

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ENGENHARIA DE PRODUÇÃO**

DEBORA CORRÊA PENA

**DESENVOLVIMENTO DE UMA METAHEURÍSTICA PARA
MINIMIZAÇÃO DO ATRASO TOTAL EM PROBLEMAS DE
FLOW SHOP COM SOLUÇÃO NÃO PERMUTACIONAL**

LONDRINA

2018

DEBORA CORRÊA PENA

**DESENVOLVIMENTO DE UMA METAHEURÍSTICA PARA
MINIMIZAÇÃO DO ATRASO TOTAL EM PROBLEMAS DE
FLOW SHOP COM SOLUÇÃO NÃO PERMUTACIONAL**

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Produção da Universidade Tecnológica Federal do Paraná, Campus Londrina, como requisito parcial para obtenção do título de “Engenheira de Produção”.

Orientador: Prof. Dr. Rafael Henrique Palma Lima

LONDRINA

2018

TERMO DE APROVAÇÃO

DESENVOLVIMENTO DE UMA METAHEURÍSTICA PARA MINIMIZAÇÃO DO ATRASO TOTAL EM PROBLEMAS DE FLOW SHOP COM SOLUÇÃO NÃO PERMUTACIONAL

POR

DEBORA CORREA PENA

Esta Monografia foi apresentada às 16 horas do dia 27 de novembro de 2018 como requisito parcial para obtenção do título de bacharel em ENGENHARIA DE PRODUÇÃO, Universidade Tecnológica Federal do Paraná – Campus Londrina. O candidato foi arguido pela Banca Examinadora composta pelos professores relacionados abaixo. Após deliberação, a Banca Examinadora considerou o trabalho: **APROVADO**.

Prof. Dr. Rogério Tondato (UTFPR)
Banca Examinadora

Prof. Bruno Samways dos Santos (O)
Banca Examinadora

Prof. Dr. Rafael Henrique Palma Lima (UTFPR)
Presidente da Banca Examinadora
Orientador

RESUMO

O sequenciamento da produção é fundamental para otimizar a utilização dos recursos de um processo produtivo. Seu papel é central na coordenação do planejamento de produção, determinando a designação de tarefas às máquinas visando o atendimento ao cliente, sejam estes internos ou externos, em tempo hábil de entrega. Uma das possíveis configurações de sistemas produtivos é por meio de arranjos sendo destes o *flow shop*. Neste tipo de layout ou arranjo os *jobs* passam pela mesma série de máquinas, não necessariamente em todas, mas sempre seguindo o mesmo fluxo sem que tenham que voltar um passo para que sua finalização seja concluída. Por se tratar de um problema computacionalmente complexo, a literatura especializada tem utilizado metaheurísticas para encontrar boas soluções. Este trabalho descreve a implementação de um algoritmo baseado na metaheurísticas *Iterated Local Search* (ILS) para minimização do atraso total em problemas de *flow shop* com solução não permutacional ou non-permutation flow shop (NPFS). Considerando que a literatura atual não apresenta instâncias nos modelos necessários para os testes da metaheurística, foram desenvolvidas instâncias aleatórias para sua aplicação. As instâncias foram criadas e posteriormente processadas em três ILS cada um com um tipo de perturbação diferente visando abranger o maior universo de possibilidades possíveis para os resultados do problema.

Palavras-chave: Sequenciamento de produção; *Flow shop problem*; Não Permutacional; NPFS; ILS;.

ABSTRACT

The scheduling of production is fundamental to optimize the utilization of the resources of a productive process. Coordination of production planning, determining the assignment of tasks to the machines for customer service in a timely delivery. One of the possible configurations of productive systems is through arrangements being these the Flow Shop. In this type of layout or arrangement the jobs pass through the same series of machines, not necessarily in all but always following the same flow without having to go back one step so that their completion is completed. Because it is a computationally complex problem, specialized literature has used metaheuristics to find good solutions. This work describes the implementation of an algorithm based on Iterated Local Search (ILS) metaheuristics to minimize the total delay in flow shop problems with non-permutational solution (NPFS). Considering that the current literature doesn't present instances in the models necessary for the metaheuristic tests, random instances were developed for its application. Instances were created and later processed into three ILSs each with a different type of perturbation aiming to find the best best result for the type of problem.

Keywords: Production Scheduling; *Flow shop problem*; Non-permutation; NPFS; ILS;.

ÍNDICE DE FIGURAS

Figura 1 - Fluxo de <i>Single Machine Layout</i>	14
Figura 2 - Gráfico de Gantt com resultado do sequenciamento <i>single machine</i> com tempos de processamento da Tabela 1.....	15
Figura 3 - Fluxo de <i>Parallel Machine Layout</i>	16
Figura 4 - Gráfico de Gantt com resultado do sequenciamento em arranjo de máquinas paralelas com tempos de processamento da Tabela 2.....	16
Figura 5 - <i>Fluxo de Flow Shop Layout</i>	17
Figura 6 - Gráfico de Gantt com resultado do sequenciamento em <i>layout flow shop</i> com tempos de processamento da Tabela 3.....	18
Figura 7 - Fluxo de <i>Job Shop Layout</i>	18
Figura 8 - Gráfico de Gantt com resultado do sequenciamento em <i>layout job shop</i> com tempos de processamento da Tabela 3 e rotas de processamento da Tabela 4.....	19
Figura 9 - Fluxo de <i>Open Shop Layout</i>	20
Figura 10 - Fluxo de <i>Hybrid Layout</i>	20
Figura 11 - Representação do gráfico de Gantt para um problema de 3 máquinas e 4 <i>jobs</i> em <i>layout Flow Shop</i> Permutaciona.....	26
Figura 12 - Representação do gráfico de Gantt para um problema de 3 máquinas e 4 <i>jobs</i> em <i>layout Flow Shop</i> Nao Permutaciona.....	27
Figura 13 - <i>Layout Flow Shop</i> com variação <i>Flow Line</i>	30
Figura 14 – Como o ILS evita estagnação em ótimos locais por perturbação.	30
Figura 15 - Pseudocódigo ILS.....	31
Figura 16 - Aba de Processamento PRINCIPAL.....	33
Figura 17 - Aba de Processamento ILS.....	34
Figura 18 - Pseudocódigo Solução Inicial Aleatória.....	34
Figura 19 - Representação da Matriz Aleatória Gerada Pelo Pseudocódigo Solução Inicial Aleatória.....	35
Figura 20 – Pseudocódigo Solver ILS.....	36
Figura 21 - Aba de Processamento SOLUÇÃO.....	37
Figura 22 - Representação Logica da Vizinhança 1 (Macro Esfera).....	38

Figura 23 -Representação Logica da Vizinhança 2 (Micro Esfera)	39
Figura 24 - Modelo de Perturbação Retira <i>Jobs</i>	40
Figura 25 - Modelo de Perturbação Troca Colunas.....	41
Figura 26 - Modelo de Perturbação Inverte Matriz	41
Figura 27 - Pseudocódigo Gera_Instância	43
Figura 28 - Exemplo de nomenclatura e disposição dos dados em uma instância gerada pelo algoritmo gerador de instâncias.....	44
Figura 29 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 1 Em 10 <i>Jobs</i>	46
Figura 31 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 10 <i>Jobs</i> Com Perturbação 1.....	48
Figura 31 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 10 <i>Jobs</i> Com Perturbação 1	48
Figura 32 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 1 Em 10 <i>Jobs</i>	48
Figura 34 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 20 <i>Jobs</i> Com Perturbação 1.....	50
Figura 34 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 20 <i>Jobs</i> Com Perturbação 1	50
Figura 35 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 1 Em 10 <i>Jobs</i>	50
Figura 37 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 30 <i>Jobs</i> Com Perturbação 1.....	52
Figura 37 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 30 <i>Jobs</i> Com Perturbação 1	52
Figura 38 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 2	53
Figura 40 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 10 <i>Jobs</i> Com Perturbação 2.....	54
Figura 40 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 10 <i>Jobs</i> Com Perturbação 2.....	54
Figura 41 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 2	55

Figura 43 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 20 <i>Jobs</i> Com Perturbação 2.....	56
Figura 43 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 20 <i>Jobs</i> Com Perturbação 2.....	56
Figura 44 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 2	57
Figura 46 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 30 <i>Jobs</i> Com Perturbação 2.....	58
Figura 46 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 30 <i>Jobs</i> Com Perturbação 2.....	58
Figura 47 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 3	59
Figura 49 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 10 <i>Jobs</i> Com Perturbação 3.....	60
Figura 49 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 10 <i>Jobs</i> Com Perturbação 3.....	60
Figura 50 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 3	60
Figura 52 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 20 <i>Jobs</i> Com Perturbação 3.....	62
Figura 52 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 20 <i>Jobs</i> Com Perturbação 3.....	62
Figura 53 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 3	62
Figura 54 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 30 <i>Jobs</i> Com Perturbação 3.....	63
Figura 55 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 30 <i>Jobs</i> Com Perturbação 3.....	63

ÍNDICE DE TABELAS

Tabela 1 - Tempos de processamentos de seis <i>jobs</i> em uma máquina para <i>single machine layout</i>	15
Tabela 2 - Tempos de Processamento hipotéticos de 8 <i>jobs</i> em 3 máquinas em arranjo de máquinas paralelas.	16
Tabela 3 - Tempos de processamentos hipotéticos de 5 <i>jobs</i> e 4 máquinas para <i>layout flow shop</i>	17
Tabela 4 - Rotas de processamento para 5 <i>jobs</i> no <i>layou job shop</i>	19
Tabela 5 - Artigos publicados envolvendo NPFS e comparações de NPFS e PFS	28
Tabela 6 - Tabela de Combinações Para Instancias Geradas	45
Tabela 7 - Dados das Instâncias de 10 <i>Jobs</i> Com Perturbação 1.....	47
Tabela 8 - Dados das Instâncias de 20 <i>Jobs</i> Com Perturbação 1.....	49
Tabela 9 - Dados das Instâncias de 30 <i>Jobs</i> Com Perturbação 1.....	51
Tabela 10 - Dados das Instâncias de 10 <i>Jobs</i> Com Perturbação 2.....	53
Tabela 11 - Dados das Instâncias de 20 <i>Jobs</i> Com Perturbação 2.....	55
Tabela 12 - Dados das Instâncias de 30 <i>Jobs</i> Com Perturbação 2.....	57
Tabela 13 - Dados das Instâncias de 10 <i>Jobs</i> Com Perturbação 3.....	59
Tabela 14 - Dados das Instâncias de 20 <i>Jobs</i> Com Perturbação 3.....	61
Tabela 15 - Dados das Instâncias de 30 <i>Jobs</i> Com Perturbação 3.....	63
Tabela 16 - Analise Geral das Perturbações Para Instâncias de 10 <i>Jobs</i>	64
Tabela 17 - Analise Geral das Perturbações Para Instâncias de 20 <i>Jobs</i>	65
Tabela 18 - Analise Geral das Perturbações Para Instâncias de 30 <i>Jobs</i>	66
Tabela 19 - Tempo Médio Por Iteração Em Cada Grupo de <i>Jobs</i> x Máquinas	67

SUMÁRIO

1.INTRODUÇÃO	11
2. REFERENCIAL TEÓRICO	13
2.1 SEQUENCIAMENTO DE PRODUÇÃO	13
2.1.1 <i>Single Machine Layout</i>	14
2.1.2 <i>Parallel Machine Layout</i>	15
2.1.3 <i>Flow Shop Layout</i>	17
2.1.4 <i>Job Shop Layout</i>	18
2.1.5 <i>Open Shop Layout</i>	19
2.1.6 <i>Hybrid Layouts</i>	20
2.1.7 Regras de Sequenciamento.....	21
2.2 FLOW SHOP NÃO PERMUTACIONAL	24
2.3 ITERATED LOCAL SEARCH (ILS)	30
3. METODO DE PESQUISA	33
3.1. ESTRUTURA GERAL DO ALGORITMO (ILS)	33
3.2 ESTRUTURA DE VIZINHANÇA	37
3.3 PERTURBAÇÕES	39
4. RESULTADOS	42
4.1 GERAÇÃO DE INSTÂNCIAS	42
4.2 PERTURBAÇÕES	46
4.2.1 Resultados Perturbação 1 – Retira <i>Jobs</i>	46
4.2.2 Resultados Perturbação 2 – Troca Coluna.....	52
4.2.3 Resultados Perturbação 3 – Inverte Matriz.....	58
5. DISCUSSÃO	64
6. CONCLUSÃO	68
8. REFERÊNCIAS	70

1.INTRODUÇÃO

O sucesso de uma empresa habitualmente está atrelado ao seu bom gerenciamento e planejamento. Assim ao pensar na manufatura dos produtos, o setor de programação e controle da produção (PCP) tem um papel fundamental na boa execução e gerenciamento do plano produtivo que muda constantemente conforme a sazonalidade da demanda de produtos (KIESKOSKI, 2016).

Uma das técnicas de gestão utilizados constantemente pelos programadores é o *scheduling* ou sequenciamento da produção. Assim a partir da análise correta do arranjo físico e da fixação de uma função objetivo que corresponda aos objetivos gerais da companhia os programadores podem sequenciar as ordens de produção de linhas inteiras otimizando tempos de ciclo, número de tarefas atrasadas, atrasos ponderados e diversas outras funções objetivo (LUSTOSA, 2008).

No atual cenário econômico que nos encontramos mesmo com a constante busca das empresas em tornar seus produtos únicos e individualizados, grande parte da produção mundial permanece acontecendo em massa e de maneira seriada. A produção em massa e repetitiva enquadra-se no arranjo ou *layout* de fabrica denominado *flow shop*, onde n jobs são executados em m máquinas de maneira seriada e repetitiva (MELO, 2014).

O objetivo deste trabalho é desenvolver uma metaheurística baseada na técnica *Iterated Local Search* (ILS) para obter soluções factíveis que buscam minimizar o atraso máximo em uma sequência de produção em um layout de *flow-shop*. As sequências que serão formuladas para este problema utilizarão o sequenciamento não permutacional, ou seja, a sequência dos jobs poderá ser diferente para cada uma das máquinas. Dentre os objetivos específicos deste trabalho estão:

1. Definir por meio de pesquisa bibliográfica o tipo de problema estudado e seu método de resolução.
2. Desenvolver uma methaheurística , com base na bibliográfica, para a efetividade do sequenciamento dos pedidos de maneira generalizada sendo possível aplicar a mesma em diversas companhias com o arranjo *flow shop*.
3. Aplicar a methaheurística desenvolvida nas instâncias hipotéticas criadas pelo autor otimizando o sequenciamento a afim de minimizar o número total de atrasos.

O desenvolvimento da metaheurística visa ainda o auxílio no sequenciamento produtivo em companhias com *layout flow shop* que optam por uma solução de sequenciamento não permutacional (NPFS), e que não tem ou não fazem uso de *softwares* próprios para sequenciamento e ainda utilizam planilhas Excel para a realização do mesmo. A não utilização de um software que proporcione um sequenciamento fabril estruturado leva a realização empírica do mesmo pelo programador. Desta forma com a execução do sequenciamento por uma planilha Excel clara e objetiva o processo para estas empresas torna-se uma atividade padronizada que pode ser executada por qualquer colaborador do setor.

O trabalho tem seu texto estruturado em seis seções, descritas a seguir:

- **Capítulo 1:** Introdução e contextualização do tema, objetivos gerais e específicos, justificativas e a estrutura do mesmo.
- **Capítulo 2:** Referencial teórico subdividido em sequenciamento de produção, *flow shop* não permutacional e *iterated local search* (ILS).
- **Capítulo 3:** O método e a técnica de pesquisa, relatando o desenvolvimento e geração das instâncias, esboço do algoritmo e realização de testes.
- **Capítulo 4:** Análise e resultado das instâncias processadas pelo algoritmo.
- **Capítulo 5:** Conclusão sobre os dados
- **Capítulo 6:** Referencial bibliográfico de todas as referências utilizadas durante o trabalho.

2. REFERENCIAL TEÓRICO

2.1 SEQUENCIAMENTO DE PRODUÇÃO

Comum na vida cotidiana, o sequenciamento ou *scheduling* é utilizado em atividades e compromissos simples do dia a dia de acordo com as prioridades e o tempo útil necessário para realizá-las. De maneira semelhante, a indústria utiliza do mesmo artifício para otimizar seus recursos críticos como máquinas, mão-de-obra e tempo, afim de atingir seus objetivos, como aumento da eficiência, utilização da capacidade e rentabilidade para a organização (DILLEP, 2008).

Segundo Torres (2003), o sequenciamento de produção torna-se um artifício essencial à programação das companhias, pois atua minimizando ou maximizando múltiplos critérios, ao mesmo tempo que se adapta às necessidades e limitações da companhia e dos clientes. A literatura ainda correlaciona diretamente a solução de problemas de integração entre planejamentos táticos e operacionais da produção à boa prática do sequenciamento pois o mesmo alinha os objetivos estratégicos da organização ao desdobrar os diversos níveis de planejamento (ANTHONY; GOVINDARAJAN, 1998).

Os estudos sobre *scheduling* tiveram início por volta dos anos 50, após uma equipe de pesquisadores implementar melhorias nas linhas de montagem de uma montadora de carros, gerando a redução dos custos operacionais dos processos e consecutivamente a competitividade da companhia. Contudo somente em 1960 após cientistas da computação utilizarem a lógica de sequenciamento para o desenvolvimento de um novo software de planejamento, devido às limitações tecnológicas da época, foi que o sequenciamento da produção passou a ter suas pesquisas economicamente impulsionadas (LEUNG, 2004).

O sequenciamento da produção é utilizado na atualidade para regularização do processo produtivo de muitas indústrias, auxiliando na tomada de decisão e alocando os recursos e tarefas a fim de otimizar um ou mais objetivos. Segundo Morton e Pentico (1993), o *scheduling* busca sempre entrelaçar as melhores formas de execução de uma série de ações a uma oferta limitada de recursos por um período de tempo especificado. Isto se dá devido à complexidade do sequenciamento, pois cada um dos projetos a serem executados tem determinados *jobs*, ou tarefas, que necessitam de determinado recurso e tempo de processo. Por sua vez os recursos podem variar suas operações em termos de máquinas, células de trabalho, transporte

e atrasos. Todas essas variáveis ainda podem ser intensificadas por restrições externas ou precedências de produção aumentando ainda mais o nível de complexidade necessário para o planejamento.

De acordo com Leung (2004) a partir da década de 1980, as vertentes do sequenciamento tornaram-se mais amplas abrindo leque para que tomassem diversas direções, o que permitiu a elaboração de algoritmos para várias segmentos. Essas aberturas permitiram que os pesquisadores classificassem os problemas mais comuns de acordo com os diferentes arranjos de máquinas e suas diferentes rotas de processamentos específicos, construindo os arranjos do sistema produtivo ou *layouts*. Estes apresentam por sua vez diferentes algoritmos que ainda podem variar de acordo com a busca de um ou mais objetivos, podendo ser estes a minimização dos atrasos de pedidos, tempo de fluxo dos estoques intermediários, tempo de processamento (*makespan*) entre outros.

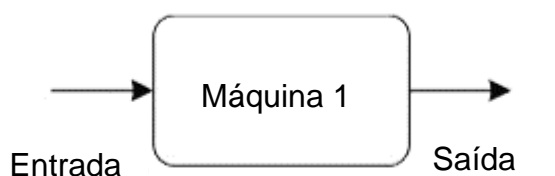
No contexto dos problemas de sequenciamento de produção Framinan, Leisten e García (2014) configuram os *layouts* em cinco tipos:

2.1.1 *Single Machine Layout*

O *layout Single Machine* ou de Uma Máquina, ilustrado na Figura 1, consiste no *layout* mais simples, pois trata-se apenas de uma máquina por onde passam todos os *jobs* de maneira única.

Na Figura 2, é apresentado um exemplo de sequenciamento de 6 trabalhos distintos em uma máquina modelado no gráfico de Gantt, que usam por base os tempos de processamento disposto na Tabela 1.

Figura 1 - Fluxo de *Single Machine Layout*



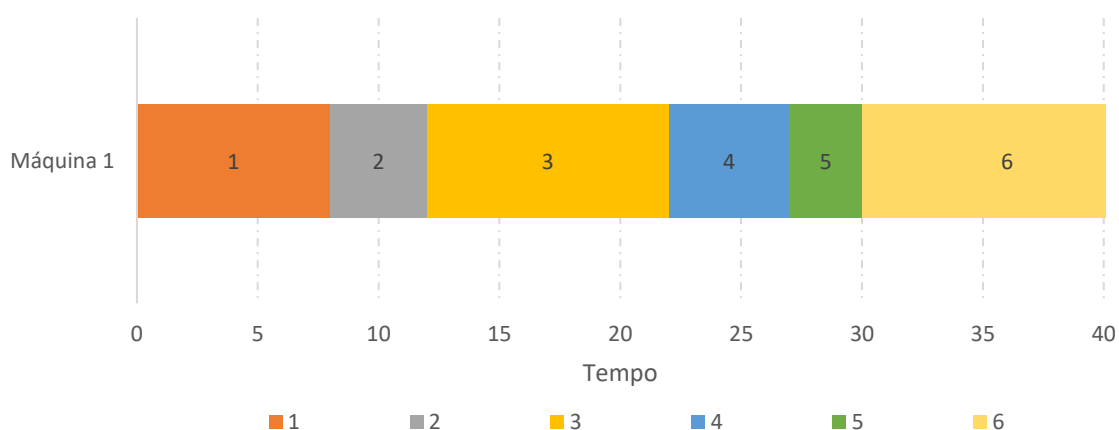
Fonte: Framinan, Leisten e García, 2014, pg 5

Tabela 1 - Tempos de processamentos de seis *jobs* em uma máquina para *single machine layout*

Máquina (<i>i</i>)	Job (<i>j</i>)					
	1	2	3	4	5	6
1	8	4	10	5	3	12

Fonte: Framinan, Leisten e García, 2014, pg 52

Figura 2 - Gráfico de Gantt com resultado do sequenciamento *single machine* com tempos de processamento da Tabela 1.



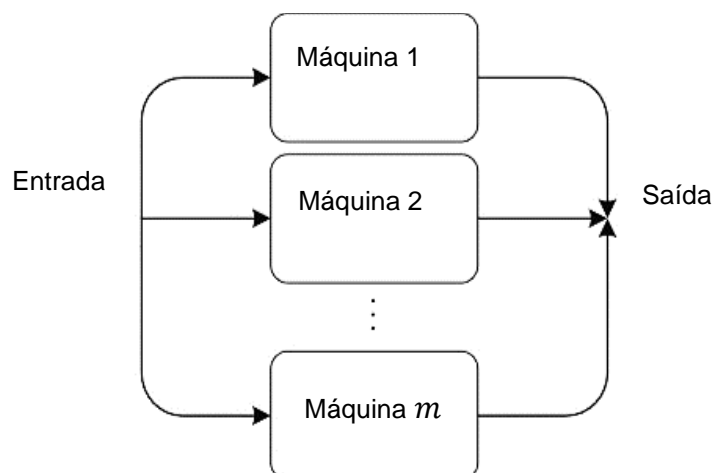
Fonte: Framinan, Leisten e García, 2014, pg 52

2.1.2 Parallel Machine Layout

Parallel Machine ou Máquinas paralelas trata-se da ampliação do modelo *Single Machine* como ilustrado na Figura 3. As máquinas m , que executam o mesmo processo, são dispostas paralelamente podendo ser utilizadas por qualquer um dos *jobs*.

Na Figura 4 é apresentada a representação gráfica para um problema com 3 máquinas e 8 *jobs* sequenciadas paralelamente. Com tempos de processamento presentes na Tabela 2, a Figura 4 deixa explícito o aumento de complexidade do arranjo paralelo ao compararmos o mesmo com o arranjo de máquina única.

Figura 3 - Fluxo de Parallel Machine Layout



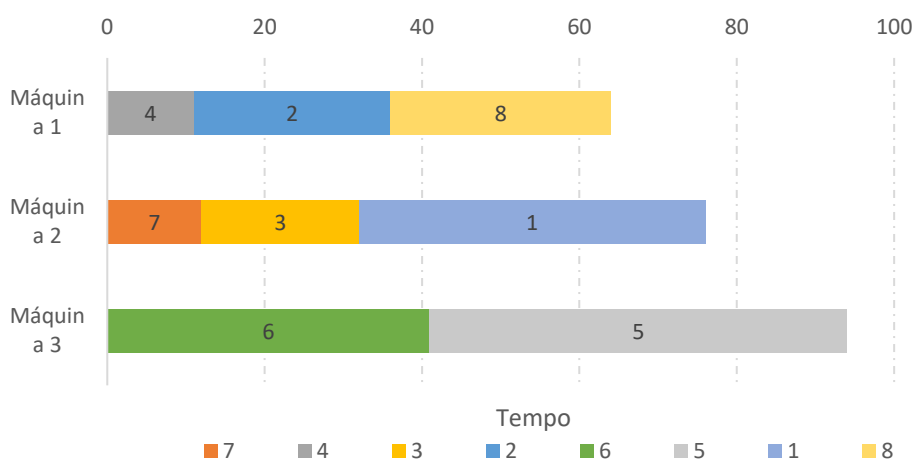
Fonte: Framinan, Leisten e García, 2014, pg 52

Tabela 2 - Tempos de processamento hipotéticos de 8 *jobs* em 3 máquinas em arranjo de máquinas paralelas.

Máquina (<i>i</i>)	<i>Job</i> (<i>j</i>)							
	1	2	3	4	5	6	7	8
Máquina 1	37	25	25	11	48	68	28	28
Máquina 2	44	42	20	27	56	59	12	39
Máquina 3	44	66	58	87	53	41	47	76

Fonte: Framinan, Leisten e García, 2014, pg 53

Figura 4 - Gráfico de Gantt com resultado do sequenciamento em arranjo de máquinas paralelas com tempos de processamento da Tabela 2

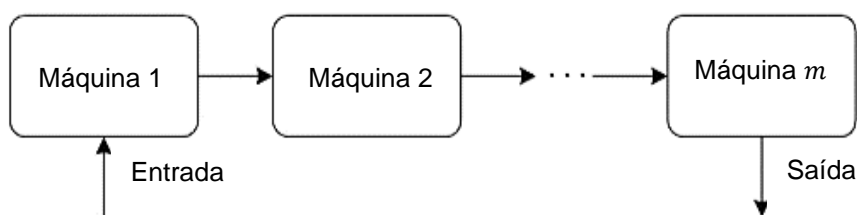


Fonte: Framinan, Leisten e García, 2014, pg 54

2.1.3 Flow Shop Layout

O *flow shop layout* apresenta diferentes máquinas dispostas de maneira seriada que executam *jobs* com a mesma sequência de processamento de máquinas, ilustrado na Figura 5. Assim todos os *jobs* passam pelas mesmas máquinas seguindo uma mesma sequência para sua conclusão, o que fica mais visível a partir do exemplo da Tabela 3 e Figura 6.

Figura 5 - Fluxo de Flow Shop Layout



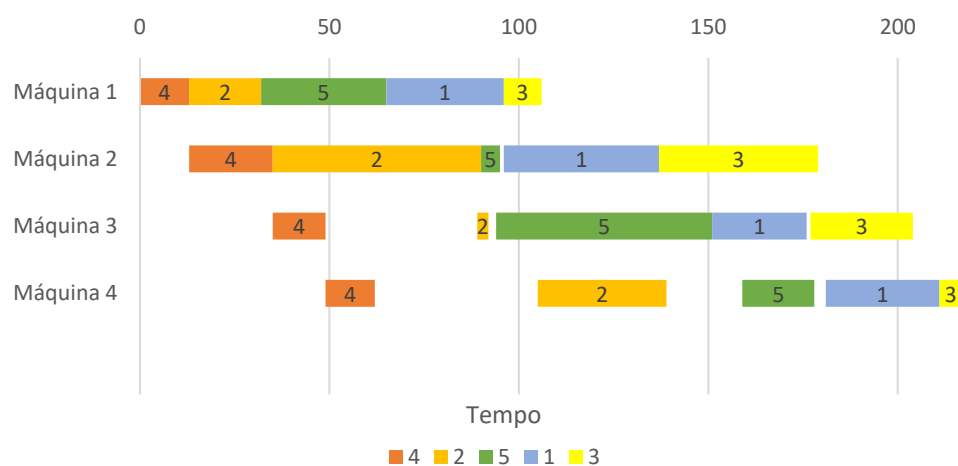
Fonte: Framinan, Leisten e García, 2014, pg 54

Tabela 3 - Tempos de processamentos hipotéticos de 5 *jobs* e 4 máquinas para *layout flow shop*

Máquina (<i>i</i>)	<i>Job</i> (<i>j</i>)				
	1	2	3	4	5
Máquina 1	31	19	10	13	33
Máquina 2	41	55	42	22	5
Máquina 3	25	3	27	14	57
Máquina 4	30	34	6	13	19

Fonte: Framinan, Leisten e García, 2014, pg 55

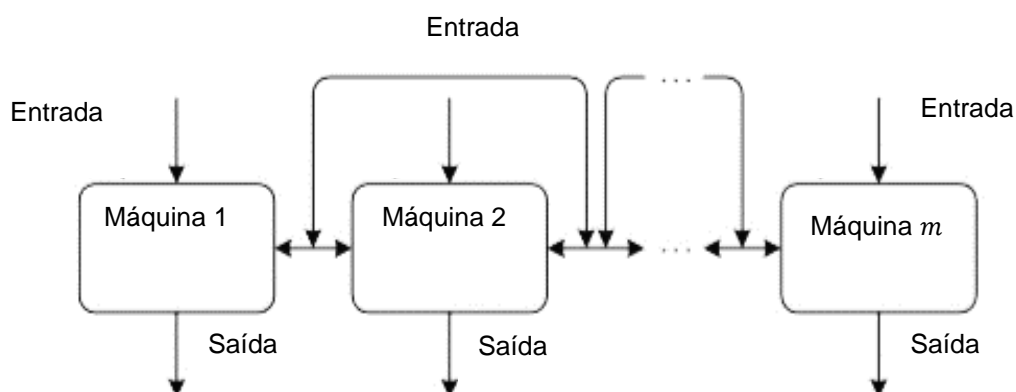
Figura 6 - Gráfico de Gantt com resultado do sequenciamento em *layout flow shop* com tempos de processamento da Tabela 3



2.1.4 Job Shop Layout

O *Job Shop* dispõe de máquinas diferentes que podem ser agrupadas de maneira seriada, por função ou por célula. Diferente do arranjo *flow shop*, neste arranjo os *jobs* realizados não apresentam necessariamente a mesma ordem de processamento, conforme ilustra a Figura 7. Dessa maneira, um *job* que inicia seu processo na máquina m_4 pode ter que voltar na sequência de máquina para seu segundo processamento que acontece na m_1 . Utilizando os tempos de processamento da Tabela 3 e o roteiro de processamento de cada *job* na Tabela 4, constrói-se um exemplo representativo do sequenciamento em *job shop* visto na Figura 8.

Figura 7 - Fluxo de *Job Shop Layout*



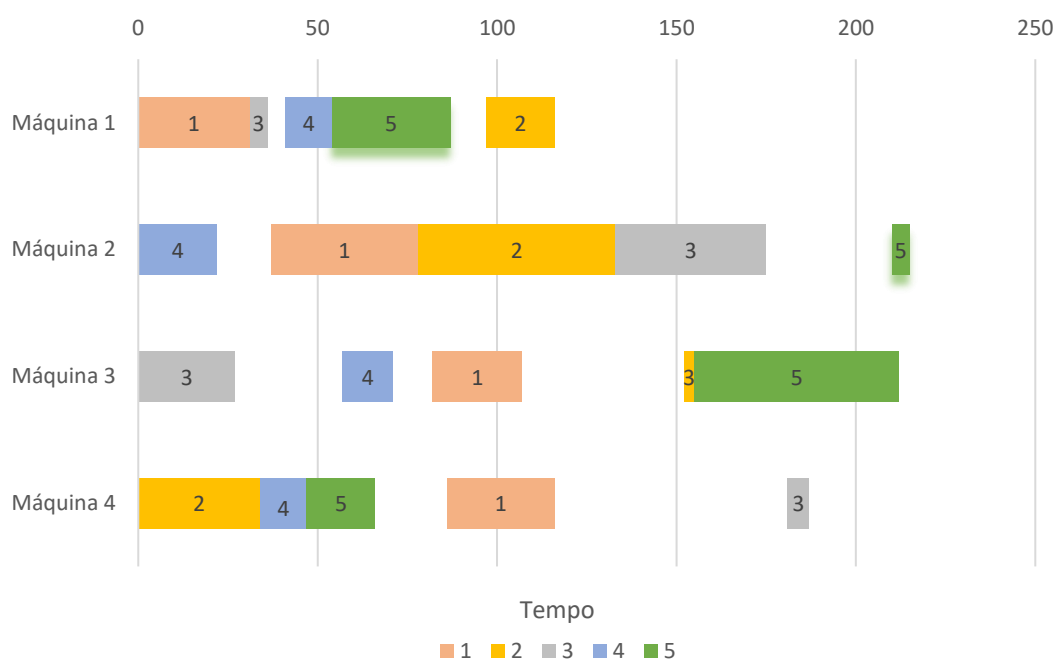
Fonte: Framinan, Leisten e García, 2014, pg 58

Tabela 4 - Rotas de processamento para 5 jobs no layout job shop

<i>(j)</i>	<i>Job</i>	R_j				
1	1	1	2	3	4	
2	2	4	2	1	3	
3	3	3	1	2	4	
4	4	2	4	3	1	
5	5	4	1	3	2	

Fonte: Framinan, Leisten e García, 2014, pg 58

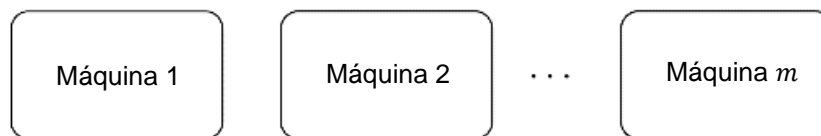
Figura 8 - Gráfico de Gantt com resultado do sequenciamento em *layout job shop* com tempos de processamento da Tabela 3 e rotas de processamento da Tabela 4.



Fonte: Framinan, Leisten e García, 2014, pg 59

2.1.5 Open Shop Layout

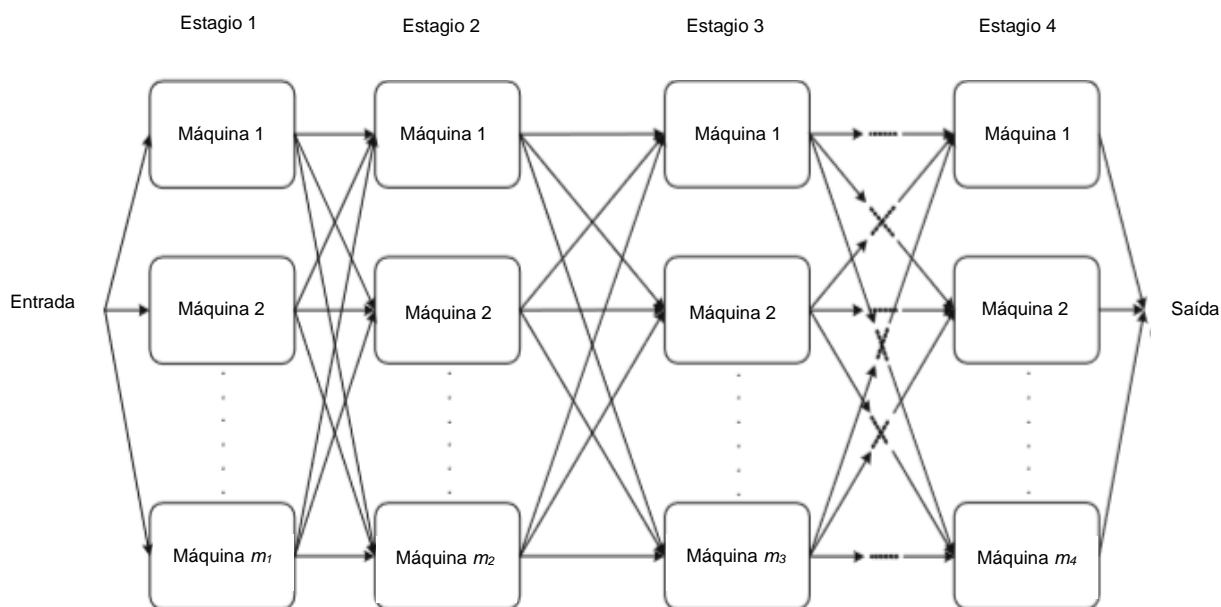
No arranjo *Open Shop*, ilustrado na Figura 9, os *jobs* não apresentam uma sequência ou roteiro específicos ou preestabelecidos para seus processamentos nas máquinas, o que o torna mais generalizado e complexo que os outros arranjos *flow* e *job shop*.

Figura 9 - Fluxo de *Open Shop Layout*

Fonte: Framinan, Leisten e García, 2014, pg 59

2.1.6 Hybrid Layouts

Os *layouts hybrid* ou flexíveis consistem em uma generalização dos *layouts* de *Flow Shop* e *Open Shop*, o qual pode ser visto na Figura 10. Trata-se realizar a disposição das máquinas para ambos os casos em paralelo, formando centros de trabalhos de máquinas equivalentes. Assim os *jobs* passam pelos centros de trabalho e podem ser executados por qualquer uma das máquinas dentro deles não apresentando de fato uma sequência fixa.

Figura 10 - Fluxo de *Hybrid Layout*

Fonte: Framinan, Leisten e García, 2014, pg 60

2.1.7 Regras de Sequenciamento

A resolução de problemas de sequenciamento ainda pode ser auxiliada por heurísticas simples baseadas em oito regras de sequenciamento (*scheduling rules*). A utilização destas regras permite uma solução mais rápida sem o uso de algoritmos matemáticos de alta complexidade que exijam processamentos computacionais (SULE, 2008). As oito regras definidas por Sule (2008) são:

- a) Regra EDD (*Earliest Due Date ou Menor Data de Entrega*): *Jobs* são sequenciados pela ordem crescente de seus prazos de entrega.
- b) Regra COVERT (*Cost Over Time ou Custo por Processamento*): A regra consiste no sequenciamento dos *jobs* por meio do cálculo do Coeficiente de Sequenciamento, abreviado por (CF_i), que é obtido em 3 etapas.
 - I. Para todos os trabalhos ainda não sequenciados é calculado um Valor de Prioridade ou (PR), correlacionando,

D_i : Prazo de entrega do *job* i

ST : Instante de início de processamento do *job* i .

P_i : Tempo total de processamento do *job* i

TT : Soma de todos os processos de processamento

L_i : Peso do *job* i

Para a atribuição do valor de priorização às tarefas ainda não sequenciadas é necessária a realização das análises condicionais a seguir:

CASO 1: Se $D_i < (ST + P_i)$, então $PR = 1$

CASO 2: Se $D_i > (ST + P_i)$ e $D_i < TT$, então $PR = \frac{TT - D_i}{Rt - P_i}$

CASO 3: $TT \leq D_i$, então $PR = 0$

- II. Após a definição dos valores de PR é realizado o calculado CF para cada um dos *jobs* por meio da Equação (1).

$$CF_i = \frac{PR \times L_i}{P_i} \quad (1)$$

III. Com os valores definidos o *job* que apresentar o maior coeficiente deve ser inserido na sequência, e consecutivamente os valores de *ST* e *RT* devem ser recalculados para que as etapas I, II e III sejam reiniciadas para o sequenciamento do próximo *job*.

c) Regra SPT (*Shortest Processing Time* ou Menor Tempo de Processamento): As tarefas são sequenciadas de acordo com seus tempos crescentes de processamento.

d) Regra LPUL (*Largest Penalty Per Unit Length* ou *Maior Penalidade Por Menor Tempo de Processamento*): Cada trabalho tem seu valor de U_i calculado usando a Equação (2). Após os cálculos as tarefas são ordenadas na ordem decrescentes dos valores U_i e, em caso de empate, o *job* com menor tempo de processamento será o próximo a ser processado.

$$U_i = \frac{P_i - D_i}{P_i} \quad (2)$$

e) Regra STP e LPUL: Consiste na mesma ordenação do método STP tendo como critério de desempate o menor índice de U_i .

f) Regra SWPT (*Shortest Weighted Processing Time* ou *Menor tempo de processamento ponderado*): Realiza-se o cálculo do indicador S_i usando a Equação (3). Após isso, o sequenciamento baseia-se na ordem crescente dos índices obtidos.

$$S_i = \frac{P_i}{P_i - D_i} \quad (3)$$

- g) Regra WT e LPUL (*Largest Weight and Largest Penalty Per Unit Length*): O sequenciamento se dá pela ordem decrescente dos pesos das tarefas, tendo como critério de desempate o menor índice de U_i .
- h) CR (*Critical Ratio ou Razão Crítica*): Tarefas são sequenciadas na ordem crescente de suas razões críticas que são calculadas pela Equação (4):

$$CR_i = \frac{D_i - TT}{P_i} \quad (4)$$

Pinedo (2016) define diversas funções objetivo, contudo argumenta que o objetivo a ser minimizado sempre será uma função do tempo de conclusão dos *jobs*, o qual é intrínseco ao sequenciamento. Segundo o autor, os principais objetivos a serem otimizados em problemas de sequenciamento de produção são:

- *Makespan*: Trata-se da minimização do tempo de conclusão do último *job* no sistema, potencializando a utilização das máquinas. Em outras palavras o *makespan* diz respeito à minimização do tempo total para a conclusão dos *jobs*.
- Tempo total de Finalização (*Total Completion Time*): Minimiza a soma dos tempos necessários para finalizar todos os *jobs*.
- Atraso máximo (*Maximum Lateness*): Esta função, ilustrada na Equação 5, tem por objetivo busca minimizar o maior atraso em termos de data de entrega do *job*, considerando a diferença entre o tempo real e o tempo de processamento planejado.

$$T_i = \max(0, (ST + P_i) - D_i). \quad (5)$$

- Atraso máximo (*Maximum Tardiness*): Assim como a função objetivo mencionada anteriormente o maximum tardiness visa minimizar o maior atraso entre os *jobs* com relação a data de entrega. A diferença entre a minimização dos dois métodos consiste que no caso do

maximum lateness, caso uma tarefa seja finalizada antes de sua data de entrega o valor da antecipação entra como tempo negativo na somatória de tempos enquanto no maximin tardiness o valor seria zero, ou seja temos apenas a mudança na interpretação final do valor resultante da Equação 5.

- Número de *Jobs* em Atraso (*The Number of Tardy Jobs*): A função realiza a minimização do número de *jobs* que ficarão em atraso.
- Atraso Ponderado (*Total Weighted Tardiness (TWT)*): Consiste minimização do atraso ponderado de um conjunto de *jobs*, por meio da soma do tempo ponderado de conclusão das tarefas determinado pela Equação 5, onde T_i representa o atraso ilustrado na Equação 5.

$$TWT = \sum_i w_i T_i \quad (6)$$

2.2 FLOW SHOP NÃO PERMUTACIONAL

Como citado na seção 2.1 os problemas de sequenciamento em *layout flow shop* (NPFS) consistem no agrupamento de diferentes máquinas (m), que executam os *jobs* (j) de maneira seriada, ou seja, todo *job* passa pela mesma sequência de máquinas durante seu processo de execução. Este tipo de *layout* ainda assume os seguintes requisitos (PINEDO, 2016):

- Cada máquina pode processar apenas um *job* por vez
- Cada *job* $j = 1, 2, 3 \dots n$ deve ter apenas um tempo de processamento p_{ij} por máquina $i = 1, 2, 3 \dots m$.
- Não há tempo de transportes entre cada uma das máquinas. Caso haja esse tempo no problema real, ele pode ser incorporado ao tempo de processamento do *job*.
- Os buffers intermediários devem ter uma capacidade de armazenamento grande o suficiente para que as tarefas sigam suas sequências sem paradas.

Considerando que o FSSP atenda aos requisitos considerados por Pinedo e tenha como função objetivo a minimização do *makespan* (C_{max}), que se trata da função mais utilizada neste tipo de arranjo, é possível estruturar um programa linear

(PL) genérico para o caso. Arenales et al. (2007), desenvolveu uma formulação de PL que tem por objetivo minimizar o *makespan* de FSSP, como mostra Equação 7, assim admitindo s_{kj} como o tempo necessário para iniciar a tarefa k na máquina j e z_{ij} uma variável $\in \{0,1\}$ que apresenta resposta 1 se a tarefa i é designada a posição j e 0 em caso negativo temos:

$$\min C_{max} = x_{mn} + \sum_{i=1}^n p_{im}z_{in} \quad (7)$$

$$\sum_{j=1}^n z_{ij} = 1, \quad i = 1, \dots, n \quad (8)$$

$$\sum_{i=1}^n z_{ij} = 1, \quad j = 1, \dots, n \quad (9)$$

$$s_{1j} + \sum_{i=1}^n p_{i1}z_{ij} = s_{1,j+1}, \quad j = 1, \dots, n-1, \quad (10)$$

$$s_{11} = 0 \quad (11)$$

$$s_{k1} + \sum_{i=1}^n p_{ik}z_{i1} = s_{k+1,1}, \quad k = 1, \dots, m-1 \quad (12)$$

$$s_{kj} + \sum_{i=1}^n p_{ik}z_{ij} \leq s_{k+1,j}, \quad j = 2, \dots, n, k = 1, \dots, m-1, \quad (13)$$

$$s_{kj} + \sum_{i=1}^n p_{ik}z_{ij} \leq s_{k,j+1}, \quad j = 1, \dots, n-1, k = 2, \dots, m, \quad (14)$$

$$s \in R_+^{nm}, z \in B^{nn} \quad (15)$$

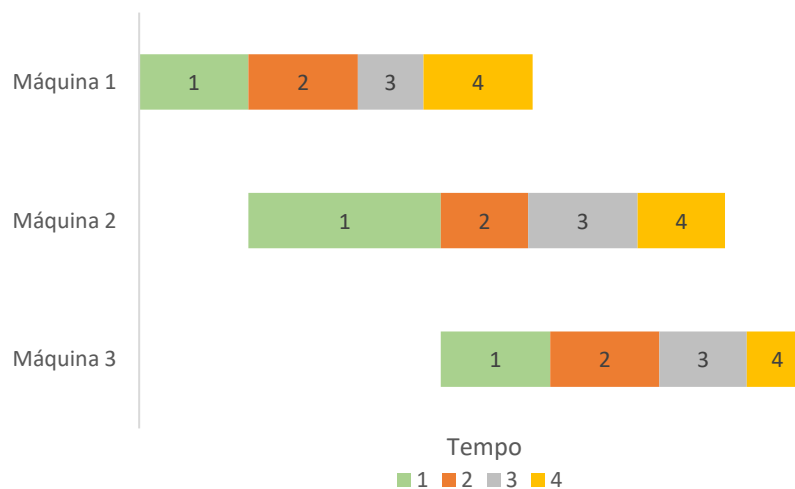
As restrições presentes na Equações 8 e 9, garantem respectivamente que cada tarefa i está associada a uma única posição, e que cada posição j está associada a uma única tarefa. A restrição ilustrada na Equação 10, força iniciação do processamento da tarefa na posição j na máquina 1 após sua tarefa predecessora ter sido totalmente processada na máquina. Na restrição subsequente, Equação 11, fica estabelecido que a primeira tarefa da sequência comece seu processamento na máquina 1 no instante 0. A Equação 12, visa garantir que a primeira tarefa na sequencia seja processada imediatamente na próxima máquina $k+1$, com a condição de que o processamento na máquina k já tenha sido finalizado. As Equações 13 e 14 formam um laço logico, no qual a primeira garante que a uma tarefa na posição j não possa ser iniciada na máquina $k+1$ até que seu processo vigente na máquina k seja

finalizado, enquanto a segunda assegura que a tarefa na posição $j + 1$ não seja iniciada em uma máquina antes que seu processamento de sua tarefa predecessora j seja finalizado na máquina k . Por fim o programa linear formulado por Arenales et al. (2007), ilustra na Equação 15 uma heurística de minimização de makespen proposta por Moccelin e Santos (2000) que indica o tipo de variáveis utilizada.

Independente da função objetivo escolhida, a determinação de soluções factíveis em *layouts flow shop* está atrelado ao tipo de permutação dos n jobs que precisam ser sequenciados (BAKER, 2009). A permutação dos jobs pode ocorrer de duas maneiras neste arranjo, a primeira seria no modelo de *Permutation Flow Shop* (PFS) no qual a os jobs são permutacionados em $n!$, o que faz com que as tarefas sejam executadas em todas as máquinas na mesma sequência. O segundo modelo, conhecido como *Non - Permutation Flow Shop* (NPFS) os n jobs podem ser permutacionados em até $(n!)^m$, onde m representa o número de máquinas, o que permite que cada máquina possui uma permutação própria (TANDON, 1991).

No arranjo PFS as permutações ocorrem considerando o job por completo, ou seja, a trajetória do job pelas m máquinas, o que faz com que o primeiro trabalho inserido na máquina 1 também seja o primeiro em todas as demais máquinas m , como ilustrado na Figura 11. Já no caso NPFS os jobs podem ser alterados individualmente por processo de máquina, ilustrado na mostra a Figura 12, onde o job 1 e 3 foram

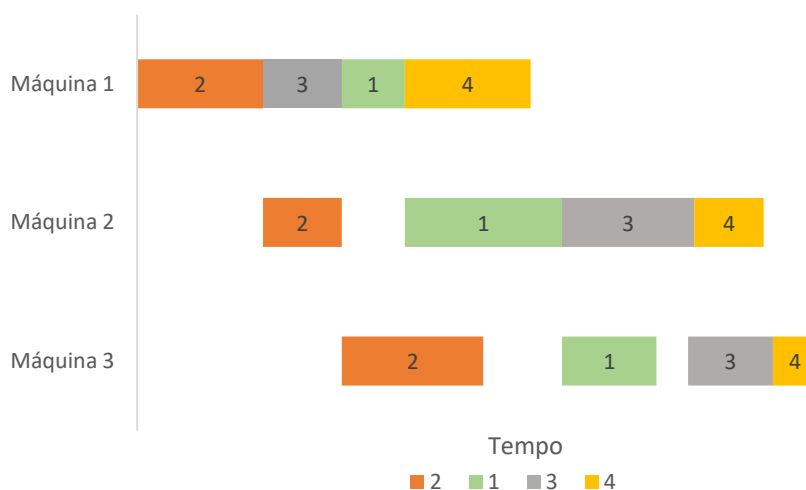
Figura 11 - Representação do gráfico de Gantt para um problema de 3 máquinas e 4 jobs em *layout Flow Shop* Permutaciona



Fonte: Framinan, Leisten e García, 2014, pg 57

alterados apenas no processamento da máquina 1, o que torna a resolução destes problemas mais complexas.

Figura 12 - Representação do gráfico de Gantt para um problema de 3 máquinas e 4 jobs em *layout Flow Shop* Não Permutaciona



Fonte: Framinan, Leisten e García, 2014, pg 57

A problematização a respeito da utilização do PFS e NPFS tem tornando-se um tema recorrente de alguns autores nos últimos anos a partir das comparações de eficiência entre os dois modelos de permutação. As comparações realizadas pelos autores mesmo com funções objetivos diferentes demonstram constantemente que a modelagem a partir do NPFS é mais eficiente do que a modelagem em PFS. Xiao (2015) processou cerca de 19.000 instâncias na busca de seu ótimo global, dentre a melhor solução factível encontrada, para um problema de sequenciamento multicritério, e em cerca de 30% dos casos o NPFS apresentou-se mais eficiente, potencialmente nas instâncias com maior número de trabalhos e máquinas. Liao (2006) realizou a comparação entre os modelos em um problema de minimização do total de atrasos e concluiu a perda de 10% de eficiência quando utilizou PFS. Quatro anos depois com instâncias de valores e tamanhos diferentes sob a mesma função objetivo Liao (2006) confirmou mais uma vez agora com uma porcentagem de cerca de 8% que o NPFS é mais eficiente que o PFS. A baixo na Tabela 5 estão 12 artigos

que descrevem a comparação de eficiência e eficácia de PFS e NPFS assim como artigos que tratam somente de NPFS ambos para diversas funções objetivos.

Tabela 5 - Artigos publicados envolvendo NPFS e comparações de NPFS e PFS

AUTOR	PROBLEMA	TITULO	OBJETIVO	HEURISTICA	Ano
Ye, H.	NPFS	<i>An improved heuristic for no-wait flow shop to minimize makespan</i>	minimize makespan	AIT	2017
Rossit , D. A.	NPFS/PFS	<i>The Non-Permutation Flow-Shop scheduling problem: A literature</i>	minimizing total tardiness	Tabu Search	2017
L.-M. Liao, C.-J. Huang	NPFS	<i>Tabu search for non-permutation flowshop scheduling problem with minimizing total tardiness</i>	minimizing total tardiness	Tabu Search	2017
B. Vahedi-Nouri	NPFS	<i>Minimizing total flow time for the non-permutation flow shop scheduling problem with learning effects and availability constraints</i>	makespan and total completion time	VRF	2013
Y. Xiao	NPFS	<i>Non-permutation flow shop scheduling with order acceptance and weighted tardiness</i>	weighted tardiness	GA	2015
Y. Mehravaran, R. Logendran	NPFS	<i>Non-permutation flowshop scheduling with dual resources</i>	minimize weighted	TSEPP	2013
Y. Mehravaran, R. Logendran	NPFS	<i>Non-permutation flowshop scheduling in a supply chain with sequence-dependent setup times</i>	minimize the work-in-process	TSEPP	2012
S. Pugazhendhi	NPFS	<i>Performance enhancement by using non-permutation schedules in flowline-based manufacturing systems</i>	minimization of makespan	NEH	2002
Kuo-Ching Ying	NPFS	<i>Solving non-permutation flowshop scheduling problems by an effective iterated greedy heuristic</i>	minimization of makespan	IG heuristic	2008
H. Ye	NPFS/PFS	<i>An effective heuristic for no-wait flow shop production to minimize</i>	minimization of	R, RAJ and NEH	2016
C. J. Liao	NPFS/PFS	<i>A performance evaluation of permutation vs. non-permutation schedules in a flowshop</i>	minimizing total tardiness	TS	2006
MatheusdeFreitasAra	NPFS/PFS	<i>MétodoProximitySearchpara a resolução do problema de flowshop scheduling não permutacional com trabalhos heterogêneos (FSNPTH)</i>	makespan	AG	2016
Alexander J. Benavide	NPFS/PFS	<i>Two simple and effective heuristics for minimizing the makespan in non-permutation flow shops</i>	makespan	LS	2015
M. TANDON	FLOWSHOP WITH NON-PERMUTATION SCHED		makespan	AS	1991

A relação entre o desempenho do NPFS e PFS está condicionada à diferença entre o tempo máximo e mínimo de processamento de um grupo de *jobs*. Tandon (1991) relata que quanto maior diferença entre os tempos de processamento melhor é o desempenho de uma solução *non-permutation* frente a uma solução *permutation*. O autor ainda presume que um problema *flow shop* para uma solução *non-permutation* apresenta pelo menos uma solução ótima para um problema de sequenciamento utilizando essa propriedade de que as duas primeiras máquinas apresentem o mesmo sequenciamento assim como as duas últimas. O uso desta ainda reduz o espaço de busca de $(n!)^2$ para $(n!)^{m-2}$ (CONWAY et al., 1967).

Benavides (2016) em trabalho estruturou um PL para o NPFS, onde admitindo x_{ij} como o tempo necessário para iniciar a tarefa j na máquina i e $y_{ijj'}$, uma variável $\in \{0,1\}$ que indica se o *job* j precede o *job* j' na máquina m_i , temos:

$$\min C_{max} \quad (16)$$

$$\text{s. t } x_{ij} + p_{ij} \leq C_{max}, \quad \forall j \in [n], \quad (17)$$

$$x_{ij} + p_{ij} \leq x_{i+1j}, \quad \forall i \in [m-1], \forall j \in [n], \quad (18)$$

$$x_{ij} + p_{ij} \leq x_{ij'} + M(1 - y_{ijj'}), \quad \forall i \in [m], j \neq j' \in [n], \quad (19)$$

$$y_{ijj'} + y_{ij'j} = 1, \quad \forall i \in [m], j \neq j' \in [n], \quad (20)$$

$$x_{ij} > 0, \quad \forall i \in [m], j \in [n], \quad (21)$$

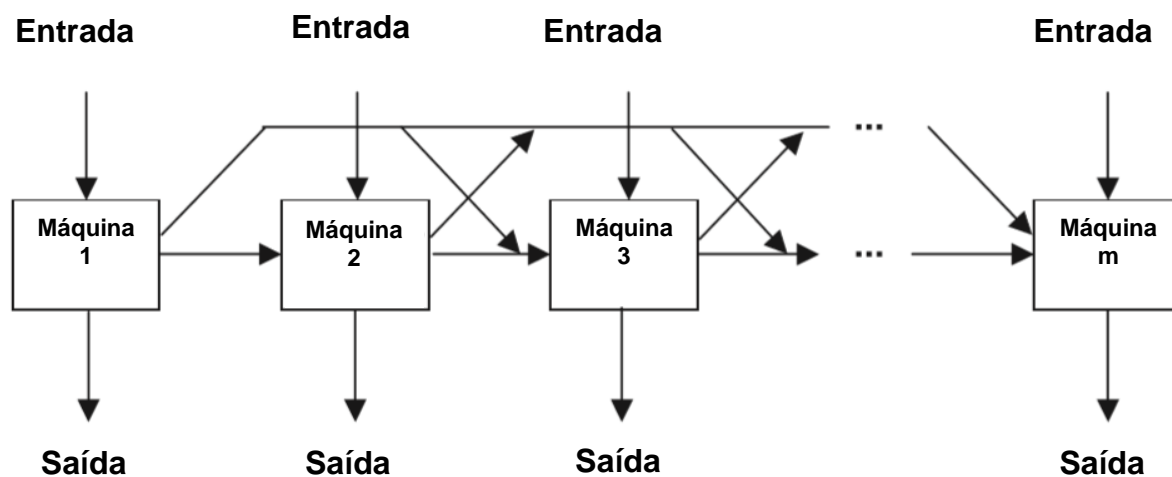
$$y_{ijj'} \in \{0,1\}, \quad \forall i \in [m], j \neq j' \in [n], \quad (22)$$

A estruturação do PL para o NPFS processa quatro restrições. A primeira restrição, ilustrada na Equação 7, consiste na definição do *makespan* como sendo menor que a soma entre o tempo de processamento e o tempo de início da tarefa j na máquina i . A segunda restrição, descrita na Equação 8, requer que nenhum trabalho seja iniciado antes que o anterior tenha terminado. A Equação 9, trata-se da restrição que garante que os trabalhos sejam processados na ordem definida pela variável y . E por fim a última restrição descrita na Equação 10, força uma ordem linear dos trabalhos ao longo de todas as máquinas.

Uma outra variante do modelo de *flow shop* de permutação é a chamada linha de fluxo na qual todos os trabalhos seguem a mesma rota relativa visitando as máquinas, mas algumas máquinas podem ser ignoradas por alguns trabalhos, como ilustrado na Figura 13. As máquinas ignoradas podem ser vistas como processamento

opcional de tratamentos que não são necessários para todos os trabalhos (FRAMINAN; LEISTEN; GARCÍA, 2014).

Figura 13 - Layout Flow Shop com variação Flow Line

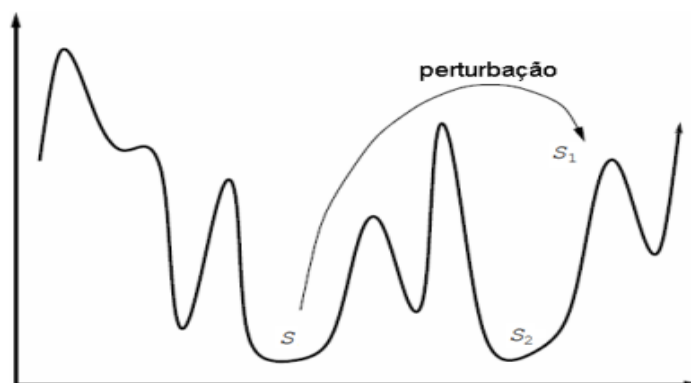


Fonte: Framinan, Leisten e García, 2014, pg 57

2.3 ITERATED LOCAL SEARCH (ILS)

De acordo com Lourenço, Martin e Stützle (2002) nos últimos anos o *Iterated Local Search* ou Busca Local Iterada (ILS) tem se mostrado como uma metaheurística promissora para diferentes problemas de natureza combinatória. Desenvolvida em

Figura 14 – Como o ILS evita estagnação em ótimos locais por perturbação.



Fonte: Adaptado Lourenço, Martin e Stützle 2002

torno da Busca Local (BL), os autores definem o ILS como sendo um método de busca iterativa que gera novas soluções, por intermédio de perturbações na solução ótima

local, retirando assim as soluções encontradas de pontos de estagnação, como ilustrado na Figura 14.

Inicialmente o ILS realiza um procedimento básico de busca local na busca da melhor solução factível. Esta passa por uma série de perturbações em sua vizinhança de modo a ser observado um pequeno conjunto de soluções e não todo o espaço de soluções. Assim por intermédio de critérios de aceitação e perturbação uma nova solução de ótimo local é gerada permitindo que o ciclo continue até seus parâmetros de parada (BRITO, MONTENEGRO, e OCHI, 2009).

O pseudocódigo apresentado por Lourenço, Martin e Stützle (2002) do algoritmo ILS, ilustrado na Figura 15, possui quatro etapas principais sendo essas: solução inicial, busca local, perturbação e critério de aceitação. A solução inicial consiste na obtenção de uma solução S_1 . Na sequência, a segunda etapa realizada a busca local usando a solução inicial S_1 , gerando uma nova solução S_2 a qual é um ótimo local. Para permitir que uma nova região do espaço de soluções próxima a S_2 seja pesquisado, na terceira etapa é aplica a perturbação na solução S_2 , sendo por fim realizada na quarta etapa a validação do critério de aceitação da solução atual encontrada, ou seja, caso a solução não se enquadre no critério de aceitação exceto a primeira etapa todas as demais devem ser executadas iterativamente até que o critério de aceitação seja obtido, e a solução passe a ser uma solução candidata para a próxima iteração.

Figura 15 - Pseudocódigo ILS

```

Procedimento ILS (CritérioParada)
1    $S_1 \leftarrow$  Construção_Solução_Inicial;
2    $S \leftarrow S_1$ ; //Melhor solução encontrada
3   enquanto não CritérioParada faça
4        $S_2 \leftarrow$  Busca_Local( $S_1$ );
5        $S_2 \leftarrow$  Perturbação( $S_2$ ,  $d$ );
6       Critério_Aceitação( $S^*$ ,  $S_1$ ,  $S_2$ );
7   fim-enquanto;
8   retorne  $S$ ;
fim ILS;

```

Fonte: Lourenço, Martin e Stützle 2002

O critério de parada utilizado dentro da metaheurística é um dos fatores muito importante para a boa execução da mesma, visto que sua escolha controla o balanço

entre a intensificação e a diversidade dos ótimos locais. De modo geral o critério de parada é um dado número de iterações (N), que se limitam a um valor imposto pelo programador determinando assim o número máximo de iterações consecutivas que a solução pode permanecer sem melhorias antes de ser descartada e definindo como o melhor ótimo global a última melhor solução factível encontrada (MELO, 2014). Outros fatores que podem ser utilizados como critério de parada são a quantidade de elementos perturbados e o tempo máximo de busca, sendo em ambos os casos escolhida a melhor última melhor solução factível obtida (KAMPKE, 2009).

3. METODO DE PESQUISA

3.1. ESTRUTURA GERAL DO ALGORITMO (ILS)

Como já mencionado anteriormente a metaheurística utilizada no processamento de instâncias no presente trabalho será o ILS. O algoritmo como mostrado anteriormente na Figura 15, consiste na realização de iterações partindo de uma solução inicial que posteriormente passará pela busca local e por fim sendo perturbada até que o critério de parada seja atendido e gerando ótimos locais para as iterações e o ótimo global da solução.

Na aba identificada como “PRINCIPAL”, Figura 16, a instância originária de um arquivo de texto txt é transferido para o arquivo Excel, identificando assim o número de trabalhos, máquinas, tempos de processamento *job* x máquina e datas de entrega de cada um dos trabalhos em questão. Após os dados serem carregados o usuário define na aba “ILS” os parâmetros que deseja para execução, tais como critério de parada, o tipo de perturbação a ser utilizada e a quantidade de perturbações a serem feitas. Para a realização da parametrização é necessário sejam respectivamente preenchidos os campos “TEMPO DE PROCESSAMENTO”, “TIPO DE PERTURBAÇÃO” e “QTD DE PERTURBAÇÕES” como ilustrado na Figura 17.

Figura 16 - Aba de Processamento PRINCIPAL

Jobs	Maq1	Maq2	Maq3	Maq4	Maq5	Prazo
1	0	20	0	8	49	146
2	42	9	28	55	39	260
3	19	35	0	38	9	195
4	0	42	21	11	33	187
5	23	0	44	9	19	174
6	19	53	12	44	19	265
7	0	19	29	49	33	188
8	8	10	0	28	44	166
9	17	54	51	40	9	217
10	51	0	0	47	33	259

Fonte: Do Autor, 2018

Figura 17 - Aba de Processamento ILS

	A	B	C	D	E	F	G	H	I	J
1										
2	ILS									
3				Melhor Global	7623					
4	Iteração	Inicial	Final	ContPert	Global			TEMPO DE PROCESSAMENTO	600	
5								TIPO DE PERTURBAÇA	1	
6	1	11522	7623	3	7623			QTD DE PERTURBAÇÕES	5	
7	2	13709	8336	4	7623			TEMPO REAL DE EXECUÇÃO	671,859375	
8	3	10891	7747	4	7623					
9	4	13142	7832	4	7623					
10	5	13218	7694	2	7623					
11	6	11132	7809	4	7623					
12	7	14432	7869	2	7623					
13	8	13265	7649	2	7623					
14										
15										

Fonte: Do Autor, 2018

Após a definição dos parâmetros o ILS inicia o processamento da instância pela criação de uma solução inicial aleatória, que consiste em um algoritmo que sequênciamos de forma aleatória os *jobs* da instância em processamento. Após a criação da sequência na primeira máquina a mesma é replicada para todas as demais, como ilustrado no pseudocódigo da Figura 18, gerando assim uma matriz aleatória de *jobs* x máquina como ilustrado pela Figura 19.

Figura 18 - Pseudocódigo Solução Inicial Aleatória

```

Procedimento SOLUÇÃO ALEATÓRIA (Instância)
1  MAQ, JOB ← Atribuindo Valor as Variáveis
2  matA, count, numA ← Declarando Variável
3
4  Gerando uma sequencia aleatória primaria
5  De i = 1 até (JOB)
6    Faça
7      Randomize
8      count = 0
9      numA ← Atribuindo Número Aleatorio Que Pode Variar de 1 até JOB
10     De k = 1 até JOB
11       Se matA (k,1) = numA faça
12         count = 1
13     Fim
14     Proximo k
15     Enquanto count >= 1
16       matA (i,1) = numA
17     Proximo i
18
19     Replicando sequencia primaria aleatória
20     De i = 1 até JOB
21       De j = 1 até MAQ
22         matA (i,j) = matA(i,1)
23       Proximo j
24     Proximo i

```

Fonte: Do Autor, 2018

Figura 19 - Representação da Matriz Aleatória Gerada Pelo Pseudocódigo
Solução Inicial Aleatória

Primeira Sequencia Aleatória	Matriz Aleatória <i>Jobs</i> x Máquina		
	M1	M2	M3
1	1	1	1
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
8	8	8	8
9	9	9	9
10	10	10	10
2	2	2	2
7	7	7	7

Fonte: Do Autor, 2018

Os procedimentos de vizinhança e perturbação acontecem de forma sequenciada tendo seu número de repetições atribuído pelo usuário no campo “QTD DE PERTURBAÇÕES”. Desta maneira após a criação da solução inicial aleatória, inicia-se o procedimento de busca local utilizando uma estrutura de vizinhança, o que permite atingir o ótimo local da solução inicial. Após atingir o ótimo local ocorre uma perturbação, também escolhida pelo usuário no campo “TIPO DE PERTURBAÇÃO”, sendo a busca local reiniciada usando a solução perturbada. As repetições de vizinhanças e perturbações são realizadas até que a quantidade solicitada pelo usuário, no campo “QTD PERTURBAÇÕES”, seja atendida obtendo o ótimo local da iteração.

O critério de parada do ILS consiste em um determinado tempo de processamento, sendo este por sua vez definido pelo usuário no campo “TEMPO DE PROCESSAMENTO”, permitindo assim a inicialização de uma nova iteração contanto que o tempo de processamento atual seja menor que o tempo imputado pelo usuário, como ilustra a Figura 20 com o pseudocódigo estrutural do ILS. Entretanto, para o processamento das 96 instâncias presentes foi padronizado o tempo de 600 segundos, não importando o número de trabalhos ou quantidade de máquinas em análise.

Figura 20 – Pseudocódigo Solver ILS

```

Procedimento Solver ILS (Instância)
1  MAQ, JOB, qtdPert, tempoFinal ← Atribuindo Valor as Variáveis da Aba Principal
2  count, it ← Declarando Variável de Controle
3  matGlobal, matLocal, matMelhorPert ← Declarando matriz de posições dos jobs
4  sumGlobal, sumLocal, sumMelhorPert, sumPert ← Declarando Variáveis de Soma de Atraso Total
5  tempolnicial ← Atribuindo Valor Inicial do Timer Excel
6
7  it = 1  'Variavel de contagem de iterações
8  matGlobal ← Atribui um valor muito alto tal como 10^8
9
10 Faça (Timer - tempolnicial) < TempoFinal
11   matAnalise ← Atribui a Matriz Aleatória
12   sumAnalise ← Atribui o valor do atraso total da matAnalise
13   countPert = 0
14
15   count = 1
16   De j = 1 até qtdPert
17     matLocal ← Atribui a melhor vizinhança apartir da matAnalise
18     sumLocal ← Atribui o valor do atraso total da matLocal
19     sumMelhorPert ← Atribui um valor muito alto tal como 10^8
20
21     Se sumLocal < sumMelhorPert
22       matMelhorPert = matLocal
23       sumMelhorPert = sumLocal
24       contPert = j
25     Fim
26   matAnalise ← Perturba a matMelhorPert de acordo com a perturbação solicitada
27   Proximo j
28
29   Se sumMelhorPert < sumGlobal
30     sumGlobal = sumMelhorPert
31     matGlobal = matMelhorPert
32   Fim
33
34   it = it +1
35 Fim
36
37 ESCREVE O RELATORIO DE ITERAÇÕES NA ABA ILS
39 ESCREVE A MELHOR SOLUÇÃO E OS ATRASOS POR MÁQUINA NA ABA SOLUÇÃO

```

Fonte: Do Autor, 2018

Ao final da execução da metaheurística, cada uma das iterações com seus respectivos valores de atraso inicial, atraso final, melhor perturbação e atraso global são impressos na aba “ILS”. Posteriormente na aba “SOLUÇÃO”, ilustrado na Figura 21, são impressos o atraso máximo total por *job*, seu prazo de entrega e posteriormente o valor de seu atraso máximo individual. Na sequência a matriz que

contém a melhor solução encontrada para instância também é impressa na mesma aba de forma matricial apresentando a posição de cada um dos *jobs* por máquina.

Figura 21 - Aba de Processamento SOLUÇÃO

	A	B	C	D	E	F	G	H	I	J	K
1	Job	Termino	Prazo	Atraso		Solução					
2	1	77	146	0		1	1	1	1	1	
3	2	121	166	0		8	8	8	8	8	
4	3	160	188	0		7	7	7	7	7	
5	4	179	174	5		5	5	5	5	5	
6	5	188	195	0		3	3	3	3	3	
7	6	221	187	34		4	4	10	4	4	
8	7	265	259	6		10	10	4	10	10	
9	8	281	217	64		9	9	9	9	9	
10	9	335	265	70		6	6	6	6	6	
11	10	410	260	150		2	2	2	2	2	
12											
13											

PRINCIPAL | SOLUÇÃO ALEATORIA | ILS | **SOLUÇÃO** | TESTE | + | : | < |

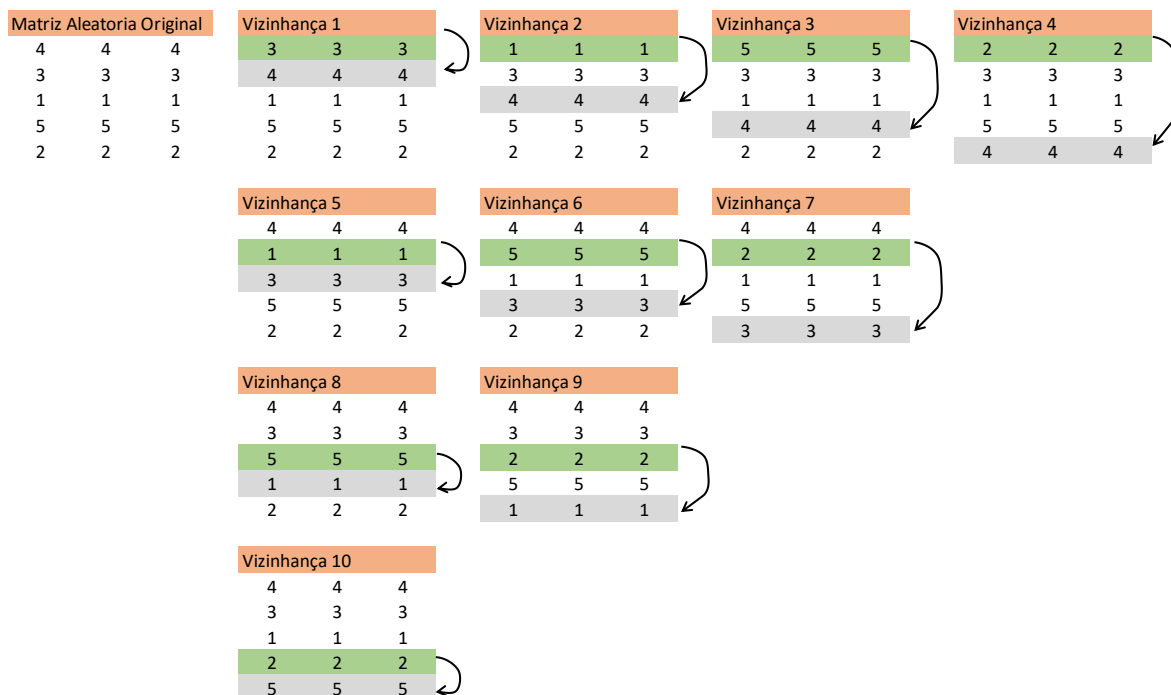
Fonte: Do Autor, 2018

3.2 ESTRUTURA DE VIZINHANÇA

A estrutura de vizinhança desenvolvida para execução do ILS, tem sua execução realizada em duas etapas, sendo a primeira na esfera macro da matriz que posteriormente é aprofundada em uma esfera micro. Denominadas como Vizinhança 1 e Vizinhança 2, a estrutura de ambas consistem em trocas simples de posição dentro de uma matriz que buscam sempre o menor atraso máximo da mesma.

A primeira vizinhança realizada pelo algoritmo trata-se da que contempla a esfera macro da matriz, consistindo na troca de cada uma das linhas com as demais, conforme ilustra a Figura 22. Dentre todas as soluções vizinhas geradas, a melhor é selecionada.

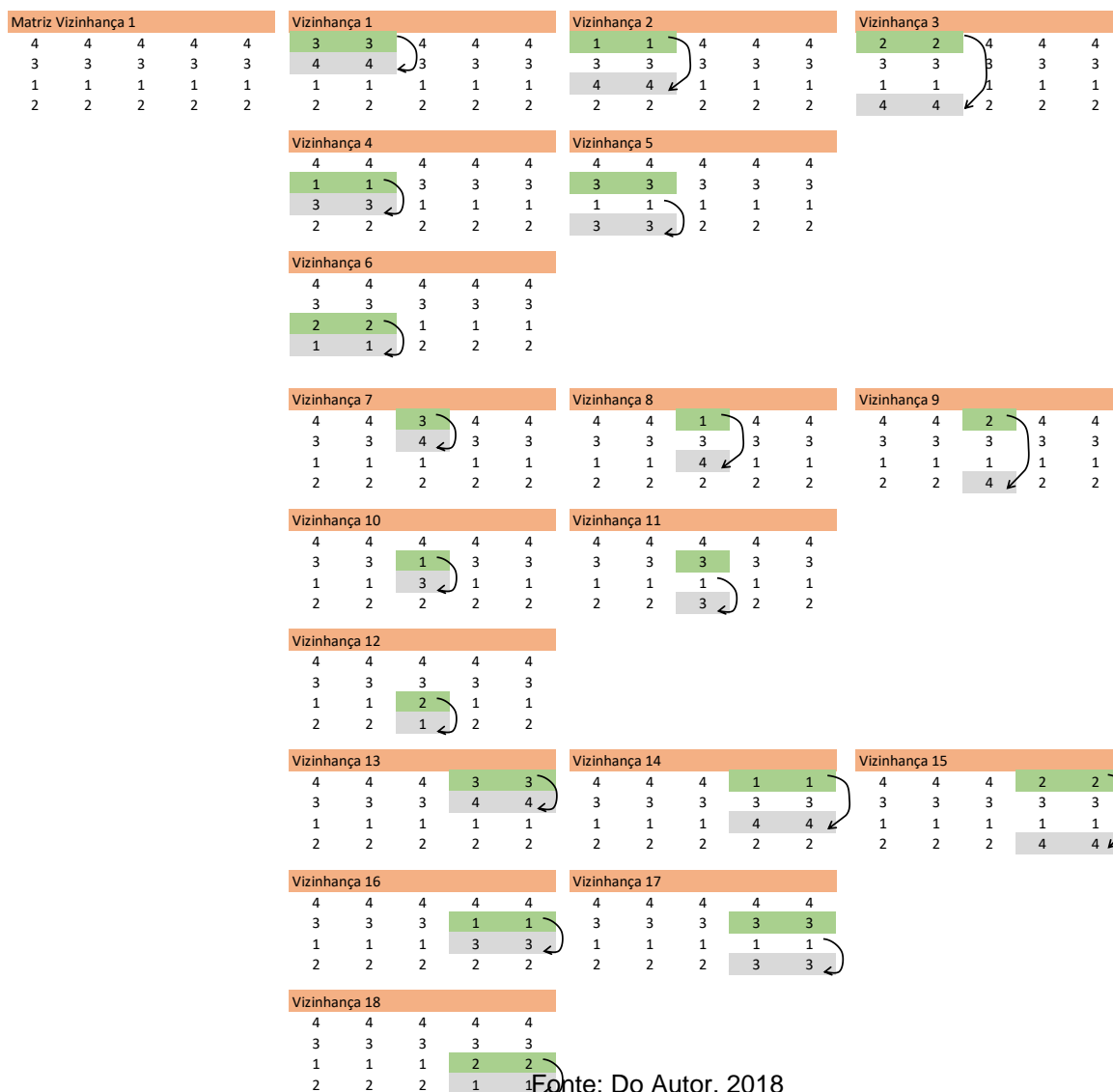
Figura 22 - Representação Lógica da Vizinhança 1 (Macro Esfera)



Fonte: Do Autor, 2018

Partindo da matriz que contém o ótimo local da vizinhança 1, inicia-se a execução da segunda vizinhança, analisando o âmbito micro da solução. A vizinhança se inicia com as trocas individuais dos *jobs* dentro de cada uma das máquinas sempre respeitando a teoria de Tandon (1991), em que as duas primeiras máquinas devem apresentar o mesmo sequenciamento, tal como as duas últimas também devem apresentar sequenciamentos similares. A Figura 23, representa de forma resumida as trocas entre os *jobs* respeitando a teoria.

Figura 23 -Representação Logica da Vizinhança 2 (Micro Esfera)



Caso a melhor solução da Vizinhança 2, ainda não é o ótimo local, as estruturas V1 e V2 continuam se repetindo até que não haja mais melhoria no valo de máximo atraso total obtido. Assim após a definição do ótimo local a matriz que representa o resultado de menos valor para o máximo atraso total, segue sendo utilizada pelo ILS para a operação de perturbação que ocorre na sequência.

3.3 PERTURBAÇÕES

Visando diversificar resultados em busca de uma solução melhor para as instâncias foram desenvolvidos três tipos de perturbações para o ILS, Perturbação Retira *Jobs*, Perturbação Troca Colunas e Perturbação Inverte Matriz.

A primeira perturbação denominada *Retira Jobs*, ilustrada na Figura 24, consiste na geração aleatória de três números que serão retirados da matriz, enquanto a sequência de *jobs* abaixo deles serão reposicionados nas lacunas deixadas pelos mesmos, sendo estes por fim introduzidos novamente a matriz nas últimas três posições de cada uma das máquinas ali presentes.

Figura 24 - Modelo de Perturbação *Retira Jobs*

Matriz Original			Retirando Aleatórios			Preenchendo Lacunas		
1	6	7	1	6	7	1	6	7
2	5	8	2			2	4	9
3	4	9		4	9	4	2	10
4	3	10	4		10	6	1	1
5	2	1		2	1	7	7	4
6	1	4	6	1	4	8	9	2
7	7	2	7	7	2	10	10	6
8	8	3	8			3	3	3
9	9	6		9	6	5	5	5
10	10	5	10	10		8	8	8

Aleatórios: 3,5 e 8

Fonte: Do Autor, 2018

Similar ao sistema de Vizinhança 1, a Perturbação Troca Colunas consiste na troca das colunas de uma matriz retornando aquela com menor atraso total máximo no sistema. Como ilustrado na Figura 25, as trocas realizadas durante a perturbação permanecem respeitando a teoria de Tandon (1991), garantindo que as duas primeiras colunas sejam iguais assim como as duas últimas.

Figura 25 - Modelo de Perturbação Troca Colunas

Matriz Original					Perturbação 1					Perturbação 2				
6	6	1	7	7	1	6	7	7	7	7	1	6	6	
5	5	2	8	8	2	5	8	8	8	8	2	5	5	
4	4	3	9	9	3	4	9	9	9	9	3	4	4	
10	10	4	2	2	4	4	10	2	2	2	4	10	10	
2	2	5	1	1	5	5	2	1	1	1	5	2	2	
1	1	6	4	4	6	6	1	4	4	4	6	1	1	
7	7	7	10	10	7	7	7	10	10	10	7	7	7	
8	8	8	3	3	8	8	8	3	3	3	8	8	8	
9	9	9	6	6	9	9	9	6	6	6	9	9	9	
3	3	10	5	5	10	10	3	5	5	5	10	3	3	

Perturbação 3				
6	6	7	1	1
5	5	8	2	2
4	4	9	3	3
10	10	2	4	4
2	2	1	5	5
1	1	4	6	6
7	7	10	7	7
8	8	3	8	8
9	9	6	9	9
3	3	5	10	10

Fonte: Do Autor, 2018

O último modelo de perturbação denominado Perturbação Inverte Matriz representa a sistemática mais simples do trio. O algoritmo em questão apenas inverte completamente a matriz em análise, colocando a primeira linha com última, a segunda como penúltima e assim por diante como mostra a Figura 25.

Figura 26 - Modelo de Perturbação Inverte Matriz

Matriz Original					Inversao da Matriz				
1	1	1	1	1	10	10	10	10	10
2	2	2	2	2	9	9	9	9	9
3	3	3	3	3	8	8	8	8	8
4	4	4	4	4	7	7	7	7	7
5	5	5	5	5	6	6	6	6	6
6	6	6	6	6	5	5	5	5	5
7	7	7	7	7	4	4	4	4	4
8	8	8	8	8	3	3	3	3	3
9	9	9	9	9	2	2	2	2	2
10	10	10	10	10	1	1	1	1	1

Fonte: Do Autor, 2018

4. RESULTADOS

4.1 GERAÇÃO DE INSTÂNCIAS

Na busca por instâncias que apresentassem datas de entrega (*Due Date*) o presente trabalho desenvolveu um algoritmo de gerações aleatórias de instâncias. Ao todo foram processadas 96 combinações com diferentes quantidades de máquinas, *jobs* e datas de entrega, que posteriormente foram processadas pelo algoritmo de sequenciamento orientado pela metaheurística ILS a fim de obter a melhor solução factível para cada um dos problemas.

O algoritmo de geração de instâncias utiliza os parâmetros inseridos pelo usuário na planilha de acordo com a necessidade de tamanho e complexidade do problema. Desta maneira devem ser definidos respectivamente a quantidade de máquinas, quantidade de *jobs*, a porcentagem de folga no tempo entre os tempos de processamento data de entrega e pôr fim a probabilidade de alguma das tarefas geradas dentro do *job* ser zero.

Os laços de repetição principais baseiam-se respectivamente nos valores de *jobs* e máquinas imputados pelo usuário, como podemos ver nas linhas 3 e 4 do pseudocódigo Gera_Instância ilustrado pela Figura 27. Subsequentemente é atribuído um tempo de processamento a tarefa $p_{i,j}$, este varia de acordo com a probabilidade do mesmo ser zero, dado também imputado pelo usuário. Nesta etapa a única restrição presente consta na linha 6, que evidencia que a última máquina nunca vai terá uma tarefa sem tempo de processamento, evitando que um *job* tenha seu tempo total de processamento zerado. Por fim é definido o tempo para a entrega do *job*, pela soma de todos os tempos de processamentos gerados multiplicando pela geração aleatória de um número entre 0 e a porcentagem de folga estipulada pelo usuário. Os laços são realizados duas vezes gerando assim duas versões aleatórias para o conjunto de dado.

Figura 27 - Pseudocódigo Gera_Instância

Procedimento	Gera_Instância
1	MAQ, JOB, FOLGA, PROB \leftarrow Atribuindo valor as variáveis
2	P(i,j), D(i,j), NUM, COUNT \leftarrow Declarando variáveis
3	De i = 1 Até (JOB)
4	De j = 1 Até (MAQ)
5	Se j = MAQ Então
6	P(i,j) Atribuir \leftarrow Num. Aleatório entre 0 e 60 segundos para posição i,j
7	Se não
8	NUM \leftarrow Atribuir num. aleatório entre 0 e 1
9	Se NUM < PROB ou NUM = PROB Então
10	P(i, j) = 0
11	Se não
12	P(i,j) \leftarrow Atribuir Num aleatório entre 0 e 60 segundos para posição i,j
13	Final se
14	Final se
15	COUNT = P(i, j) + COUNT
16	Próximo j
17	NUM \leftarrow Atribuir número aleatório entre 0 e a FOLGA
18	D(i,j) = COUNT*NUM
19	Próximo i

Fonte: Do Autor, 2018

O algoritmo é encerrado após realizar todos os procedimentos descritos no pseudocódigo e salvar os valores de $P_{i,j}$ e $D_{i,j}$ obtidos em um arquivo TXT. de forma matricial onde a última coluna representa o valor de $D_{i,j}$, e a primeira linha respectivamente o valor de *jobs* e máquinas descritas pelo usuário. A nomenclatura do arquivo também é automática e fica representada pelo número da instância, versão gerada, quantidade de máquinas, quantidade de *jobs*, porcentagem de folga e probabilidade de número zero, como ilustrado na Figura 28.

Figura 28 - Exemplo de nomenclatura e disposição dos dados em uma instância gerada pelo algoritmo gerador de instâncias

1-(V1)(MAQ 3)(JOB 10)(FOLGA 10)(PROB 20) - Bloco de notas			
Arquivo	Editar	Formatar	Exibir Ajuda
10	3		
40	35	37	203,84
29	33	10	134,64
56	33	34	151,29
23	5	29	70,11
53	0	50	123,6
25	32	49	208,82
59	24	53	174,08
15	15	45	141,75
37	0	25	104,16
0	18	14	42,88

Fonte: Do Autor, 2018

Ao todo foram parametrizados com o algoritmo 48 combinações de quantidade de trabalhos, máquinas, porcentagem de folgas para a data de entrega e porcentagem de probabilidade de um dos trabalhos ser zero, como ilustra Tabela 6. Assim para cada combinação de quantidade de *job* e máquina a porcentagem de folga na data de entrega podia variar de 0 a 10% ou de 0 a 20%, ao mesmo tempo que a probabilidade de um trabalho ser zero poderia variar de 0 a 20% ou de 0 a 40%. Como dito anteriormente cada uma das combinações foi processada duas vezes gerando assim duas versões para cada um dos modelos totalizando 96 instancias.

Tabela 6 - Tabela de Combinações Para Instancias Geradas

Instância	Qt Máquinas	Qt Jobs	% Folga Tempo	Probabilidade 0
1	3	10	10	20
2	3	10	10	40
3	3	10	20	20
4	3	10	20	40
5	3	20	10	20
6	3	20	10	40
7	3	20	20	20
8	3	20	20	40
9	3	30	10	20
10	3	30	10	40
11	3	30	20	20
12	3	30	20	40
13	5	10	10	20
14	5	10	10	40
15	5	10	20	20
16	5	10	20	40
17	5	20	10	20
18	5	20	10	40
19	5	20	20	20
20	5	20	20	40
21	5	30	10	20
22	5	30	10	40
23	5	30	20	20
24	5	30	20	40
25	8	10	10	20
26	8	10	10	40
27	8	10	20	20
28	8	10	20	40
29	8	20	10	20
30	8	20	10	40
31	8	20	20	20
32	8	20	20	40
33	8	30	10	20
34	8	30	10	40
35	8	30	20	20
36	8	30	20	40
37	10	10	10	20
38	10	10	10	40
39	10	10	20	20
40	10	10	20	40
41	10	20	10	20
42	10	20	10	40
43	10	20	20	20
44	10	20	20	40
45	10	30	10	20
46	10	30	10	40
47	10	30	20	20
48	10	30	20	40

Fonte: Do Autor, 2018

4.2 PERTURBAÇÕES

O algoritmo proposto foi executado usando a mesma estrutura da Figura 19, porém alternando o tipo de perturbação utilizada. Por essa razão, as 3 próximas seções apresentam os resultados obtidos pela utilização de cada uma das perturbações ao algoritmo ILS.

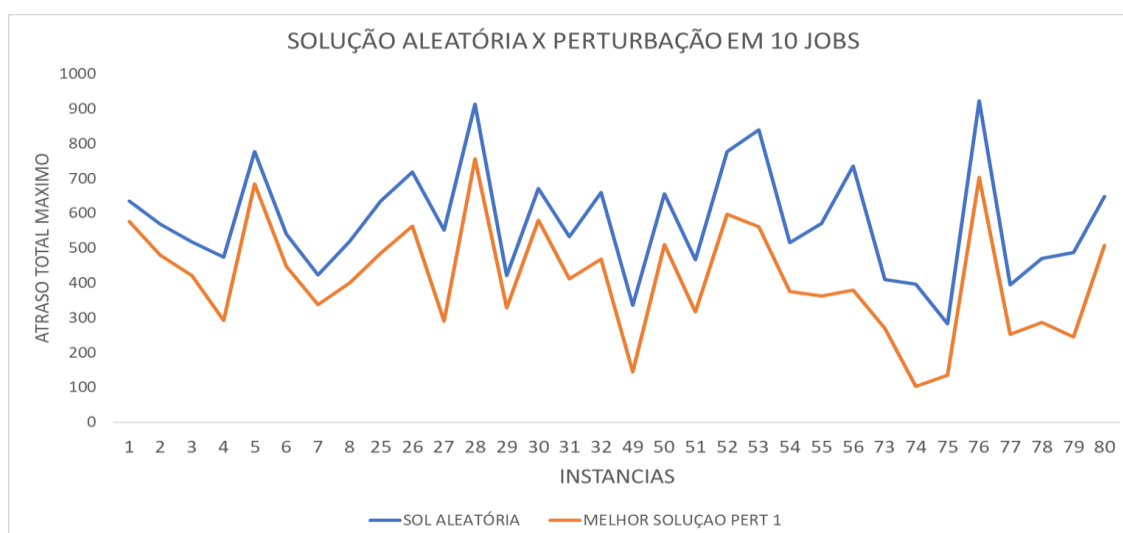
Foram avaliados o desempenho de cada uma das perturbações nas instâncias de 10, 20 e 30 *jobs* relatando o mínimo atraso total obtido, tempo de processamento individual por iteração e porcentagem de melhora individual da solução com relação ao resultado aleatório.

4.2.1 Resultados Perturbação 1 – Retira *Jobs*

A perturbação 1 ou Retira *Jobs* como dito na seção anterior consiste na retirada aleatória de números de uma matriz e reordenação dos demais sobressalentes. Analisando os valores encontrados para as melhores soluções do ILS com perturbação 1 com os valores das 10 mil soluções aleatórias geradas, demonstrando uma redução geral média de cerca de 25% para as soluções obtidas.

Subdividindo a análise dos resultados para as instâncias de apenas dez *jobs*, a Tabela 7 e a Figura 29 ilustram o comportamento da melhor solução aleatória e a melhor solução do ILS com perturbação 1, observando média de 30% nos valores de melhor solução para as instâncias, assim como um tempo de iteração médio de cerca de 3,44 segundos por iteração.

Figura 29 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 1 Em 10 *Jobs*



Fonte: Do Autor, 2018

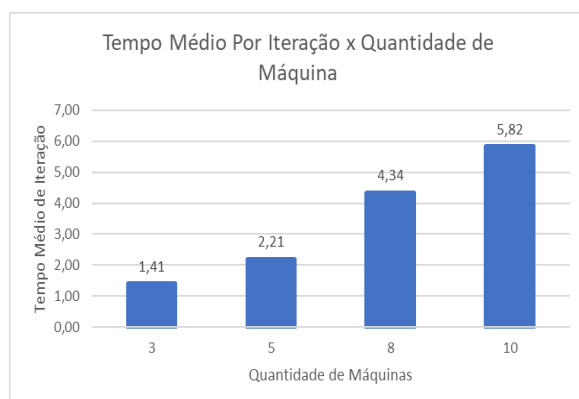
Tabela 7 - Dados das Instâncias de 10 Jobs Com Perturbação 1

INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
1	3	636	577	9,3%	601,7	470	1,28
2	3	569	479	15,8%	601,6	394	1,53
3	3	517	422	18,4%	600,1	405	1,48
4	3	475	293	38,3%	600,4	380	1,58
5	3	777	685	11,8%	600,3	476	1,26
6	3	541	448	17,2%	600,9	493	1,22
7	3	423	338	20,1%	600,5	366	1,64
8	3	520	400	23,1%	601,3	477	1,26
25	5	635	485	23,6%	600,5	289	2,08
26	5	719	563	21,7%	600,8	261	2,30
27	5	552	291	47,3%	602,1	250	2,41
28	5	914	756	17,3%	600,2	302	1,99
29	5	422	329	22,0%	601,1	270	2,23
30	5	672	580	13,7%	601,1	283	2,12
31	5	533	412	22,7%	600,4	251	2,39
32	5	659	468	29,0%	602,0	282	2,13
49	8	335	145	56,7%	601,8	139	4,33
50	8	657	511	22,2%	602,6	117	5,15
51	8	466	317	32,0%	601,4	144	4,18
52	8	778	598	23,1%	603,3	137	4,40
53	8	839	562	33,0%	601,8	140	4,30
54	8	516	375	27,3%	602,2	147	4,10
55	8	570	362	36,5%	600,0	138	4,35
56	8	735	379	48,4%	601,0	153	3,93
73	10	410	270	34,1%	600,5	102	5,89
74	10	397	102	74,3%	604,3	95	6,36
75	10	282	136	51,8%	603,6	90	6,71
76	10	924	703	23,9%	603,2	116	5,20
77	10	395	253	35,9%	603,4	110	5,49
78	10	470	286	39,1%	605,3	100	6,05
79	10	488	245	49,8%	604,0	105	5,75
80	10	649	508	21,7%	601,8	117	5,14

Fonte: Do Autor, 2018

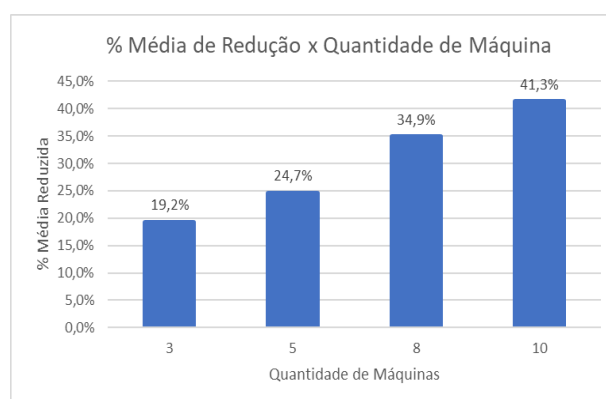
O tempo médio das iterações assim como a porcentagem de redução da solução obtida, aumenta conforme aumentam as quantidades de máquinas na sequência dos *jobs*. Assim como ilustrado nas Figuras 30 e 31, o maior tempo médio de processamento para instâncias de 10 *jobs* se faz presente em sequências com 10 máquinas sob o valor médio de 5,82 segundo, assim como a maior média de redução na solução obtida em cerca de 41,3%.

Figura 31 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 10 Jobs Com Perturbação 1



Fonte: Do Autor, 2018

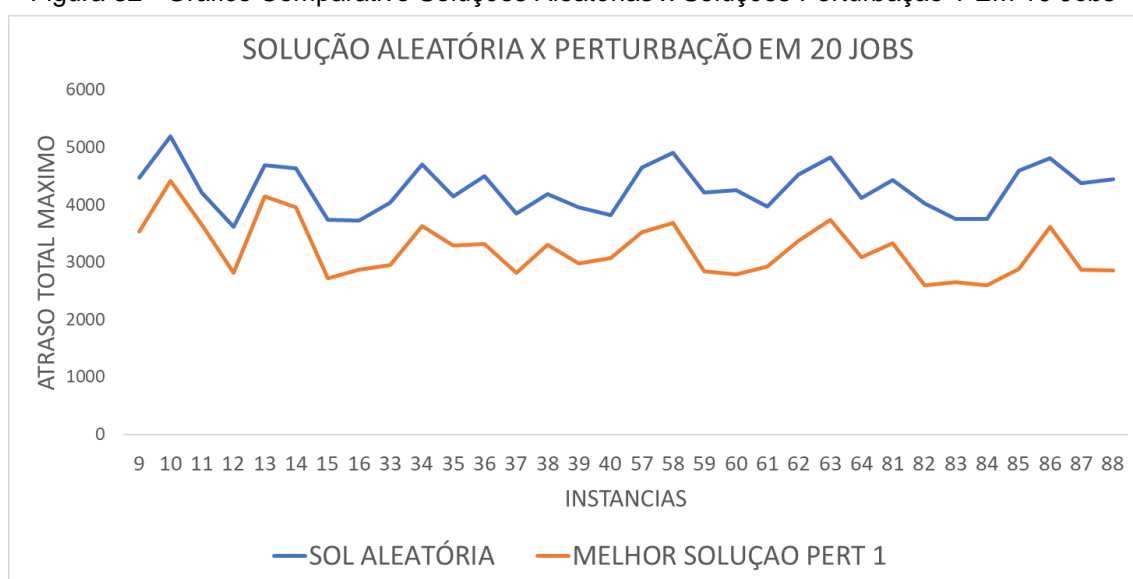
Figura 31 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 10 Jobs Com Perturbação 1



Fonte: Do Autor, 2018

No processamento de instâncias de 20 jobs, ilustrados pela Figura 32 e pela Tabela 8, apresentam uma queda em seu desempenho quando comparados a solução aleatória anterior com 10 jobs. Ao dobrar a quantidade de jobs processados o tempo de processamento passou de 5,82 segundos para uma média de 36s segundos juntamente com a queda da porcentagem média de melhora para 25%.

Figura 32 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 1 Em 10 Jobs



Fonte: Do Autor, 2018

Tabela 8 - Dados das Instâncias de 20 *Jobs* Com Perturbação 1

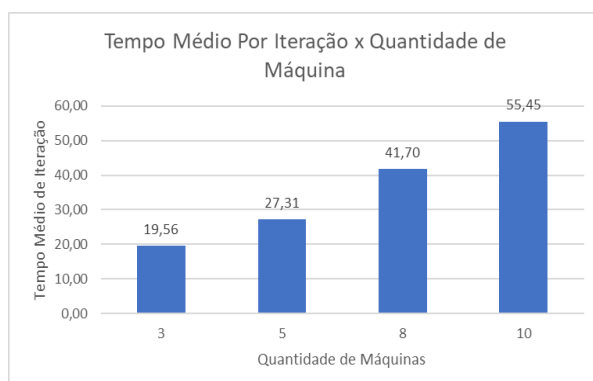
INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
9	3	4469	3530	21,0%	600,2	37	16,22
10	3	5189	4411	15,0%	607,6	38	15,99
11	3	4213	3655	13,2%	617,0	27	22,85
12	3	3613	2815	22,1%	602,8	29	20,79
13	3	4685	4142	11,6%	602,5	31	19,43
14	3	4637	3961	14,6%	614,1	36	17,06
15	3	3739	2718	27,3%	613,7	25	24,55
16	3	3730	2875	22,9%	608,6	31	19,63
33	5	4041	2955	26,9%	611,3	25	24,45
34	5	4708	3632	22,9%	609,7	22	27,71
35	5	4141	3284	20,7%	603,5	20	30,17
36	5	4497	3315	26,3%	612,2	21	29,15
37	5	3849	2819	26,8%	622,0	18	34,55
38	5	4190	3305	21,1%	610,3	26	23,47
39	5	3954	2973	24,8%	610,7	22	27,76
40	5	3825	3073	19,7%	614,1	29	21,18
57	8	4655	3519	24,4%	601,4	15	40,09
58	8	4902	3677	25,0%	603,7	18	33,54
59	8	4216	2839	32,7%	633,3	13	48,72
60	8	4248	2792	34,3%	648,3	15	43,22
61	8	3968	2925	26,3%	603,7	16	37,73
62	8	4526	3376	25,4%	633,3	15	42,22
63	8	4820	3742	22,4%	627,2	14	44,80
64	8	4124	3088	25,1%	606,3	14	43,31
81	10	4437	3334	24,9%	616,0	12	51,33
82	10	4019	2595	35,4%	608,0	11	55,27
83	10	3754	2648	29,5%	636,1	10	63,61
84	10	3752	2595	30,8%	640,5	11	58,23
85	10	4598	2880	37,4%	607,3	12	50,61
86	10	4805	3617	24,7%	623,7	12	51,98
87	10	4375	2869	34,4%	643,4	11	58,49
88	10	4443	2851	35,8%	649,0	12	54,08

Fonte: Do Autor, 2018

Realizando uma análise individual por máquina as instâncias de 20 *jobs* apresentam o mesmo comportamento das analisadas anteriormente, ilustrado nas Figuras 33 e 34. Assim com o aumento do número de máquinas temos o aumento do tempo de processamento assim como o aumento na porcentagem de redução da

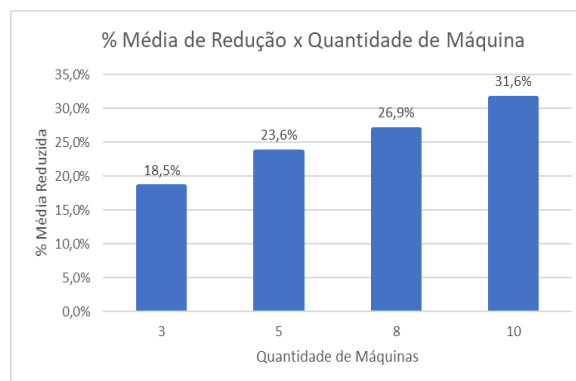
solução obtida quando comparada com o parâmetro aleatório. A instância 87 com dez máquinas apresentou o maior tempo de processamento em 58,49 segundos enquanto a maior porcentagem reduzida ficou com a instância 88 também com dez máquinas.

Figura 34 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 20 Jobs Com Perturbação 1



Fonte: Do Autor, 2018

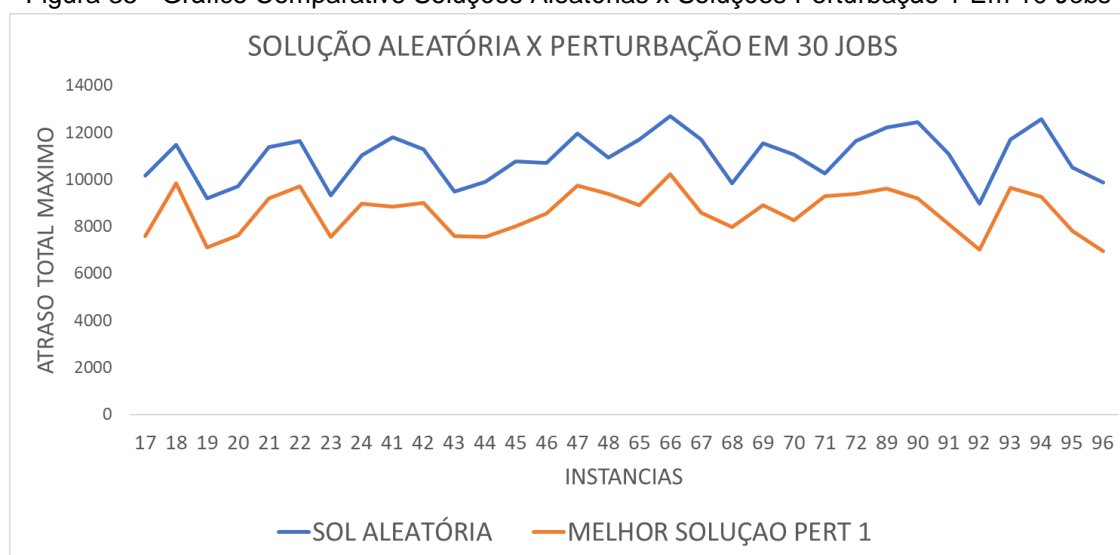
Figura 34 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 20 Jobs Com Perturbação 1



Fonte: Do Autor, 2018

Por fim as instâncias com 30 jobs, ilustradas na Tabela 9 e Figura 35, perpetuaram o comportamento das instâncias de 20 jobs aumentando seus valores médios com o aumento de jobs. O tempo de processamento para o conjunto chegou a média de 213 segundos por instâncias sendo a obtendo uma porcentagem de redução de apenas para apenas 21% para as soluções.

Figura 35 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 1 Em 10 Jobs



Fonte: Do Autor, 2018

Tabela 9 - Dados das Instâncias de 30 *Jobs* Com Perturbação 1

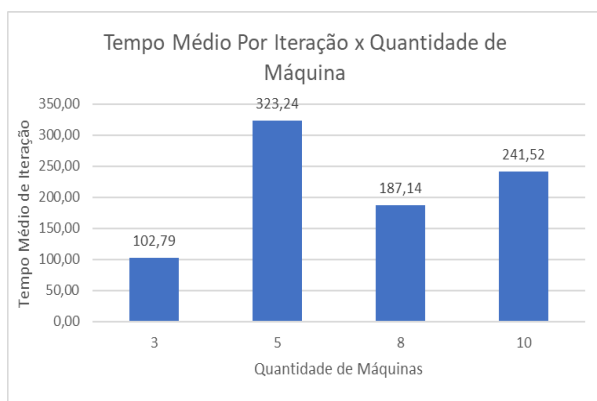
INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
17	3	10153	7598	25,2%	605,6	7	86,51
18	3	11469	9851	14,1%	638,3	6	106,38
19	3	9210	7105	22,9%	660,3	6	110,05
20	3	9727	7623	21,6%	671,9	8	83,98
21	3	11375	9215	19,0%	616,0	6	102,67
22	3	11626	9724	16,4%	685,6	6	114,27
23	3	9327	7565	18,9%	619,1	6	103,19
24	3	11028	8962	18,7%	691,7	6	115,28
41	5	11809	8835	25,2%	675,9	6	112,65
42	5	11295	9023	20,1%	675,4	6	112,57
43	5	9494	7590	20,1%	682,7	4	170,67
44	5	9903	7554	23,7%	754,3	2	377,17
45	5	10771	8015	25,6%	708,6	2	354,31
46	5	10701	8545	20,1%	831,9	2	415,96
47	5	11953	9743	18,5%	877,4	2	438,72
48	5	10933	9401	14,0%	603,9	1	603,87
65	8	11721	8917	23,9%	669,3	4	167,31
66	8	12717	10217	19,7%	690,8	4	172,69
67	8	11710	8590	26,6%	624,6	4	156,15
68	8	9836	7974	18,9%	802,6	4	200,65
69	8	11536	8922	22,7%	759,2	5	151,83
70	8	11076	8255	25,5%	764,8	4	191,21
71	8	10253	9287	9,4%	763,0	3	254,33
72	8	11637	9405	19,2%	811,8	4	202,94
89	10	12216	9622	21,2%	778,3	4	194,58
90	10	12444	9203	26,0%	878,6	3	292,88
91	10	11081	8097	26,9%	746,5	3	248,84
92	10	8968	7009	21,8%	718,5	3	239,49
93	10	11690	9656	17,4%	633,8	3	211,27
94	10	12561	9255	26,3%	751,5	3	250,52
95	10	10501	7825	25,5%	740,2	3	246,75
96	10	9870	6936	29,7%	743,6	3	247,86

Fonte: Do Autor, 2018

Analisando o comportamento do conjunto de instâncias com relação ao número de máquinas, as instâncias de 30 *jobs* apresentaram comportamento peculiar com relação as demais já analisadas. A curva dos tempos por iteração apresentou um pico para instâncias de cinco máquinas, com o máximo de tempo de processamento em cerca de dez minutos, como mostra a Figura 36. Na sequência o valor reduz para instâncias de oito máquinas e volta a crescer para instâncias de dez máquinas. Em

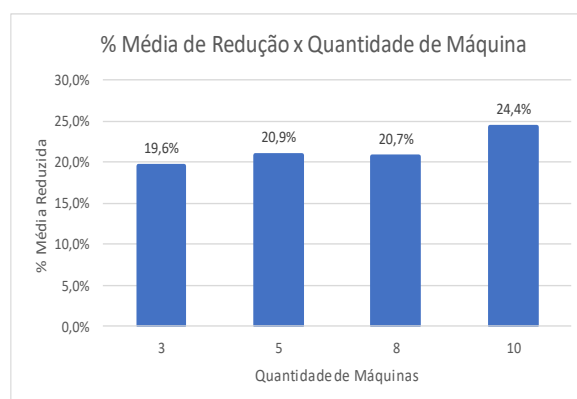
contrapartida, como ilustra a Figura 37, a porcentagem de redução permaneceu crescente, apresentam um valor máximo de 27,9% na instância 96.

Figura 37 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 30 Jobs Com Perturbação 1



Fonte: Do Autor, 2018

Figura 37 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 30 Jobs Com Perturbação 1



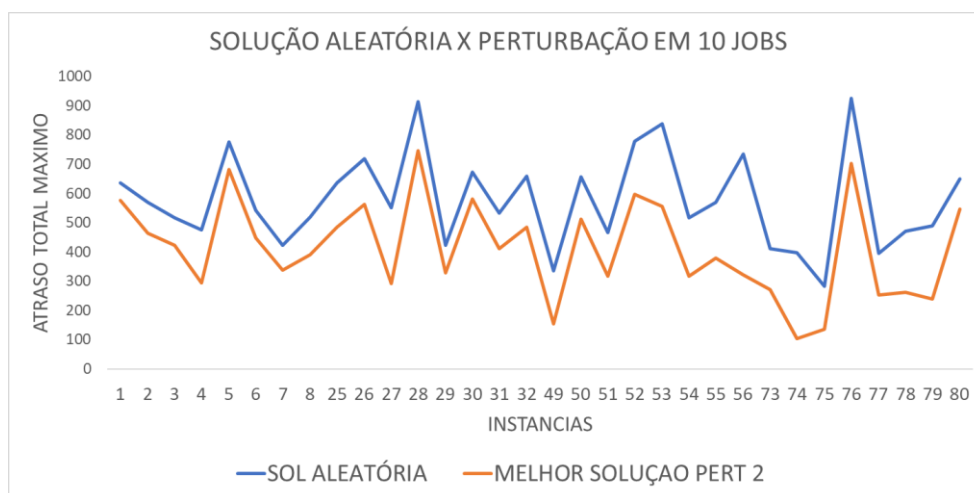
Fonte: Do Autor, 2018

4.2.2 Resultados Perturbação 2 – Troca Coluna

Partindo de um conceito similar ao de vizinhança, a perturbação 2 ou Troca Coluna, realiza a troca das colunas de uma matriz e retorna o melhor resultado obtido. As instâncias que tiveram o ILS executado com a segunda perturbação, assim como a perturbação 1, apresentaram melhores valores do que as soluções aleatórias com uma média de 26% de redução.

Realizando uma análise por blocos assim como feito na permutação 1, a Figura 38 e a Tabela 10, exibem os resultados dos processamentos das instâncias de 10 jobs com a segunda permutação, com uma redução média de 31% e tempo médio de processamento de 1,5 segundos, valores superiores aos alcançados na permutação 1.

Figura 38 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 2



Fonte: Do Autor, 2018

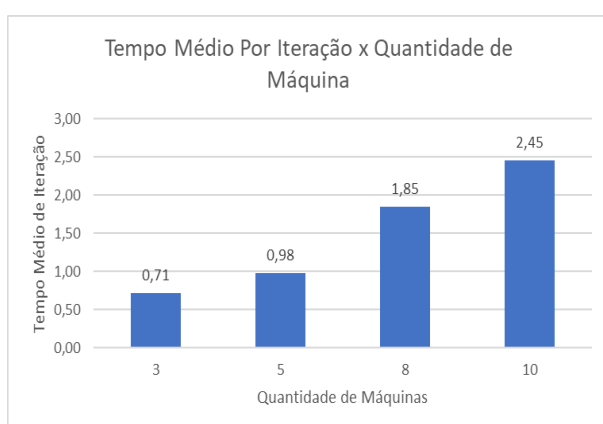
Tabela 10 - Dados das Instâncias de 10 Jobs Com Perturbação 2

INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
1	3	636	577	9,3%	600,2	884	0,68
2	3	569	465	18,3%	600,4	844	0,71
3	3	517	422	18,4%	600,5	806	0,75
4	3	475	293	38,3%	600,7	764	0,79
5	3	777	683	12,1%	600,1	901	0,67
6	3	541	448	17,2%	600,7	945	0,64
7	3	423	338	20,1%	600,6	746	0,81
8	3	520	390	25,0%	600,8	907	0,66
25	5	635	485	23,6%	601,1	634	0,95
26	5	719	563	21,7%	600,5	621	0,97
27	5	552	291	47,3%	600,8	588	1,02
28	5	914	746	18,4%	600,1	646	0,93
29	5	422	329	22,0%	600,5	616	0,97
30	5	672	580	13,7%	600,1	605	0,99
31	5	533	412	22,7%	600,0	595	1,01
32	5	659	484	26,6%	600,0	613	0,98
49	8	335	153	54,3%	601,0	317	1,90
50	8	657	511	22,2%	601,6	306	1,97
51	8	466	317	32,0%	600,9	333	1,80
52	8	778	598	23,1%	601,6	323	1,86
53	8	839	556	33,7%	600,1	330	1,82
54	8	516	318	38,4%	600,4	331	1,81
55	8	570	379	33,5%	601,2	337	1,78
56	8	735	321	56,3%	600,7	330	1,82
73	10	410	270	34,1%	601,8	238	2,53
74	10	397	103	74,1%	602,9	241	2,50
75	10	282	136	51,8%	601,4	239	2,52
76	10	924	703	23,9%	601,5	255	2,36
77	10	395	253	35,9%	602,2	244	2,47
78	10	470	261	44,5%	602,4	248	2,43
79	10	488	238	51,2%	601,6	245	2,46
80	10	649	546	15,9%	600,7	256	2,35

Fonte: Do Autor, 2018

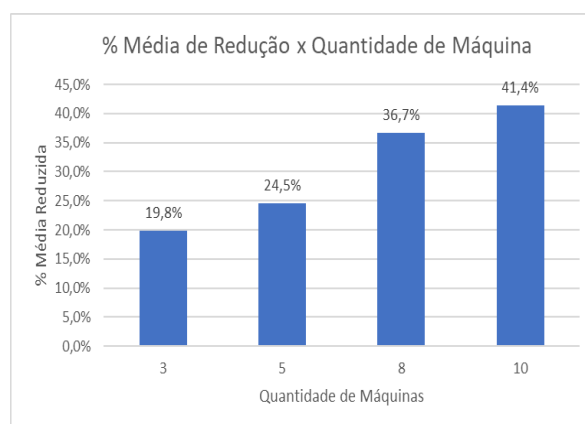
Assim como na permutação anterior as instâncias de 10 *jobs* comportaram-se de maneira similar na perturbação 2. O Tempo médio das iterações por máquina permaneceu crescente, Figura 39, tendo seu pico médio em 2,45 segundos para instâncias de dez máquinas enquanto a porcentagem de redução por máquina também crescia, Figura 40, tendo o seu maior valor médio em 41,4 e seu pico na instância 74 com cerca de 74,1%.

Figura 40 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 10 *Jobs* Com Perturbação 2



Fonte: Do Autor, 2018

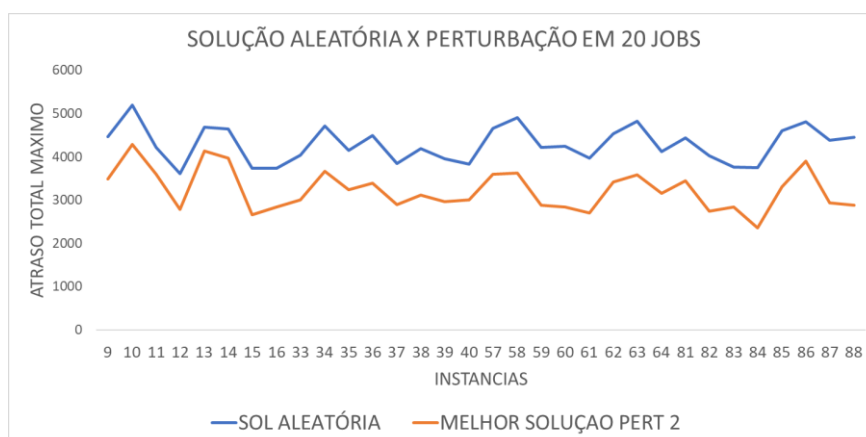
Figura 40 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 10 *Jobs* Com Perturbação 2



Fonte: Do Autor, 2018

No processamento de instâncias de 20 *jobs*, exibido na Tabela 11 e ilustrado pela Figura 41, comportou-se de maneira similar ao encontrado na perturbação 1. Ao dobrar a quantidade de *jobs* processados o tempo de processamento também aumentou de 1,5 para 16,15 segundos, e a porcentagem média de melhora caiu em 5,6% ao comparado a instâncias de 10 *jobs*.

Figura 41 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 2



Fonte: Do Autor, 2018

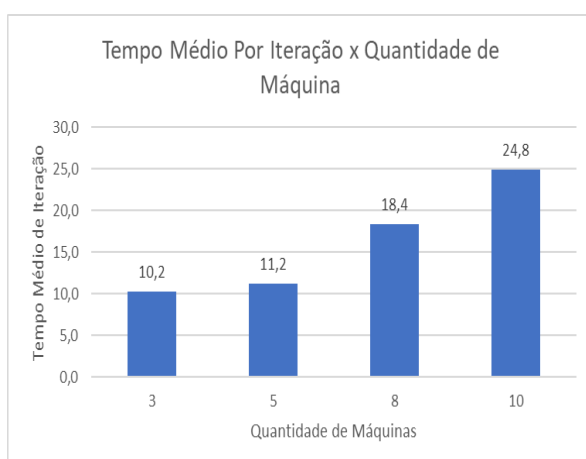
Tabela 11 - Dados das Instâncias de 20 Jobs Com Perturbação 2

INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
9	3	4469	3479	22,2%	603,6	68	8,88
10	3	5189	4278	17,6%	605,7	71	8,53
11	3	4213	3599	14,6%	603,1	53	11,38
12	3	3613	2790	22,8%	607,4	53	11,46
13	3	4685	4130	11,8%	608,1	60	10,14
14	3	4637	3968	14,4%	603,1	61	9,89
15	3	3739	2653	29,0%	607,3	52	11,68
16	3	3730	2839	23,9%	605,4	63	9,61
33	5	4041	3007	25,6%	604,3	56	10,79
34	5	4708	3665	22,2%	604,1	53	11,40
35	5	4141	3242	21,7%	604,1	55	10,98
36	5	4497	3394	24,5%	603,1	52	11,60
37	5	3849	2888	25,0%	606,2	48	12,63
38	5	4190	3120	25,5%	601,3	57	10,55
39	5	3954	2965	25,0%	605,3	54	11,21
40	5	3825	3009	21,3%	604,2	59	10,24
57	8	4655	3598	22,7%	616,8	36	17,13
58	8	4902	3621	26,1%	610,0	33	18,49
59	8	4216	2874	31,8%	611,4	31	19,72
60	8	4248	2836	33,2%	613,5	35	17,53
61	8	3968	2695	32,1%	601,0	33	18,21
62	8	4526	3416	24,5%	611,1	32	19,10
63	8	4820	3584	25,6%	616,4	34	18,13
64	8	4124	3154	23,5%	616,7	33	18,69
81	10	4437	3450	22,2%	612,3	26	23,55
82	10	4019	2747	31,6%	606,7	25	24,27
83	10	3754	2843	24,3%	620,2	24	25,84
84	10	3752	2357	37,2%	619,2	23	26,92
85	10	4598	3308	28,1%	600,1	26	23,08
86	10	4805	3899	18,9%	606,3	26	23,32
87	10	4375	2930	33,0%	620,0	25	24,80
88	10	4443	2880	35,2%	619,5	23	26,94

Fonte: Do Autor, 2018

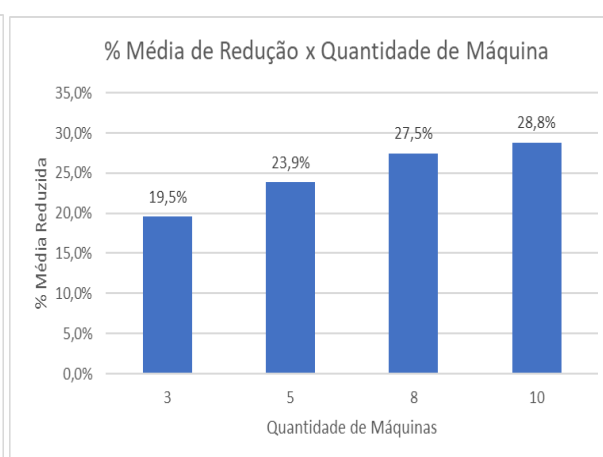
As instâncias de 20 *jobs*, ilustrado nas Figuras 42 e 43, apresentam tempo por iteração e a porcentagem de redução crescentes juntamente com o aumento no número de máquinas, sendo as instâncias com dez máquinas portadoras dos maiores valores médios obtidos.

Figura 43 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 20 *Jobs* Com Perturbação 2



Fonte: Do Autor, 2018

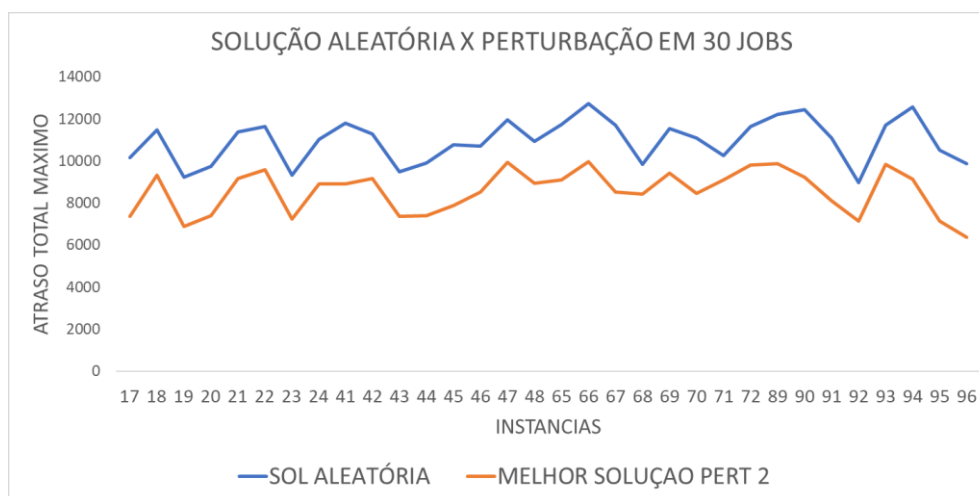
Figura 43 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 20 *Jobs* Com Perturbação 2



Fonte: Do Autor, 2018

Ilustradas na Tabela 12 encontram-se as instâncias de 30 *jobs* processadas pela perturbação 2. Como ilustra a Figura 44 estas mantiveram mesmo comportamento das demais instâncias, obtendo uma média de 76,4 segundos de processamento por instância e 22% de redução média nas soluções.

Figura 44 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 2



Fonte: Do Autor, 2018

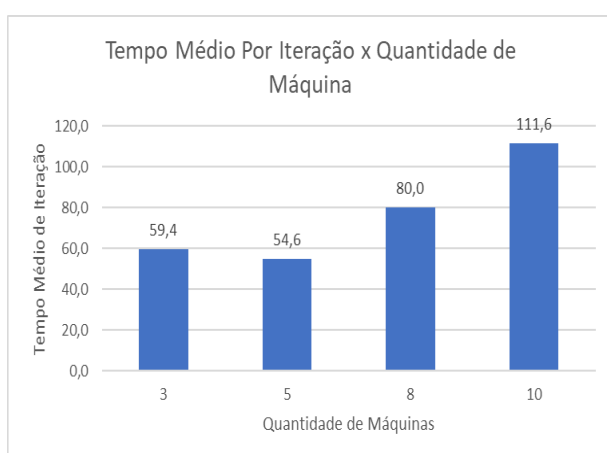
Tabela 12 - Dados das Instâncias de 30 Jobs Com Perturbação 2

INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
17	3	10153	7352	27,6%	651,6	12	54,30
18	3	11469	9315	18,8%	638,0	9	70,89
19	3	9210	6894	25,1%	629,5	10	62,95
20	3	9727	7381	24,1%	601,2	12	50,10
21	3	11375	9151	19,6%	628,4	12	52,37
22	3	11626	9567	17,7%	656,0	12	54,67
23	3	9327	7224	22,5%	676,2	10	67,62
24	3	11028	8893	19,4%	622,9	10	62,29
41	5	11809	8890	24,7%	606,7	12	50,56
42	5	11295	9146	19,0%	618,3	12	51,53
43	5	9494	7363	22,4%	603,9	12	50,32
44	5	9903	7391	25,4%	640,5	12	53,38
45	5	10771	7865	27,0%	630,8	11	57,35
46	5	10701	8508	20,5%	621,5	11	56,50
47	5	11953	9945	16,8%	650,6	12	54,22
48	5	10933	8940	18,2%	630,0	10	63,00
65	8	11721	9101	22,4%	636,7	8	79,59
66	8	12717	9957	21,7%	647,0	9	71,89
67	8	11710	8514	27,3%	609,4	9	67,71
68	8	9836	8417	14,4%	610,5	7	87,22
69	8	11536	9414	18,4%	625,5	9	69,50
70	8	11076	8467	23,6%	614,6	7	87,80
71	8	10253	9082	11,4%	624,9	7	89,27
72	8	11637	9798	15,8%	607,0	7	86,71
89	10	12216	9873	19,2%	615,4	6	102,56
90	10	12444	9220	25,9%	700,2	6	116,70
91	10	11081	8085	27,0%	632,9	6	105,48
92	10	8968	7146	20,3%	709,7	6	118,28
93	10	11690	9822	16,0%	638,0	6	106,34
94	10	12561	9142	27,2%	714,5	6	119,09
95	10	10501	7139	32,0%	641,0	6	106,83
96	10	9870	6352	35,6%	704,2	6	117,37

Fonte: Do Autor, 2018

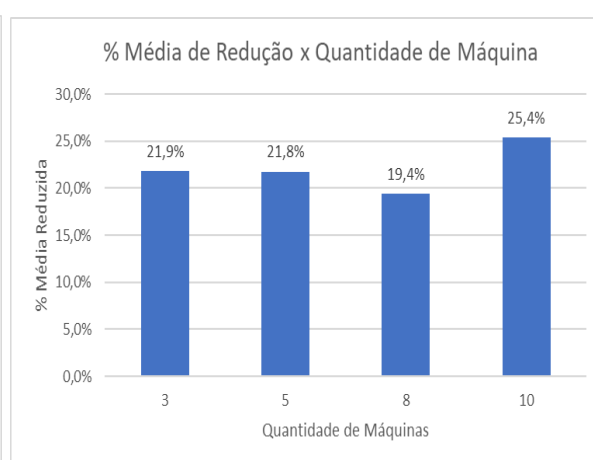
Assim como as instâncias de 30 *jobs* na perturbação 1, estas comportam-se de forma fora dos padrões esperados. Como é possível observar na Figura 45, o tempo médio por iteração caiu para máquinas de 5 *jobs* e após isso voltou a subir. No caso da porcentagem média de redução, Figura 46, observou-se uma queda em instâncias com cinco máquinas voltando a crescer nas de oito máquinas.

Figura 46 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 30 *Jobs* Com Perturbação 2



Fonte: Do Autor, 2018

Figura 46 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 30 *Jobs* Com Perturbação 2



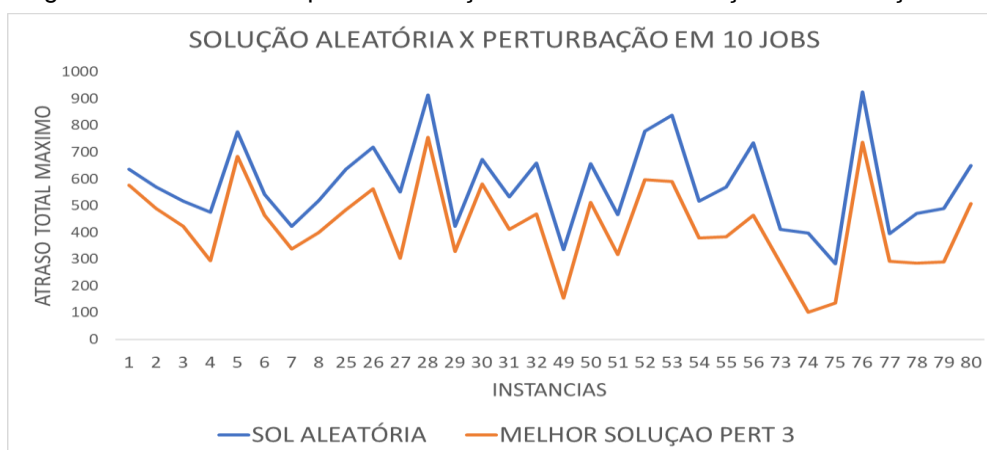
Fonte: Do Autor, 2018

4.2.3 Resultados Perturbação 3 – Inverte Matriz

A última perturbação utilizada no processamento do ILS para as instâncias consiste na perturbação de inversão, onde as linhas primeiras linhas de uma matriz são trocadas pelas ultimas assim como contextualizado na seção 3.3. O comportamento das instâncias executadas no ILS com perturbação 3 comparados as aos resultados das matrizes aleatórios, tem como média na porcentagem de redução dos valores 23%.

A Tabela 13 e a Figura 47, apresentam os dados de processamento das instâncias de 10 *jobs* processadas na perturbação 3. O tempo médio de processamento das instâncias para uma iteração foi de cerca de 4,4 segundos, com uma porcentagem média de 28% de melhora em relação a solução aleatória.

Figura 47 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 3



Fonte: Do Autor, 2018

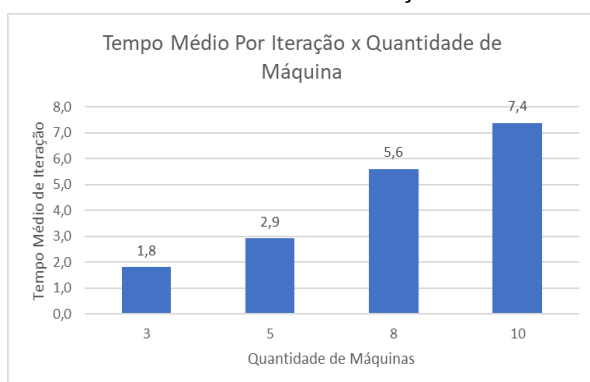
Tabela 13 - Dados das Instâncias de 10 Jobs Com Perturbação 3

INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
1	3	636	577	9,3%	600,9	366	1,64
2	3	569	490	13,9%	601,0	332	1,81
3	3	517	422	18,4%	601,9	291	2,07
4	3	475	293	38,3%	600,6	313	1,92
5	3	777	685	11,8%	600,3	462	1,30
6	3	541	464	14,2%	601,4	368	1,63
7	3	423	338	20,1%	600,8	253	2,37
8	3	520	400	23,1%	602,3	343	1,76
25	5	635	485	23,6%	601,2	220	2,73
26	5	719	563	21,7%	601,3	201	2,99
27	5	552	304	44,9%	600,9	173	3,47
28	5	914	756	17,3%	601,9	241	2,50
29	5	422	329	22,0%	602,8	219	2,75
30	5	672	580	13,7%	600,1	230	2,61
31	5	533	412	22,7%	602,5	173	3,48
32	5	659	468	29,0%	600,7	207	2,90
49	8	335	155	53,7%	600,4	99	6,06
50	8	657	511	22,2%	600,0	97	6,19
51	8	466	317	32,0%	605,8	110	5,51
52	8	778	598	23,1%	601,9	107	5,62
53	8	839	591	29,6%	607,6	112	5,42
54	8	516	380	26,4%	603,3	116	5,20
55	8	570	383	32,8%	607,9	107	5,68
56	8	735	463	37,0%	605,1	117	5,17
73	10	410	285	30,5%	602,2	77	7,82
74	10	397	102	74,3%	603,6	79	7,64
75	10	282	136	51,8%	607,7	79	7,69
76	10	924	737	20,2%	606,7	87	6,97
77	10	395	292	26,1%	602,7	88	6,85
78	10	470	286	39,1%	603,8	76	7,94
79	10	488	289	40,8%	602,0	86	7,00
80	10	649	508	21,7%	606,4	85	7,13

Fonte: Do Autor, 2018

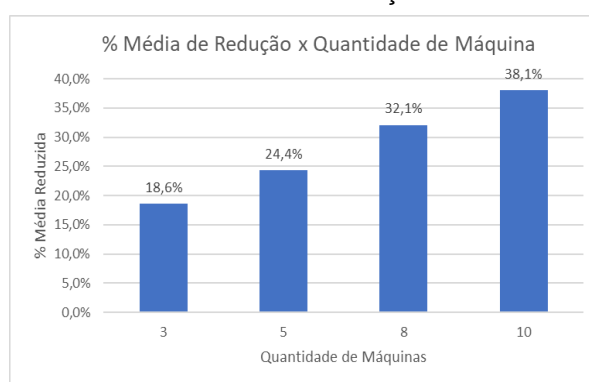
Com relação ao comportamento das instâncias com 10 *jobs* nas máquinas ao aumento de máquinas, ilustrados nas Figuras 48 e 49, o mesmo permaneceu similar ao encontrado nas outras duas perturbações. Em ambos os casos quanto mais máquinas maiores o tempo de processamento e maior a porcentagem de redução, obtendo respectivamente os valores máximos de 7,7 e 51,77 na instância 75.

Figura 49 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 10 *Jobs* Com Perturbação 3



Fonte: Do Autor, 2018

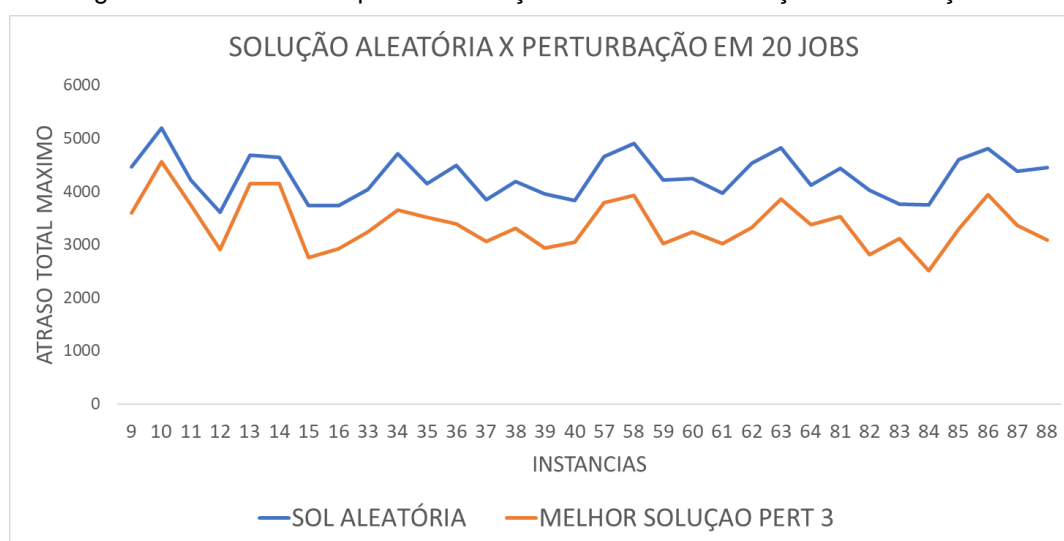
Figura 49 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 10 *Jobs* Com Perturbação 3



Fonte: Do Autor, 2018

As instâncias 20 *jobs* executadas na permutação 3 ilustrados na a Tabela 14 e pela Figura 50, apresentam o tempo médio por iteração em cerca de 58 segundos, e a menor porcentagem de redução com uma média de cerca de 21% .

Figura 50 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 3



Fonte: Do Autor, 2018

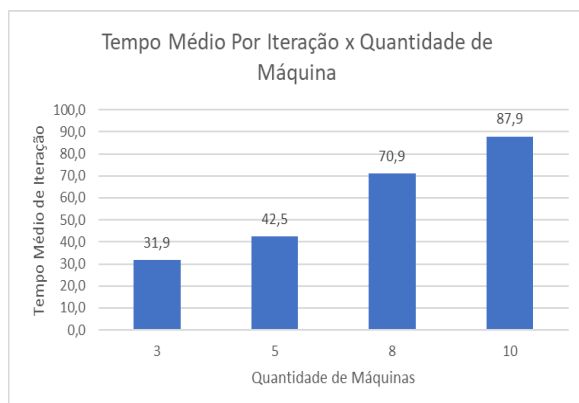
Tabela 14 - Dados das Instâncias de 20 Jobs Com Perturbação 3

INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
9	3	4469	3601	19,4%	631,4	29	21,77
10	3	5189	4561	12,1%	605,6	27	22,43
11	3	4213	3753	10,9%	641,5	17	37,74
12	3	3613	2913	19,4%	601,0	17	35,35
13	3	4685	4144	11,5%	630,2	21	30,01
14	3	4637	4141	10,7%	630,4	19	33,18
15	3	3739	2755	26,3%	617,8	14	44,13
16	3	3730	2915	21,8%	605,1	20	30,26
33	5	4041	3242	19,8%	631,0	15	42,06
34	5	4708	3652	22,4%	634,6	17	37,33
35	5	4141	3507	15,3%	641,4	13	49,34
36	5	4497	3393	24,5%	627,8	14	44,84
37	5	3849	3052	20,7%	658,0	12	54,83
38	5	4190	3304	21,1%	609,9	17	35,88
39	5	3954	2941	25,6%	628,3	15	41,89
40	5	3825	3041	20,5%	612,6	18	34,03
57	8	4655	3783	18,7%	607,7	11	55,24
58	8	4902	3923	20,0%	621,6	10	62,16
59	8	4216	3017	28,4%	627,0	7	89,56
60	8	4248	3235	23,8%	611,1	8	76,38
61	8	3968	3017	24,0%	648,5	10	64,85
62	8	4526	3323	26,6%	651,3	10	65,13
63	8	4820	3852	20,1%	648,6	9	72,06
64	8	4124	3369	18,3%	654,7	8	81,83
81	10	4437	3521	20,6%	636,8	7	90,98
82	10	4019	2808	30,1%	646,1	9	71,79
83	10	3754	3120	16,9%	680,9	6	113,48
84	10	3752	2503	33,3%	665,4	8	83,17
85	10	4598	3288	28,5%	677,1	8	84,64
86	10	4805	3941	18,0%	670,4	9	74,49
87	10	4375	3362	23,2%	615,5	7	87,92
88	10	4443	3092	30,4%	679,7	7	97,09

Fonte: Do Autor, 2018

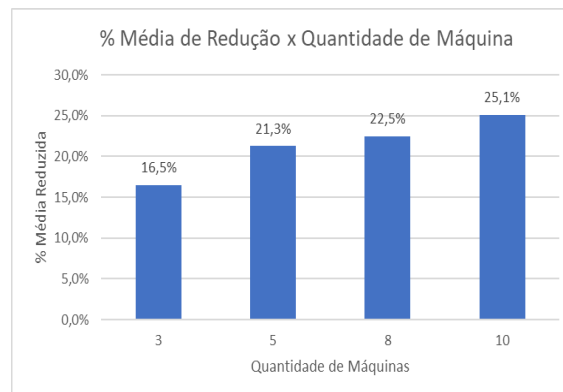
As Figuras 51 e 52 ilustram o comportamento da porcentagem média reduzida das soluções e o tempo médio por iteração de acordo com o aumento no número de máquinas. Mantendo o mesmo padrão das demais perturbações, ou seja, com o aumento das máquinas o tempo de processamento tende a aumentar e a porcentagem de redução também.

Figura 52 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 20 Jobs Com Perturbação 3



Fonte: Do Autor, 2018

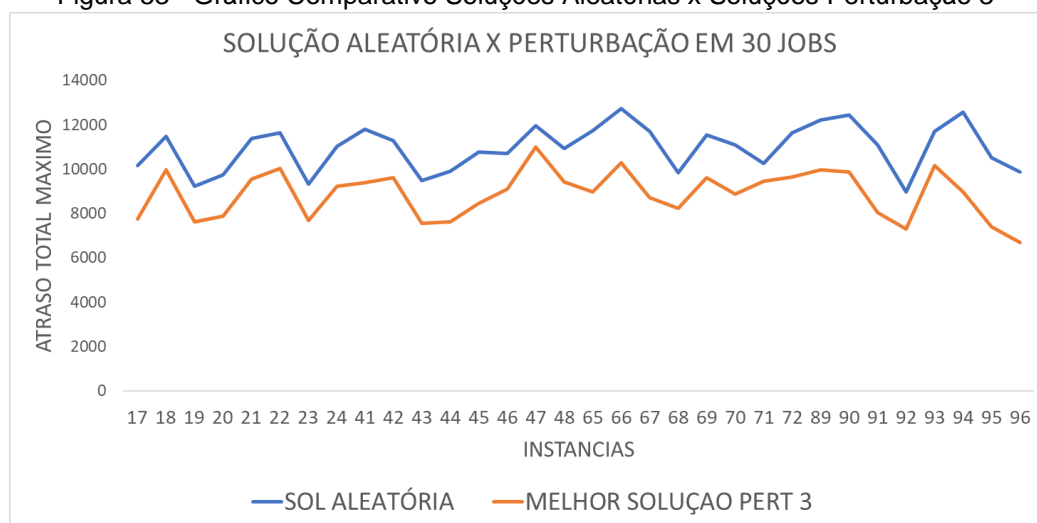
Figura 52 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 20 Jobs Com Perturbação 3



Fonte: Do Autor, 2018

Ilustradas na Tabela 15 e pela Figura 53, as instâncias de 30 jobs um tempo médio por iterações 292,5 e uma redução média de 19% para as instâncias, valores menos satisfatórios do que os obtidos nas demais perturbações. Contudo a análise dos conjuntos de instâncias por máquina, exibidos nas Figuras 54 e 55, ilustra o mesmo padrão de comportamento visto anteriormente, exceto por na máquina 5 o tempo médio de processamento por iterações ter apresentando-se menor do que na máquina 3.

Figura 53 - Gráfico Comparativo Soluções Aleatórias x Soluções Perturbação 3



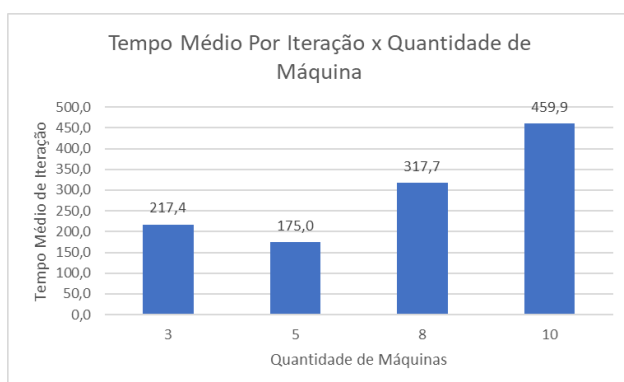
Fonte: Do Autor, 2018

Tabela 15 - Dados das Instâncias de 30 Jobs Com Perturbação 3

INSTANCIA	QTD MAQUINAS	SOL ALEATÓRIA	MELHOR SOLUÇÃO PERT 1	% DE MELHORA	TEMPO DE PROCESSAMENTO	NUM ITERAÇÕES	TEMPO/ITERAÇÃO
17	3	10153	7738	23,8%	728,7	4	182,18
18	3	11469	9963	13,1%	800,0	3	266,66
19	3	9210	7633	17,1%	663,3	3	221,09
20	3	9727	7885	18,9%	644,3	5	128,86
21	3	11375	9550	16,0%	658,7	3	219,57
22	3	11626	10028	13,7%	741,6	3	247,19
23	3	9327	7688	17,6%	750,4	3	250,13
24	3	11028	9222	16,4%	671,1	3	223,71
41	5	11809	9377	20,6%	755,6	5	151,12
42	5	11295	9624	14,8%	746,9	4	186,72
43	5	9494	7548	20,5%	633,2	4	158,29
44	5	9903	7624	23,0%	616,0	4	154,00
45	5	10771	8453	21,5%	628,2	4	157,04
46	5	10701	9106	14,9%	735,8	5	147,17
47	5	11953	11002	8,0%	608,5	3	202,83
48	5	10933	9417	13,9%	971,5	4	242,88
65	8	11721	8978	23,4%	709,8	2	354,88
66	8	12717	10301	19,0%	854,3	3	284,75
67	8	11710	8700	25,7%	646,5	3	215,51
68	8	9836	8232	16,3%	859,1	2	429,56
69	8	11536	9607	16,7%	664,1	2	332,07
70	8	11076	8877	19,9%	722,1	3	240,69
71	8	10253	9458	7,8%	602,8	2	301,41
72	8	11637	9648	17,1%	765,4	2	382,71
89	10	12216	9975	18,3%	785,0	2	392,52
90	10	12444	9855	20,8%	891,1	2	445,57
91	10	11081	8047	27,4%	1054,4	2	527,22
92	10	8968	7286	18,8%	1026,5	2	513,23
93	10	11690	10163	13,1%	785,0	2	392,49
94	10	12561	8984	28,5%	991,6	2	495,80
95	10	10501	7406	29,5%	948,3	2	474,16
96	10	9870	6690	32,2%	875,9	2	437,93

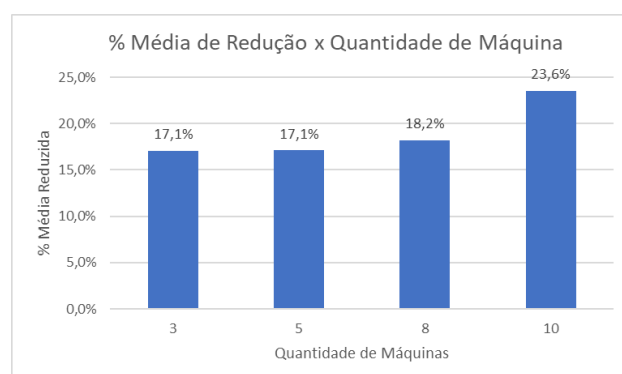
Fonte: Do Autor, 2018

Figura 54 - Gráfico do Tempo Médio Por Iteração x Quantidade de Máquina Para instâncias com 30 Jobs Com Perturbação 3



Fonte: Do Autor, 2018

Figura 55 - Gráfico % Média de Redução x Quantidade de Máquina Para Instâncias com 30 Jobs Com Perturbação 3



Fonte: Do Autor, 2018

5. DISCUSSÃO

Ao comparar os resultados das iterações de 10 *jobs* para cada uma das iterações como mostra a Tabela 16, é possível constatar que em 47% das o melhor resultado foi encontrado nas três perturbações. Contudo em cerca de 84% das vezes a perturbação 2 foi retornou o menor atraso máximo total, contra 72% da perturbação 1 e cerca de 53% da perturbação 3.

Tabela 16 - Análise Geral das Perturbações Para Instâncias de 10 *Jobs*

INSTANCIAS	PERTURBAÇÃO 1	PERTURBAÇÃO 2	PERTURBAÇÃO 3	MELHOR SOLUÇÃO
1	577	577	577	1,2,3
2	479	465	490	2
3	422	422	422	1,2,3
4	293	293	293	1,2,3
5	685	683	685	2
6	448	448	464	1,2,3
7	338	338	338	1,2,3
8	400	390	400	2
25	485	485	485	1,2,3
26	563	563	563	1,2,3
27	291	291	304	1, 2
28	756	746	756	1,2,3
29	329	329	329	1,2,3
30	580	580	580	1,2,3
31	412	412	412	1,2,3
32	468	484	468	1,3
49	145	153	155	1
50	511	511	511	1,2,3
51	317	317	317	1,2,3
52	598	598	598	1,2,3
53	562	556	591	2
54	375	318	380	2
55	362	379	383	1
56	379	321	463	2
73	270	270	285	1,2
74	102	103	102	1,3
75	136	136	136	1,2,3
76	703	703	737	1,2
77	253	253	292	1,2
78	286	261	286	2
79	245	238	289	2
80	508	546	508	1,3

Fonte: Do Autor, 2018

As instâncias de 20 *jobs* não apresentaram resultados iguais entre as perturbações como no caso das iterações de 10 *jobs*. Como ilustrado na Tabela 17 para esse tamanho de instância cerca de 58% das vezes a perturbação 1

apresentou-se como melhor opção, contra 41% da perturbação 2 e 6% da perturbação 3.

Tabela 17 - Análise Geral das Perturbações Para Instâncias de 20 *Jobs*

INSTANCIAS	PERTURBAÇÃO 1	PERTURBAÇÃO 2	PERTURBAÇÃO 3	MELHOR SOLUÇÃO
9	3530	3479	3601	2
10	4411	4278	4561	2
11	3655	3599	3753	2
12	2815	2790	2913	2
13	4142	4130	4144	2
14	3961	3968	4141	1
15	2718	2653	2755	2
16	2875	2839	2915	2
33	2955	3007	3242	1
34	3632	3665	3652	1
35	3284	3242	3507	2
36	3315	3394	3393	1
37	2819	2888	3052	1
38	3305	3120	3304	2
39	2973	2965	2941	3
40	3073	3009	3041	2
57	3519	3598	3783	1
58	3677	3621	3923	2
59	2839	2874	3017	1
60	2792	2836	3235	1
61	2925	2695	3017	1
62	3376	3416	3323	3
63	3742	3584	3852	2
64	3088	3154	3369	1
81	3334	3450	3521	1
82	2595	2747	2808	1
83	2648	2843	3120	1
84	2595	2357	2503	2
85	2880	3308	3288	1
86	3617	3899	3941	1
87	2869	2930	3362	1
88	2851	2880	3092	1

Fonte: Do Autor, 2018

O processamento de instâncias de 30 *jobs* apresentou maior efetividade com a utilização da perturbação 2 com 56% das melhores soluções contra 38% da perturbação 1 e 6% da perturbação 3. Por intermédio da Tabela 18 fica visível a superioridade de resultados com melhor desempenho para a perturbação 2.

Tabela 18 - Análise Geral das Perturbações Para Instâncias de 30 Jobs

INSTANCIAS	PERTURBAÇÃO 1	PERTURBAÇÃO 2	PERTURBAÇÃO 3	MELHOR SOLUÇÃO
17	7598	7352	7738	2
18	9851	9315	9963	2
19	7105	6894	7633	2
20	7623	7381	7885	2
21	9215	9151	9550	2
22	9724	9567	10028	2
23	7565	7224	7688	2
24	8962	8893	9222	2
41	8835	8890	9377	1
42	9023	9146	9624	1
43	7590	7363	7548	2
44	7554	7391	7624	2
45	8015	7865	8453	2
46	8545	8508	9106	2
47	9743	9945	11002	1
48	9401	8940	9417	2
65	8917	9101	8978	1
66	10217	9957	10301	2
67	8590	8514	8700	2
68	7974	8417	8232	1
69	8922	9414	9607	1
70	8255	8467	8877	1
71	9287	9082	9458	2
72	9405	9798	9648	1
89	9622	9873	9975	1
90	9203	9220	9855	1
91	8097	8085	8047	3
92	7009	7146	7286	1
93	9656	9822	10163	1
94	9255	9142	8984	3
95	7825	7139	7406	2
96	6936	6352	6690	2

Fonte: Do Autor, 2018

Mediante a análise dos resultados individuais das iterações a perturbação 2 apresentou-se como a melhor opção de processamento para problemas de *flow shop* que buscam soluções não permutacionais. Assim ao aprofundar a análise foi possível identificar que o tempo médio de processamento das iterações para instâncias que sofreram aplicação do ILS com perturbação 2 apresentaram-se inferiores as demais perturbações em todas os grupos de instâncias como mostra a Tabela 19.

Tabela 19 - Tempo Médio Por Iteração Em Cada Grupo de *Jobs* x Máquinas

Jobs x Máquinas	TEMPO MÉDIO POR ITERAÇÃO		
	PERTURBAÇÃO 1	PERTURBAÇÃO 2	PERTURBAÇÃO 3
10 X 3	1,4	0,7	1,8
10 X 5	2,2	1,0	2,9
10 X 8	4,3	1,8	5,6
10 X 10	5,8	2,5	7,4
20 X 3	19,6	10,2	31,9
20 X 5	27,3	11,2	42,5
20 X 8	41,7	18,4	70,9
20 X 10	55,4	24,8	87,9
30 X 3	102,8	59,4	217,4
30 X 5	323,2	54,6	175,0
30 X 8	187,1	80,0	317,7
30 X 10	241,5	111,6	459,9

Fonte: Do Autor, 2018

A rapidez no processamento das iterações sob a perturbação 2, permitiu um maior número das mesmas o que consecutivamente leva a um maior conjunto de soluções visitadas tendo assim a maior probabilidade de um resultado melhor. Outro fator que influencia o desempenho superior da perturbação dois seria o fato de a mesma consistir em uma terceira vizinhança dentro do ILS, o que intensifica as buscas sempre em volta de um melhor ótimo local.

6. CONCLUSÃO

O presente trabalho desenvolveu um algoritmo para resolução de problemas de sequenciamento em *layouts flow shop*, visando uma solução não permutacional que minimizasse o máximo atraso total de sequenciamentos produtivos. Utilizando como estrutura base para o algoritmo a metaheurística ILS foram desenvolvidos um modelo de solução inicial aleatória, duas classes de vizinhanças denominadas de Vizinhança 1 e Vizinhança 2, que analisavam respectivamente as esferas macro e micro do sequenciamento e três modelos de perturbação sendo estes respectivamente Perturbação Retira Jobs, Perturbação Troca Colunas e Perturbação Inverte Matriz.

Na literatura atual existem várias bases de instâncias para problemas de *flow shop*, contudo não foram encontrados modelos que apresentassem tempos iguais a zero em uma grande proporção. Considerando que neste tipo de situação as soluções permutacionais são mais relevantes, foi desenvolvido um algoritmo gerador de instancias que com base em 48 combinações distintas de *jobs*, máquinas, porcentagem de folga para entrega e probabilidade de *jobs* zerados, gerou 96 instancias que foram processadas posteriormente pelo algoritmo do ILS para cada uma das três perturbações desenvolvidas.

Para analisar as instancias partindo de um mesmo parâmetro cada uma destas foi executada 10 mil vezes dentro do algoritmo de geração de solução aleatórias retornando apenas o melhor resultado de minimização de atraso máximo total encontrado que foi utilizada como base para as comparações. Desta maneira ao realizar a comparação dos dados das perturbações com as soluções aleatórias o modelo matemático criado demonstrou-se superior a qualquer aleatório. Ainda ao comparar as perturbações entre si, constatou-se uma superioridade de resultados para o modelo que continha a perturbação de tipo dois, que realizava a troca de colunas da matriz de *jobs* x máquinas em análise.

A superioridade do algoritmo com Perturbação 2 pode ser explicado devido a mesma ser similar a um modelo de vizinhança pois busca dentre as trocas de colunas da matriz retorna a melhor solução para o atraso máximo total obtida dentro da mesma. Outro fator que pode influenciar sua superioridade é o fato de que seu tempo de processamento foi inferior aos demais, e permitindo que em casos de 30 *jobs* com

10 máquinas fossem realizadas até seis iterações, enquanto as perturbações 1 e 3 realizavam no máximo duas.

De modo geral o algoritmo criado cumpre seu objetivo central, sendo capaz de processar instâncias e criar sequenciamentos produtivos melhores que sequenciamentos aleatórios. Contudo considerando que as instâncias criadas não foram processadas por nenhuma outra metaheurística, não é possível afirmar que os resultados obtidos são os melhores que podem ser encontrados para cada uma destas.

Como propostas de trabalhos futuros, sugere-se como continuidade para o presente trabalho, o processamento das instâncias com o mesmo algoritmo de ILS e perturbações, mas buscando soluções permutacionais ao invés de não permutacionais, processamento das instâncias criadas por outras heurísticas e metaheurística a fim de verificar a efetividade das soluções obtidas no presente trabalho e a aplicação do algoritmo criado em casos reais de sequenciamento.

8. REFERÊNCIAS

- ANTHONY, R. N; GOVINDARAJAN, V. **Management control systems**, Boston, McGrawHill,1998.
- ARENALES, M. ARMENTANO, V.; MORABITO, R. & YANASSE, H. **Pesquisa Operacional**. Rio de Janeiro, Elsevier, 2007.
- BAKER, K. R; TRIETSCH, D., **Principles of sequencing and scheduling**, US, Copyright, 2009.
- BENAVIDES A. J., RITTN M., **Two simple and effective heuristics for minimizing the makespan in non-permutation flow shops**, Computers & Operations Research, 2016.
- BRITO, J. A. M.; MONTENEGRO, F. M. T.; Ochi, L. S., **Um algoritmo ILS para melhoria de eficiência da estratificação estatística**, Anais do XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2009.
- BULMER, M. **Sociological research methods**. London, Macmillan, 1977
- CONWAY R. W., MAXWELL W. L.; MILLER, L. W., **Theory of Scheduling**. Addison-Wesley, Reading, MA, 1967.
- FRAMINAN, J. M.; LEISTEN, R.; GARCÍA, R. R., **Manufacturing Scheduling Systems - An Integrated View on Models, Methods and Tools**, Londres, Springer, 2014.
- KAMPKE E. H.; ARROYO J. E. C. **Busca Local Iterativa Para Resolver O Problema De Programação De Tarefas Em Máquinas Paralelas Com Setup Times Dependentes Da Sequência E De Recursos**, Salvador, ENEGEP, 2009.
- KIESKOSKI, A. **Um Estudo Do Problema De Flow Shop Permutacional: Uma Proposta De Solução Através Da Metaheurística Colônia De Formigas**. Tese de Doutorado: Universidade Federal Do Paraná, Curitiba, 2016.
- LEUNG, J. Y-T. **Handbook Of Scheduling: Algorithms, Models, And Performance Analysis**, New York, CRC Press LLC, 2004.
- LOURENÇO, H.R.; MARTIN, O.; STÜTZLE, T., **Iterated Local Search**, Handbook of Metaheuristics, Kluwer Academic Publishers, Norwell, 2002.
- LUSTOSA, L. J.; MESQUITA, M. A.; QUELHAS, O. L. G.; OLIVEIRA, R. J. **Planejamento e Controle da Produção**. Rio de Janeiro, Elsevier, 2008
- MELO, Everton L., **Meta-heurística Iterated Local Search, GRASP e artificial Bee Colony aplicadas ao Job Shop flexível para minimização do atraso total**. Tese de Doutorado: Universidade de São Paulo, São Paulo, 2014.
- MOCCELLIN, J.V. & SANTOS, M.O, **An Adaptive Hybrid Metaheuristic for Permutation Flowshop Scheduling**. *Journal of Control and Cybernetics*, 29: n. 3, 2000.
- MORTON, Tomas E.; PENTICO, David W.. - **Heuristic Scheduling Systems: With Applications To Production Systems And Project Management**, New York, John Wiley & Sons, 1993.
- PINEDO, M. L, **Scheduling: Theory, Algorithms, and Systems.**, New York, Springer, 2016.
- SULE D. R., **Production Planning and Industrial scheduling: examples, case studies, and applications**, New York, CRC Press LLC, 2008.
- TANDON, M.; CUMMINGS, P. T.; LEVAN, M. D., **Flowshop With Non-Permutation Schedules**, Charlottesville, Copyright, 1991.

TORRES, M. S.; LEITAO, F.; RODRIGUES, L. H.; ANTUNES JR., J. A. V. **Os benefícios da manufatura sincronizada: uma aplicação prática em uma empresa metalmecânica do setor de autopeças**. Porto Alegre, Publicações e SErviços LTDA, 2003

XIAO, Y.Y, YUANB Y., ZHANG R.Q., KONAK A., ***Non-permutation flow shop scheduling with order acceptance and weighted tardiness***, *Appl. Math. Comput*, 2015.