em ambas as tabelas é para o método FRB3+ENG. Analisando a tabela 4.14 também se verifica que o método FRB3+ENG apresenta menor desvio relativo em relação aos demais métodos estudados neste trabalho.

Tabela 4.21 – Tempo de execução para os problemas de Taillard com o método FRB3+ENG

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>20</b>	0,00123	0,00103	0,00663	0,00790	0,00807
<b>50</b>	0,01940	0,01747	0,12547	0,27787	0,27660
<b>100</b>	0,17463	0,13817	1,10083	4,18087	4,21953
<b>200</b>	1,71525	1,25745	11,54595	80,65705	80,99400
<b>500</b>	33,92190	22,59820	260,79230	4220,77680	4251,11580
<b>Média</b>	7,16648	4,80246	54,71424	861,18010	867,32280

Tabela 4.22 – Tempo de execução para os problemas de Ruben Ruiz com o método FRB3+ENG

<b>Tarefas</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>50</b>	0,05	0,02	0,37	0,72	0,72
<b>100</b>	0,39	0,16	3,26	11,26	11,19
<b>150</b>	1,33	0,52	11,26	55,45	55,40
<b>200</b>	3,21	1,22	26,98	173,64	173,19
<b>250</b>	6,26	2,38	53,38	416,74	418,23
<b>300</b>	10,84	4,10	92,55	859,02	860,51
<b>350</b>	17,13	6,45	146,16	1574,86	1577,74
<b>400</b>	25,57	9,56	216,69	2666,66	2670,70
<b>450</b>	36,48	13,61	312,82	4271,38	4267,86
<b>500</b>	63,72	23,68	546,66	8078,71	8078,47
<b>Média</b>	16,50	6,17	141,01	1810,84	1811,40

As tabelas 4.15 e 4.16 mostram que o método FRB3+ENG é o método de maior custo computacional, entretanto comparando as soluções obtidas não existe diferença significativa



em comparação conjunta ao melhor método FRB3, o acréscimo de tempo é praticamente nulo e pode ser desprezado, pois é menor que 1% em média. Desta forma o método FRB3+ENG apresenta a mesma eficiência computacional que o método FRB3, mas obtém soluções de melhor qualidade.

# CAPÍTULO 5

---

## 5. CONCLUSÕES

O desenvolvimento de métodos heurísticos construtivos para a programação de operações em ambiente *flow shop* permutacional motivou o desenvolvimento deste projeto. A grande quantidade de publicações de métodos heurísticos cada vez mais eficientes na busca por soluções melhores leva a motivação de pesquisadores na busca de resultados que atinjam as soluções ótimas ou semi ótimas em tempo computacional inferior em comparação aos métodos de solução exata.

O procedimento proposto neste projeto, método ENG, usado em conjunto com o método heurístico construtivo FRB3, apresentou resultados muito satisfatórios melhorando a solução encontrada. A porcentagem de sucesso do FRB3+ENG é superior à dos demais métodos para as duas bases de dados utilizadas, Taillard e de Ruben Ruiz.

Em relação ao desvio relativo médio, o algoritmo também apresentou uma redução no desvio relativo médio para os problemas de Taillard e de Ruben Ruiz.

O tempo de computação do método FRB3+ENG é superior a todos os demais métodos, se forem considerados somente os tempos totais de computação. No entanto, a diferença entre o tempo de execução do método FRB3 e FRB3+ENG é menor que 1% em média. Por este fato a incorporação do ENG ao método FRB3 original apresenta uma melhoria real no desempenho do método, resultando em soluções melhores com pequeno acréscimo do tempo de computação do método.

O método apresentado neste trabalho de mestrado, ENG, se mostrou eficiente para a solução de problemas de *flow shop* permutacional. Os resultados apresentados no capítulo anterior fortalecem que sua utilização pode ser expandida para outros tipos de problemas de programação de operações em máquinas, *no-idle flow shop*, *no-wait flow shop*, *flow shop* com tempo de *setup*, entre outros.

## REFERÊNCIAS

- BAKER, K.R. (1974). *Introduction to sequencing and scheduling*. New York: John Wiley.
- CAMPBELL, H.G.; DUDEK, R.A.; SMITH, M.L. (1970). A Heuristic algorithm for the n-job, m-machine sequencing problem. *Management Science*, Providence, v.16, n.9, p.630-637, May.
- CHAKRABORTY, U.K.; LAHA, D. (2007). An Improved heuristic for permutation flowshop scheduling. *International Journal Information and Communication Technology*, v.1, p.89-97.
- DANNENBRING, D.G. (1977). An Evaluation of flow-shop sequencing heuristics. *Management Science*, Providence, v.23, n.11, p.1174-1182, July.
- DAVOUD POUR, H. (2001). A New heuristic for the n-job, m-machine flow-shop problem. *Production Planning and Control*, London, v.12, n.7, p.648-653.
- DONG, X.; HUANG, H.; CHEN, P. (2008). An Improved NEH-based heuristic for the permutation flowshop problem. *Computer & Operations Research*, Amsterdam, v.35, n.12, p.3962-3968, Dez.
- DUDEK, R.A.; PANWALKAR, S.S.; SMITH, M.L. (1991). The Lessons of flow-shop scheduling research. *Operations Research*, Linthicum, v.40, n.1, p.7-13, Jan.
- FRAMINAN, J.M.; GUPTA, J.N.D.; LEISTEN, R. (2004). A Review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*, London, v.55, n.12, p.1243-1255, Dec.
- FRAMINAN, J.M.; LEISTEN, R.; RAJENDRAN, C. (2003). Different initial sequences for the heuristic of nawaz, enscore and ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. *International Journal of Production Research*, London, v.41, n.1, p.121-148.
- FRAMINAN, J.M.; LEISTEN, R.; RUIZ-USANO, R. (2005). Comparison of heuristics for flowtime minimisation in permutation flowshops. *Computers & Operations Research*, Amsterdam, v.32, n.5, p.1237-1254, May.
- FRENCH, S. (1982). *Sequencing and scheduling: an introduction to the mathematics of the job shop*. New York: John Wiley.
- GAREY; M.R.; JOHNSON, D.S. (1979). *Computers and intractability: a guide of NP-completeness*. New York: W.H.Freemon. p.300-340.

GAREY, M.R.; JOHNSON, D.S.; SETHI, R. (1976). The Complexity of flow-shop and job-shop scheduling. *Mathematics of Operations Research*, Providence, v.1, n.2, p.117-129, May.

GRABOWSKI, J.; WODECKI, M. (2004). A Very fast tabu search algorithm for the permutation flow shop problem with makespan criterion. *Computer & Operations Research*, Amsterdam, v.31, n.11, p.1891-1909, Sept.

GRAVES, S.G. (1981). A Review of production scheduling. *Operations Research*, Linthicum, v.29, n.4, p.646-675, July.

GUPTA, J.N.D. (1971). A Functional heuristic algorithm for the flow-shop scheduling problem. *Operational Research Quarterly*, London, v.22, n.1, p.39-47, Mar.

GUPTA, J.N.D.; STAFFORD, E.F. (2005). Flow shop scheduling research after five decades. *European Journal of Operational Research*, Amsterdam, v.169, n.3, p.699-711, Mar.

HEJAZI, S.R.; SAGHAFIAN, S. (2005). Flowshop scheduling problems with makespan criterion: a review. *International Journal of Production Research*, London, v.43, n.14, p.2895-2929.

HO, J.C.; CHANG, Y.L. (1991). A New heuristic for the n-job, m-machine flow-shop problem. *European Journal of Operational Research*, Amsterdam, v.52, n.2, p.194-202, May.

HUNDAL, T.S.; RAJGOPAL, J. (1988). An Extension of Palmer's heuristic for the flow-shop scheduling problem. *International Journal of Production Research*, London, v.26, n.6, p.1119-1124.

JOHNSON, S.M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, Washington, v.1, n.1, p.61-68, Mar.

KALCZYNSKI, P.J.; KAMBUROWSKI, J. (2007). On the NEH heuristic for minimizing the makespan in permutation flow shops. *Omega: the international journal of management science*, Oxford, v.35, n.1, p.53-60, Feb.

KALCZYNSKI, P.J.; KAMBUROWSKI, J. (2008). An Improved NEH heuristic to minimize makespan in permutation flow shops. *Computers & Operations Research*, Amsterdam, v.35, n.9, p.3001-3008, Sept.

KOULAMAS, C. (1998). A New constructive heuristic for the flowshop scheduling problem. *European Journal of Operational Research*, Amsterdam, v.105, n.1, p.66-71, Feb.

LOURENÇO, H.L. (1996). Sevast'yanov's algorithm for the flow-shop scheduling problem. *European Journal of Operational Research*, Amsterdam, v.91, n.1, p.176-189, May.

MACCARTHY, B.L.; LIU, J. (1993). Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, London, v.31, n.1, p.59-79.

MOCCELLIN, J.V. (1992). *Uma contribuição à programação de operações em sistemas de produção intermitente flow-shop*. 126p. Tese (Livre-Docência) – Escola de engenharia de São Carlos, Universidade de São Paulo, São Carlos, 1992.

MOCCELLIN, J.V. (1995). A New heuristic method for the permutation flow shop scheduling problem. *Journal of the Operational Research Society*, London, v.46, n.7, p.883-886, July.

MOCCELLIN, J.V.; NAGANO, M.S. (1998). Evaluating the performance of tabu search procedures for flow shop sequencing. *Journal of the Operational Research Society*, London, v.49, n.12, p.1296-1302, Dec.

NAGANO, M.S.; MOCCELLIN, J.V. (2002). A High quality solution constructive for flow shop sequencing. *Journal of the Operational Research Society*, London, v.53, n.12, p.1374-1379, Dec.

NAGANO, M.S.; MOCCELLIN, J.V.; LORENA, L.A.N. (2005). Redução do estoque em processamento em sistemas de produção flow shop permutacional. *Revista Produção On Line*, Florianópolis, v.5, n.3, Sept. Disponível em: <<http://producaoonline.org.br/index/php/rpo/article/view/363/447>>. Acesso em: 10 ago. 2010.

NAWAZ, M.; ENSCORE JR., E.E.; HAM, I. (1983). A Heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega: the international journal of management science*, Oxford, v.11, n.1, p.91-95.

NOWICKI, E.; SMUTNICKI, C. (1996). A Fast tabu algorithm for the permutation flow-shop problem. *European Journal of Operational Research*, Amsterdam, v.91, n.1, p.160-175, May.

OGBU, F.A.; SMITH, D.K. (1990). The Application of the simulated annealing algorithms to the solution of the  $n/m/C_{max}$  flowshop problem. *Computers & Operations Research*, Amsterdam, v.17, n.3, p.243-253.

OSMAN, I.H.; POTTS, C.N. (1989). Simulated annealing for permutation flow-shop scheduling. *Omega: the international journal of management science*, Oxford, v.17, n.6, p.551-557.

PALMER, D.S. (1965). Sequencing jobs through a multi-stage process in the minimum total time - A quick method of obtaining a near optimum. *Operational Research Quarterly*, London, v.16, n.1, p.101-107, Mar.

POTTS, C.N. (1980). An Adaptive branching rule for the permutation flow-shop problem. *European Journal of Operational Research*, Amsterdam, v.5, n.1, p.19-25, July.

RAD, S.F.; RUIZ, R.; BOROOJERDIAN, N. (2009). New high performing heuristic for minimizing makespan in permutation flowshops. *Omega: the international journal of management science*, Oxford, v.37, n.2, p.331-345, Apr.

RAJENDRAN, C.; ZIEGLER, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, Amsterdam, v.155, n.2, p.426-436, June.

REEVES, C.R. (1993). Improving the efficiency of tabu search for machine sequencing problems. *Journal of the Operational Research Society*, London, v.44, n.4, p.375-382, Apr.

REEVES, C.R. (1995). A Genetic algorithm for flowshop sequencing. *Computers & Operations Research*, v.22, n.1, p.5-13, Jan.

RICKEL, J. (1988). Issues in the design of scheduling systems. In: OLIFF, M. (Ed.). *Expert systems and intelligent manufacturing*. New York: Elsevier Science. p.70-89.

RODAMMER, F.A.; WHITE JR., K.P. (1988). A Recent survey of production scheduling. *IEEE Transactions on Systems, Man and Cybernetics*, New York, v.18, n.6, p.841-851.

RUIZ, R.; MAROTO, C. (2005). A Comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, Amsterdam, v.165, n.2, p.479-494, Sept.

SEVASTJANOV, S. (1995). Vector summation in banach space and polynomial algorithms for flow shops and open shops. *Mathematics of Operations Research*, Providence, v.20, n.1, p.90-103, Feb.

STÜTZLE, T. (1998). *Applying iterated local search to the permutation flow shop problem*. Darmstadt: Darmstad University of Technology. (Technical report, AIDA-98-04).

TAILLARD, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, Amsterdam, v.47, n.1, p.65-74, July.

WIDMER, M.; HERTZ, A. (1989). A New heuristic method for the flow shop sequencing problem. *European Journal of Operational Research*, Amsterdam, v.41, n.2, p.186-19, July.

# APÊNDICE A

## TABELAS DE RESULTADOS DOS MÉTODOS

Tabela A.1 – Resultados dos métodos para os problemas de Taillard.

<b>Tarefas</b>	<b>Máquinas</b>	<b>Problema</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FBR4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>20</b>	<b>5</b>	<b>1</b>	1286	1297	1289	1289	1184
		<b>2</b>	1365	1383	1365	1365	1348
		<b>3</b>	1132	1132	1085	1100	1098
		<b>4</b>	1325	1306	1304	1299	1299
		<b>5</b>	1305	1283	1250	1250	1197
		<b>6</b>	1225	1264	1210	1210	1130
		<b>7</b>	1251	1251	1241	1251	1257
		<b>8</b>	1227	1221	1214	1214	1205
		<b>9</b>	1274	1289	1258	1258	1242
		<b>10</b>	1127	1131	1151	1129	1109
	<b>10</b>	<b>1</b>	1654	1662	1600	1598	1599
		<b>2</b>	1776	1718	1713	1713	1713
		<b>3</b>	1538	1538	1543	1543	1543
		<b>4</b>	1433	1427	1401	1401	1421
		<b>5</b>	1502	1473	1458	1438	1444
		<b>6</b>	1433	1447	1431	1431	1454
		<b>7</b>	1555	1524	1526	1531	1454
		<b>8</b>	1620	1529	1598	1572	1582
		<b>9</b>	1647	1663	1620	1620	1620
		<b>10</b>	1656	1656	1613	1613	1619
	<b>20</b>	<b>1</b>	2443	2380	2365	2365	2287
		<b>2</b>	2134	2162	2147	2147	2126
		<b>3</b>	2418	2387	2399	2378	2378
		<b>4</b>	2263	2248	2249	2247	2208
		<b>5</b>	2384	2363	2373	2373	2373
		<b>6</b>	2349	2378	2273	2273	2234
		<b>7</b>	2383	2366	2345	2345	2348
		<b>8</b>	2249	2279	2232	2232	2232
		<b>9</b>	2313	2292	2279	2272	2248
		<b>10</b>	2220	2308	2263	2263	2263
<b>50</b>	<b>5</b>	<b>1</b>	2732	2724	2729	2729	2733



2	2843	2882	2838	2838	2747
3	2650	2654	2625	2630	2564
4	2782	2768	2762	2782	2750

continua

continuação

Tarefas	Máquinas	Problema	NEHKK	NEH-D	FBR4	FRB3	FRB3+ENG
		5	2871	2890	2864	2864	2784
		6	2845	2846	2835	2835	2795
		7	2735	2736	2736	2725	2706
		8	2695	2704	2683	2683	2626
		9	2607	2571	2581	2564	2539
		10	2786	2790	2782	2782	2799
	10	1	3149	3177	3089	3126	3126
		2	3059	3008	2965	2961	2943
		3	3000	2988	2936	2932	2923
		4	3196	3176	3095	3128	3109
		5	3160	3136	3065	3055	3061
		6	3158	3126	3091	3158	3114
		7	3252	3271	3198	3226	3145
		8	3137	3162	3088	3088	3107
		9	3025	3001	2963	2945	2956
		10	3187	3179	3160	3177	3163
	20	1	4093	4041	4004	4004	4018
		2	3938	3937	3888	3810	3822
		3	3868	3820	3840	3774	3809
		4	3913	3897	3868	3860	3847
		5	3874	3827	3728	3754	3776
		6	3878	3844	3804	3822	3790
		7	3959	3971	3925	3865	3865
		8	3969	3982	3847	3900	3838
		9	3951	3941	3883	3874	3874
		10	4015	3972	3898	3942	3921
100	5	1	5514	5493	5493	5493	5406
		2	5284	5286	5289	5284	5240
		3	5196	5215	5194	5175	5145
		4	5023	5027	5023	5023	5012
		5	5266	5255	5253	5253	5175
		6	5139	5139	5139	5139	5065
		7	5288	5283	5261	5261	5256
		8	5129	5110	5107	5105	5118
		9	5489	5489	5465	5454	5405
		10	5341	5346	5334	5334	5352
	10	1	5886	5845	5829	5801	5829
		2	5460	5426	5419	5389	5368
		3	5743	5740	5691	5691	5670
		4	5926	5935	5865	5860	5884

5	5647	5586	5599	5564	5540
6	5373	5389	5334	5324	5290
7	5681	5736	5673	5659	5634

continua

conclusão

Tarefas	Máquinas	Problema	NEHKK	NEH-D	FBR4	FRB3	FRB3+ENG
		8	5752	5684	5678	5693	5680
		9	5993	5993	5983	5979	5979
		10	5922	5903	5882	5860	5802
	20	1	6502	6511	6448	6467	6418
		2	6494	6506	6434	6361	6389
		3	6534	6535	6433	6455	6415
		4	6611	6606	6532	6498	6474
		5	6590	6584	6536	6520	6475
		6	6725	6666	6608	6556	6528
		7	6631	6682	6481	6480	6491
		8	6839	6770	6683	6596	6596
		9	6630	6599	6505	6482	6436
		10	6683	6658	6585	6614	6555
200	10	1	10986	10942	10893	10893	10804
		2	10722	10649	10612	10560	10531
		3	11025	11027	11026	10969	10905
		4	11057	11057	10918	11057	11049
		5	10575	10628	10572	10544	10560
		6	10476	10403	10390	10391	10323
		7	10965	10945	10939	10863	10834
		8	10798	10825	10798	10798	10757
		9	10594	10525	10492	10470	10452
		10	10795	10789	10776	10735	10758
	20	1	11640	11580	11510	11472	11479
		2	11781	11632	11544	11485	11490
		3	11789	11768	11613	11592	11584
		4	11764	11738	11588	11564	11459
		5	11624	11596	11514	11456	11492
		6	11658	11631	11524	11488	11470
		7	11798	11799	11608	11539	11586
		8	11883	11763	11602	11626	11623
		9	11665	11617	11540	11507	11449
		10	11805	11783	11619	11560	11600
500	20	1	26743	26643	26192	26021	26065
		2	27167	27143	26568	26549	26476
		3	26925	26827	27010	26904	26899
		4	26988	26937	26791	26580	26602
		5	26788	26651	26903	26998	26998
		6	27111	26883	26975	26975	26975
		7	26731	26716	26938	26793	26831

<b>8</b>	27027	27064	27066	27036	26948
<b>9</b>	26627	26467	26992	26963	27029
<b>10</b>	26987	26865	26582	26576	26496

Tabela A.2 – Tabela de resultados dos problemas de Ruben Ruiz

<b>Tarefas</b>	<b>Máquinas</b>	<b>Problema</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>50</b>	<b>10</b>	<b>1</b>	3276	3278	3224	3224	3146
		<b>2</b>	3477	3364	3363	3363	3315
		<b>3</b>	3190	3235	3146	3111	3145
		<b>4</b>	3150	3054	3010	3024	3062
		<b>5</b>	3385	3304	3250	3241	3303
	<b>20</b>	<b>1</b>	4053	4094	3984	3953	3988
		<b>2</b>	4054	3989	3948	3896	3942
		<b>3</b>	4086	4000	3896	3897	3897
		<b>4</b>	4123	4033	3947	3951	3920
		<b>5</b>	4175	4034	3949	3912	3914
	<b>30</b>	<b>1</b>	4975	4852	4788	4736	4777
		<b>2</b>	4868	4783	4679	4619	4619
		<b>3</b>	4750	4673	4592	4584	4584
		<b>4</b>	4752	4720	4678	4665	4630
		<b>5</b>	4783	4727	4695	4657	4643
	<b>40</b>	<b>1</b>	5442	5392	5352	5336	5345
		<b>2</b>	5513	5445	5368	5336	5336
		<b>3</b>	5660	5571	5485	5448	5451
		<b>4</b>	5528	5589	5483	5428	5428
		<b>5</b>	5398	5377	5298	5297	5297
	<b>50</b>	<b>1</b>	6284	6181	6143	6141	6106
		<b>2</b>	6335	6256	6171	6163	6122
		<b>3</b>	6064	6041	5964	5890	5890
		<b>4</b>	6228	6260	6091	6095	6114
		<b>5</b>	6379	6274	6179	6184	6184
<b>100</b>	<b>10</b>	<b>1</b>	5941	5803	5729	5771	5651
		<b>2</b>	5834	5661	5617	5591	5595
		<b>3</b>	5776	5578	5481	5493	5426
		<b>4</b>	5831	5690	5550	5542	5532
		<b>5</b>	5982	5838	5782	5783	5727
	<b>20</b>	<b>1</b>	6827	6687	6628	6513	6560
		<b>2</b>	6725	6659	6505	6501	6490
		<b>3</b>	6866	6642	6596	6496	6490
		<b>4</b>	6794	6554	6430	6421	6421
		<b>5</b>	7015	6776	6655	6681	6691
	<b>30</b>	<b>1</b>	7651	7516	7361	7316	7316
		<b>2</b>	7749	7464	7354	7304	7304
		<b>3</b>	7712	7616	7504	7451	7454
		<b>4</b>	7813	7635	7533	7484	7484

	<b>5</b>	7840	7740	7606	7569	7535
<b>40</b>	<b>1</b>	8519	8406	8229	8238	8221
	<b>2</b>	8468	8226	8171	8104	8095
	<b>3</b>	8359	8324	8251	8210	8161

continua

continuação

<b>Tarefas</b>	<b>Máquinas</b>	<b>Problema</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>	
			<b>4</b>	8550	8443	8189	8160	8233
			<b>5</b>	8375	8332	8163	8108	8108
	<b>50</b>		<b>1</b>	9327	9190	9091	9035	9012
			<b>2</b>	9237	9130	8966	8952	8931
			<b>3</b>	9137	9108	9030	8951	8981
			<b>4</b>	9326	9156	9010	8954	8946
			<b>5</b>	9303	9097	8997	8921	8914
<b>150</b>	<b>10</b>		<b>1</b>	8790	8427	8367	8360	8348
			<b>2</b>	8419	8140	8063	8029	8046
			<b>3</b>	8549	8112	8049	8061	8041
			<b>4</b>	8671	8410	8415	8412	8412
			<b>5</b>	8080	8089	8062	8044	7984
	<b>20</b>		<b>1</b>	9435	9185	9043	9009	8977
			<b>2</b>	9471	9216	9068	9063	9030
			<b>3</b>	9562	9272	9056	9037	9109
			<b>4</b>	9279	8975	8877	8813	8834
			<b>5</b>	9417	9275	9133	9047	9073
	<b>30</b>		<b>1</b>	10557	10285	10117	10045	10028
			<b>2</b>	10342	10176	9969	9962	9912
			<b>3</b>	10467	10297	10160	10095	10036
			<b>4</b>	10595	10444	10227	10136	10136
			<b>5</b>	10370	10178	9939	9940	9905
	<b>40</b>		<b>1</b>	11214	11117	10988	10877	10901
			<b>2</b>	11424	11134	11035	10936	10876
			<b>3</b>	11323	11092	10855	10804	10795
			<b>4</b>	11145	10971	10866	10764	10719
			<b>5</b>	11450	11331	11139	11016	11004
	<b>50</b>		<b>1</b>	12066	11980	11821	11708	11746
			<b>2</b>	12207	12086	11733	11781	11781
			<b>3</b>	12065	11844	11720	11756	11756
			<b>4</b>	12062	11987	11803	11722	11722
			<b>5</b>	12177	12125	11917	11856	11797
	<b>200</b>	<b>10</b>	<b>1</b>	11107	11012	11001	10991	10899
			<b>2</b>	11257	10922	10870	10837	10813
			<b>3</b>	11109	10984	10980	10980	10799
			<b>4</b>	11182	11073	11073	11073	11010
			<b>5</b>	11059	10673	10638	10638	10652
	<b>20</b>		<b>1</b>	12055	11782	11612	11556	11556
			<b>2</b>	12141	11649	11582	11441	11427

	<b>3</b>	12336	11985	11911	11890	11782
	<b>4</b>	11950	11745	11581	11531	11561
	<b>5</b>	12144	11846	11641	11677	11655
<b>30</b>	<b>1</b>	12937	12685	12478	12433	12446

continua

continuação

<b>Tarefas</b>	<b>Máquinas</b>	<b>Problema</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
		<b>2</b>	13242	12796	12574	12534	12496
		<b>3</b>	13076	12635	12455	12418	12453
		<b>4</b>	13365	13092	12785	12772	12751
		<b>5</b>	13255	12958	12703	12661	12690
	<b>40</b>	<b>1</b>	14048	13936	13725	13532	13494
		<b>2</b>	14169	13995	13714	13577	13533
		<b>3</b>	14320	13799	13727	13605	13598
		<b>4</b>	14037	13645	13517	13395	13457
		<b>5</b>	14231	13835	13692	13534	13571
	<b>50</b>	<b>1</b>	14957	14906	14718	14446	14446
		<b>2</b>	14926	14765	14535	14427	14421
		<b>3</b>	15068	14819	14543	14518	14506
		<b>4</b>	15092	14848	14553	14435	14426
		<b>5</b>	15035	14702	14536	14480	14454
<b>250</b>	<b>10</b>	<b>1</b>	13972	13669	13623	13623	13559
		<b>2</b>	13865	13443	13413	13400	13417
		<b>3</b>	13674	13346	13289	13270	13207
		<b>4</b>	14112	13613	13581	13578	13675
		<b>5</b>	13899	13401	13352	13352	13289
	<b>20</b>	<b>1</b>	14458	14000	13888	13775	13767
		<b>2</b>	14833	14276	14113	13997	13979
		<b>3</b>	14649	14290	14119	14027	14038
		<b>4</b>	14889	14334	14132	14061	14056
		<b>5</b>	14866	14319	14000	13932	13983
	<b>30</b>	<b>1</b>	15610	15316	15048	14972	14967
		<b>2</b>	15892	15333	15197	15079	15148
		<b>3</b>	15750	15429	15192	15090	15152
		<b>4</b>	15905	15476	15228	15111	15073
		<b>5</b>	15644	15361	15006	14884	14918
	<b>40</b>	<b>1</b>	16728	16251	16127	16040	15986
		<b>2</b>	16820	16198	16039	15939	15972
		<b>3</b>	16737	16366	16124	16073	15998
		<b>4</b>	16956	16754	16443	16336	16323
		<b>5</b>	16888	16534	16244	16206	16138
	<b>50</b>	<b>1</b>	17869	17529	17203	17099	17133
		<b>2</b>	17610	17369	17068	16944	16938
		<b>3</b>	17614	17236	17033	16891	16926
		<b>4</b>	17670	17232	17018	16828	16955
		<b>5</b>	17851	17521	17274	17141	17106

<b>300</b>	<b>10</b>	<b>1</b>	16301	15882	15864	15861	15832
		<b>2</b>	16033	15904	15904	15904	15971
		<b>3</b>	16376	16087	16030	16026	15963
		<b>4</b>	16557	16191	16173	16172	16050

continua

continuação

<b>Tarefas</b>	<b>Máquinas</b>	<b>Problema</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
		<b>5</b>	16457	16326	16227	16227	16205
	<b>20</b>	<b>1</b>	17436	16911	16759	16664	16634
		<b>2</b>	17597	16883	16601	16583	16529
		<b>3</b>	17360	16800	16546	16470	16419
		<b>4</b>	17299	16629	16537	16440	16450
		<b>5</b>	17688	16947	16885	16692	16678
	<b>30</b>	<b>1</b>	18290	17846	17645	17526	17522
		<b>2</b>	18662	18230	17915	17855	17846
		<b>3</b>	18551	17978	17827	17608	17655
		<b>4</b>	18719	18301	18083	17863	17804
		<b>5</b>	18441	17926	17562	17498	17568
	<b>40</b>	<b>1</b>	19433	19002	18767	18602	18558
		<b>2</b>	19585	18929	18645	18521	18577
		<b>3</b>	19728	19268	18977	18841	18895
		<b>4</b>	19497	18948	18678	18582	18567
		<b>5</b>	19563	19113	18850	18647	18681
	<b>50</b>	<b>1</b>	20744	20186	19938	19753	19814
		<b>2</b>	20506	20098	19717	19609	19540
		<b>3</b>	20728	20402	20111	19875	19870
		<b>4</b>	20480	20056	19772	19665	19636
		<b>5</b>	20448	20001	19743	19612	19501
<b>350</b>	<b>10</b>	<b>1</b>	19198	18782	18819	18774	18684
		<b>2</b>	18249	17834	17821	17818	17895
		<b>3</b>	19412	18769	18801	18763	18685
		<b>4</b>	18977	18437	18448	18420	18369
		<b>5</b>	19226	18765	18749	18749	18783
	<b>20</b>	<b>1</b>	20049	19297	19213	19090	19059
		<b>2</b>	20592	19857	19679	19646	19662
		<b>3</b>	20163	19439	19276	19243	19272
		<b>4</b>	20271	19480	19299	19264	19296
		<b>5</b>	19703	19170	19030	18897	18866
	<b>30</b>	<b>1</b>	20973	20429	20312	20034	20067
		<b>2</b>	20808	20288	20011	19829	19814
		<b>3</b>	21121	20662	20293	20249	20186
		<b>4</b>	21116	20393	20141	19950	19970
		<b>5</b>	20894	20473	20257	19984	19991
	<b>40</b>	<b>1</b>	22700	21890	21597	21417	21403
		<b>2</b>	22052	21775	21383	21180	21212
		<b>3</b>	22084	21445	21149	20946	20936

	<b>4</b>	22166	21598	21251	21127	20982
	<b>5</b>	22252	21626	21370	21168	21166
<b>50</b>	<b>1</b>	23171	22748	22406	22238	22223
	<b>2</b>	23264	22734	22473	22361	22363

continua

continuação

<b>Tarefas</b>	<b>Máquinas</b>	<b>Problema</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
		<b>3</b>	23284	22539	22263	22159	22137
		<b>4</b>	23262	22660	22339	22116	22141
		<b>5</b>	23239	22636	22337	22216	22137
<b>400</b>	<b>10</b>	<b>1</b>	21878	21259	21248	21228	21141
		<b>2</b>	21650	20937	20901	20901	20879
		<b>3</b>	21972	21876	21876	21876	21873
		<b>4</b>	21595	20755	20755	20755	20699
		<b>5</b>	21893	20953	20890	20884	20891
	<b>20</b>	<b>1</b>	22776	21765	21576	21416	21439
		<b>2</b>	22788	22008	21815	21747	21715
		<b>3</b>	22743	22098	22000	21999	21999
		<b>4</b>	22631	21876	21718	21680	21657
		<b>5</b>	22739	21981	21845	21694	21693
	<b>30</b>	<b>1</b>	24025	23104	22838	22723	22757
		<b>2</b>	23982	23152	22956	22797	22744
		<b>3</b>	23971	22942	22804	22609	22603
		<b>4</b>	23959	23242	22810	22636	22717
		<b>5</b>	23879	23233	23089	22847	22811
	<b>40</b>	<b>1</b>	25053	24370	24026	23830	23835
		<b>2</b>	25120	24426	24062	23797	23726
		<b>3</b>	24831	24239	23824	23719	23629
		<b>4</b>	25059	24163	23791	23489	23603
		<b>5</b>	24710	24268	23777	23649	23609
	<b>50</b>	<b>1</b>	26062	25407	25017	24787	24917
		<b>2</b>	26114	25469	25079	24817	24877
		<b>3</b>	26317	25554	25212	25008	24942
		<b>4</b>	26253	25421	25116	24850	24846
		<b>5</b>	25843	25377	24936	24711	24709
<b>450</b>	<b>10</b>	<b>1</b>	23483	22558	22485	22469	22442
		<b>2</b>	23985	23411	23333	23340	23340
		<b>3</b>	24104	23034	23034	23034	22896
		<b>4</b>	23629	23100	23035	23028	23037
		<b>5</b>	24647	23897	23896	23895	23841
	<b>20</b>	<b>1</b>	25426	24331	24135	24007	24012
		<b>2</b>	25508	24316	24227	24090	24118
		<b>3</b>	25029	24269	23965	23950	23950
		<b>4</b>	25171	24013	23874	23741	23742
		<b>5</b>	25274	24234	23968	23879	23857
	<b>30</b>	<b>1</b>	27002	26104	25812	25651	25716

2	26641	25517	25282	25095	25054
3	26276	25431	25006	24995	24927
4	26532	25405	25076	24970	24990
5	26947	26149	25762	25678	25719

continua

conclusão

<b>Tarefas</b>	<b>Máquinas</b>	<b>Problema</b>	<b>NEHKK</b>	<b>NEH-D</b>	<b>FRB4</b>	<b>FRB3</b>	<b>FRB3+ENG</b>
<b>500</b>	<b>40</b>	<b>1</b>	27619	26918	26631	26470	26499
		<b>2</b>	27529	26881	26458	26224	26154
		<b>3</b>	27674	27014	26725	26369	26375
		<b>4</b>	27409	26711	26386	26073	26110
		<b>5</b>	27798	26910	26624	26395	26396
	<b>50</b>	<b>1</b>	29050	28396	28040	27838	27844
		<b>2</b>	28275	27782	27379	27194	27142
		<b>3</b>	28877	28088	27681	27484	27425
		<b>4</b>	28706	28019	27644	27481	27534
		<b>5</b>	28687	28140	27711	27548	27571
	<b>10</b>	<b>1</b>	26763	26088	26059	26057	25930
		<b>2</b>	26262	26054	26054	26048	25928
		<b>3</b>	27074	26533	26516	26516	26495
		<b>4</b>	27060	26466	26460	26459	26422
		<b>5</b>	26214	25292	25285	25273	25274
	<b>20</b>	<b>1</b>	28687	27545	27432	27355	27357
		<b>2</b>	28705	27715	27358	27263	27263
		<b>3</b>	28304	27254	27022	27005	26975
		<b>4</b>	28017	26778	26415	26299	26375
		<b>5</b>	28131	26890	26648	26534	26567
	<b>30</b>	<b>1</b>	29028	27953	27679	27411	27438
		<b>2</b>	29035	27985	27669	27564	27524
		<b>3</b>	29103	28321	28080	27925	27809
		<b>4</b>	28919	27835	27592	27399	27429
		<b>5</b>	29310	28304	28046	27846	27862
<b>40</b>	<b>1</b>	30240	29487	29147	28974	28870	
	<b>2</b>	30340	29540	29213	29075	29075	
	<b>3</b>	30152	29167	28785	28592	28715	
	<b>4</b>	30149	29441	29176	28787	28765	
	<b>5</b>	30747	29697	29507	29184	29148	
<b>50</b>	<b>1</b>	31605	30704	30359	30091	30078	
	<b>2</b>	31672	30721	30440	30248	30248	
	<b>3</b>	31877	30931	30535	30281	30331	
	<b>4</b>	31302	30601	30082	29793	29864	
	<b>5</b>	31344	30477	29945	29732	29833	





# APÊNDICE B

---

## CÓDIGO FONTE DO MÉTODO PROPOSTO

```
int h;      // posição onde a tarefa foi inserida
int l;      // tamanho da sequência parcial.
int i, aux; // variáveis auxiliares.
int seq_aux[n];      // Sequência teste
int best_seq[n];     // Resultado atual

if((h > 1) && (l >=3) && (h < l)){

    for(i=0;i <= l;i++){
        seq_aux[i] = best_seq[i];
    }

    aux = seq_aux[h-1];
    seq_aux[h-1] = seq_aux[h+1];
    seq_aux[h+1] = aux;

    Makespan_Teste = DataDeTermino(l);

    if(Makespan_Teste < Makespan_Atual){
        for(i=0; i <= l; i++){
            best_seq[i] = seq_aux[i];
        }

        Makespan_Atual = Makespan_Teste;
    }
}

if((h == 1) & (l >= 3)){

    for(i=0;i<=l;i++){
```

```
seq_aux[i] = best_seq[i];
}
```

```
aux = seq_aux[h+1];
seq_aux[h+1] = seq_aux[h+2];
seq_aux[h+2] = aux;
```

```
Makespan_Teste = DataDeTermino (l);
```

```
if(Makespan_Teste < Makespan_Atual){
```

```
    for(i=0;i<=l;i++){
        best_seq[i] = seq_aux[i];
    }
```

```
    Makespan_Atual = Makespan_Teste;
```

```
}
```

```
if((h == l) & (l >= 3)){
```

```
    for(i=0;i<=l;i++){
        seq_aux[i] = best_seq[i];
    }
```

```
aux = seq_aux[h-1];
seq_aux[h-1] = seq_aux[h-2];
seq_aux[h-2] = aux;
```

```
Makespan_Teste = DataDeTermino (l);
```

```
if(Makespan_Teste < Makespan_Atual){
```

```
    for(i=0;i<=l;i++){
        best_seq[i] = seq_aux[i];
    }
```

```
    Makespan_Atual = Makespan_Teste;
```

```
}
```

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)