UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

AUGUSTO ALMEIDA DA SILVA

**Heurística Evolutiva para problemas de programação em no-
wait flowshop com tempos de setup**

São Carlos

2012

Augusto Almeida da Silva

**Heurística Evolutiva para problemas de programação em no-
wait flowshop com tempos de setup**

Dissertação apresentada à Escola de Engenharia
de São Carlos da Universidade de São Paulo para
obtenção do Título de Mestre em Engenharia de
Produção.

Área de concentração: Processos e Gestão de
Operações.

Orientador: Prof. Dr. Marcelo Seido Nagano

São Carlos

2012

RESUMO

Este trabalho aborda o problema de *no-wait flowshop* em um ambiente com custos de *setup* apartados dos tempos de processamento, são investigados os casos de *setups* dependentes e independentes da seqüência para *makespan* e *total flowtime*. Diversas aplicações práticas podem ser modeladas sob estas suposições, dentre elas destacamos a indústria química e alimentícia. É proposta uma metaheurística evolutiva baseada em algoritmo genético e *clustering search* e seus resultados são comparados com os métodos de Brown et al (2004), França et al (2006) e Ruiz e Allahverdi (2007) através dos bancos de dados de Ruiz e Stützle (2008) e Ruiz e Allahverdi (2007). Os métodos são avaliados segundo o percentual de sucesso e desvio relativo médio. Os resultados obtidos demonstram a superioridade do método proposto para problemas de grande porte.

Palavras-chave: *No-wait Flowshop*, *Setup* Dependente, *Setup* Independente, *Evolutionary Clustering Search*, *Makespan*, *Total Flowtime*.

ABSTRACT

This work intends to research the no-wait flowshop scheduling problem with setup times separated from the processing costs; the both cases where the sequence is dependent and independent are targeted for makespan and total flowtime. There are numerous practical situations that can be modeled under these assumptions, such as, chemical industry, food processing, etc. A hybrid metaheuristic method based on a genetic algorithm and *clustering search* is proposed and its results are compared to the methods of Brown et al (2004), França et al (2006) e Ruiz e Allahverdi (2007) using the data base from Ruiz e Stützle (2008) and Ruiz e Allahverdi (2007). The methods are evaluated as regarding the success rate and average relative deviation. The results show that the proposed method delivers better solutions for problems with higher complexity.

Key-words: *No-wait Flowshop, Dependent Setup, Independent Setup, Evolutionary Clustering Search, Makespan, Total Flowtime.*

Lista de Símbolos

C_{max}	Duração total da programação (<i>makespan</i>)
$\sum C_j$	Soma das datas de término (<i>total completion time</i>)
$F2$	<i>Flowshop</i> com duas máquinas
$\bar{F}$	Tempo médio de fluxo (<i>mean flowtime</i>)
$\sum F_j$	Tempo de fluxo total (<i>total flowtime</i>)
L_{max}	Atraso ou adiantamento máximo (<i>maximum lateness</i>)
m	número de máquinas do problema
n	número de tarefas do problema
r_j	Datas de liberação
R_{si}	Tempos de remoção
ST_{sd}	Tempos de <i>setup</i> dependentes da seqüência
ST_{si}	Tempos de <i>setup</i> independentes da seqüência

Lista de Siglas

AM	<i>Analyzer Module</i>
ATSP	<i>Asymmetric Traveling Salesman Problem</i>
ATSP-RT	<i>Asymmetric Travelling Salesman Problem with Ready Times</i>
CS	<i>Clustering Search</i>
DPSO	<i>Discrete Particle Swarm Optimization</i>
ECS	<i>Evolutionary Clustering Search</i>
GA	<i>Genetic Algorithm</i>
GRASP	<i>Greedy Randomized Search Procedure</i>
IC	<i>Iterative Clustering</i>
ILS	<i>Iterated Local Search</i>
LPT	<i>Longest Processing Time</i>
LS	<i>Local Search</i>
MILP	<i>Mixed-Integer Linear Programming</i>
MIP	<i>Mixed Integer Programming</i>
NWFS	<i>No-wait Flowshop</i>
PSO	<i>Particle Swarm Optimization</i>
SDST	<i>Sequence Dependent Setup Times</i>
SM	<i>Search Metaheuristic</i>
SPT	<i>Shortest Processing Time</i>
RAI	<i>Recursive Arc Insert</i>
TSP	<i>Traveling Salesman Problem</i>
WIP	<i>Work In Process</i>

Lista de Tabelas

Tabela 1 – Dados do Flowshop.	20
Tabela 2 – Dados do Flowshop com setups separados.....	20
Tabela 3 – Mapeamento dos métodos e problemas.....	53
Tabela 4 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_10_0,5 - Makespan.....	57
Tabela 5 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_10_1,0 - Makespan.....	58
Tabela 6 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_10_1,5 - Makespan	59
Tabela 7 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_100_0,5 - Makespan.....	61
Tabela 8 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_100_1,0 - Makespan.....	62
Tabela 9 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_100_1,5 - Makespan.....	63
Tabela 10 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_10_0,5 - Makespan.	64
Tabela 11 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_10_1,0 - Makespan.	65
Tabela 12 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_10_1,5 - Makespan.	65
Tabela 13 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_100_0,5 - Makespan.	67
Tabela 14 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_100_1,0 - Makespan.	67
Tabela 15 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_100_1,5 - Makespan.	68
Tabela 16 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_10_0,5 - TFT.....	69
Tabela 17 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_10_1,0 - TFT.....	70
Tabela 18 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_10_1,5 - TFT.....	71
Tabela 19 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_100_0,5 - TFT.....	72
Tabela 20 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_100_1,0 - TFT.....	73
Tabela 21 – Percentual de Sucesso e Desvio Rel. Médio SI_SM_100_1,5 - TFT.....	74
Tabela 22 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_10_0,5 - TFT.....	75
Tabela 23 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_10_1,0 - TFT.....	76

Tabela 24 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_10_1,5 - TFT.....	76
Tabela 25 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_100_0,5 - TFT.....	78
Tabela 26 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_100_1,0 - TFT.....	78
Tabela 27 – Percentual de Sucesso e Desvio Rel. Médio SI_LG_100_1,5 - TFT.....	79
Tabela 28 – Percentual de Sucesso e Desvio Rel. Médio SDST10 - Makespan.....	81
Tabela 29 – Percentual de Sucesso e Desvio Rel. Médio SDST50 - Makespan.....	81
Tabela 30 – Percentual de Sucesso e Desvio Rel. Médio SDST100 - Makespan.....	82
Tabela 31 – Percentual de Sucesso e Desvio Rel. Médio SDST125 - Makespan.....	82
Tabela 32 – Percentual de Sucesso e Desvio Rel. Médio SDST10 - TFT.....	84
Tabela 33 – Percentual de Sucesso e Desvio Rel. Médio SDST50 - TFT.....	84
Tabela 34 – Percentual de Sucesso e Desvio Rel. Médio SDST100 - TFT.....	85
Tabela 35 – Percentual de Sucesso e Desvio Rel. Médio SDST125 - TFT.....	85

Lista de Figuras

Figura 1 – Relação entre classes de <i>scheduling</i>	13
Figura 2 – Relação entre classes de complexidade.	16
Figura 3 – Gráfico de Gantt para Flowshop.....	20
Figura 4 – Gráfico de Gantt para Flowshop com setups.....	20
Figura 5 – Gráfico de Gantt para Flowshop em No-Wait.....	28
Figura 6 – Gráfico de Gantt para No-Wait Flowshop com setups.	33
Figura 7 – Fluxo entre Módulos do Clustering Search.	40

Sumário

<i>CAPÍTULO 1 - INTRODUÇÃO</i>	9
1.1 Problemas de <i>Scheduling</i>	100
1.2 Complexidade e Métodos de solução	155
1.3 Objetivos da Pesquisa	177
<i>CAPÍTULO 2 - FLOWSHOP COM TEMPOS DE SETUP</i>	19
2.1 Flowshop com <i>Setup</i> Independente	211
2.1.1 Principais Métodos	211
2.2 Flowshop com <i>Setup</i> Dependente	233
2.2.1 Principais Métodos	244
<i>CAPÍTULO 3 - NO-WAIT FLOWSHOP</i>	277
3.1 Principais Métodos	288
<i>CAPÍTULO 4 - NO-WAIT FLOWSHOP COM TEMPOS DE SETUP</i>	333
4.1 Principais Métodos	344
<i>CAPÍTULO 5 - EVOLUTIONARY CLUSTERING SEARCH</i>	388
5.1 ECS aplicado em No-Wait Flowshop com tempos de setup	541
<i>CAPÍTULO 6 - EXPERIMENTAÇÃO COMPUTACIONAL</i>	51
6.1 Delineamento do Experimento	52
6.2 Processo de Análise	54
<i>CAPÍTULO 7 - ANÁLISE DOS RESULTADOS</i>	51
7.1 No-Wait Flowshop com setup independente	52
7.1.1 Critério de Minimização do <i>Makespan</i>	57
7.1.2 Critério de Minimização do <i>Total Flowtime</i>	69

7.2 No-Wait Flowshop com setup dependente	80
7.2.1 Critério de Minimização do <i>Makespan</i>	80
7.2.2 Critério de Minimização do <i>Total Flowtime</i>	83
<i>CAPÍTULO 8 - CONSIDERAÇÕES FINAIS</i>	86
<i>REFERÊNCIAS</i>	88

CAPÍTULO 1

INTRODUÇÃO

No atual cenário de produção e consumo globalizados, a disputa industrial torna-se cada vez mais acirrada ocasionando uma busca intensa por maior eficiência dos métodos de produção e das técnicas de gerenciamento da atividade industrial. Objetiva-se diferenciar os produtos e serviços prestados de forma a torná-los mais atrativos ao público consumidor. Inseridos neste contexto, o estudo e aperfeiçoamento dos sistemas de produção passaram a ser fundamentais para as organizações se manterem atualizadas e competitivas.

Segundo Wild (1981), um sistema de produção pode ser definido como o arranjo de recursos combinados para a elaboração de bens e/ou serviços. Os sistemas produtivos possuem diversas classificações distintas. Erdmann (2000) define os sistemas em probabilísticos ou determinísticos, de acordo com o caráter da previsibilidade das operações envolvidas. Johnson e Montgomery (1974) propõem uma divisão em três categorias, segundo o processo de fluxo dos materiais no sistema:

- 1) Sistema grande projeto: produtos com alto grau de complexidade e/ou dimensões, freqüentemente produzidos em lotes únicos.
- 2) Sistema contínuo: produtos com alto grau de similaridades entre si, produzidos em larga escala.
- 3) Sistema intermitente: produtos variados que usualmente demandam diversas adaptações entre os estágios de produção. Esta categoria em especial divide-se em dois subgrupos, a saber:

- a) *Flowshop* : ambiente de produção em que todos os produtos seguem o mesmo roteiro ao longo de toda a linha de produção.
- b) *Jobshop* : ambiente de produção em que cada produto tem seu próprio roteiro ao longo da linha de produção.

Dentre os diversos aspectos que envolvem a otimização de um sistema de produção intermitente, o desafio mais complexo é o seqüenciamento de tarefas. A complexidade deve-se ao fato de que suas soluções possuem caráter combinatorial. Por outro lado, uma ordenação adequada das tarefas na linha de produção pode resultar em grandes benefícios para a organização, como economia de tempo e/ou recursos.

1.1 Problemas de *Scheduling*

O processo de *scheduling* consiste em especificar, em uma base de tempo, como os recursos disponíveis serão empregados na realização de tarefas (Pinedo, 2008). Levando-se em consideração a complexidade dos ambientes de produção em que existem diversos itens a serem produzidos e cada item necessita ser processado por várias máquinas distintas, o *scheduling* compreende em determinar as datas de início e fim das operações e qual meio utilizar nos processamentos. No entanto, problemas de programação de tarefas, são muito comuns, segundo Conway et al (1967) tais problemas surgem sempre que as tarefas existentes no sistema admitem uma ordenação, assim como observado no

controle de pouso e decolagem de aeronaves em um aeroporto, clientes em um banco aguardando por atendimento, instruções a serem processadas por um sistema computacional, etc.

Por se tratar de um conjunto de problemas facilmente encontrado no cotidiano muitas vezes a solução surge de maneira casual ou se baseia em conceitos intuitivos, como o de fila por exemplo, que a princípio pode ser uma estratégia justa, porém, na prática, pode gerar uma solução de baixa qualidade, segundo algum critério previamente determinado. Conforme a relevância do *scheduling* aumenta, como o encontrado em uma linha de produção, surge a necessidade de se analisar e aplicar estratégias mais elaboradas que resultem na programação das operações com soluções cada vez melhores.

Como mencionado anteriormente os problemas de *scheduling* são muito distintos e podem possuir características muito particulares. Ainda segundo Conway (1967), cada programação de tarefas pode ser classificada através de um conjunto de quatro parâmetros no formato A/B/C/D:

- A) Tarefas e operações a serem processadas
- B) Número e tipo de máquinas que formam a linha de produção.
- C) Regras que restringem a maneira pela qual as tarefas devem ser processadas.
- D) Critério de avaliação de solução.

Dessa forma os problemas se distinguem pela quantidade e maneira em que as tarefas chegam ao sistema, a quantidade de recursos disponíveis para executá-las, o roteiro em que cada tarefa deverá seguir ao longo de seu processamento e a maneira em que uma solução é avaliada.

O formato em que as tarefas chegam ao sistema o classifica em estático

ou dinâmico.

Chama-se estático o sistema que estava ocioso, recebe simultaneamente um determinado número de tarefas e está apto a iniciar imediatamente o processamento de qualquer uma delas. Nesse caso o parâmetro A representa o número de tarefas a serem processadas.

Classifica-se por dinâmico o sistema em que o processo de chegada das tarefas é contínuo, ou seja, pode haver tarefas que não estão disponíveis ao longo de todo o período de programação. Nesse caso o parâmetro A representa a distribuição de probabilidade que as tarefas devem chegar ao sistema.

Os parâmetros B e C referem-se, respectivamente, aos recursos (máquinas) e o conjunto restrições de processamento do *scheduling* que são determinadas principalmente pelo fluxo que as tarefas apresentam ao longo da linha de produção. Os autores sugerem a seguinte classificação:

a) *Jobshop*: cada tarefa possui seu próprio roteiro ao longo da linha de produção.

b) *Flowshop*: todas as tarefas possuem o mesmo roteiro ao longo da linha de produção.

c) *Openshop*: nenhuma das tarefas possui um roteiro padrão de produção preestabelecido.

d) *Flowshop* Permutacional: caso específico de *flowshop* em que a ordem de processamento das tarefas em todas as máquinas deve ser a mesma.

e) Máquina Única: uma única máquina é alocada para processar o conjunto de tarefas.

f) Máquinas Paralelas: máquinas idênticas são alocadas para processar

o conjunto de tarefas sendo que cada tarefa deve ser processada por uma única máquina.

g) *Jobshop* com Múltiplas Máquinas: caso específico de *Jobshop* em que pode haver mais de uma máquina idêntica para cada estágio de produção sendo que cada tarefa necessita de apenas uma máquina por estágio.

h) *Flowshop* com Múltiplas Máquinas: caso específico de *flowshop* em que pode haver mais de uma máquina idêntica para cada estágio de produção sendo que cada tarefa necessita de apenas uma máquina por estágio.

A Figura 1 exprime a relação entre os diferentes problemas de *scheduling*.

Figura 1 – Relação entre classes de *scheduling* (MACCARTHY; LIU, 1993).

Todo problema de *scheduling* possui um conjunto de soluções que satisfazem todas as restrições do problema também chamadas de soluções factíveis. Existem casos em que as características do sistema são restritivas ao ponto de o conjunto de soluções ser vazio, ou seja, não há maneira de alocar os

recursos disponíveis de forma que todas as restrições sejam satisfeitas. Para tais problemas uma alternativa é reavaliar as regras do modelo e relaxar um subconjunto de restrições menos relevantes até que o problema apresente solução factível. Para os problemas em que o conjunto de soluções é finito e não é trivial, ou seja, existe mais de uma solução viável, surge a necessidade de se determinar dentre as soluções factíveis aquelas que são as mais adequadas segundo algum critério.

Diversas medidas de desempenho, parâmetro D da classificação utilizada, são encontradas na literatura. O critério de avaliação da programação pode ser classificado como único ou múltiplo, segundo o número de variáveis que compõe a função objetivo em questão. Baker (1967) destaca os critérios únicos mais utilizados no estudo de *scheduling*, são eles:

- 1) Duração Total da Programação (*Makespan* – C_{max}): Otimiza a utilização dos recursos disponíveis.
- 2) Tempo Médio de Fluxo (*Mean Flowtime* – $\bar{F}$): Otimiza o tempo de resposta à demanda.
- 3) Atraso Máximo da Programação (Maximum Lateness – L_{max}): Otimiza o tempo máximo que uma tarefa atrasa na programação.

Os problemas com múltiplos critérios também possuem participação relevante na literatura. Em Smith (1956) foi resolvido o problema *flowshop* de em um ambiente de máquina única sujeito ao conjunto de tarefas atrasadas ser nulo, o critério de otimização utilizado foi a minimização do tempo médio de fluxo. Emmons (1975) estende o algoritmo de Smith propondo solução para o problema de múltiplo critério onde se minimiza o tempo médio de fluxo e o atraso máximo

da programação simultaneamente.

1.2 Complexidade e Métodos de solução

A teoria de complexidade computacional é fundamental para o estudo da eficiência dos métodos e a determinação prática de quais tipos e tamanhos de problemas são viáveis de se solucionar através da computação. Em linhas gerais a teoria busca estimar a quantidade de recursos, tais como tempo e memória, que um determinado método utiliza para solucionar um tipo de problema e posteriormente o classifica segundo os custos despendidos (Arora e Barak, 2009).

As principais classes de complexidade são $\mathcal{P}$, $\mathcal{NP}$, $\mathcal{NP}$ - *completo* e $\mathcal{NP}$ - *difícil*. Todos os problemas pertencentes à classe $\mathcal{P}$ possuem um algoritmo em que os custos envolvidos podem ser descritos por um polinômio no formato n^k , onde n é o tamanho do problema e k constante. A classe $\mathcal{NP}$ tem por característica principal o fato de uma solução poder ser verificada em tempo polinomial. Os $\mathcal{NP}$ - *completos* são problemas mais difíceis de que $\mathcal{NP}$ em que caso se reduza todos os problemas de $\mathcal{NP}$ a um problema p , este pertence à $\mathcal{NP}$ - *completo*. Por fim, os problemas pertencentes ao conjunto $\mathcal{NP}$ - *difícil* são pelo menos tão complexos quanto os $\mathcal{NP}$ - *completos*, se todos os problemas pertencentes a $\mathcal{NP}$ e $\mathcal{NP}$ - *completo* podem ser reduzidos a um problema p , este pertence aos $\mathcal{NP}$ - *difíceis*. Ressalta-se que o conjunto $\mathcal{NP}$ - *difícil* é aberto, pois ainda não foram mapeados

todos os problemas pertencentes a essa classe. A Figura 2 ilustra a relação entre as classes de complexidade.

Figura 2 – Relação entre classes de complexidade.

Em Garey e Johnson (1976) é apresentado um estudo detalhado sobre a complexidade computacional dos problemas de *Flowshop* e *Jobshop*, é demonstrado por meio de análise matemática que a otimização de C_{max} em um ambiente de produção *Flowshop* com número de máquinas $m \geq 3$ é pertencente aos $\mathcal{NP}$ - completos. Ressalta-se que o algoritmo proposto por Johnson (1954) soluciona em tempo polinomial o caso em que $m = 2$, comumente denotado por *F2*.

Dentre as estratégias utilizadas para se solucionar os problemas de *scheduling* destacam-se os métodos exatos e heurísticos. Os métodos exatos se baseiam em modelos matemáticos de Programação Inteira Mista e usualmente utilizam estratégias de pesquisa e exploração de soluções do tipo *Branch and Bound*, em que uma árvore com topologia particular é gerada e a utilização de limitantes superiores e inferiores possibilita excluir certos subconjuntos de soluções, reduzindo assim o espaço de busca. Ao final do procedimento determina-se a solução ótima do problema e, muito embora sua aplicação prática esteja

altamente restrita às dimensões do problema, sua relevância na fundamentação teórica de novas abordagens é alta. Métodos heurísticos se apóiam no conhecimento prévio do problema e em estratégias racionais para encontrar soluções computacionalmente viáveis e de boa qualidade. Outra motivação para se utilizar recursos heurísticos é o fato de a modelagem ser um processo de representação de um sistema real e que, em geral, estes modelos utilizam suposições simplificadas e, nesse caso, o ótimo do modelo não é garantido ser o ótimo do cenário original.

Os métodos heurísticos se dividem principalmente em dois tipos, as construtivas e melhorativas. Os algoritmos construtivos geram as soluções iterativamente, sua principal característica é sempre obter uma única solução factível para um problema específico. Heurísticas melhorativas, ou metaheurísticas, partem de uma solução factível e através de processos de perturbações e buscas por vizinhanças aprimoram a solução inicial.

1.3 Objetivos da Pesquisa

Esse trabalho tem como objetivo investigar o problema de programação de tarefas em *No-wait Flowshop* no qual os tempos envolvidos na preparação do maquinário (*setup*) são considerados separadamente dos tempos de processamento das tarefas em si. Por sua vez, este universo é geralmente separado em dois grupos, o primeiro se refere aos problemas em que os *setups* independem da sequência (ST_{si}) de programação, logo, o tempo que será gasto para adequar o

maquinário depende apenas da tarefa a ser processada. O segundo caso modela as situações em que os custos de *setup* são dependentes (ST_{sd}) da sequência de programação, ou seja, o custo de preparação do maquinário para uma tarefa qualquer depende tanto da tarefa a ser processada como da tarefa que a antecede imediatamente na sequência.

É proposta uma meta-heurística híbrida de algoritmo genético e clustering search adaptada para ambos os casos de dependência e independência do *setup*. A eficiência do método será avaliada segundo duas medidas de desempenho, C_{max} e TFT.

Os resultados obtidos serão comparados com os melhores métodos heurísticos reportados na literatura.

Este texto está organizado da seguinte maneira:

- O Capítulo 2 expõe o problema de flowshop com tempos de setup explícitos, é feita uma revisão bibliográfica com os principais métodos para os casos de setup independentes e dependentes;
- O Capítulo 3 apresenta uma revisão bibliográfica sobre o problema no-wait flowshop básico;
- O Capítulo 4 é dedicado ao problema de no-wait flowshop com tempos de setup independentes e dependentes da sequência, é apresentada uma revisão bibliográfica dos principais métodos;
- O Capítulo 5 apresenta o método heurístico genético ECS;
- O Capítulo 6 detalha o projeto de experimentação computacional conduzida.

CAPÍTULO 2

FLOWSHOP COM TEMPOS DE SETUP

Os custos de *setup* se referem a toda preparação necessária para se executar uma tarefa, tais como: o tempo gasto para limpar o ambiente, buscar as ferramentas, separar a matéria prima, ajustar as máquinas, etc (Allahverdi et al., 1999).

A maioria dos problemas de *Flowshop* considera os gastos envolvidos nas operações de preparação das máquinas inclusos nos tempos de processamento das tarefas. Essa suposição simplifica os modelos e pode ser válida quando os custos de *setups* são baixos, independentes da seqüência e difíceis de serem separados do processamento. No entanto, existem casos em que essa simplificação não é apropriada e pode resultar em desperdícios de proporções relevantes ao sistema, tais como observado nos casos em que os custos com *setups* são grandes.

Considerar os tempos envolvidos na preparação das máquinas separadamente dos tempos de processamento possibilita realizar uma série de operações simultaneamente, fato que aumenta a utilização dos recursos presentes no sistema e resulta em economias. Utilizando os dados da Tabela 1, a Figura 2 apresenta o gráfico de Gantt para a seqüência (T1, T2, T3) considerando os tempos de *setup* embutidos no processamento.

	Tarefa 1	Tarefa 2	Tarefa 3
M1	7	8	10
M2	11	7	7
M3	12	8	8

Tabela 1 – Dados do Flowshop.

Figura 3 – Gráfico de Gantt para Flowshop.

A Tabela 2 apresenta explicitamente os valores envolvidos nas operações de *setup* das tarefas consideradas na Tabela 1. A Figura a seguir ilustra a solução para o problema com custos explícitos.

		Tarefa 1	Tarefa 2	Tarefa 3
M1	setup	3	4	4
	processamento	4	4	6
M2		4	3	3
		7	4	4
M3		5	3	4
		7	5	4

Tabela 2 – Dados do Flowshop com *setups* separados. Figura 4 – Gráfico de Gantt para Flowshop com *setups*.

Graficamente é possível observar as grandes possibilidades de redução dos custos de produção ao se considerar os tempos de *setup* separadamente dos tempos processamento em si.

2.1 Flowshop com *Setup* Independente

Existem várias aplicações práticas que motivam o estudo de ambientes de *Flowshop* com tempos de *setup* independentes. Em geral, todo sistema em que o tempo de preparação para se realizar uma tarefa qualquer dependa apenas dela mesma é considerado independente. Um exemplo seria uma indústria alimentícia onde a limpeza e esterilização dos materiais envolvidos no processamento deve ser feita sempre com o mesmo padrão de qualidade, independente do alimento.

2.1.1 Principais Métodos

Yoshida e Hitomi (1979) foram os pioneiros no estudo dos problemas de *scheduling* com custos de preparação apartados do processamento. O modelo clássico $n / 2 / F / C_{max}$ foi estendido para o caso ST_{si} e um método ótimo baseado na regra de Johnson (1954) foi apresentado.

Khurana e Bagga (1985) propuseram um método ótimo para o problema proposto por Yoshida e Hitomi (1979) sujeito a restrição adicional de

existir um prazo máximo para o término do processamento de um subconjunto de tarefas.

Allahverdi (1995) apresentou uma variação para o problema $n / 2 / F / C_{max}$ onde foi considerado também os períodos em que as máquinas não estavam disponíveis (em manutenção), foi estabelecida uma regra de dominância para a minimização do *makespan*.

Sule e Huang (1983) acrescentam uma máquina ao problema de Yoshida e Hitomi (1979) e passam a considerar também os custos envolvidos na remoção do item. Dois métodos heurísticos são propostos para solucionar o problema.

Allahverdi (1997) considerou o problema com duas máquinas presente em Sule e Huang (1983) e demonstram que o algoritmo estocástico dos autores também minimiza o *makespan* quando as máquinas passam por períodos aleatórios de manutenção.

Em Proust (1991) é apresentado um método ótimo baseado nas estratégias de Branch and Bound em $n/m/F, ST_{si}, R_{si}/C_{max}$ para problemas de pequeno porte, além disso, são propostos quatro algoritmos heurísticos para problemas de grande porte. Posteriormente Rajendran and Ziegler (1997) também apresentam uma heurística para o caso em que os tempo de *setup*, processamento e remoção são considerados separadamente.

Outros critérios de otimização também foram propostos para esse tipo de problema. Dileepan e Sen (1991) determinaram duas condições suficientes de otimalidade para o problema $n/2/F, ST_{si}/L_{max}$, apoiados nesses resultados propuseram um algoritmo de *Branch and Bound* e dois métodos heurísticos. O

ambiente $n/2/F, ST_{si}/\bar{F}$ foi investigado em Allahverdi e Aldowaisan (2000) que propõe um método heurístico para sua solução.

2.2 Flowshop com *Setup* Dependente

Diversos ambientes de manufatura requerem um planejamento mais detalhado dos processos quem envolvem as trocas de produtos na linha produção. O problema em que o custo de setup da próxima tarefa em uma máquina depende da tarefa que está sendo processada atualmente é bastante comum. Por exemplo, a temperatura ideal do processo de fusão de metais pode variar de acordo com a liga em questão, neste caso o tempo de adequação da temperatura da caldeira para a próxima liga é dependente da temperatura atual. Em uma fábrica de embalagens o tempo de *setup* depende das similaridades entre os lotes consecutivos, tais como coloração e dimensões.

Wilbrecht e Prescott (1969) verificaram que os *setups* dependentes são especialmente significantes para os casos em que o ambiente de produção funciona em nível muito próximo à sua capacidade total. Em Panwalkar et al. (1973) é apresentada uma pesquisa realizada com gerentes de produção onde foi levantada a quantidade de operações que envolviam *setup* dependentes em suas atividades, cerca de 75% dos entrevistados reportaram que em suas rotinas havia pelo menos uma operação de *Schedule* com *setup* dependente, enquanto aproximadamente 15% informaram possuir atividades em que todas as tarefas requerem *setups* dependentes. A importância de se considerar ambientes com custos de *setup*

dependentes da seqüência para alavancar a eficiência do gerenciamento da produção foi destacada em Wortman (1992). Em Krajewski (1987) foram investigados os fatores que mais influenciam o desempenho de produção de um ambiente e concluíram que independente do sistema produtivo adotado a minimização dos tempos de *setup* somados à otimização dos tamanhos de lotes é a maneira mais eficiente de se reduzir os níveis de estoque e melhorar a qualidade do serviço prestado.

2.2.1 Principais Métodos

Corwin e Esogbue (1974) avaliaram o problema de *flowshop* permutacional em $F2$ para a situação especial onde em uma das máquinas é considerado o *setup* dependente e na outra independente. Eles demonstraram as regras que garantiam a otimalidade das soluções para ambos os casos e propuseram um método exato apoiado nas técnicas de programação dinâmica. Este problema foi estendido por Gupta e Darrow (1985) para o caso em que nas duas máquinas o ambiente é ST_{sd} , dois algoritmos heurísticos foram apresentados para o problema.

Supondo que os custos de *setup* são diretamente proporcionais aos tempos de preparação do maquinário, Uskup e Smith (1975) propuseram um método de solução exata, baseado nas estratégias de Branch and Bound, que

otimiza os custos de *setup* em ambiente *F2* onde os atrasos das tarefas não são permitidos.

Utilizando técnicas de Programação Linear Inteira Mista (Mixed Integer Linear Programming – MILP), Srikar e Ghosh (1986) propuseram uma modelagem matemática para o problema $n/m/F, ST_{sd}/C_{max}$. Embora a formulação possua um número de restrições consideravelmente menor do que os modelos baseados no problema do Caixeiro Viajante (Travelling Sales Problem – TSP), verificou-se que na prática era computacionalmente viável apenas a solução ótima de problemas com até seis máquinas e seis itens. Gupta et al. (1995) apresentaram um algoritmo de Branch and Bound capaz de solucionar o mesmo problema de Srikar e Ghosh (1986) para até 20 itens.

Em Simons (1992) foram apresentados quatro métodos heurísticos para o ambiente $n/m/F, ST_{sd}/C_{max}$, os resultados obtidos superaram três *benchmarks* e indicam que o grau de sofisticação dos métodos não necessariamente acarreta em melhores soluções. Considerando o mesmo problema, Ríos-Mercado e Bard (1998) apresentaram uma heurística construtiva baseada no método NEH proposto em Nawaz et al. (1983) e uma técnica também heurística chamada *Greedy Randomized Adaptive Search Procedure* (GRASP).

O problema de *flowshop* com custos de *setup* dependentes e tempos de remoção dos itens foi abordado em Rajendran e Ziegler (1997), três métodos heurísticos foram propostos assumindo diversas simplificações do problema de *scheduling*. Também é apresentado um método para elevar a qualidade das soluções já obtidas.

Moccellin e Nagano (2007) apresentaram uma característica presente nos problemas de *scheduling* com tempos de *setup* separados em geral, ST_{sd} e ST_{si} .

Esse resultado permite que seja calculado um limitante superior entre o tempo de *setup* e a execução da tarefa propriamente dita, permitindo então a resolução dos problemas de maneira heurística utilizando analogia com o Problema do Caixeiro Viajante Assimétrico (*Asymmetric Traveling Salesman Problem* – ATSP).

CAPÍTULO 3

NO-WAIT FLOWSHOP

O problema clássico de *flowshop* permutacional pressupõe diversas simplificações sobre o ambiente de produção, dentre elas é considerado que existe entre cada par de máquinas consecutivas um estoque de capacidade infinita capaz de abrigar as tarefas por tempo indeterminado até que a próxima máquina esteja disponível para processá-la. Dessa maneira, o ambiente clássico permite que as tarefas sofram interrupções por tempos indeterminados ao longo de toda a linha de produção.

No entanto, uma classe importante de *flowshop* é caracterizada por um ambiente de produção em que não existe estoque intermediário entre as máquinas, essa configuração é também conhecida como *No-Wait Flowshop* (NWFS). As tarefas desta classe necessitam ser processadas ininterruptamente da primeira a última máquina. Inseridos nesse contexto, uma forma prática de garantir que uma tarefa não sofrerá interrupções durante seu processamento é atrasar a sua execução na primeira máquina até que as datas de término de todas as operações coincidam com os inícios de suas sucessoras.

Um exemplo prático que motiva o NWFS, é o caso em que se produz barras de metais fundidos, a liga em alta temperatura não pode aguardar em um estágio intermediário entre a caldeira e o molde, pois a mistura resfriaria perdendo suas propriedades essenciais de manipulação. Outro exemplo surge da indústria alimentícia onde os produtos devem ser enlatados logo após sua

preparação para garantir sua qualidade. Em geral, problemas NWFS surgem sempre que a tarefa em execução pode perder ao longo do tempo alguma propriedade importante para o processamento ou sua posterior utilização. Diversos outros exemplos são levantados e detalhados em Hall and Sriskandarajah (1996), que, além disso, também apresentam o resultado que possibilita a transformação do problema de *No-wait Flowshop* em um ATSP. A figura a seguir ilustra o exemplo de NWFS utilizando os dados da Tabela 1. Como era de se esperar, ao incluir-se uma nova restrição ao sistema a nova solução é no máximo tão boa quanto a solução do problema relaxado.

Figura 5 – Gráfico de Gantt para Flowshop em No-Wait.

O NWFS para mais de duas máquinas é classificado como $\mathcal{NP}$ -difícil em Garey e Johnson (1976), logo, exceto no caso onde $\mathcal{P} = \mathcal{NP}$, a complexidade do problema cresce exponencialmente com o aumento da dimensão do problema.

3.1 Principais Métodos

O problema de NWFS com critério de otimização a soma das datas de término das tarefas ($\sum C_j$) foi investigado pela primeira vez por Deman e Baker

(1974), também foi proposto um algoritmo baseado nas técnicas de Branch and Bound com capacidade para resolver problemas de pequeno porte, 12 tarefas e 6 máquinas.

Adiri e Pohoryles (1982) apresentaram algumas características presentes nas soluções ótimas para problemas $F2$ e demonstraram teoremas que servem de base para algoritmos heurísticos para m máquinas.

Em Rajendran e Chaudhuri (1990) foram apresentados dois algoritmos heurísticos tendo como critério de avaliação de solução o *total flowtime* ($\sum F_j$) das tarefas. Ambos os métodos são compostos de duas etapas, inicialmente é feita uma ordenação das tarefas e na sequência o módulo de inserção é responsável por gerar a solução final.

Van Der Veen e Van Dal (1991) demonstraram que o problema para m máquinas pode ser solucionado em tempo computacional polinomial quando as matrizes de tempo de processamento estão ordenadas, ou seja, $p_{ki} \leq p_{k+1i}$, para $k \leq m$ máquinas e $i \leq n$ tarefas.

Gangadharan e Rajendran (1993) propuseram dois métodos heurísticos para problemas NWFS visando minimizar o C_{max} . Os algoritmos são baseados em duas estratégias, inicialmente um método é utilizado para obter uma solução básica, chamada de semente, posteriormente os resultados são alavancados através de técnicas melhorativas.

Bertolissi (1999) considerou o critério de otimização $\sum F_j$ em um método heurístico de fase única. O método consiste no cálculo do *flowtime* parcial de cada par de tarefas consecutivas, e em seguida é obtido uma ordenação das tarefas. Os resultados da experimentação computacional mostraram que o método

proposto não obtém soluções melhores que os métodos de Rajendran e Chaudhuri (1990).

Bertolissi (2000) apresentou um novo método heurístico, composto por dois módulos. O primeiro passo foi inspirado em seu trabalho anterior em Bertolissi (1999). Na fase complementar, foi utilizado o procedimento similar de inserção de tarefas apresentada em Rajendran e Chaudhuri (1990). Os resultados computacionais mostraram que a heurística é superior à de Rajendran e Chaudhuri (1990).

Fink e Voß (2003) avaliaram os diferentes tipos de métodos heurísticos, como Busca Tabu, *Steepest Descent* e *Simulated Annealing*, de um ponto de vista prático, examinando a relação custo benefício entre esforço computacional e qualidade da solução. Concluíram que a escolha do método mais adequado a se utilizar depende muito das particularidades do problema em questão.

Allahverdi e Aldowaisan (2004) investigaram o problema onde foi considerada uma soma ponderada de dois critérios ($\alpha \times C_{max} + \beta \times L_{max}$), ressalta-se que o critério transforma-se em único sempre que um dos ponderadores é igual a zero. Um algoritmo heurístico foi apresentado além de um método de Branch and Bound e uma Regra de Dominância (Dominance Rule – DR).

Aldowaisan e Allahverdi (2004) apresentaram quatro novos métodos heurísticos (PH1, PH2, PH3 e PH4) para o problema NWFS utilizando o critério $\sum F_j$ para m máquinas. Também foi proposta uma heurística melhorativa de solução utilizando-se a vizinhança de permutação de pares de tarefas. O método PH1(p), composto da heurística PH1 acrescida do procedimento melhorativo, foi o

que apresentou os melhores resultados quando comparados com os métodos de Rajendran e Chaudhuri (1990) e o método apresentados em Chen et al. (1996).

Pan et al. (2007a) é proposto um método baseado nas técnicas de Enxame de Partículas (*Particle Swarm Optimization* – PSO) tendo como critério de otimização tanto a minimização C_{max} como $\sum F_j$, avaliados separadamente. A estratégia considera as permutações de tarefas como partículas e atualiza sua posição baseada em um domínio discretizado. A solução encontrada é então aperfeiçoada de um método de busca por vizinhança.

Pan et al. (2007b) apresenta uma extensão de seu trabalho anterior (Pan et al., 2007a). O método anterior foi hibridizado com uma busca local baseado na estratégia de inserção por vizinhança. Os resultados obtidos com a integração das heurísticas foram superiores aos encontrados no PSO proposto em Pan et al. (2007a).

Em Nagano et al. (2008) é considerado o problema NWFS para m máquinas buscando a minimização do C_{max} . É proposto um algoritmo genético construtivo (Constructive Genetic Algorithm – CGA) que utiliza a informação genética contida em dois indivíduos para gerar um terceiro elemento. O método foi hibridizado com uma busca local por inserção de vizinhança chamada IH. Os resultados obtidos foram comparados com outros métodos genéticos, de *tabu search* e *simulated annealing* e se mostraram bastante competitivos e promissores.

Framinan et al. (2010) apresentou uma nova heurística construtiva com critério de otimização $\sum F_j$. O método se baseia na analogia de um problema $F2$ onde se procura selecionar um candidato a ser inserido no final da seqüência parcial. A solução parcial encontrada é então submetida a um processo de inserção em diversas posições distintas através de inserções e trocas por vizinhança.

Um método híbrido de Algoritmo Genético (Genetic Algorithm – GA) e Busca Local Variável (Variable Neighborhood Search – VNS) foi proposto por Jarboui et al. (2010) para minimizar o C_{max} e $\sum F_j$, considerados separadamente. A idéia principal do VNS é permitir a mudança sistemática da vizinhança pertencente à melhor solução encontrada (incumbente) através de buscas locais aleatórias.

Outro algoritmo de híbrido de GA e busca local foi proposto recentemente por Tseng e Lin (2010) onde o objetivo da heurística é a minimização de C_{max} . O procedimento de busca local combina dois métodos distintos, uma por inserção, responsável por investigar pequenas vizinhanças, e outra por inserção com corte e reconstrução, responsável por fazer buscas em grandes regiões.

CAPÍTULO 4

NO-WAIT *FLOWSHOP* COM TEMPOS DE SETUP

Uma classe importante de problemas de seqüenciamento é encontrada quando em problemas de NWFS se acrescenta a variável em que o custo de *setup* é considerado separadamente do tempo de processamento. Esse cenário composto implica que a preparação do ambiente de produção deve ser realizada antes mesmo de a tarefa chegar à máquina, isso se deve ao fato de que em todo ambiente *no-wait* a espera entre etapas de produção não é permitida. Dessa maneira, pode-se concluir que a simples combinação dos ambientes, por si só, enfatiza a condição de *no-wait* no processo.

Exemplos práticos deste cenário são facilmente encontrados na indústria química em que, dada a propriedade de instabilidade de diversos elementos, o processamento deve ocorrer o mais rápido possível. Nesta configuração fica evidente a possível necessidade de se reter um processo antes de seu início para que todos os instrumentos envolvidos estejam previamente adequados. A figura 6 ilustra esse ambiente com os dados apresentados na Tabela 2.

Figura 6 – Gráfico de Gantt para No-Wait Flowshop com setups.

4.1 Principais Métodos

Em Stafford e Tseng (1990) foram apresentados dois modelos de programação inteira mista (Mixed Integer Linear Programming – MILP) para problemas de NWFS com *setups* separados. Uma formulação matemática trata a configuração em que a preparação do maquinário é ST_{si} e outra, mais elaborada, considera o ambiente ST_{sd} . O critério de otimização para ambos foi C_{max} .

Aldowaisan e Allahverdi (1998) investigaram o ambiente de produção $n/2/no - wait, ST_{si}/\sum F_j$. Foram apresentados três resultados matemáticos, o primeiro estabelece um critério de eliminação que serve de fundamento para um método heurístico proposto no mesmo trabalho, este teorema também pode ser utilizado para melhorar a eficiência dos métodos exatos de Branch and Bound. Os outros dois resultados consistem em critérios de eliminação para casos particulares e suas aplicações permitem obter soluções ótimas para os problemas.

Em Bianco, Dell’Olmo e Giordani (1999) é demonstrado que a configuração $n/m/no - wait, ST_{si}/C_{max}$ corresponde a um caso específico do problema de ATSP onde são consideradas as datas de liberação. É realizada uma análise de complexidade computacional e conclui-se que o problema pertence a classe dos $\mathcal{NP}$ -*diffceis* mesmo quando são desconsideradas as datas de liberação. Além de dois métodos heurísticos, a formulação matemática e dois *lower bounds* também são apresentados.

Aldowaisan e Allahverdi (2000) estabeleceram três seqüências ótimas para casos específicos de $n/3/no - wait, ST_{si}/\sum F_j$ além de uma regra de

dominância de tarefas para o caso geral. Cinco métodos heurísticos foram propostos e avaliados para problemas de portes variados. O algoritmo H1 determina a seqüência de maneira dinâmica, escolhe-se dentre as tarefas ainda não seqüenciadas aquela que minimiza o $\sum F_j$ da solução parcial. H2 e H3 utilizam a solução de H1 como semente e estabelecem a seqüência final através de sucessivas inserções das tarefas ainda não seqüenciadas na solução parcial. A heurística H4 utiliza a mesma estratégia de H3 apenas com uma mudança na solução inicial. Por fim, H5 repete o procedimento de H4 por dez vezes utilizando a solução de H4 como semente para a primeira repetição e a solução obtida serve de semente para a próxima iteração.

Aldowaisan e Allahverdi (2001) apresentaram soluções ótimas para dois casos especiais de $n/2/no - wait, ST_{sd}/\sum C_j$ além de uma regra de dominância que pode ser aplicada para casos mais genéricos. Métodos heurísticos com tempo computacional polinomial foram construídos, eles se dividem em duas etapas, inicialmente uma solução rápida é encontrada e posteriormente técnicas melhorativas são aplicadas a fim de determinar a solução final.

Em Brown et al (2004) foi investigado o problema de *NWFS* para n tarefas e m máquinas em ambiente ST_{si} , foram considerados $\sum F_j$ e C_{max} como medidas de desempenho. Os autores apresentaram um método heurístico que divide problemas de grande porte em problemas menores e de fácil solução. A proposta da heurística TRIPS é examinar todas as triplas de tarefas ainda não seqüenciadas, escolhe-se o subconjunto $\{x_i, x_j, x_k\}$ tal que o critério de desempenho seja o melhor. Posteriormente, insere-se a tarefa x_i no final da seqüência parcial e reinicia-se o procedimento. O algoritmo se repete até que o

conjunto das não seqüenciadas possua apenas três tarefas, determina-se então qual a tripla que minimiza o critério de otimização e a insere no final da seqüência.

Em França et al (2006) foi apresentado um algoritmo genético para o problema de *no-wait* com ST_{sd} e datas de liberação das tarefas, C_{max} foi o critério de desempenho utilizado. A estratégia utilizada foi organizar os indivíduos em uma estrutura hierárquica de árvore ternária de profundidade três, ou seja, ao todo o sistema possui treze elementos. Estes elementos estão organizados segundo o critério de desempenho, técnicas de recombinação genética são utilizadas para que os indivíduos interajam entre si gerando novas soluções. O método é hibridizado com um esquema recursivo de inserção por arco (Recursive Arc Insertion – RAI). Nesse contexto, a eficácia da busca local é primordial para a obtenção de boas soluções uma vez que aproximadamente 90% dos recursos computacionais são direcionados a ela.

Ruiz e Allahverdi (2007) estabeleceram uma regra de dominância eficiente para o problema $n/4/no - wait, ST_{si}/\sum C_j$, muito embora não haja demonstração matemática, testes empíricos sugerem que a relação pode ser aplicada para o caso genérico de m máquinas. Os cinco métodos heurísticos encontrados em Aldowaisan e Allahverdi (2000) foram estendidos para o ambiente com quatro máquinas. Além disso, foram apresentados dois métodos de buscas locais estocásticas, sendo um deles baseado em busca local iterativa (*Iterated Local Search* - ILS).

Em Araújo e Nagano (2010) foi apresentado um método para o cálculo do período de ociosidade das máquinas entre tarefas (GAP) presentes nos problemas de $n/m/no - wait, ST_{sd}/C_{max}$. Baseado nessa estrutura, um método

heurístico chamado GAPH (GAP Heuristic) foi desenvolvido. São construídas quatro seqüências simultaneamente através dos cálculos de GAP e critérios de inserções distintos, no final, é adotada como a solução a seqüência que apresenta o menor *makespan*.

CAPÍTULO 5

EVOLUTIONARY CLUSTERING SEARCH

O processo de *clustering* consiste, basicamente, em agrupar conjuntos de objetos de forma que os indivíduos pertencentes a um mesmo grupo apresentem forte semelhança entre si. O algoritmo de Clustering Search (CS), proposto em Oliveira e Lorena (2004), aplica técnicas de agrupamento que possibilitam a construção de conjuntos de soluções promissoras para problemas de otimização numérica. A identificação precoce dessas regiões permite direcionar de maneira mais efetiva os recursos computacionais disponíveis, gerando uma economia de processamento que possibilita melhorar a qualidade das soluções. Uma região pode ser interpretada como um subespaço de busca definido por uma relação de vizinhança entre soluções.

No método *Evolutionary Clustering Search* (ECS), o processo de *clustering* é executado paralelamente a um algoritmo genético, identificando grupos de indivíduos que merecem uma atenção especial. No método CS tradicional, o algoritmo genético é substituído por uma metaheurística qualquer tal como GRASP, *Simulated Annealing*, *Tabu Search*, dentre outras.

O CS visa localizar áreas promissoras através da divisão do espaço de soluções em *clusters*. Um *cluster* pode ser definido como a tripla $G = (c, r, \beta)$ onde c , r e β são, respectivamente, o centro, raio e a estratégia de exploração associada ao *cluster*.

O centro do *cluster* c é definido como a solução que representa o conjunto, identificando a sua localização dentro do espaço de soluções. Inicialmente, o centro de um *cluster* pode ser obtido aleatoriamente, mas com o passar do tempo, eles tendem a convergir em torno das melhores soluções presentes no subespaço. O raio r estabelece a distância máxima em que as soluções podem ser associadas ao *cluster*; esse intervalo é medido a partir da solução c . A estratégia de busca β define a intensidade da técnica de exploração na qual as soluções interagem entre si gerando novos indivíduos.

O método *CS* é composto de quatro módulos distintos, a saber:

- Metaheurística (*Search Metaheuristic – SM*);
- Agrupador Iterativo (*Iterative Clustering – IC*);
- Analisador de Agrupamento (*Analyzer Module – AM*);
- Algoritmo de Otimização (*Local Search – LS*).

A Figura 7 ilustra a maneira em que os componentes do método interagem entre si.

O módulo *SM* funciona de maneira independente dos outros componentes e tem como objetivo abastecer com novas soluções o processo de agrupamento.

Figura 7 – Fluxo entre Módulos do Clustering Search.

O componente *IC* possui a tarefa de agrupar as soluções similares em *clusters*. Também é responsável por identificar o indivíduo *c* que melhor representa o conjunto. O elemento é implementado como um processo *online* em que cada nova solução encaminhada à *IC* é devidamente agrupada no *cluster* em que ela mais se assemelhe ou um novo cluster é gerado para lhe abrigar. Ressalta-se que um número máximo de clusters é previamente estipulado com o intuito de evitar que sejam criados *clusters* deliberadamente. Para o bom funcionamento do algoritmo é essencial definir inicialmente uma métrica de similaridade que viabilize a comparação efetiva entre soluções.

O módulo *AM* avalia regularmente a qualidade dos *clusters* existentes no sistema, ressaltando aqueles subespaços mais promissores. Para tanto, é definida a propriedade δ_i que representa a densidade de um *cluster* qualquer, essa

característica é capaz de medir o volume de movimentação realizada dentro de um agrupamento. Por simplicidade, δ_i pode ser definido como o número de novos indivíduos associados ao *cluster* i . Dessa forma, sempre que um padrão de solução seja predominantemente gerado no sistema, a densidade do agrupamento que o abriga necessariamente aumenta, indicando que tal região deve ser investigada posteriormente com mais cautela.

Por fim, o componente *LS* é responsável por investigar com maior rigor as supostas regiões promissoras indicadas pelo módulo *AM*. Este procedimento consiste em aplicar uma busca local no centro do *cluster*.

5.1 ECS aplicado em No-wait Flowshop com tempos de setup

Inspirado no algoritmo ECS apresentado por Filho et al (2007) para solucionar o problema clássico de *flowshop* permutacional com objetivo de minimizar o TFT, foi adaptada a metaheurística que combina algoritmo genético e CS para problemas de no-wait *flowshop* com tempos de *setup*. Conforme mencionado anteriormente, o ECS utiliza um algoritmo genético (Genetics Algorithm – GA) que corresponde ao módulo SM do CS, esse algoritmo é responsável por gerar soluções que permitam aos outros componentes do CS realizar buscas por regiões promissoras.

O método Evolutionary Clustering Search para *no-wait flowshop* com tempos de *setup* (ECS_NSL) proposto neste trabalho pode ser representado pelo seguinte pseudocódigo.

Procedimento ECS_NSL () {

```

1  Inicializar população P;
2  Inicializar conjunto C de clusters;
3  Enquanto (critério de parada = Falso) {
4      Enquanto (i < novos_indivíduos) {
5          Base ← Escolha dentro os 40% melhores de P;
6          Guide ← Escolha qualquer indivíduo de P;
7          Offspring ← Recombine(Base, Guide);
8          SLS (Offspring);
9          Se (aleatório(0; 100) <= 60)
10             LS1 (Offspring);
11             Se (insira_em_P (Offspring) = Verdadeiro)
12                 IC (Offspring);
13         i ← i + 1;
14     }
15     Para cada cluster c ∈ C{
16         Se (AM (c))
17             LO (c);
18     }
19 }
```

O método ECS_NSL será detalhado na sequência iniciando pelo seu componente SM.

Um vetor ordenado de n posições foi utilizado no GA como forma de representação de cromossomos, dessa forma, cada posição do vetor armazena uma tarefa que integra a programação da produção.

A qualidade dos indivíduos que compõe a população inicial é de extrema importância para o desempenho do método. A fim de garantir esta

característica da população inicial (linha 1 do Procedimento ECS_NSL), foi desenvolvida uma heurística que resulta em uma solução de qualidade aceitável com baixos custos computacionais. A solução inicial é obtida através de um procedimento de preenchimento de uma solução parcial que inicialmente se encontra vazia. O primeiro passo consiste em escolher, dentre as tarefas ainda não programadas, aquela que caso inserida na primeira posição disponível da solução parcial minimizaria o critério de avaliação. Remove-se essa tarefa do conjunto de tarefas ainda não programadas e repete-se o procedimento até que o conjunto de tarefas ainda não programadas esteja vazio.

O primeiro elemento da população P é o resultado da aplicação deste método seguido por um procedimento de busca local (Standard Local Search – SLS). O algoritmo do SLS é inspirado na busca local presente em Ruiz e Allahverdi (2007) e pode ser resumido da seguinte maneira.

```

SLS (solução_atual) {
    cs ← solução_atual;
    x ← ∅;
    x(1) ← cs(1);
    Para k ← 2 até n {
        testar tarefa cs(k) em todas posições de 1 até k de x;
        inserir cs(k) na posição de x que minimizar o critério de avaliação;
    }
    Retorna(x);
}

```

As outras $n-1$ soluções foram obtidas através de uma perturbação simples aplicada à solução inicial. Essa perturbação segue a ideia apresentada por

Stützle (1998) em que são realizadas duas trocas de posicionamento entre as tarefas. A primeira consiste na seleção aleatória de uma tarefa qualquer da programação, em seguida, permuta-se esta tarefa pela que a sucede na programação. A segunda movimentação se dá pela troca de posições de duas tarefas quaisquer escolhidas de maneira aleatória. Com a intenção de evitar mudanças drásticas na solução inicial faz-se necessário garantir que as duas tarefas que serão trocadas estejam separadas ao máximo de $\max(n/50; 30)$ posições entre si e o procedimento de perturbação é aplicado apenas uma única vez para cada solução.

Após exaustivos testes computacionais a população inicial foi fixada em $(n + m)$ indivíduos. Este parâmetro busca equilibrar o custo computacional despendido, a diversidade da populacional e a qualidade de seus indivíduos.

A rotina de inserção de indivíduos na população é responsável por manter a população ordenada de maneira não decrescente segundo o critério de otimização adotado. Além disso, é atribuição da rotina garantir que na população existirá apenas uma cópia de cada indivíduo.

A distância entre os indivíduos e o centro do um *cluster* é estipulada como sendo o número necessário de trocas de tarefas para transformar este indivíduo em uma solução idêntica ao centro do *cluster*. Dessa forma, todas as posições do indivíduo são comparadas com a posição correspondente da solução do centro de um *cluster*, a cada tarefa não coincidente encontrada aplica-se uma troca de tarefas tal que a sequencia parcial do individuo passe a ser idêntica a do centro de *cluster*. Após efetuar todas as trocas necessárias a sequencia do indivíduo se torna igual a do centro de *cluster*, o número trocas realizadas é considerado como a medida de distância.

Os *clusters* iniciais do método ECS_NSL (linha 2 do procedimento ECS_NSL) são formados através da varredura da população, do seu melhor indivíduo ao pior, criando-se novos *clusters* ou agrupando tais indivíduos em *clusters* previamente criados. Um novo *cluster* é criado toda vez que distância entre um indivíduo e o centro de cada *cluster* já existente for maior que $r = 0.85 \times n$, neste caso o próprio indivíduo é adotado como centro do novo *cluster*. Caso contrário, o indivíduo é assimilado ao *cluster* de menor distância. O parâmetro r foi escolhido após diversos testes buscando manter-se um número razoável de *clusters*.

A fase de construção dos *clusters* iniciais termina assim que toda a população já tenha sido assimilada por algum *cluster* ou quando o número máximo de 200 *clusters* for alcançado, nesse último caso, o algoritmo encerra o processo de inserção de novos indivíduos e dirige-se para a próxima etapa do método. O valor do número máximo de *clusters* foi estipulado após vários testes para diversas classes de problemas.

O tempo de execução do método foi considerado como sendo o critério de parada neste trabalho (linha 3 do procedimento ECS_NSL), adotou-se o parâmetro $10 \times n \times m$ milissegundos. O parâmetro foi ajustado em uma tentativa de equilibrar o custo despendido na obtenção da população inicial e a aplicação dos módulos melhorativos do método.

Após submeter o método GA a diversos testes foi observado que o algoritmo produziu melhores resultados quando programado para gerar 20 novos indivíduos a cada iteração (linha 4 do procedimento ECS_NSL), quando parametrizado para produzir menos indivíduos por iteração verificou-se que o método tende a convergir precocemente, por outro lado, com muitos novos indivíduos por iteração o método tende a desprender muito esforço computacional nesta etapa resultando em

soluções de baixa qualidade. Cada uma dessas soluções são candidatas a fazer parte da população.

A produção de novos indivíduos (linhas 5 a 7 do procedimento ECS_NSL) foi construída selecionando-se aleatoriamente dois indivíduos presentes na população, o primeiro antepassado chamado *base* deve pertencer ao grupo dos 40% melhores indivíduos de toda a população, o segundo antepassado chamado *guide* é selecionado em toda a população sem restrições. Após seleção dos antepassados aplica-se um processo de recombinação proposto por Syswerda (1989) conhecido como *Block Order Crossover* (BOX) representado na Figura 7, este processo consiste na cópia de genes de cada antepassado gerando assim um único descendente. Neste trabalho o descendente foi obtido com 50% de seus genes vindos de cada um dos antepassados.

Figura 8 – Esquema do processo de recombinação.

Após o processo de recombinação o descendente é submetido à busca local SLS (linha 8 do procedimento ECS_NSL). Esta estratégia foi adotada por se tratar de um processo de baixo custo computacional e pela sua eficiência em melhorar a qualidade do indivíduo que está prestes a ser inserido na população.

O indivíduo é então testado estatisticamente de forma que em 60% dos casos seja submetido a outro processo de busca local chamado de LS1 (linha 9 do procedimento ECS_NSL). Por se tratar de um processo exaustivo e consequentemente computacionalmente custoso, o procedimento não é aplicado em todos os indivíduos, o parâmetro 0,6 foi estabelecido após diversos testes computacionais, verificou-se que para probabilidades menores a baixa incidência da aplicação do procedimento resultou em soluções de baixa qualidade, por outro lado, probabilidades maiores resultaram em perda demasiada de tempo prejudicando assim a qualidade da solução final.

Procedimento LS1 (solução atual) {

cs $\leftarrow$ solução atual;

stop $\leftarrow$ falso;

Enquanto (stop = falso) {

P $\leftarrow$ Vizinha de permutação (cs);

sp $\leftarrow$ Primeiro s em P tal que valor(s) < valor(cs), ou valor(s) < valor(t) para todo t em P ;

I $\leftarrow$ Vizinha de Inserção (cs);

si $\leftarrow$ Primeiro s em I tal que valor(s) < valor(cs), ou valor(s) < valor(t) para todo t em I;

Se (valor(sp) < min(valor(si), valor(cs)))

cs $\leftarrow$ sp;

Caso contrário {

Se (valor(si) < min(valor(sp), valor(cs)))

cs $\leftarrow$ si;

Caso contrário

stop $\leftarrow$ true;

}

}

Retorna (cs);

}

Este procedimento se utiliza de dois tipos de buscas de vizinhança a permutacional e a de inserção. A vizinhança de permutação de um indivíduo é obtida através da permutação de todos os pares de tarefas resultando em $n \times (n - 1)/2$ indivíduos distintos. Já a vizinhança de inserção é obtida através da remoção de cada um dos genes de suas posições originais e os inserindo em todas as outras posições possíveis do cromossomo, produzindo $n \times (n - 1)$ indivíduos distintos. Posteriormente este indivíduo é inserido na população na posição referente ao valor obtido pelo critério de avaliação, consequentemente o pior indivíduo de toda a população é eliminado.

Os indivíduos que são de fato inseridos na população passam pelo componente IC do ECS_NSL. A assimilação de um indivíduo por um *cluster* se dá através do processo de *Path Relinking* apresentado por Glover (1996). Iniciando-se através de um indivíduo, sucessivas trocas de tarefas são executadas até que o cromossomo deste indivíduo seja idêntico ao do centro do *cluster*. O par de genes escolhido para a permuta é aquele que melhorar, ou prejudicar menos o critério de avaliação. A nova configuração do cromossomo é avaliada a cada permuta de genes, ao final da transformação caso o indivíduo resultante ou algum cromossomo intermediário obtido durante o processo de transformação apresentar solução de melhor qualidade que o centro do *cluster*, o centro do *cluster* é deslocado (ou substituído) por esta nova configuração. Caso uma melhora deste nível não seja observada, o centro do *cluster* permanece inalterado.

O componente IC do método (linha 12 do procedimento ECS_NSL) busca pelos *clusters* que estejam dentro do raio r do indivíduo. Assim que tal *cluster* é encontrado o indivíduo é assimilado, caso contrário, um novo *cluster* é criado e o

próprio indivíduo passa a representar o seu centro. Novos *clusters* são gerados apenas em caso de o número total de *clusters* ativos no sistema não exceder 200. Foi observado através de testes que para as primeiras iterações do ECS_NSL o número de *clusters* tende a aumentar rapidamente e após alcançar um pico o número de *clusters* tende a diminuir gradativamente, esse efeito é se deve ao fato de o método eliminar os *clusters* menos ativos.

Após gerar novos indivíduos, melhorá-los e inseri-los na população, o método ECS_NSL aciona seu componente AM. Trata-se de um procedimento de análise de *clusters* que realiza duas tarefas: remover *clusters* que não assimilaram novos indivíduos nas últimas 5 iterações e para cada *cluster* que efetivamente assimilou algum indivíduo na iteração atual é executado um processo de otimização chamado LS2, correspondente ao componente LO do método ECS_NSL (Lina 17 do procedimento ECS_NSL).

Procedimento LS2 (solução atual) {

cs ← solução atual;

stop ← falso;

Enquanto (stop = falso) {

I ← Vizinhaça de inserção (cs);

si ← Primeiro s de I tal que valor(s) < valor(cs) ou

valor(s) < valor(t) para todo t de I;

Se valor(si) < valor(cs) então

cs ← si;

P ← Vizinhaça de permutação (cs);

sp ← Primeiro s de P tal que valor(s) < valor(cs) ou

valor(s) < valor(t) para todo t de P;

Se(valor(sp) < valor(cs))

cs ← sp;

Caso contrário {

```

    Pnh ← Vizinhaça de permutação (cs);
    sp ← Buscar Pnh até que sp seja melhor que cs, ou
    sp seja melhor de Pnh;
    Se (valor(sp) < valor(cs))
        cs ← sp;
    Caso contrário
        stop ← verdadeiro;
    }
}
Retorna (cs);
}

```

A solução final produzida pelo algoritmo é o centro de *cluster* com melhor valor segundo o critério de avaliação adotado.

CAPÍTULO 6

EXPERIMENTAÇÃO COMPUTACIONAL

A experimentação computacional neste trabalho tem por objetivo avaliar o desempenho do método proposto ECS para 4 problemas de *scheduling*, são eles:

- 1) NWFS com *setup* independente para TFT;
- 2) NWFS com *setup* independente para *makespan*;
- 3) NWFS com *setup* dependente para TFT;
- 4) NWFS com *setup* dependente para *makespan*;

Os métodos TRIPS, ILS_RA e MA_RAI foram utilizados como parâmetros de comparação. O método que em sua apresentação original não possuir as características do ambiente em estudo será adequadamente adaptado para o novo cenário.

Todos os métodos foram implementados utilizando a linguagem de programação C, os problemas foram processados sob uma plataforma Linux utilizando a distribuição Ubuntu 64 Bits sem interface gráfica, para que todo recurso computacional fosse focado apenas na resolução dos problemas. A configuração da máquina utilizada foi um Intel Core 2 Duo 2.53 GHz com 4 GB de memória RAM.

6.1 Delineamento do Experimento

A experimentação computacional será dividida em quatro partes:

- Parte 1: os métodos ECS, TRIPS, ILS_RA e MA_RAI serão avaliados para o problema NWFS com *setup* independente com critério de minimização do TFT.
- Parte 2: os métodos ECS, TRIPS, ILS_RA e MA_RAI serão avaliados para o problema NWFS com *setup* independente com critério de minimização do *makespan*.
- Parte 3: os métodos ECS, TRIPS, ILS_RA e MA_RAI serão avaliados para o problema NWFS com *setup* dependente com critério de minimização do TFT.
- Parte 4: os métodos ECS, TRIPS, ILS_RA e MA_RAI serão avaliados para o problema NWFS com *setup* dependente com critério de minimização do *makespan*.

A Tabela 3 apresenta resumo dos métodos utilizados, os problemas resolvidos e a condição de aplicação do método: original ou adaptado.

MÉTODO	OBJETIVO	SETUP			
		DEPENDENTE		INDEPENDENTE	
		ORIGINAL	ADAPTADO	ORIGINAL	ADAPTADO
ECS	TFT	X	-	X	-
	MAKESPAN	X	-	X	-
TRIPS	TFT	-	X	X	-
	MAKESPAN	-	X	X	-
ILS_RA	TFT	X	-	-	X
	MAKESPAN	-	X	-	X
MA_RAI	TFT	-	X	-	X
	MAKESPAN	X	-	-	X

Tabela 3 – Mapeamento dos métodos e problemas.

Alguns métodos passaram por um processo de adaptação para outros problemas devido à escassez de bibliografia para o tema abordado. Pelo fato de nenhum dos métodos estudados depender de uma ordenação inicial das tarefas para se iniciar o processo de solução, a exemplo da ordem não-decrescente dos tempos de processamento (*Shortest Processing Time* – SPT) e ordem não-crescente dos tempos de programação (*Longest Processing Time* – LPT), todas as adaptações foram feitas apenas no cálculo da função objetivo. Torna-se importante ressaltar que o método MA_RAI considera as datas de liberação das tarefas, no entanto, essas datas foram consideradas iguais a zero para todas as tarefas, ou seja, todas as tarefas se encontram disponíveis para processamento a partir do instante zero.

Os algoritmos para *setups* independentes foram avaliados com o banco de dados de Ruiz e Allahverdi (2007). Esse banco de dados possui 5,400 problemas que se dividem em duas classes principais: *small* e *large*. O conjunto *small* se subdivide em tempo máximo de processamento $t \in \{10, 100\}$, os custos de *setup* são calculados em função de t no formato $t \times s$, onde $s \in \{0.5, 1, 1.5\}$,

$n \in \{15, 20, 25, 30\}$ e $m \in \{2, 3, 4, 5, 6\}$. O grupo *large* se subdivide em $t \in \{10, 100\}$, $s \in \{0.5, 1, 1.5\}$, $n \in \{15, 20, 25, 30\}$ e $m \in \{10, 20, 30, 40\}$.

Os métodos para *setup* dependentes foram avaliados com o banco de dados de Ruiz e Stützle (2008). Essa base contém 480 problemas, para cada conjunto de problemas, há 4 diferentes distribuições dos tempos de *setup*, com 10 problemas cada: SSD-10, SSD-50, SSD-100 e SSD-125. Os tempos de processamento são uniformemente distribuídos no intervalo de $[1, 99]$ e os tempos de *setup* são distribuídos uniformemente nos intervalos $[1, 9]$, $[1, 49]$, $[1, 99]$ e $[1, 124]$, para as classes de setup SSD-10, SSD-50, SSD-100 e SSD-125, respectivamente.

6.2 Processo de Análise

Os resultados obtidos na experimentação computacional foram analisados com relação à Porcentagem de Sucesso e ao Desvio Relativo Médio.

O Tempo Médio de Computação não foi avaliado por se tratar do critério de parada de três dos quatro métodos, o algoritmo TRIPS é a exceção à regra isso se deve pelo fato de ser o único procedimento construtivo, dessa forma, não faria sentido que fosse interrompido durante sua execução. Vale ressaltar que para problemas grandes o tempo utilizado pelo algoritmo TRIPS é por diversas vezes superior ao gasto pelos outros métodos.

A Porcentagem de Sucesso é definida pelo quociente entre o número de problemas para os quais um determinado método obteve a melhor solução (menor *makespan/total flowtime*) e o número total de programas resolvidos. Logo, quando os métodos obtêm a melhor solução para um mesmo problema, suas Porcentagens de Sucesso são simultaneamente melhoradas.

O Desvio Relativo (DR_h) quantifica o desvio que o método h obtém em relação ao melhor *makespan/flowtime* obtido para o mesmo problema, sendo calculado conforme segue:

$$DR_h(\%) = \frac{D_h - D^*}{D^*} \times 100 \quad (10)$$

onde:

D_h é o *makespan/total flowtime* obtido pelo método h ;

D^* é o melhor *makespan/total flowtime* obtido pelo(s) método(s)

para um determinado problema.

CAPÍTULO 7

ANÁLISE DE RESULTADOS

Os resultados obtidos experimentação computacional serão analisados isoladamente para cada problema e seus critérios de otimização. A apresentação dos dados é feita através de tabelas separadas por grupos de problemas.

O tópico 7.1 se refere aos problemas NWFS com *setup* independente da sequência, seus subtópicos dividem-se entre *makespan* e TFT. De maneira análoga, o tópico 7.2 tratará de apresentar os problemas NWFS com *setup dependente* e suas subdivisões também serão feitas entre *makespan* e TFT.

7.1 No-wait Flowshop com setup independente

A análise dos problemas é subdividida em duas partes de acordo com o critério de otimização em estudo. A base de dados pode ser dividida entre dois tipos de problemas os de menor complexidade chamados *small* representados pela sigla SM e os de maior complexidade chamados *large*, representados pela sigla LG. Conforme mencionado anteriormente, para cada problema será apresentado estatísticas referentes ao percentual de sucesso e ao desvio relativo médio.

7.1.1 Critério de Minimização do *Makespan*

As tabelas 4, 5 e 6 apresentam os resultados obtidos para os problemas de menor complexidade e tempo de processamento máximo igual a 10. A única diferença entre os grupos é a variação presente nos custos de *setup*.

SI_SM_10_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	96,00	0,03	8,00	2,68	100,00	0,00	100,00	0,00
15 x 3	84,00	0,11	0,00	5,57	76,00	0,17	88,00	0,08
15 x 4	64,00	0,39	0,00	6,25	76,00	0,16	68,00	0,37
15 x 5	60,00	0,31	0,00	6,92	80,00	0,15	84,00	0,12
15 x 6	44,00	0,54	0,00	6,49	80,00	0,16	40,00	0,49
20 x 2	84,00	0,09	8,00	4,15	88,00	0,07	100,00	0,00
20 x 3	44,00	0,46	0,00	5,50	20,00	0,66	92,00	0,04
20 x 4	44,00	0,54	0,00	6,57	32,00	0,40	52,00	0,30
20 x 5	48,00	0,28	0,00	8,34	68,00	0,15	84,00	0,07
20 x 6	72,00	0,19	0,00	8,37	80,00	0,11	88,00	0,07
25 x 2	92,00	0,06	4,00	3,08	76,00	0,12	100,00	0,00
25 x 3	64,00	0,24	0,00	5,73	16,00	0,68	84,00	0,11
25 x 4	36,00	0,35	0,00	7,69	24,00	0,48	88,00	0,05
25 x 5	36,00	0,43	0,00	7,65	16,00	0,51	96,00	0,02
25 x 6	52,00	0,26	0,00	8,45	40,00	0,41	72,00	0,10
30 x 2	96,00	0,02	0,00	3,00	72,00	0,13	96,00	0,02
30 x 3	48,00	0,31	0,00	5,42	8,00	0,71	76,00	0,26
30 x 4	12,00	0,59	0,00	7,57	8,00	0,93	96,00	0,01
30 x 5	36,00	0,36	0,00	8,00	4,00	0,87	92,00	0,04
30 x 6	56,00	0,22	0,00	8,55	4,00	0,72	64,00	0,18
Média	58,40	0,29	1,00	6,30	48,40	0,38	83,00	0,12

Tabela 4 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_0,5.

SI_SM_10_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	100,00	0,00	8,00	3,01	100,00	0,00	100,00	0,00
15 x 3	80,00	0,13	0,00	4,77	64,00	0,22	88,00	0,07
15 x 4	80,00	0,09	0,00	5,89	68,00	0,21	84,00	0,08
15 x 5	68,00	0,15	0,00	5,69	48,00	0,31	72,00	0,27
15 x 6	64,00	0,21	0,00	5,42	48,00	0,35	56,00	0,36
20 x 2	96,00	0,02	8,00	2,26	96,00	0,02	100,00	0,00
20 x 3	64,00	0,16	0,00	4,93	64,00	0,32	88,00	0,08
20 x 4	56,00	0,27	0,00	5,96	24,00	0,60	64,00	0,20
20 x 5	44,00	0,20	0,00	7,50	64,00	0,13	96,00	0,01
20 x 6	40,00	0,28	0,00	6,95	64,00	0,14	88,00	0,04
25 x 2	92,00	0,04	0,00	2,84	80,00	0,07	100,00	0,00
25 x 3	48,00	0,39	0,00	4,47	16,00	0,48	76,00	0,09
25 x 4	32,00	0,31	0,00	6,23	44,00	0,25	100,00	0,00
25 x 5	40,00	0,32	0,00	7,11	24,00	0,46	84,00	0,05
25 x 6	48,00	0,22	0,00	6,76	20,00	0,34	88,00	0,03
30 x 2	96,00	0,01	4,00	2,59	84,00	0,06	100,00	0,00
30 x 3	64,00	0,18	0,00	5,00	20,00	0,47	72,00	0,13
30 x 4	24,00	0,30	0,00	5,70	0,00	0,52	96,00	0,01
30 x 5	32,00	0,29	0,00	6,19	4,00	0,58	96,00	0,02
30 x 6	68,00	0,09	0,00	6,89	4,00	0,72	52,00	0,19
Média	61,80	0,18	1,00	5,31	46,80	0,31	85,00	0,08

Tabela 5 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_1,0.

SI_SM_10_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	100,00	0,00	16,00	2,15	100,00	0,00	100,00	0,00
15 x 3	92,00	0,04	0,00	3,89	84,00	0,09	96,00	0,02
15 x 4	100,00	0,00	0,00	4,04	84,00	0,09	100,00	0,00
15 x 5	84,00	0,09	0,00	4,82	36,00	0,36	68,00	0,17
15 x 6	64,00	0,17	0,00	5,12	52,00	0,27	60,00	0,28
20 x 2	100,00	0,00	0,00	1,98	96,00	0,02	100,00	0,00
20 x 3	72,00	0,10	0,00	3,56	68,00	0,14	84,00	0,09
20 x 4	68,00	0,15	0,00	4,93	32,00	0,38	56,00	0,27
20 x 5	40,00	0,30	0,00	5,75	68,00	0,10	96,00	0,01
20 x 6	40,00	0,18	0,00	5,62	68,00	0,09	96,00	0,01
25 x 2	100,00	0,00	0,00	2,31	100,00	0,00	92,00	0,02
25 x 3	56,00	0,19	0,00	4,26	24,00	0,32	92,00	0,02
25 x 4	44,00	0,16	0,00	5,15	40,00	0,20	100,00	0,00
25 x 5	48,00	0,20	0,00	5,45	32,00	0,29	84,00	0,05
25 x 6	24,00	0,30	0,00	5,33	4,00	0,44	88,00	0,04
30 x 2	100,00	0,00	0,00	1,99	92,00	0,03	100,00	0,00
30 x 3	56,00	0,15	0,00	3,91	32,00	0,28	84,00	0,04
30 x 4	32,00	0,24	0,00	5,54	16,00	0,35	92,00	0,02
30 x 5	32,00	0,23	0,00	5,48	4,00	0,49	84,00	0,04
30 x 6	24,00	0,24	0,00	5,39	12,00	0,42	92,00	0,01
Média	63,80	0,14	0,80	4,33	52,20	0,22	88,20	0,05

Tabela 6 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 10_1,5.**

Em uma primeira análise podemos verificar que o comportamento dos métodos é semelhante entre as classes de problemas, sendo observadas apenas pequenas variações entre grupos. Muito embora não tenha sido originalmente construído com a finalidade de resolver problemas para *setup* independente, o desempenho do método MA_RAI superou os outros métodos chegando a obter a melhor solução para 88,20% dos casos, como observado na Tabela 6. Na média entre os três grupos de problemas o método foi capaz de gerar a melhor solução

em 85% dos casos. No quesito desvio relativo médio, o método também apresentou os melhores resultados, na média apresentou um desvio pouco superior a 0,08.

Observa-se que o método ECS_NSL também foi capaz de gerar boas soluções chegando à marca de 63,80% de percentual de sucesso para o terceiro grupo de problemas, como verificado na Tabela 6. Na média entre os três grupos o método apresentou um percentual de sucesso de 61,33%. O desvio padrão médio observado entre os três grupos foi de aproximadamente 0,23.

O método ILS_RA, também adaptado para o problema, apresentou soluções consideravelmente inferiores ao ECS_NSL, tanto na medida de percentual de sucesso que na média ficou em 49,13% quanto no desvio padrão médio calculado em 0,30. O método TRIPS, um dos pioneiros no estudo de problemas com custos de *setup* apartados do processamento, apresentou os piores resultados ficando muito aquém dos outros métodos avaliados o percentual de sucesso médio foi inferior a 1% e seu desvio relativo médio calculado em 5,31.

As tabelas 7, 8 e 9 exibem os resultados atingidos para os problemas de baixa complexidade baixa, porém com os tempos de processamento variando até 100.

SI_SM_100_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	96,00	0,01	8,00	2,87	84,00	0,02	100,00	0,00
15 x 3	64,00	0,20	0,00	6,29	44,00	0,35	68,00	0,13
15 x 4	72,00	0,12	0,00	6,23	36,00	0,25	32,00	0,40
15 x 5	72,00	0,10	0,00	7,05	44,00	0,29	36,00	0,45
15 x 6	60,00	0,23	0,00	7,23	48,00	0,32	28,00	0,53
20 x 2	56,00	0,10	0,00	4,06	36,00	0,15	96,00	0,02
20 x 3	56,00	0,32	0,00	5,77	4,00	0,47	56,00	0,11
20 x 4	40,00	0,38	0,00	6,83	16,00	0,69	52,00	0,26
20 x 5	28,00	0,44	0,00	7,17	20,00	0,30	88,00	0,02
20 x 6	32,00	0,33	0,00	7,19	44,00	0,18	76,00	0,04
25 x 2	52,00	0,12	0,00	4,09	32,00	0,25	100,00	0,00
25 x 3	28,00	0,43	0,00	5,38	8,00	0,69	68,00	0,11
25 x 4	4,00	0,52	0,00	8,66	0,00	0,66	96,00	0,00
25 x 5	32,00	0,32	0,00	7,78	4,00	0,64	84,00	0,01
25 x 6	20,00	0,34	0,00	9,54	12,00	0,44	80,00	0,05
30 x 2	48,00	0,13	0,00	3,03	20,00	0,31	88,00	0,02
30 x 3	20,00	0,55	0,00	5,63	4,00	1,00	80,00	0,08
30 x 4	12,00	0,51	0,00	8,18	0,00	0,99	92,00	0,01
30 x 5	20,00	0,42	0,00	8,97	0,00	0,99	84,00	0,03
30 x 6	24,00	0,37	0,00	8,78	16,00	0,70	64,00	0,11
Média	41,80	0,30	0,40	6,54	23,60	0,48	73,40	0,12

Tabela 7 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100_0,5.

SI_SM_100_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	84,00	0,01	0,00	3,37	84,00	0,03	100,00	0,00
15 x 3	72,00	0,04	0,00	5,57	52,00	0,14	68,00	0,05
15 x 4	48,00	0,14	0,00	5,73	36,00	0,21	56,00	0,23
15 x 5	56,00	0,17	0,00	6,60	48,00	0,31	36,00	0,38
15 x 6	64,00	0,16	0,00	5,65	40,00	0,25	28,00	0,47
20 x 2	88,00	0,02	0,00	2,86	72,00	0,03	100,00	0,00
20 x 3	48,00	0,18	0,00	4,82	24,00	0,32	52,00	0,09
20 x 4	56,00	0,15	0,00	6,71	12,00	0,56	36,00	0,21
20 x 5	40,00	0,25	0,00	6,34	48,00	0,10	88,00	0,01
20 x 6	20,00	0,27	0,00	6,61	28,00	0,19	80,00	0,04
25 x 2	80,00	0,04	0,00	2,89	56,00	0,17	92,00	0,00
25 x 3	36,00	0,21	0,00	4,86	8,00	0,51	60,00	0,17
25 x 4	8,00	0,37	0,00	6,40	0,00	0,50	100,00	0,00
25 x 5	12,00	0,28	0,00	6,34	4,00	0,39	88,00	0,02
25 x 6	12,00	0,27	0,00	7,24	8,00	0,41	84,00	0,04
30 x 2	60,00	0,07	0,00	2,71	24,00	0,17	76,00	0,02
30 x 3	36,00	0,25	0,00	5,11	4,00	0,45	76,00	0,05
30 x 4	8,00	0,41	0,00	5,97	0,00	0,71	100,00	0,00
30 x 5	12,00	0,37	0,00	7,06	0,00	0,73	96,00	0,00
30 x 6	36,00	0,22	0,00	6,89	0,00	0,66	72,00	0,08
Média	43,80	0,19	0,00	5,49	27,40	0,34	74,40	0,09

Tabela 8 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100_1,0.

SI_SM_100_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	92,00	0,01	4,00	2,13	92,00	0,02	96,00	0,00
15 x 3	64,00	0,05	0,00	4,50	44,00	0,12	100,00	0,00
15 x 4	68,00	0,05	0,00	5,62	16,00	0,28	56,00	0,18
15 x 5	48,00	0,10	0,00	4,51	32,00	0,27	52,00	0,12
15 x 6	80,00	0,07	0,00	5,48	36,00	0,22	24,00	0,35
20 x 2	96,00	0,00	0,00	2,13	76,00	0,02	96,00	0,00
20 x 3	88,00	0,03	0,00	3,30	28,00	0,17	60,00	0,07
20 x 4	76,00	0,05	0,00	4,69	4,00	0,46	20,00	0,32
20 x 5	20,00	0,29	0,00	6,10	24,00	0,15	96,00	0,00
20 x 6	28,00	0,22	0,00	5,57	28,00	0,19	100,00	0,00
25 x 2	84,00	0,02	0,00	2,39	52,00	0,08	100,00	0,00
25 x 3	56,00	0,09	0,00	4,20	4,00	0,37	56,00	0,11
25 x 4	12,00	0,23	0,00	5,30	16,00	0,25	100,00	0,00
25 x 5	12,00	0,29	0,00	6,25	4,00	0,36	92,00	0,00
25 x 6	20,00	0,25	0,00	5,06	0,00	0,40	80,00	0,02
30 x 2	80,00	0,02	0,00	1,98	68,00	0,04	100,00	0,00
30 x 3	20,00	0,27	0,00	3,66	8,00	0,31	84,00	0,02
30 x 4	4,00	0,39	0,00	5,00	0,00	0,56	96,00	0,00
30 x 5	16,00	0,28	0,00	5,50	8,00	0,52	84,00	0,01
30 x 6	32,00	0,23	0,00	5,67	0,00	0,59	72,00	0,04
Média	49,80	0,15	0,20	4,45	27,00	0,27	78,20	0,06

Tabela 9 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 100_1,5.**

A característica da estabilidade de resultados entre grupos similares presente na análise anterior permanece inalterada para estes grupos de problemas. No entanto verifica-se que os percentuais de sucesso de todos os métodos sofreram reduções se comparados aos problemas cujos tempos de processamento variavam até 10. Nota-se que o melhor método tanto para percentual de sucesso quanto para desvio relativo médio continua sendo o MA_RAI, seu percentual médio de sucesso considerando os três grupos foi de 75,3%, a média dos desvios relativos foi observada em 0,06. O método ECS_NSL

apresentou resultados significativamente inferior ao MA_RAI, a maior média de percentual observada foi ligeiramente inferior a 50%, Tabela 9, na média entre os grupos o método foi capaz de gerar a melhor solução em apenas 45,13% dos casos. Na média entre grupos, o desvio relativo observado ficou em 0,21, o que indica que muito embora não tenha gerado a melhor solução para mais da metade dos casos, o método foi capaz de gerar uma solução de qualidade razoável. Já os métodos ILS_RA e TRIPS, geraram soluções de qualidade inferior obtendo 26% e 0,2% de percentual de sucesso respectivamente, no quesito desvio relativo médio o método ILS_RA apresentou média de 0,36 enquanto o algoritmo TRIPS obteve 5,49.

As tabelas 10, 11 e 12 apresentam os resultados obtidos para os problemas de maior complexidade e tempo de processamento máximo igual a 10.

SI_LG_10_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
50 X 10	68,00	0,14	0,00	7,25	4,00	0,80	52,00	0,14
50 X 20	76,00	0,08	0,00	8,08	4,00	0,67	28,00	0,45
50 X 30	88,00	0,03	0,00	8,99	4,00	0,74	12,00	0,53
50 X 40	88,00	0,02	0,00	9,30	4,00	0,70	12,00	0,53
100 X 10	56,00	0,20	0,00	5,92	0,00	0,87	52,00	0,19
100 X 20	96,00	0,01	0,00	7,00	0,00	0,82	12,00	0,54
100 X 30	100,00	0,00	0,00	6,83	0,00	0,85	4,00	0,95
100 X 40	96,00	0,01	0,00	7,35	0,00	0,77	8,00	0,87
150 X 10	84,00	0,03	0,00	4,91	4,00	0,83	12,00	0,43
150 X 20	92,00	0,01	0,00	5,95	8,00	0,69	0,00	0,94
150 X 30	100,00	0,00	0,00	6,35	0,00	0,73	0,00	1,01
150 X 40	96,00	0,01	0,00	6,27	0,00	0,82	4,00	1,23
200 X 10	92,00	0,01	0,00	4,09	8,00	0,62	0,00	0,52
200 X 20	96,00	0,01	0,00	4,78	0,00	0,63	4,00	1,09
200 X 30	100,00	0,00	0,00	5,42	0,00	0,71	0,00	1,17
200 X 40	100,00	0,00	0,00	5,65	0,00	0,72	0,00	1,24
Média	89,25	0,03	0,00	6,51	2,25	0,75	12,50	0,74

Tabela 10 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_0,5.

SI_LG_10_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
50 X 10	72,00	0,10	0,00	6,55	4,00	0,83	28,00	0,39
50 X 20	80,00	0,05	0,00	7,22	0,00	0,61	28,00	0,39
50 X 30	72,00	0,08	0,00	6,96	0,00	0,72	40,00	0,30
50 X 40	84,00	0,07	0,00	7,14	4,00	0,54	20,00	0,48
100 X 10	68,00	0,07	0,00	4,69	0,00	0,69	40,00	0,25
100 X 20	96,00	0,01	0,00	5,65	0,00	0,71	4,00	0,62
100 X 30	92,00	0,01	0,00	5,81	0,00	0,77	8,00	0,69
100 X 40	84,00	0,03	0,00	5,74	0,00	0,56	16,00	0,56
150 X 10	100,00	0,00	0,00	4,06	0,00	0,71	4,00	0,41
150 X 20	100,00	0,00	0,00	4,96	4,00	0,60	0,00	0,84
150 X 30	100,00	0,00	0,00	5,21	0,00	0,71	0,00	1,03
150 X 40	100,00	0,00	0,00	5,14	0,00	0,69	0,00	1,12
200 X 10	92,00	0,01	0,00	3,30	0,00	0,53	8,00	0,41
200 X 20	100,00	0,00	0,00	3,82	0,00	0,65	0,00	0,86
200 X 30	100,00	0,00	0,00	4,48	0,00	0,58	0,00	1,07
200 X 40	100,00	0,00	0,00	4,61	0,00	0,47	0,00	0,97
Média	90,00	0,03	0,00	5,33	0,75	0,65	12,25	0,65

Tabela 11 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_1,0.

SI_LG_10_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
50 X 10	72,00	0,07	0,00	5,68	0,00	0,65	40,00	0,21
50 X 20	60,00	0,15	0,00	5,93	0,00	0,64	52,00	0,22
50 X 30	68,00	0,06	0,00	6,37	4,00	0,55	32,00	0,22
50 X 40	68,00	0,05	0,00	6,95	4,00	0,54	28,00	0,36
100 X 10	56,00	0,15	0,00	4,33	0,00	0,68	48,00	0,14
100 X 20	84,00	0,02	0,00	4,55	0,00	0,60	16,00	0,38
100 X 30	92,00	0,02	0,00	4,70	0,00	0,59	8,00	0,50
100 X 40	92,00	0,01	0,00	4,64	0,00	0,60	12,00	0,47
150 X 10	84,00	0,05	0,00	3,36	0,00	0,60	20,00	0,29
150 X 20	100,00	0,00	0,00	3,82	0,00	0,53	0,00	0,56
150 X 30	96,00	0,02	0,00	4,09	4,00	0,53	4,00	0,70
150 X 40	96,00	0,00	0,00	4,19	4,00	0,50	0,00	0,74
200 X 10	84,00	0,02	0,00	3,13	0,00	0,44	16,00	0,37
200 X 20	96,00	0,00	0,00	3,30	0,00	0,48	8,00	0,65
200 X 30	100,00	0,00	0,00	3,84	0,00	0,47	4,00	0,71
200 X 40	92,00	0,00	0,00	4,03	8,00	0,59	0,00	0,92
Média	83,75	0,04	0,00	4,56	1,50	0,56	18,00	0,46

Tabela 12 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_1,5.

Uma primeira característica que volta a chamar atenção é o fato que, assim como para os problemas de complexidade inferior, os resultados não apresentam muita diferenciação entre os grupos.

O método ECS_NSL apresentou a melhor solução para os três grupos, chegando a atingir a marca de 90% de percentual de sucesso como observado na Tabela 11. A média dos percentuais de sucesso entre grupos também se manteve alta representando acima de 85% dos casos, o desvio padrão médio foi baixo para todas as classes. O método MA_RAI, que foi considerado o melhor nos casos de menor complexidade, para esses casos não foi capaz de atingir altos percentuais de sucesso obtendo uma média entre grupos de aproximadamente 14%, no entanto, seu desvio padrão médio se manteve baixo o que indica que embora não tenha sido capaz de gerar as melhores soluções, o método gerou soluções de qualidade razoável.

Por outro lado os métodos TRIPS e ILS_RA não foram capazes de gerar boas soluções, o ILS_RA obteve uma média de percentual de sucesso entre grupos de 1,5%, já o algoritmo TRIPS não gerou nenhuma das melhores soluções. Embora o ILS_RA não tenha obtido resultados muito positivos em termos percentuais de sucesso, o seu desvio relativo médio pode ser considerado baixo o que indica que o método gerou soluções razoáveis como um todo. O algoritmo TRIPS foi o que apresentou o maior desvio padrão entre todos os métodos.

As tabelas 13, 14 e 15 apresentam os resultados obtidos para os problemas de maior complexidade e tempo de processamento máximo igual a 100.

SI_LG_100_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
50 X 10	60,00	0,12	0,00	7,93	4,00	0,96	40,00	0,32
50 X 20	64,00	0,08	0,00	8,19	8,00	0,73	32,00	0,37
50 X 30	84,00	0,03	0,00	9,09	4,00	0,73	12,00	0,45
50 X 40	72,00	0,06	0,00	8,65	4,00	0,72	28,00	0,49
100 X 10	72,00	0,07	0,00	6,25	0,00	1,11	28,00	0,43
100 X 20	92,00	0,02	0,00	7,09	0,00	0,83	8,00	0,68
100 X 30	96,00	0,00	0,00	7,57	0,00	0,74	4,00	0,71
100 X 40	88,00	0,02	0,00	7,49	0,00	0,80	12,00	0,86
150 X 10	88,00	0,04	0,00	5,53	0,00	0,91	12,00	0,54
150 X 20	100,00	0,00	0,00	6,08	0,00	1,03	0,00	1,08
150 X 30	100,00	0,00	0,00	6,30	0,00	0,96	0,00	1,20
150 X 40	100,00	0,00	0,00	6,43	0,00	0,93	0,00	1,24
200 X 10	72,00	0,11	0,00	4,73	4,00	0,70	24,00	0,41
200 X 20	96,00	0,02	0,00	5,10	4,00	0,59	0,00	0,88
200 X 30	96,00	0,00	0,00	5,53	4,00	0,66	0,00	1,19
200 X 40	100,00	0,00	0,00	3,79	0,00	3,18	0,00	3,41
Média	86,25	0,04	0,00	6,61	2,00	0,97	12,50	0,89

Tabela 13 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100_0,5.

SI_LG_100_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
50 X 10	88,00	0,02	0,00	6,60	4,00	0,59	96,00	0,01
50 X 20	68,00	0,08	0,00	7,07	4,00	0,45	96,00	0,01
50 X 30	60,00	0,17	0,00	6,95	32,00	0,44	68,00	0,07
50 X 40	68,00	0,13	0,00	7,42	32,00	0,32	68,00	0,07
100 X 10	84,00	0,02	0,00	4,81	4,00	0,66	96,00	0,00
100 X 20	88,00	0,03	0,00	5,44	36,00	0,29	64,00	0,08
100 X 30	100,00	0,00	0,00	5,54	32,00	0,22	68,00	0,07
100 X 40	88,00	0,02	0,00	7,03	4,00	1,48	4,00	1,59
150 X 10	100,00	0,00	0,00	3,98	4,00	0,53	96,00	0,00
150 X 20	100,00	0,00	0,00	3,97	60,00	0,09	40,00	0,15
150 X 30	92,00	0,02	0,00	4,50	76,00	0,07	24,00	0,25
150 X 40	84,00	0,01	0,00	16,44	0,00	11,20	0,00	11,47
200 X 10	84,00	0,01	0,00	3,09	20,00	0,25	80,00	0,03
200 X 20	84,00	0,03	0,00	3,70	72,00	0,05	28,00	0,22
200 X 30	88,00	0,03	0,00	3,98	72,00	0,05	28,00	0,33
200 X 40	88,00	0,02	0,00	4,69	0,00	0,69	0,00	0,98
Média	85,25	0,04	0,00	5,95	28,25	1,09	53,50	0,96

Tabela 14 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100_1,0.

SI_LG_100_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
50 X 10	56,00	0,09	0,00	5,54	0,00	0,80	48,00	0,13
50 X 20	72,00	0,08	0,00	6,16	4,00	0,73	24,00	0,29
50 X 30	68,00	0,12	0,00	6,10	4,00	0,59	28,00	0,26
50 X 40	92,00	0,04	0,00	6,88	0,00	0,63	8,00	0,46
100 X 10	48,00	0,10	0,00	4,13	0,00	0,70	52,00	0,13
100 X 20	88,00	0,02	0,00	4,55	0,00	0,63	12,00	0,35
100 X 30	84,00	0,02	0,00	5,26	0,00	0,76	16,00	0,56
100 X 40	96,00	0,01	0,00	5,17	0,00	0,69	4,00	0,59
150 X 10	80,00	0,01	0,00	3,77	0,00	0,58	24,00	0,22
150 X 20	100,00	0,00	0,00	4,01	0,00	0,63	0,00	0,65
150 X 30	100,00	0,00	0,00	4,71	0,00	0,80	0,00	0,80
150 X 40	96,00	0,01	0,00	4,61	0,00	0,54	4,00	0,74
200 X 10	92,00	0,01	0,00	3,30	4,00	0,56	4,00	0,29
200 X 20	92,00	0,00	0,00	3,41	4,00	0,48	4,00	0,55
200 X 30	96,00	0,00	0,00	3,89	0,00	0,55	4,00	0,69
200 X 40	100,00	0,00	0,00	4,10	0,00	0,55	0,00	0,80
Média	85,00	0,03	0,00	4,72	1,00	0,64	14,50	0,47

Tabela 15 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 100_1,5.**

O método ECS_NSL apresentou as melhores soluções, seu percentual de sucesso foi de 85,5%, o desvio relativo médio manteve-se baixo para todas os grupos. MA_RAI foi o segundo método com melhor aproveitamento no percentual de sucesso com média de 26,8% seu desvio relativo médio foi de 0,77. ILS_RA apresentou percentual de sucesso médio de 10,42% e seu desvio relativo ficou em 0,9. O método TRIPS não obteve sucesso e seu desvio relativo médio foi o maior de todos.

O Gráfico 1 apresenta as médias dos DRMs para os problemas apresentados acima, nota-se que o método *TRIPS* apresentou o maior DRM dentre os métodos testados. Para a classe de problemas pequenos, classe SM, verifica-se

que os métodos ECS_NSL, ILS_RA e MA_RAI possuíram desempenhos semelhantes, no entanto, para os problemas grandes, classe LG, verifica-se que o método proposto ECS_NSL apresenta as melhores soluções.

Gráfico 1 – DRM para problemas SI com critério Makespan.

O Gráfico 2 apresenta as médias dos PS para os problemas apresentados acima. Para a classe de problemas pequenos, classe SM, nota-se que o método MA_RAI é responsável por produzir as melhores soluções seguido do ECS_NSL e ILS_RA. No entanto para problemas mais complexos nota-se que o método proposto gera as melhores soluções dentre os métodos avaliados.

Gráfico 2 – DRM para problemas SI com critério Makespan.

7.1.2 Critério de Minimização do *Total Flowtime*

As tabelas 16, 17 e 18 apresentam os resultados obtidos para os problemas de menor complexidade e tempo de processamento máximo igual a 10. A única diferença entre os grupos é a variação presente nos custos de *setup*.

SI_SM_10_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	72,00	0,09	16,00	1,03	60,00	0,16	4,00	1,60
15 x 3	64,00	0,28	4,00	1,58	56,00	0,21	4,00	1,43
15 x 4	48,00	0,38	16,00	1,74	56,00	0,21	20,00	1,00
15 x 5	72,00	0,18	4,00	1,59	40,00	0,47	0,00	1,26
15 x 6	72,00	0,20	8,00	1,28	68,00	0,14	8,00	1,33
20 x 2	48,00	0,40	8,00	1,27	48,00	0,20	0,00	2,76
20 x 3	28,00	0,53	12,00	1,85	64,00	0,22	0,00	2,35
20 x 4	56,00	0,25	8,00	1,45	48,00	0,23	0,00	2,25
20 x 5	24,00	0,68	0,00	2,47	64,00	0,10	48,00	0,11
20 x 6	44,00	0,33	0,00	1,75	64,00	0,08	52,00	0,17
25 x 2	48,00	0,17	16,00	1,14	40,00	0,32	0,00	3,77
25 x 3	68,00	0,25	0,00	2,03	32,00	0,38	0,00	3,51
25 x 4	28,00	0,67	4,00	1,99	24,00	0,38	52,00	0,23
25 x 5	36,00	0,50	0,00	2,23	36,00	0,21	32,00	0,35
25 x 6	36,00	0,51	0,00	2,77	24,00	0,33	44,00	0,21
30 x 2	76,00	0,09	12,00	1,12	16,00	0,44	0,00	3,86
30 x 3	56,00	0,29	4,00	1,34	40,00	0,43	0,00	3,65
30 x 4	28,00	0,47	0,00	2,44	12,00	0,51	60,00	0,20
30 x 5	32,00	0,51	0,00	2,79	20,00	0,53	48,00	0,35
30 x 6	40,00	0,32	0,00	2,34	24,00	0,42	36,00	0,44
Média	48,80	0,36	5,60	1,81	41,80	0,30	20,40	1,54

Tabela 16 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_0,5.

SI_SM_10_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	72,00	0,10	16,00	1,27	48,00	0,27	4,00	1,97
15 x 3	76,00	0,07	4,00	1,98	24,00	0,36	4,00	1,68
15 x 4	68,00	0,19	0,00	1,69	48,00	0,26	8,00	1,20
15 x 5	80,00	0,07	8,00	1,62	48,00	0,23	4,00	1,12
15 x 6	76,00	0,15	0,00	1,86	44,00	0,28	8,00	1,04
20 x 2	76,00	0,12	0,00	1,55	48,00	0,16	0,00	2,57
20 x 3	76,00	0,14	0,00	2,06	40,00	0,20	0,00	2,03
20 x 4	48,00	0,35	0,00	2,06	56,00	0,19	0,00	2,38
20 x 5	24,00	0,57	0,00	2,45	60,00	0,04	32,00	0,13
20 x 6	32,00	0,41	0,00	2,22	72,00	0,03	40,00	0,10
25 x 2	84,00	0,07	0,00	1,35	20,00	0,37	0,00	2,84
25 x 3	72,00	0,20	4,00	1,70	24,00	0,44	0,00	3,10
25 x 4	32,00	0,53	0,00	2,78	44,00	0,17	32,00	0,33
25 x 5	28,00	0,27	0,00	2,37	24,00	0,20	52,00	0,20
25 x 6	28,00	0,34	0,00	2,57	52,00	0,16	20,00	0,36
30 x 2	84,00	0,03	8,00	1,30	8,00	0,53	0,00	3,90
30 x 3	64,00	0,18	8,00	1,64	28,00	0,48	0,00	3,37
30 x 4	40,00	0,36	0,00	2,47	28,00	0,36	32,00	0,28
30 x 5	52,00	0,21	0,00	1,99	28,00	0,43	20,00	0,43
30 x 6	60,00	0,22	0,00	2,55	28,00	0,33	12,00	0,55
Média	58,60	0,23	2,40	1,97	38,60	0,27	13,40	1,48

Tabela 17 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_1,0.

SI_SM_10_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	84,00	0,02	24,00	1,00	80,00	0,03	0,00	1,76
15 x 3	56,00	0,16	4,00	1,94	84,00	0,03	0,00	1,59
15 x 4	72,00	0,14	4,00	1,55	68,00	0,10	0,00	1,34
15 x 5	60,00	0,16	0,00	1,37	44,00	0,17	8,00	1,21
15 x 6	40,00	0,47	12,00	1,45	84,00	0,10	0,00	1,52
20 x 2	60,00	0,17	0,00	1,70	56,00	0,11	0,00	3,09
20 x 3	72,00	0,10	0,00	1,66	36,00	0,14	0,00	2,72
20 x 4	48,00	0,28	0,00	1,85	56,00	0,27	0,00	2,76
20 x 5	24,00	0,49	0,00	2,11	72,00	0,04	32,00	0,10
20 x 6	28,00	0,36	0,00	2,15	72,00	0,03	48,00	0,13
25 x 2	76,00	0,07	0,00	1,62	32,00	0,32	0,00	3,43
25 x 3	60,00	0,11	0,00	1,41	44,00	0,15	0,00	2,99
25 x 4	28,00	0,42	0,00	2,54	52,00	0,15	24,00	0,30
25 x 5	36,00	0,37	0,00	1,96	32,00	0,19	36,00	0,26
25 x 6	44,00	0,30	0,00	2,31	44,00	0,13	12,00	0,31
30 x 2	80,00	0,04	0,00	1,38	20,00	0,25	0,00	3,66
30 x 3	56,00	0,14	0,00	1,42	44,00	0,25	0,00	3,43
30 x 4	28,00	0,38	0,00	1,99	44,00	0,22	32,00	0,36
30 x 5	68,00	0,13	0,00	2,22	20,00	0,31	20,00	0,46
30 x 6	52,00	0,21	0,00	2,11	12,00	0,23	36,00	0,47
Média	53,60	0,22	2,20	1,79	49,80	0,16	12,40	1,60

Tabela 18 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 10_1,5.**

Em uma primeira análise nota-se uma mudança drástica de resultados com relação aos dados obtidos para o problema de *makespan*. Enquanto naquele cenário o método MA_RAI apresentou as melhores soluções, para TFT o melhor método foi o ECS_NSL com percentual de sucesso média entre grupos superior a 53%, o desvio médio relativo manteve-se baixo, na casa 0,27. O método ILS_RA apresentou soluções de boa qualidade, em média o percentual de sucesso ficou em 43,4% o desvio médio relativo foi de 0,24. Já o algoritmo MA_RAI obteve um percentual médio de sucesso de 15,4% e desvio médio relativo foi ligeiramente

superior a 1,54. O método TRIPS apresentou o pior desempenho com 3,4% de percentual de sucesso médio entre grupos e um desvio relativo médio de 1,86.

As tabelas 19, 20 e 21 apresentam os resultados obtidos para os problemas de menor complexidade e tempo de processamento máximo igual a 100.

SI_SM_100_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	84,00	0,06	8,00	1,07	48,00	0,13	0,00	1,50
15 x 3	84,00	0,19	12,00	1,36	40,00	0,31	0,00	1,64
15 x 4	88,00	0,05	0,00	1,49	20,00	0,33	4,00	1,27
15 x 5	80,00	0,22	4,00	1,66	48,00	0,34	4,00	1,22
15 x 6	72,00	0,20	4,00	1,93	56,00	0,28	12,00	1,24
20 x 2	52,00	0,27	12,00	0,95	36,00	0,27	4,00	2,23
20 x 3	64,00	0,24	4,00	1,30	32,00	0,50	0,00	2,10
20 x 4	64,00	0,45	8,00	1,85	32,00	0,37	4,00	2,20
20 x 5	24,00	0,49	0,00	2,07	64,00	0,09	52,00	0,11
20 x 6	28,00	0,34	0,00	2,05	64,00	0,04	44,00	0,09
25 x 2	64,00	0,11	8,00	1,21	28,00	0,57	0,00	2,95
25 x 3	72,00	0,08	8,00	1,28	20,00	0,40	0,00	2,59
25 x 4	16,00	1,00	4,00	2,64	20,00	0,48	64,00	0,13
25 x 5	28,00	0,60	0,00	2,48	28,00	0,31	48,00	0,12
25 x 6	24,00	0,34	0,00	2,74	44,00	0,22	36,00	0,28
30 x 2	52,00	0,17	8,00	1,15	40,00	0,40	0,00	2,83
30 x 3	60,00	0,34	12,00	1,17	28,00	0,61	0,00	2,84
30 x 4	20,00	0,61	0,00	2,39	28,00	0,45	52,00	0,30
30 x 5	44,00	0,22	0,00	2,32	24,00	0,35	32,00	0,48
30 x 6	48,00	0,26	0,00	2,51	32,00	0,38	20,00	0,41
Média	53,40	0,31	4,60	1,78	36,60	0,34	18,80	1,33

Tabela 19 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 100_0,5.**

SI_SM_100_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	72,00	0,14	0,00	1,44	56,00	0,12	4,00	1,38
15 x 3	84,00	0,04	0,00	1,97	36,00	0,33	4,00	1,31
15 x 4	48,00	0,25	8,00	1,51	52,00	0,22	8,00	0,94
15 x 5	68,00	0,14	4,00	1,87	40,00	0,28	12,00	1,14
15 x 6	64,00	0,19	4,00	1,61	48,00	0,26	8,00	1,12
20 x 2	48,00	0,24	8,00	1,25	48,00	0,15	0,00	2,22
20 x 3	64,00	0,18	0,00	1,65	36,00	0,19	0,00	2,16
20 x 4	56,00	0,30	4,00	1,19	48,00	0,19	0,00	2,05
20 x 5	20,00	0,47	0,00	2,08	68,00	0,06	56,00	0,06
20 x 6	28,00	0,39	4,00	2,20	48,00	0,09	48,00	0,06
25 x 2	72,00	0,17	8,00	1,24	20,00	0,41	0,00	3,20
25 x 3	64,00	0,13	0,00	1,55	36,00	0,30	0,00	2,72
25 x 4	32,00	0,44	0,00	2,65	40,00	0,16	28,00	0,28
25 x 5	24,00	0,41	4,00	2,35	44,00	0,17	28,00	0,20
25 x 6	28,00	0,39	0,00	2,05	60,00	0,12	16,00	0,33
30 x 2	40,00	0,33	4,00	1,47	56,00	0,20	0,00	3,11
30 x 3	48,00	0,15	8,00	1,35	44,00	0,15	0,00	2,70
30 x 4	40,00	0,35	0,00	2,38	28,00	0,29	32,00	0,32
30 x 5	52,00	0,31	0,00	2,21	24,00	0,33	24,00	0,43
30 x 6	44,00	0,29	0,00	2,24	44,00	0,22	12,00	0,57
Média	49,80	0,27	2,80	1,81	43,80	0,21	14,00	1,31

Tabela 20 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 100_1,0.**

SI_SM_100_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
15 x 2	72,00	0,08	4,00	1,05	72,00	0,09	0,00	1,69
15 x 3	68,00	0,22	0,00	1,60	36,00	0,19	4,00	1,25
15 x 4	64,00	0,22	20,00	1,51	48,00	0,25	0,00	1,53
15 x 5	80,00	0,07	0,00	1,67	44,00	0,21	0,00	1,38
15 x 6	76,00	0,08	0,00	1,97	48,00	0,07	4,00	1,09
20 x 2	56,00	0,17	12,00	1,14	40,00	0,17	0,00	2,19
20 x 3	64,00	0,19	4,00	1,66	40,00	0,18	0,00	2,15
20 x 4	56,00	0,20	0,00	1,66	48,00	0,30	0,00	1,98
20 x 5	16,00	0,53	0,00	2,31	64,00	0,06	44,00	0,15
20 x 6	36,00	0,30	0,00	2,04	36,00	0,08	56,00	0,13
25 x 2	68,00	0,14	4,00	1,24	40,00	0,22	0,00	2,92
25 x 3	76,00	0,10	4,00	1,64	20,00	0,48	0,00	2,58
25 x 4	32,00	0,31	0,00	1,87	40,00	0,14	28,00	0,32
25 x 5	32,00	0,31	0,00	2,56	40,00	0,14	28,00	0,23
25 x 6	24,00	0,25	0,00	2,25	28,00	0,24	48,00	0,18
30 x 2	76,00	0,05	0,00	0,96	24,00	0,24	0,00	2,98
30 x 3	68,00	0,10	0,00	1,60	32,00	0,31	0,00	3,12
30 x 4	48,00	0,18	0,00	2,25	36,00	0,26	20,00	0,41
30 x 5	60,00	0,22	0,00	2,60	24,00	0,38	16,00	0,71
30 x 6	56,00	0,19	0,00	2,16	8,00	0,35	36,00	0,37
Média	56,40	0,19	2,40	1,79	38,40	0,22	14,20	1,37

Tabela 21 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 100_1,5.**

A mudança radical com relação aos resultados obtidos com critério de otimização *makespan* permanecem inalterados. O método ECS_NSL novamente apresentou os melhores resultados com percentual de sucesso média entre grupos superior a 53%, o desvio médio relativo manteve-se baixo por volta de 0,25. O método ILS_RA apresentou soluções de boa qualidade, em média o percentual de sucesso ficou em 39,6% o desvio médio relativo foi de 0,26. Já o algoritmo MA_RAI obteve um percentual médio de sucesso de 15,6% e desvio médio relativo foi

ligeiramente superior a 1,34. O método TRIPS apresentou o pior desempenho com 3,27% de percentual de sucesso médio entre grupos e um desvio relativo médio de 1,79.

As tabelas 22, 23 e 24 apresentam os resultados obtidos para os problemas de maior complexidade e tempo de processamento máximo igual a 10.

SI_LG_10_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	OS	DRM	PS	DRM
50 X 10	80,00	0,09	0,00	2,67	20,00	0,54	0,00	1,01
50 X 20	64,00	0,11	0,00	2,95	20,00	0,45	16,00	0,89
50 X 30	80,00	0,08	0,00	3,36	16,00	0,42	4,00	0,77
50 X 40	64,00	0,11	0,00	3,36	32,00	0,29	4,00	0,90
100 X 10	96,00	0,00	0,00	2,45	4,00	0,92	0,00	2,54
100 X 20	88,00	0,02	0,00	2,74	12,00	0,64	0,00	2,45
100 X 30	88,00	0,02	0,00	2,79	12,00	0,66	0,00	2,59
100 X 40	84,00	0,04	0,00	3,48	16,00	0,72	0,00	2,52
150 X 10	92,00	0,03	0,00	1,99	8,00	0,94	0,00	3,37
150 X 20	92,00	0,01	0,00	2,79	8,00	0,71	0,00	3,21
150 X 30	80,00	0,04	0,00	3,04	20,00	0,62	0,00	3,22
150 X 40	96,00	0,00	0,00	3,20	4,00	0,68	0,00	3,12
200 X 10	96,00	0,02	0,00	1,23	4,00	0,67	0,00	3,61
200 X 20	84,00	0,03	0,00	1,95	16,00	0,55	0,00	3,34
200 X 30	76,00	0,04	0,00	2,66	24,00	0,61	0,00	3,50
200 X 40	92,00	0,01	0,00	2,64	8,00	0,66	0,00	3,28
Média	84,50	0,04	0,00	2,71	14,00	0,63	1,50	2,52

Tabela 22 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_0,5.

SI_LG_10_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	OS	DRM	PS	DRM
50 X 10	80,00	0,07	0,00	2,45	16,00	0,62	4,00	0,91
50 X 20	72,00	0,14	0,00	2,63	12,00	0,51	16,00	0,77
50 X 30	80,00	0,07	0,00	2,60	16,00	0,44	4,00	0,98
50 X 40	72,00	0,08	0,00	2,80	24,00	0,43	4,00	0,83
100 X 10	96,00	0,01	0,00	2,21	4,00	0,78	0,00	2,30
100 X 20	92,00	0,05	0,00	2,65	8,00	0,75	0,00	2,27
100 X 30	96,00	0,02	0,00	2,65	4,00	0,66	0,00	1,92
100 X 40	96,00	0,00	0,00	2,50	4,00	0,64	0,00	2,18
150 X 10	100,00	0,00	0,00	1,82	0,00	0,74	0,00	2,92
150 X 20	96,00	0,00	0,00	2,21	4,00	0,66	0,00	2,73
150 X 30	92,00	0,01	0,00	2,39	8,00	0,70	0,00	2,59
150 X 40	88,00	0,03	0,00	2,50	12,00	0,60	0,00	2,85
200 X 10	92,00	0,02	0,00	0,93	8,00	0,47	0,00	3,05
200 X 20	100,00	0,00	0,00	1,62	0,00	0,61	0,00	2,89
200 X 30	80,00	0,07	0,00	2,06	20,00	0,41	0,00	2,81
200 X 40	88,00	0,04	0,00	2,39	12,00	0,57	0,00	3,03
Média	88,75	0,04	0,00	2,28	9,50	0,60	1,75	2,19

Tabela 23 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_1,0.

SI_LG_10_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	OS	DRM	PS	DRM
50 X 10	72,00	0,09	0,00	2,24	20,00	0,50	8,00	1,07
50 X 20	68,00	0,14	0,00	2,21	24,00	0,32	8,00	0,74
50 X 30	76,00	0,10	0,00	2,48	20,00	0,41	4,00	0,72
50 X 40	68,00	0,06	0,00	2,45	28,00	0,32	4,00	0,63
100 X 10	96,00	0,00	0,00	1,77	4,00	0,55	0,00	2,05
100 X 20	96,00	0,01	0,00	2,21	4,00	0,59	0,00	1,85
100 X 30	100,00	0,00	0,00	2,71	0,00	0,71	0,00	1,99
100 X 40	96,00	0,01	0,00	2,61	4,00	0,64	0,00	2,06
150 X 10	100,00	0,00	0,00	1,42	0,00	0,68	0,00	2,53
150 X 20	96,00	0,01	0,00	1,89	4,00	0,52	0,00	2,34
150 X 30	92,00	0,03	0,00	2,06	8,00	0,55	0,00	2,40
150 X 40	92,00	0,01	0,00	2,32	8,00	0,63	0,00	2,34
200 X 10	84,00	0,03	0,00	1,03	16,00	0,36	0,00	2,86
200 X 20	96,00	0,01	0,00	1,41	4,00	0,47	0,00	2,39
200 X 30	88,00	0,04	0,00	1,57	12,00	0,38	0,00	2,49
200 X 40	92,00	0,02	0,00	1,76	8,00	0,45	0,00	2,33
Média	88,25	0,03	0,00	2,01	10,25	0,50	1,50	1,92

Tabela 24 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10_1,5.

Novamente emerge a característica de que os resultados não apresentam muita diferenciação entre os grupos.

O método ECS_NSL apresentou a melhor solução, a média dos percentuais de sucesso entre grupos também se manteve alta representando acima de 87,17% dos casos, o desvio padrão médio foi baixo para todas as classes. O método ILS_RA atingiu 11,25%, seu desvio médio relativo se manteve baixo o que indica que embora não tenha sido capaz de gerar as melhores soluções, o método gerou soluções de boa qualidade.

Por outro lado os métodos TRIPS e MA_RAI não foram boas soluções, o MA_RAI obteve uma média de percentual de sucesso entre grupos de 1,58%, já o algoritmo TRIPS não gerou nenhuma das melhores soluções. Já o desvio relativo médio do método MA_RAI ficou em 2,21%. O algoritmo TRIPS foi o que apresentou o maior desvio padrão entre todos os métodos.

As tabelas 25, 26 e 27 apresentam os resultados obtidos para os problemas de maior complexidade e tempo de processamento máximo igual a 100.

SI_LG_100_0,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	OS	DRM	PS	DRM
50 X 10	76,00	0,10	0,00	2,48	20,00	0,59	4,00	1,28
50 X 20	68,00	0,20	0,00	3,25	32,00	0,38	0,00	1,05
50 X 30	76,00	0,09	0,00	3,06	16,00	0,48	8,00	1,07
50 X 40	76,00	0,16	0,00	3,28	24,00	0,47	0,00	1,15
100 X 10	88,00	0,05	0,00	2,35	12,00	0,72	0,00	2,30
100 X 20	100,00	0,00	0,00	2,98	0,00	0,85	0,00	2,87
100 X 30	96,00	0,01	0,00	3,44	4,00	0,94	0,00	3,05
100 X 40	96,00	0,01	0,00	3,66	4,00	0,76	0,00	2,77
150 X 10	96,00	0,00	0,00	1,94	4,00	0,80	0,00	2,90
150 X 20	96,00	0,00	0,00	2,58	4,00	0,78	0,00	3,22
150 X 30	88,00	0,09	0,00	3,21	12,00	0,63	0,00	3,26
150 X 40	96,00	0,03	0,00	3,18	4,00	0,75	0,00	3,46
200 X 10	92,00	0,02	0,00	1,39	8,00	0,63	0,00	3,09
200 X 20	72,00	0,07	0,00	2,09	28,00	0,53	0,00	3,36
200 X 30	84,00	0,06	0,00	2,48	16,00	0,49	0,00	3,33
200 X 40	80,00	0,13	0,00	3,08	20,00	0,50	0,00	3,27
Média	86,25	0,06	0,00	2,78	13,00	0,64	0,75	2,59

Tabela 25 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100_0,5.

SI_LG_100_1,0								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	OS	DRM	OS	DRM
50 X 10	76,00	0,11	0,00	2,51	20,00	0,54	4,00	0,76
50 X 20	76,00	0,08	0,00	2,77	24,00	0,34	0,00	0,82
50 X 30	80,00	0,04	0,00	2,64	16,00	0,42	4,00	0,87
50 X 40	76,00	0,07	0,00	3,08	16,00	0,37	8,00	0,94
100 X 10	88,00	0,02	0,00	1,87	12,00	0,74	0,00	2,06
100 X 20	92,00	0,02	0,00	2,46	8,00	0,68	0,00	2,03
100 X 30	84,00	0,03	0,00	2,67	16,00	0,71	0,00	2,31
100 X 40	88,00	0,07	0,00	3,23	12,00	0,73	0,00	2,09
150 X 10	100,00	0,00	0,00	1,76	0,00	0,73	0,00	2,69
150 X 20	96,00	0,00	0,00	2,16	4,00	0,72	0,00	2,59
150 X 30	88,00	0,02	0,00	2,49	12,00	0,69	0,00	2,59
150 X 40	88,00	0,02	0,00	2,81	12,00	0,56	0,00	2,80
200 X 10	76,00	0,06	4,00	1,29	20,00	0,38	0,00	2,83
200 X 20	100,00	0,00	0,00	1,75	0,00	0,63	0,00	2,76
200 X 30	80,00	0,03	0,00	2,07	20,00	0,55	0,00	2,87
200 X 40	68,00	0,06	0,00	2,45	32,00	0,34	0,00	3,03
Média	84,75	0,04	0,25	2,38	14,00	0,57	1,00	2,13

Tabela 26 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100_1,0.

SI_LG_100_1,5								
Problemas	ECS_NSL		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
50 X 10	88,00	0,02	0,00	2,13	4,00	0,60	8,00	0,98
50 X 20	68,00	0,08	0,00	2,63	32,00	0,43	0,00	0,82
50 X 30	60,00	0,17	0,00	2,81	32,00	0,29	8,00	0,82
50 X 40	68,00	0,13	0,00	2,62	20,00	0,35	12,00	0,67
100 X 10	84,00	0,02	0,00	1,81	16,00	0,55	0,00	1,74
100 X 20	88,00	0,03	0,00	2,42	12,00	0,59	0,00	2,00
100 X 30	100,00	0,00	0,00	2,50	0,00	0,82	0,00	1,90
100 X 40	88,00	0,02	0,00	2,48	12,00	0,61	0,00	1,92
150 X 10	100,00	0,00	0,00	1,61	0,00	0,65	0,00	2,38
150 X 20	100,00	0,00	0,00	1,91	0,00	0,77	0,00	2,26
150 X 30	92,00	0,02	0,00	2,33	8,00	0,51	0,00	2,27
150 X 40	84,00	0,01	0,00	2,35	16,00	0,53	0,00	2,41
200 X 10	84,00	0,01	0,00	1,10	16,00	0,34	0,00	2,49
200 X 20	84,00	0,03	0,00	1,47	16,00	0,42	0,00	2,40
200 X 30	88,00	0,03	0,00	1,76	12,00	0,48	0,00	2,38
200 X 40	88,00	0,02	0,00	1,98	12,00	0,34	0,00	2,35
Média	85,25	0,04	0,00	2,12	13,00	0,52	1,75	1,86

Tabela 27 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 100_1,5.**

O método ECS_NSL apresentou a melhor solução, a média dos percentuais de sucesso entre grupos também se manteve alta representando acima de 85,42% dos casos, o desvio padrão médio foi baixo para todas as classes. O método ILS_RA atingiu 13,33%, seu desvio médio relativo se manteve baixo o que indica que embora não tenha sido capaz de gerar as melhores soluções, o método gerou soluções de boa qualidade.

Por outro lado os métodos TRIPS e MA_RAI não foram geraram boas soluções, o MA_RAI obteve uma média de percentual de sucesso entre grupos de 1,16%, já o algoritmo TRIPS não gerou nenhuma das melhores soluções. O desvio relativo médio do método MA_RAI ficou em 2,19. O algoritmo TRIPS foi o que apresentou o maior desvio padrão entre todos os métodos.

O Gráfico 3 apresenta as médias dos DRMs para os problemas apresentados acima, nota-se que os métodos ECS_NSL e ILS_RA obtiveram as melhores soluções para problemas de pequeno porte, no entanto verifica-se a mesma tendência observada para problemas com critério de Makespan onde em problemas de maior complexidade o método proposto ECS_NSL apresenta os melhores resultados.

Gráfico 3 – DRM para problemas SI com critério TFT.

O Gráfico 4 apresenta as médias dos percentuais de sucesso para os problemas apresentados acima, observa-se que o método ECS_NSL obteve as melhores soluções sendo que o método ILS_RA também gerou soluções de qualidade para problemas pequenos porte.

Gráfico 4 – PS para problemas SI com critério TFT.

7.2 No-wait Flowshop com setup dependente

A análise dos problemas é subdividida em duas partes de acordo com o critério de otimização em estudo. Conforme mencionado anteriormente, para cada problema será apresentado estatísticas referentes ao percentual de sucesso e ao desvio relativo médio.

O método TRIPS mostrou-se computacionalmente inviável de ser aplicado para problemas com 500 tarefas, logo seus resultados foram desconsiderados.

7.2.1 Critério de Minimização do *Makespan*

As tabelas 28, 29, 30 e 31 apresentam os resultados obtidos para os problemas com menor critério de otimização *makespan*.

SDST10								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	30,00	0,32	0,00	10,08	60,00	0,24	100,00	0,00
20 X 10	60,00	0,21	0,00	9,37	80,00	0,11	80,00	0,11
20 X 20	80,00	0,04	0,00	10,91	70,00	0,02	70,00	0,10
50 X 5	30,00	0,44	0,00	9,41	0,00	2,10	70,00	0,23
50 X 10	90,00	0,05	0,00	9,42	0,00	1,44	10,00	0,76
50 X 50	100,00	0,00	0,00	11,23	0,00	1,44	0,00	1,38
100 X 5	40,00	0,43	0,00	8,16	0,00	2,33	60,00	0,13
100 X 10	100,00	0,00	0,00	7,84	0,00	1,19	0,00	0,97
100 X 20	100,00	0,00	0,00	8,95	0,00	1,37	0,00	1,51
200 X 10	90,00	0,06	0,00	6,73	0,00	1,08	10,00	0,87
200 x 20	90,00	0,02	0,00	5,95	10,00	0,61	0,00	1,41
500 X 20	70,00	0,08	-	-	30,00	0,24	0,00	1,32
Média	73,33	0,14	0,00	8,91	20,83	1,01	33,33	0,73

Tabela 28 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 10.

SDST50								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	60,00	0,17	0,00	9,34	40,00	0,27	100,00	0,00
20 X 10	60,00	0,07	0,00	8,79	60,00	0,04	70,00	0,04
20 X 20	80,00	0,03	0,00	9,85	40,00	0,10	50,00	0,09
50 X 5	60,00	0,15	0,00	8,06	0,00	1,75	40,00	0,26
50 X 10	90,00	0,03	0,00	8,01	0,00	1,19	10,00	0,68
50 X 50	80,00	0,07	0,00	9,61	0,00	0,99	20,00	0,92
100 X 5	80,00	0,05	0,00	6,69	0,00	1,90	20,00	0,35
100 X 10	80,00	0,05	0,00	7,66	0,00	1,26	20,00	0,83
100 X 20	100,00	0,00	0,00	7,71	0,00	1,15	0,00	1,42
200 X 10	100,00	0,00	0,00	4,80	0,00	0,83	0,00	0,68
200 x 20	90,00	0,03	0,00	4,57	0,00	0,58	10,00	1,08
500 X 20	60,00	0,04	-	-	40,00	0,13	0,00	0,93
Média	78,33	0,06	0,00	7,74	15,00	0,85	28,33	0,61

Tabela 29 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 50.

SDST100								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	30,00	0,39	0,00	9,66	20,00	0,47	100,00	0,00
20 X 10	40,00	0,19	0,00	8,73	60,00	0,13	70,00	0,15
20 X 20	70,00	0,06	0,00	9,48	40,00	0,17	70,00	0,07
50 X 5	30,00	0,51	0,00	6,82	0,00	2,56	70,00	0,13
50 X 10	60,00	0,06	0,00	6,38	0,00	1,64	40,00	0,38
50 X 50	100,00	0,00	0,00	7,25	0,00	1,14	0,00	1,04
100 X 5	40,00	0,50	0,00	6,22	0,00	2,66	60,00	0,24
100 X 10	50,00	0,12	0,00	5,07	0,00	1,26	50,00	0,15
100 X 20	80,00	0,01	0,00	5,56	10,00	0,89	10,00	0,93
200 X 10	90,00	0,05	0,00	3,68	0,00	1,17	10,00	0,73
200 x 20	100,00	0,00	0,00	3,35	0,00	0,59	0,00	0,95
500 X 20	50,00	0,08	-	-	50,00	0,17	0,00	0,78
Média	61,67	0,16	0,00	6,56	15,00	1,07	40,00	0,46

Tabela 30 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100.

SDST125								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	40,00	0,37	0,00	9,02	0,00	0,49	100,00	0,00
20 X 10	50,00	0,17	0,00	7,21	30,00	0,27	70,00	0,05
20 X 20	80,00	0,04	0,00	8,02	80,00	0,07	60,00	0,09
50 X 5	50,00	0,39	0,00	8,92	0,00	2,77	50,00	0,11
50 X 10	50,00	0,23	0,00	6,39	0,00	1,46	50,00	0,19
50 X 50	100,00	0,00	0,00	6,99	0,00	1,10	0,00	0,60
100 X 5	50,00	0,21	0,00	5,88	0,00	3,00	50,00	0,29
100 X 10	50,00	0,25	0,00	4,58	0,00	1,39	50,00	0,29
100 X 20	90,00	0,02	0,00	5,38	0,00	1,14	10,00	0,62
200 X 10	100,00	0,00	0,00	3,02	0,00	1,19	0,00	0,66
200 x 20	70,00	0,07	0,00	2,91	30,00	0,37	0,00	0,61
500 X 20	70,00	0,02	-	-	30,00	0,16	0,00	0,61
Média	66,67	0,15	0,00	6,21	14,17	1,12	36,67	0,34

Tabela 31 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 125.**

Os resultados obtidos sugerem que a variabilidade dos custos de *setup* influencia pouco na solução. ECS_NSL foi o melhor método utilizado para os problemas, seu percentual de sucesso médio chegou foi de 70% enquanto seu desvio relativo médio ficou em 0,13. A experimentação utilizando o MA_RAI resultou em um percentual de sucesso médio de 34, 58% e seu desvio médio relativo foi de 0,35. O método ILS_RA obteve sucesso em 16,25% dos testes realizados, seu desvio relativo médio foi observado em 1,01. Por fim, o algoritmo TRIPS não obteve sucesso com suas soluções e seu desvio relativo médio foi o maior de todos os métodos.

desvio padrão entre todos os métodos.

O Gráfico 5 sumariza a informação de DRMs obtida para os problemas apresentados acima, nota-se que os métodos ECS_NSL, ILS_RA e MA_RAI obtiveram boas soluções, no entanto observa-se que o método ECS_NSL foi o que apresentou as melhores soluções.

Gráfico 5 – DRM para problemas SD com critério Makespan.

O Gráfico 6 apresenta as médias dos percentuais de sucesso médio obtidos para os problemas acima, verifica-se que o método proposto foi gerador das melhores soluções, seguido pelo método MA_RAI.

Gráfico 6 – PS para problemas SD com critério Makespan.

7.2.1 Critério de Minimização do TFT

As tabelas 32, 33, 34 e 35 apresentam os resultados obtidos para os problemas com menor critério de otimização TFT.

De maneira análoga ao que foi observado para os problemas de *setup* independente, os comportamentos dos métodos mudam significativamente conforme o critério de desempenho. Observou-se que embora a sua vantagem relativa aos outros métodos tenha diminuído, o algoritmo ECS_NSL continua a gerar as melhores soluções. O percentual médio de sucesso entre todos os grupos de problemas foi de 49,38% o seu desvio relativo médio foi de 0,40. ILS_RA e MA_RAI obtiveram resultados de boa qualidade onde seus percentuais de sucesso

foram respectivamente de 30,41% e 28,54%, além disso, seus desvios médios relativos foram de 0,75 e 1,42 respectivamente. O método TRIPS obteve as piores soluções com 11,88% de percentual de sucesso e 1,98 de desvio médio relativo.

SDST10								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	40,00	0,38	0,00	2,06	60,00	0,04	80,00	0,09
20 X 10	60,00	0,23	0,00	2,58	100,00	0,00	90,00	0,01
20 X 20	50,00	0,27	0,00	1,94	100,00	0,00	10,00	0,13
50 X 5	0,00	0,92	0,00	2,97	10,00	1,28	90,00	0,04
50 X 10	70,00	0,18	0,00	3,21	10,00	0,92	20,00	0,62
50 X 50	90,00	0,21	0,00	4,16	10,00	0,69	0,00	1,65
100 X 5	90,00	0,15	0,00	1,85	0,00	1,90	10,00	0,98
100 X 10	100,00	0,00	0,00	3,36	0,00	1,00	0,00	2,54
100 X 20	80,00	0,01	0,00	3,38	20,00	0,61	0,00	3,06
200 X 10	100,00	0,00	0,00	1,89	0,00	1,01	0,00	3,88
200 X 20	60,00	0,26	0,00	2,28	40,00	0,30	0,00	4,36
500 X 20	20,00	0,47	-	-	80,00	0,12	0,00	4,40
Média	63,33	0,26	0,00	2,70	35,83	0,65	25,00	1,81

Tabela 32 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 10.**

SDST50								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	0,00	0,52	0,00	2,46	50,00	0,09	70,00	0,02
20 X 10	30,00	0,31	0,00	3,05	90,00	0,01	70,00	0,12
20 X 20	50,00	0,31	0,00	1,65	60,00	0,02	20,00	0,17
50 X 5	40,00	0,67	0,00	2,91	0,00	1,65	60,00	0,05
50 X 10	30,00	0,59	0,00	3,16	0,00	0,72	70,00	0,22
50 X 50	70,00	0,12	0,00	3,31	30,00	0,42	0,00	1,17
100 X 5	60,00	0,14	10,00	1,31	0,00	1,57	30,00	0,73
100 X 10	100,00	0,00	0,00	2,14	0,00	0,94	0,00	1,74
100 X 20	80,00	0,08	0,00	3,53	20,00	0,98	0,00	3,05
200 X 10	70,00	0,04	10,00	0,95	20,00	0,73	0,00	3,20
200 X 20	50,00	0,14	0,00	1,13	50,00	0,27	0,00	3,38
500 X 20	10,00	0,47	-	-	90,00	0,01	0,00	3,63
Média	49,17	0,28	1,67	2,33	34,17	0,62	26,67	1,46

Tabela 33 – Percentual de Sucesso* e Desvio Rel. Médio para a classe 50.**

SDST100								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	30,00	0,86	0,00	2,59	50,00	0,12	100,00	0,00
20 X 10	10,00	0,64	0,00	2,15	60,00	0,05	60,00	0,05
20 X 20	30,00	0,18	0,00	1,69	60,00	0,04	80,00	0,06
50 X 5	0,00	1,79	0,00	2,22	0,00	2,13	100,00	0,00
50 X 10	60,00	0,38	0,00	2,38	10,00	0,84	30,00	0,34
50 X 50	80,00	0,18	0,00	2,46	20,00	0,60	0,00	0,76
100 X 5	30,00	0,73	30,00	0,61	0,00	1,88	40,00	0,60
100 X 10	80,00	0,04	20,00	0,58	0,00	0,97	0,00	1,35
100 X 20	100,00	0,00	0,00	1,95	0,00	0,72	0,00	2,10
200 X 10	40,00	0,29	50,00	0,18	10,00	1,06	0,00	3,01
200 x 20	80,00	0,01	10,00	0,62	10,00	0,40	0,00	2,78
500 X 20	20,00	0,48	-	-	80,00	0,02	0,00	3,14
Média	46,67	0,47	17,50	1,58	25,00	0,74	34,17	1,18

Tabela 34 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 100.

SDST125								
Problemas	ECS		TRIPS		ILS_RA		MA_RAI	
	PS*	DRM**	PS	DRM	PS	DRM	PS	DRM
20 X 5	0,00	1,12	10,00	1,79	60,00	0,17	80,00	0,01
20 X 10	40,00	0,73	0,00	2,04	50,00	0,11	90,00	0,02
20 X 20	50,00	0,32	0,00	1,60	50,00	0,08	30,00	0,17
50 X 5	10,00	1,66	20,00	1,91	0,00	2,42	70,00	0,13
50 X 10	30,00	0,68	0,00	2,02	20,00	1,10	50,00	0,24
50 X 50	70,00	0,22	0,00	2,19	20,00	0,72	10,00	0,63
100 X 5	20,00	0,78	70,00	0,11	0,00	2,87	10,00	1,24
100 X 10	50,00	0,09	50,00	0,39	0,00	1,07	0,00	1,24
100 X 20	90,00	0,05	0,00	1,82	10,00	0,94	0,00	2,02
200 X 10	20,00	1,05	80,00	0,01	0,00	2,00	0,00	3,78
200 x 20	60,00	0,16	10,00	0,46	30,00	0,48	0,00	2,72
500 X 20	20,00	0,29	-	-	80,00	0,05	0,00	2,64
Média	38,33	0,60	28,33	1,30	26,67	1,00	28,33	1,24

Tabela 35 – Percentual de Sucesso* e Desvio Rel. Médio** para a classe 125.

O Gráfico 7 sintetiza a informação de DRMs obtida para os problemas apresentados acima, nota-se que o método proposto ECS_NSL obteve as melhores soluções para todas as classes de problemas.

Gráfico 7 – DRM para problemas SD com critério Makespan.

O Gráfico 8 sintetiza a informação de percentuais de sucesso obtida para os problemas apresentados acima. O método ECS_NSL obteve as melhores soluções para todas as classes de problemas.

Gráfico 8 – PS para problemas SD com critério Makespan.

CAPÍTULO 8

CONSIDERAÇÕES FINAIS

O presente trabalho abordou o problema de sequenciar n tarefas em m máquinas em um ambiente de produção NWFS com tempos de setup dependentes e independentes. Foram considerados dois objetivos para a programação, a minimização do *makespan* e do *total flowtime*, ambos de alta relevância nas linhas de produção.

Um longo período de experimentação computacional foi necessário para que fosse possível solucionar a base de dados que compôs este trabalho. Os

resultados obtidos e discutidos no capítulo anterior nos levam a crer que o melhor método para estes tipos de problemas é o ECS_NSL.

O método se mostrou altamente flexível, sendo capaz de gerar soluções de qualidade para os mais diversos tipos de problemas sob as mais diversas complexidades. Faz-se necessário ressaltar que durante a construção e adaptação do método, priorizou-se o desempenho para problemas de grande porte que geralmente são os mais difíceis de se resolver, nessa busca sacrificou-se parcialmente o desempenho do método para problemas de pequeno porte.

Por se tratar de um algoritmo relativamente complexo e com diversos critérios parametrizáveis, há diversas combinações possíveis de parâmetros que podem resultar em melhorias no seu desempenho. Outro ponto de pesquisa é a técnica de recombinação genética, existem outros diversos métodos relatados na literatura que também podem ocasionar em melhorias.

Trabalhos futuros podem incluir a adaptação do método para os mais diversos problemas de planejamento como Flowshop No-idle, Flowshop com datas de liberação, Jobshop, Openshop, etc.

REFERÊNCIAS

ADIRI, T.; POHORYLES, D. Flowshop/no-idle or no-wait scheduling to minimize the sum of completion times. **Naval Research Logistics Quartely**, Hoboken, vol. 29, n. 3, p. 495-504, set. 1982.

ALDOWAISAN, T. A new heuristic and dominance relations for no-wait flowshops with setups. **Computers & Operations Research**, Oxford, vol. 28, n. 6, p. 563-584, mai. 2001.

_____.; ALLAHVERDI, A. New heuristics for m-machine no-wait flowshop to minimize total completion time. **Omega, The international Journal of Management Science**, Oxford, vol. 32, n. 5, p. 345-352, out. 2004a.

_____.; ALLAHVERDI, A. Total flowtime in no-wait flowshops with separated setup times. **Computers & Operations Research**, Oxford, vol. 25, n. 9, p. 757-765, set. 1998.

ALLAHVERDI, A. Scheduling in stochastic flowshops with independent setup, processing and removal times. **Computers & Operations Research**, Oxford, v. 24, p. 955-960, 1997.

ALLAHVERDI, A. Two-stage production scheduling with separated set-up times and stochastic breakdowns. **Journal of the Operational Research Society**, v. 46, p. 895-904, 1995.

_____.; ALDOWAISAN, T. No-wait and separate setup three-machine flowshop with total completion time criterion. **International Transactions in Operational Research**, vol. 7, n. 3, p. 245-264, mai. 2000.

_____.; ALDOWAISAN, T. No-wait flowshops with bicriteria of makespan and maximum lateness. **European Journal of Operational Research**, Amsterdam, vol. 152, n. 1, p. 132-147, jan. 2004.

_____.; GUPTA, J. N. D.; ALDOWAISAN, T. A review of scheduling research involving setup considerations. **Omega, The international Journal of Management Science**, Oxford, vol. 27, n. 2, p. 219-239, abr. 1999.

ARAÚJO, D. C.; NAGANO, M. S. An effective heuristic for the no-wait flowshop with sequence-dependent setup times problem. **Lecture Notes in Computer Science**, vol. 6437, p. 187-196, 2010.

ARORA, S.; BARAK, B. **Computational Complexity: A Modern Approach**. Cambridge University Press, New York, 2009.

BAKER, K. R. **Introduction to Sequencing and Scheduling**, Wiley, New York, 1974.

BERTOLISSI, E. A simple no-wait flowshop scheduling for the no-wait flowshop problem. In: THE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED PRODUCTION ENGINEERING, 15., 1999, Durham. **Proceedings...** Durham: University of Durham Publishers, 1999, p. 750–755.

BERTOLISSI, E. Heuristic algorithm for scheduling in the no-wait flow-shop. **Journal of Materials Processing Technology**, Oxford, vol. 107, n. 1-3, p. 459-465, nov. 2000.

BIANCO, L.; DELL'OLMO, P.; GIORDANI, S. Flow shop no-wait scheduling with sequence dependent setup times and release dates. **INFOR Journal**, Toronto, vol. 37, n. 1, p. 3-19, 1999.

BROWN, S. I.; MCGARVEY, R.; VENTURA, J. A. Total flowtime and makespan for a no-wait m-machine flowshop with set-up times separated. **Journal of the Operational Research Society**, Oxford, vol. 55, n. 6, p. 614-621, mai. 2004.

BRUNO, J.; DOWNEY, P. Complexity of task sequencing with deadlines, set-up times and changeover costs. **SIAM Journal on Computing**, vol. 7, p. 393-404, 1978.

CHEN, C.-L.; NEPPALLI, R. V.; ALJABER, N. Genetic algorithms applied to the continuous flow shop problem. **Computers & Industrial Engineering**, Oxford, vol. 30, n. 4, p. 919-929, set. 1996.

CONWAY, R. W.; MAXWELL, W. L.; MILLER, L.W. **Theory of Scheduling**, Addison Wesley, Reading, Mass, 1967.

CORWIN, B. D.; ESOGBUE, A. O. Two machine flow shop scheduling problems with sequence dependent setup times: a dynamic programming approach. **Naval Research Logistics Quarterly**, Hoboken, vol. 21, n. 3, p. 515–524, set. 1974.

DILEEPAN, P.; SEN T. Job lateness in a two-machine flowshop with setup times separated. **Computers & Operations Research**, vol. 18, p. 549-556, 1991.

EMMONS, H. One Machine Sequencing to Minimize Mean Time with Minimum Number Tardy. **Naval Research Logistics Quarterly**, vol. 22, n. 3, set. 1975.

ERDMANN, R. H. **Administração da produção: planejamento, programação e controle**. Florianópolis: Papa Livro, 2000.

FILHO, G. R.; NAGANO, M. S.; LORENA, L. A. N. Evolutionary Clustering Search for Flowtime Minimization in Permutation Flow Shop. **Lecture Notes in Computer Science** 4771: 69-81.

FINK, A.; VOß, S. Solving the continuous flow-shop scheduling problem by metaheuristics. **European Journal of Operational Research**, Amsterdam, vol. 151, n. 2, p. 400-414, dez. 2003.

_____.; NAGANO, M. S.; MOCCELLIN, J. V. An efficient heuristic for total flowtime minimization in no-wait flowshops. **International Journal of Advanced Manufacturing Technology**, London, vol. 46, n. 9-12, p. 1049-1057, fev. 2010.

FRANÇA, P. M.; TIN JR, G.; BURRIOL, L. S. Genetic Algorithms for the no-wait flowshop sequencing problem with time restrictions. **International Journal of Production Research**, London, vol. 44, n.5, p. 939-957, 2006.

GANGADHARAN, R.; RAJENDRAN, C. Heuristic algorithms for scheduling in the no-wait flowshop. **International Journal of Production Economics**, Amsterdam, vol. 32, n. 3, p. 285-290, nov. 1993.

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability**: a guide to the theory of NP-completeness. W. H Freeman and Company, 1979.

GAREY, M. R., JOHNSON, D. S., SETHI, R. The Complexity of flowshop and jobshopscheduling. **Mathematics of Operations Research**, vol. 1, p. 117-129, 1976.

_____.; DARROW, W. P. Approximate schedules for the two-machine flowshop with sequence dependent setup times. **Indian Journal of Management Systems**, Delhi, vol. 1, p. 6-11, 1985.

_____.; DAS, S. R.; GHOSH, S. Flowshop **Scheduling with Sequence Dependent Setup Times**. Muncie: Department of Management - Ball State University, 1995. (Working paper, IN 47306).

GLOVER, F. Tabu search and adaptive memory programming: Advances, applications and challenges. **Interfaces in Computer Science and Operations Research**: 1–75 1996.

HALL, N. G.; SRISKANDARAJAH, C. A survey of machine scheduling problems with blocking and no-wait in process. **Operations Research**, Hannover, vol. 44, n. 3, p. 510-525, mai./jun. 1996.

JARBOUI, B.; EDDALY, M.; SIARRY, P. A hybrid genetic algorithm for solving no-wait flowshop scheduling problems. **International Journal of Advanced Manufacturing Technology**, DOI 10.1007/s00170-010-3009-4, 2010.

JOHNSON, L. A.; MONTGOMERY, D. C. **Operations research in production planning, scheduling, and inventory control**. New York: John Wiley & Sons, 1974.

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

KHURANA, K.; BAGGA P. C. Scheduling of job-block with deadline in $n \times 2$ flowshop problem with separated setup times. **Indian Journal of Pure and Applied Mathematics**, v. 16, p. 213-224, 1985.

KRAJEWSKI, L. J. et al. Kanban, MRP, and shaping the manufacturing environment. **Faculty of Management Science**, London, vol. 33, n. 1, p. 39-57, 1987.

_____.; NAGANO, M. S. Uma propriedade estrutural do problema de programação da produção flow shop permutacional com tempos de setup. **Pesquisa Operacional**, Rio de Janeiro, vol. 27, n. 3, p. 487-515, set./dez. 2007.

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

NAGANO, M. S.; MOCCELLIN, J. V. A high quality solution constructive heuristic for flow shop sequencing. **Journal of the Operational Research Society**, Oxford, v. 53, n. 12, p. 1374-1379, dez. 2002.

_____.; RUIZ, R.; LORENA, L. A. N. A constructive genetic algorithm for permutation flowshop scheduling. **Computers & Industrial Engineering**, vol. 55, p. 195-207, 2008.

NAWAZ, M.; ENSCORE JR., E. E.; HAM, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. **Omega, The International Journal of Management Science**, Oxford, vol. 11, n. 1, p. 91-95, 1983.

OLIVEIRA, A.C.M.; LORENA, L.A.N. Detecting promising areas by evolutionary clustering search. In: BAZZAN, A.L.C., LABIDI, S. **SBIA 2004: Advances in Artificial Intelligence**. São Luis: Springer Heidelberg, 2004. LNAI 3171, p. 385-394.

PAN, Q.-P.; TASGETIREN, M. F.; LIANG, Y.-C. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. **Computers & Operations Research**, Oxford, vol. 45, p. 2807-2839, 2007a.

PAN, Q.-P.; WANG, L.; TASGETIREN, M. F. A hybrid discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem with makespan criterion. **International Journal of Advanced Manufacturing Technology**, London, vol. 38, p. 337-347, 2007b.

PANWALKAR, S. S.; DUDEK, R. A.; SMITH, M. L. Sequencing research and the industrial scheduling problem. In: **THEORY OF SCHEDULING AND ITS APPLICATIONS**, 1973, New York. **Symposium...** New York: Springer-Verlag, 1973, p. 29–38.

PINEDO, M. **Scheduling: theory, algorithms, and systems**. New Jersey: Prentice Hall, 2008.

PROUST, C.; GUPTA, J. N. D.; DESCHAMPS, V. Flowshops scheduling with set-up, processing and removal times separated. **International Journal of Production Research**, London, v. 29, n. 3, p. 479-493, mar. 1991.

_____.; CHAUDHURI, D. Heuristic algorithms for continuous flow-shop problem. **Naval Research Logistics**, Hoboken, vol. 37, n. 5, p. 695-705, 1990.

_____.; ZIEGLER, H. Heuristics for scheduling in a flowshop with setup, processing and removal times separated. **Production Planning & Control**, Abington, vol. 8, n. 6, p. 568-576, set. 1997.

RIBEIRO FILHO, G.; NAGANO, M. S.; LORENA, L.A.N. Hybrid Evolutionary Algorithm for Flowtime Minimisation in No-Wait Flowshop Scheduling. In: GELBUKH, A.; MORALES, A. F. K. **MICAI 2007: Advances in Artificial Intelligence**. Aguascalientes: Springer Berlin/Heidelberg, 2007. LNAI 4827, p. 1099-1109.

_____.; BARD, J. Heuristics for the flow line problem with setup costs. **European Journal of Operational Research**, Amsterdam, vol. 110, n. 1, p. 76-98, out. 1998b.

_____.; ALLAHVERDI, A. Some effective heuristics for no-wait flowshops with setup times to minimize total completion time. **Annals of Operations Research**, Amsterdam, vol. 156, n. 1, p. 143-171, dez. 2007.

SIMONS, J. V. Heuristics in flow shop scheduling with sequence dependent setup times. **Omega, The International Journal of Management Science**, Oxford, vol. 20, n. 2, p. 215-225, mar. 1992.

SMITH, W. E. **Various optimizers for single-stage production.** Naval Research Logistics Quarterly, vol. 3, p. 59-66, 1956.

SRIKAR, B. N.; GHOSH, S. A MILP model for the n-job, m-stage flowshop, with sequence dependent setup times. **International Journal of Production Research**, London, vol. 24, n. 6, p. 1459-72, nov. 1986.

_____.; TSENG, F.T. Two models for a family of flowshop sequencing problems. **European Journal of Operational Research**, Amsterdam, vol. 142, n. 2, p. 282–293, out. 2002.

SULE, D. R.; Huang, K. Y. Sequency on two and three machines with setup, processing and removal times separated. **International Journal of Production Research**, vol. 21, p. 723-732, 1983.

SYSWERDA, G. Uniform crossover in genetic algorithms. **International Conference on Genetic Algorithms (ICGA)**, Virginia, USA: 2–9.

TAILLARD, E. Benchmarks for basic scheduling problems. **European Journal of Operational Research**, Amsterdam, v. 64, n. 2, p. 278-285, jan. 1993.

TSENG, L.-Y; LIN, Y.-T. A hybrid genetic algorithm for no-wait flowshop scheduling problem. . **International Journal of Production Economics**, vol. 128, p. 144-152, 2010.

USKUP G.; SMITH S. B. A branch and bound for two- stage production sequencing. **Operations Research**, Hannover, vol. 23, n. 1, p. 118-136, jan./fev. 1975.

VAN DER VEEN, J. A. A.; VAN DAL, R. Solvable Cases of the No-Wait Flow-Shop Scheduling Problem. *The Journal of the Operational Research Society*, Oxford, vol. 42, n. 11, p. 971-980, nov. 1991.

WILD, R. **Concepts for Operations Management**, London, Wiley, 1981.

WILBRECHT J. K.; PRESCOTT, W. B. The influence of setup time on job shop performance. **Faculty of Management Science**, London, vol. 16, n. 4, p. B274-B280, 1969.

WORTMAN, D. B. Managing capacity: getting the most from your firm's assets. **Industrial Engineering**, vol. 24, n. 2, p. 47-9, 1992.

YOSHIDA, T.; HITOMI, K. Optimal two-stage production scheduling with setup times separated. **AIIE Trans**, vol. 11, p. 261-263, 1979.