

Lucas Yamada Scardoelli

NOVOS MÉTODOS HEURÍSTICOS PARA A PROGRAMAÇÃO DE OPERAÇÕES *NO-WAIT FLOW SHOP* COM CRITÉRIO DE MINIMIZAÇÃO DO TEMPO TOTAL DE FLUXO

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para a obtenção do Título de Mestre em Engenharia de Produção.

Orientador: Prof. Dr. Marcelo Seido Nagano

São Carlos – SP
2006

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

S285n Scardoelli, Lucas Yamada
Novos métodos heurísticos para a programação de operações *no-wait flow shop* com critério de minimização do tempo total de fluxo / Lucas Yamada Scardoelli. -- São Carlos, 2006.

Dissertação (Mestrado) -- Escola de Engenharia de São Carlos-Universidade de São Paulo, 2006.

Área: Engenharia de Produção.

Orientador: Prof. Dr. Marcelo Seido Nagano.

1. Programação da produção. 2. *Flow shop*. 3. *No-wait*.
4. Tempo total de fluxo. I. Título.

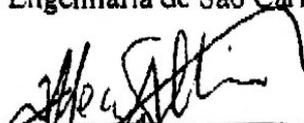
FOLHA DE JULGAMENTO

Candidato: Bacharel LUCAS YAMADA SCARDOELLI

Dissertação defendida e julgada em 31-03-2006 perante a Comissão Julgadora:


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

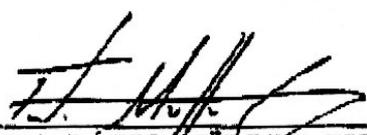
aprovado


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

APROVADO


 Prof. Dr. ALCEU SALLES CAMARGO JUNIOR
 (Faculdade de Economia, Administração e Contabilidade de Ribeirão Preto/USP)

aprovado


 Prof. Associado FÁBIO MÜLLER GUERRINI
 Coordenador do Programa de Pós-Graduação em
 Engenharia de Produção


 Profa. Titular MARIA DO CARMO CALNURI
 Presidente da Comissão de Pós-Graduação

Dedico este trabalho:
Ao pessoal da Contábil Scarty S/S;
Aos meus familiares e minha namorada.

AGRADECIMENTOS

Meus sinceros e eternos agradecimentos ao Prof. Dr. Marcelo Seido Nagano, pelas orientações, respeito e confiança. Sinto muito orgulho de ter sido orientado por pessoa tão competente e apaixonada pelo que faz. Mais uma vez, muito obrigado.

Ao meu pai, minha mãe, meu irmão, minha irmã e minha namorada, pela compreensão, motivação e incentivo.

Aos professores, funcionários e amigos da área de Engenharia de Produção da EESC-USP, obrigado pela atenção e apoio.

RESUMO

SCARDOELLI, L.Y. (2006). *Novos métodos heurísticos para a programação de operações no-wait flow shop com critério de minimização do tempo total de fluxo*. Dissertação(Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

Este trabalho apresenta o problema de programação de operações em máquinas no ambiente *no-wait flow shop* com critério de minimização do tempo total de fluxo (*total flowtime*). Aplicações para o problema *no-wait flow shop* podem ser encontradas em muitas indústrias. Por exemplo, em indústrias de metais, o metal aquecido deve seguir por uma série de operações antes de ser resfriado, com a finalidade de prevenir defeitos na sua composição. Um segundo exemplo é a indústria de plásticos que requer uma série de processos, um imediatamente após o outro, a fim de prevenir a degradação. Situações similares também são encontradas em outros processos industriais, como químicos e farmacêuticos. Neste trabalho, são propostos e avaliados novos métodos heurísticos para o problema, tão bons quanto os já existentes na literatura. Os métodos são avaliados através de um grande número de problemas de vários tamanhos. Os resultados da experimentação computacional também são analisados.

Palavras-chave: programação da produção, *flow shop*, *no-wait*, tempo total de fluxo.

ABSTRACT

SCARDOELLI, L.Y. (2006). *New heuristics methods for no-wait flow shop scheduling to minimize total flowtime*. M.Sc. Dissertation – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

This work deals with the no-wait Flow Shop scheduling problem with the objective of minimizing total flowtime. Applications of no-wait flow shop can be found in many industries. For example, in steel factories, the heated metal continuously goes through a sequence of operations before it is allowed to cool in order to prevent defects in the composition of the steel. A second example is a plastic product that requires a series of processes to immediately follow on another in order to prevent degradation. Similar situations arise in other process industries such as the chemical and pharmaceutical. In this work, we propose evaluate new heuristics methods for the problem, as well as the existing in the literature. The methods are evaluated over a large number of problems of various sizes. The results of the computational experience are also reported.

Keywords: production scheduling, flow shop, no-wait, total flowtime.

LISTA DE FIGURAS

Figura 1	Número de possíveis soluções, para o caso <i>flow shop</i> permutacional, de acordo com o número de tarefas.....	20
Figura 1.1	Fluxo da programação no problema <i>flow shop</i>	23
Figura 1.2	Exemplos de Gráficos de Gantt para o problema <i>no-wait flow shop</i>	25
Figura 1.3	Exemplo de tempos de processamento e matriz D de <i>delays</i>	26
Figura 2.1	Gráfico de Gantt para problema <i>no-wait flow shop</i>	37
Figura 3.1	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	42
Figura 3.2	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	42
Figura 3.3	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	43
Figura 3.4	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	43
Figura 3.5	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	44
Figura 3.6	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	45
Figura 3.7	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	45
Figura 3.8	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	46
Figura 3.9	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	46
Figura 3.10	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	47
Figura 3.11	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	47
Figura 3.12	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	48
Figura 3.13	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	48
Figura 3.14	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	49
Figura 3.15	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	49
Figura 3.16	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	50
Figura 3.17	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	51
Figura 3.18	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	51
Figura 3.19	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	52
Figura 3.20	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	52
Figura 3.21	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	53
Figura 3.22	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	53
Figura 3.23	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	54
Figura 3.24	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	54
Figura 3.25	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	55
Figura 3.26	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	55
Figura 3.27	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	56

Figura 3.28	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	56
Figura 3.29	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	57
Figura 3.30	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	57
Figura 3.31	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	58
Figura 3.32	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	58
Figura 3.33	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	59
Figura 3.34	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	59
Figura 3.35	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	60
Figura 3.36	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	60
Figura 3.37	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	61
Figura 3.38	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	61
Figura 3.39	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	62
Figura 3.40	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	62
Figura 3.41	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	63
Figura 3.42	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	63
Figura 3.43	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	64
Figura 3.44	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	64
Figura 3.45	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	65
Figura 4.1	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	67
Figura 4.2	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	67
Figura 4.3	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	68
Figura 4.4	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	68
Figura 4.5	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	69
Figura 4.6	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	70
Figura 4.7	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	70
Figura 4.8	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	71
Figura 4.9	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	71
Figura 4.10	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	72
Figura 4.11	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	72
Figura 4.12	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	73
Figura 4.13	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	73
Figura 4.14	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	74
Figura 4.15	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	74

Figura 4.16	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	75
Figura 4.17	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	76
Figura 4.18	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	76
Figura 4.19	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	77
Figura 4.20	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	77
Figura 4.21	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	78
Figura 4.22	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	78
Figura 4.23	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	79
Figura 4.24	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	79
Figura 4.25	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	80
Figura 4.26	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	81
Figura 4.27	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	81
Figura 4.28	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	82
Figura 4.29	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	82
Figura 4.30	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	83
Figura 4.31	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	84
Figura 4.32	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	84
Figura 4.33	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	85
Figura 4.34	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	85
Figura 4.35	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	86
Figura 4.36	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	87
Figura 4.37	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	87
Figura 4.38	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	88
Figura 4.39	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	88
Figura 4.40	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	89
Figura 4.41	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	89

Figura 4.42	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	90
Figura 4.43	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	90
Figura 4.44	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	91
Figura 4.45	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	91
Figura 5.1	Esquema de fluxograma do método proposto C1_SN.....	95
Figura 5.2	Esquema de fluxograma do método proposto C2_SN.....	97
Figura 6.1	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	100
Figura 6.2	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	100
Figura 6.3	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	101
Figura 6.4	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	101
Figura 6.5	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	102
Figura 6.6	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	103
Figura 6.7	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	103
Figura 6.8	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	104
Figura 6.9	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	104
Figura 6.10	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	105
Figura 6.11	Porcentagem de Sucesso vs. Número de Tarefas, para 5 máquinas...	106
Figura 6.12	Porcentagem de Sucesso vs. Número de Tarefas, para 10 máquinas.	106
Figura 6.13	Porcentagem de Sucesso vs. Número de Tarefas, para 15 máquinas.	107
Figura 6.14	Porcentagem de Sucesso vs. Número de Tarefas, para 20 máquinas.	107
Figura 6.15	Porcentagem de Sucesso vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	108
Figura 6.16	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	109
Figura 6.17	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	109
Figura 6.18	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	110
Figura 6.19	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	110
Figura 6.20	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	111
Figura 6.21	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	112
Figura 6.22	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	112
Figura 6.23	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 15 máquinas.....	113
Figura 6.24	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	113
Figura 6.25	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	114
Figura 6.26	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 5 máquinas.....	115
Figura 6.27	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 10 máquinas.....	115
Figura 6.28	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para	

	15 máquinas.....	116
Figura 6.29	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para 20 máquinas.....	116
Figura 6.30	Porcentagem de Desvio Médio Relativo vs. Número de Tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	117
Figura 6.31	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	118
Figura 6.32	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	118
Figura 6.33	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	119
Figura 6.34	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	119
Figura 6.35	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	120
Figura 6.36	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	120
Figura 6.37	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	121
Figura 6.38	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	121
Figura 6.39	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	122
Figura 6.40	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	122
Figura 6.41	Tempo Médio de Computação vs. Número de tarefas, para 5 máquinas.....	123
Figura 6.42	Tempo Médio de Computação vs. Número de tarefas, para 10 máquinas.....	123
Figura 6.43	Tempo Médio de Computação vs. Número de tarefas, para 15 máquinas.....	124
Figura 6.44	Tempo Médio de Computação vs. Número de tarefas, para 20 máquinas.....	124
Figura 6.45	Tempo Médio de Computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).....	125

LISTA DE TABELAS

Tabela 1	Porcentagem de Sucesso para 5 máquinas.....	133
Tabela 2	Porcentagem de Sucesso para 10 máquinas.....	133
Tabela 3	Porcentagem de Sucesso para 15 máquinas.....	133
Tabela 4	Porcentagem de Sucesso para 20 máquinas.....	133
Tabela 5	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	133
Tabela 6	Porcentagem de Sucesso para 5 máquinas.....	134
Tabela 7	Porcentagem de Sucesso para 10 máquinas.....	134
Tabela 8	Porcentagem de Sucesso para 15 máquinas.....	134
Tabela 9	Porcentagem de Sucesso para 20 máquinas.....	134
Tabela 10	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	135
Tabela 11	Porcentagem de Sucesso para 5 máquinas.....	135
Tabela 12	Porcentagem de Sucesso para 10 máquinas.....	135
Tabela 13	Porcentagem de Sucesso para 15 máquinas.....	135
Tabela 14	Porcentagem de Sucesso para 20 máquinas.....	135
Tabela 15	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	135
Tabela 16	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	136
Tabela 17	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	136
Tabela 18	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	136
Tabela 19	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	136
Tabela 20	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	136
Tabela 21	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	136
Tabela 22	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	137
Tabela 23	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	137
Tabela 24	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	137
Tabela 25	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	137
Tabela 26	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	137
Tabela 27	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	138
Tabela 28	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	138
Tabela 29	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	138
Tabela 30	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	138
Tabela 31	Tempo Médio de Computação (s) para 5 máquinas.....	138
Tabela 32	Tempo Médio de Computação (s) para 10 máquinas.....	138
Tabela 33	Tempo Médio de Computação (s) para 15 máquinas.....	139
Tabela 34	Tempo Médio de Computação (s) para 20 máquinas.....	139
Tabela 35	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	139
Tabela 36	Tempo Médio de Computação (s) para 5 máquinas.....	139
Tabela 37	Tempo Médio de Computação (s) para 10 máquinas.....	139
Tabela 38	Tempo Médio de Computação (s) para 15 máquinas.....	140
Tabela 39	Tempo Médio de Computação (s) para 20 máquinas.....	140
Tabela 40	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	140
Tabela 41	Tempo Médio de Computação (s) para 5 máquinas.....	140
Tabela 42	Tempo Médio de Computação (s) para 10 máquinas.....	140
Tabela 43	Tempo Médio de Computação (s) para 15 máquinas.....	141
Tabela 44	Tempo Médio de Computação (s) para 20 máquinas.....	141
Tabela 45	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	141
Tabela 46	Porcentagem de Sucesso para 5 máquinas.....	142
Tabela 47	Porcentagem de Sucesso para 10 máquinas.....	142

Tabela 48	Porcentagem de Sucesso para 15 máquinas.....	142
Tabela 49	Porcentagem de Sucesso para 20 máquinas.....	142
Tabela 50	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	142
Tabela 51	Porcentagem de Sucesso para 5 máquinas.....	143
Tabela 52	Porcentagem de Sucesso para 10 máquinas.....	143
Tabela 53	Porcentagem de Sucesso para 15 máquinas.....	143
Tabela 54	Porcentagem de Sucesso para 20 máquinas.....	143
Tabela 55	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	144
Tabela 56	Porcentagem de Sucesso para 5 máquinas.....	144
Tabela 57	Porcentagem de Sucesso para 10 máquinas.....	144
Tabela 58	Porcentagem de Sucesso para 15 máquinas.....	144
Tabela 59	Porcentagem de Sucesso para 20 máquinas.....	144
Tabela 60	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	144
Tabela 61	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	145
Tabela 62	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	145
Tabela 63	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	145
Tabela 64	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	145
Tabela 65	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	145
Tabela 66	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	145
Tabela 67	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	146
Tabela 68	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	146
Tabela 69	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	146
Tabela 70	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	146
Tabela 71	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	146
Tabela 72	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	147
Tabela 73	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	147
Tabela 74	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	147
Tabela 75	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	147
Tabela 76	Tempo Médio de Computação (s) para 5 máquinas.....	147
Tabela 77	Tempo Médio de Computação (s) para 10 máquinas.....	147
Tabela 78	Tempo Médio de Computação (s) para 15 máquinas.....	148
Tabela 79	Tempo Médio de Computação (s) para 20 máquinas.....	148
Tabela 80	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	148
Tabela 81	Tempo Médio de Computação (s) para 5 máquinas.....	148
Tabela 82	Tempo Médio de Computação (s) para 10 máquinas.....	148
Tabela 83	Tempo Médio de Computação (s) para 15 máquinas.....	149
Tabela 84	Tempo Médio de Computação (s) para 20 máquinas.....	149
Tabela 85	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	149
Tabela 86	Tempo Médio de Computação (s) para 5 máquinas.....	149
Tabela 87	Tempo Médio de Computação (s) para 10 máquinas.....	149
Tabela 88	Tempo Médio de Computação (s) para 15 máquinas.....	150
Tabela 89	Tempo Médio de Computação (s) para 20 máquinas.....	150
Tabela 90	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	150
Tabela 91	Porcentagem de Sucesso para 5 máquinas.....	151
Tabela 92	Porcentagem de Sucesso para 10 máquinas.....	151
Tabela 93	Porcentagem de Sucesso para 15 máquinas.....	151
Tabela 94	Porcentagem de Sucesso para 20 máquinas.....	151
Tabela 95	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	151
Tabela 96	Porcentagem de Sucesso para 5 máquinas.....	152
Tabela 97	Porcentagem de Sucesso para 10 máquinas.....	152
Tabela 98	Porcentagem de Sucesso para 15 máquinas.....	152

Tabela 99	Porcentagem de Sucesso para 20 máquinas.....	152
Tabela 100	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	153
Tabela 101	Porcentagem de Sucesso para 5 máquinas.....	153
Tabela 102	Porcentagem de Sucesso para 10 máquinas.....	153
Tabela 103	Porcentagem de Sucesso para 15 máquinas.....	153
Tabela 104	Porcentagem de Sucesso para 20 máquinas.....	153
Tabela 105	Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas.....	153
Tabela 106	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	154
Tabela 107	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	154
Tabela 108	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	154
Tabela 109	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	154
Tabela 110	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	154
Tabela 111	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	154
Tabela 112	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	155
Tabela 113	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	155
Tabela 114	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	155
Tabela 115	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	155
Tabela 116	Porcentagem de Desvio Médio Relativo para 5 máquinas.....	155
Tabela 117	Porcentagem de Desvio Médio Relativo para 10 máquinas.....	156
Tabela 118	Porcentagem de Desvio Médio Relativo para 15 máquinas.....	156
Tabela 119	Porcentagem de Desvio Médio Relativo para 20 máquinas.....	156
Tabela 120	Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas.....	156
Tabela 121	Tempo Médio de Computação (s) para 5 máquinas.....	156
Tabela 122	Tempo Médio de Computação (s) para 10 máquinas.....	156
Tabela 123	Tempo Médio de Computação (s) para 15 máquinas.....	157
Tabela 124	Tempo Médio de Computação (s) para 20 máquinas.....	157
Tabela 125	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	157
Tabela 126	Tempo Médio de Computação (s) para 5 máquinas.....	157
Tabela 127	Tempo Médio de Computação (s) para 10 máquinas.....	157
Tabela 128	Tempo Médio de Computação (s) para 15 máquinas.....	158
Tabela 129	Tempo Médio de Computação (s) para 20 máquinas.....	158
Tabela 130	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	158
Tabela 131	Tempo Médio de Computação (s) para 5 máquinas.....	158
Tabela 132	Tempo Médio de Computação (s) para 10 máquinas.....	158
Tabela 133	Tempo Médio de Computação (s) para 15 máquinas.....	159
Tabela 134	Tempo Médio de Computação (s) para 20 máquinas.....	159
Tabela 135	Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas.....	159

SUMÁRIO

INTRODUÇÃO	18
1. PROBLEMA DE PROGRAMAÇÃO DE OPERAÇÕES EM AMBIENTE NO-WAIT <i>FLOW SHOP</i>	23
1.1 Principais características do <i>Flow Shop</i> Permutacional.....	23
1.2 Problema <i>No-Wait Flow Shop</i>	24
1.2.1 Problema <i>NWFS</i> com critério de minimização da duração total da programação (<i>makespan</i>).....	26
1.2.2 Problema <i>NWFS</i> com critério de minimização do <i>flowtime</i>	27
1.3 Investigação e Análise dos Melhores Métodos e suas Modificações.....	29
1.3.1 Rajendran e Chaudhuri (1990).....	29
1.3.2 Bertolissi (2000).....	31
1.3.3 Aldowaisan e Allahverdi (2004).....	32
1.3.4 Algoritmos Modificados.....	34
2. PROPOSIÇÃO DE UM MÉTODO HEURÍSTICO CONSTRUTIVO DE DUAS FASES	36
2.1 Fase 1: Ordenação inicial das tarefas.....	36
2.2 Fase 2: Construção da seqüência final das tarefas.....	38
3. EXPERIMENTAÇÃO COMPUTACIONAL - PARTE A	40
3.1 Planejamento dos Experimentos.....	40
3.2 Obtenção de Dados.....	40
3.3 Método de Análise.....	41
3.4 Análise dos Resultados	41
3.4.1 Análise da Porcentagem de Sucesso.....	42
3.4.2 Análise da Porcentagem de Desvio Médio Relativo.....	50
3.4.3 Tempo Médio de Computação.....	58
4. EXPERIMENTAÇÃO COMPUTACIONAL - PARTE B	66
4.1 Análise dos Resultados	66
4.1.1 Análise da Porcentagem de Sucesso.....	66
4.1.2 Análise da Porcentagem de Desvio Médio Relativo.....	75
4.1.3 Tempo Médio de Computação.....	83
5. PROPOSIÇÃO DE DOIS MÉTODOS HEURÍSTICOS CONSTRUTIVOS DE FASE ÚNICA	93
5.1 Método Heurístico C1_SN.....	93

5.2 Método Heurístico C2_SN.....	95
6. EXPERIMENTAÇÃO COMPUTACIONAL – PARTE FINAL	99
6.1 Análise dos Resultados	99
6.1.1 Análise da Porcentagem de Sucesso.....	99
6.1.2 Análise da Porcentagem de Desvio Médio Relativo.....	108
6.1.3 Tempo Médio de Computação.....	117
7. CONSIDERAÇÕES FINAIS	126
REFERÊNCIAS BIBLIOGRÁFICAS	128
ANEXO A – Tabelas Referentes às Experimentações Computacionais	132

INTRODUÇÃO

A análise da natureza global da administração da produção, feita por Slack et al. (1999), enaltece que a produção é o foco central para as organizações porque é responsável por produzir bens e serviços que são a razão da sua existência.

A administração da produção, na maioria das situações, é conduzida de forma sistemática. Segundo Johnson e Montgomery (1974), entre os sistemas de produção, os intermitentes são aqueles que comportam o processamento do mesmo produto ou de uma grande variedade de produtos, em lotes de diferentes tamanhos.

Os sistemas de produção intermitentes são considerados mais flexíveis, pois apresentam diferentes tarefas a serem executadas em uma ou mais máquinas e, naturalmente, exigem do administrador maior perspicácia no planejamento e controle da produção.

Para o administrador planejar e controlar a produção são necessárias diversas atividades, entre elas, pode-se considerar como uma das principais, a programação da produção, que visa alocar materiais, máquinas, pessoas e estoques de acordo com o planejado, determinando quando e onde cada tarefa ou operação produtiva deve ser realizada.

Uma das aplicações da programação da produção é na tentativa de resolução dos problemas de programação da produção de operações em máquinas, que referem-se principalmente às decisões de curto prazo no ambiente produtivo. Maiores detalhes sobre estes problemas serão abordados de forma introdutória na seqüência, pois o tema principal desta dissertação é um tipo específico de problema que será apresentado no capítulo 1.

- **Problemas de Programação de Operações em Máquinas**

De acordo com Pinedo (1995), os problemas de programação de operações em máquinas podem ser descritos com a notação de três parâmetros: $\alpha / \beta / \gamma$. O parâmetro α descreve o ambiente das máquinas, o parâmetro β contém as características do processamento e o parâmetro γ possui a função objetivo, ou seja, o critério de avaliação da

programação.

Segundo Moccellin (1994), a programação refere-se à ordenação de tarefas a serem executadas, em uma ou diversas máquinas, considerando-se uma base de tempo, ou seja, determinando-se, principalmente, as datas de início e fim de cada tarefa.

As tarefas são conhecidas, determinadas e devem ser executadas. Cada tarefa corresponde a um dado conjunto de operações que tem uma seqüência a ser seguida para a execução completa.

Porém, a maioria dos problemas de programação da produção que as organizações enfrentam é muito complexa e de difícil solução. Além disso, cada ambiente de fabricação possui suas particularidades, apresentando diferentes tipos de problemas, que podem ser definidos e classificados da seguinte forma:

- **JOB SHOP:** cada tarefa tem sua própria ordem de processamento nas máquinas;
- **FLOW SHOP:** todas as tarefas têm o mesmo fluxo de processamento nas máquinas;
- **OPEN SHOP:** não há fluxo definido (específico) para as tarefas serem processadas nas máquinas;
- **FLOW SHOP PERMUTACIONAL:** trata-se de *flow shop* no qual a ordem de processamento das tarefas deve ser a mesma em todas as máquinas;
- **MÁQUINA ÚNICA:** existe apenas uma máquina a ser utilizada;
- **MÁQUINAS PARALELAS:** são disponíveis mais de uma máquina, geralmente idênticas, para as mesmas operações;
- **JOB SHOP COM MÚLTIPLAS MÁQUINAS:** *job shop* no qual em cada estágio de produção existe um conjunto de máquinas paralelas;
- **FLOW SHOP COM MÚLTIPLAS MÁQUINAS:** *flow shop* no qual em cada estágio de produção existe um conjunto de máquinas paralelas.

Dos problemas acima, teoricamente, o caso mais simples é o problema de ordenação de n tarefas em uma única máquina. O número de soluções possíveis neste caso é $n!$ (n fatorial).

No problema *flow shop* permutacional (tipo de problema abordado nesta dissertação), a seqüência das tarefas deve ser a mesma em todas as máquinas. Porém, verifica-se que mesmo para problemas relativamente pequenos, o número de soluções possíveis é extremamente elevado.

Número de tarefas	Número de Soluções Possíveis
5	120
6	720
7	5.040
8	40.320
9	362.880
10	3.628.800

Figura 1. Número de possíveis soluções, para o caso *flow shop* permutacional, de acordo com o número de tarefas.

Uma forma de buscar a solução ótima seria gerar todas as soluções possíveis e identificar a melhor. Porém, na maioria dos casos, isto é inviável. Observando por exemplo, na Figura 1: no caso de 10 tarefas a serem seqüenciadas, se o computador demorar 0,1 segundo para avaliar cada solução, levaria 4,2 dias de processamento ininterruptos para encontrar todas as soluções possíveis.

Existem problemas que são relativamente simples de resolver através de programação linear para se obter a solução ótima. Entretanto, a grande maioria dos problemas de programação da produção é intrinsecamente muito difícil de ser resolvida. Estes problemas podem ser classificados como *NP-hard* e sua complexidade cresce exponencialmente com relação à dimensão do problema.

Segundo Pinedo (1995), os problemas podem ser classificados em dois grupos principais:

- Problemas simples que, mesmo em grandes dimensões garantem a solução ótima em tempo polinomial;
- Problemas complexos (mais comuns) conhecidos como *NP-hard*. Para estes problemas não são conhecidos métodos que garantem encontrar a solução ótima em tempo aceitável.

Em relação a este último grupo, os métodos heurísticos têm sido amplamente estudados e aplicados, para obtenção de soluções viáveis e, às vezes, próximas das soluções ótimas.

Os métodos heurísticos podem ser classificados de diversas maneiras. Uma delas classifica-os em Construtivos ou Melhorativos, dependendo da forma de obtenção da solução. Os métodos construtivos se caracterizam pelo fato de gerarem apenas uma solução, a qual será a solução final do problema, por exemplo, o método NEH de Nawaz et al. (1983), o método N&M de Nagano e Moccellini (2002), o método proposto em Bertolissi (2000) e o proposto por Rajendran e Chaudhuri (1990). No caso dos métodos melhorativos, obtém-se uma solução inicial e posteriormente através de algum procedimento iterativo (geralmente envolvendo trocas de posições das tarefas na seqüência) busca-se obter uma seqüência das tarefas melhor que a atual quanto à medida de desempenho adotada, por exemplo, os métodos propostos por Widmer e Hertz (1989) e Osman e Potts (1989).

Segundo Moccellini (1994), os problemas de seqüenciamento de tarefas também podem ser classificados como estáticos ou dinâmicos. No caso estático, o conjunto de tarefas a serem executadas é conhecido (determinado) e os problemas podem ser solucionados através de métodos exatos ou heurísticos. No caso dinâmico, as tarefas chegam à fábrica de forma intermitente, em datas diferentes que somente podem ser previstas com bases probabilísticas. Os principais métodos para solução dos problemas são: teoria dos processos estocásticos, sistemas de filas e simulação.

Seja *flow shop* ou *job shop*, estático ou dinâmico, todos os problemas também devem possuir um critério de desempenho a ser considerado. Estes critérios referem-se ao objeto da análise, à variável a ser medida.

Segundo Baker (1974), as principais medidas de desempenho utilizadas são:

- Minimização da duração total da programação (*makespan*), indicado por M , F_{\max} ou C_{\max} ;
- Minimização do tempo total de fluxo (*total flowtime*) ou tempo médio de fluxo (*mean flowtime*), indicado por \bar{F} ;
- Minimização do atraso médio, indicado por \bar{T} ou Minimização do atraso máximo (*tardiness*), indicado por T_{\max} .

Com base nesta breve introdução acerca dos problemas de programação de operações em máquinas, no próximo tópico são delineados os principais objetivos deste trabalho, principalmente com relação a um tipo específico de problema (abordado nesta dissertação), mais conhecido como *no-wait flow shop* (NWFS).

• Objetivos e Estrutura do Trabalho

Neste contexto, esta dissertação terá como objetivo explorar um tipo específico de problema, existente em algumas organizações, devido às suas particularidades.

Algumas indústrias, entre elas as que produzem produtos plásticos, químicos, farmacêuticos, metais e alimentos, podem apresentar um problema bastante específico: quando uma tarefa é iniciada, ela não pode ser interrompida durante seu processamento nas máquinas, nem mesmo entre as máquinas.

Nas indústrias de aço, por exemplo, no caso dos cilindros ou rodas de aço, o metal aquecido precisa ser continuamente processado (sem interrupções) por diversas operações antes de ser resfriado, com a finalidade de evitar defeitos na composição do material. Nas indústrias de produtos alimentares, a operação de enlatamento deve ser executada logo após o cozimento para garantir a qualidade do produto final.

Assim, esta pesquisa tem como objetivo principal investigar o problema de programação de operações, considerando a impossibilidade de interrupções para as tarefas iniciadas (*no-wait flow shop*) propondo métodos heurísticos para solução do problema com o objetivo de minimização do tempo total de fluxo das tarefas (*total flowtime*).

Segundo a notação adotada por Pinedo (1995), este problema pode ser denotado como $F_m/\text{no-wait}/\bullet C_j$.

Adjacente a este objetivo central acima exposto, são objetivos também do trabalho: revisão bibliográfica acerca do tema *no-wait flow shop* com critério de minimização do tempo total de fluxo e a implementação dos principais algoritmos encontrados na literatura, para fins de comparação.

Com base nos objetivos anteriormente descritos, a estrutura da dissertação será a seguinte:

- Capítulo 1 – caracterização do problema de programação de operações *no-wait flow shop* e revisão bibliográfica acerca do tema em questão;
- Capítulo 2 – proposição de um método heurístico construtivo de duas fases;
- Capítulo 3 – experimentação computacional (Parte A), que avalia os principais métodos heurísticos construtivos de duas fases existentes na literatura, comparando-os entre si e com o método proposto no capítulo 2;
- Capítulo 4 – experimentação computacional (Parte B), que avalia os principais métodos heurísticos construtivos de três fases existentes na literatura e os modificados, comparando-os entre si;
- Capítulo 5 – proposição de dois métodos heurísticos construtivos de fase única;
- Capítulo 6 – experimentação computacional final, que compara os melhores métodos dos capítulos 3 e 4 com os métodos propostos no capítulo 5;
- Capítulo 7 - conclusão do trabalho considerando os resultados alcançados e as possibilidades para pesquisas futuras.

CAPÍTULO 1

Problema de Programação de Operações em Ambiente *No-Wait Flow Shop*

Neste capítulo serão descritas as principais características do *Flow Shop* Permutacional, bem como do problema *No-Wait Flow Shop* e dos principais métodos existentes para sua solução.

1.1 Principais Características do *Flow Shop* Permutacional

Aldowaisan e Allahverdi (2004), Framinan e Liesten (2003), Nagano e Moccellini (2002), e outros, relatam que o problema de programação de operações *flow shop* é aquele em que cada tarefa de um conjunto de n tarefas tem que ser processada, na mesma ordem, nas m máquinas existentes no ambiente.

Quando a seqüência de processamento das tarefas é a mesma em todas as máquinas, tem-se o ambiente de programação *flow shop* permutacional.

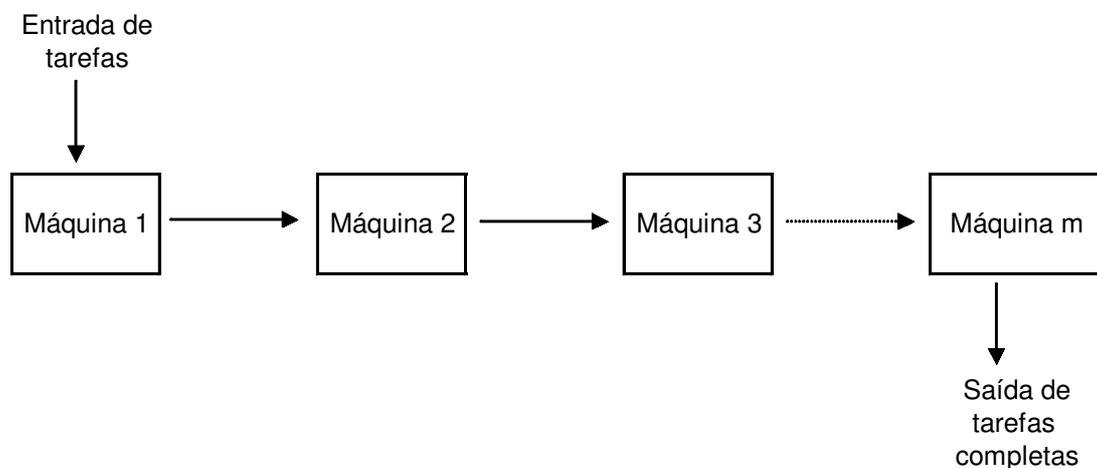


Figura 1.1. Fluxo da programação no problema *flow shop*.

Reisman et al. (1997), numa revisão bibliográfica, identificaram cerca de cento e setenta trabalhos envolvendo o tema *flow shop*, desde 1952 até 1994. Entre outras conclusões, verificaram principalmente que o número de artigos vem crescendo exponencialmente, pois trata-se de uma área de estudo abrangente, que desperta o interesse de muitos pesquisadores.

Segundo Moccellini e Nagano (1998) este problema geralmente é modelado com as seguintes considerações:

- Os tempos de processamento das tarefas nas máquinas são conhecidos, fixos e assumem o valor zero quando não é processado em determinada máquina;
- Os tempos de *setup* estão incluídos no tempo de processamento e independem da posição da tarefa na ordenação das tarefas;
- Toda tarefa é processada em somente uma máquina por vez e toda máquina processa uma tarefa por vez;
- As operações das tarefas nas máquinas não podem ser interrompidas.

Na maioria dos estudos, as seguintes notações são usadas no problema *flow shop* permutacional:

- Conjunto J de tarefas $J = \{J_1, J_2, J_3, \dots, J_i, \dots, J_n\}$
- Conjunto J_i de operações $J_i = \{op_{i1}, op_{i2}, op_{i3}, \dots, op_{ik}, \dots, op_{im}\}$
- Conjunto M de máquinas $M = \{m_1, m_2, m_3, \dots, m_k, \dots, m_m\}$

Assim, as tarefas são numeradas por $J_1, J_2, J_3, \dots, J_i, \dots, J_n$, e as máquinas são numeradas por $m_1, m_2, m_3, \dots, m_k, \dots, m_m$.

Cada tarefa, quando executada em cada máquina, consome um tempo de processamento, que é denotado pela variável t_{ik} , onde $1 \leq i \leq n$ e $1 \leq k \leq m$, que expressa o tempo de processamento da tarefa i na máquina k .

Através destas variáveis é possível analisar a seguinte expressão, para obtenção do tempo total de fluxo de uma seqüência de tarefas estabelecida:

$$F = \sum_{i=1}^n F_i \quad (1)$$

F = tempo total de fluxo;

F_i = *flowtime* da tarefa J_i

Trata-se do critério de desempenho utilizado neste trabalho, que é a soma do tempo total de fluxo para as n tarefas programadas.

1.2 Problema *No-Wait Flow Shop*

Chen et al. (1996) relatam que muitos algoritmos propostos para solução de problemas assumem que existe capacidade de estocagem intermediária infinita, decorrente do fato de que, depois de processada uma tarefa, existe a possibilidade da máquina subsequente estar ocupada por outra tarefa. Porém, algumas vezes, esta situação não é aceitável.

Allahverdi e Aldowaisan (2002) explicam que o problema pode ser encontrado em muitas indústrias. Entre elas, nas indústrias de plástico, em que o produto plástico requer uma série de processos seguidos a fim de prevenir a sua degradação.

A situação acima, conhecida como *no-wait flow shop*, *continuous flow shop*, *flow shop with no intermediate storage* (REDDI e RAMAMOORTHY¹, 1972 *apud* CHEN et al., 1996, p.919), *flow shop with no intermediate queues* (DEMAN e BAKER, 1974), é definida como um problema *flow shop* de m máquinas no qual, quando se inicia o processamento de uma tarefa, esta deve ser processada por todas as máquinas sem nenhuma interrupção nas máquinas, nem entre as máquinas.

Allahverdi e Aldowaisan (2002) complementam que, quando necessário, o início do processamento da tarefa pode ser retardado (Figura 1.2), mas alerta que devem sempre coincidir, o início do processamento de uma tarefa em uma máquina e, o fim do processamento da mesma tarefa na máquina anterior.

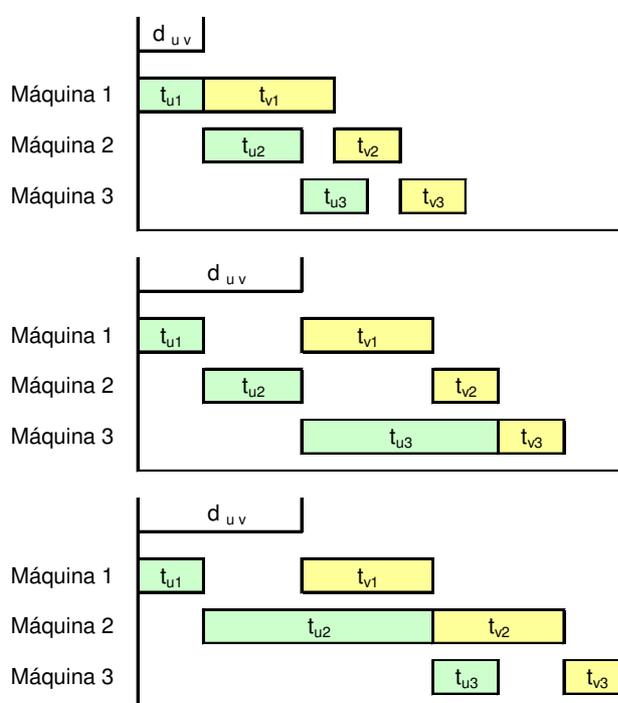


Figura 1.2. Exemplos de Gráficos de Gantt para o problema *no-wait flow shop*

¹ REDDI, S.S.; RAMAMOORTHY, C.V. (1972). On the flow-shop sequencing problem with no wait in process. *Op. Res.* q.23, p.323-331 *apud* CHEN, C.; NEPPALLI, R.V.; ALJABER N. (1996). Genetic Algorithms Applied to the Continuous Flow Shop Problem. *Computers Indl. Engineers*, v.30, n.4, p.919

Ao analisar a Figura 1.2, e também, segundo Bertolissi (2000) e Fink e Voã (2003) verifica-se que o problema *no-wait flow shop* geralmente determina um *delay* d_{uv} ($1 \bullet u \bullet n$, $1 \bullet v \bullet n$, $u \bullet v$), na primeira máquina, entre o início da tarefa u e o início da tarefa v , quando a tarefa v é processada diretamente após a tarefa u . O *delay* d_{uv} pode ser calculado através da seguinte expressão, considerando $2 \leq k \leq m$:

$$d_{uv} = t_{u,1} + \max_k \left(\sum_{p=2}^k t_{up} - \sum_{p=1}^{k-1} t_{vp}, 0 \right) \quad (2)$$

A partir do conjunto de dados dos tempos de processamentos, os resultados podem ser obtidos em uma matriz assimétrica de *delay*, denominada matriz D .

Um exemplo de três tarefas e três máquinas de Fink e Voã (2003) possibilita a visualização dos tempos de processamento e a matriz D , obtida através da expressão (2):

Matriz D			
	J ₁	J ₂	J ₃
Máquina 1	1	1	4
Máquina 2	3	2	1
Máquina 3	3	2	4

	J ₁	J ₂	J ₃
J ₁	-	4	2
J ₂	2	-	1
J ₃	5	6	-

Figura 1.3. Exemplo de tempos de processamento e matriz D de *delays*

A matriz D é importante e é utilizada pelos últimos métodos heurísticos propostos para solução do problema NWFS com critério de minimização do tempo total de fluxo (*flowtime*).

1.2.1 Problema NWFS com critério de minimização da duração total da programação (*makespan*)

Bonney e Gundry (1976) foram os primeiros a estabelecer método heurístico, através da adaptação de algoritmos criados para problemas de máquina única e utilizando-os para solução do problema com critério de minimização do *makespan*.

Na década de 80, Papadimitriou e Kanellakis (1980) e Adiri e Pohoryles (1982) também deram continuidade na busca por melhores soluções. Papadimitriou e Kanellakis (1980) apresentaram um caso particular para solução de problemas de pequeno porte, de 3 e 4 máquinas, enquanto Adiri e Pohoryles (1982) apresentaram pesquisas para o ambiente *no-idle* (o trabalho nas máquinas deve ser contínuo) e *no-wait* para obtenção de soluções ótimas de problemas com 2 máquinas.

Porém, na década seguinte, Gangadharan e Rajendran (1993) desenvolveram métodos heurísticos construtivos que comprovaram ser melhores que as heurísticas

desenvolvidas anteriormente. Tratam-se de dois algoritmos de duas fases: na primeira é obtida uma ordenação inicial e na segunda é executada construção da seqüência final através da inserção de tarefas.

Rajendran (1994) desenvolveu um método heurístico semelhante aos propostos por Gangadharan e Rajendran (1993). Na primeira fase também é obtida a ordenação inicial. Na segunda fase é aplicado um método de inserção de tarefas em seqüências parciais, obedecendo-se a ordenação inicial obtida na fase anterior. Atualmente, ainda é o algoritmo heurístico que apresenta o melhor desempenho para solução do problema NWFS com critério de minimização do *makespan*.

Através da utilização de metaheurísticas, Aldowaisan e Allahverdi (2003) desenvolveram três heurísticas denominadas SA, SA-1 e SA-2, baseadas no método *Simulated Annealing* e outras três heurísticas denominadas GA, GA-1 e GA-2 baseados no método Algoritmo Genético, que, por sua vez, mostraram-se superiores aos métodos propostos anteriormente.

1.2.2 Problema NWFS com critério de minimização do *flowtime*

Como já foi mencionado, o objetivo considerado neste trabalho é a minimização do tempo total de fluxo.

De acordo com Framinan e Leisten (2003), o critério do tempo total de fluxo, é conhecido por conduzir à estabilidade ou à constância na utilização dos recursos, rápido processamento das tarefas (início ao fim sem interrupções) e minimização do estoque em processamento.

Para o problema *no-wait flow shop*, o tempo total de fluxo pode ser calculado através da seguinte expressão:

$$F = \sum_{i=2}^n (n+1-i)d_{[i-1][i]} + \sum_{i=1}^n \sum_{k=1}^m t_{ik} \quad (3)$$

Resumidamente, esta expressão obtém a soma dos tempos de processamento das tarefas em todas as máquinas e a soma dos *delays* entre as tarefas seqüenciadas, e os termos $d_{[i-1][i]}$ indicam determinada posição na seqüência.

Demian e Baker (1974) foram alguns dos pioneiros a estudar o problema de *no-wait flow shop*, com o critério de minimização do tempo total de fluxo. Eles propuseram um algoritmo *branch and bound* para estabelecer todas as seqüências parciais, considerando a utilização de limitantes inferiores. Os autores concluíram que os resultados alcançados foram satisfatórios e que o problema podia ser solucionado tão rapidamente quanto os problemas tradicionais que utilizavam como critério a minimização do *makespan*.

Rajendran e Chaudhuri (1990) apresentaram dois algoritmos heurísticos construtivos, compostos de duas fases: uma primeira fase de ordenação inicial das tarefas e outra de inserção de tarefas para a obtenção das melhores seqüências parciais, a fim de estabelecer a seqüência final das tarefas. A experimentação computacional provou que as soluções obtidas eram melhores que os métodos anteriormente propostos por Bonney e Gundry (1976) e King e Spachis (1980), cujo critério de desempenho considerado foi o *makespan*.

Chen et al. (1996) desenvolveram uma heurística, baseada no algoritmo genético, e, através da sua parametrização, conseguiram melhorar as soluções obtidas pelo método construtivo proposto por Rajendran e Chaudhuri (1990).

Bertolissi (1999) apresentou um método heurístico construtivo de apenas uma fase, na qual são definidas as parcelas das somas dos tempos de fluxo de duas tarefas adjacentes J_i e J_j , a partir do início de J_i (ou seja, no intervalo de tempo entre o início de J_i e o término de J_j) e em seguida é obtida uma ordenação das tarefas. Os experimentos computacionais mostraram que o método proposto não obtém soluções melhores que os métodos de Rajendran e Chaudhuri (1990).

Bertolissi (2000) apresenta um novo método heurístico construtivo composto de duas fases. Na primeira fase, similar a Bertolissi (1999), é definida uma ordenação inicial, de acordo com as parcelas das somas dos tempos de fluxo de duas tarefas adjacentes J_i e J_j , a partir do início de J_i (ou seja, no intervalo de tempo entre o início de J_i e o término de J_j). Na segunda fase, é utilizado o mesmo procedimento de inserção de tarefas de Rajendran e Chaudhuri (1990). Os resultados dos experimentos computacionais permitem concluir que o método heurístico construtivo proposto obtém melhores soluções quando comparado aos métodos heurísticos construtivos existentes.

Fink e Voça (2003) utilizaram os procedimentos metaheurísticos de Busca na Vizinhança, Busca Tabu e *Simulated Annealing* e concluíram que a efetividade dos procedimentos metaheurísticos utilizados na solução do problema depende da qualidade da solução desejada e do tempo computacional disponível. Ou seja, os autores não contemplam um método como o melhor entre todos, mas afirmam que, dependendo das características do problema, existe um melhor método.

Aldowaisan e Allahverdi (2004) propuseram novos métodos heurísticos compostos de três fases. Segundo os experimentos e a conclusão dos próprios pesquisadores, o algoritmo PH1(p) é o que apresenta melhores resultados. Na primeira fase é desenvolvida a ordenação inicial através de regras de prioridade estabelecidas. Na segunda fase é obtida uma solução inicial através do método de inserção NEH, proposto por Nawaz et al. (1983).

Na última fase é aplicado o método de melhoria utilizando a permutação de tarefas. Trata-se de um método heurístico composto, cuja experimentação computacional comprova obter soluções melhores que os métodos heurísticos de Rajendran e Chaudhuri (1990) e o método metaheurístico de Chen et al. (1996).

Com base nos fatos levantados na literatura acerca do assunto, pode-se classificar os melhores métodos heurísticos construtivos e compostos existentes, da seguinte forma:

Métodos Heurísticos Construtivos de Duas Fases: **Raj1** e **Raj2** de Rajendran e Chaudhuri (1990) e **Bert** de Bertolissi (2000);

Método Heurístico Composto de Três Fases: **PH1(p)** de Aldowaisan e Allahverdi (2004).

Estes métodos serão investigados e analisados com maiores detalhes na próxima seção. Além disso, para fins de comparação com os métodos propostos, serão sugeridas algumas modificações nos algoritmos Raj2 de Rajendran e Chaudhuri e PH1(p) de Aldowaisan e Allahverdi (2004).

1.3 Investigação e Análise dos Melhores Métodos e suas Modificações

1.3.1 Rajendran e Chaudhuri (1990)

Este é o trabalho mais referenciado na literatura e apresenta dois algoritmos para solução do problema em questão. Ambos os algoritmos desenvolvidos são simples e de fácil implementação, principalmente pelo fato de considerarem os índices T_i e T_i' , que são base para a regra da ordenação inicial.

Os algoritmos são compostos de quatro “passos”. No primeiro é formada a ordenação inicial, levando-se em conta os índices T_i e T_i' . O cálculo do valor de T_i é obtido através da soma dos produtos do tempo de processamento da tarefa i e $(m-k+1)$. Já T_i' é simplesmente obtido através da soma dos tempos de processamento da tarefa i nas máquinas.

Nos “passos” 2 e 3 é utilizado um método de inserção de tarefas nas seqüências parciais, de forma a prevalecer, a cada tarefa inserida, a seqüência que apresentar menor tempo total de fluxo. No último “passo” consta a condição de parada, ou seja, quando todas as tarefas da ordenação inicial estiverem inseridas na programação, o algoritmo é encerrado, senão o processo volta ao passo 3.

A seguir são demonstrados os algoritmos “passo a passo”.

- **Raj1 (algoritmo 1)**

Passo 1: Formar um vetor de tarefas, ordenando-as de forma crescente em relação ao valor de T_i :

$$T_i = \sum_{k=1}^m (m - k + 1)t_{ik} \quad (4)$$

Qualquer empate deve ser resolvido através da escolha da tarefa com o menor valor de T_i' .

$$T_i' = \sum_{k=1}^m t_{ik} \quad (5)$$

Passo 2: Remover a primeira tarefa do vetor e denominá-la como sendo a tarefa “principal”, formando a seqüência parcial \acute{a} com $n' = 1$. Atualizar os dados do vetor de tarefas.

Passo 3: Remover a primeira tarefa do vetor atualizado e inserir na i -ésima posição da seqüência parcial, considerando $[n'+1]/2 \leq i \leq [n'+1]$.

Avaliar a seqüência parcial com relação à expressão:

$$\sum_{i=2}^{n'+1} (n'+2 - i)d_{[i-1], [i]} \quad (6)$$

Escolher a melhor seqüência parcial com o menor valor calculado pela expressão (6). A seqüência parcial escolhida torna-se a nova, e atualizada \acute{a} , com $n' = n' + 1$. Atualizar os dados do novo vetor.

Passo 4: Voltar para o passo 3 se o vetor ordenado inicialmente ainda contiver dados, senão parar. Calcular o tempo total de fluxo através da expressão (3).

- **Raj2 (algoritmo 2)**

Este algoritmo usa o valor de T_i' como critério de ordenação inicial no passo 1 e o valor de T_i para desempates, enquanto os demais passos de 2 a 4 são os mesmos.

1.3.2 Bertolissi (2000)

Este trabalho também propõe um método heurístico construtivo baseado na ordenação inicial. Através da adaptação da propriedade de Chan e Bedworth² (1990 apud BERTOLISSI, 2000, p.460), é apresentado um método de ordenação inicial das tarefas para a construção da seqüência final.

A principal característica da ordenação inicial é que, para construí-la, deve-se calcular as parcelas das somas dos tempos de fluxo de duas tarefas adjacentes J_i e J_j , a partir do início de J_i (ou seja, no intervalo de tempo entre o início de J_i e o término de J_j), através da expressão:

$$F_{m(uv)} = 2t_{u1} + \sum_{k=2}^m t_{uk} + R_{m(uv)} \quad (7)$$

Onde:

$$\begin{aligned} R_{1(ij)} &= t_{j1}, \\ R_{2(ij)} &= t_{j2} + \max(R_{1(ij)}, t_{i2}) \\ R_{3(ij)} &= t_{j3} + \max(R_{2(ij)}, t_{i2} + t_{i3}) \\ R_{m(ij)} &= t_{jm} + \max(R_{m-1(ij)}, \sum_{k=2}^m t_{ik}) \end{aligned} \quad (8)$$

Para o processamento de duas tarefas em duas máquinas, por exemplo, o termo $R_{m(uv)}$ é igual ao cálculo de $\max(t_{v1}, t_{u2})$.

Assim, a heurística gera a ordenação de tarefas através da comparação entre as parcelas das somas dos tempos de fluxo de duas tarefas adjacentes J_i e J_j , a partir do início de J_i (ou seja, no intervalo de tempo entre o início de J_i e o término de J_j).

Entre as duas parcelas de soma dos tempos de fluxo de cada par de tarefas, escolhe-se a ordem do par que apresenta a menor soma e identifica-se a primeira tarefa.

A ordenação inicial é realizada através da ordem decrescente de número de identificações nas tarefas.

A segunda fase é semelhante ao processo de inserção de tarefas, proposta por Rajendran e Chaudhuri (1990).

A seguir é apresentado o algoritmo passo a passo.

- **Bert (algoritmo)**

Passo 1: Calcular as parcelas da soma dos tempos de fluxo de duas tarefas adjacentes J_u e J_v , nas duas seqüências possíveis ((uv) e (vu)) através da expressão (7);

Passo 2: Comparar os pares ($F_{m(uv)}$ e $F_{m(vu)}$), selecionar o menor e identificar a tarefa inicial do par de tarefas correspondente;

Passo 3: Executar o passo 2 para todos os pares de tarefas;

Passo 4: Contar o número de identificações de cada tarefa e ordená-las em ordem decrescente do número de identificações. Se algumas tarefas tiverem associadas igual número de identificações, efetuar o desempate ordenando primeiramente a tarefa que apresentar o menor tempo de processamento total;

Passo 5: Remover a primeira tarefa da ordenação e colocá-la como tarefa “principal”, formando a seqüência parcial \acute{o} com $n' = 1$.

Passo 6: Remover a primeira tarefa da ordenação de tarefas atualizada e inseri-la na i -ésima posição da seqüência parcial, de forma que $(n' + 1)/2 \bullet i \bullet (n' + 1)$;

Passo 7: Calcular a programação parcial de acordo com a expressão (6);

Passo 8: Escolher a seqüência parcial que tem o menor valor na expressão (6). A seqüência parcial escolhida compõe o novo \acute{o} com $n' = n' + 1$. Atualizar a nova seqüência de tarefas;

Passo 9: Se a seqüência inicialmente estabelecida ainda contiver dados voltar para o passo 6, senão parar. Calcule o tempo total de fluxo através da expressão (3).

² CHAN, D.; BEDWORTH, D.D. (1990). Design of a scheduling system for flexible manufacturing cells. *Int. J. Prod. Res.* 28(11), p.2037-2049 apud BERTOLISSI, E. (2000). Heuristic algorithm for scheduling in th no-wait flow-shop. *Journal of Materials Processing Technology*, v.107, p.460.

1.3.3 Aldowaisan e Allahverdi (2004)

Diferente dos três algoritmos anteriores, o algoritmo PH1(p) é um método heurístico composto de três fases. A primeira fase é chamada de ISA (*insertion sequence algorithm*) cujo objetivo é a obtenção da ordenação inicial.

A segunda fase é a aplicação, a partir da ordenação inicial, do método de inserção NEH proposto por Nawaz et al. (1983), resultando numa solução inicial.

Na terceira fase, o processo de melhoria que utiliza a permutação de tarefas é aplicado na seqüência obtida na fase anterior, resultando na solução final do problema.

- **PH1(p) (algoritmo)**

Passo 1: Desenvolva a seqüência inicial utilizando o procedimento ISA;

ISA - Passo 1: Escolha a tarefa i de modo que:

$$\sum_{j=1}^m t_{i,j} \leq \sum_{j=1}^m t_{r,j}, \text{ para toda tarefa } r \text{ não ordenada} \quad (9)$$

Remova-a para a primeira posição na ordenação inicial R.

ISA – Passo 2: Se não existir mais nenhuma tarefa a ser ordenada execute o passo 4. Senão, calcule o TCT (tempo total de fluxo) para cada tarefa i ainda não ordenada, após inseri-la na próxima posição disponível e escolha aquela que apresentar o menor TCT, removendo a tarefa i escolhida para a próxima posição disponível na ordenação inicial R.

ISA – Passo 3: Execute o passo 2.

ISA – Passo 4: Pare. A ordenação inicial ISA é a constante em R.

Passo 2: Remover a primeira tarefa de R e denominá-la como sendo a tarefa “principal”, formando a seqüência parcial \acute{o} com $n' = 1$.

Passo 3: Remover a próxima tarefa de R e inseri-la na i -ésima posição da seqüência parcial, considerando $1 \bullet i \bullet n'$.

Avaliar cada possibilidade de seqüência parcial e escolher a melhor

seqüência parcial com o menor valor calculado pela expressão (6). A seqüência parcial escolhida torna-se a nova, e atualizada á, com $n' = n' + 1$.

Passo 4: Voltar para o passo 3 se R contiver dados.

Passo 5: Executar o processo de melhoria que utiliza a permutação de tarefas na seqüência parcial obtida nos passos 3 e 4. Calcular o tempo total de fluxo da melhor seqüência escolhida através da expressão (3).

1.3.4 Algoritmos Modificados

Com base nos algoritmos apresentados anteriormente, foram elaboradas algumas adaptações no intuito de efetuar pequenas modificações que proporcionem melhorias nos resultados dos algoritmos originais.

As adaptações foram feitas no melhor método heurístico construtivo de duas fases (**Raj2**) e no melhor método heurístico composto de três fases (**PH1(p)**), também com o objetivo de forçar a comparação dos métodos propostos com os métodos existentes e com os métodos existentes modificados (melhorados).

- **Raj2Mod1**

A modificação é feita na segunda fase. O método de inserção utilizado no Raj2 é substituído pelo método de inserção do NEH, que contempla um maior número de seqüências parciais a cada tarefa a ser inserida, possibilitando melhores seqüências.

- **Raj2Mod2**

As adaptações são feitas na segunda e terceira fases. O método de inserção utilizado no Raj2 é substituído pelo método de inserção do NEH.

Além disso, uma terceira fase é inserida, contendo um método de melhoria que utiliza a permutação de tarefas, possibilitando uma análise ainda maior das seqüências parciais possíveis.

- **Raj2Mod3**

O algoritmo Raj2Mod3 é similar ao Raj2Mod2, porém na terceira fase é proposto um

método de melhoria que utiliza a inserção de tarefas, possibilitando uma análise ainda maior das seqüências possíveis em relação ao algoritmo modificado anterior.

- **PH1(p)Mod**

A única adaptação é na terceira fase, ou mais precisamente no passo 5, onde o método de melhoria que utiliza a permutação de tarefas é substituído pelo método de melhoria que utiliza a inserção de tarefas, fortalecendo ainda mais o algoritmo que a literatura considera como o melhor método heurístico composto de três fases para solução do problema NWFS, com minimização do tempo total de fluxo.

Após a análise dos melhores métodos e suas adaptações sugeridas, no próximo capítulo será proposto, um novo método heurístico construtivo de duas fases.

CAPÍTULO 2

Proposição de Um Método Heurístico Construtivo de Duas Fases

O método heurístico proposto neste capítulo possui duas fases, e será denominado C_SN(2F), para fins de abreviação, identificação e comparação com os demais métodos já existentes:

- Fase 1: Ordenação inicial das tarefas;
- Fase 2: Construção da seqüência através do método de inserção de tarefas não seqüenciadas e, a cada iteração, do processo de melhoria que utiliza a inserção de tarefas.

2.1 Fase 1: Ordenação inicial das tarefas

- **Propriedade Utilizada**

Como já foi verificado no capítulo anterior, o método heurístico de Bertolissi (2000) apresenta a adaptação de uma propriedade que o próprio autor considera como a base da sua pesquisa, trata-se da possibilidade de cálculo das parcelas das somas dos tempos de fluxo de duas tarefas adjacentes J_i e J_j , a partir do início de J_i (ou seja, no intervalo de tempo entre o início de J_i e o término de J_j).

Para cada par de tarefas, existem duas seqüências parciais possíveis, independente da posição que elas ocupam na seqüência de tarefas, como por exemplo, demonstrado na figura 2.1.

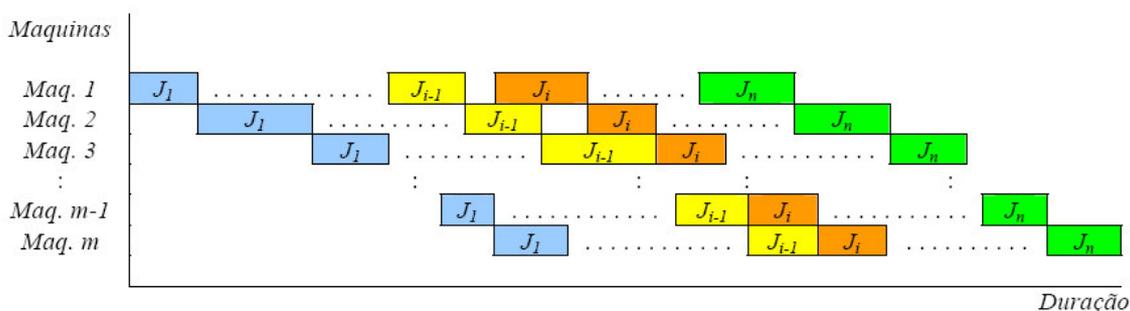


Figura 2.1- Gráfico de Gantt para problema *no-wait flow shop*

Dessa forma, segundo a adaptação de Bertolissi (2000), seria possível calcular a parcela da soma dos tempos de fluxo de todos os pares de tarefas adjacentes possíveis através da expressão (7) mencionada na seção 1.3.2.

Com a parcela da soma dos tempos de fluxo de duas tarefas adjacentes, é possível construir a matriz F ($n \times n$), considerando todas as máquinas. O método proposto neste capítulo, assim como os outros dois métodos propostos no capítulo 5, utilizam-se desta propriedade.

- **Primeira parte do algoritmo (fase 1 - ordenação inicial das tarefas)**

Na expectativa de minimização do *flowtime*, a matriz F ($n \times n$) sugere a possibilidade da criação de uma ordenação inicial a partir das menores parcelas das somas dos tempos de fluxo de duas tarefas adjacentes.

A adoção desta forma de ordenação inicial é baseada no fato de que, para minimização do tempo total de fluxo de determinada seqüência de tarefas, deve-se ordenar as tarefas a partir daquelas que apresentam menor tempo de processamento. Porém, para atender ao caso particular de NWFS, é necessário considerar também o *delay* entre o início de determinada tarefa e o início da tarefa subsequente na primeira máquina.

Assim como em Bertolissi (2000), a ordenação inicial é obtida através da utilização da matriz F ($n \times n$). Dessa forma, seja $J = \{J_1, J_2, \dots, J_n\}$ um conjunto de n tarefas que devem ser processadas, e S o conjunto das tarefas programadas.

Passo 1

$$J = \{J_1, J_2, \dots, J_i, \dots, J_n\};$$

$$S = \emptyset;$$

Selecione o menor elemento $F_m(J_i, J_j)$;

$S = \{J_i, J_j\}$;

$J \leftarrow J - \{J_i, J_j\}$;

$u \leftarrow J_{[2]}$;

$k = 3$;

Passo 2

Selecione o menor elemento $F_m(J_u, J_v)$ tal que $J_v \in J$;

$S \leftarrow S \cup \{J_v\}$;

$J \leftarrow J - \{J_v\}$;

$u \leftarrow J_{[k]}$;

$k \leftarrow k + 1$;

Se $J \neq \emptyset$, volte para o **Passo 2**. Caso contrário (ordenação inicial das tarefas concluída) vá para o **Passo 3**.

2.2 Fase 2: Construção da seqüência final das tarefas

- **Método de construção utilizado**

A construção da seqüência final (solução) é uma modificação dos passos 3 e 4 do algoritmo proposto por Framinan e Leisten (2003), para o caso *no-wait flow shop*.

A modificação ocorre na segunda fase. A partir da ordenação inicial obtida na Fase 1, a segunda fase é baseada na fase final da heurística NEH e a cada iteração é aplicado o método de melhoria que utiliza a inserção de tarefas.

Esta estrutura possibilita maior número de ordenações parciais a serem testadas e não inviabiliza o tempo de computação, como será visto no capítulo 3.

- **Segunda parte do algoritmo (fase 2 – construção da seqüência final)**

A segunda parte do algoritmo proposto é definida pelos seguintes passos:

Passo 3

Selecione as duas primeiras tarefas da ordenação inicial S ;

Passo 4

Para $L = 3$ a n ;

- Selecione a tarefa que ocupa a $L - \text{ésima}$ posição na ordenação inicial S ;
- Examine as L possibilidades de inserir a tarefa na seqüência parcial até então obtida, adotando aquela que leva a menor soma dos tempos de fluxo;
- Considerando toda a Vizinhança de Inserção da seqüência parcial com L tarefas, constituída de $(k - 1)^2$ seqüências, determine a seqüência S^* associada a menor soma dos tempos de fluxo;
- Atualize, com a seqüência S^* as L primeiras posições da seqüência S .

Após a apresentação, no capítulo anterior, dos métodos heurísticos existentes mais importantes e suas modificações, e após a proposição de um novo método heurístico construtivo de duas fases (neste capítulo), serão apresentados nos próximos dois capítulos, as experimentações computacionais.

CAPÍTULO 3

Experimentação Computacional – Parte A

3.1 Planejamento dos Experimentos

A experimentação computacional foi dividida em três partes:

Parte A: serão avaliados os algoritmos já existentes Raj1, Raj2 e Bert, o algoritmo modificado Raj2Mod1, e o método proposto C_SN(2F), todos métodos heurísticos construtivos de duas fases;

Parte B: serão avaliados os algoritmos modificados Raj2Mod2, Raj2Mod3 e PH1(p)Mod e o algoritmo existente PH1(p), todos métodos heurísticos compostos de 3 fases;

Parte Final: serão avaliados os dois melhores métodos heurísticos respectivamente das experimentações Parte A e Parte B, e os dois algoritmos de fase única que serão propostos no capítulo 5.

Em todas as experimentações foram testados 7.200 problemas, divididos em três grupos. O primeiro grupo de 2.000 problemas, considerados de pequeno porte, subdivididos em 20 classes, de acordo com o número de tarefas $n \in \{5, 6, 7, 8, 9\}$ e o número de máquinas $m \in \{5, 10, 15, 20\}$.

O segundo grupo, envolveu 3.200 problemas, considerados de médio porte, com número de tarefas $n \in \{10, 20, 30, 40, 50, 60, 70, 80\}$ e máquinas $m \in \{5, 10, 15, 20\}$, totalizando 32 classes de problemas

O terceiro grupo foi constituído por 2.000 problemas, subdivididos em 20 classes, considerados de grande porte, com número de tarefas $n \in \{90, 100, 110, 120, 130\}$ e $m \in \{5, 10, 15, 20\}$.

Dessa forma, para cada classe (n,m), foram testados 100 problemas com objetivo principal de minimizar a possibilidade de erro nas amostras.

3.2 Obtenção dos Dados

Os dados dos problemas foram gerados de forma aleatória, sendo os tempos de processamento, números inteiros, distribuídos uniformemente no intervalo [1, 99].

Os métodos heurísticos foram codificados na linguagem de programação Borland Delphi e o equipamento utilizado foi um Intel Celeron 2,66 GHz, com 256 mb de memória RAM e disco rígido de 40 Gb.

3.3 Método de Análise

Os resultados obtidos nas experimentações computacionais foram analisados através das porcentagens de sucesso, das porcentagens dos desvios médios relativos e dos tempos médios de computação. Para facilitar a visualização dos resultados, eles são apresentados em forma de gráficos, elaborados através das tabelas constantes no Anexo A.

A porcentagem de sucesso é calculada pelo número de vezes que o algoritmo apresentou a melhor solução (isoladamente ou não), para o caso, menor tempo total de fluxo, dividido pelo número de problemas resolvidos.

A porcentagem do desvio médio relativo quantifica o desvio de determinado método heurístico em relação ao melhor resultado obtido para o mesmo problema, sendo calculado como segue:

$$100 * \left(\frac{Flowtime_{heur} - Flowtime_{melhor_heur}}{Flowtime_{melhor_heur}} \right)$$

O tempo médio de computação (em segundos) é calculado pela soma dos tempos de processamento em cada problema, dividido pelo número total de problemas resolvidos.

Os gráficos e tabelas do Anexo A foram elaborados para cada classe de problemas, ou seja, problemas com 5, 10, 15 e 20 máquinas. Além disso, os resultados foram agrupados com relação ao número de máquinas (de 5 a 20 máquinas juntos), para possibilitar uma análise global dos resultados.

3.4 Análise dos Resultados

Os resultados obtidos são analisados em três partes e na seguinte ordem: Porcentagem de Sucesso; Porcentagem de Desvio Médio Relativo; e Tempo Médio de

Computação

3.4.1 Análise da Porcentagem de Sucesso

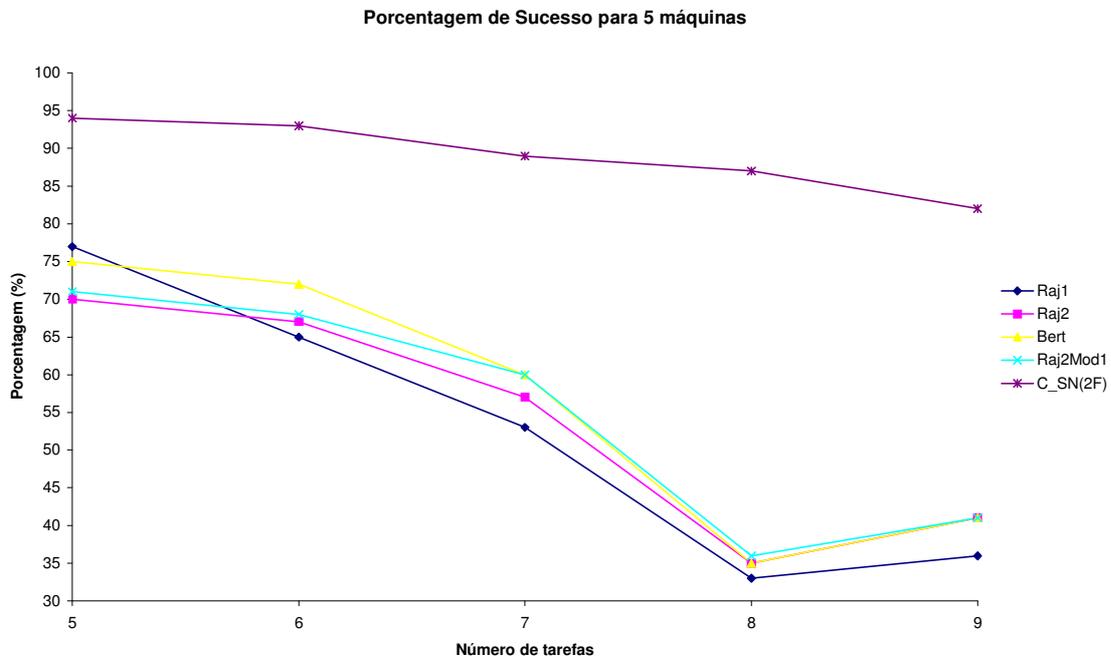


Figura 3.1– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

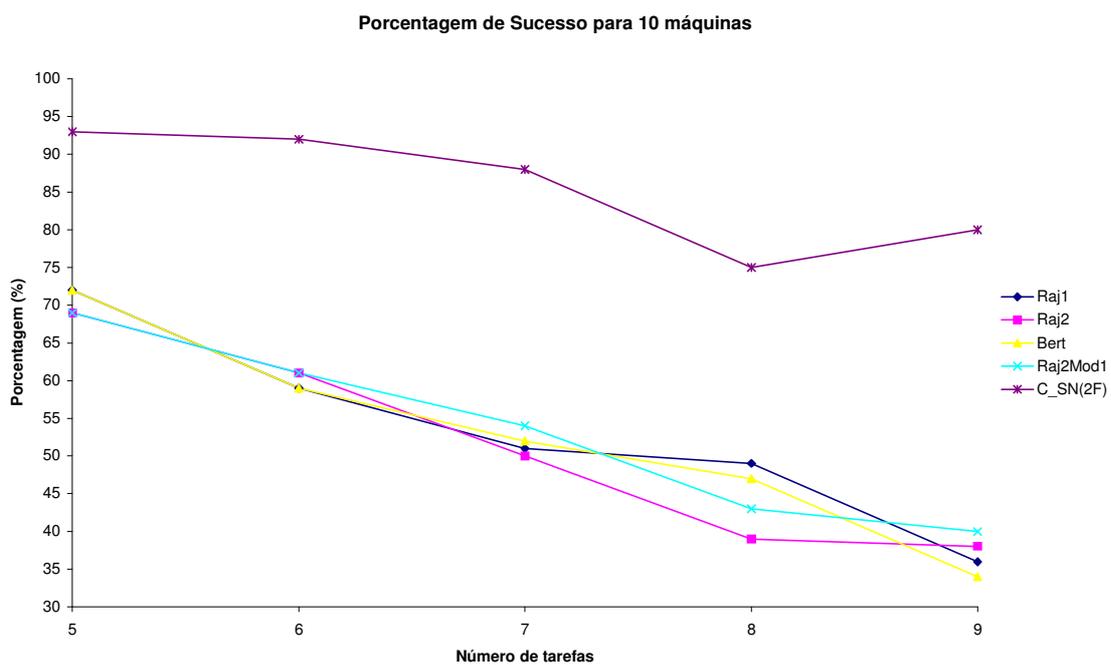


Figura 3.2– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

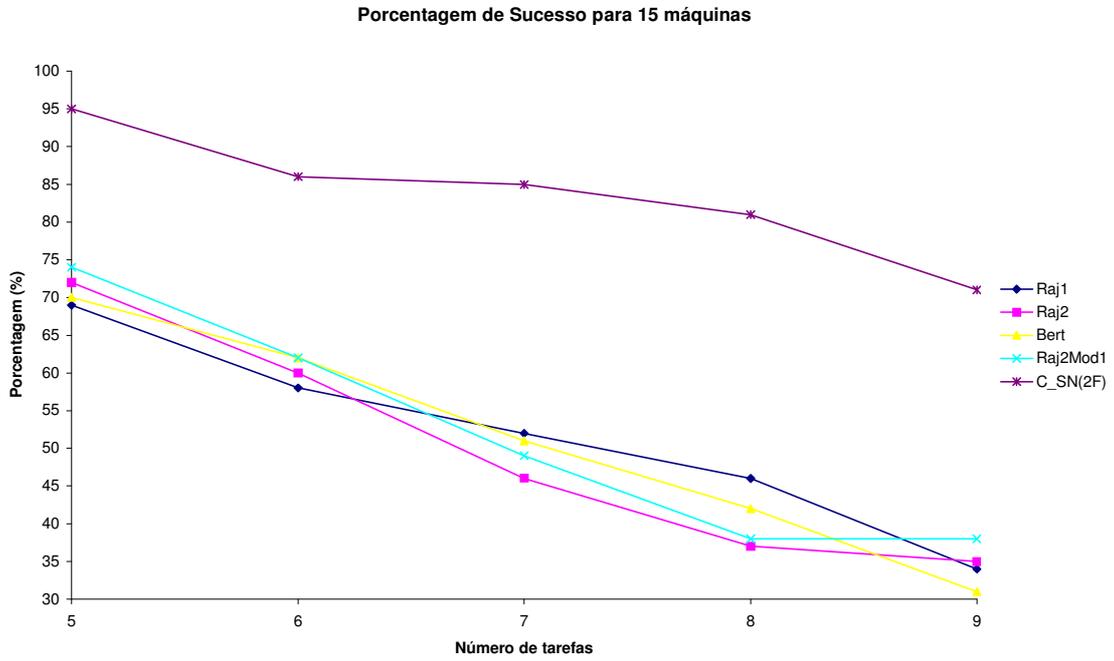


Figura 3.3– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

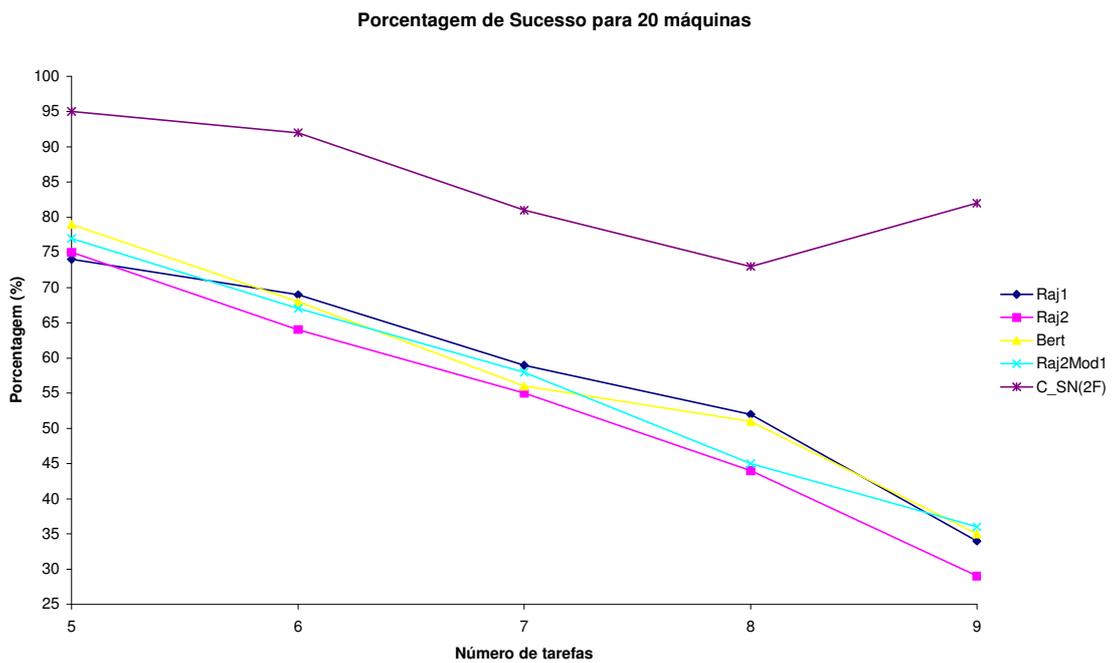


Figura 3.4– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

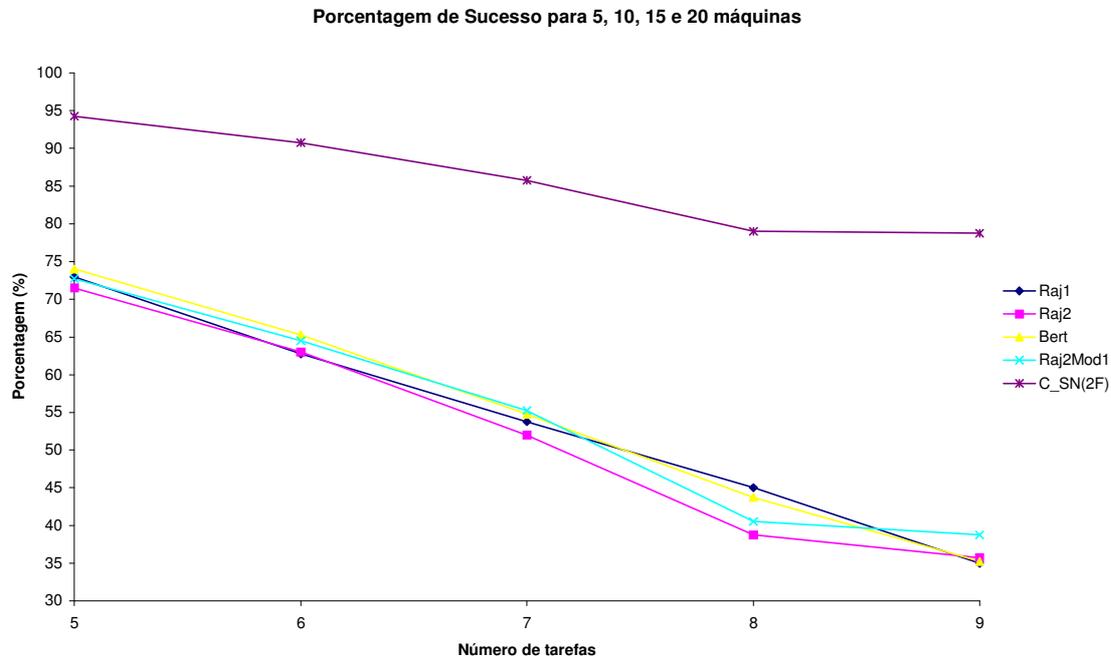


Figura 3.5– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Uma primeira análise é realizada com relação ao desempenho dos algoritmos quanto a variação do número de tarefas. Percebe-se que, pelos fatos de ocorrerem menos empates e de diminuir as possibilidades de se alcançar o resultado ótimo, a porcentagem de sucesso é menor para cada método com 9 tarefas, em relação aos problemas de 5 tarefas. A figura 3.5 mostra esses resultados.

Analisando-se as figuras 3.1 a 3.5, verifica-se que a porcentagem de sucesso do algoritmo C_SN(2F) é superior a todos os outros algoritmos, em todos os casos de variação de 5 a 9 tarefas e 5 a 20 máquinas.

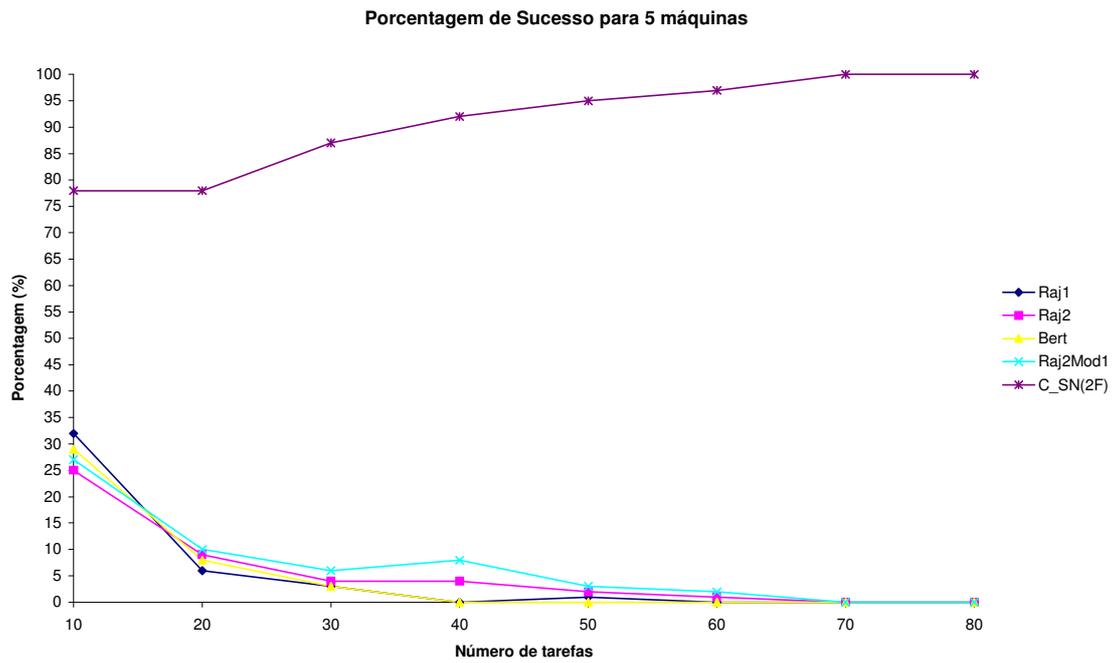


Figura 3.6– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

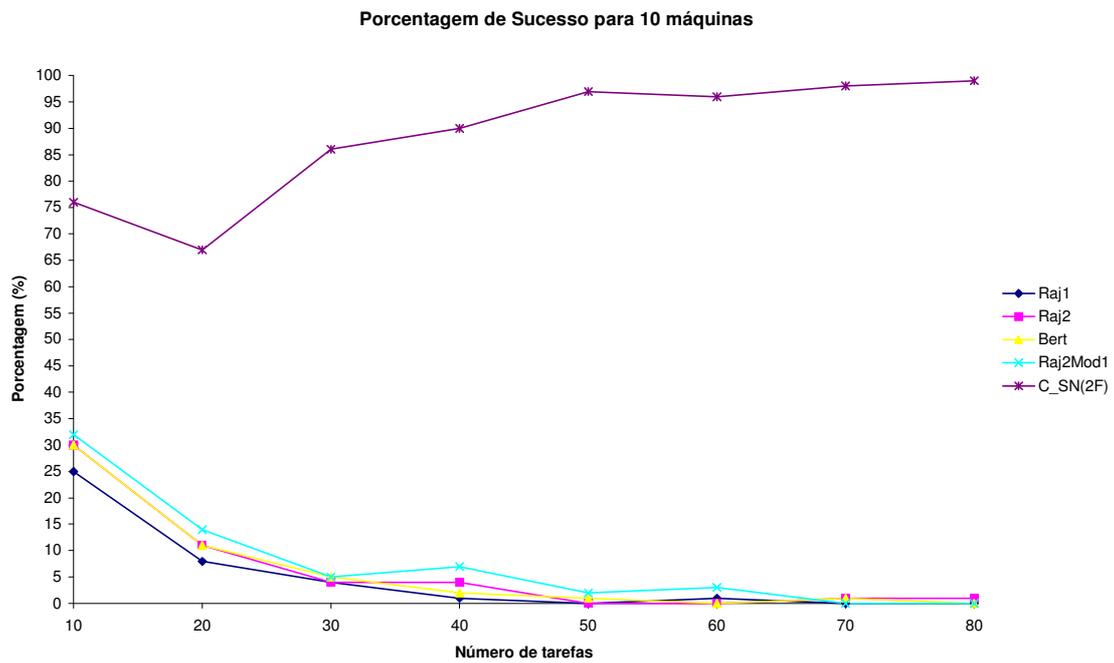


Figura 3.7– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

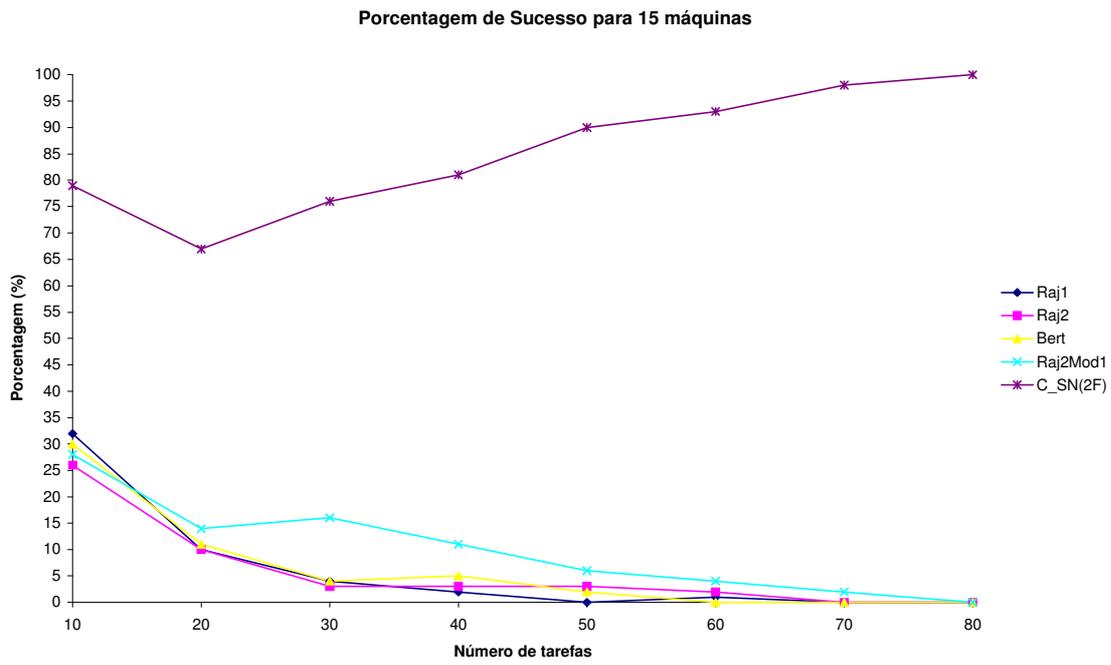


Figura 3.8– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

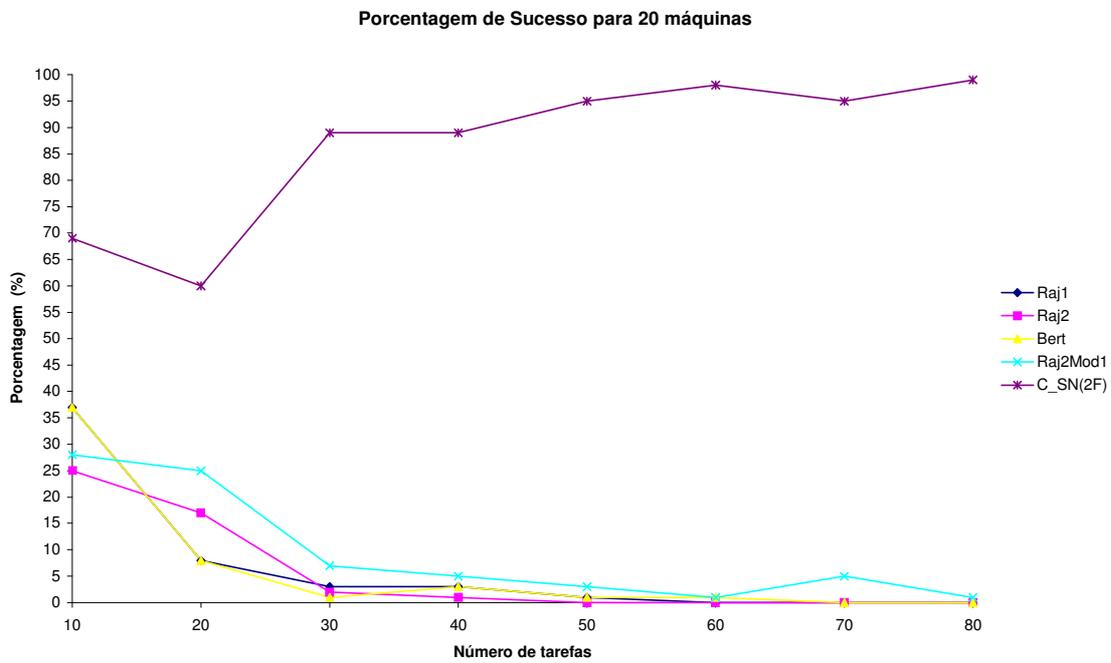


Figura 3.9– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

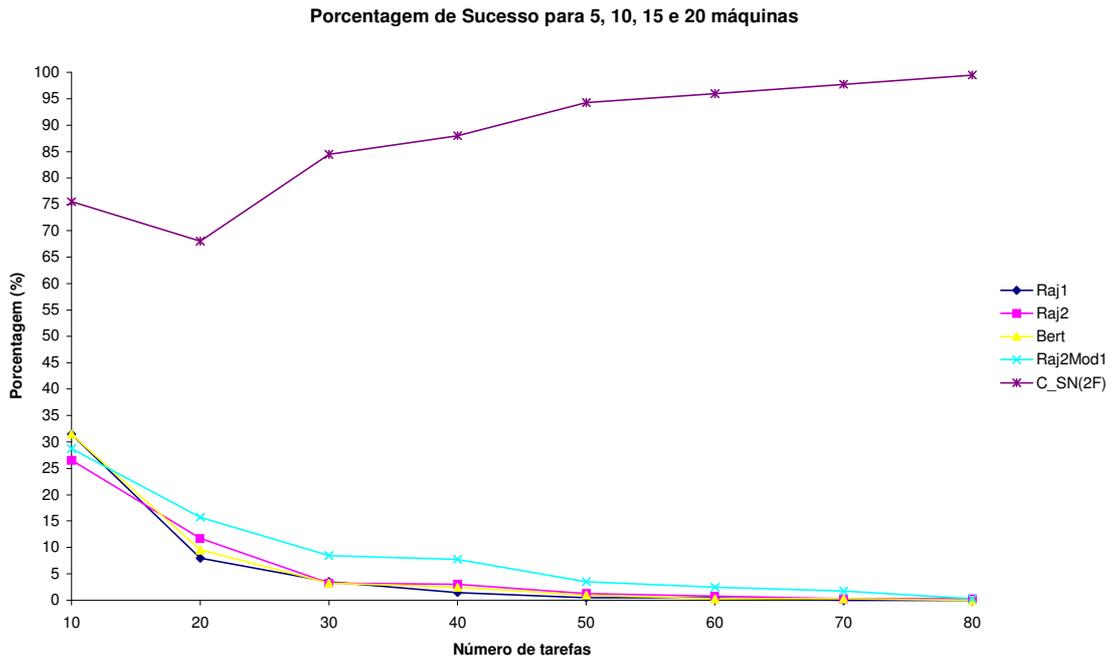


Figura 3.10– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Nas figuras 3.6 a 3.10, verifica-se que a partir de problemas com 20 tarefas, o algoritmo C_SN(2F) demonstra tendência de aumento da porcentagem de sucesso à medida que aumenta o número de tarefas.

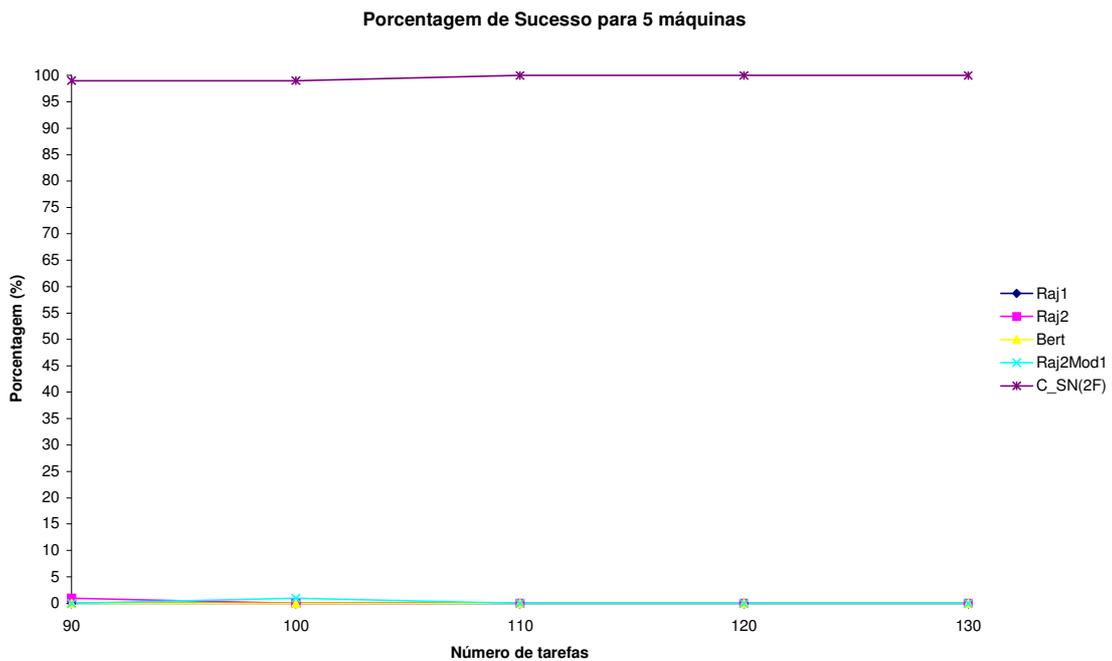


Figura 3.11– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

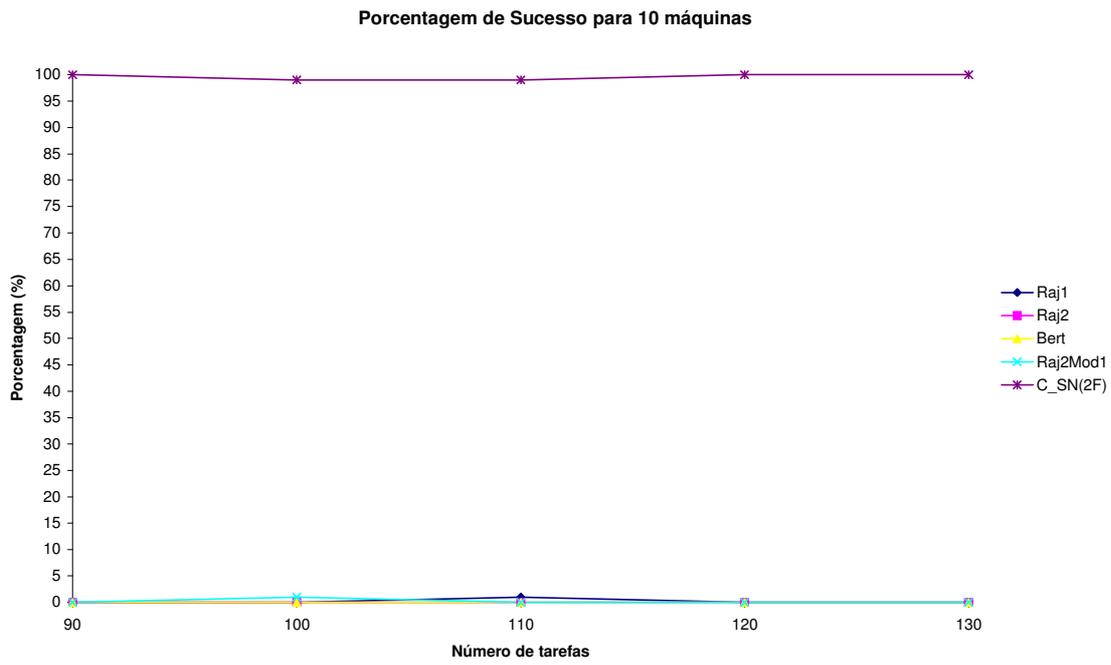


Figura 3.12– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

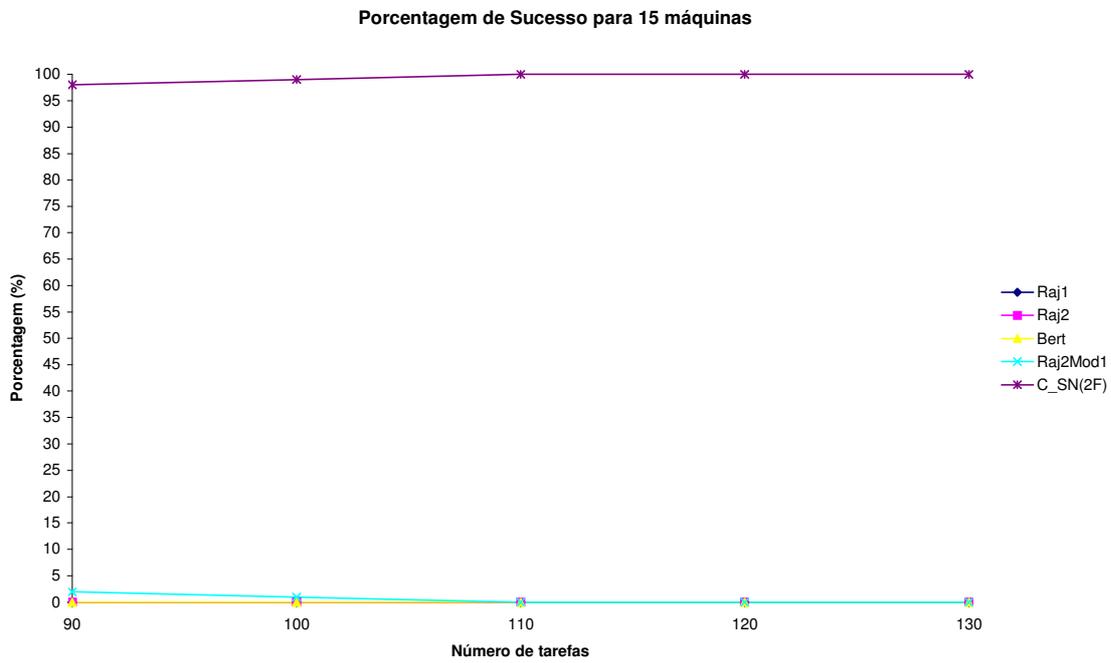


Figura 3.13– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

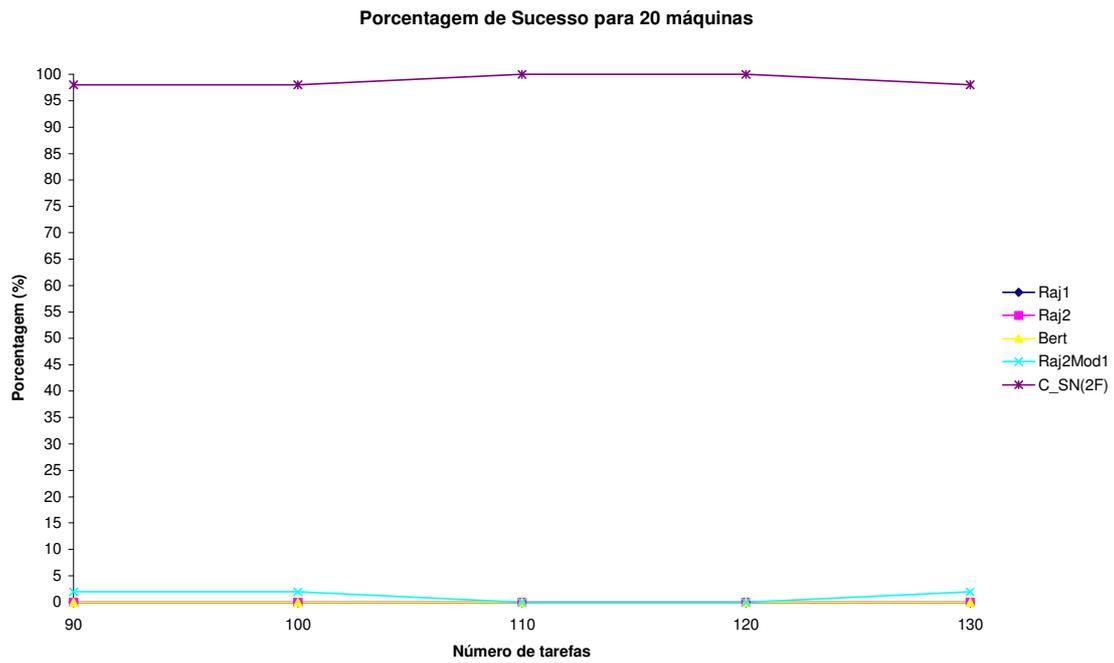


Figura 3.14– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

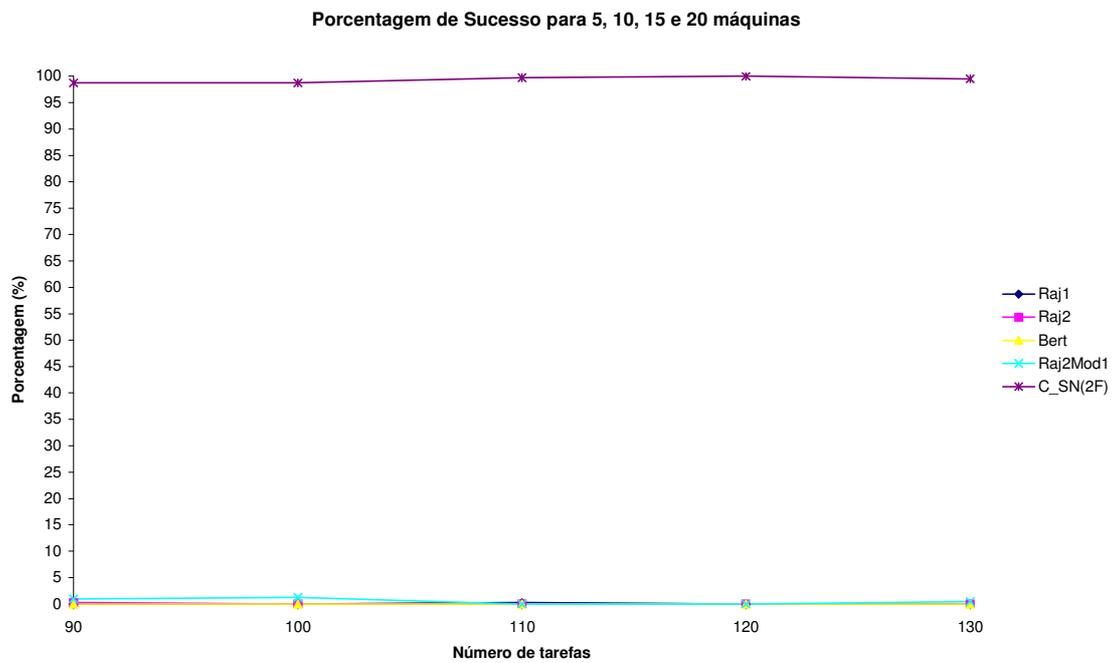


Figura 3.15– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Nas figuras 3.11 a 3.15 fica consolidado que independente do número de máquinas, o método C_SN(2F) é superior aos demais existentes, principalmente para problemas de grande porte.

3.4.2 Análise da Porcentagem de Desvio Médio Relativo

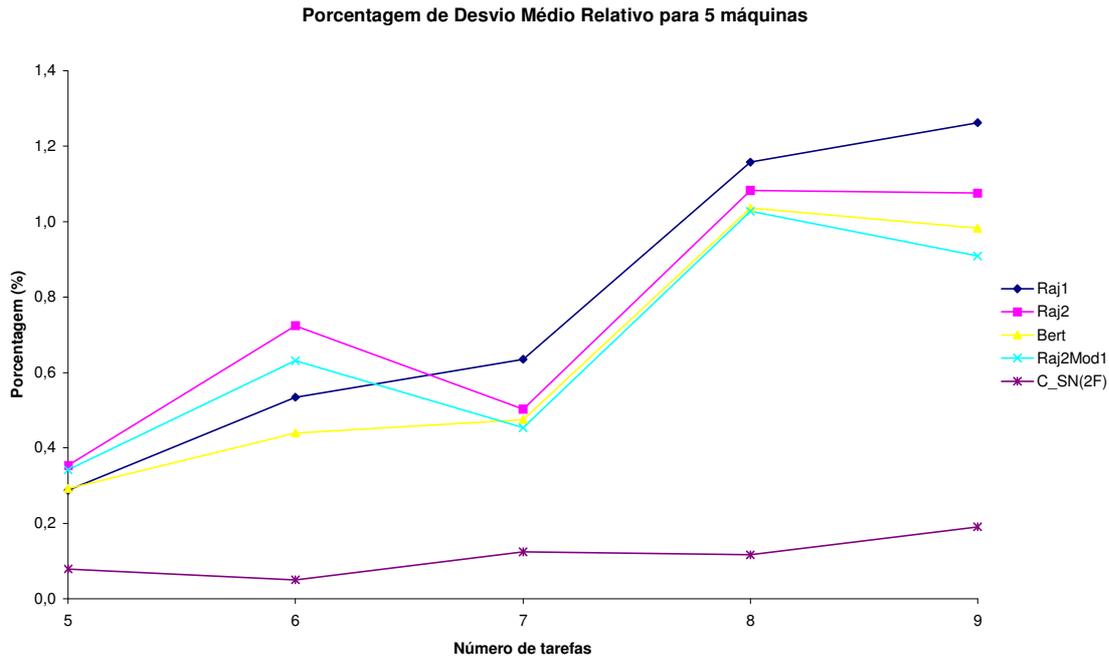


Figura 3.16– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas

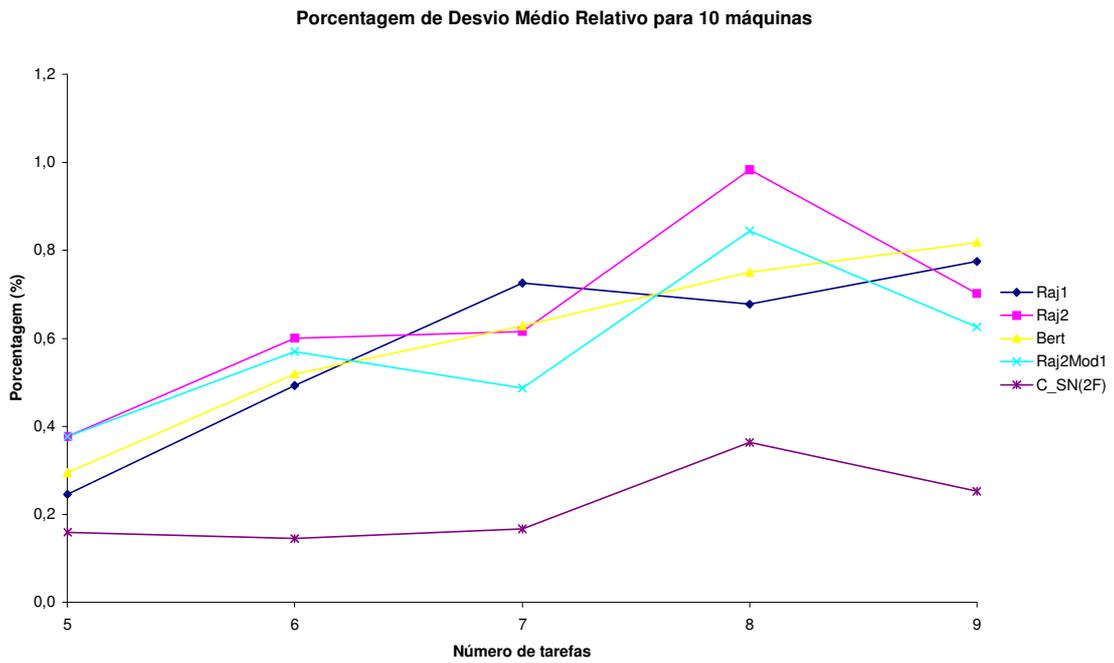


Figura 3.17– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas

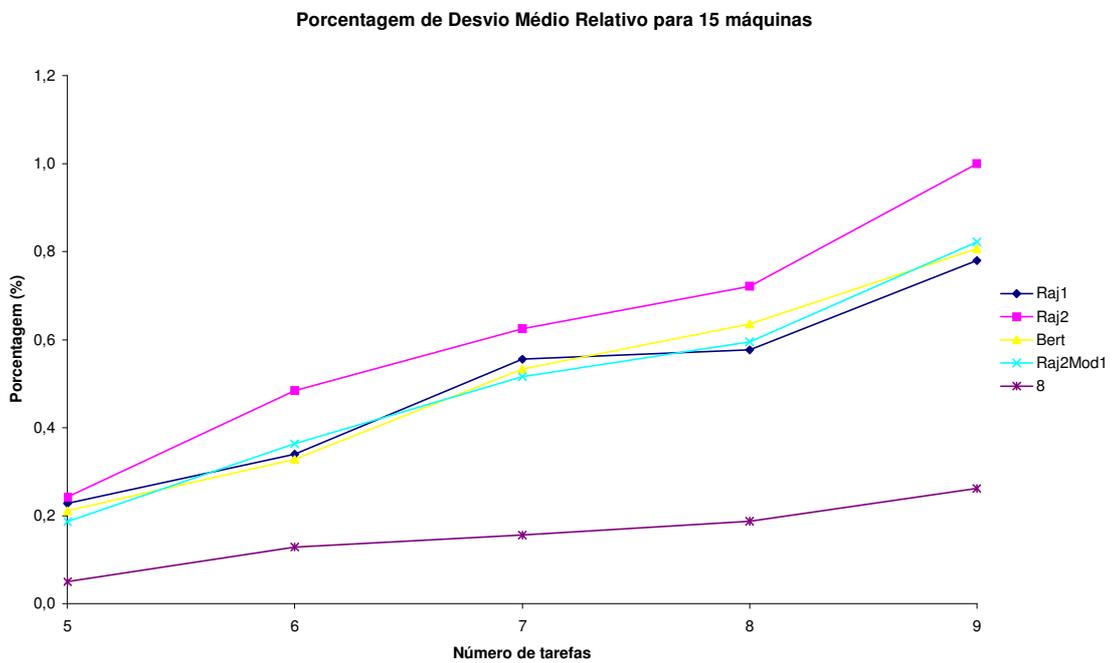


Figura 3.18– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas

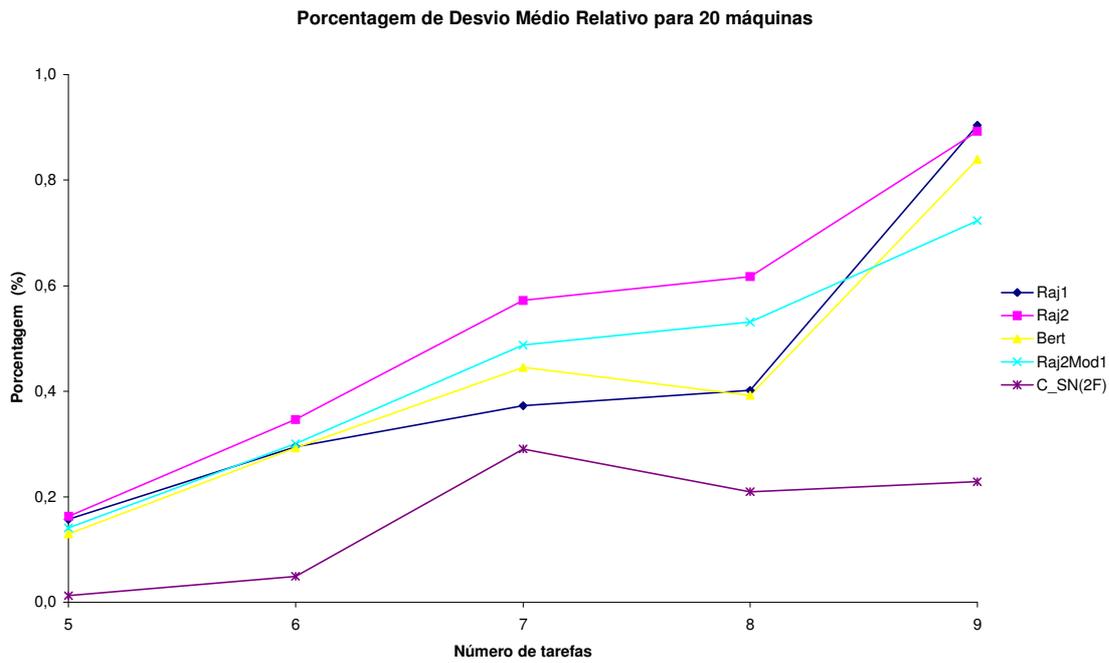


Figura 3.19– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas

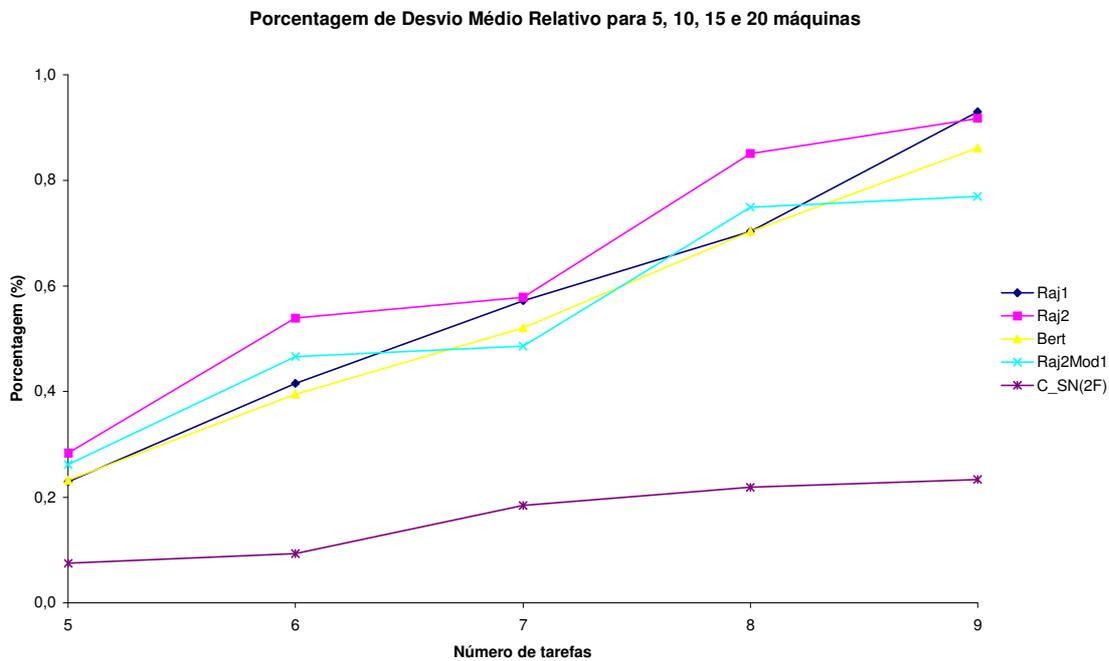


Figura 3.20– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Percebe-se que o desvio médio relativo, nos problemas de pequeno porte, é semelhante em quase todos os algoritmos, porém o método C_SN(2F) apresenta menor desvio médio relativo que os demais métodos, principalmente devido ao fato de apresentar

soluções de melhor qualidade.

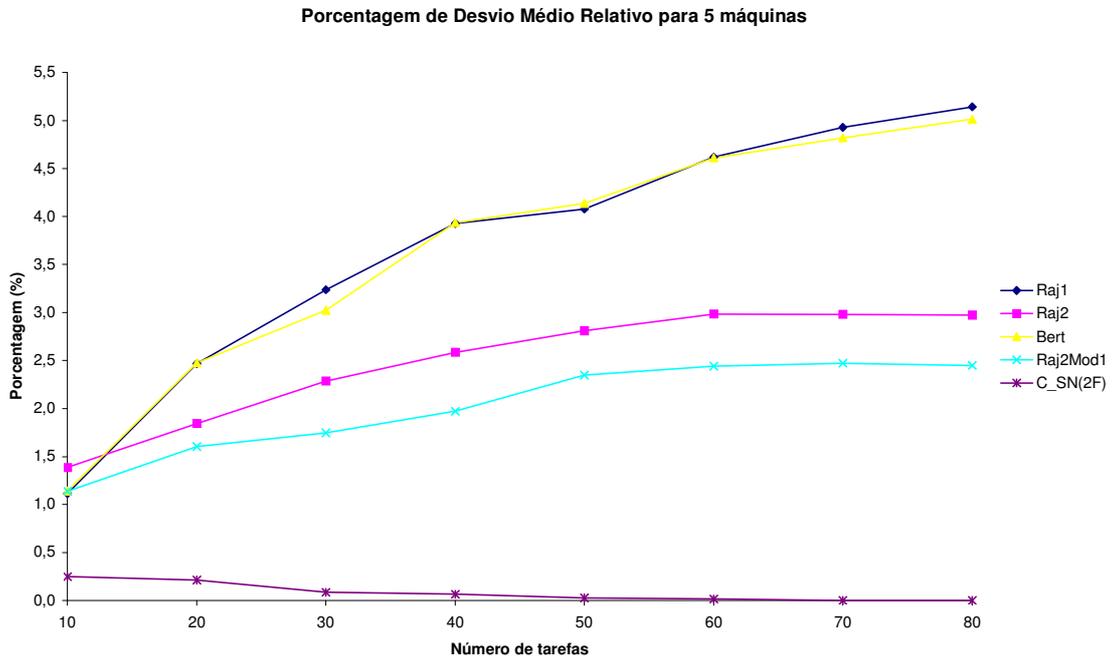


Figura 3.21– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas

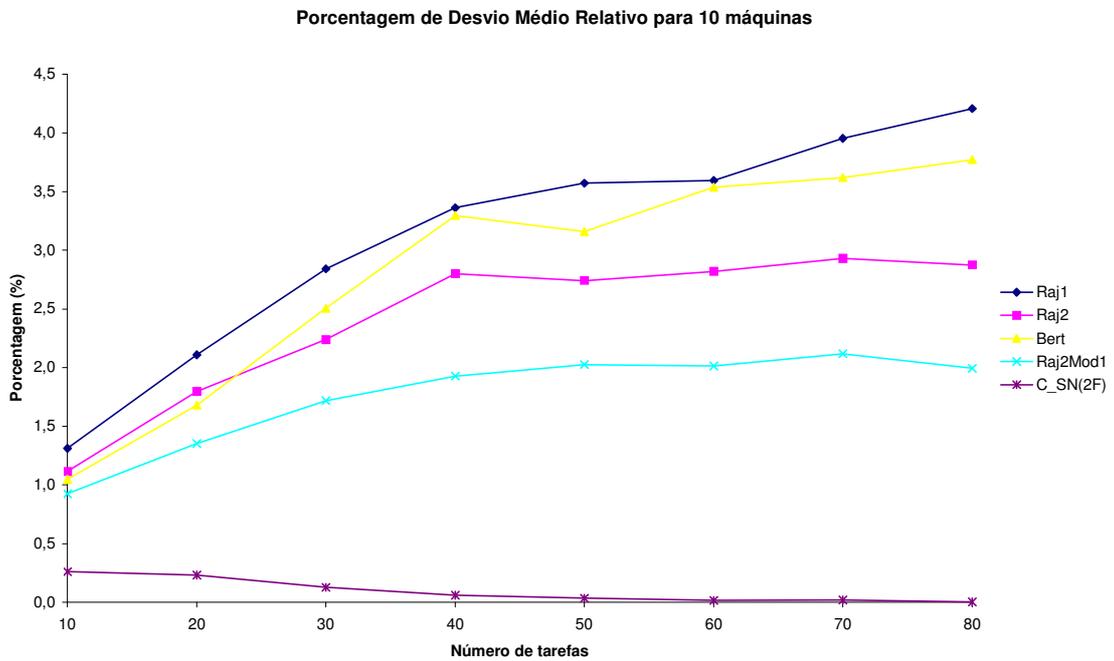


Figura 3.22– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas

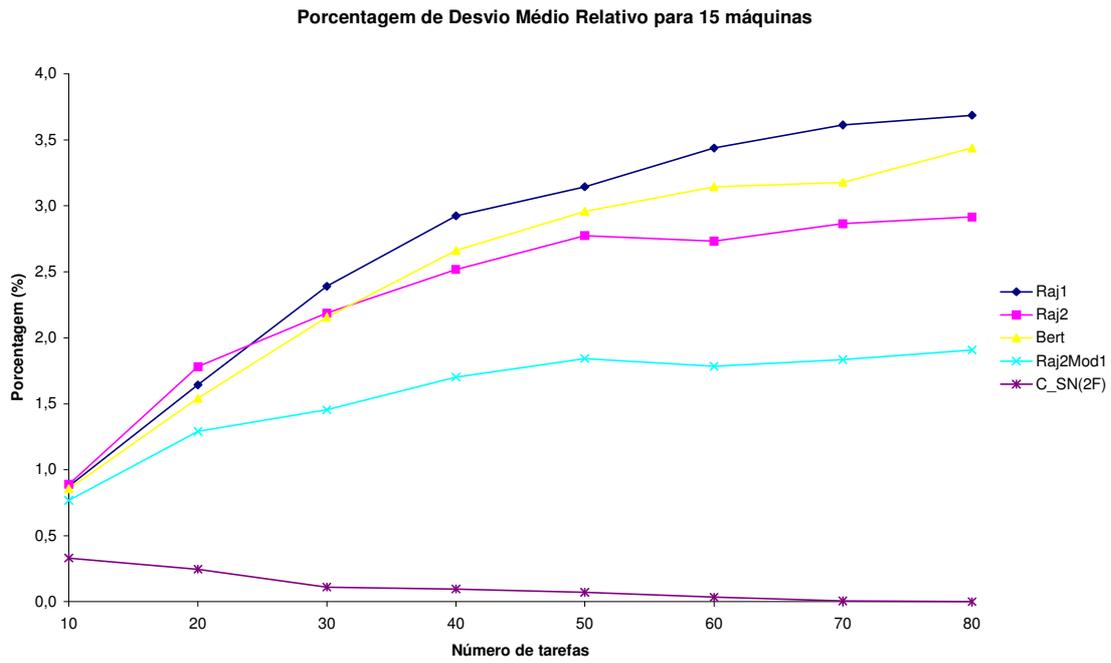


Figura 3.23– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas

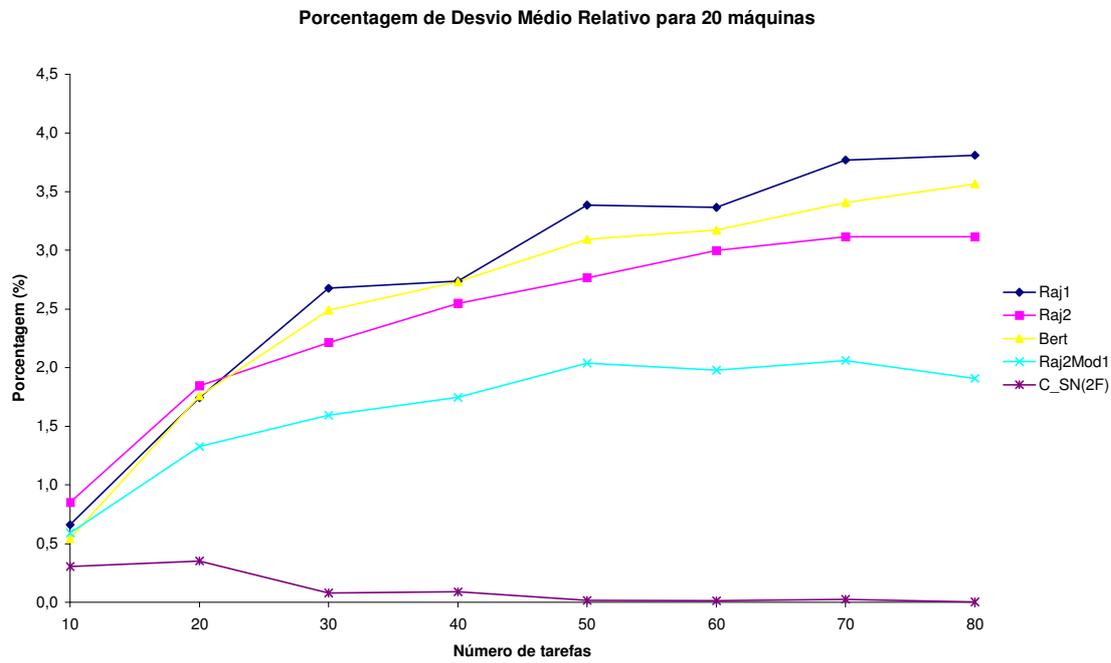


Figura 3.24– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas

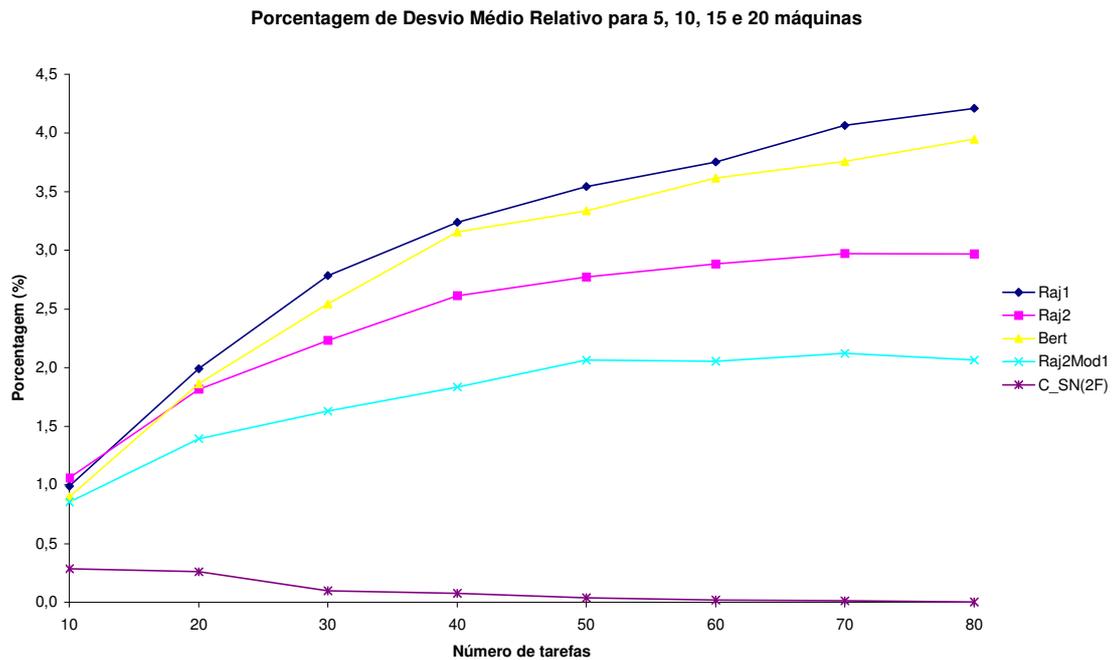


Figura 3.25– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

A partir dos problemas com 20 tarefas, o algoritmo C_SN(2F) destaca-se, demonstrando um desvio médio relativo bem inferior aos demais algoritmos, principalmente em relação aos algoritmos Raj1, Raj2 e Bert.

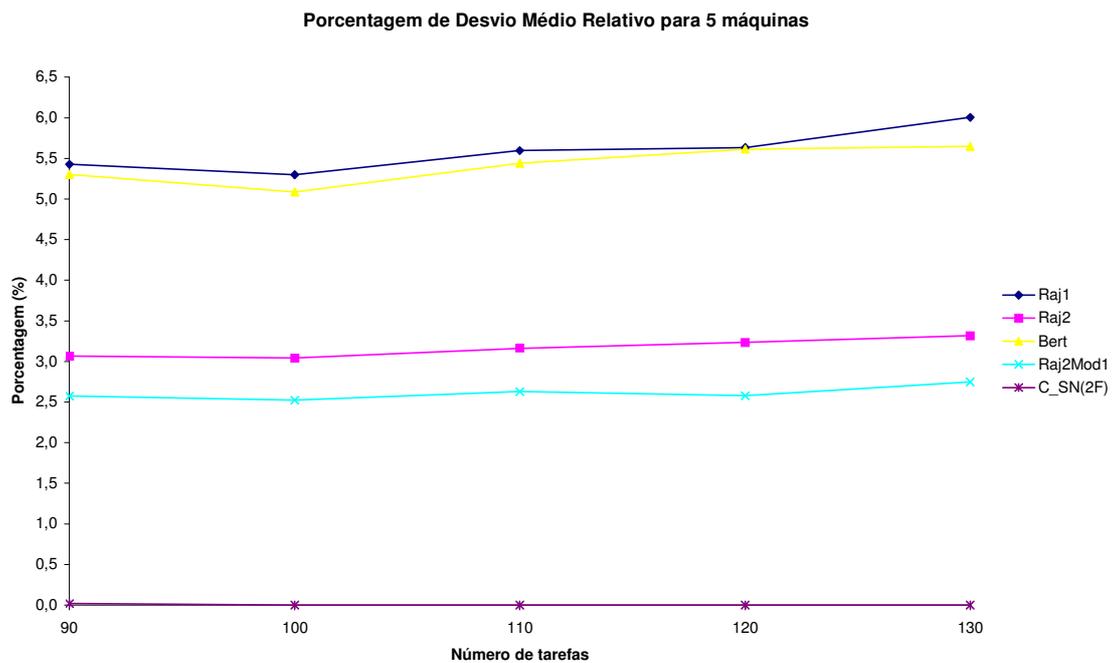


Figura 3.26– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas

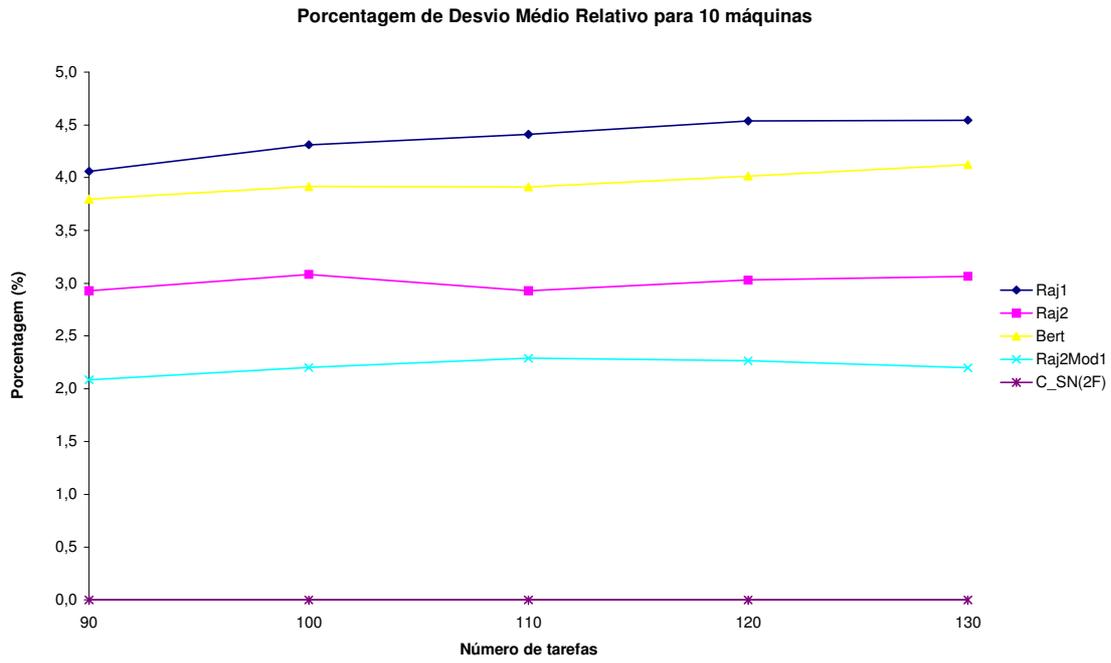


Figura 3.27– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas

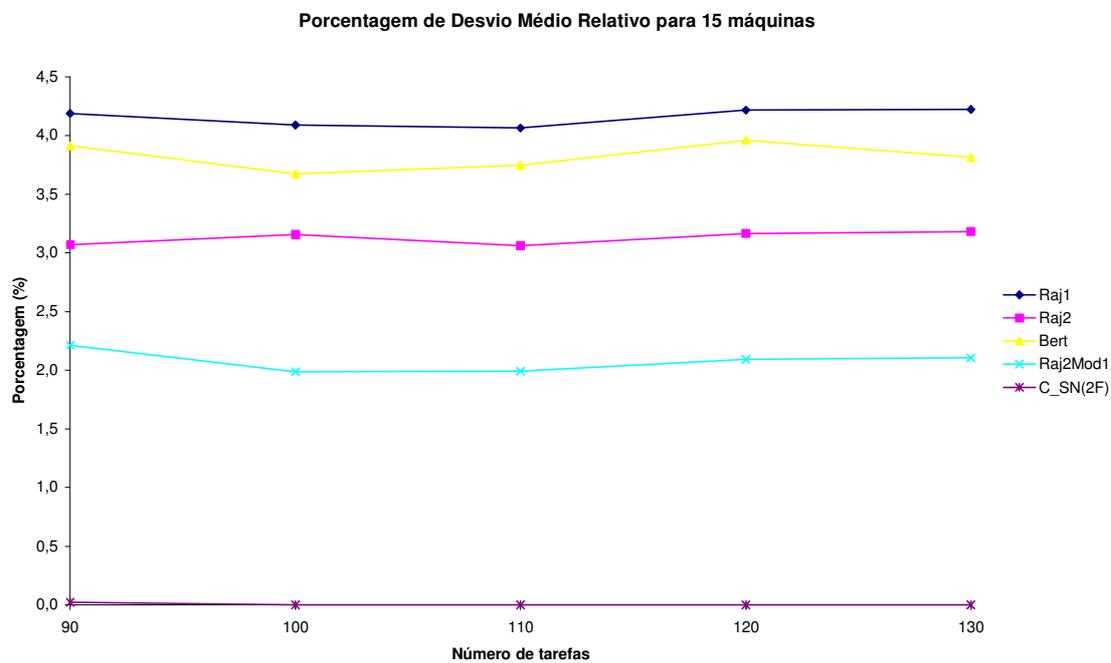


Figura 3.28– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas

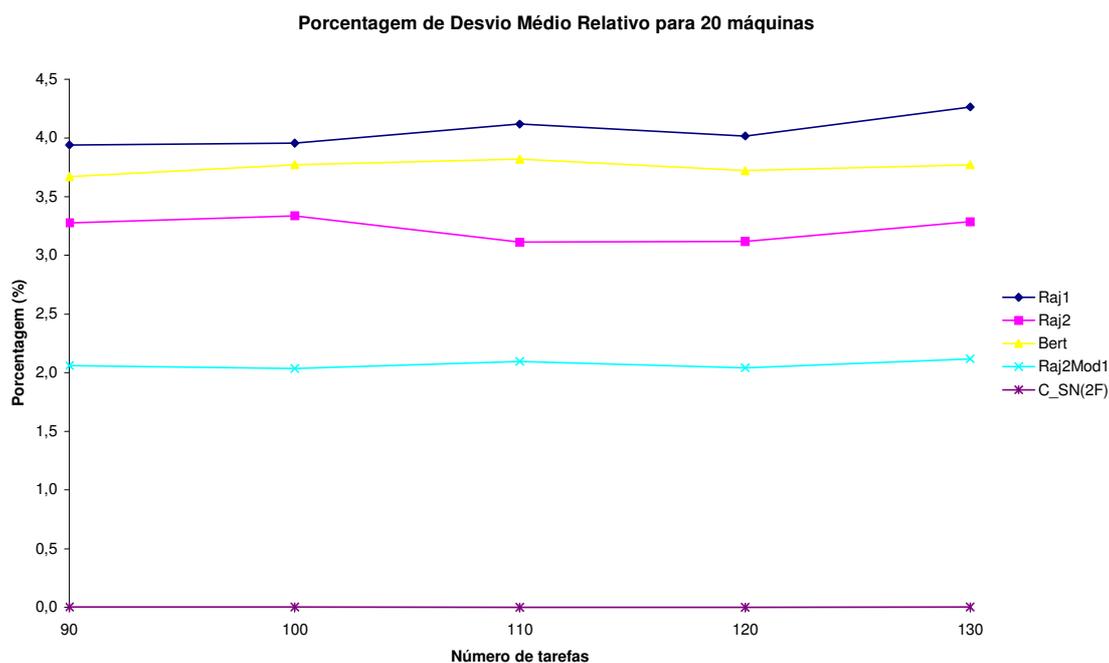


Figura 3.29– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas

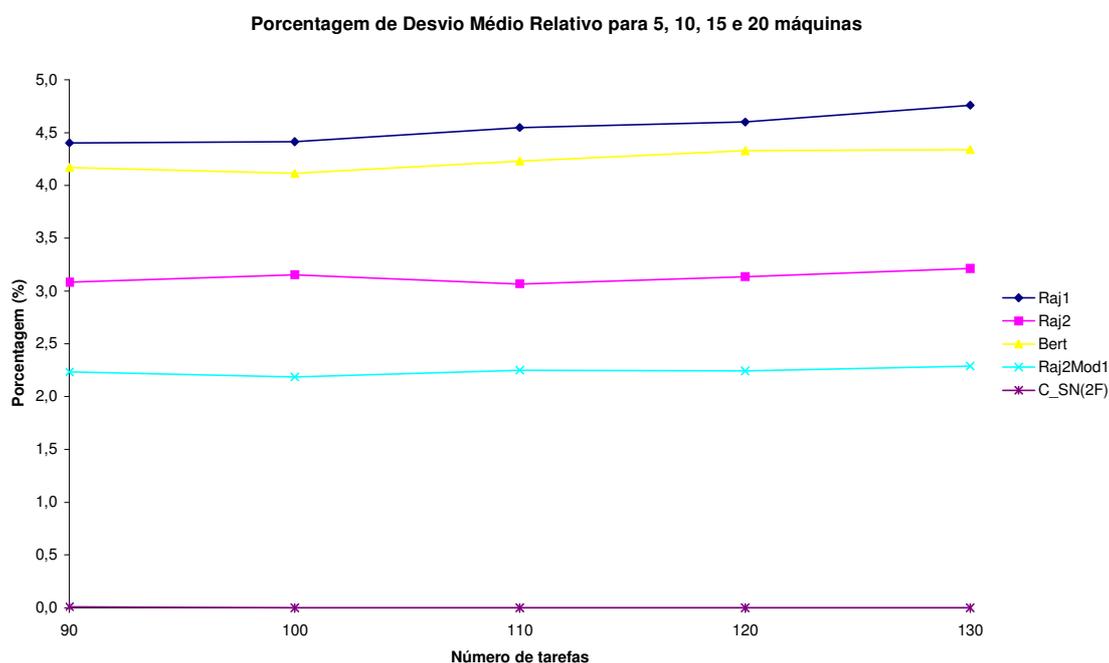


Figura 3.30– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Verifica-se na figura 3.30 que, para os problemas de grande porte, a heurística C_SN(2F) também é a que apresenta menor porcentagem de desvio médio relativo (quase totalmente nulo), enquanto o segundo melhor método apresenta uma porcentagem de desvio médio relativo entre 2% e 2,5%.

Em relação aos algoritmos já existentes, o método C_SN(2F) apresenta soluções pelo menos 3% melhores, como pode ser observado na figura 3.30.

3.4.3 Tempo Médio de Computação

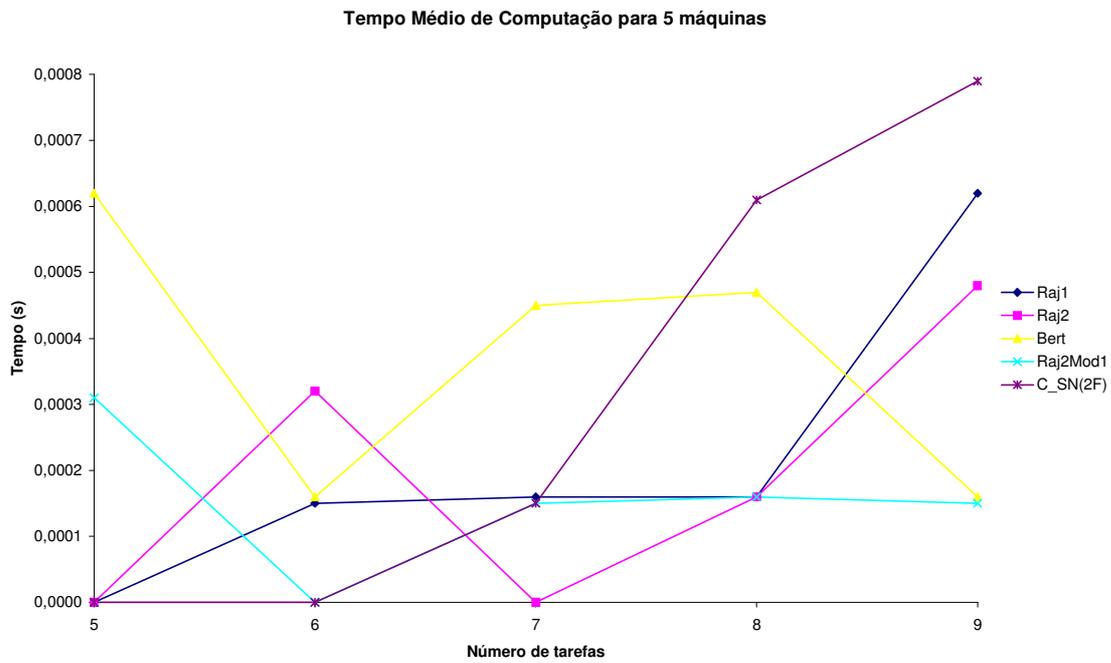


Figura 3.31– Tempo médio de computação vs. Número de tarefas, para 5 máquinas

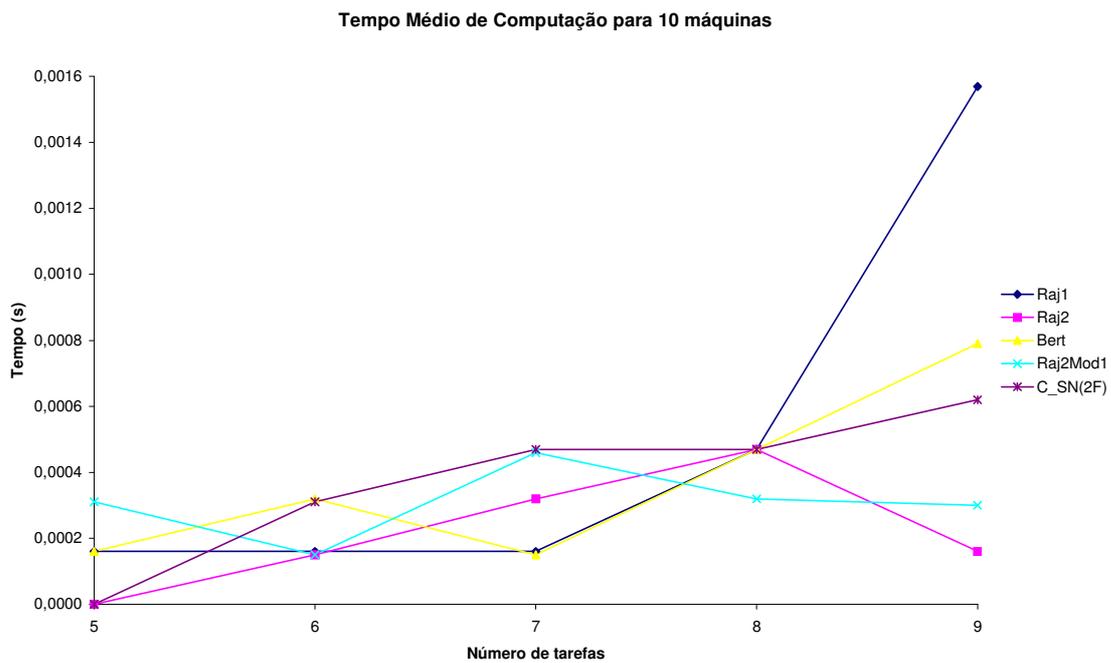


Figura 3.32– Tempo médio de computação vs. Número de tarefas, para 10 máquinas

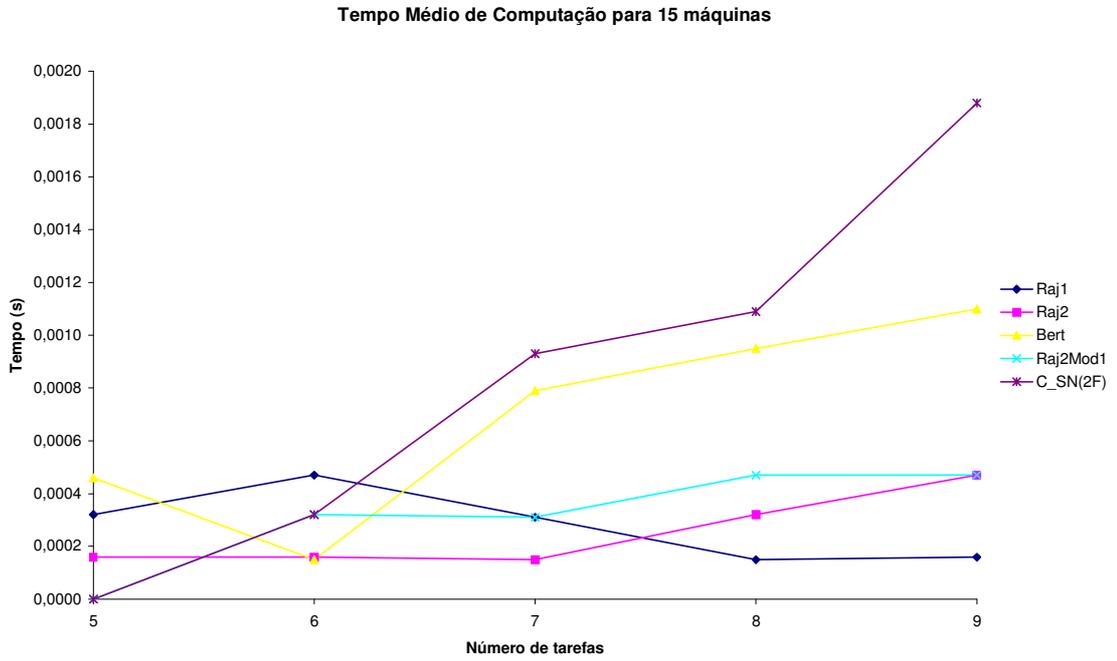


Figura 3.33– Tempo médio de computação vs. Número de tarefas, para 15 máquinas

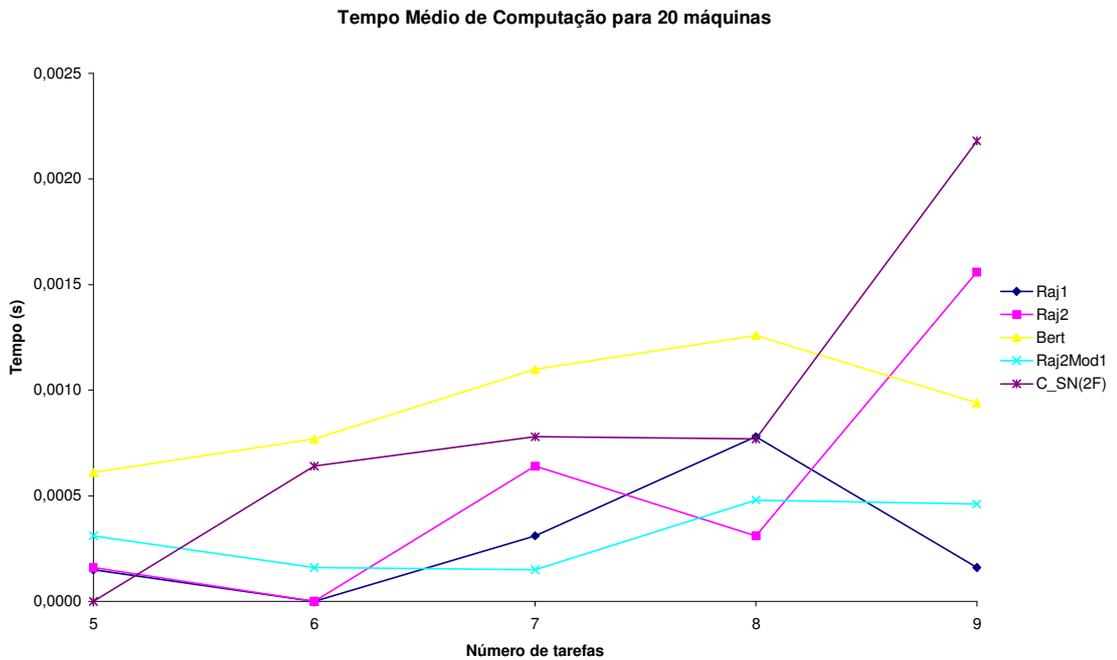


Figura 3.34– Tempo médio de computação vs. Número de tarefas, para 20 máquinas

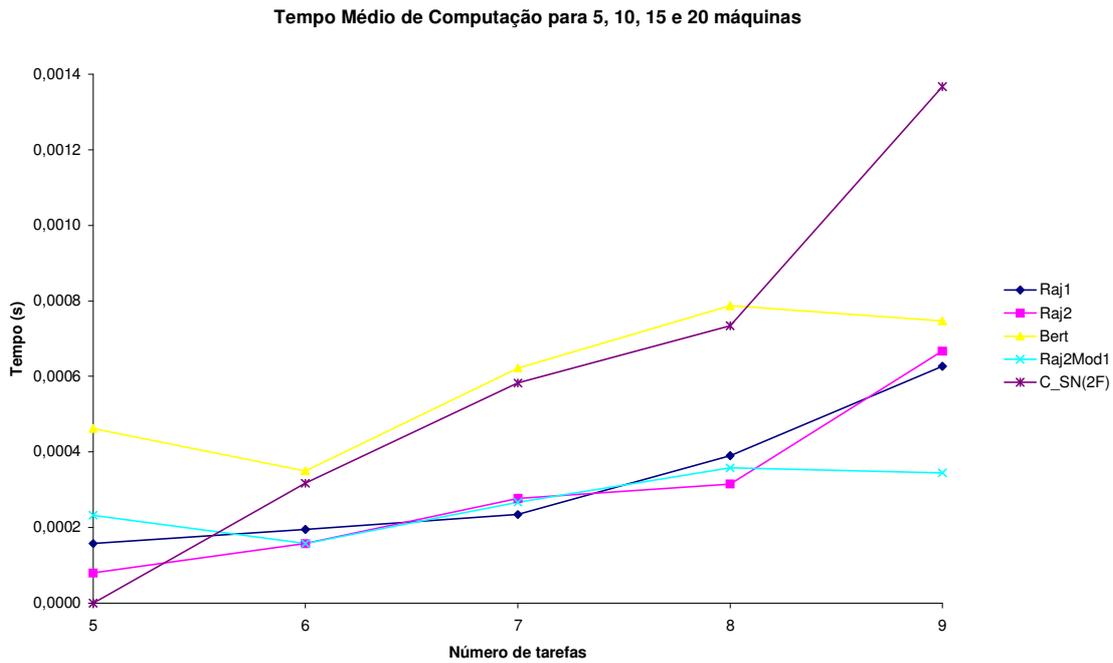


Figura 3.35– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Verifica-se, através das figuras 3.31 a 3.35, que para problemas de pequeno porte, o tempo de computação é muito baixo, não comprometendo a viabilidade das soluções apresentadas pelos algoritmos.

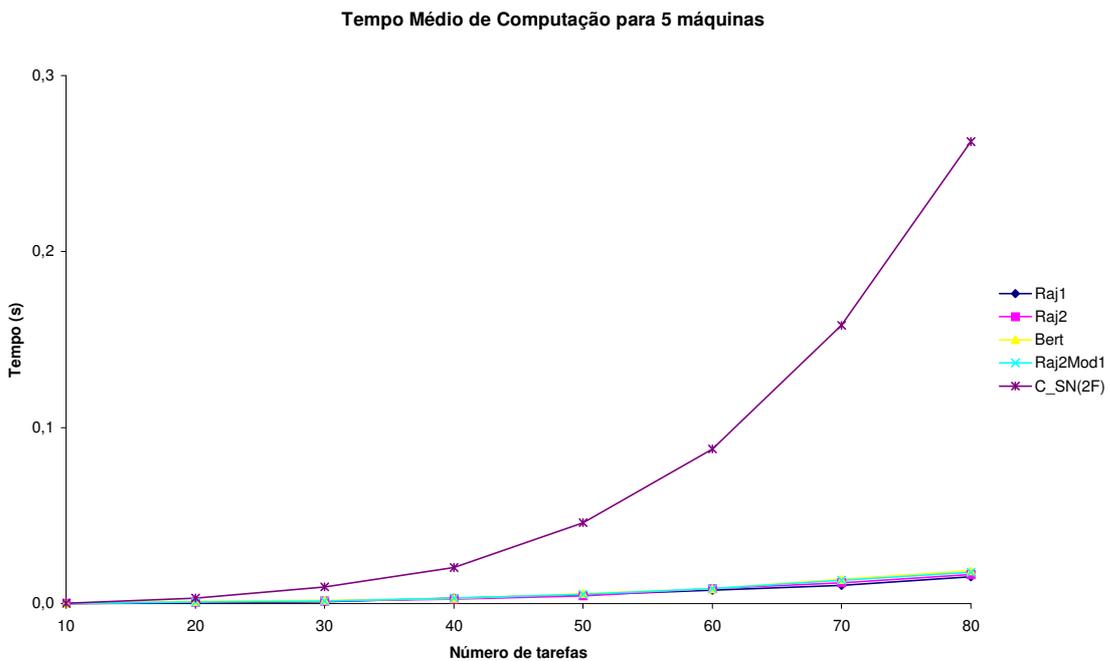


Figura 3.36– Tempo médio de computação vs. Número de tarefas, para 5 máquinas

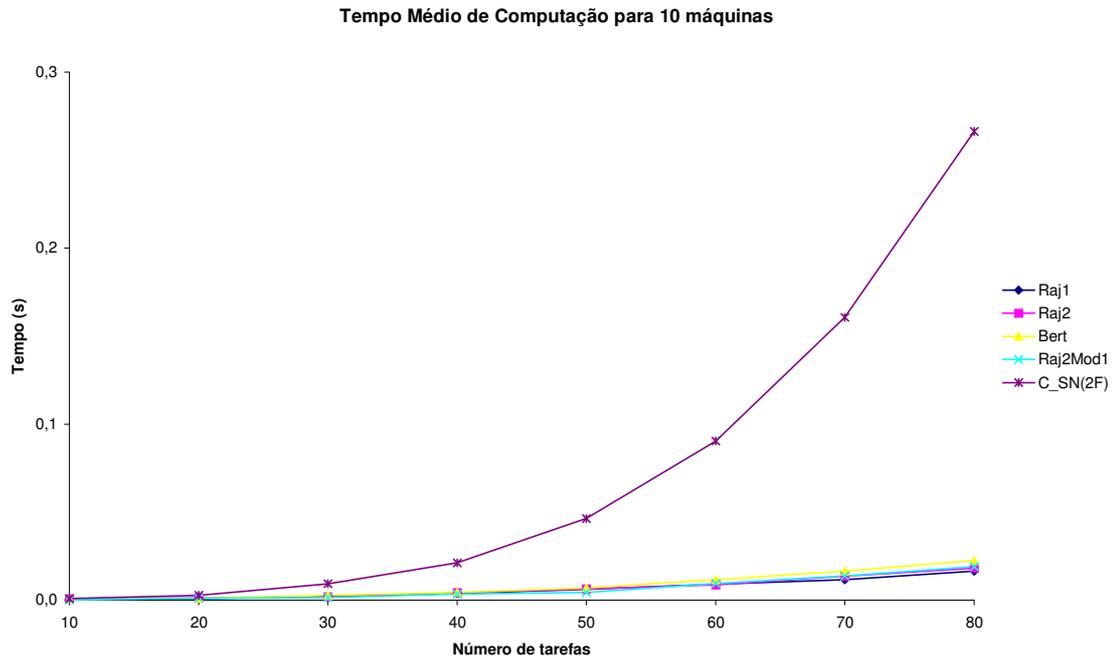


Figura 3.37– Tempo médio de computação vs. Número de tarefas, para 10 máquinas

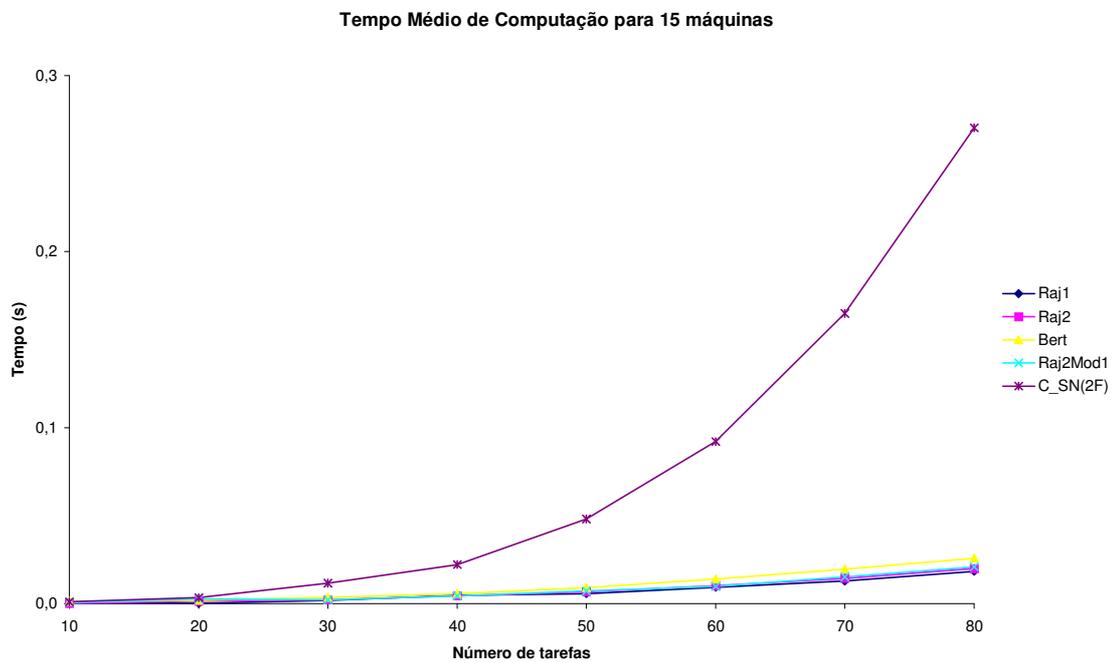


Figura 3.38– Tempo médio de computação vs. Número de tarefas, para 15 máquinas

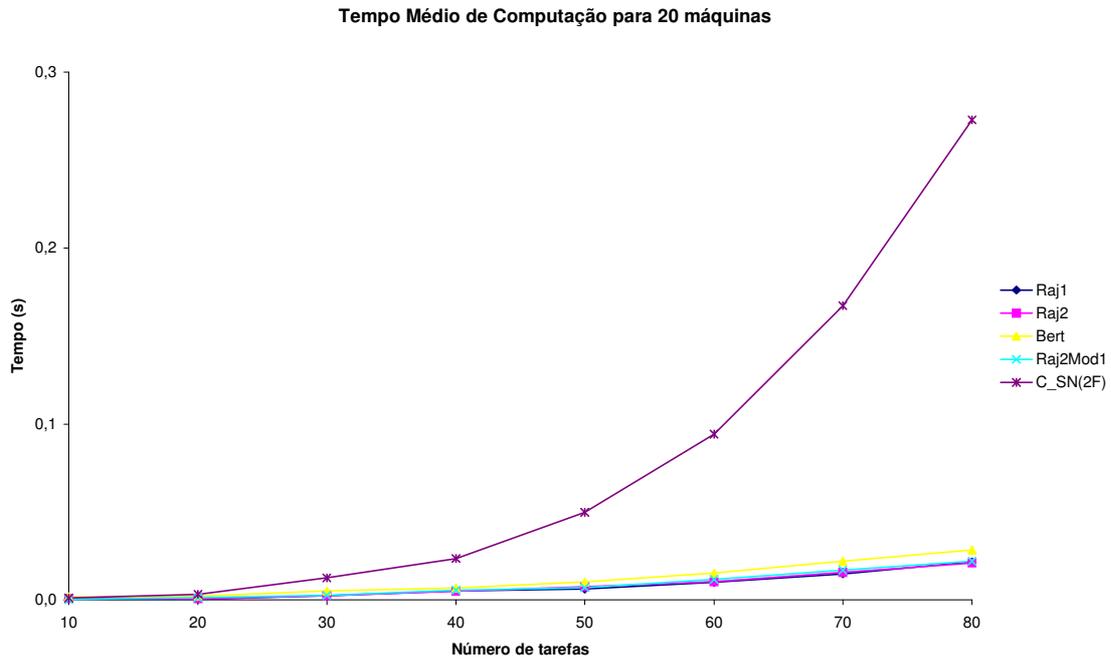


Figura 3.39– Tempo médio de computação vs. Número de tarefas, para 20 máquinas

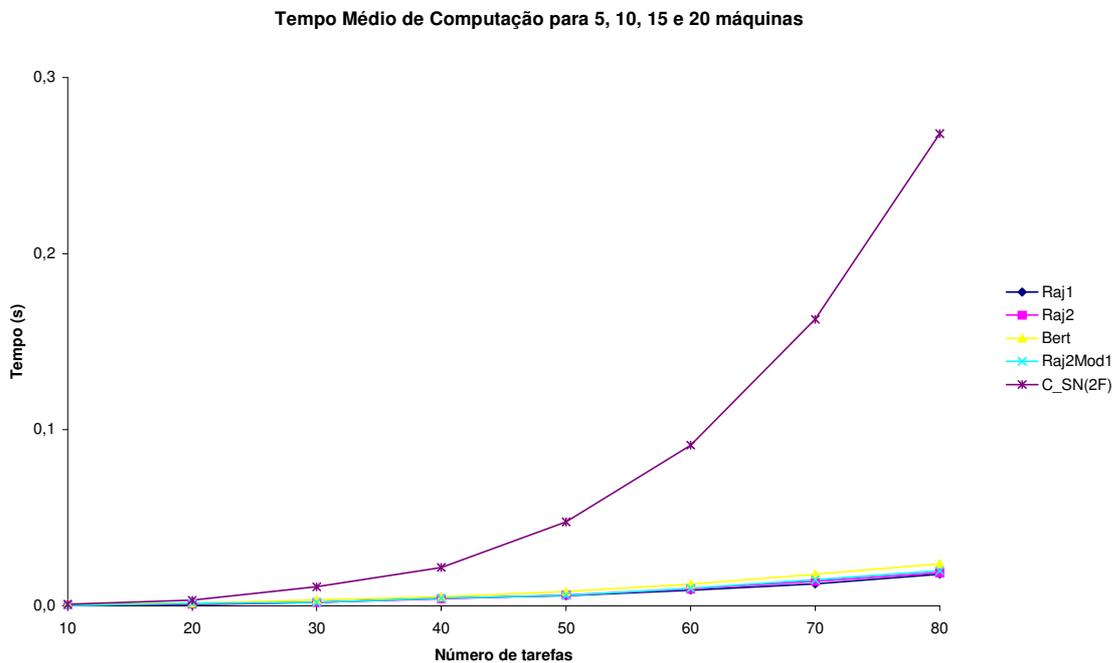


Figura 3.40– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Para os problemas de médio porte, constata-se de forma agrupada (figura 3.40), que o algoritmo C_SN(2F) demanda três vezes mais tempo de computação que os demais algoritmos. Porém, o método não consome mais do que 0,3 segundos em nenhum caso.

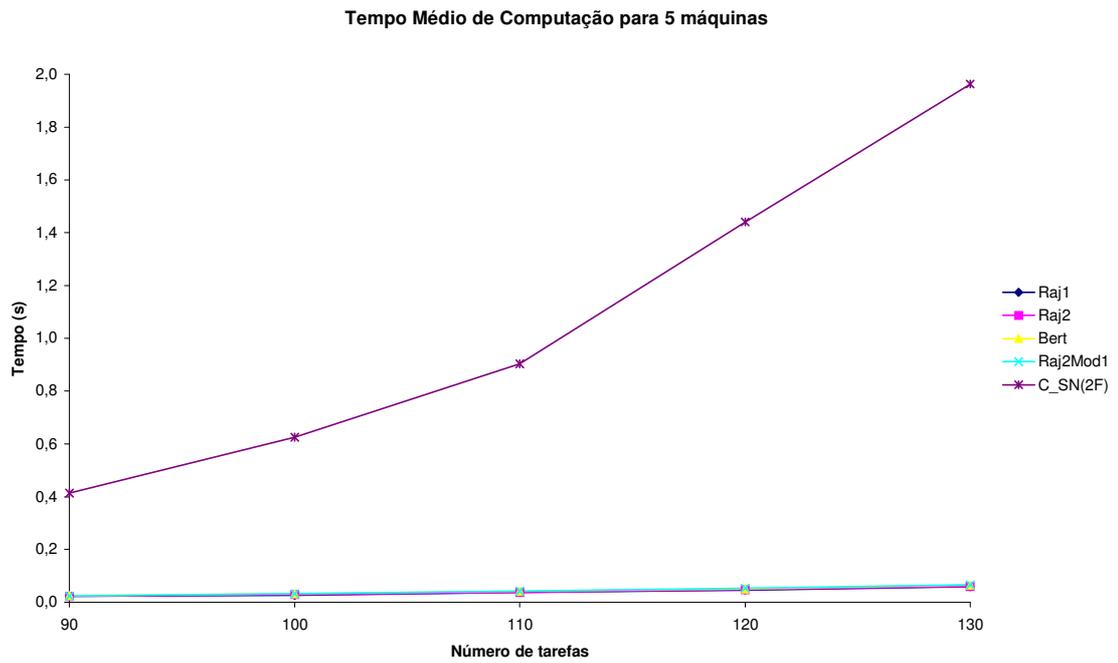


Figura 3.41– Tempo médio de computação vs. Número de tarefas, para 5 máquinas

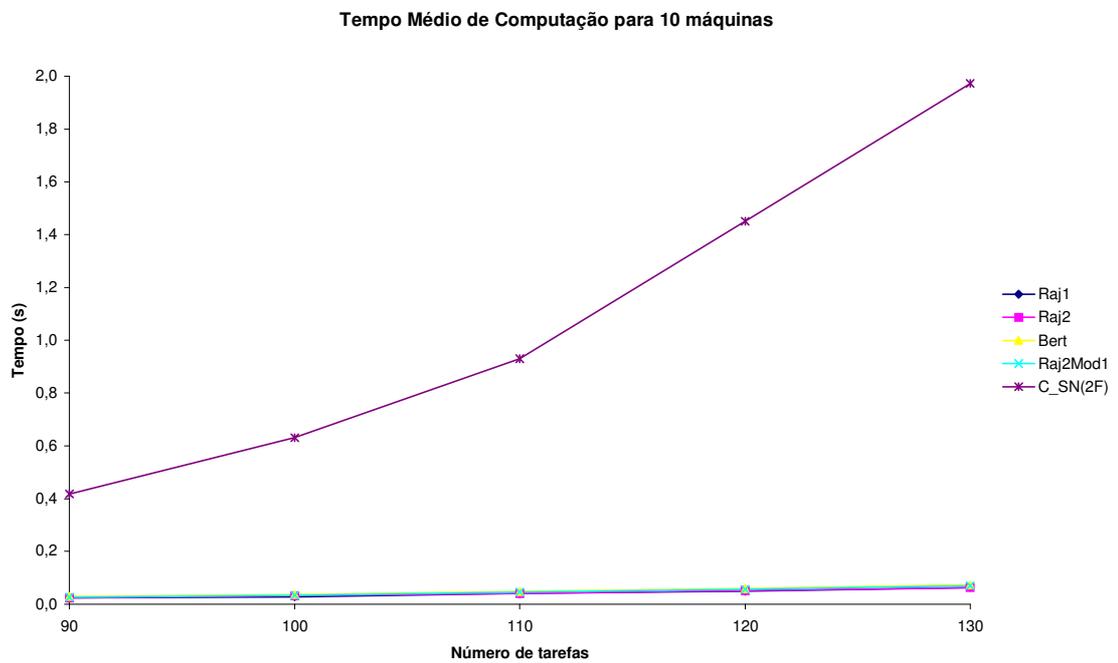


Figura 3.42– Tempo médio de computação vs. Número de tarefas, para 10 máquinas

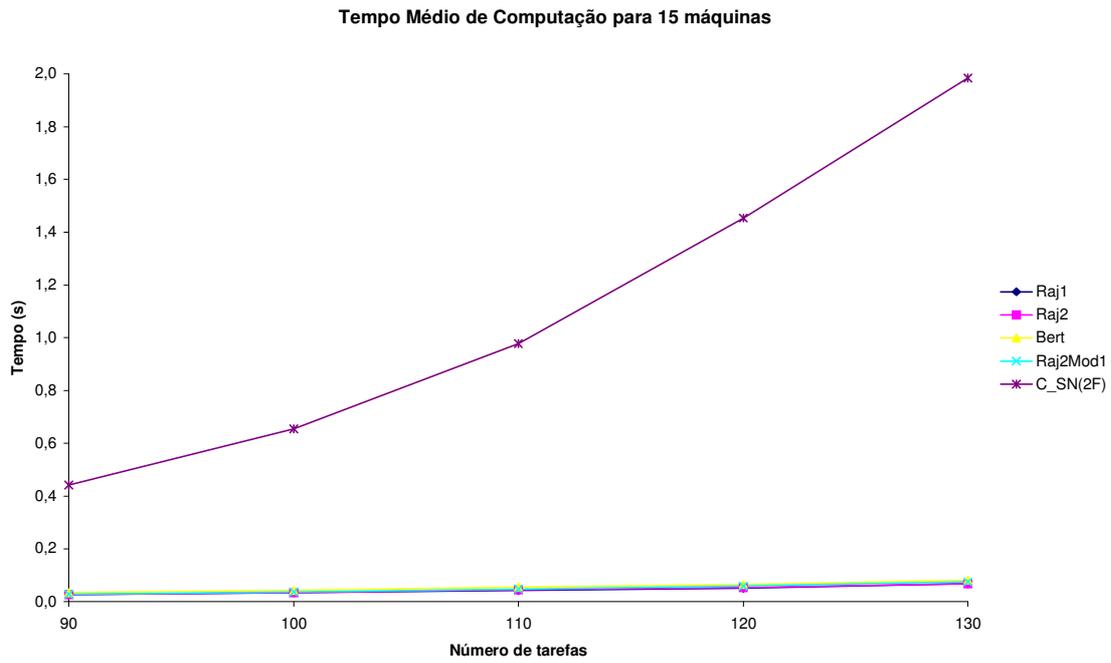


Figura 3.43– Tempo médio de computação vs. Número de tarefas, para 15 máquinas

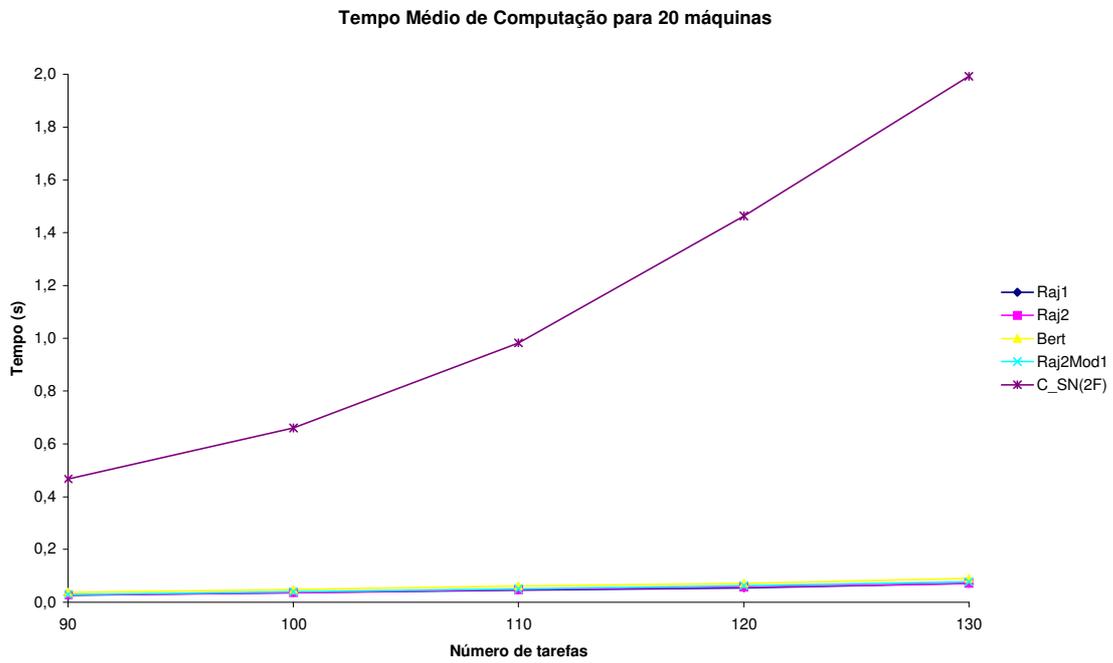


Figura 3.44– Tempo médio de computação vs. Número de tarefas, para 20 máquinas

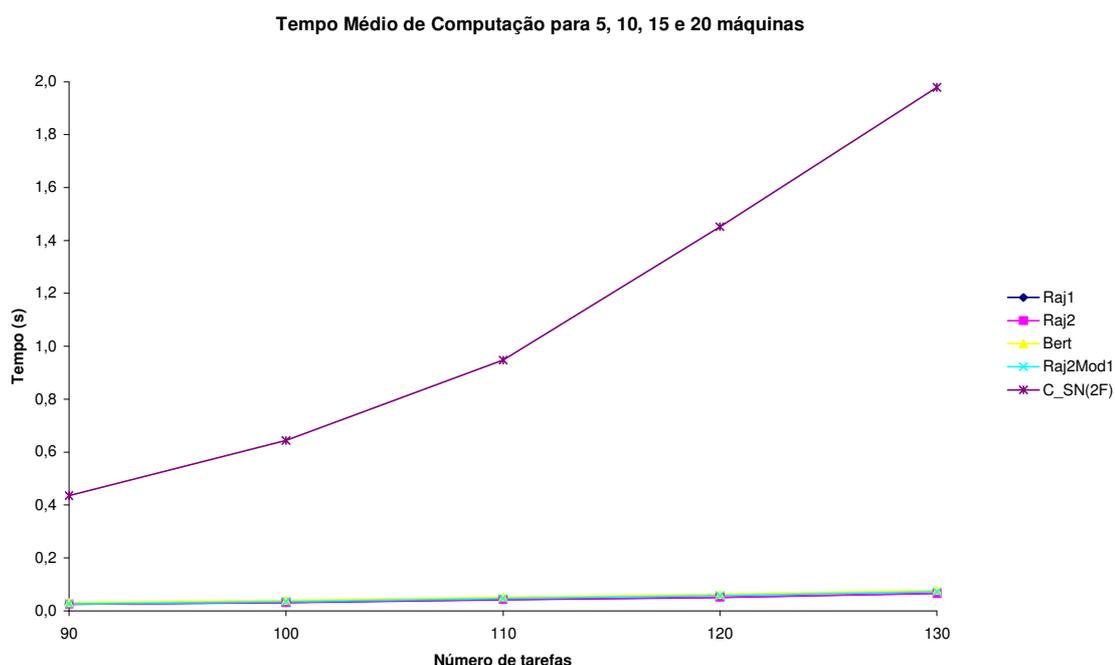


Figura 3.45– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

O algoritmo C_SN(2F) nos problemas de grande porte, assim como nos problemas de médio porte, tem uma característica particular em relação aos demais: quanto maior o número de tarefas, maior o tempo de computação médio. Para problemas com 130 tarefas o tempo de processamento aproxima-se de 2 segundos.

De forma geral, entre todos os métodos avaliados nesta experimentação, verifica-se que o método proposto C_SN(2F) é o que apresenta melhores resultados tanto em termos de porcentagem de sucesso quanto em termos de porcentagem de desvio médio relativo.

Percebe-se também que o tempo de computação médio do método proposto C_SN(2F) cresce exponencialmente em relação ao número de tarefas. Porém, para 130 tarefas, o método consome cerca de apenas 2 segundos por problema, não inviabilizando o método proposto.

No capítulo seguinte será apresentada a experimentação computacional (Parte B), que conterà a avaliação do melhor método heurístico composto de três fases e os métodos heurísticos modificados sugeridos no capítulo 1, todos também de três fases.

CAPÍTULO 4

Experimentação Computacional – Parte B

A base de dados e a metodologia de análise são as mesmas utilizadas na experimentação computacional (Parte A) Porém, neste capítulo são analisados os resultados gerados pelos algoritmos modificados Raj2Mod2, Raj2Mod3 e PH1(p)Mod e o algoritmo PH1(p), todos métodos heurísticos compostos de 3 fases.

4.1 Análise dos Resultados

Os resultados obtidos também são analisados em três partes e na seguinte ordem: Porcentagem de Sucesso; Porcentagem de Desvio Médio Relativo e Tempo Médio de Computação

4.1.1 Análise da Porcentagem de Sucesso

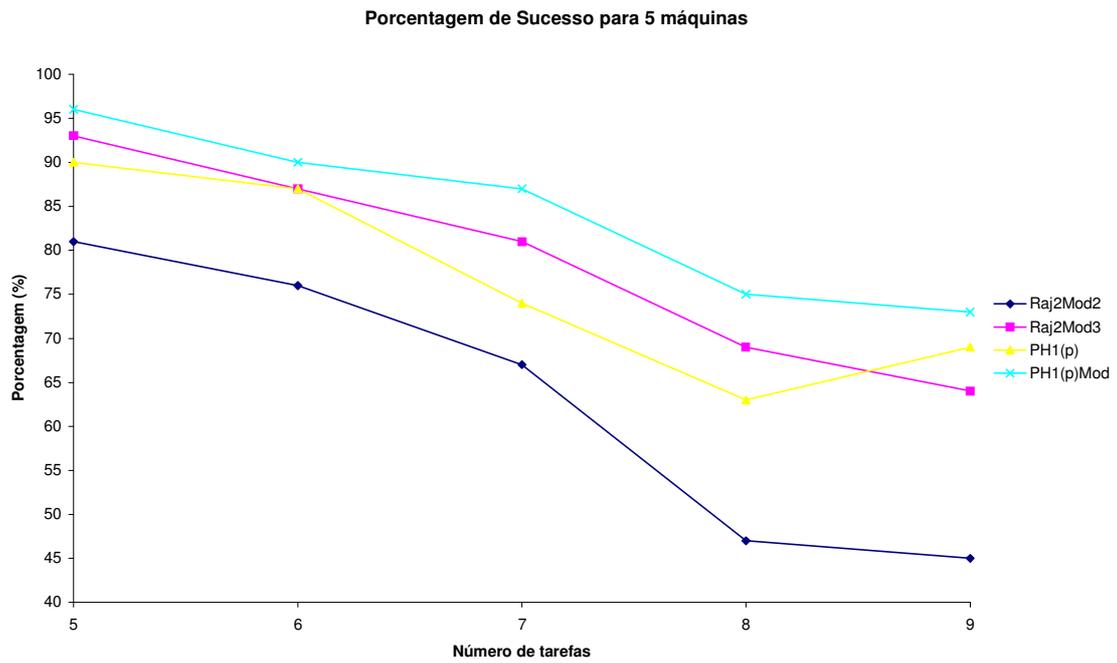


Figura 4.1– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

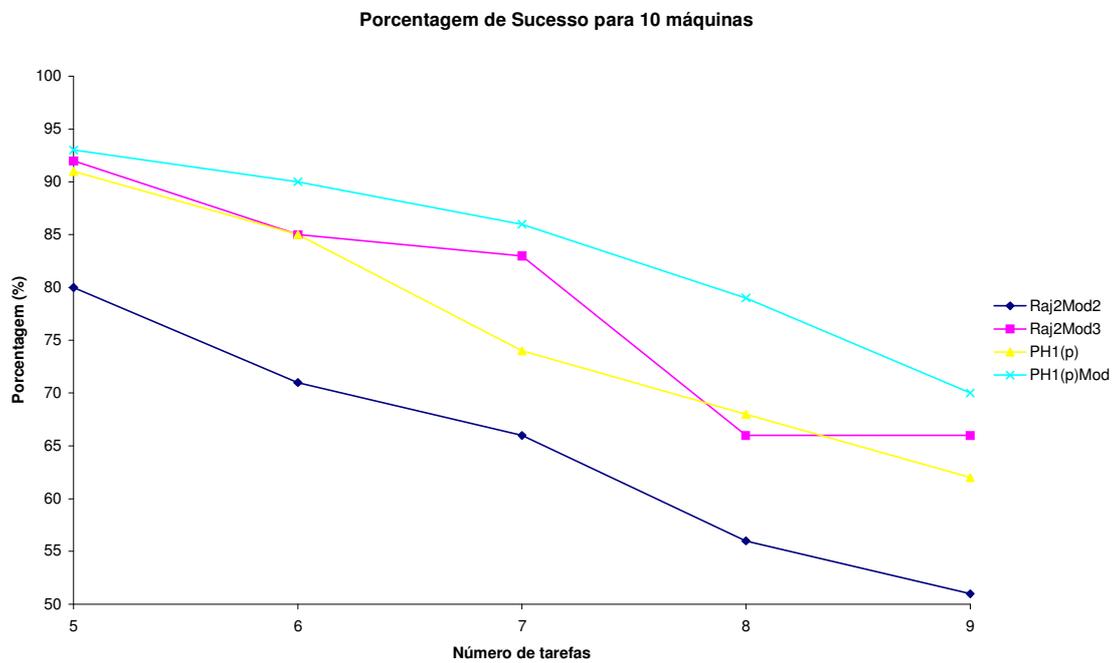


Figura 4.2– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

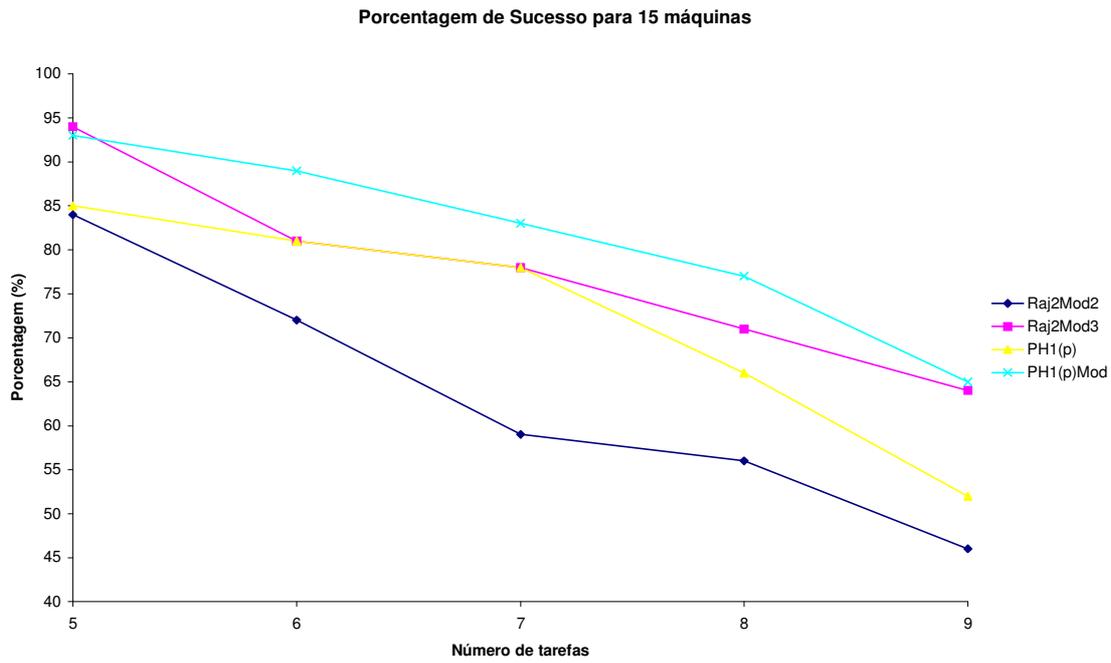


Figura 4.3– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

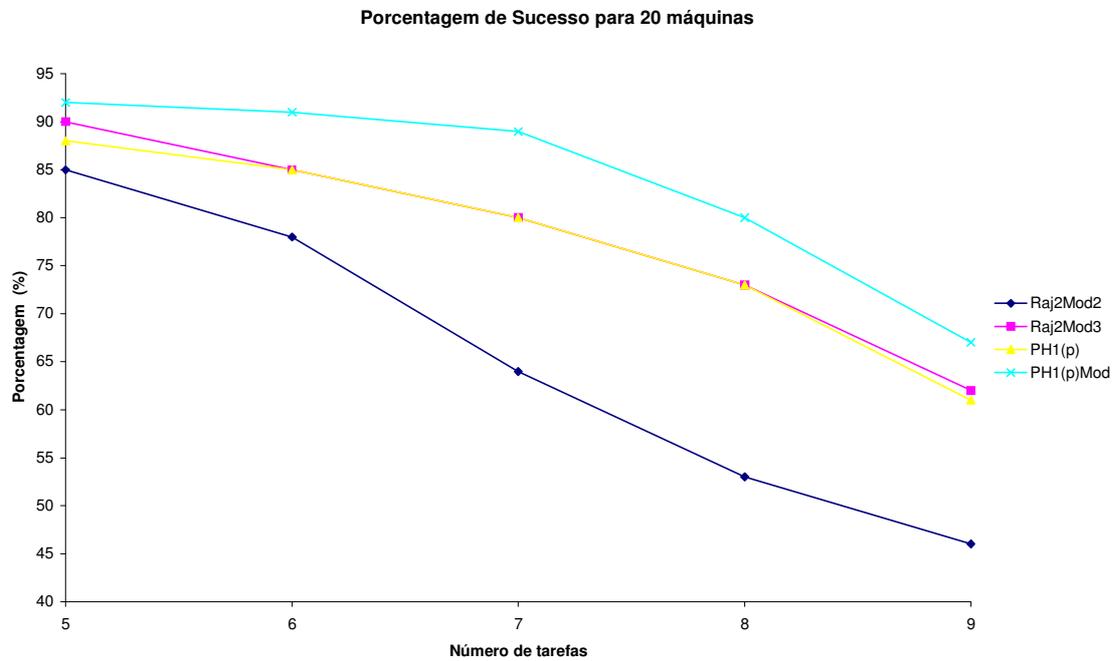


Figura 4.4– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

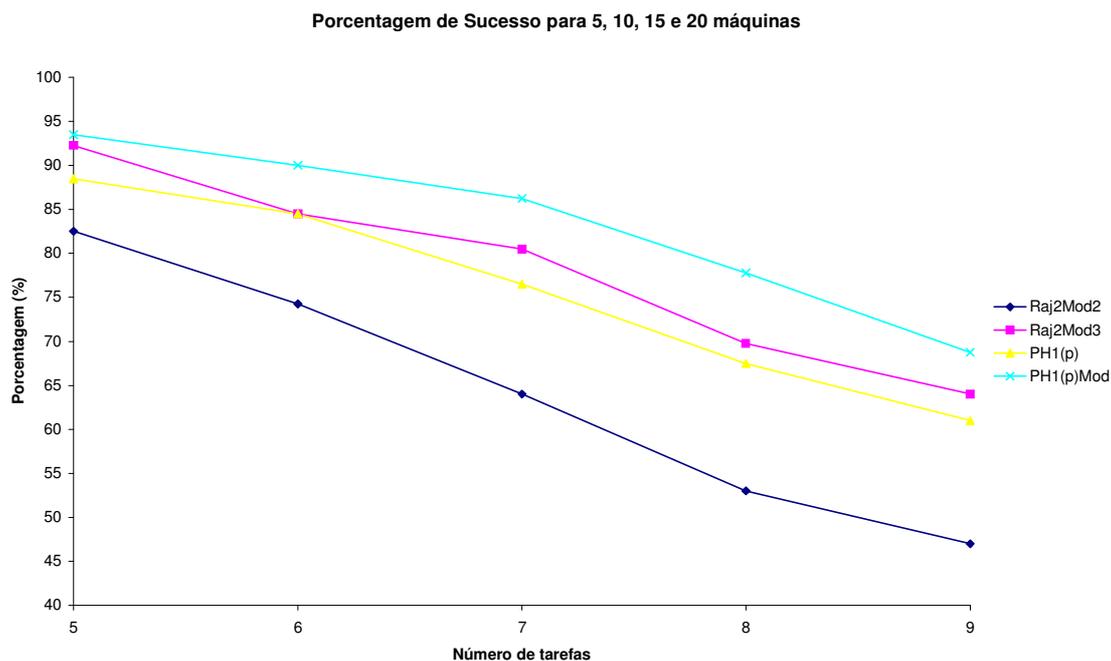


Figura 4.5– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Percebe-se que, pelos fatos de ocorrerem menos empates e de diminuir as possibilidades de se alcançar o resultado ótimo, a porcentagem de sucesso é menor para cada método com 9 tarefas, em relação aos problemas de 5 tarefas. A figura 4.5 mostra que para 5 tarefas, os algoritmos apresentam soluções com porcentagem de sucesso entre 80% e 95%. Já para 9 tarefas a porcentagem de sucesso dos quatro algoritmos está entre 45% e 75%.

Analisando-se as figuras 4.1 a 4.5, verifica-se que a porcentagem de sucesso do algoritmo adaptado PH1(p)Mod é superior aos outros algoritmos, na maioria dos casos de variação de 5 a 9 tarefas e 5 a 20 máquinas.

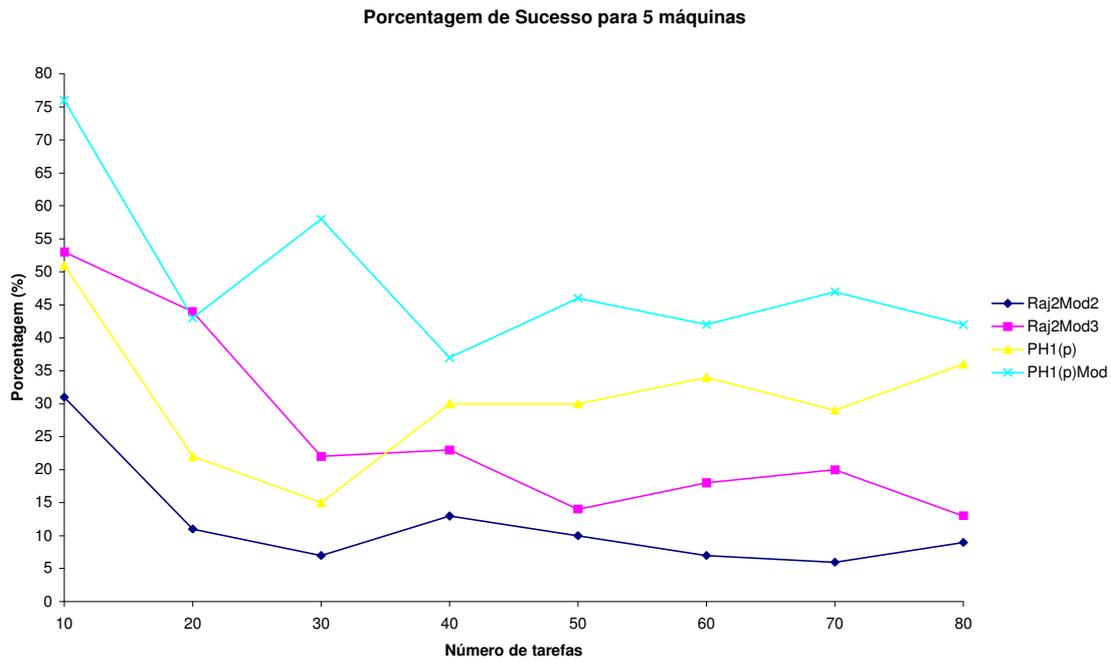


Figura 4.6– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

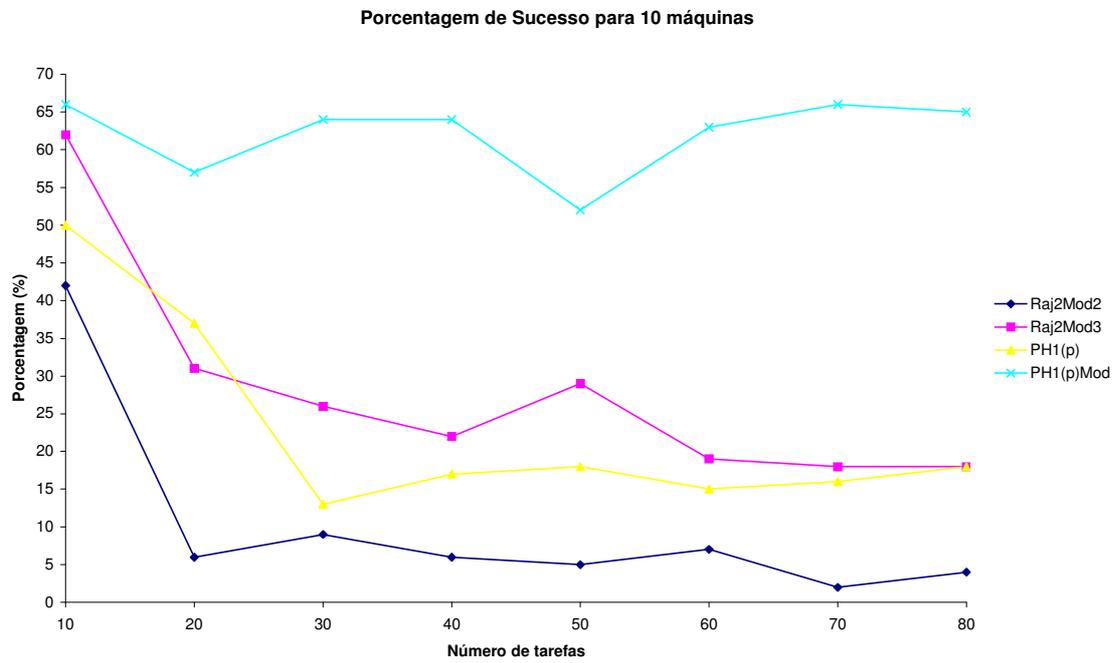


Figura 4.7– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

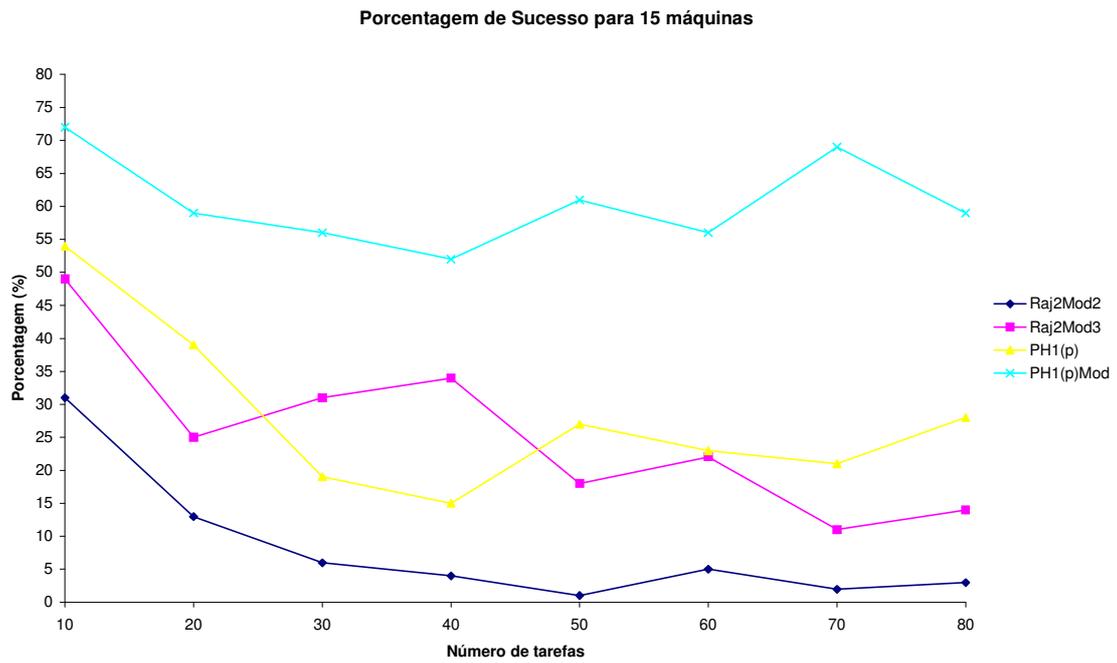


Figura 4.8– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

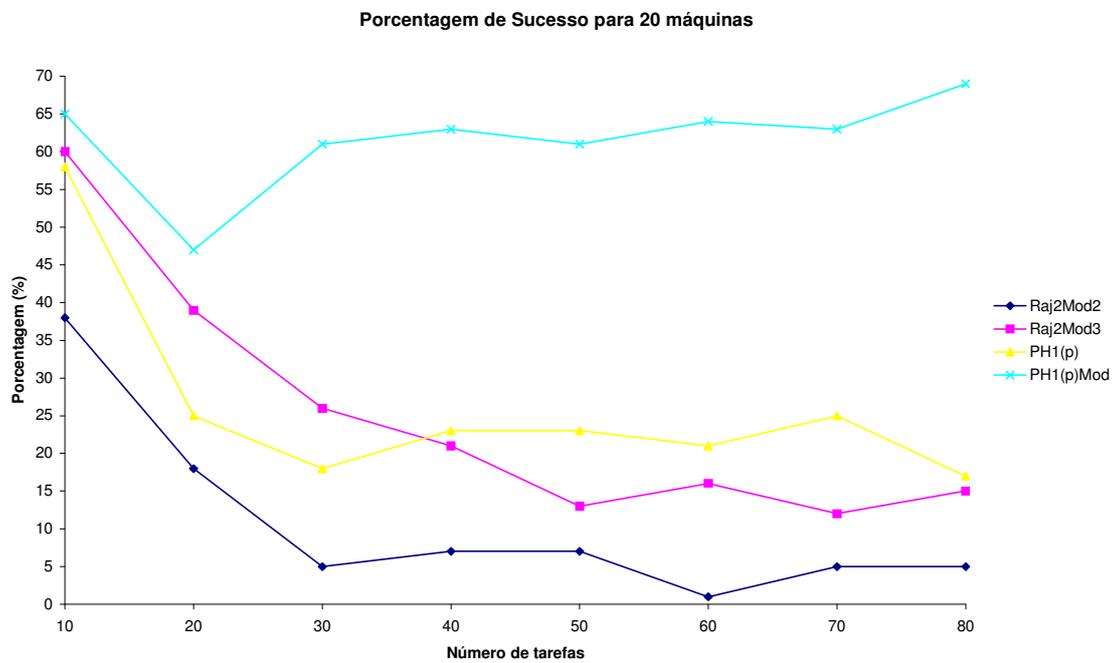


Figura 4.9– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

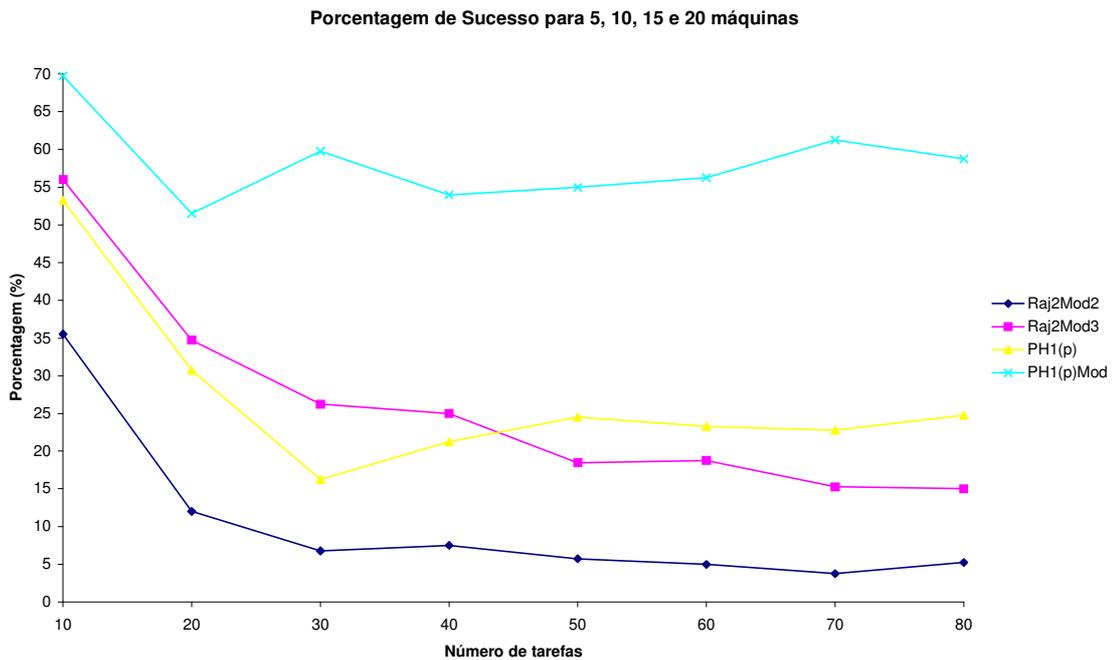


Figura 4.10– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Nos problemas com mais de 20 tarefas, considerados de médio porte, o algoritmo PH1(p)Mod consolida sua superioridade em relação aos demais, atingindo de forma agrupada porcentagem de sucesso entre 50% e 65%, enquanto os demais algoritmos apresentam no máximo 35% de porcentagem de sucesso.

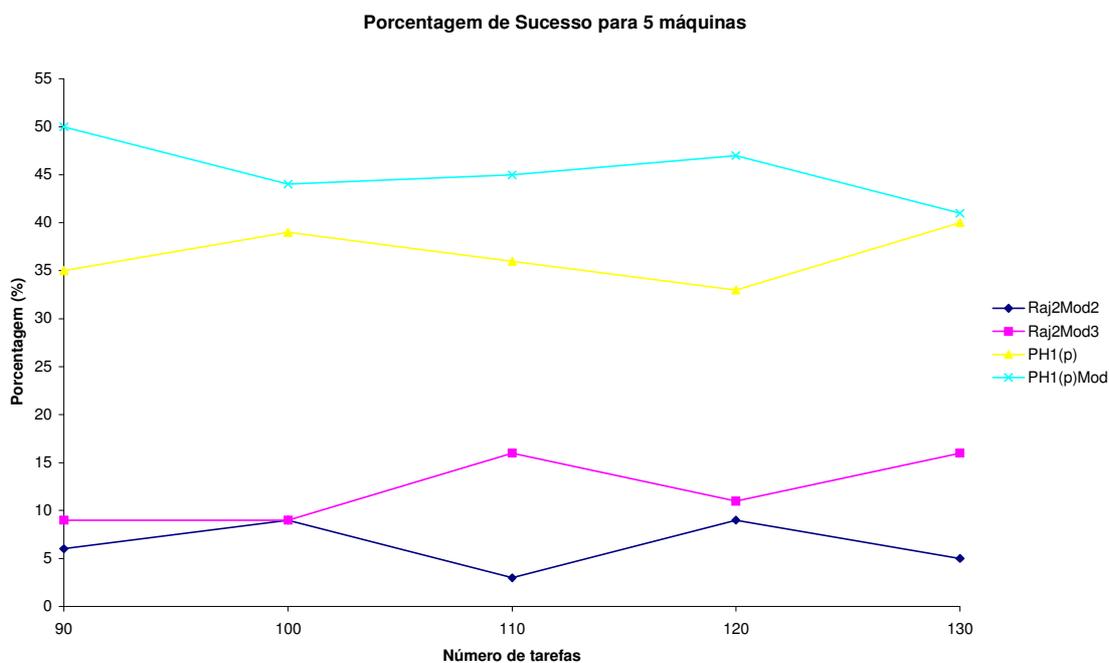


Figura 4.11– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

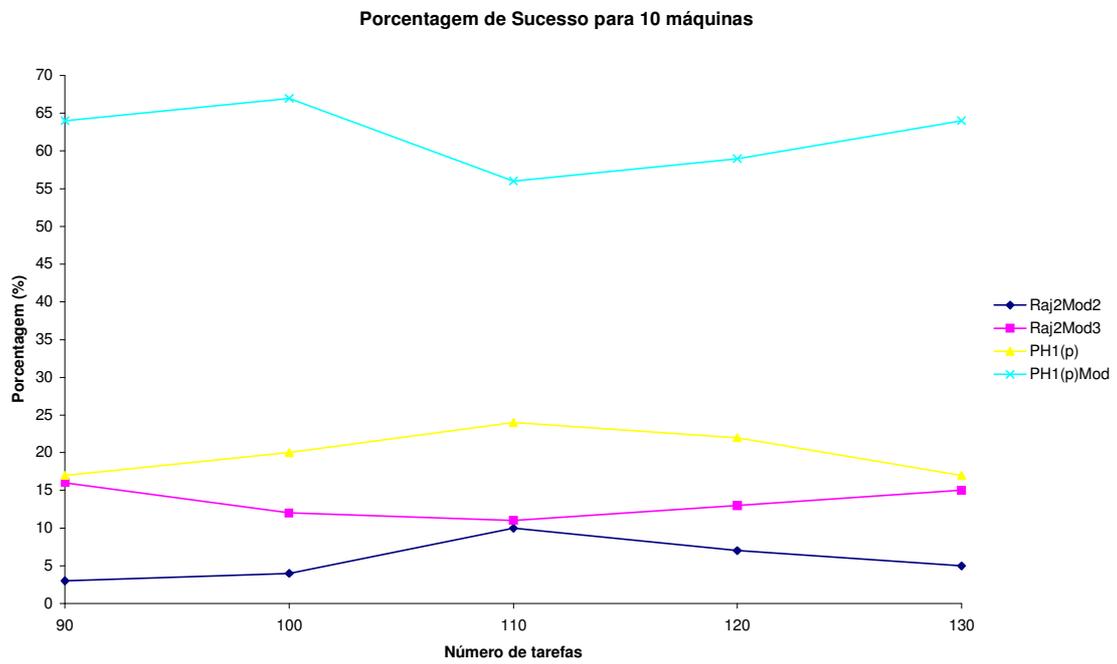


Figura 4.12– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

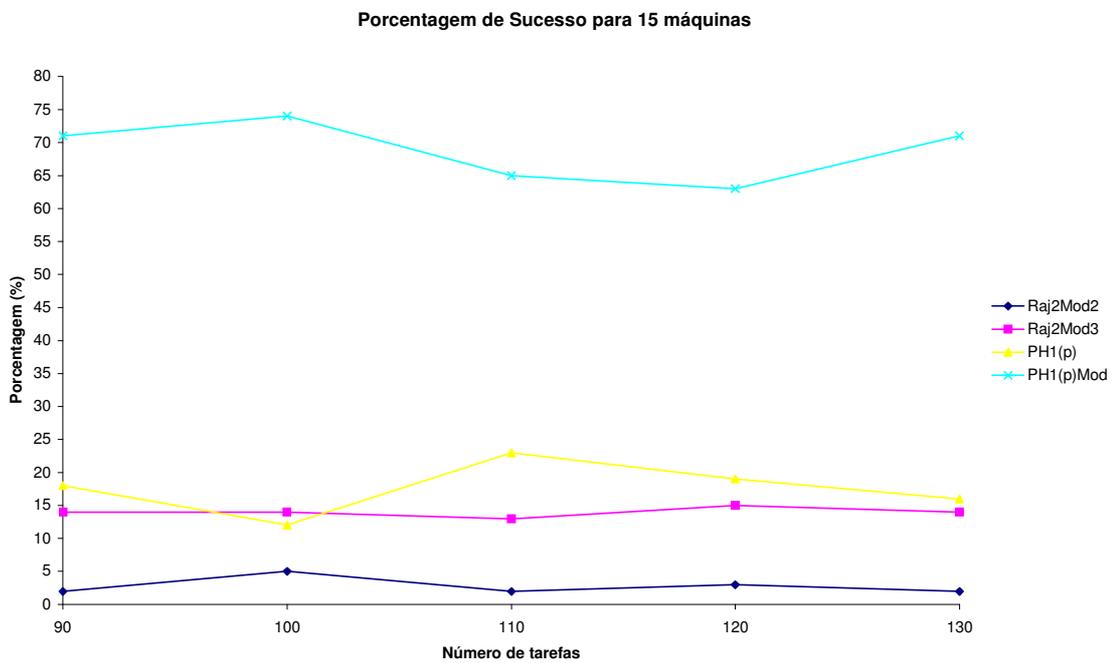


Figura 4.13– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

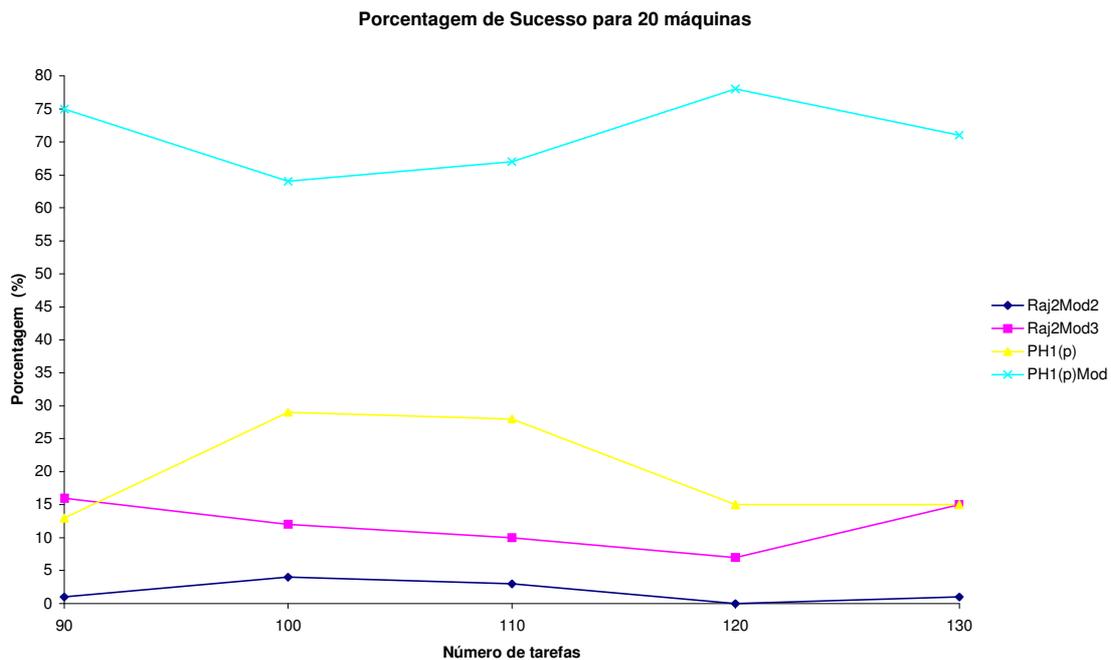


Figura 4.14– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

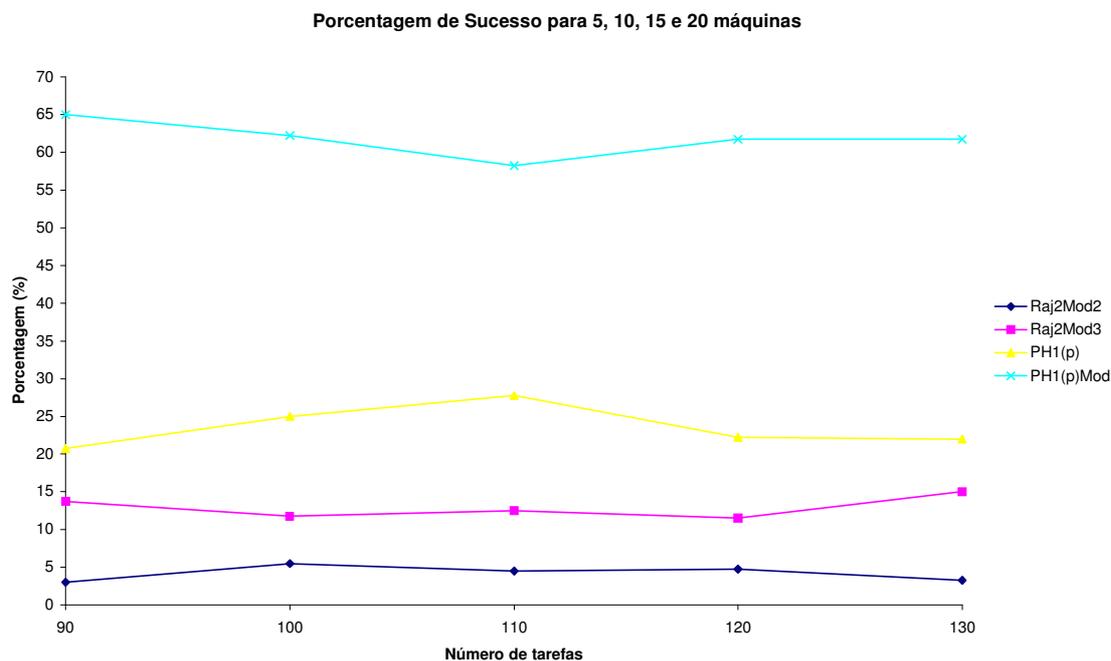


Figura 4.15– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Nos problemas de grande porte, verifica-se que o algoritmo modificado PH1(p)Mod e sua versão original, o algoritmo PH1(p), são os que apresentam maior percentagem de sucesso. O algoritmo PH1(p)Mod apresenta de forma agrupada, percentagem de sucesso entre 55% e 65%, enquanto o segundo melhor tem percentagem de sucesso entre 20% e 30% conforme ilustra a figura 4.15.

Outra observação refere-se ao fato de que nas figuras 4.11 a 4.14 evidencia-se que a porcentagem de sucesso do algoritmo PH1(p)Mod aumenta quanto maior o número de máquinas envolvidas nos problemas.

4.1.2 Análise da Porcentagem do Desvio Médio Relativo

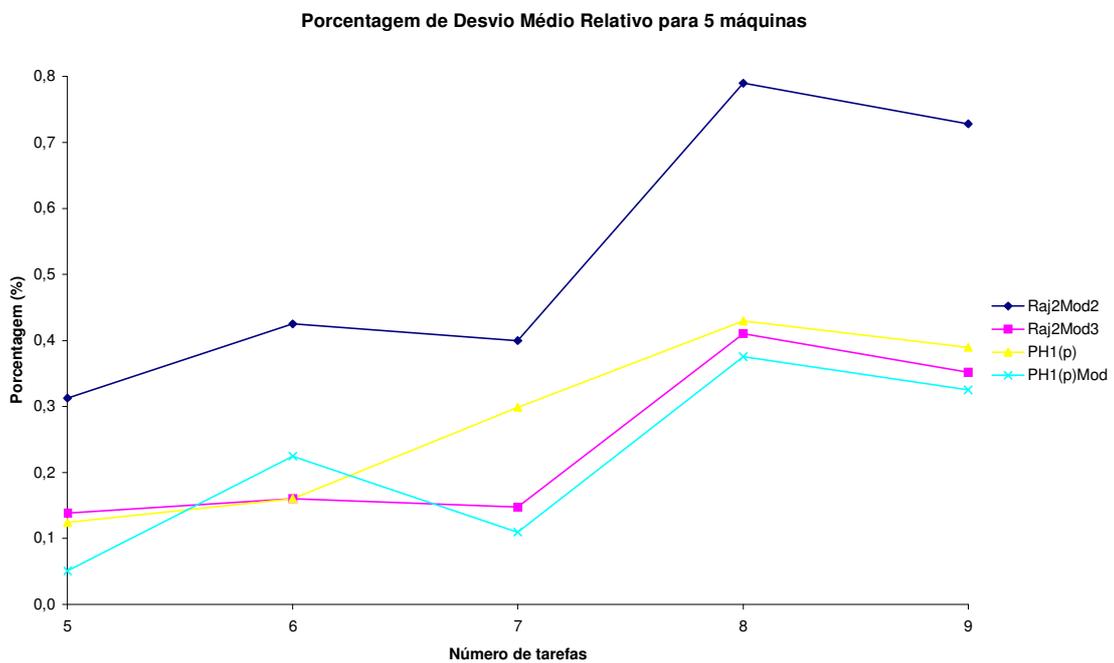


Figura 4.16– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas.

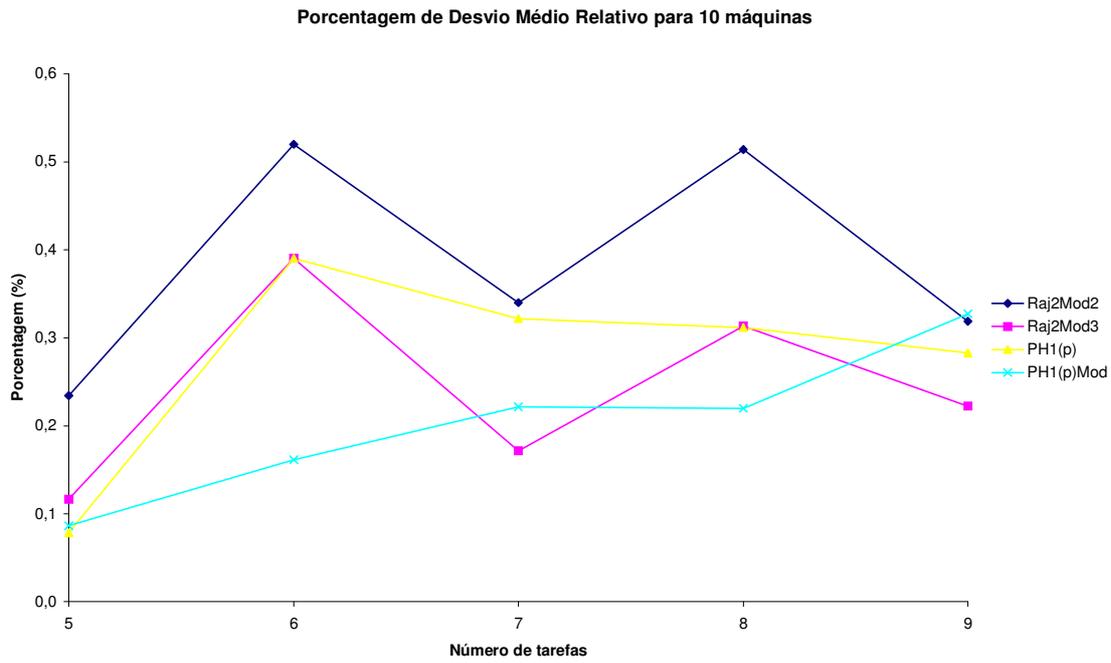


Figura 4.17– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas.

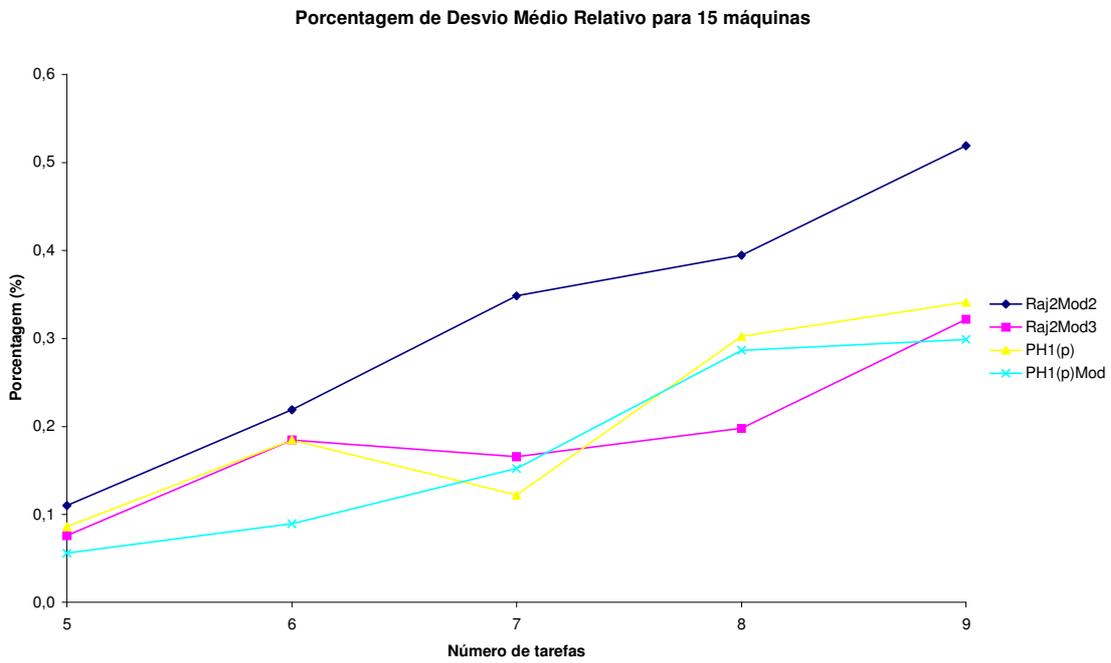


Figura 4.18– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas.

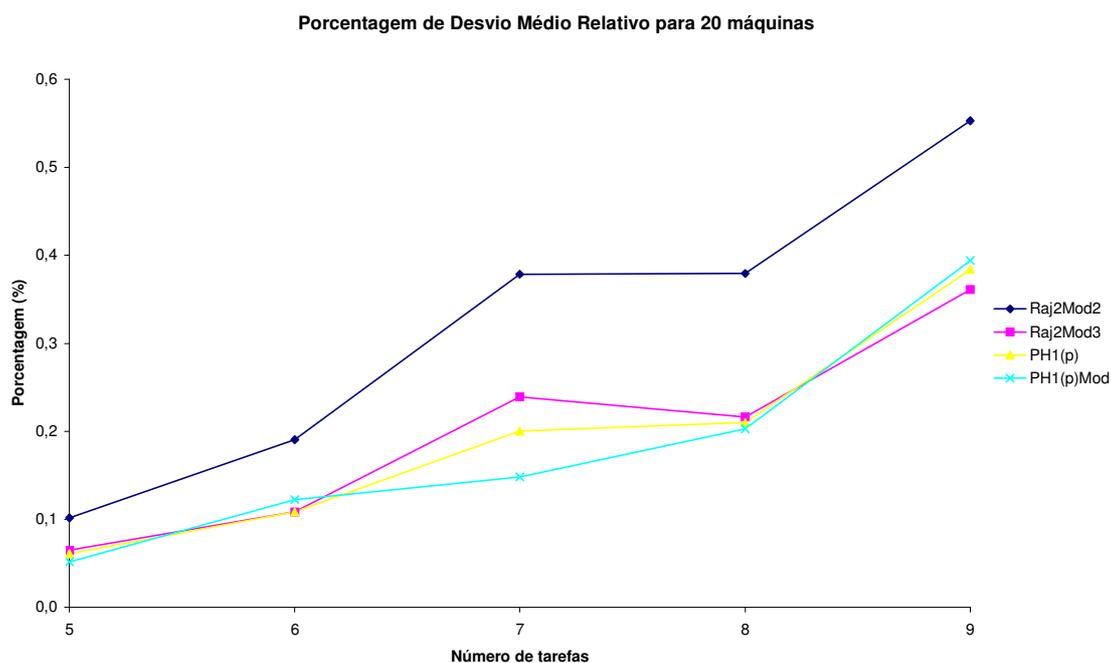


Figura 4.19– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas.

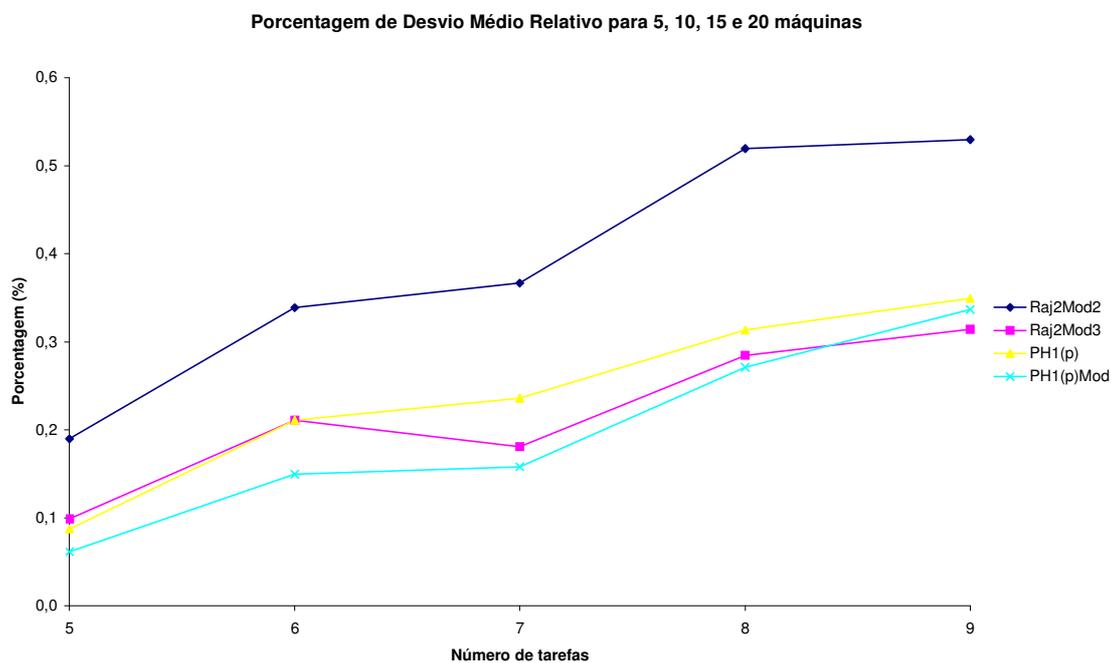


Figura 4.20– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

As figuras 4.16 a 4.20, confirmam que para problemas de pequeno porte, apesar do método PH1(p)Mod apresentar soluções com porcentagem de sucesso um pouco superior aos demais, a porcentagem de desvio médio relativo é muito próxima dos outros dois melhores métodos.

Além disso, para problemas de 9 tarefas, o algoritmo Raj2Mod3 apresenta menor porcentagem de desvio médio relativo, mesmo tendo a segunda melhor marca em termos de porcentagem de sucesso, como observado anteriormente na figura 4.4.

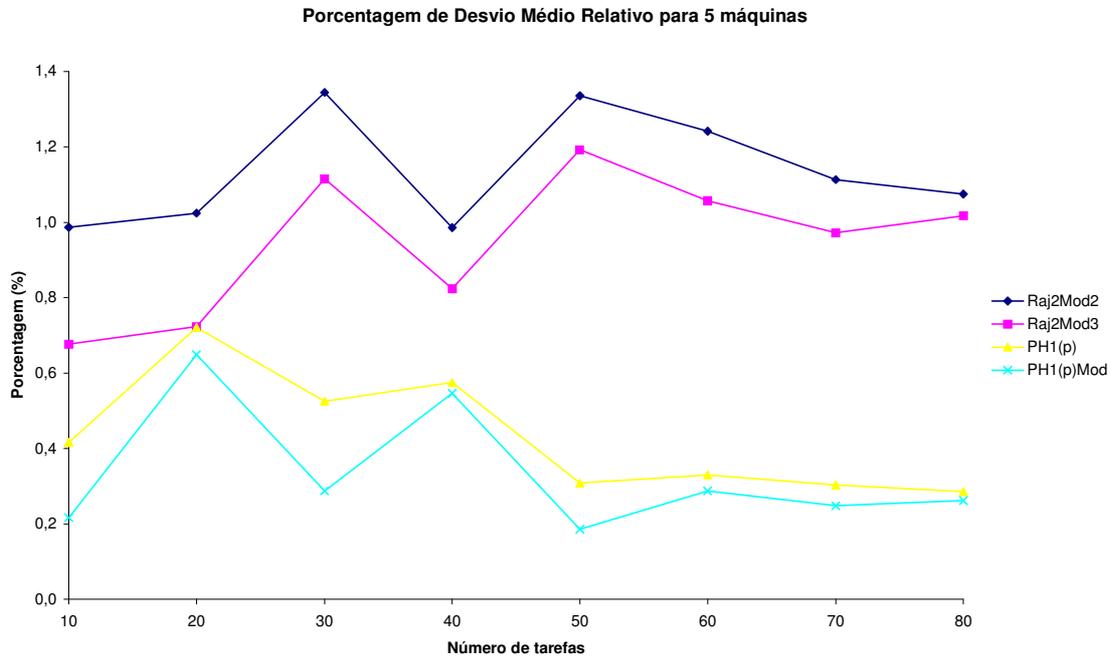


Figura 4.21– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas.

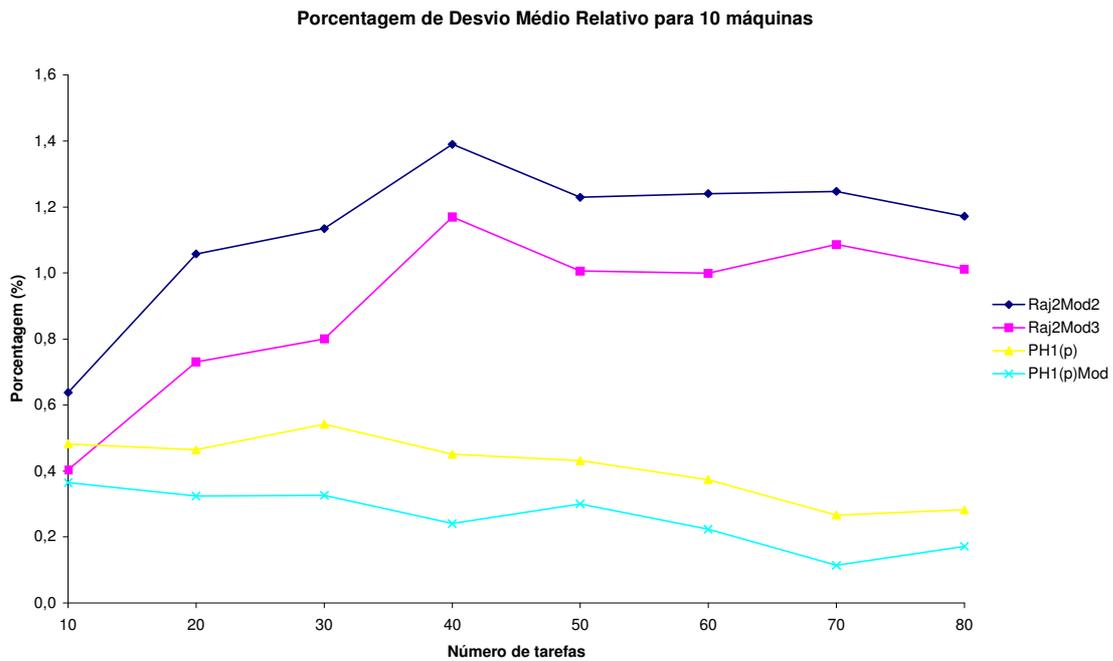


Figura 4.22– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas.

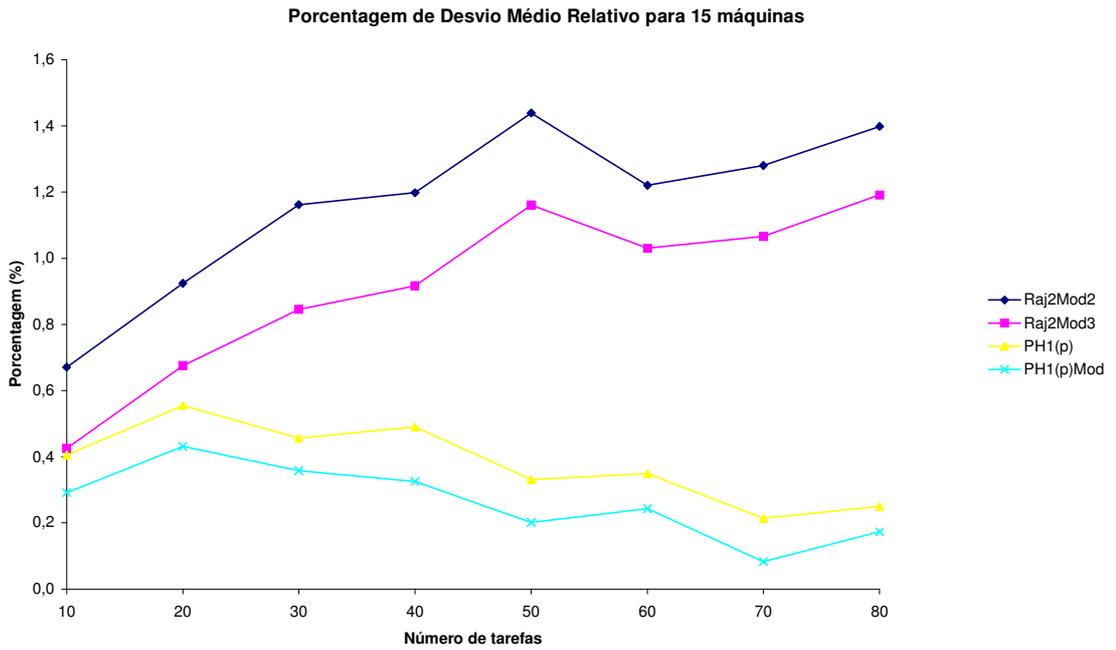


Figura 4.23– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas.

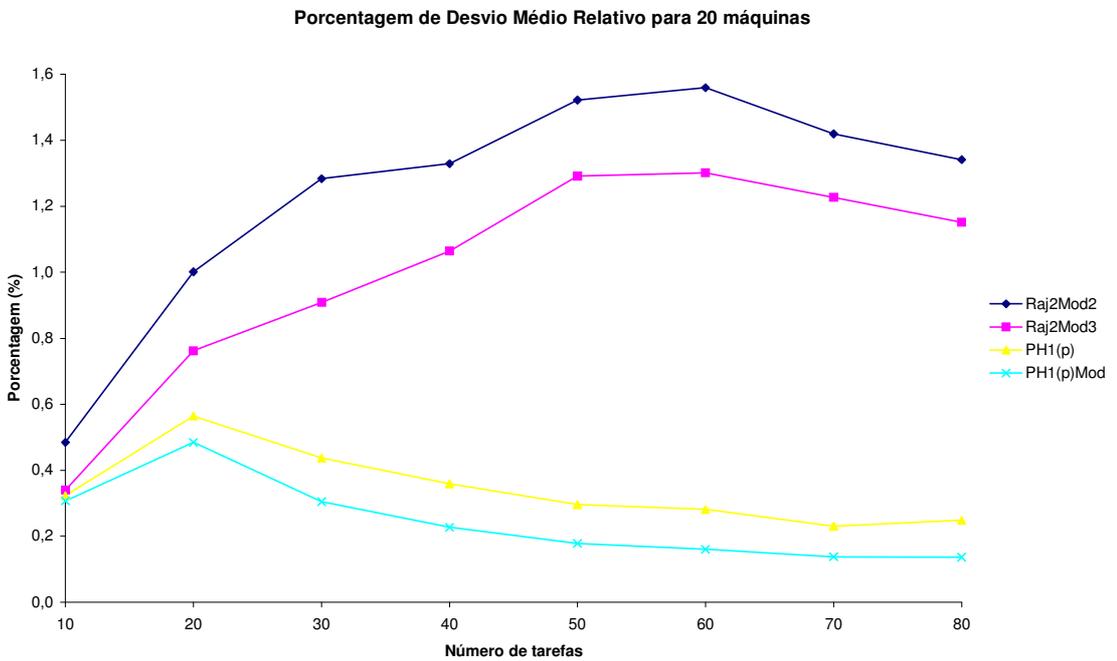


Figura 4.24– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas.

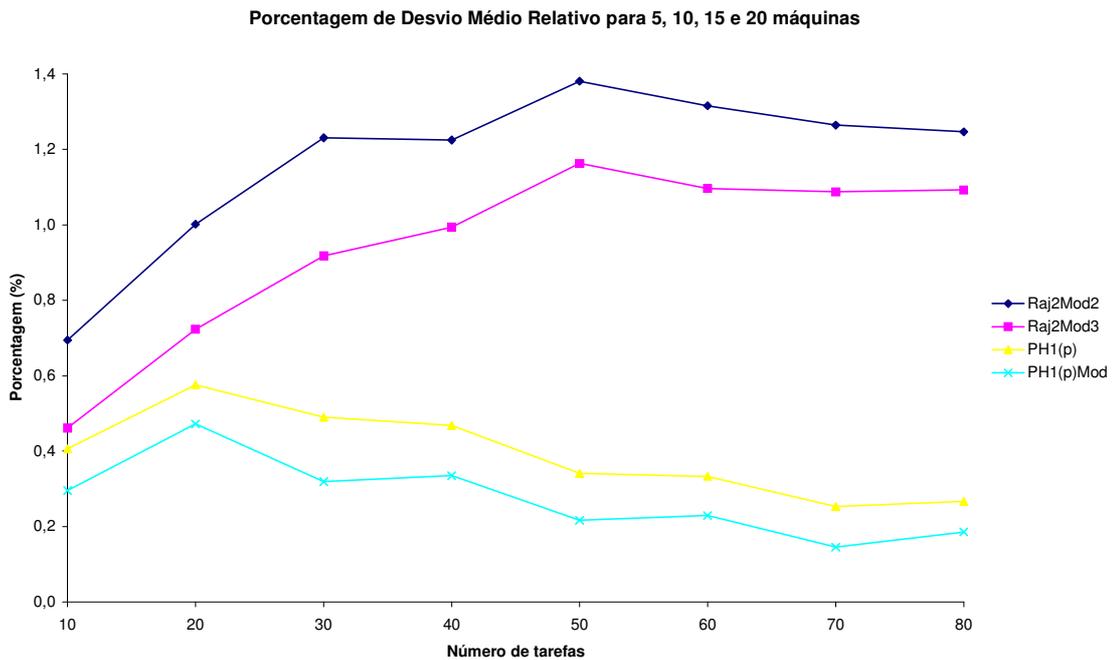


Figura 4.25– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

A figura 4.25 mostra de forma agrupada, para problemas de médio porte, que o algoritmo PH1(p)Mod apresenta a menor porcentagem de desvio médio relativo, confirmando os melhores resultados em termos de porcentagem de sucesso.

É uma porcentagem de desvio médio relativo abaixo de 0,5%, seguido de perto pelo segundo melhor método, o algoritmo PH1(p). Os outros dois algoritmos apresentam porcentagens de desvio médio relativo maiores, entre de 0,8% e 1,4% para problemas acima de 30 tarefas.

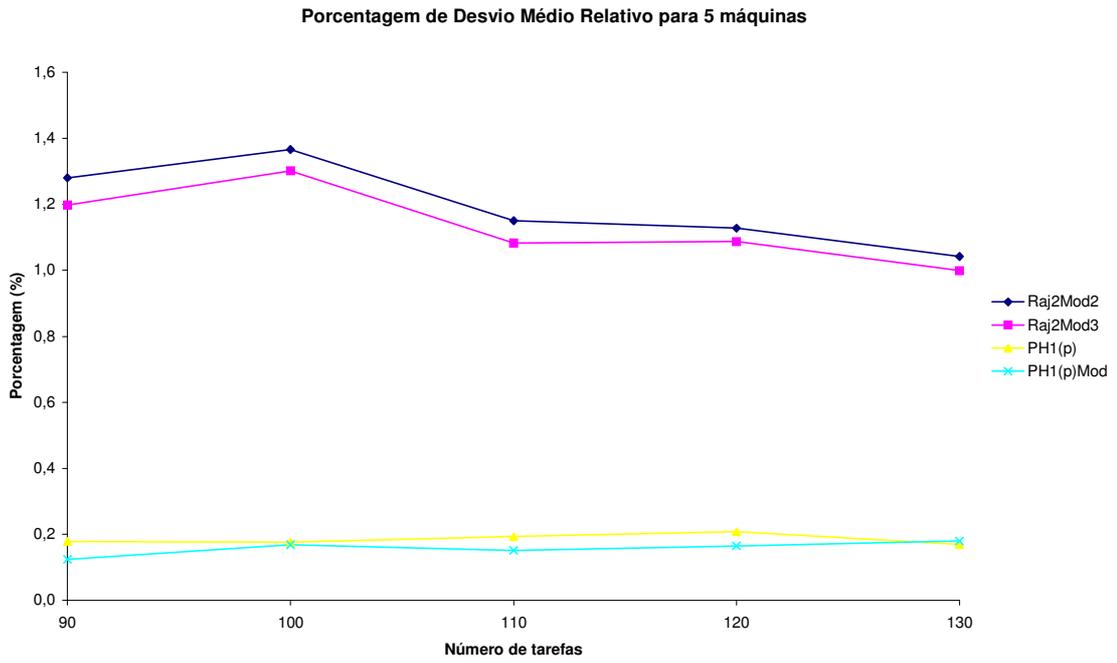


Figura 4.26– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas.

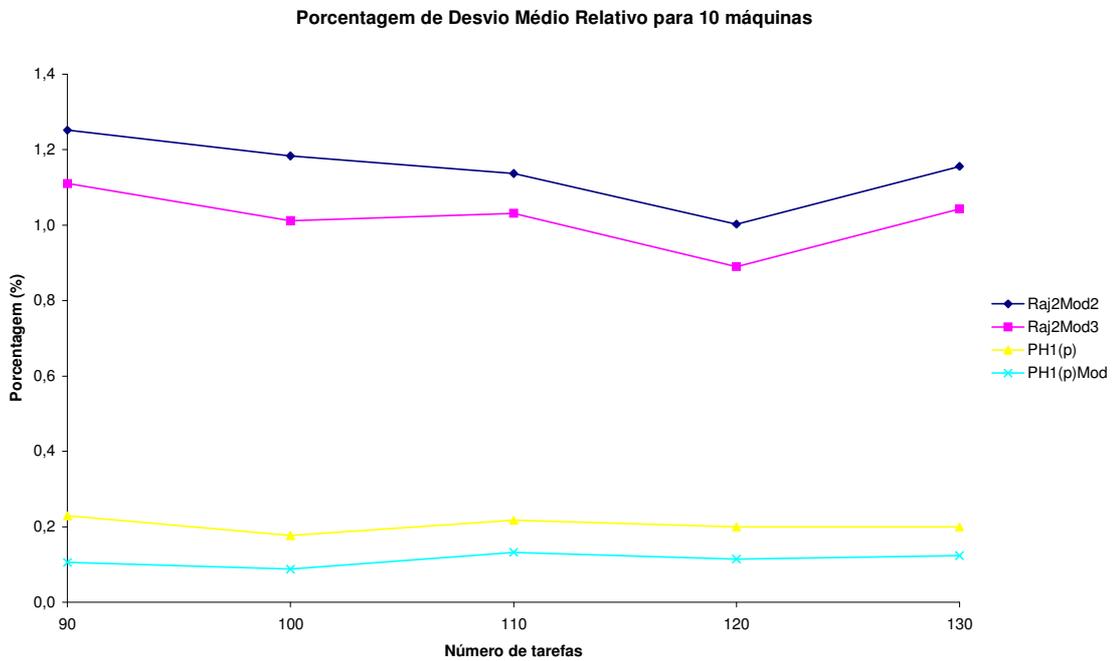


Figura 4.27– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas.

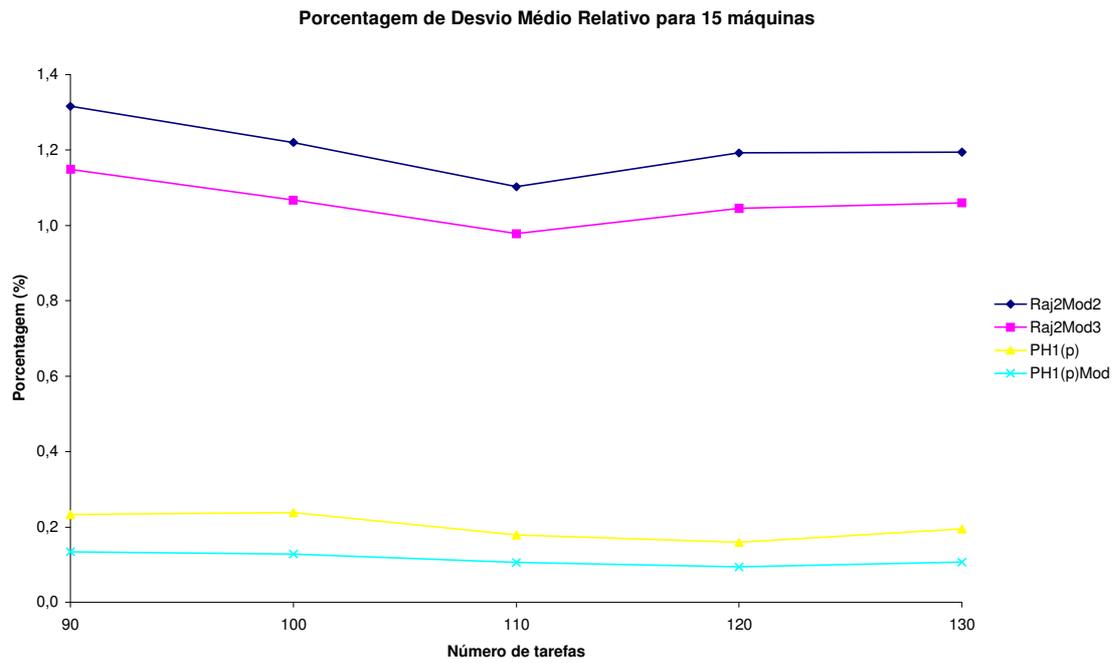


Figura 4.28– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas.

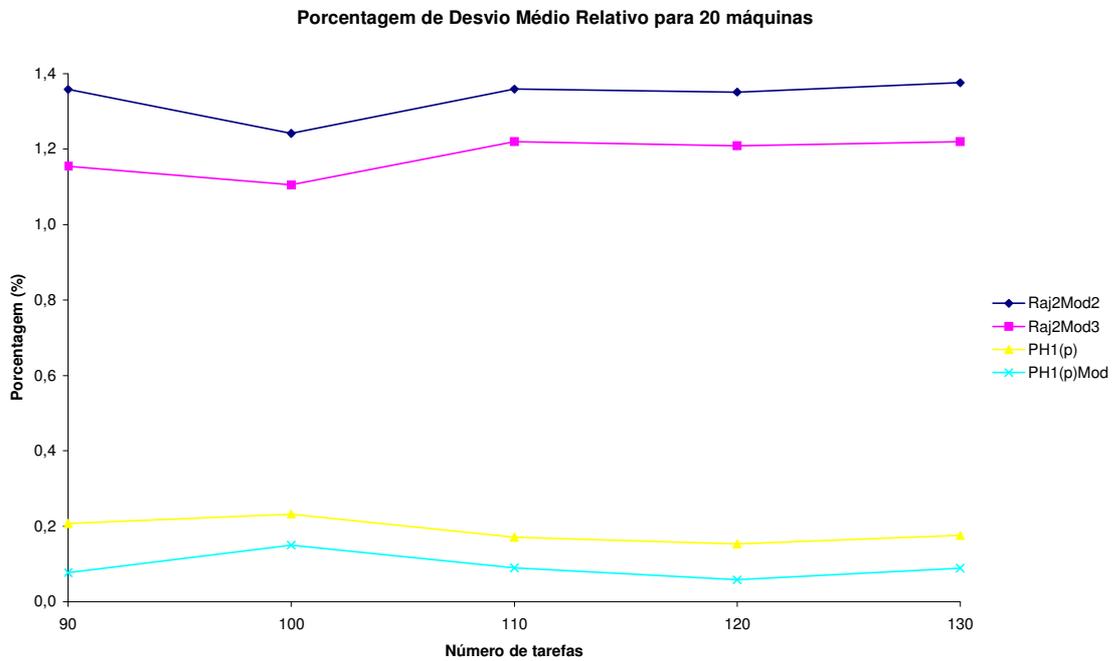


Figura 4.29– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas.

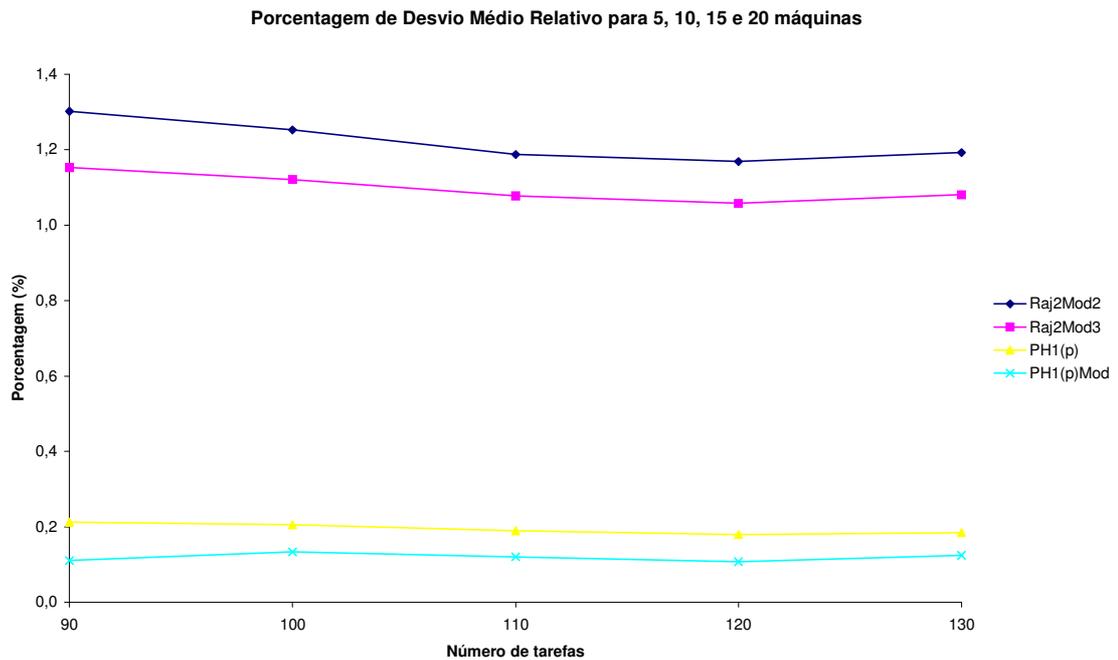


Figura 4.30– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Para problemas de grande porte, figuras 4.26 a 4.30, ocorre algo similar com os problemas de médio porte, o algoritmo PH1(p)Mod apresenta porcentagem de desvio médio relativo abaixo de 0,2%, seguido de perto pelo segundo melhor método, o algoritmo PH1(p). Os outros dois algoritmos apresentam porcentagens de desvio médio relativo maiores, entre de 1% e 1,4%.

4.1.3 Tempo Médio de Computação

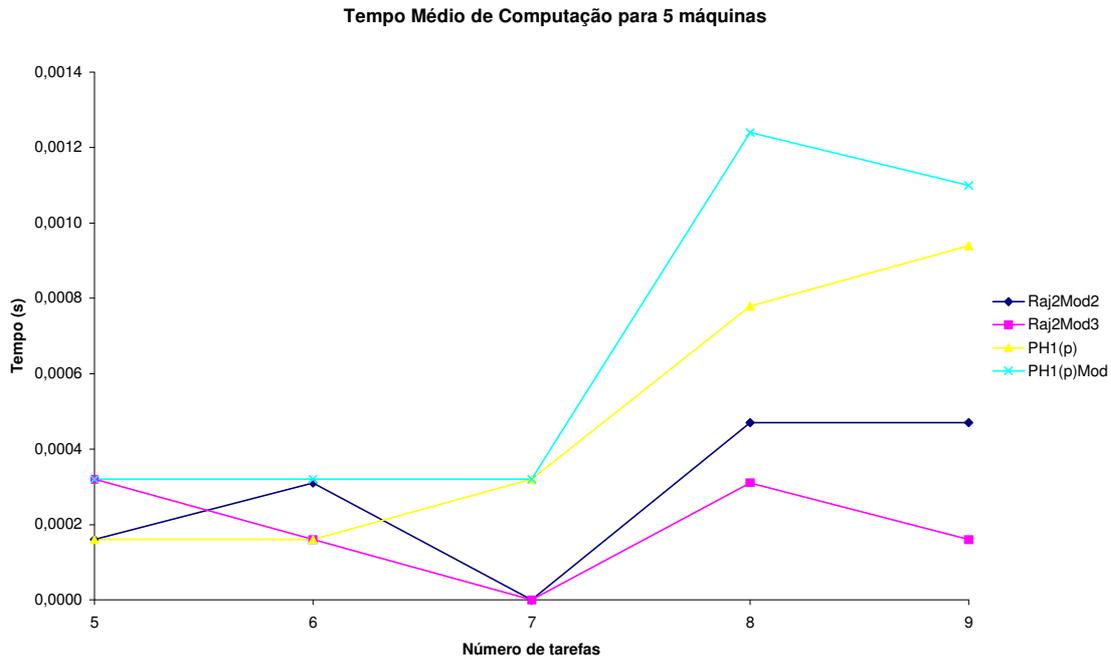


Figura 4.31– Tempo médio de computação vs. Número de tarefas, para 5 máquinas.

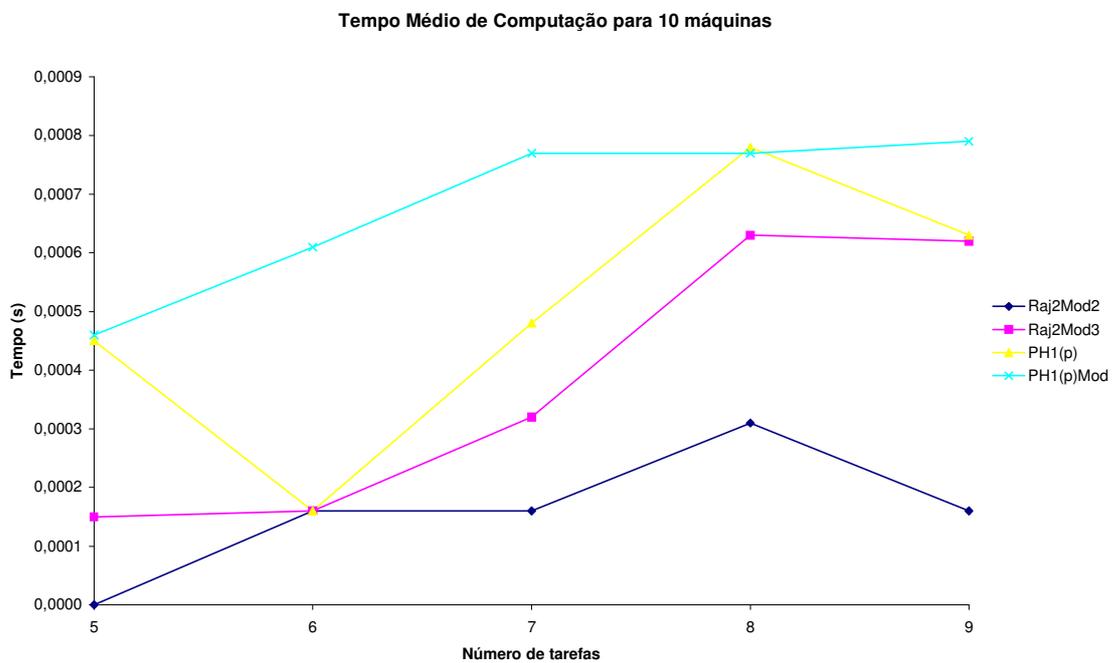


Figura 4.32– Tempo médio de computação vs. Número de tarefas, para 10 máquinas.

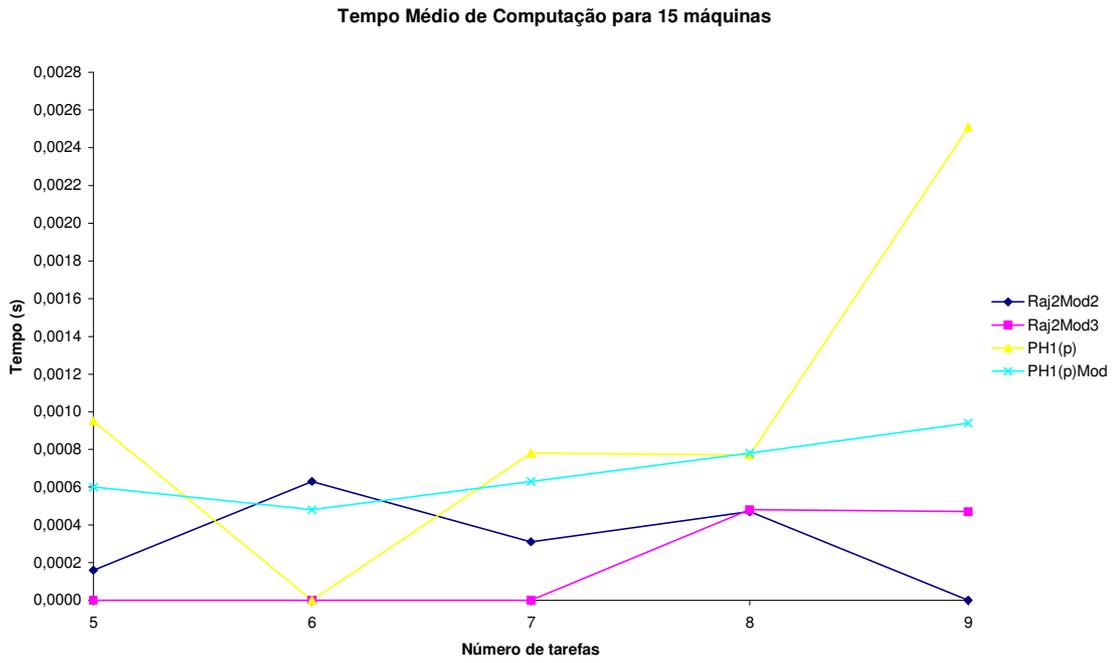


Figura 4.33– Tempo médio de computação vs. Número de tarefas, para 15 máquinas.

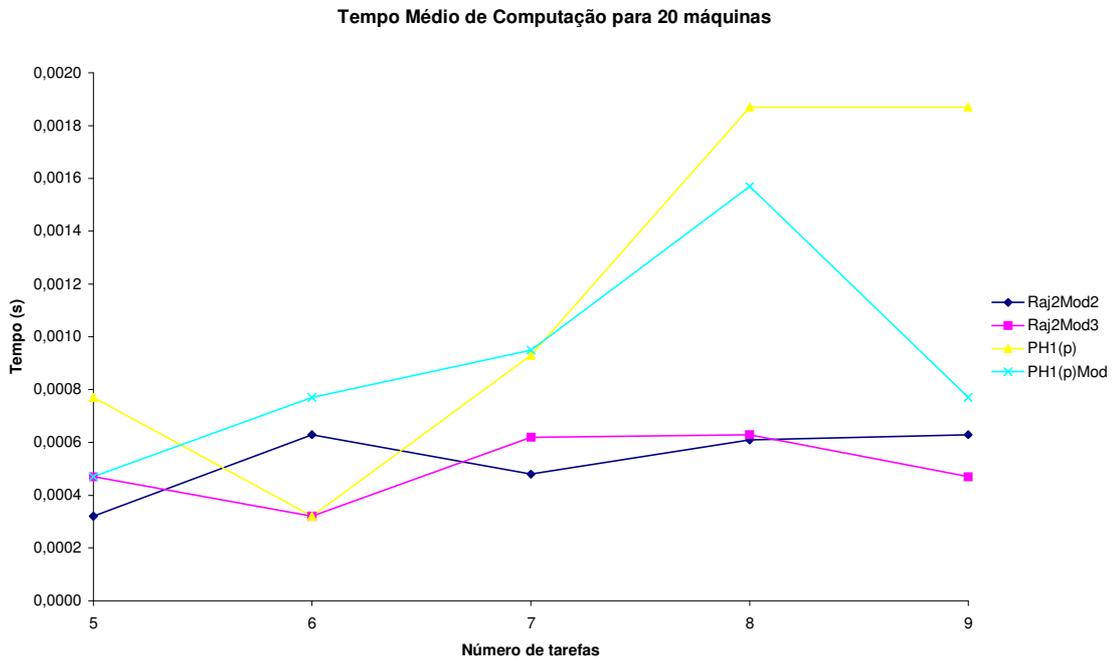


Figura 4.34– Tempo médio de computação vs. Número de tarefas, para 20 máquinas.

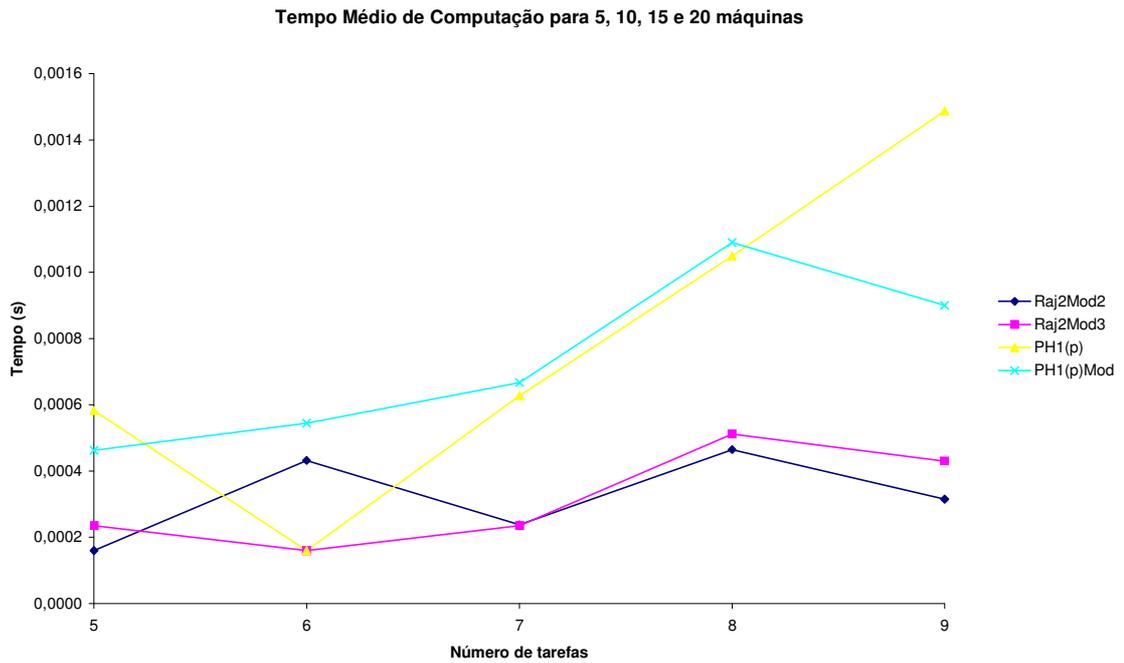


Figura 4.35– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20)

De forma agrupada, para problemas de pequeno porte, percebe-se que os melhores métodos (PH1(p) e sua versão modificada) consomem maior tempo médio de computação que os demais. Porém, trata-se de menos de 0,0016 segundos.

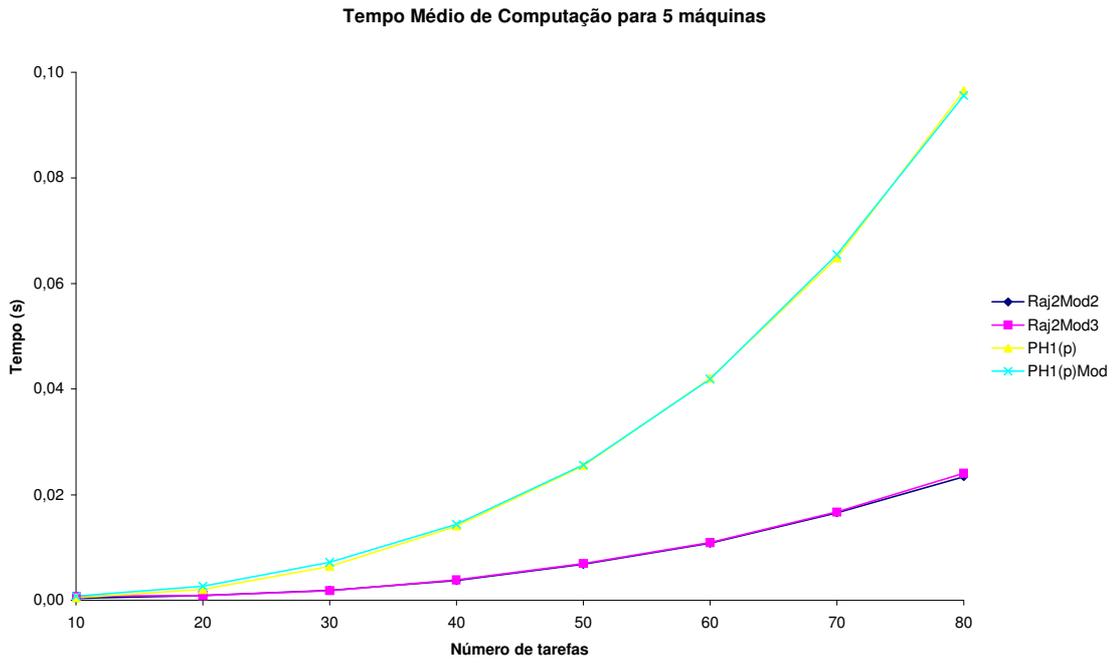
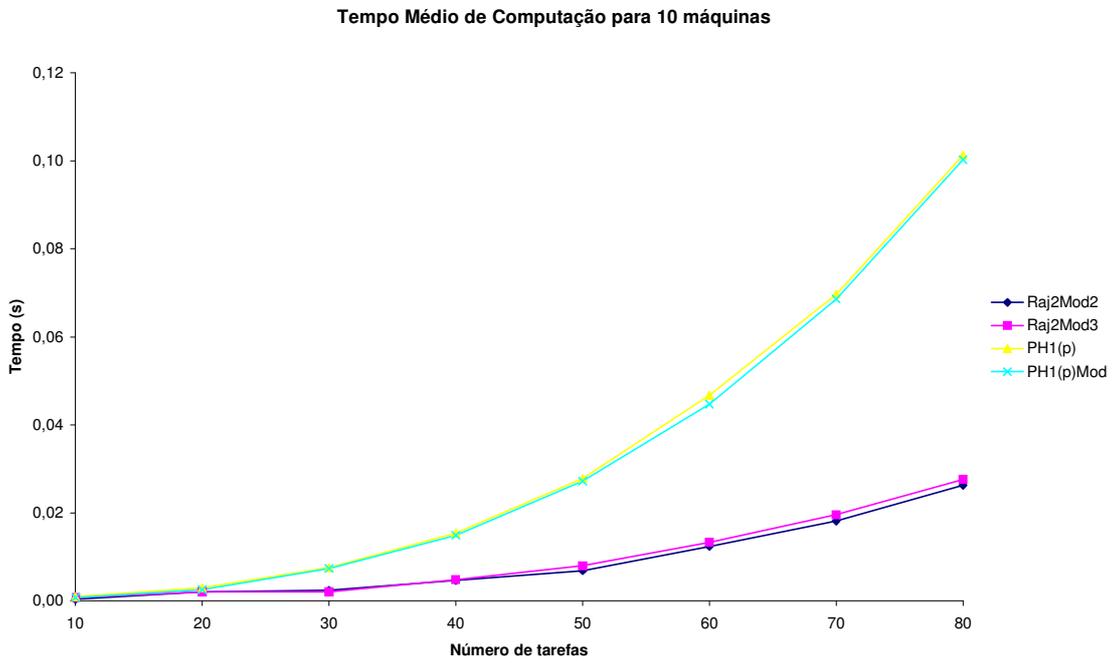


Figura 4.36– Tempo médio de computação vs. Número de tarefas, para 5 máquinas.



4.37– Tempo médio de computação vs. Número de tarefas, para 10 máquinas.

Figura

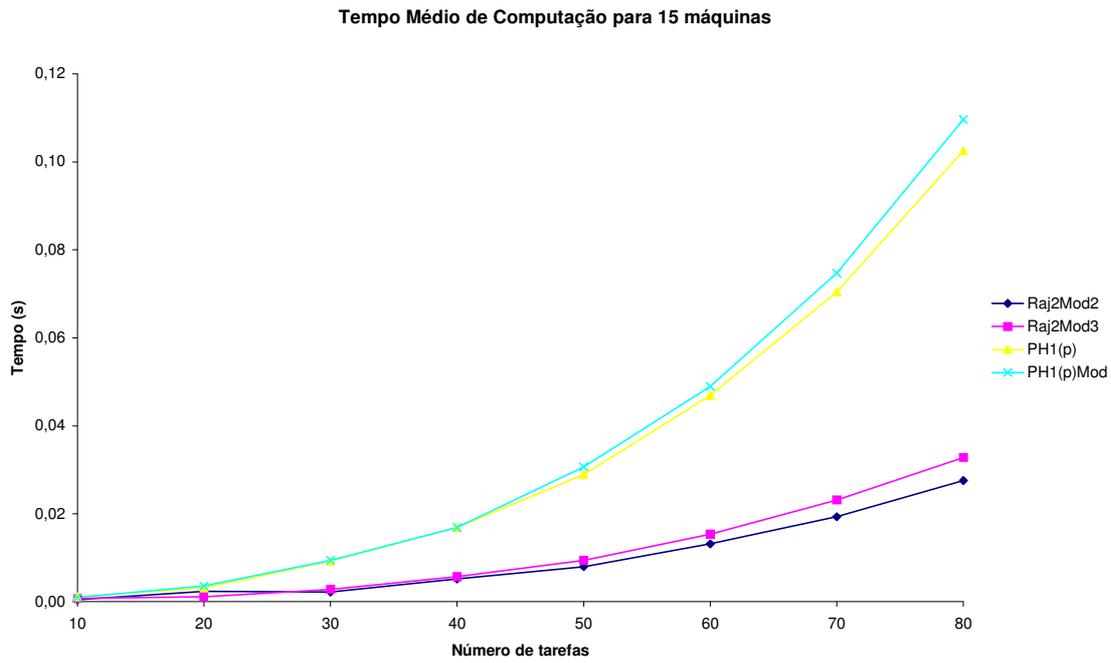


Figura 4.38– Tempo médio de computação vs. Número de tarefas, para 15 máquinas.

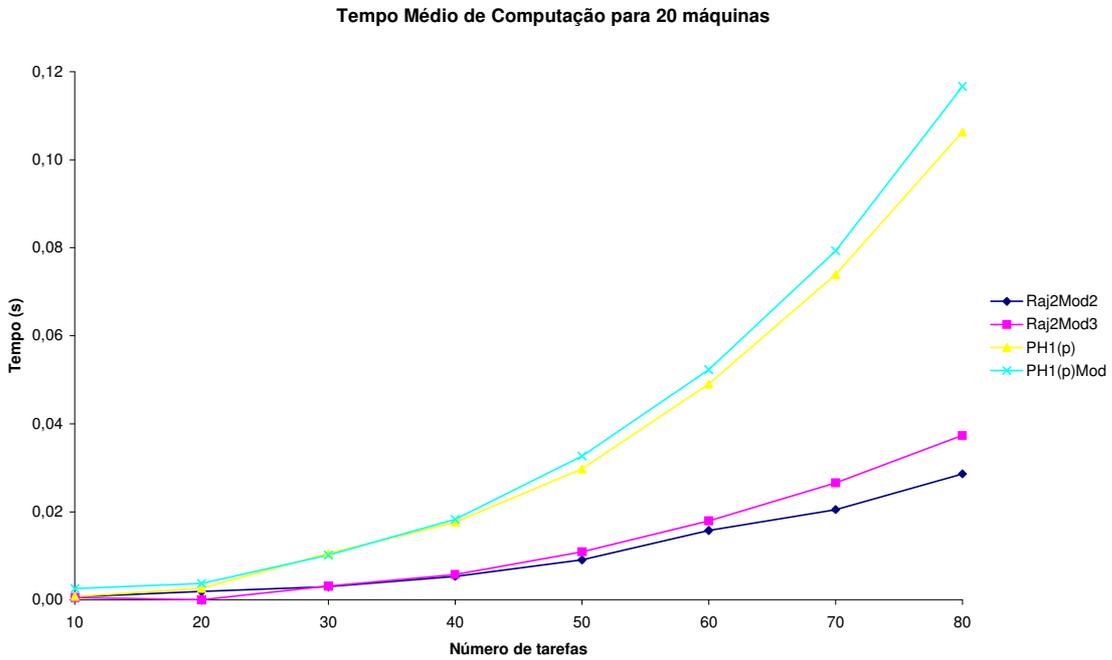


Figura 4.39– Tempo médio de computação vs. Número de tarefas, para 20 máquinas.

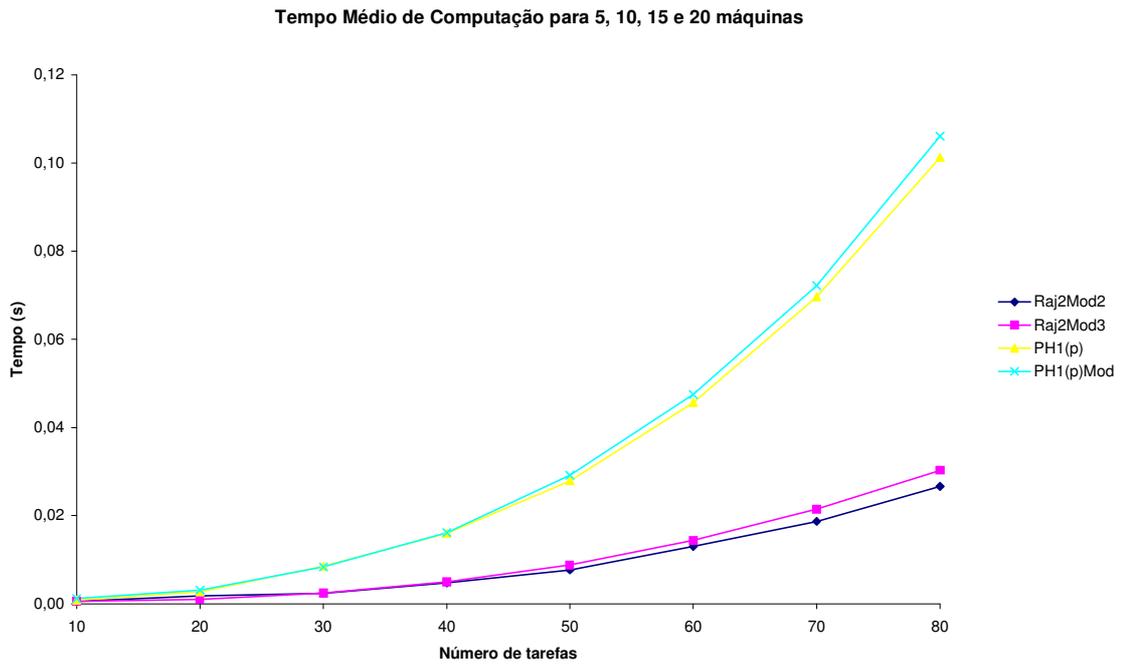


Figura 4.40– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

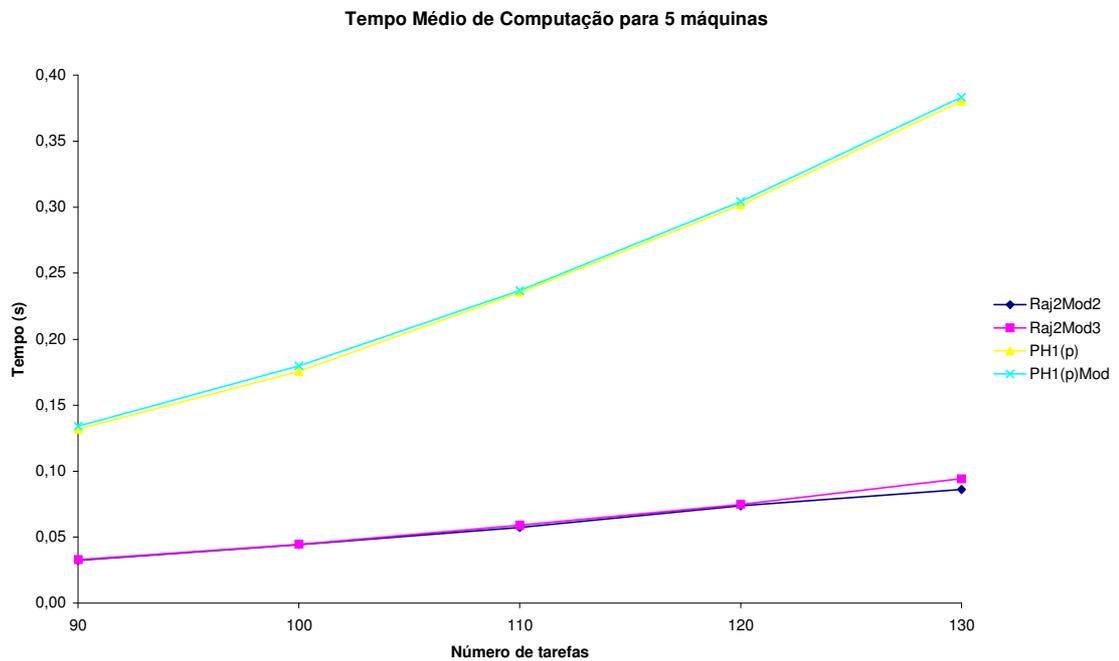


Figura 4.41– Tempo médio de computação vs. Número de tarefas, para 5 máquinas.

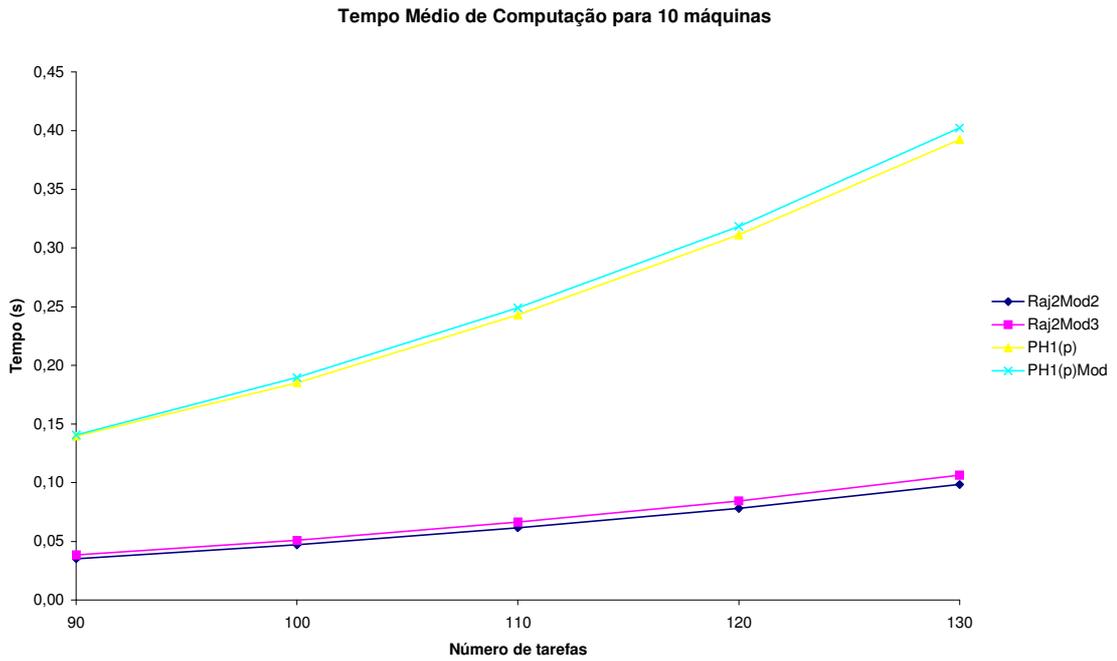


Figura 4.42– Tempo médio de computação vs. Número de tarefas, para 10 máquinas.

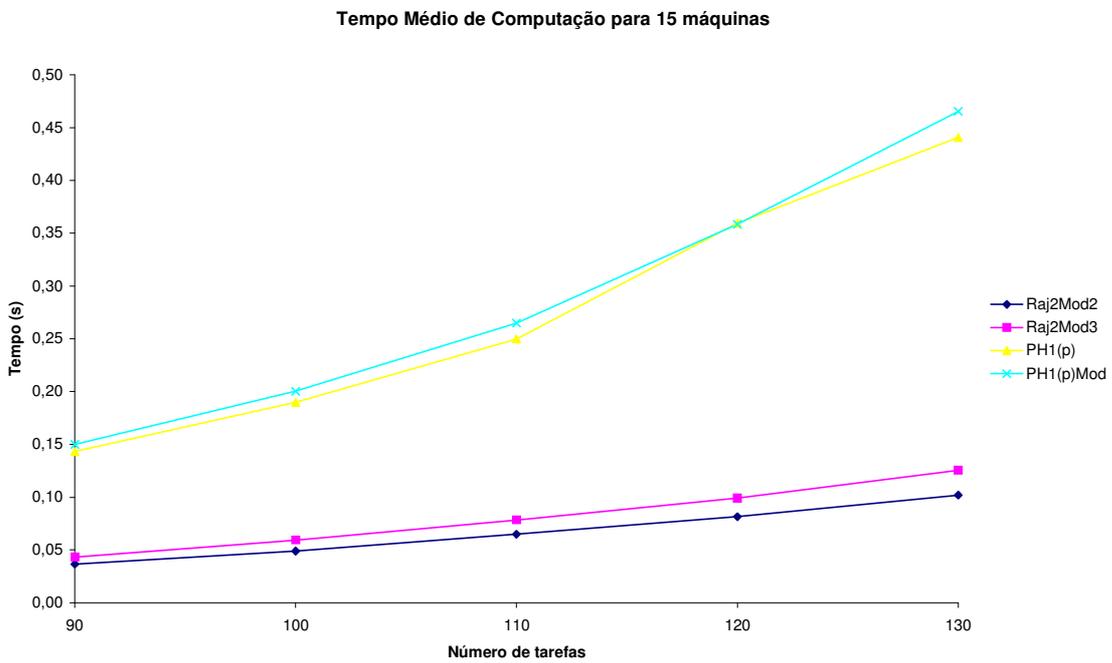


Figura 4.43– Tempo médio de computação vs. Número de tarefas, para 15 máquinas.

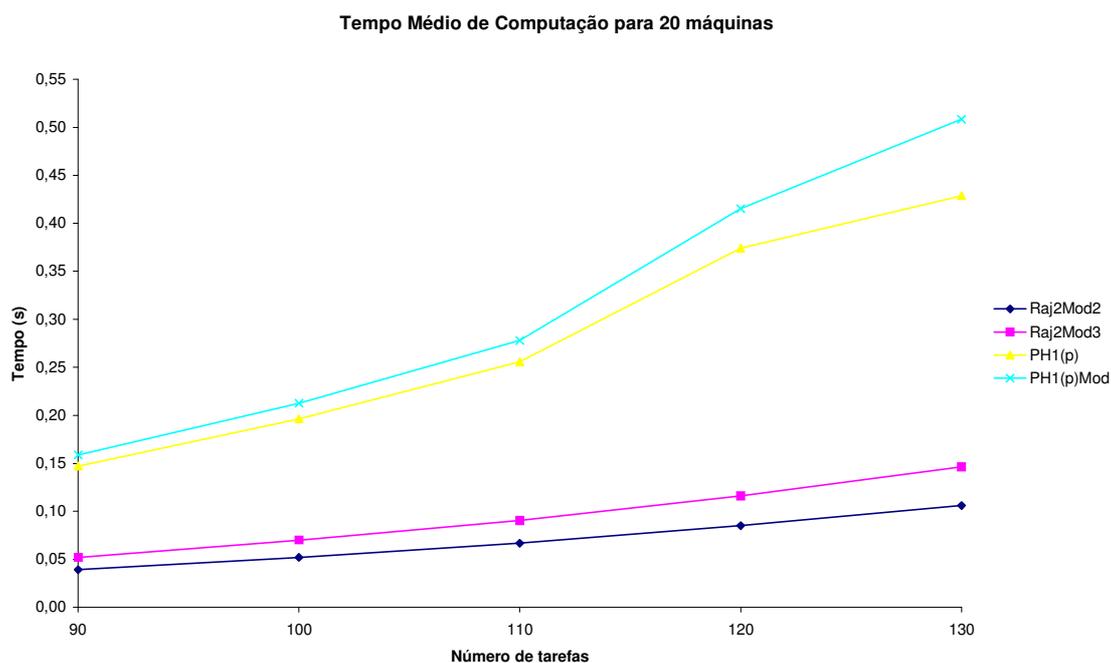


Figura 4.44– Tempo médio de computação vs. Número de tarefas, para 20 máquinas.

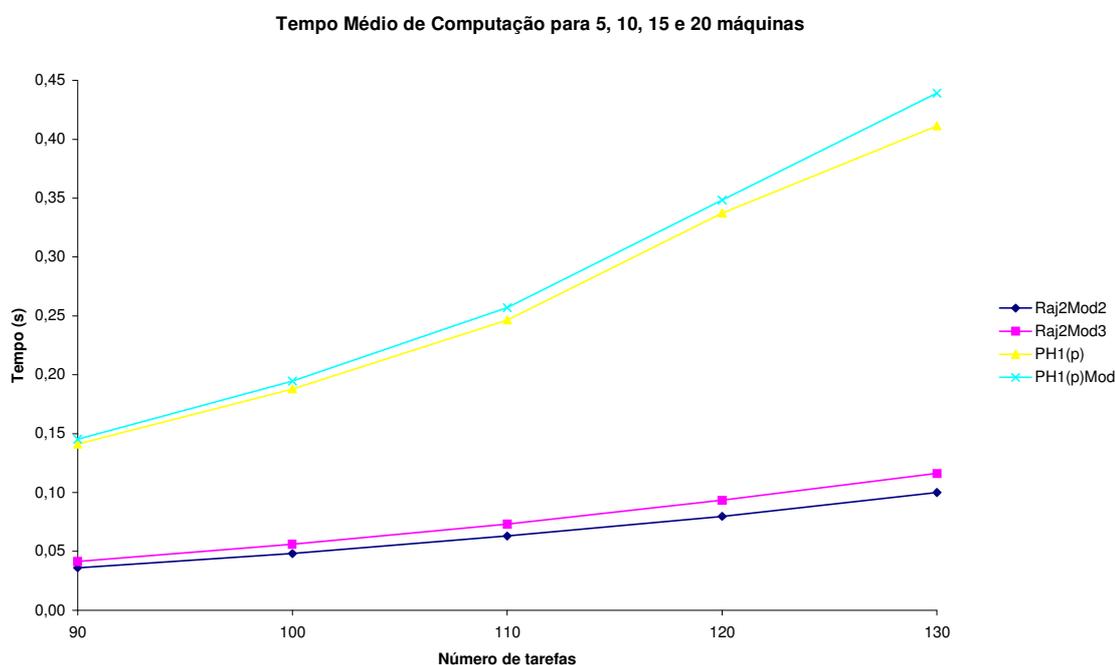


Figura 4.45– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Nos problemas de médio e grande porte, analisados através das figuras 4.36 a 4.45, fica evidente que quanto maior o número de tarefas, maior o tempo médio de computação dos algoritmos, principalmente dos algoritmos PH1(p) e PH1(p)Mod.

Porém, mesmo para problemas de 130 tarefas, a viabilidade das soluções não estão comprometidas, uma vez que o tempo médio de computação não é superior a 0,45 segundos para nenhum algoritmo.

Com base nas experimentações computacionais (Parte A e Parte B), podemos concluir que os dois melhores métodos heurísticos são respectivamente C_SN(2F), de duas fases e PH1(p)Mod, de três fases.

No próximo capítulo serão propostos dois novos métodos heurísticos construtivos de fase única e mais adiante serão comparados com os dois melhores métodos citados no parágrafo anterior.

CAPÍTULO 5

Proposição de Dois Métodos Heurísticos Construtivos de Fase Única

Os dois métodos heurísticos propostos neste capítulo possuem uma única fase, e serão denominados C1_SN e C2_SN, para fins de abreviação, identificação e comparação com os demais métodos já existentes.

Tratam-se de dois métodos elaborados através da combinação dos procedimentos de melhoria que utiliza a permutação de tarefas (*pairwise interchange*) e de melhoria que utiliza a inserção de tarefas (*insertion*).

Nos dois métodos, o ponto de partida mais uma vez é através da parcela da soma dos tempos de fluxo de duas tarefas adjacentes J_i e J_j , a partir do início de J_i (ou seja, no intervalo de tempo entre o início de J_i e o término de J_j), mas não existe a ordenação inicial. Assim, as tarefas são seqüenciadas a partir da matriz F ($n \times n$) e, a cada iteração, a seqüência parcial é submetida ao método de melhoria que utiliza a permutação de tarefas e ao método de melhoria que utiliza a inserção de tarefas.

5.1 Método Heurístico C1_SN

Seja $J = \{J_1, J_2, \dots, J_n\}$ um conjunto de n tarefas que devem ser processadas, e S o conjunto das tarefas programadas; S_p a melhor seqüência parcial obtida após a execução do processo de melhoria que utiliza a permutação de tarefas; e S_i a melhor seqüência parcial obtida após a execução do processo de melhoria que utiliza a inserção de tarefas, o método heurístico C1_SN é composto pelos seguintes passos:

Passo 1

$$J = \{J_1, J_2, \dots, J_i, \dots, J_n\};$$

$$S = \emptyset;$$

Selecione o menor elemento $F_m(J_i, J_j)$;

$$S = \{J_i, J_j\};$$

$$J \leftarrow J - \{J_i, J_j\};$$

$$u \leftarrow J_{[2]};$$

$$k = 3;$$

Passo 2

Selecione o menor elemento $F_m(J_u, J_v)$ tal que $J_v \in J$;

$$S \leftarrow S \cup \{J_v\};$$

$$J \leftarrow J - \{J_v\};$$

$$u \leftarrow J_{[k]};$$

$$k \leftarrow k + 1;$$

Passo 3

- Considerando toda a Vizinhança de Permutação da seqüência parcial com $(k-1)$ tarefas, constituída de $k(k-1)/2$ seqüências, determine a seqüência S_p associada a menor soma dos tempos de fluxo;
- Considerando toda a Vizinhança de Inserção da seqüência parcial com $(k-1)$ tarefas, constituída de $(k-1)^2$ seqüências, determine a seqüência S_i associada a menor soma dos tempos de fluxo;
- Se S_p é melhor que S e S_i , atualize S com S_p ;
- Se S_i é melhor que S e S_p , atualize S com S_i ;
- $u \leftarrow J_{[k-1]}$;
- Volte ao *Passo 2* até que todas as tarefas estejam seqüenciadas.

Como pode-se verificar, trata-se de um método simples, com poucos passos e que a cada inserção de uma nova tarefa (a partir da terceira), simultaneamente são executados os procedimentos de melhoria que utiliza a permutação de tarefas e de melhoria que utiliza inserção de tarefas, prevalecendo sempre o melhor método a cada parcial avaliada.

A seguir é apresentado o esquema de fluxograma para os passos 2 e 3 do método C1_SN:

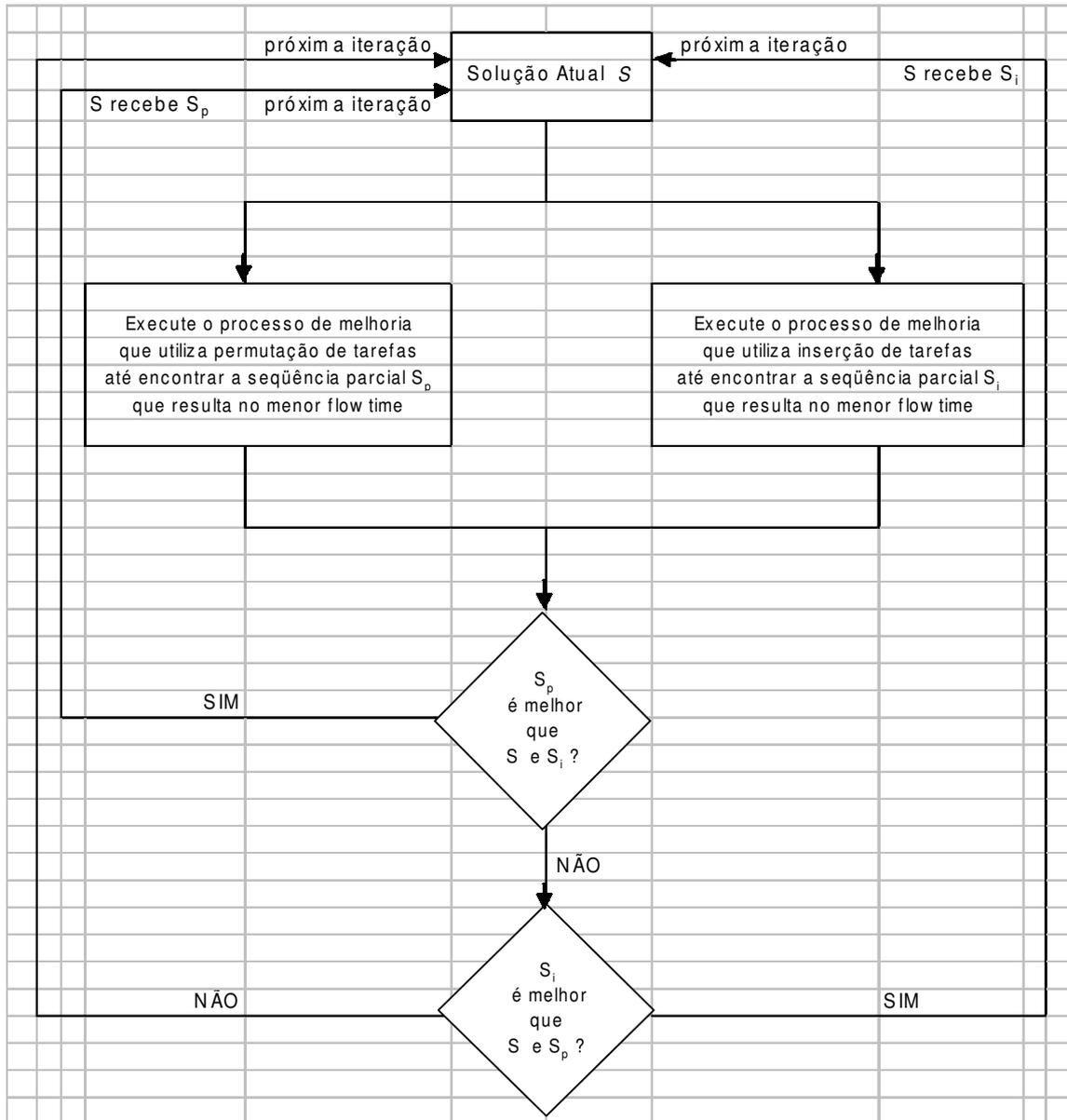


Figura 5.1- Esquema de fluxograma do método proposto C1_SN

5.2 Método Heurístico C2_SN

O segundo método é similar ao primeiro, com algumas poucas alterações. A base é praticamente a mesma: matriz F ($n \times n$), processo de melhoria que utiliza a permutação de tarefas e processo de melhoria que utiliza a inserção de tarefas.

O algoritmo proposto para o método heurístico construtivo C2_SN é definido pelos seguintes passos:

Passo 1

$$J = \{J_1, J_2, \dots, J_i, \dots, J_n\};$$

$$S = \emptyset;$$

Selecione o menor elemento $F_m(J_i, J_j)$;

$$S = \{J_i, J_j\};$$

$$J \leftarrow J - \{J_i, J_j\};$$

$$u \leftarrow J_{[2]};$$

$$k = 3;$$

Passo 2

Selecione o menor elemento $F_m(J_u, J_v)$ tal que $J_v \in J$;

$$S \leftarrow S \cup \{J_v\};$$

$$J \leftarrow J - \{J_v\};$$

$$u \leftarrow J_{[k]};$$

$$k \leftarrow k + 1;$$

Passo 3

- Considerando toda a Vizinhança de Inserção da seqüência parcial com $(k-1)$ tarefas, constituída de $(k-1)^2$ seqüências, determine a seqüência S_i associada a menor soma dos tempos de fluxo;
- Se S_i é melhor que S , atualize S com S_i e, considerando toda a Vizinhança de Permutação da seqüência parcial com $(k-1)$ tarefas, constituída de $k(k-1)/2$ seqüências, determine a seqüência S_p associada a menor soma dos tempos de fluxo;
- Se S_p é melhor que S , atualize S com S_p ;
- Se S_i não é melhor que S , considerando toda a Vizinhança de Permutação da seqüência parcial com $(k-1)$ tarefas, constituída de $k(k-1)/2$ seqüências, determine a seqüência S_p associada a menor soma dos tempos de fluxo;
- Se S_p é melhor que S , atualize S com S_p ;
- $u \leftarrow J_{[k-1]}$;
- Volte ao *Passo 2* até que todas as tarefas estejam seqüenciadas.

Trata-se também de um método simples, com poucos passos e que a cada inserção de uma

nova tarefa (a partir da terceira), é executado o procedimento de melhoria que utiliza a inserção de tarefas. Em seguida, é aplicado o processo de melhoria que utiliza a permutação de tarefas, prevalecendo sempre o melhor método após cada seqüência parcial apurada.

A seguir é apresentado o esquema de fluxograma para os passos 2 e 3 do método C2_SN:

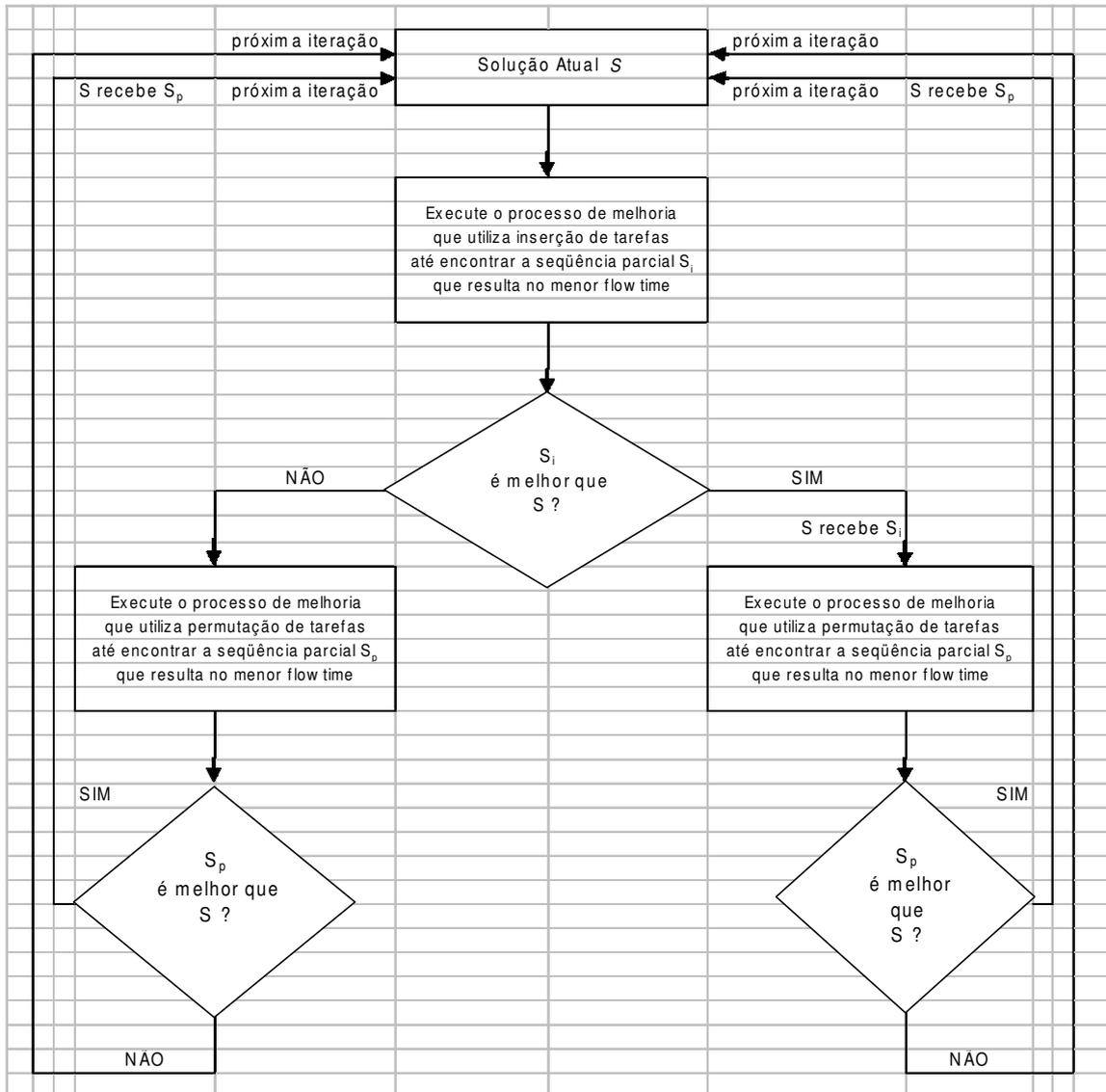


Figura 5.2- Esquema de fluxograma do método proposto C2_SN

A principal diferença entre os dois métodos decorre da combinação na aplicação dos processos de melhoria que utilizam permutação e inserção de tarefas. No método C1_SN os dois processos de melhoria são aplicados de forma simultânea na seqüência parcial S. Já no método C2_SN, primeiro é aplicado o processo de melhoria que utiliza a inserção de tarefas na seqüência parcial S e em seguida é aplicado o processo de melhoria que utiliza a permutação de tarefas sobre a seqüência parcial atualizada S.

No capítulo 6, a seguir, serão comparados os dois melhores métodos heurísticos de duas fases e três fases, com os dois métodos heurísticos construtivos de fase única propostos neste capítulo. Trata-se da experimentação computacional final, que identificará o melhor método heurístico para solução do problema.

CAPÍTULO 6

Experimentação Computacional – Parte Final

A base de dados e a metodologia de análise são as mesmas utilizadas nas experimentações computacionais – parte A e parte B. Porém, neste capítulo são avaliados os melhores algoritmos: PH1(p)Mod de três fases, o algoritmo proposto C_SN(2F) (método heurístico construtivo de 2 fases) e os algoritmos C1_SN e C2_SN, ambos métodos heurísticos construtivos de fase única.

6.1 Análise dos Resultados

Os resultados obtidos também são analisados em três partes e na seguinte ordem: Porcentagem de Sucesso; Porcentagem de Desvio Médio Relativo e Tempo Médio de Computação.

6.1.1 Análise da Porcentagem de Sucesso

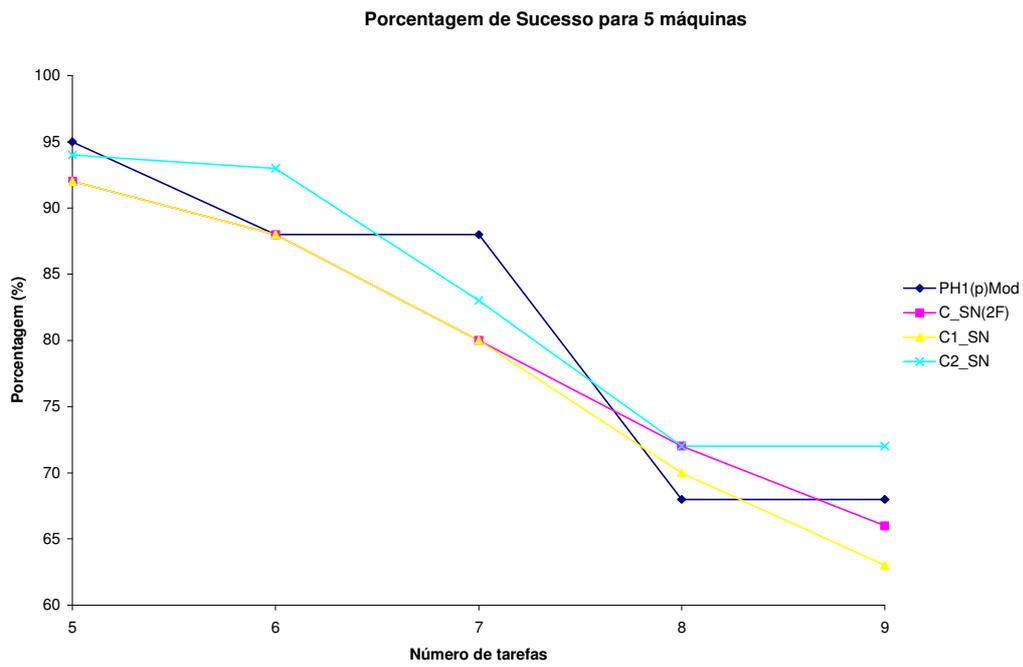


Figura 6.1– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

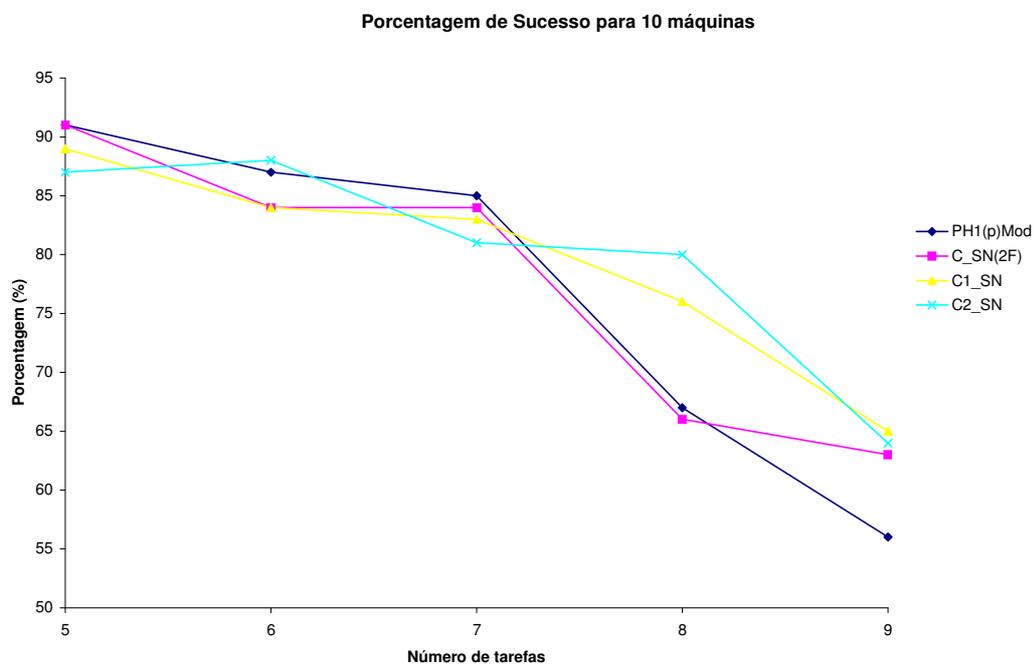


Figura 6.2– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

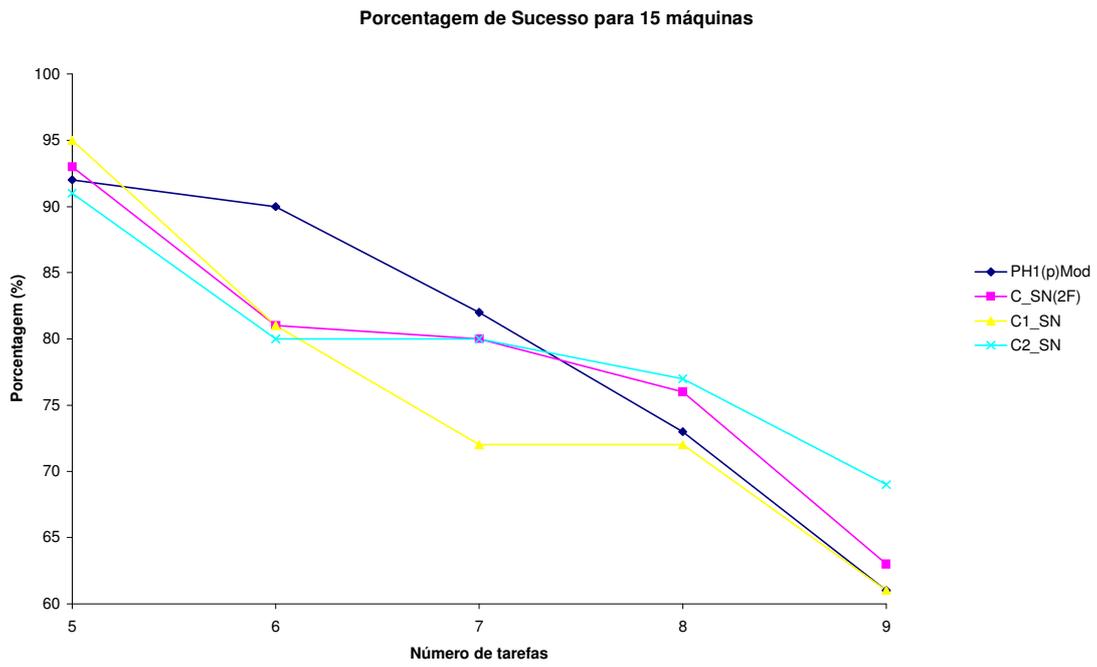


Figura 6.3– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

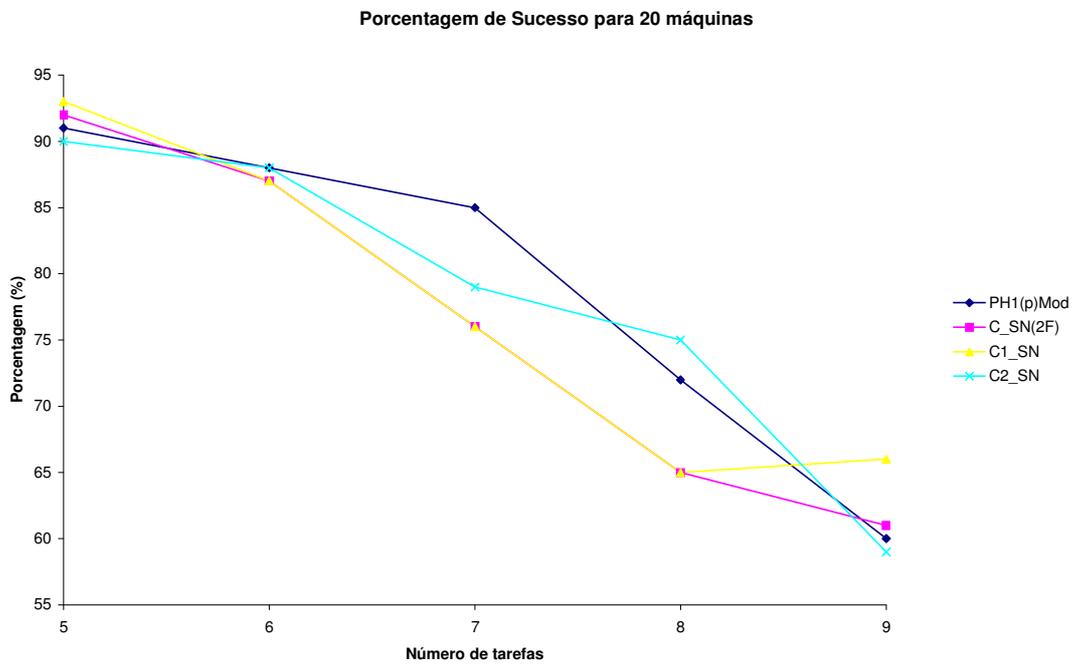


Figura 6.4– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

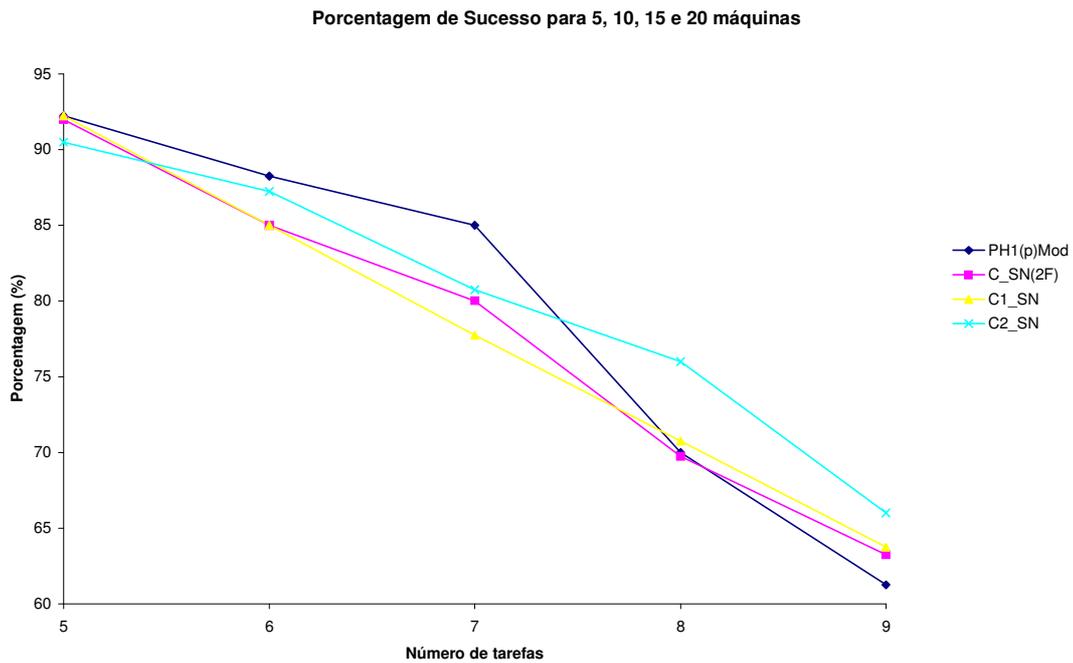


Figura 6.5– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Para os problemas de pequeno porte, percebe-se que, pelos fatos de ocorrerem menos empates e de diminuir as possibilidades de se alcançar o resultado ótimo, a porcentagem de sucesso é menor para cada método com 9 tarefas, em relação aos problemas de 5 tarefas. Para 5 tarefas, os algoritmos atingem entre 90% a 95% de sucesso, enquanto para 9 tarefas, a porcentagem de sucesso fica abaixo de 70%. De acordo com a figura 6.5, constata-se a similaridade dos resultados apresentados por todos os algoritmos.

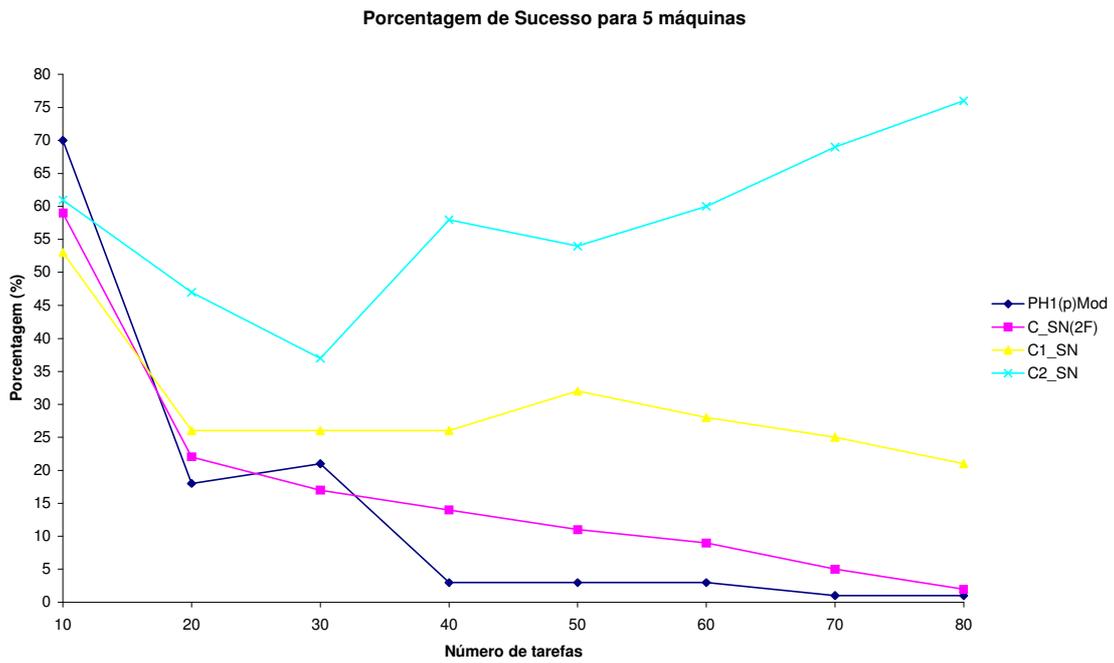


Figura 6.6– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

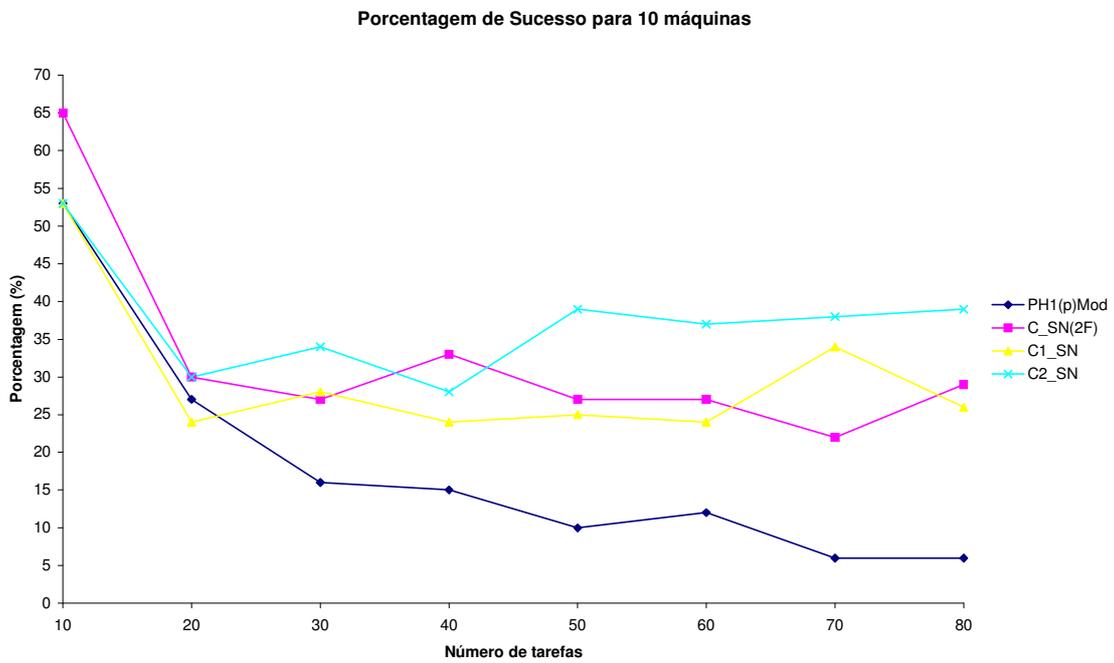


Figura 6.7– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

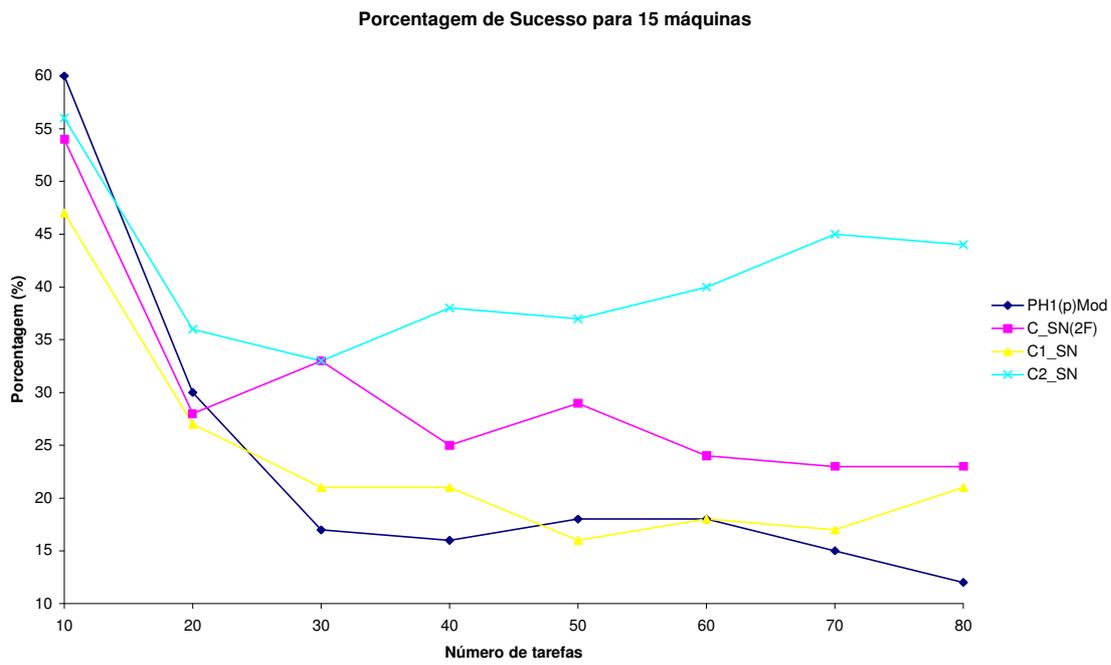


Figura 6.8– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

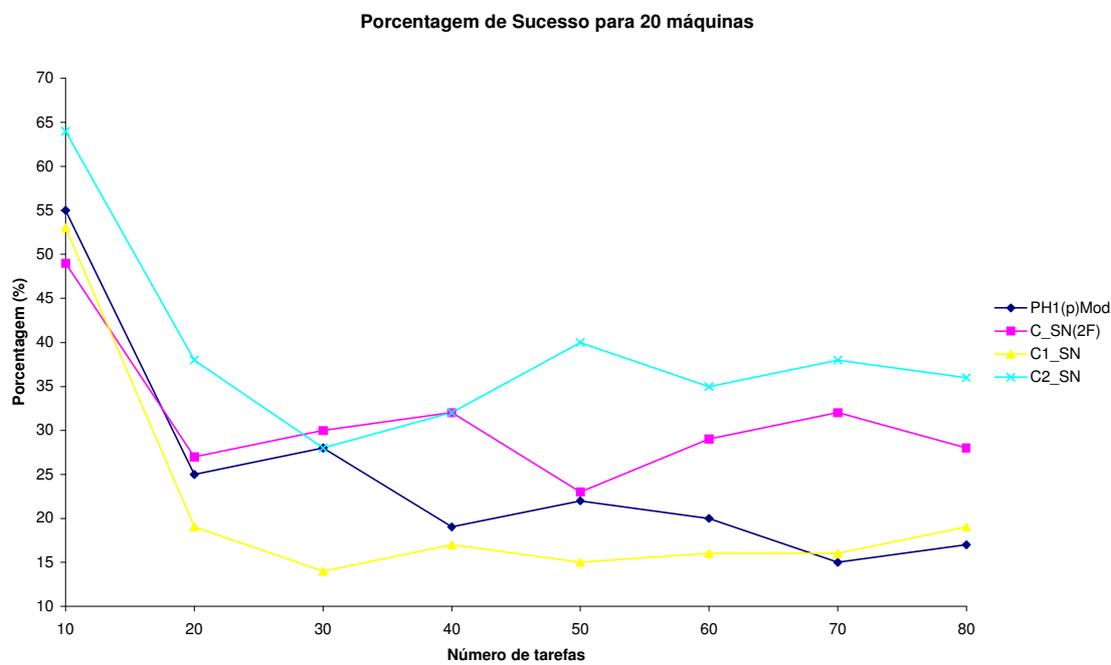


Figura 6.9– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

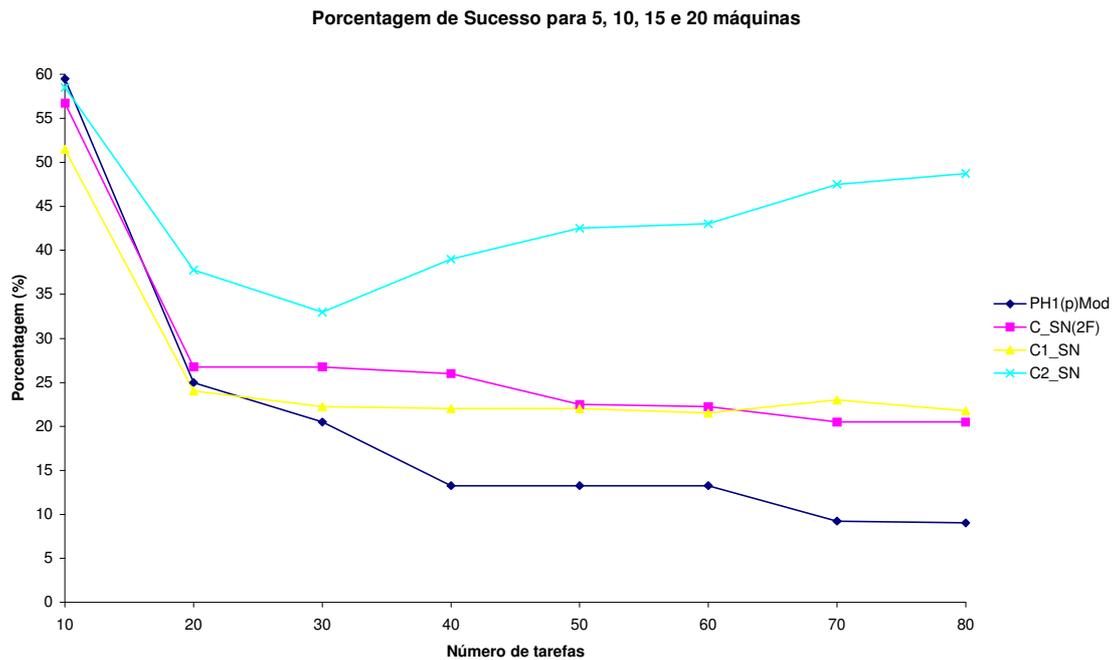


Figura 6.10– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Nos problemas de médio porte com 5 máquinas, a partir de 20 tarefas, o método C2_SN supera os demais, atingindo maiores porcentagens de sucesso, conforme figura 6.6.

O mesmo ocorre nas figuras 6.7 a 6.9, porém somente a partir de problemas com mais de 40 ou 50 tarefas. De forma consolidada, a figura 6.10 deixa claro que a partir de problemas com 20 tarefas o método que apresenta melhor porcentagem de sucesso é o C2_SN.

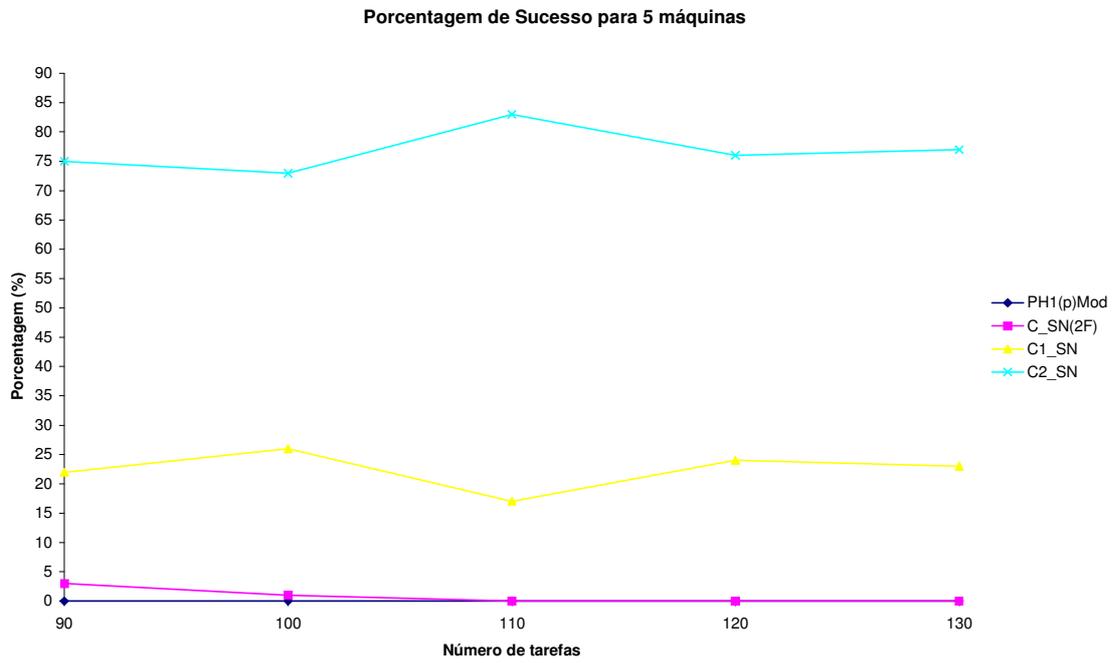


Figura 6.11– Porcentagem de sucesso vs. Número de tarefas, para 5 máquinas

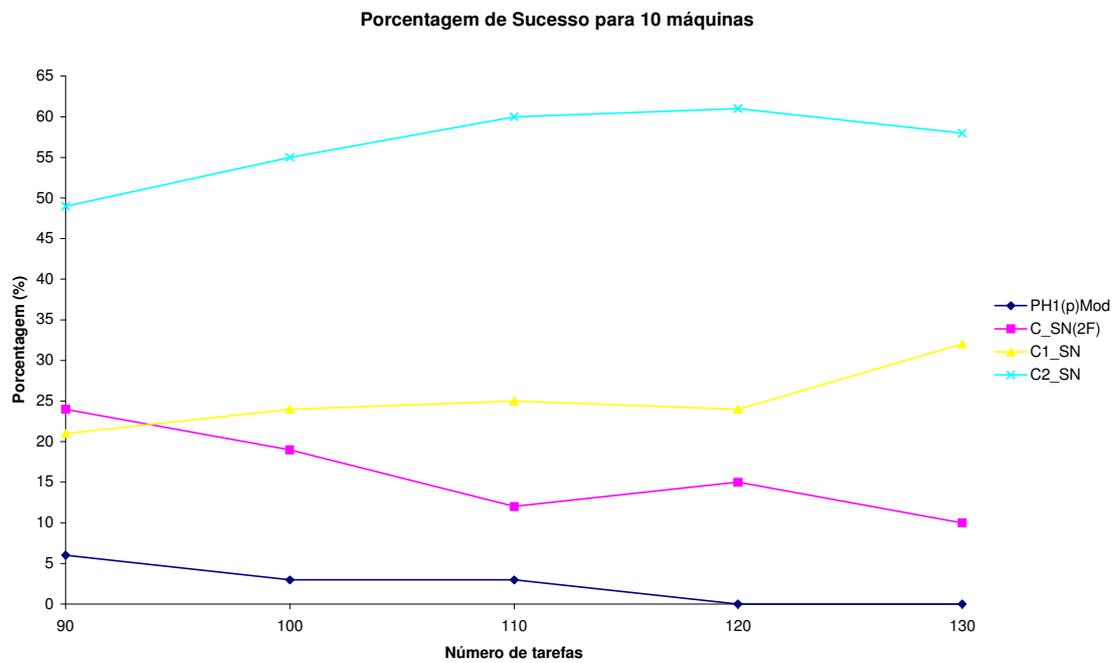


Figura 6.12– Porcentagem de sucesso vs. Número de tarefas, para 10 máquinas

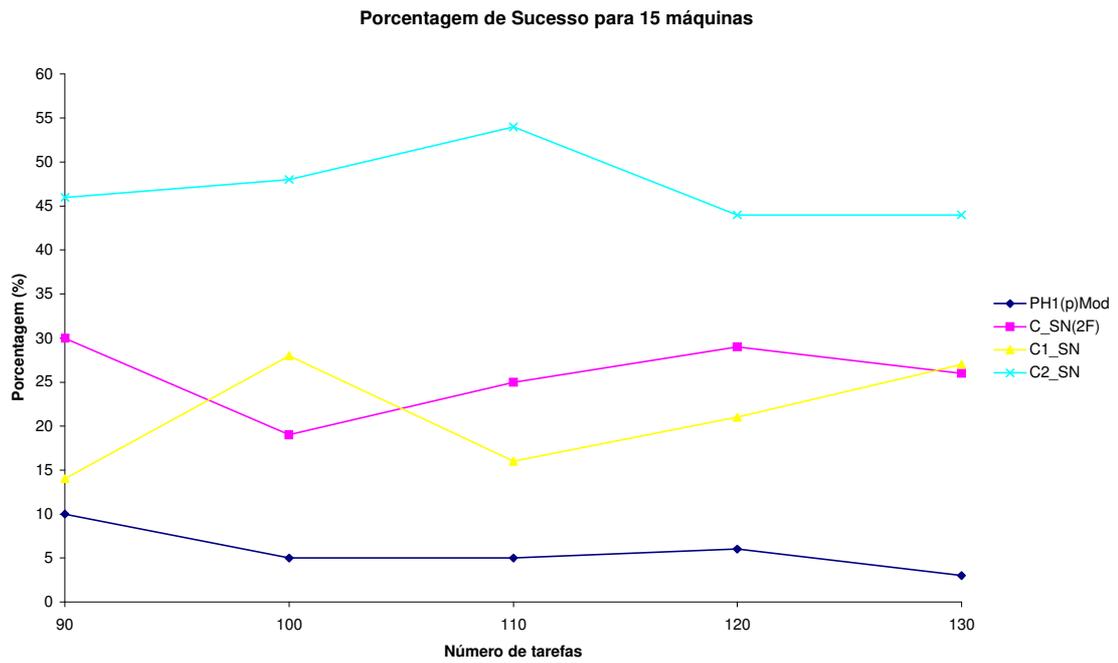


Figura 6.13– Porcentagem de sucesso vs. Número de tarefas, para 15 máquinas

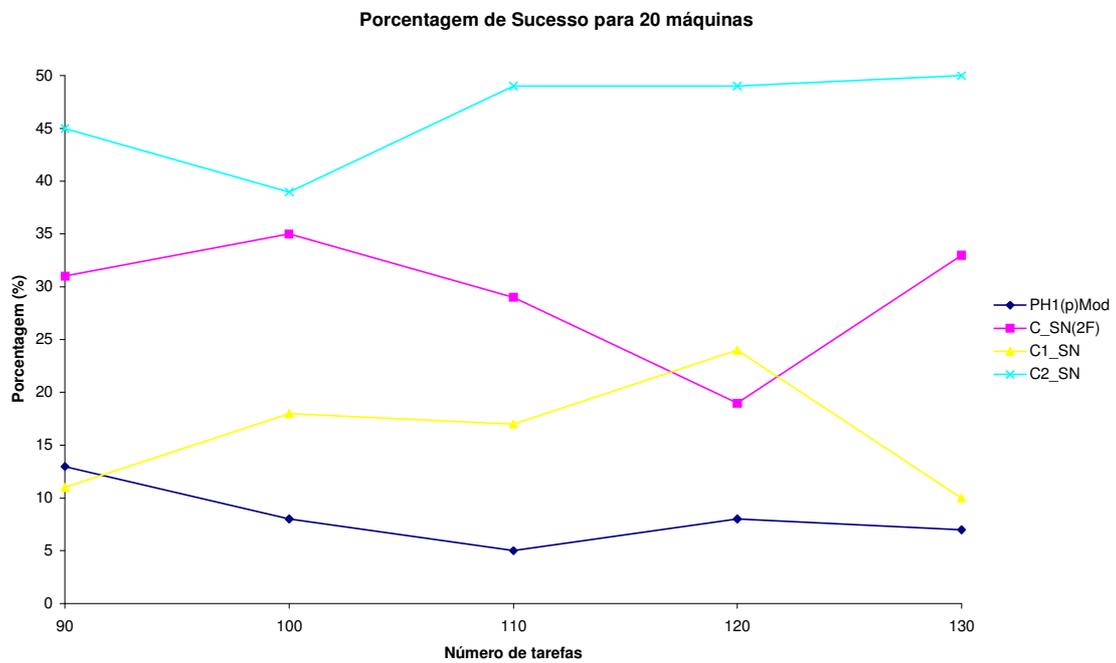


Figura 6.14– Porcentagem de sucesso vs. Número de tarefas, para 20 máquinas

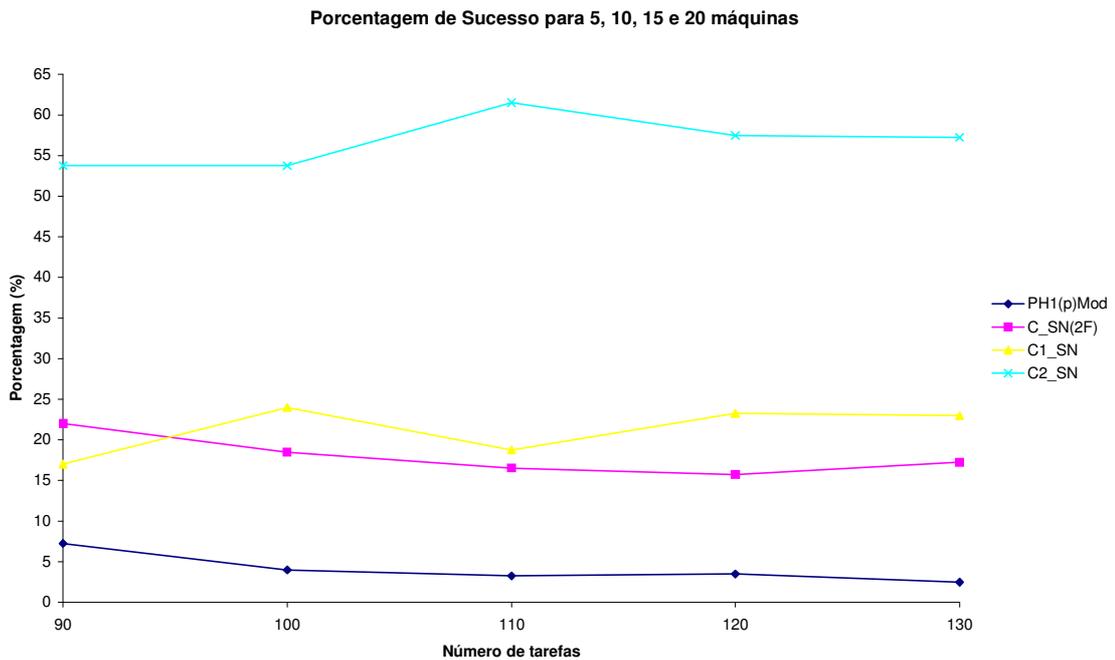


Figura 6.15– Porcentagem de sucesso vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

De forma consolidada (figura 6.15), para problemas de grande porte, o método C2_SN é superior aos demais, apresentando porcentagem de sucesso acima de 50%, enquanto os demais algoritmos apresentam porcentagens de sucesso abaixo de 25%.

6.1.2 Análise da Porcentagem do Desvio Médio Relativo

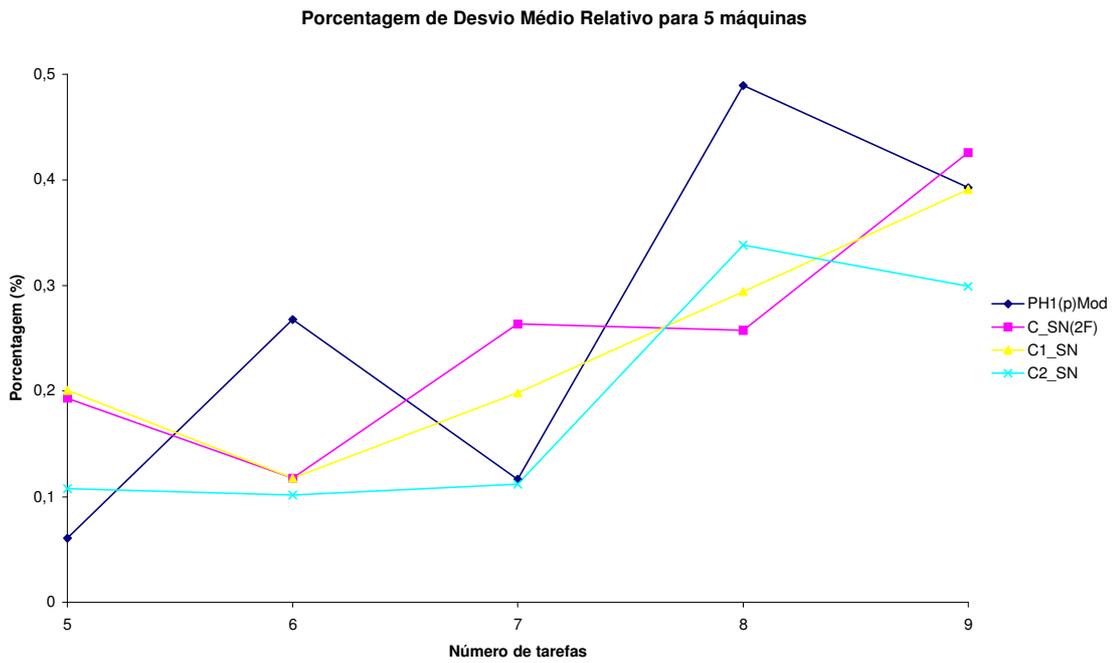


Figura 6.16– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas

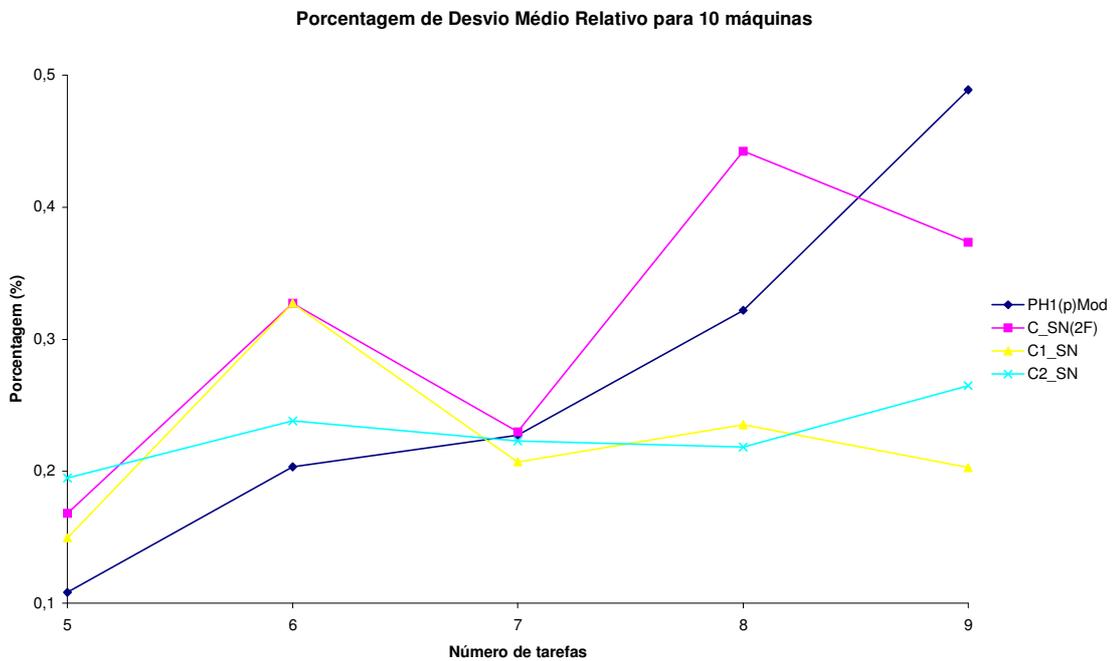


Figura 6.17– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas

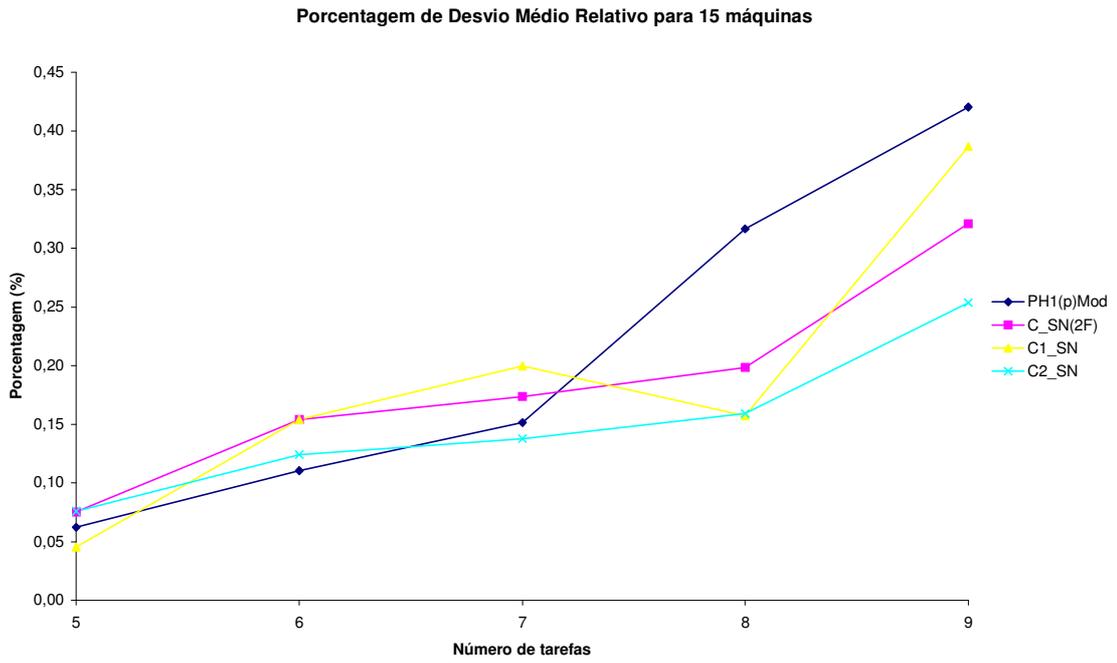


Figura 6.18– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas

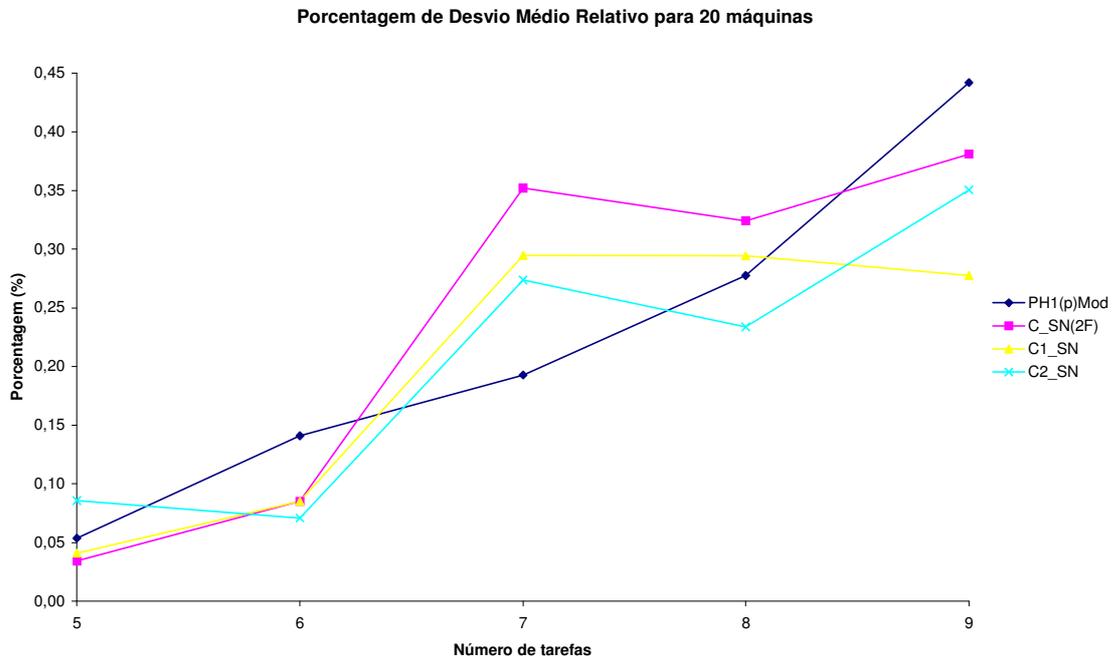


Figura 6.19– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas

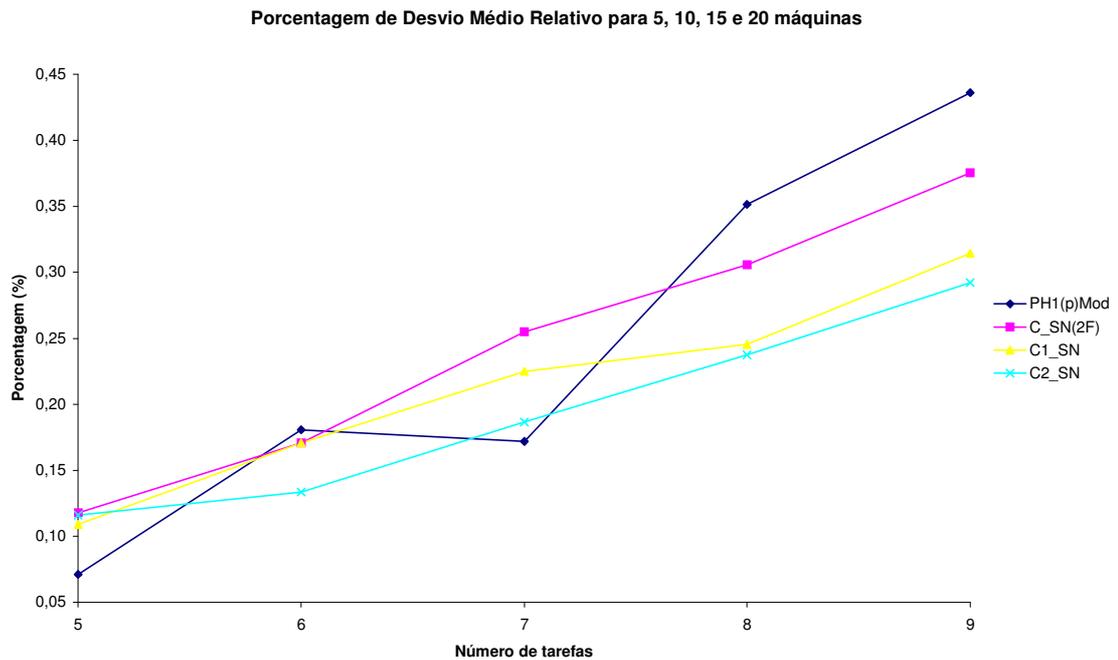


Figura 6.20– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

Assim como na análise de porcentagem de sucesso, através das figuras 6.16 a 6.20 verifica-se que os algoritmos de pequeno porte apresentam porcentagens de desvio médio relativo bastante similares e pequenos, sempre inferiores a 0,5%.

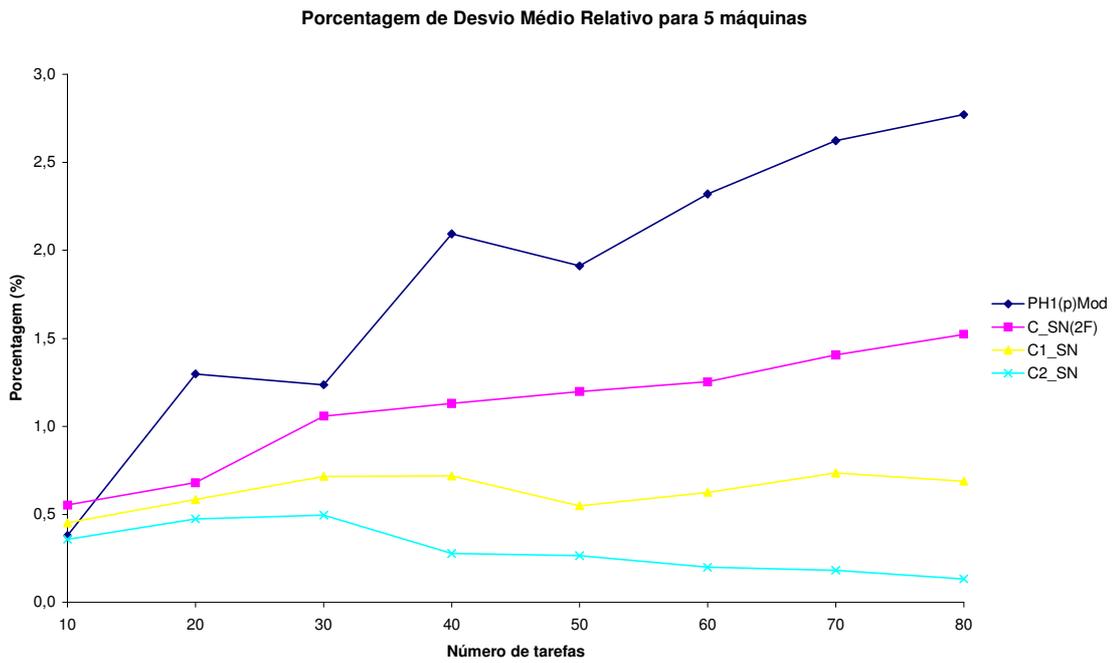


Figura 6.21– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas

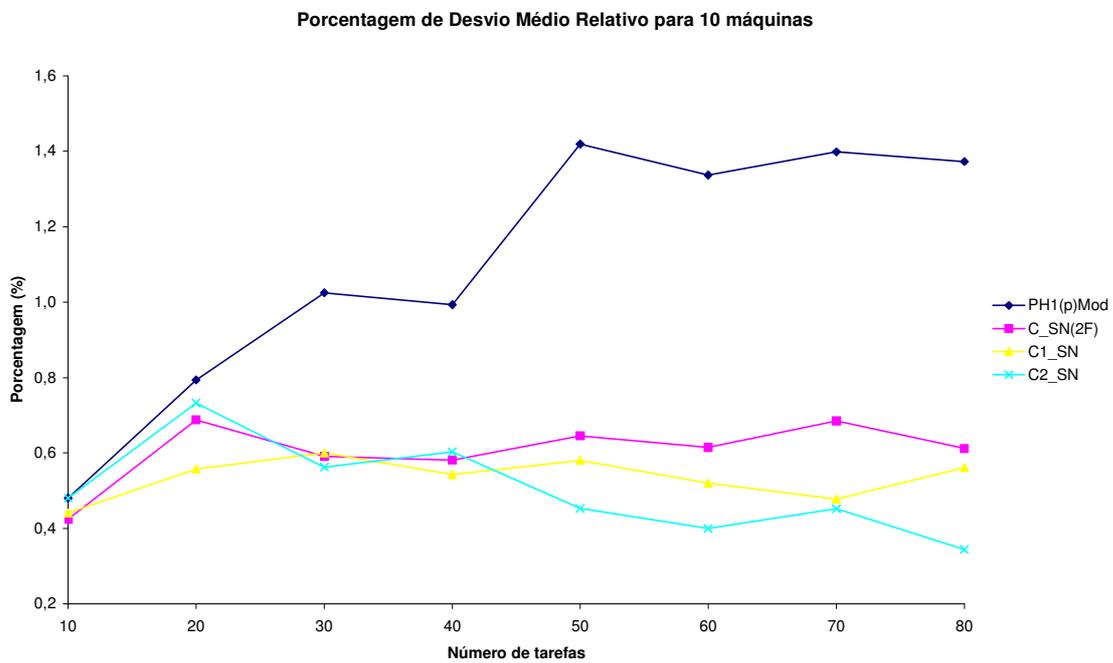


Figura 6.22– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas

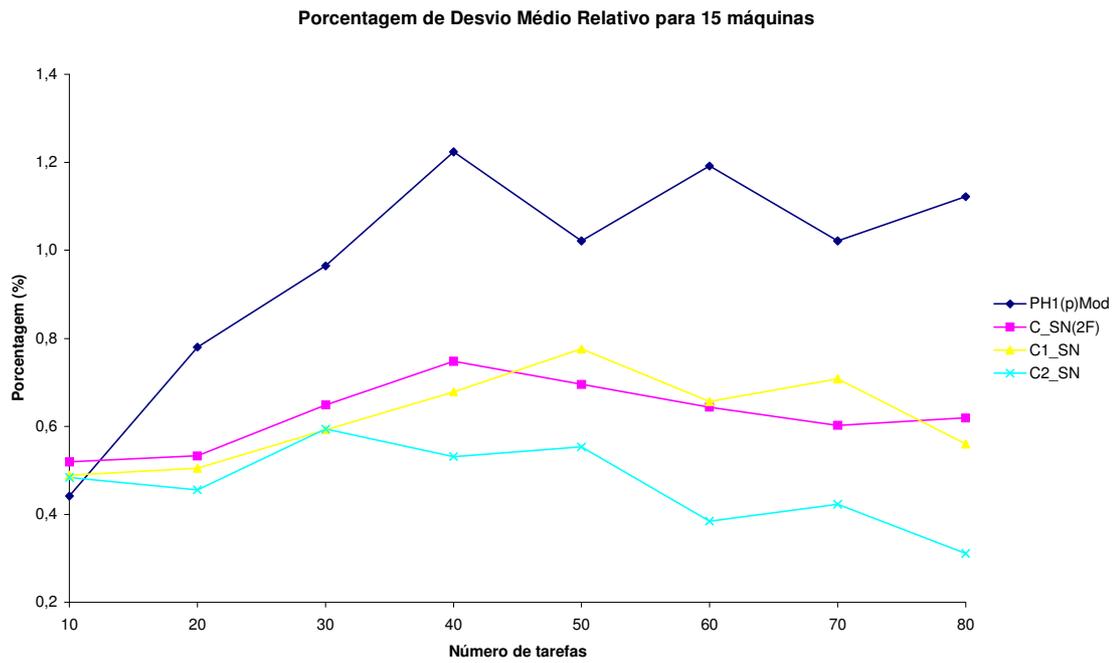


Figura 6.23– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas

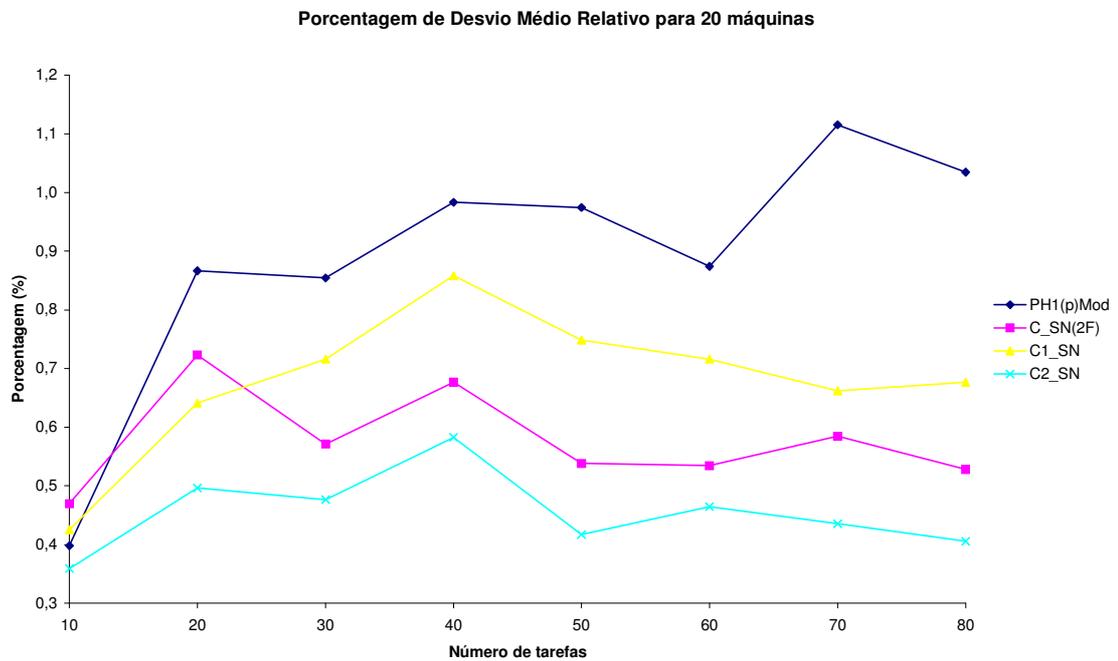


Figura 6.24– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas

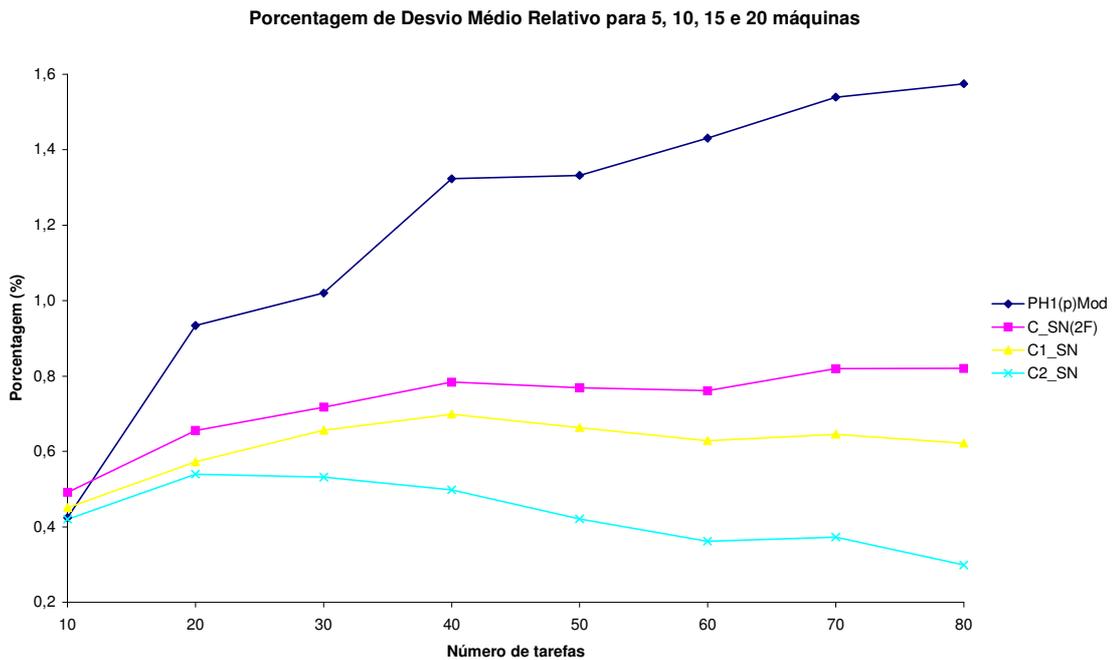


Figura 6.25– Porcentagem de desvio médio relativo vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

De forma consolidada, a figura 6.25 evidencia a superioridade do método proposto C2_SN.

É uma porcentagem de desvio médio relativo abaixo de 0,6, seguido pelo segundo melhor método, o algoritmo C1_SN. Os outros dois algoritmos apresentam porcentagens de desvio médio relativo maiores, entre de 0,7% e 1,6% para problemas acima de 40 tarefas.

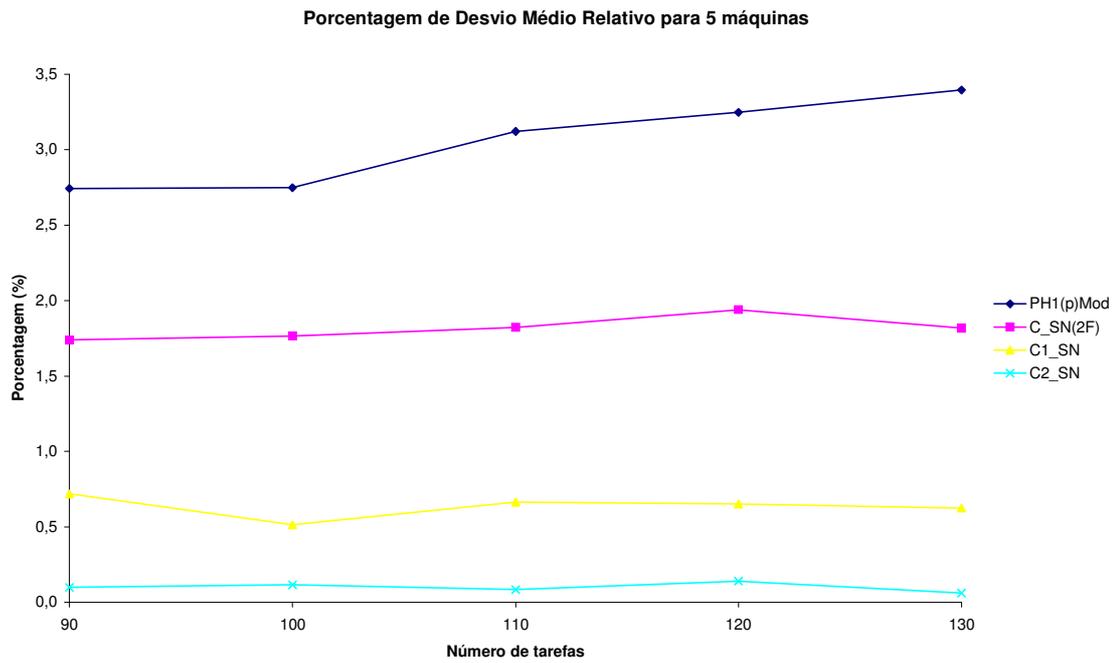


Figura 6.26– Porcentagem de desvio médio relativo vs. Número de tarefas, para 5 máquinas

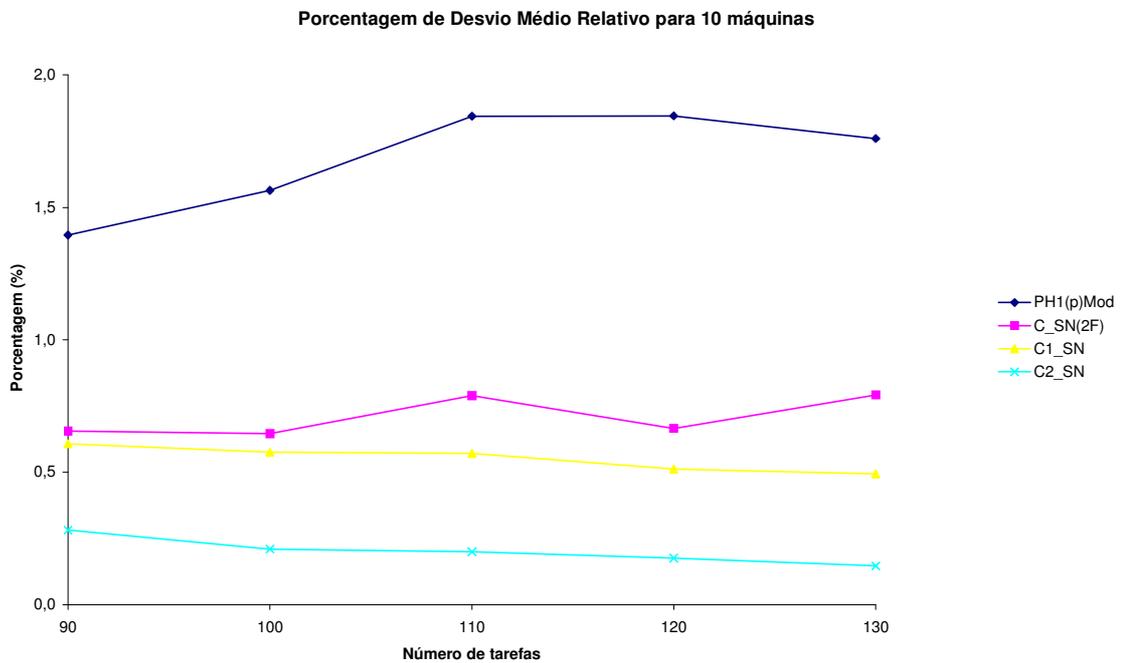


Figura 6.27– Porcentagem de desvio médio relativo vs. Número de tarefas, para 10 máquinas

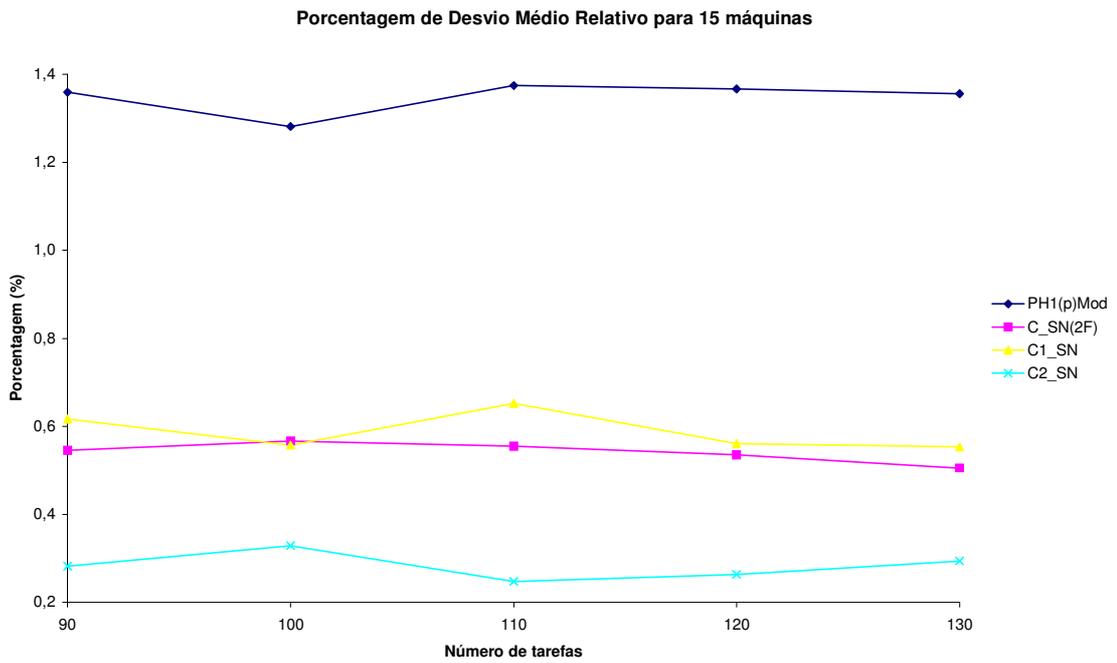


Figura 6.28– Porcentagem de desvio médio relativo vs. Número de tarefas, para 15 máquinas

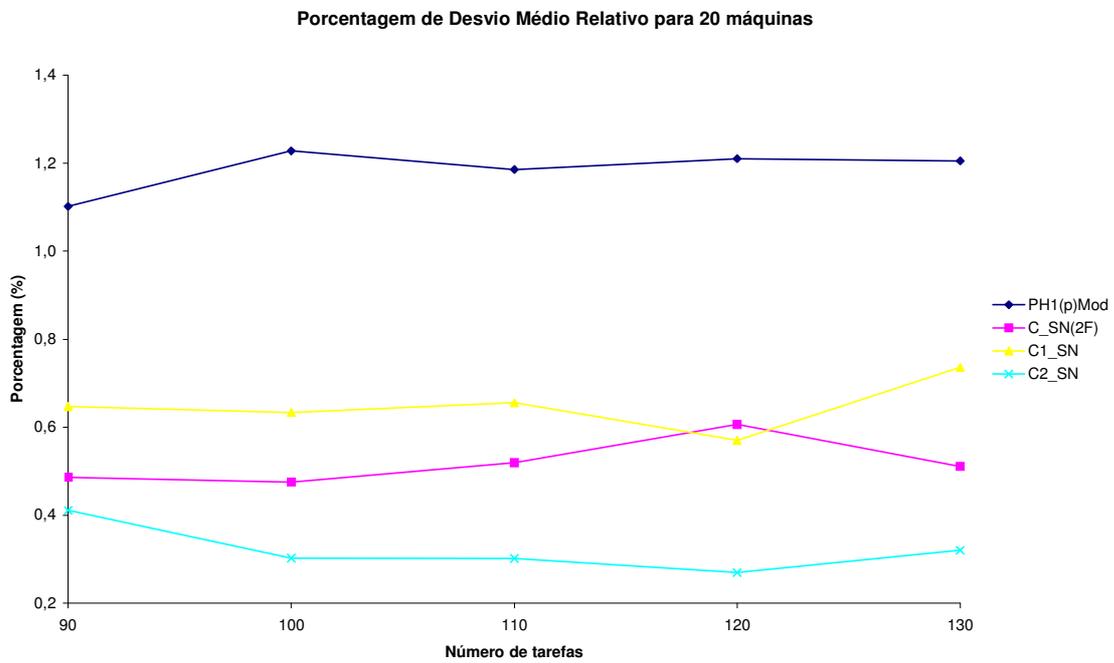


Figura 6.29– Porcentagem de desvio médio relativo vs. Número de tarefas, para 20 máquinas

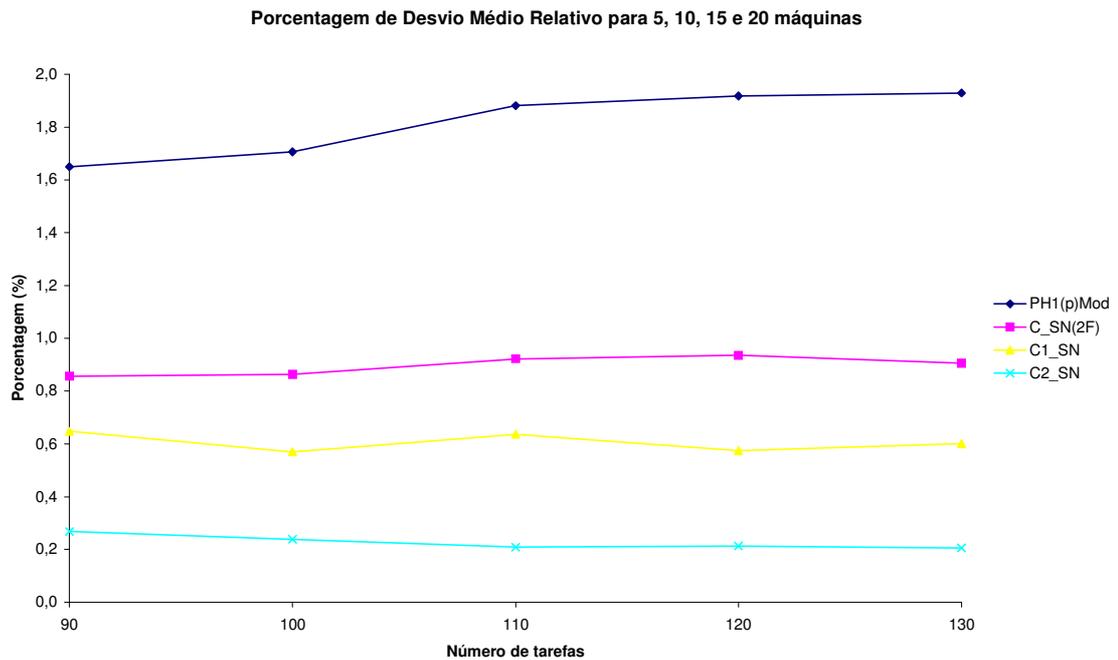


Figura 6.30– Porcentagem de desvio médio relativo vs. Número de tarefas, para m Máquinas agrupadas (5, 10, 15 e 20).

Para problemas de grande porte, fica confirmada a tendência demonstrada nos problemas de médio porte. O algoritmo C2_SN apresenta porcentagem de desvio médio relativo abaixo de 0,3%. Enquanto a versão melhorada (modificada) do algoritmo PH1(p), aqui denominado PH1(p)Mod, é o pior, conforme figura 6.30, com porcentagens de desvio médio relativo acima de 1,6%.

6.1.3 Tempo Médio de Computação

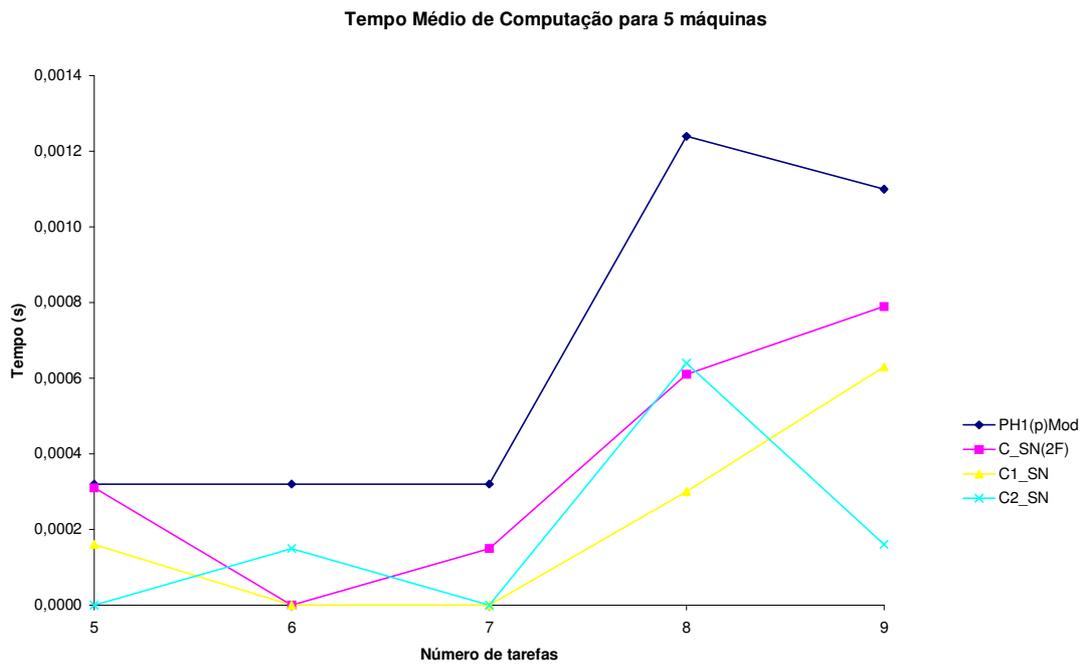


Figura 6.31– Tempo médio de computação vs. Número de tarefas, para 5 máquinas

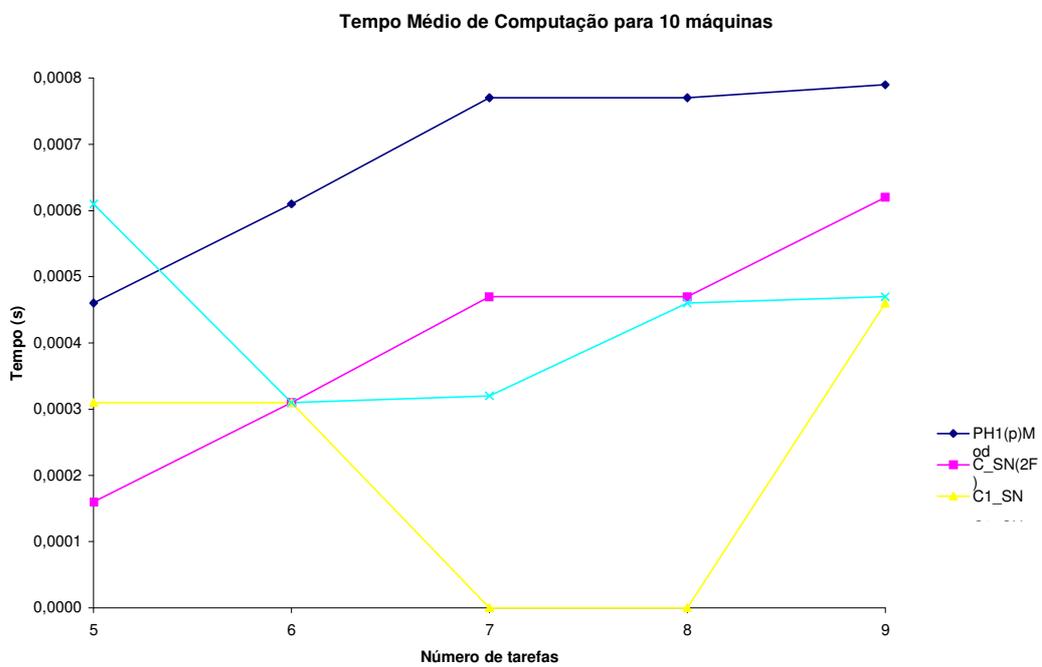


Figura 6.32– Tempo médio de computação vs. Número de tarefas, para 10 máquinas

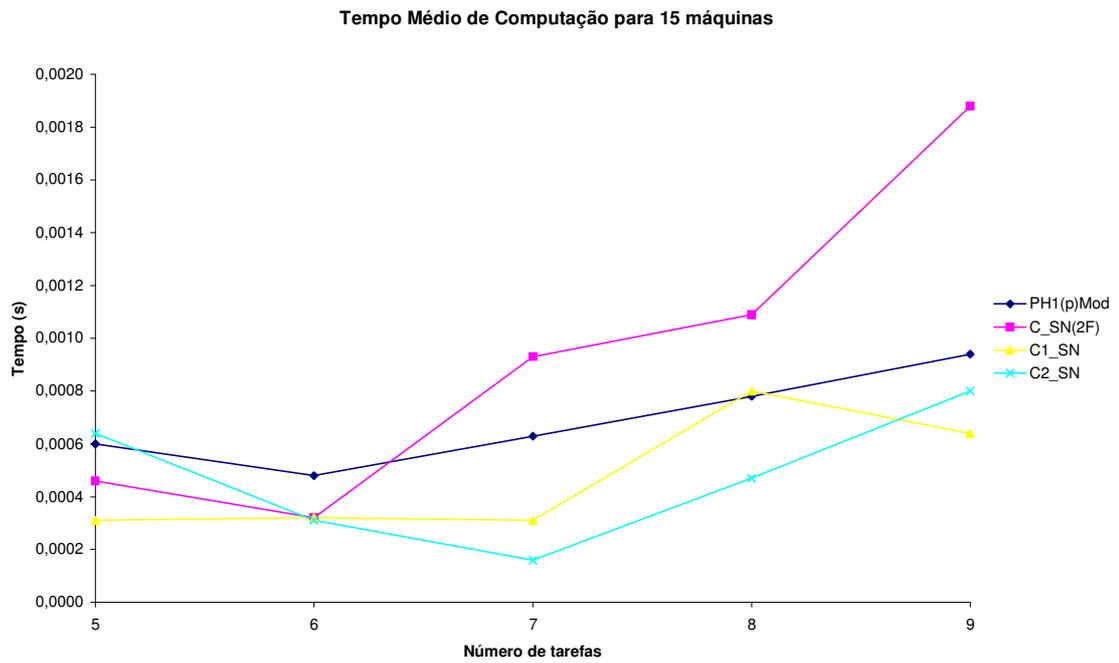


Figura 6.33– Tempo médio de computação vs. Número de tarefas, para 15 máquinas

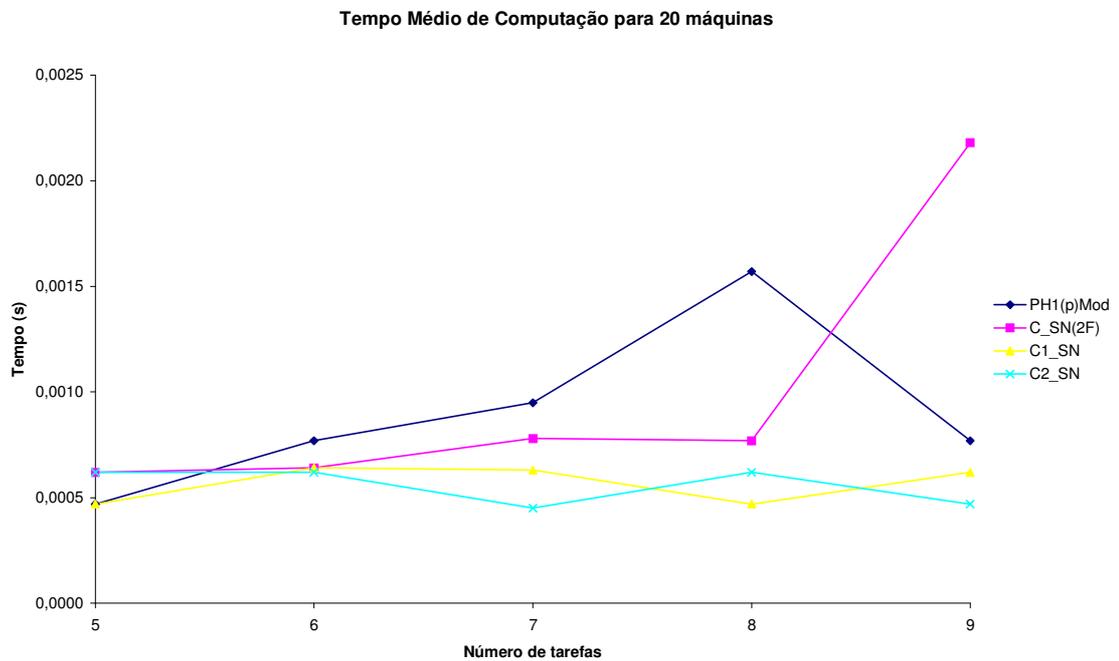


Figura 6.34– Tempo médio de computação vs. Número de tarefas, para 20 máquinas

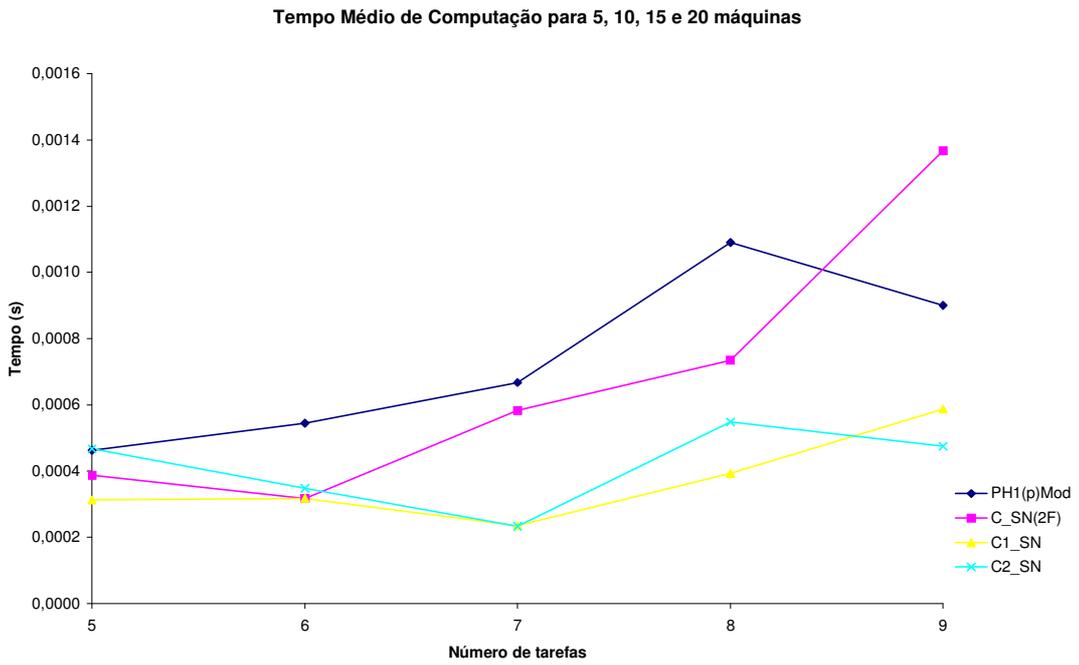


Figura 6.35– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5,10, 15 e 20).

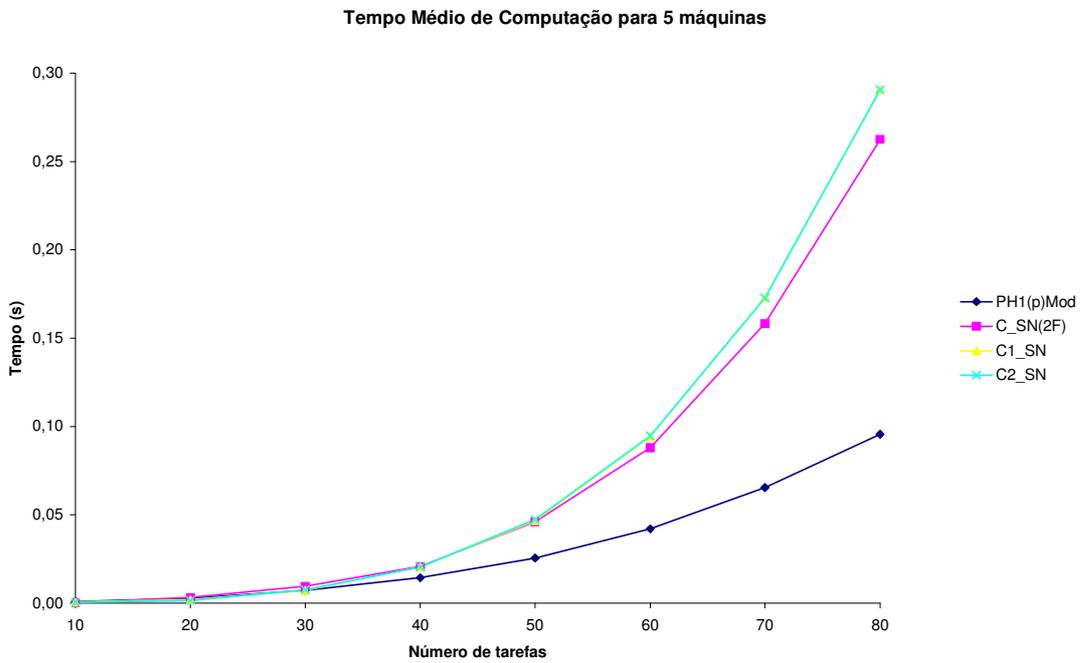


Figura 6.36– Tempo médio de computação vs. Número de tarefas, para 5 máquinas

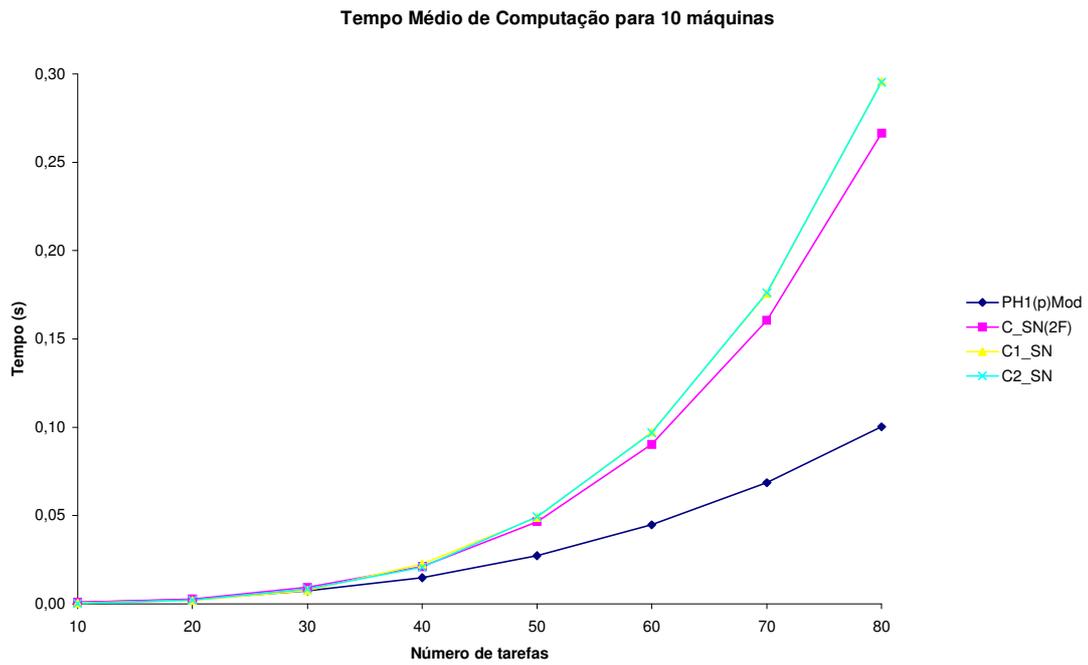


Figura 6.37– Tempo médio de computação vs. Número de tarefas, para 10 máquinas

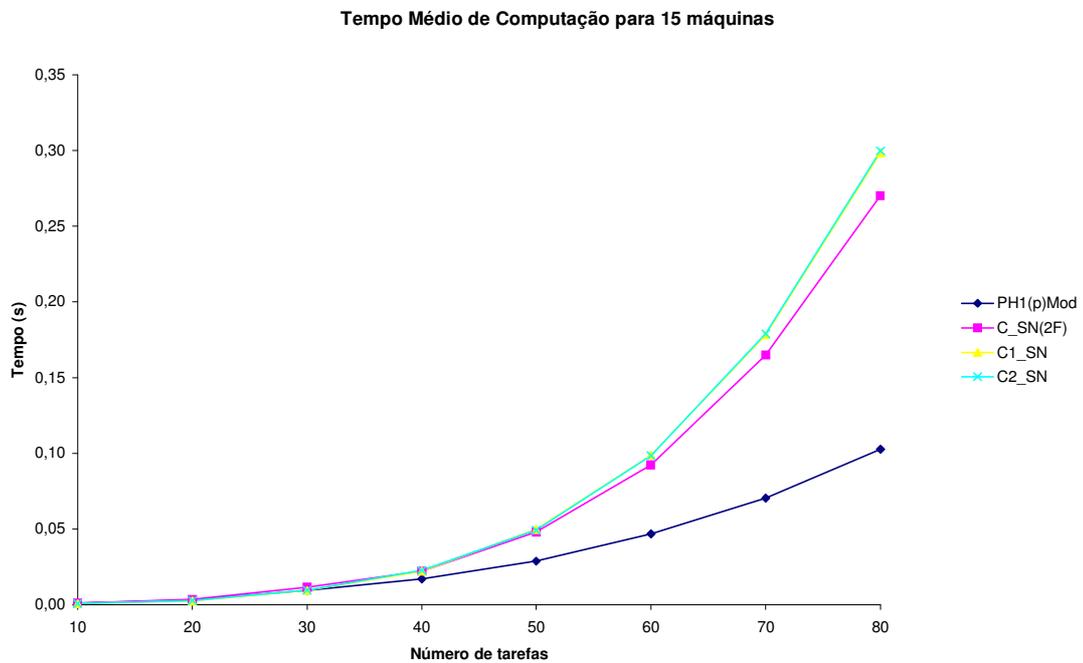


Figura 6.38– Tempo médio de computação vs. Número de tarefas, para 15 máquinas

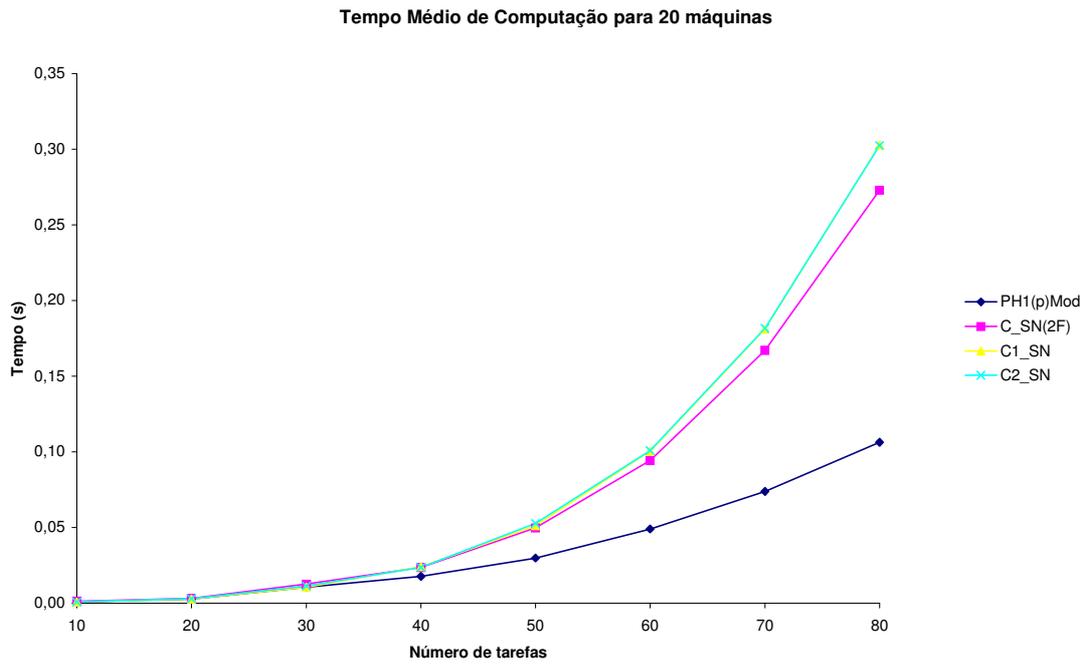


Figura 6.39– Tempo médio de computação vs. Número de tarefas, para 20 máquinas

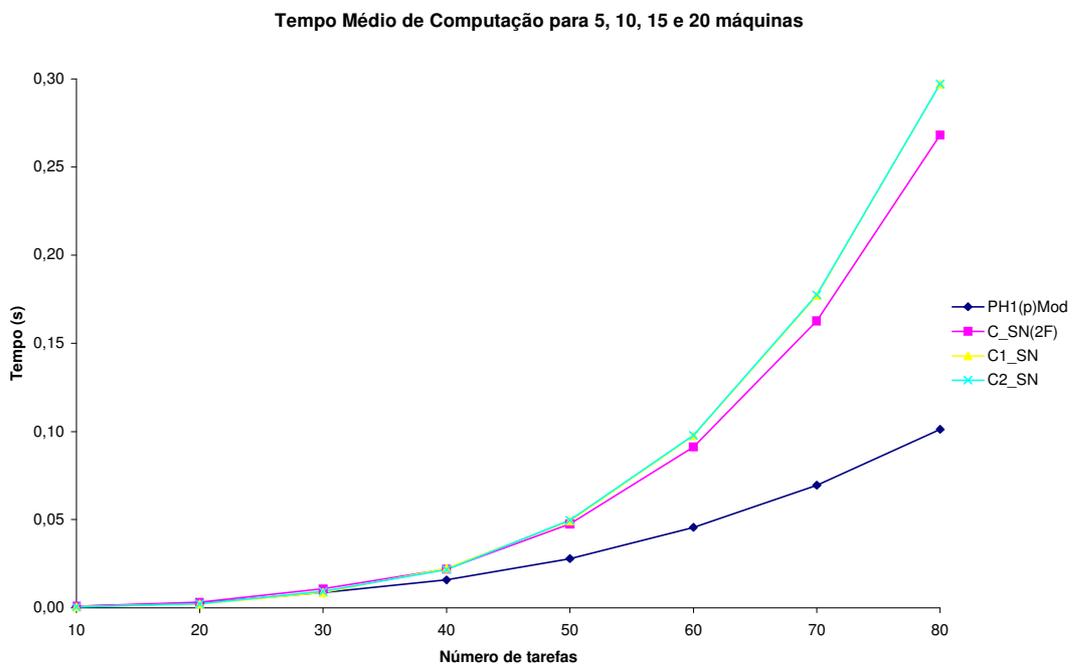


Figura 6.40– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

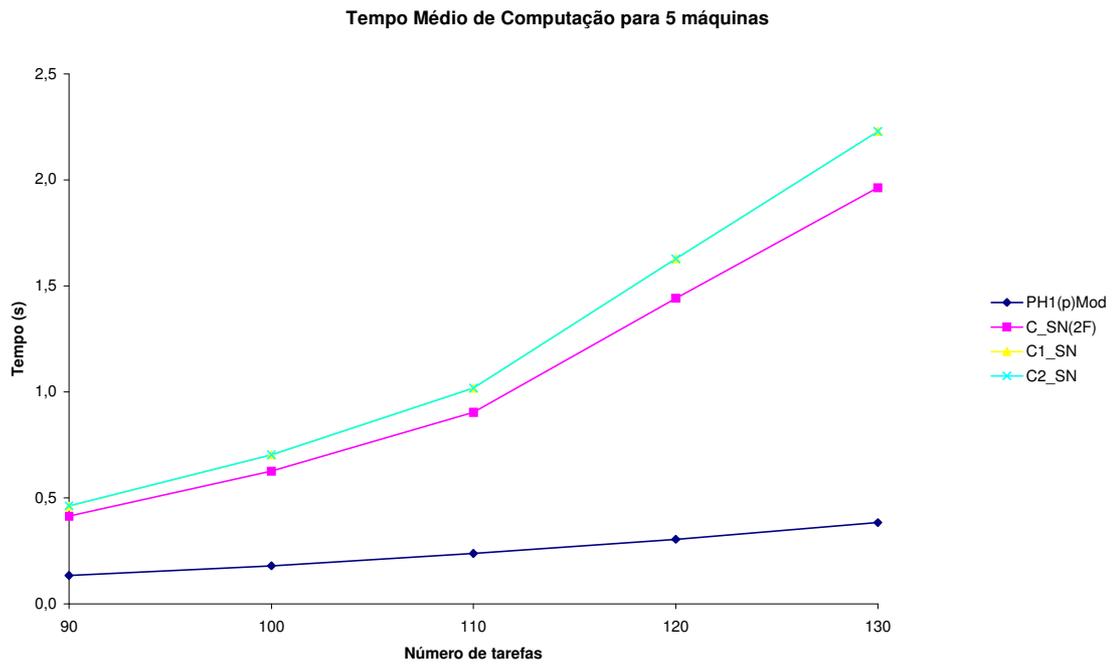


Figura 6.41– Tempo médio de computação vs. Número de tarefas, para 5 máquinas

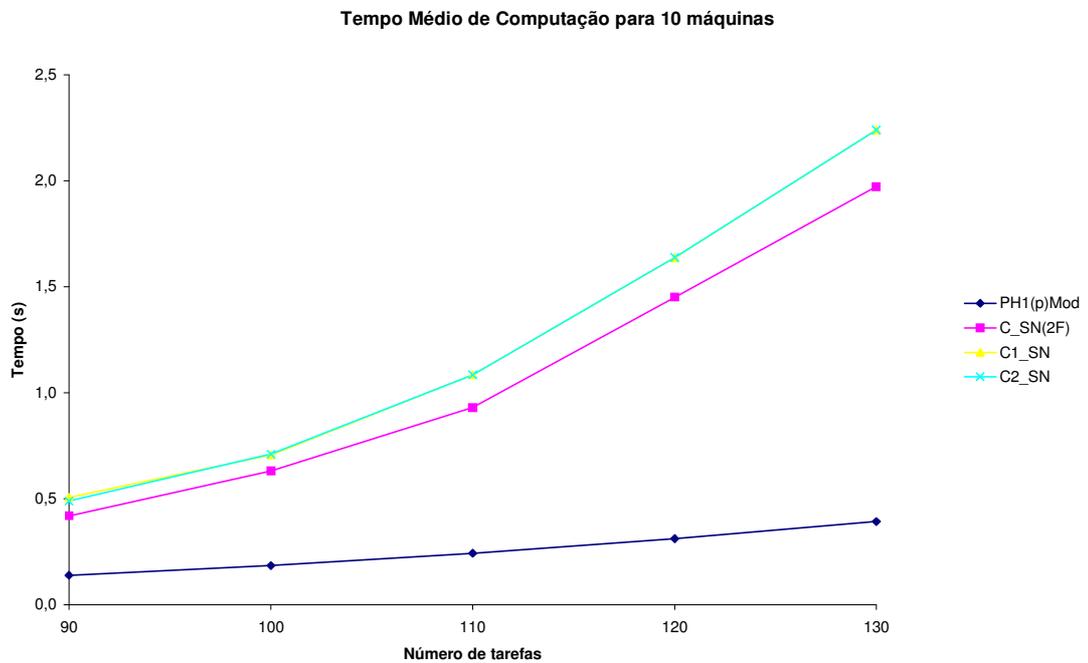


Figura 6.42– Tempo médio de computação vs. Número de tarefas, para 10 máquinas

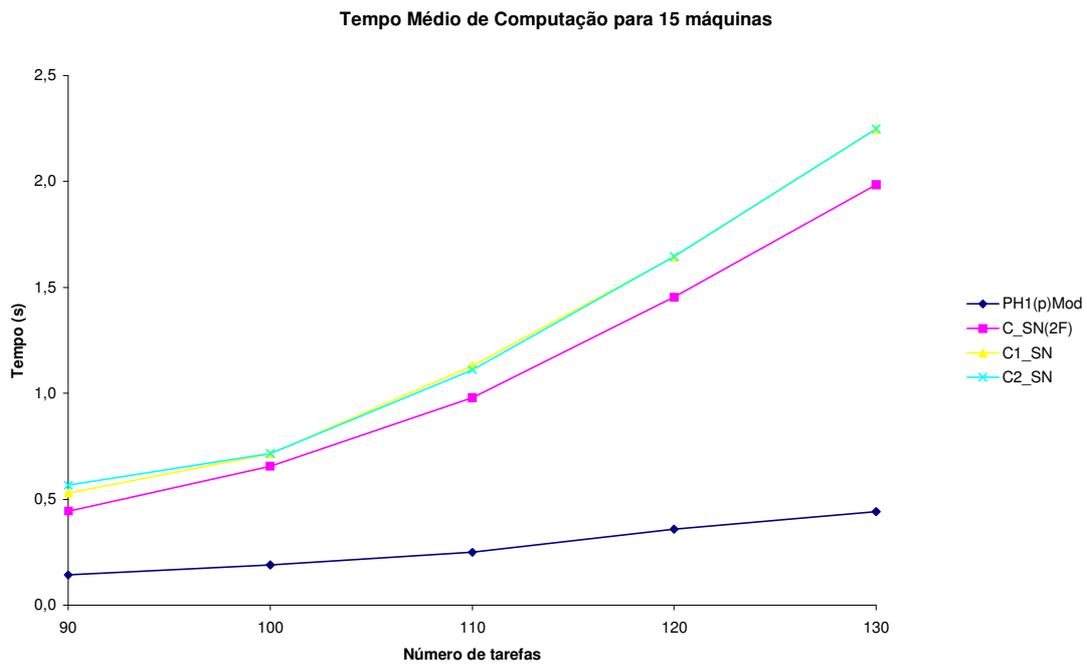


Figura 6.43– Tempo médio de computação vs. Número de tarefas, para 15 máquinas

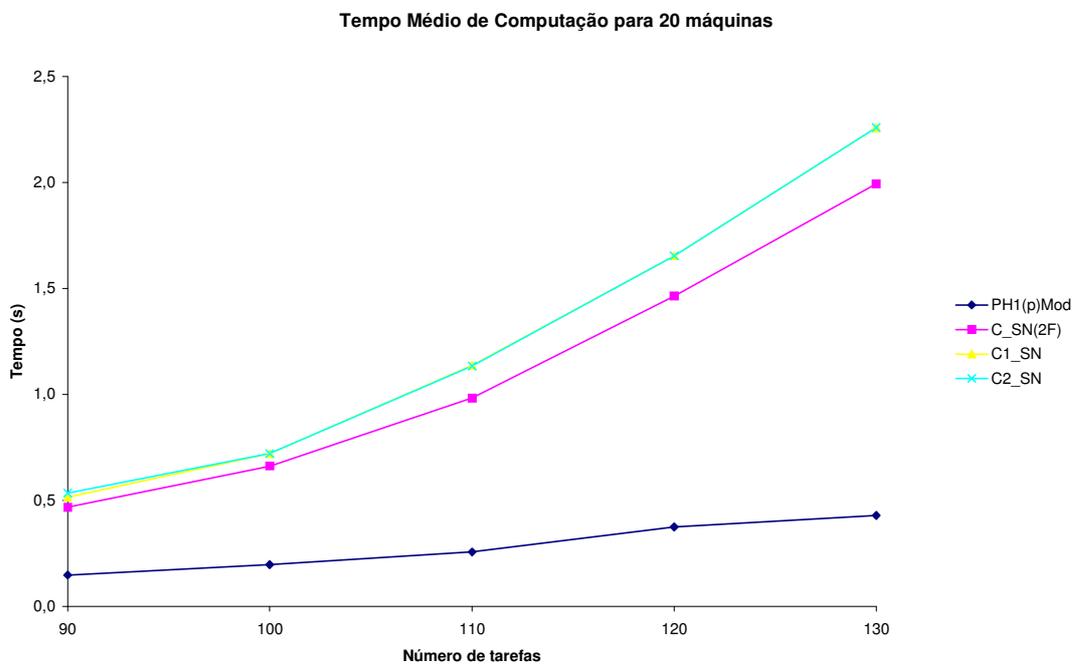


Figura 6.44– Tempo médio de computação vs. Número de tarefas, para 20 máquinas

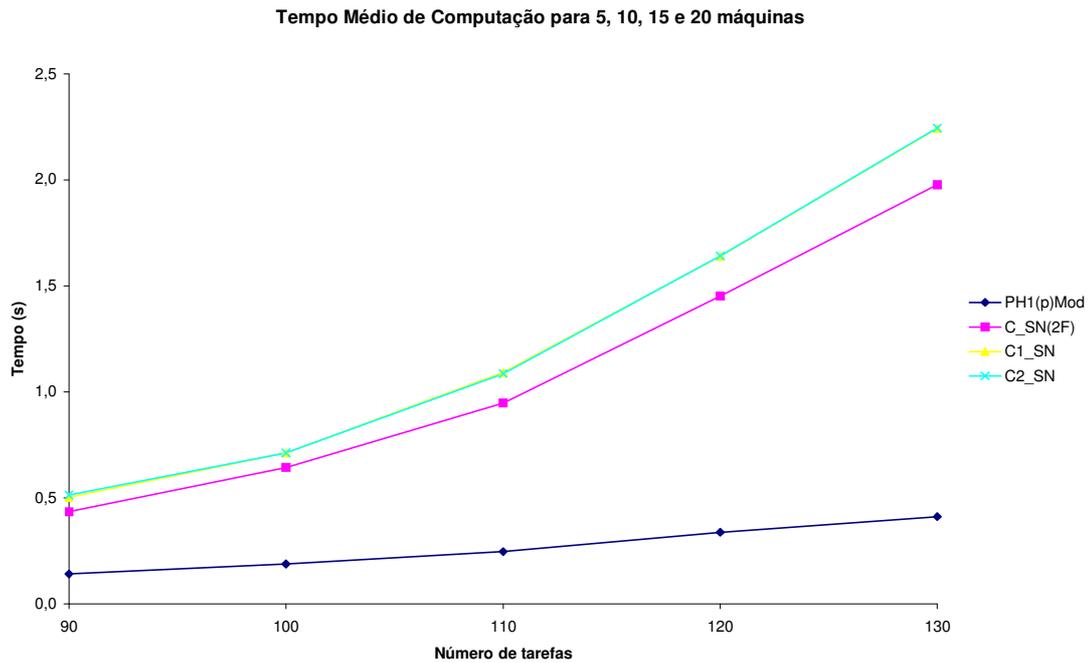


Figura 6.45– Tempo médio de computação vs. Número de tarefas, para máquinas agrupadas (5, 10, 15 e 20).

O tempo médio de computação para os problemas de médio e grande porte são maiores para os métodos propostos C2_SN, C1_SN e C_SN(2F) se comparados ao algoritmo PH1(p)Mod. Enquanto este apresenta tempo médio de computação de menos de 0,5 segundos para problemas de 130 tarefas, os demais consomem entre 1,9 e 2,3 segundos.

Porém, mesmo que quase 5 vezes mais lento que o método modificado PH1(1)Mod, o tempo médio de computação do melhor método C2_SN não compromete a viabilidade das soluções.

Concluídas as experimentações computacionais, no capítulo seguinte serão apresentadas as considerações finais do trabalho.

CAPÍTULO 7

Considerações Finais

Preliminarmente, ressalta-se que para fins práticos, as soluções obtidas pelos métodos heurísticos já existentes, para o problema NWFS, são suficientes, ou seja, tal problema pode ser considerado como já resolvido.

Porém, tendo em vista a complexidade do problema em questão, a busca por novos métodos que contêm adequado equilíbrio entre a qualidade da solução e a eficiência computacional, simplicidade e facilidade de implementação, ainda continua como uma direção para novas pesquisas.

Dessa forma, este trabalho apresenta uma pequena contribuição que procura evidenciar que apesar dos algoritmos existentes proporcionarem boas soluções, é possível, por meio da propriedade já existente, referenciada e adaptada em Bertolissi (2000) e de uma estruturação diferenciada dos métodos heurísticos (fase única), obter soluções melhores e em tempo computacional viável.

O principal aspecto a ser destacado neste trabalho refere-se ao desenvolvimento de métodos heurísticos de fase única (C2_SN e C1_SN) e de duas fases (C_SN(2F)), que demonstraram, através das experimentações computacionais, ser superiores aos melhores métodos heurísticos de duas e três fases já existentes e suas modificações sugeridas.

Dentre estes métodos propostos, os resultados experimentais mostraram também que o método heurístico proposto C2_SN tem um desempenho superior aos demais, para solução do problema em questão.

O “passo 3” é a principal parte da estrutura do algoritmo do método proposto C2_SN, uma vez que nele está contida a combinação dos dois métodos de melhoria, que possibilita a avaliação, a cada iteração, de uma maior variedade de seqüências parciais. Aliado a este passo, o passo anterior permite a utilização de uma propriedade muito interessante que auxilia a escolha da próxima tarefa a ser seqüenciada, através da análise da matriz F, que contém as parcelas das somas dos tempos de fluxo de duas tarefas adjacentes J_i e J_j , a partir do início de J_i (ou seja, no intervalo de tempo entre o início de J_i e o término de J_j).

Trata-se de uma combinação entre uma propriedade existente e uma estrutura diferenciada, cuja experimentação apresentou um tempo computacional maior em relação aos métodos já existentes, porém tal aumento é compensado pela facilidade de implementação do método e pela melhoria na qualidade das soluções.

Ainda com relação à experimentação computacional, uma consideração interessante não pode deixar de ser pautada. Quando avaliados separadamente os métodos Bert, Raj1 e Raj2, ficou demonstrado que o método proposto por Bertolissi (2000) apenas equipara-se aos métodos propostos por Rajendran e Chaudhuri (1990) para problemas de pequeno porte, sendo inferior nos problemas de médio e grande porte. Assim, diferentemente da conclusão apresentada por Bertolissi (2000), seu método não é superior aos métodos Raj1 e Raj2.

No futuro, as pesquisas poderiam observar, por exemplo, os efeitos da aplicação do “passo 3” do método proposto C2_SN ao problema Flow Shop Permutacional e os efeitos da aplicação também em algoritmos metaheurísticos para o problema Flow Shop Permutacional.

Além disso, pesquisas futuras poderiam observar a combinação do “passo 3” com outras propriedades ou outras formas de seleção das próximas tarefas a serem inseridas, ao invés da propriedade adotada referenciada em Bertolissi (2000).

Para finalizar, ressalta-se a estrutura diferenciada do principal método proposto. Ela não segue a rigidez das estruturas dos métodos heurísticos observados até então: ordenação inicial, construção e melhoria da solução. Este trabalho evidencia que a forma de um método heurístico pode ser mais flexível se os resultados apresentados forem mais interessantes, assim como no caso do C2_SN.

REFERÊNCIAS BIBLIOGRÁFICAS

ALDOWAISAN T.; ALLAHVERDI A. (2003). New heuristics for no-wait flowshops to minimize makespan. *Computers e Operations Research*, v.30, p.1219-1231.

_____. (2004). New heuristics for m-machine no-wait flowshop to minimize total completion time. *OMEGA – The International Journal of Management Science*, v.32, p.345- 352.

ALLAHVERDI, A.; ALDOWAISAN T. (2002). No-wait flowshops with bicriteria of makespan and total completion time. *Journal of the Operational Research Society*, v.53, p.1004-1015.

BAKER, K.R. (1974). *Introduction to sequencing and scheduling*. New York: John Wiley & Sons, Inc.

BARROS A. D. (2002). *Algoritmo metaheurístico para busca do gargalo flutuante em flow shop permutacional com tempos de setup assimétricos e dependentes da seqüência*. 105p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 2002.

BERTOLISSI, E. (1999). A simple no-wait flowshop scheduling heuristic for the no-wait flowshop problem. *Proceedings of the 15th International Conference on Computer-Aided Production Engineering, CAPE'99*, University of Durham Publishers, p.750-755.

_____. (2000). Heuristic algorithm for scheduling in th no-wait flow-shop. *Journal of Materials Processing Technology*, v.107, p.459-465.

BONNEY, M.C.; GUNDRY S.W. (1976). Solutions to the Constrained FlowShop Sequencing Problem. *Operations Research Quarterly*, v.24, p.869-883.

CHEN, C.; NEPPALLI, R.V.; ALJABER N. (1996). Genetic Algorithms Applied to the Continuous Flow Shop Problem. *Computers Industrial Engineers*, v.30, n.4, p. 919-929.

DEMAN, J.M.V.; BAKER K.R. (1974). Minimizing Mean Flowtime in the Flow Shop with No Intermediate Queues. *AIIE Transactions*, v.6, n.1, p.28-34.

FINK, A.; VOß, S. (2003). Solving the continuous flow-shop scheduling problem by metaheuristics. *European Journal of Operational Research*, v.151, p.400-414.

FRAMINAN, J.M.; LEISTEN R. (2003). An efficient constructive heuristic for flowtime minimisation in permutation flow shops. *OMEGA – The International Journal of Management Science*, v.31, p.311-317.

GANGADHARAN, R.; RAJENDRAN C. (1993). Heuristic algorithms for scheduling in the no-wait flowshop. *International Journal of Production Economics*, v.32, p.285-290.

JOHNSON S.M.; MONTGOMERY D.C. (1974). *Operations Research in Production, Planning, Scheduling and Inventory Control*, Wiley, New York.

KING, R.; SPACHIS, A.S. (1980). Heuristics for FlowShop Scheduling. *International Journal of Production Research*, v.18, p.343-357.

LEE, C.; LEI, L; PINEDO, M. (1997). Current trends in deterministic scheduling. *Annals of Operations Research* 70, p.1-41.

MOCCELLIN J.V. (1994). *Técnicas de Seqüenciamento e Programação de Operações em Máquinas*. 79p. Publicação 013/94 Escola de Engenharia de São Carlos, Universidade de São Paulo, Apostila.

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

NAGANO, M.S. (1995). *Novos procedimentos de Busca Tabu para o Problema de Programação de Operações Flow-Shop Permutacional*. 117p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 1995.

NAGANO, M.S. (1999). *Um novo método heurístico construtivo de alto desempenho para a programação de operações flow shop permutacional*. 169p. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 1999.

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

NAGANO, M.S.; MOCCELLIN, J.V.; LORENA, L.A.N. (2004). *Programação da Produção Flow Shop Permutacional com Minimização do Tempo Médio de Fluxo*. In: XXXVI Simpósio Brasileiro de Pesquisa Operacional 2004, São João Del-Rei - MG. Anais do XXXVI SBPO, 2004.

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

OSMAN, I.H., POTTS, C.N. (1989). Simulated Annealing for Permutation Flow-Shop Scheduling. *OMEGA - The International Journal of Management Science*, v.17, p.551-557.

PINEDO, M. (1995). *Scheduling: theory, algorithms, and systems*. New Jersey: Prentice-Hall

RAJENDRAN, C. (1994). A no-wait Flowshop Scheduling Heuristic to Minimize Makespan. *J. Operational Research Society*, v.45, n.4, p.472-478.

RAJENDRAN, C.; CHAUDHURI D. (1990). Heuristic Algorithms for Continuous Flow-Shop Problem. *Naval Research Logistics*, v.37, p.695-705.

REISMAN, A.; KUMAR A.; MOTWANI J. (1997). Flowshop Scheduling/Sequencing

Research: A Statistical Review of the Literature, 1952-1994. *IEEE Transactions on Engineering Management*, v.44, n.3, p.316-329.

SLACK, N. et al. (1999). *Administração da Produção*. Edição Compacta. São Paulo: Atlas.

WIDMER, M. and HERTZ, A. (1989). A New Heuristic Method for the Flow-Shop Sequencing Problem. *European Journal of Operational Research*, v.41, p.186-193.

ANEXO A - Tabelas Referentes às Experimentações Computacionais

Experimentação Computacional - Parte A

Tabela 1: Porcentagem de Sucesso para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	77,00	70,00	75,00	71,00	94,00
6	65,00	67,00	72,00	68,00	93,00
7	53,00	57,00	60,00	60,00	89,00
8	33,00	35,00	35,00	36,00	87,00
9	36,00	41,00	41,00	41,00	82,00

Tabela 2: Porcentagem de Sucesso para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	72,00	69,00	72,00	69,00	93,00
6	59,00	61,00	59,00	61,00	92,00
7	51,00	50,00	52,00	54,00	88,00
8	49,00	39,00	47,00	43,00	75,00
9	36,00	38,00	34,00	40,00	80,00

Tabela 3: Porcentagem de Sucesso para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	69,00	72,00	70,00	74,00	95,00
6	58,00	60,00	62,00	62,00	86,00
7	52,00	46,00	51,00	49,00	85,00
8	46,00	37,00	42,00	38,00	81,00
9	34,00	35,00	31,00	38,00	71,00

Tabela 4: Porcentagem de Sucesso para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	74,00	75,00	79,00	77,00	95,00
6	69,00	64,00	68,00	67,00	92,00
7	59,00	55,00	56,00	58,00	81,00
8	52,00	44,00	51,00	45,00	73,00
9	34,00	29,00	35,00	36,00	82,00

Tabela 5: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	73,00	71,50	74,00	72,75	94,25
6	62,75	63,00	65,25	64,50	90,75
7	53,75	52,00	54,75	55,25	85,75
8	45,00	38,75	43,75	40,50	79,00
9	35,00	35,75	35,25	38,75	78,75

Tabela 6: Porcentagem de Sucesso para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	32,00	25,00	29,00	27,00	78,00
20	6,00	9,00	8,00	10,00	78,00
30	3,00	4,00	3,00	6,00	87,00
40	0,00	4,00	0,00	8,00	92,00
50	1,00	2,00	0,00	3,00	95,00
60	0,00	1,00	0,00	2,00	97,00
70	0,00	0,00	0,00	0,00	100,00
80	0,00	0,00	0,00	0,00	100,00

Tabela 7: Porcentagem de Sucesso para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	25,00	30,00	30,00	32,00	76,00
20	8,00	11,00	11,00	14,00	67,00
30	4,00	4,00	5,00	5,00	86,00
40	1,00	4,00	2,00	7,00	90,00
50	0,00	0,00	1,00	2,00	97,00
60	1,00	0,00	0,00	3,00	96,00
70	0,00	1,00	1,00	0,00	98,00
80	0,00	1,00	0,00	0,00	99,00

Tabela 8: Porcentagem de Sucesso para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	32,00	26,00	30,00	28,00	79,00
20	10,00	10,00	11,00	14,00	67,00
30	4,00	3,00	4,00	16,00	76,00
40	2,00	3,00	5,00	11,00	81,00
50	0,00	3,00	2,00	6,00	90,00
60	1,00	2,00	0,00	4,00	93,00
70	0,00	0,00	0,00	2,00	98,00
80	0,00	0,00	0,00	0,00	100,00

Tabela 9: Porcentagem de Sucesso para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	37,00	25,00	37,00	28,00	69,00
20	8,00	17,00	8,00	25,00	60,00
30	3,00	2,00	1,00	7,00	89,00
40	3,00	1,00	3,00	5,00	89,00
50	1,00	0,00	1,00	3,00	95,00
60	0,00	0,00	1,00	1,00	98,00
70	0,00	0,00	0,00	5,00	95,00
80	0,00	0,00	0,00	1,00	99,00

Tabela 10: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	31,50	26,50	31,50	28,75	75,50
20	8,00	11,75	9,50	15,75	68,00
30	3,50	3,25	3,25	8,50	84,50
40	1,50	3,00	2,50	7,75	88,00
50	0,50	1,25	1,00	3,50	94,25
60	0,50	0,75	0,25	2,50	96,00
70	0,00	0,25	0,25	1,75	97,75
80	0,00	0,25	0,00	0,25	99,50

Tabela 11: Porcentagem de Sucesso para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,00	1,00	0,00	0,00	99,00
100	0,00	0,00	0,00	1,00	99,00
110	0,00	0,00	0,00	0,00	100,00
120	0,00	0,00	0,00	0,00	100,00
130	0,00	0,00	0,00	0,00	100,00

Tabela 12: Porcentagem de Sucesso para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,00	0,00	0,00	0,00	100,00
100	0,00	0,00	0,00	1,00	99,00
110	1,00	0,00	0,00	0,00	99,00
120	0,00	0,00	0,00	0,00	100,00
130	0,00	0,00	0,00	0,00	100,00

Tabela 13: Porcentagem de Sucesso para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,00	0,00	0,00	2,00	98,00
100	0,00	0,00	0,00	1,00	99,00
110	0,00	0,00	0,00	0,00	100,00
120	0,00	0,00	0,00	0,00	100,00
130	0,00	0,00	0,00	0,00	100,00

Tabela 14: Porcentagem de Sucesso para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,00	0,00	0,00	2,00	98,00
100	0,00	0,00	0,00	2,00	98,00
110	0,00	0,00	0,00	0,00	100,00
120	0,00	0,00	0,00	0,00	100,00
130	0,00	0,00	0,00	2,00	98,00

Tabela 15: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,00	0,25	0,00	1,00	98,75
100	0,00	0,00	0,00	1,25	98,75
110	0,25	0,00	0,00	0,00	99,75
120	0,00	0,00	0,00	0,00	100,00
130	0,00	0,00	0,00	0,50	99,50

Tabela 16: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,29	0,35	0,29	0,34	0,08
6	0,53	0,72	0,44	0,63	0,05
7	0,64	0,50	0,48	0,45	0,12
8	1,16	1,08	1,04	1,03	0,12
9	1,26	1,08	0,98	0,91	0,19

Tabela 17: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,25	0,38	0,29	0,38	0,16
6	0,49	0,60	0,52	0,57	0,14
7	0,73	0,62	0,63	0,49	0,17
8	0,68	0,98	0,75	0,84	0,36
9	0,77	0,70	0,82	0,63	0,25

Tabela 18: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,23	0,24	0,21	0,19	0,05
6	0,34	0,48	0,33	0,36	0,13
7	0,56	0,62	0,53	0,52	0,16
8	0,58	0,72	0,64	0,60	0,19
9	0,78	1,00	0,81	0,82	0,26

Tabela 19: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,16	0,16	0,13	0,14	0,01
6	0,30	0,35	0,29	0,30	0,05
7	0,37	0,57	0,45	0,49	0,29
8	0,40	0,62	0,39	0,53	0,21
9	0,90	0,89	0,84	0,72	0,23

Tabela 20: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,23	0,28	0,23	0,26	0,08
6	0,42	0,54	0,39	0,47	0,09
7	0,57	0,58	0,52	0,49	0,18
8	0,70	0,85	0,70	0,75	0,22
9	0,93	0,92	0,86	0,77	0,23

Tabela 21: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	1,11	1,39	1,14	1,14	0,25
20	2,47	1,84	2,47	1,60	0,22
30	3,24	2,29	3,03	1,75	0,08
40	3,93	2,58	3,93	1,97	0,06
50	4,08	2,81	4,14	2,35	0,03
60	4,62	2,98	4,61	2,44	0,02
70	4,93	2,98	4,82	2,47	0,00
80	5,14	2,97	5,02	2,45	0,00

Tabela 22: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	1,31	1,11	1,05	0,93	0,26
20	2,11	1,80	1,68	1,35	0,23
30	2,84	2,24	2,51	1,72	0,13
40	3,36	2,80	3,30	1,93	0,06
50	3,57	2,74	3,16	2,03	0,03
60	3,59	2,82	3,54	2,01	0,02
70	3,95	2,93	3,62	2,12	0,02
80	4,21	2,87	3,77	1,99	0,00

Tabela 23: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,88	0,89	0,86	0,77	0,33
20	1,64	1,78	1,54	1,29	0,25
30	2,39	2,19	2,16	1,45	0,11
40	2,92	2,52	2,66	1,70	0,09
50	3,14	2,77	2,96	1,84	0,07
60	3,44	2,73	3,15	1,78	0,03
70	3,61	2,87	3,18	1,84	0,01
80	3,69	2,92	3,44	1,91	0,00

Tabela 24: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,66	0,85	0,54	0,59	0,31
20	1,74	1,85	1,76	1,33	0,35
30	2,68	2,21	2,49	1,59	0,08
40	2,74	2,55	2,73	1,75	0,09
50	3,39	2,76	3,09	2,04	0,02
60	3,37	3,00	3,17	1,98	0,01
70	3,77	3,11	3,41	2,06	0,02
80	3,81	3,11	3,57	1,91	0,00

Tabela 25: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,99	1,06	0,90	0,86	0,29
20	1,99	1,82	1,86	1,39	0,26
30	2,79	2,23	2,54	1,63	0,10
40	3,24	2,61	3,16	1,84	0,08
50	3,55	2,77	3,34	2,06	0,04
60	3,75	2,88	3,62	2,05	0,02
70	4,07	2,97	3,76	2,12	0,01
80	4,21	2,97	3,95	2,06	0,00

Tabela 26: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	5,42	3,06	5,30	2,57	0,02
100	5,30	3,04	5,09	2,52	0,00
110	5,60	3,16	5,44	2,63	0,00
120	5,63	3,23	5,62	2,58	0,00
130	6,00	3,32	5,65	2,75	0,00

Tabela 27: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	4,06	2,93	3,80	2,08	0,00
100	4,31	3,08	3,92	2,20	0,00
110	4,41	2,93	3,91	2,29	0,00
120	4,54	3,03	4,02	2,26	0,00
130	4,55	3,07	4,13	2,20	0,00

Tabela 28: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	4,19	3,07	3,91	2,21	0,02
100	4,09	3,16	3,68	1,99	0,00
110	4,06	3,06	3,75	1,99	0,00
120	4,22	3,17	3,96	2,09	0,00
130	4,22	3,18	3,81	2,10	0,00

Tabela 29: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	3,94	3,28	3,67	2,06	0,00
100	3,96	3,34	3,77	2,03	0,00
110	4,12	3,11	3,82	2,10	0,00
120	4,02	3,12	3,72	2,04	0,00
130	4,27	3,29	3,77	2,12	0,00

Tabela 30: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	4,40	3,08	4,17	2,23	0,01
100	4,41	3,15	4,11	2,19	0,00
110	4,55	3,07	4,23	2,25	0,00
120	4,60	3,14	4,33	2,24	0,00
130	4,76	3,21	4,34	2,29	0,00

Tabela 31: Tempo Médio de Computação (s) para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,0000	0,0000	0,0006	0,0003	0,0000
6	0,0001	0,0003	0,0002	0,0000	0,0000
7	0,0002	0,0000	0,0004	0,0001	0,0001
8	0,0002	0,0002	0,0005	0,0002	0,0006
9	0,0006	0,0005	0,0002	0,0002	0,0008

Tabela 32: Tempo Médio de Computação (s) para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,0002	0,0000	0,0002	0,0003	0,0000
6	0,0002	0,0001	0,0003	0,0001	0,0003
7	0,0002	0,0003	0,0001	0,0005	0,0005
8	0,0005	0,0005	0,0005	0,0003	0,0005
9	0,0016	0,0002	0,0008	0,0003	0,0006

Tabela 33: Tempo Médio de Computação (s) para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,0003	0,0002	0,0005	0,0000	0,0000
6	0,0005	0,0002	0,0001	0,0003	0,0003
7	0,0003	0,0001	0,0008	0,0003	0,0009
8	0,0002	0,0003	0,0010	0,0005	0,0011
9	0,0002	0,0005	0,0011	0,0005	0,0019

Tabela 34: Tempo Médio de Computação (s) para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,0001	0,0002	0,0006	0,0003	0,0000
6	0,0000	0,0000	0,0008	0,0002	0,0006
7	0,0003	0,0006	0,0011	0,0001	0,0008
8	0,0008	0,0003	0,0013	0,0005	0,0008
9	0,0002	0,0016	0,0009	0,0005	0,0022

Tabela 35: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
5	0,0002	0,0001	0,0005	0,0002	0,0000
6	0,0002	0,0002	0,0003	0,0002	0,0003
7	0,0002	0,0003	0,0006	0,0003	0,0006
8	0,0004	0,0003	0,0008	0,0004	0,0007
9	0,0006	0,0007	0,0007	0,0003	0,0014

Tabela 36: Tempo Médio de Computação (s) para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,0000	0,0003	0,0003	0,0002	0,0002
20	0,0006	0,0010	0,0013	0,0008	0,0031
30	0,0011	0,0016	0,0019	0,0016	0,0095
40	0,0031	0,0025	0,0030	0,0031	0,0206
50	0,0049	0,0045	0,0058	0,0053	0,0459
60	0,0075	0,0084	0,0085	0,0085	0,0880
70	0,0103	0,0119	0,0138	0,0133	0,1581
80	0,0153	0,0165	0,0186	0,0178	0,2625

Tabela 37: Tempo Médio de Computação (s) para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,0005	0,0008	0,0006	0,0001	0,0008
20	0,0009	0,0013	0,0011	0,0011	0,0028
30	0,0020	0,0019	0,0027	0,0014	0,0094
40	0,0038	0,0042	0,0044	0,0034	0,0211
50	0,0061	0,0063	0,0070	0,0044	0,0464
60	0,0091	0,0088	0,0115	0,0096	0,0903
70	0,0117	0,0136	0,0164	0,0138	0,1607
80	0,0165	0,0181	0,0225	0,0191	0,2664

Tabela 38: Tempo Médio de Computação (s) para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,0006	0,0001	0,0013	0,0006	0,0011
20	0,0002	0,0016	0,0019	0,0028	0,0034
30	0,0019	0,0022	0,0036	0,0020	0,0115
40	0,0046	0,0044	0,0056	0,0045	0,0222
50	0,0056	0,0067	0,0091	0,0072	0,0480
60	0,0094	0,0102	0,0139	0,0098	0,0922
70	0,0130	0,0144	0,0197	0,0153	0,1649
80	0,0184	0,0200	0,0258	0,0209	0,2703

Tabela 39: Tempo Médio de Computação (s) para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,0006	0,0008	0,0017	0,0002	0,0011
20	0,0005	0,0006	0,0020	0,0013	0,0031
30	0,0023	0,0023	0,0052	0,0025	0,0125
40	0,0052	0,0049	0,0069	0,0055	0,0236
50	0,0061	0,0074	0,0102	0,0070	0,0498
60	0,0099	0,0105	0,0153	0,0116	0,0942
70	0,0148	0,0156	0,0220	0,0169	0,1672
80	0,0215	0,0209	0,0284	0,0220	0,2730

Tabela 40: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
10	0,0004	0,0005	0,0010	0,0003	0,0008
20	0,0006	0,0011	0,0016	0,0015	0,0031
30	0,0018	0,0020	0,0033	0,0019	0,0107
40	0,0042	0,0040	0,0050	0,0041	0,0219
50	0,0057	0,0062	0,0080	0,0060	0,0475
60	0,0090	0,0094	0,0123	0,0099	0,0912
70	0,0124	0,0139	0,0180	0,0148	0,1627
80	0,0179	0,0189	0,0238	0,0200	0,2681

Tabela 41: Tempo Médio de Computação (s) para 5 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,0221	0,0230	0,0253	0,0244	0,4131
100	0,0275	0,0294	0,0324	0,0322	0,6253
110	0,0382	0,0384	0,0419	0,0422	0,9033
120	0,0467	0,0475	0,0505	0,0536	1,4417
130	0,0592	0,0602	0,0651	0,0665	1,9634

Tabela 42: Tempo Médio de Computação (s) para 10 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,0245	0,0247	0,0301	0,0266	0,4180
100	0,0287	0,0321	0,0374	0,0342	0,6313
110	0,0406	0,0416	0,0485	0,0469	0,9300
120	0,0500	0,0519	0,0588	0,0565	1,4504
130	0,0634	0,0642	0,0739	0,0702	1,9731

Tabela 43: Tempo Médio de Computação (s) para 15 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,0267	0,0267	0,0338	0,0284	0,4425
100	0,0337	0,0342	0,0420	0,0370	0,6553
110	0,0422	0,0446	0,0546	0,0477	0,9784
120	0,0502	0,0538	0,0647	0,0595	1,4538
130	0,0678	0,0684	0,0820	0,0752	1,9848

Tabela 44: Tempo Médio de Computação (s) para 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,0271	0,0281	0,0375	0,0300	0,4669
100	0,0368	0,0366	0,0480	0,0392	0,6608
110	0,0461	0,0475	0,0627	0,0503	0,9822
120	0,0545	0,0575	0,0726	0,0629	1,4641
130	0,0722	0,0716	0,0902	0,0778	1,9930

Tabela 45: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	Raj1	Raj2	Bert	Raj2Mod1	C_SN(2F)
90	0,0251	0,0256	0,0317	0,0273	0,4351
100	0,0317	0,0331	0,0399	0,0357	0,6432
110	0,0418	0,0430	0,0519	0,0468	0,9485
120	0,0503	0,0527	0,0616	0,0582	1,4525
130	0,0657	0,0661	0,0778	0,0724	1,9786

Experimentação Computacional - Parte B

Tabela 46: Porcentagem de Sucesso para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	81,00	93,00	90,00	96,00
6	76,00	87,00	87,00	90,00
7	67,00	81,00	74,00	87,00
8	47,00	69,00	63,00	75,00
9	45,00	64,00	69,00	73,00

Tabela 47: Porcentagem de Sucesso para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	80,00	92,00	91,00	93,00
6	71,00	85,00	85,00	90,00
7	66,00	83,00	74,00	86,00
8	56,00	66,00	68,00	79,00
9	51,00	66,00	62,00	70,00

Tabela 48: Porcentagem de Sucesso para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	84,00	94,00	85,00	93,00
6	72,00	81,00	81,00	89,00
7	59,00	78,00	78,00	83,00
8	56,00	71,00	66,00	77,00
9	46,00	64,00	52,00	65,00

Tabela 49: Porcentagem de Sucesso para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	85,00	90,00	88,00	92,00
6	78,00	85,00	85,00	91,00
7	64,00	80,00	80,00	89,00
8	53,00	73,00	73,00	80,00
9	46,00	62,00	61,00	67,00

Tabela 50: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	82,50	92,25	88,50	93,50
6	74,25	84,50	84,50	90,00
7	64,00	80,50	76,50	86,25
8	53,00	69,75	67,50	77,75
9	47,00	64,00	61,00	68,75

Tabela 51: Porcentagem de Sucesso para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	31,00	53,00	51,00	76,00
20	11,00	44,00	22,00	43,00
30	7,00	22,00	15,00	58,00
40	13,00	23,00	30,00	37,00
50	10,00	14,00	30,00	46,00
60	7,00	18,00	34,00	42,00
70	6,00	20,00	29,00	47,00
80	9,00	13,00	36,00	42,00

Tabela 52: Porcentagem de Sucesso para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	42,00	62,00	50,00	66,00
20	6,00	31,00	37,00	57,00
30	9,00	26,00	13,00	64,00
40	6,00	22,00	17,00	64,00
50	5,00	29,00	18,00	52,00
60	7,00	19,00	15,00	63,00
70	2,00	18,00	16,00	66,00
80	4,00	18,00	18,00	65,00

Tabela 53: Porcentagem de Sucesso para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	31,00	49,00	54,00	72,00
20	13,00	25,00	39,00	59,00
30	6,00	31,00	19,00	56,00
40	4,00	34,00	15,00	52,00
50	1,00	18,00	27,00	61,00
60	5,00	22,00	23,00	56,00
70	2,00	11,00	21,00	69,00
80	3,00	14,00	28,00	59,00

Tabela 54: Porcentagem de Sucesso para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	38,00	60,00	58,00	65,00
20	18,00	39,00	25,00	47,00
30	5,00	26,00	18,00	61,00
40	7,00	21,00	23,00	63,00
50	7,00	13,00	23,00	61,00
60	1,00	16,00	21,00	64,00
70	5,00	12,00	25,00	63,00
80	5,00	15,00	17,00	69,00

Tabela 55: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	35,50	56,00	53,25	69,75
20	12,00	34,75	30,75	51,50
30	6,75	26,25	16,25	59,75
40	7,50	25,00	21,25	54,00
50	5,75	18,50	24,50	55,00
60	5,00	18,75	23,25	56,25
70	3,75	15,25	22,75	61,25
80	5,25	15,00	24,75	58,75

Tabela 56: Porcentagem de Sucesso para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	6,00	9,00	35,00	50,00
100	9,00	9,00	39,00	44,00
110	3,00	16,00	36,00	45,00
120	9,00	11,00	33,00	47,00
130	5,00	16,00	40,00	41,00

Tabela 57: Porcentagem de Sucesso para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	3,00	16,00	17,00	64,00
100	4,00	12,00	20,00	67,00
110	10,00	11,00	24,00	56,00
120	7,00	13,00	22,00	59,00
130	5,00	15,00	17,00	64,00

Tabela 58: Porcentagem de Sucesso para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	2,00	14,00	18,00	71,00
100	5,00	14,00	12,00	74,00
110	2,00	13,00	23,00	65,00
120	3,00	15,00	19,00	63,00
130	2,00	14,00	16,00	71,00

Tabela 59: Porcentagem de Sucesso para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	1,00	16,00	13,00	75,00
100	4,00	12,00	29,00	64,00
110	3,00	10,00	28,00	67,00
120	0,00	7,00	15,00	78,00
130	1,00	15,00	15,00	71,00

Tabela 60: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	3,00	13,75	20,75	65,00
100	5,50	11,75	25,00	62,25
110	4,50	12,50	27,75	58,25
120	4,75	11,50	22,25	61,75
130	3,25	15,00	22,00	61,75

Tabela 61: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,31	0,14	0,12	0,05
6	0,43	0,16	0,16	0,22
7	0,40	0,15	0,30	0,11
8	0,79	0,41	0,43	0,38
9	0,73	0,35	0,39	0,33

Tabela 62: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,23	0,12	0,08	0,09
6	0,52	0,39	0,39	0,16
7	0,34	0,17	0,32	0,22
8	0,51	0,31	0,31	0,22
9	0,32	0,22	0,28	0,33

Tabela 63: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,11	0,08	0,09	0,06
6	0,22	0,18	0,18	0,09
7	0,35	0,17	0,12	0,15
8	0,39	0,20	0,30	0,29
9	0,52	0,32	0,34	0,30

Tabela 64: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,10	0,06	0,06	0,05
6	0,19	0,11	0,11	0,12
7	0,38	0,24	0,20	0,15
8	0,38	0,22	0,21	0,20
9	0,55	0,36	0,38	0,39

Tabela 65: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,19	0,10	0,09	0,06
6	0,34	0,21	0,21	0,15
7	0,37	0,18	0,24	0,16
8	0,52	0,28	0,31	0,27
9	0,53	0,31	0,35	0,34

Tabela 66: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,99	0,68	0,42	0,22
20	1,02	0,72	0,72	0,65
30	1,34	1,12	0,53	0,29
40	0,99	0,82	0,58	0,55
50	1,34	1,19	0,31	0,19
60	1,24	1,06	0,33	0,29
70	1,11	0,97	0,30	0,25
80	1,07	1,02	0,29	0,26

Tabela 67: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,64	0,40	0,48	0,36
20	1,06	0,73	0,46	0,32
30	1,13	0,80	0,54	0,33
40	1,39	1,17	0,45	0,24
50	1,23	1,01	0,43	0,30
60	1,24	1,00	0,37	0,22
70	1,25	1,09	0,27	0,11
80	1,17	1,01	0,28	0,17

Tabela 68: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,67	0,43	0,41	0,29
20	0,92	0,68	0,56	0,43
30	1,16	0,85	0,46	0,36
40	1,20	0,92	0,49	0,33
50	1,44	1,16	0,33	0,20
60	1,22	1,03	0,35	0,24
70	1,28	1,07	0,21	0,08
80	1,40	1,19	0,25	0,17

Tabela 69: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,48	0,34	0,32	0,31
20	1,00	0,76	0,56	0,48
30	1,28	0,91	0,44	0,30
40	1,33	1,06	0,36	0,23
50	1,52	1,29	0,30	0,18
60	1,56	1,30	0,28	0,16
70	1,42	1,23	0,23	0,14
80	1,34	1,15	0,25	0,14

Tabela 70: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,69	0,46	0,41	0,29
20	1,00	0,72	0,58	0,47
30	1,23	0,92	0,49	0,32
40	1,23	0,99	0,47	0,33
50	1,38	1,16	0,34	0,22
60	1,32	1,10	0,33	0,23
70	1,27	1,09	0,25	0,15
80	1,25	1,09	0,27	0,19

Tabela 71: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	1,28	1,20	0,18	0,12
100	1,37	1,30	0,18	0,17
110	1,15	1,08	0,19	0,15
120	1,13	1,09	0,21	0,17
130	1,04	1,00	0,17	0,18

Tabela 72: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	1,25	1,11	0,23	0,11
100	1,18	1,01	0,18	0,09
110	1,14	1,03	0,22	0,13
120	1,00	0,89	0,20	0,11
130	1,16	1,04	0,20	0,12

Tabela 73: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	1,32	1,15	0,23	0,13
100	1,22	1,07	0,24	0,13
110	1,10	0,98	0,18	0,11
120	1,19	1,05	0,16	0,09
130	1,19	1,06	0,19	0,11

Tabela 74: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	1,36	1,16	0,21	0,08
100	1,24	1,11	0,23	0,15
110	1,36	1,22	0,17	0,09
120	1,35	1,21	0,15	0,06
130	1,38	1,22	0,18	0,09

Tabela 75: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	1,30	1,15	0,21	0,11
100	1,25	1,12	0,21	0,13
110	1,19	1,08	0,19	0,12
120	1,17	1,06	0,18	0,11
130	1,19	1,08	0,19	0,12

Tabela 76: Tempo Médio de Computação (s) para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,0002	0,0003	0,0002	0,0003
6	0,0003	0,0002	0,0002	0,0003
7	0,0000	0,0000	0,0003	0,0003
8	0,0005	0,0003	0,0008	0,0012
9	0,0005	0,0002	0,0009	0,0011

Tabela 77: Tempo Médio de Computação (s) para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,0000	0,0001	0,0005	0,0005
6	0,0002	0,0002	0,0002	0,0006
7	0,0002	0,0003	0,0005	0,0008
8	0,0003	0,0006	0,0008	0,0008
9	0,0002	0,0006	0,0006	0,0008

Tabela 78: Tempo Médio de Computação (s) para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,0002	0,0000	0,0009	0,0006
6	0,0006	0,0000	0,0000	0,0005
7	0,0003	0,0000	0,0008	0,0006
8	0,0005	0,0005	0,0008	0,0008
9	0,0000	0,0005	0,0025	0,0009

Tabela 79: Tempo Médio de Computação (s) para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,0003	0,0005	0,0008	0,0005
6	0,0006	0,0003	0,0003	0,0008
7	0,0005	0,0006	0,0009	0,0009
8	0,0006	0,0006	0,0019	0,0016
9	0,0006	0,0005	0,0019	0,0008

Tabela 80: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
5	0,0002	0,0002	0,0006	0,0005
6	0,0004	0,0002	0,0002	0,0005
7	0,0002	0,0002	0,0006	0,0007
8	0,0005	0,0005	0,0010	0,0011
9	0,0003	0,0004	0,0015	0,0009

Tabela 81: Tempo Médio de Computação (s) para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,0003	0,0008	0,0005	0,0008
20	0,0009	0,0010	0,0020	0,0027
30	0,0019	0,0019	0,0064	0,0072
40	0,0037	0,0039	0,0141	0,0144
50	0,0069	0,0070	0,0255	0,0256
60	0,0108	0,0110	0,0420	0,0419
70	0,0165	0,0167	0,0648	0,0655
80	0,0234	0,0241	0,0966	0,0956

Tabela 82: Tempo Médio de Computação (s) para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,0003	0,0008	0,0010	0,0008
20	0,0020	0,0020	0,0030	0,0025
30	0,0023	0,0020	0,0075	0,0074
40	0,0047	0,0048	0,0155	0,0149
50	0,0069	0,0080	0,0278	0,0272
60	0,0124	0,0133	0,0467	0,0447
70	0,0181	0,0195	0,0697	0,0686
80	0,0263	0,0276	0,1012	0,1003

Tabela 83: Tempo Médio de Computação (s) para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,0005	0,0008	0,0011	0,0009
20	0,0024	0,0011	0,0031	0,0036
30	0,0022	0,0028	0,0092	0,0094
40	0,0052	0,0057	0,0169	0,0169
50	0,0080	0,0094	0,0289	0,0306
60	0,0131	0,0153	0,0469	0,0489
70	0,0193	0,0231	0,0704	0,0747
80	0,0275	0,0328	0,1025	0,1097

Tabela 84: Tempo Médio de Computação (s) para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,0006	0,0005	0,0008	0,0025
20	0,0019	0,0000	0,0027	0,0037
30	0,0030	0,0031	0,0105	0,0102
40	0,0053	0,0058	0,0176	0,0183
50	0,0090	0,0109	0,0297	0,0326
60	0,0158	0,0180	0,0491	0,0524
70	0,0205	0,0266	0,0739	0,0794
80	0,0286	0,0373	0,1064	0,1167

Tabela 85: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
10	0,0006	0,0006	0,0009	0,0012
20	0,0018	0,0010	0,0027	0,0031
30	0,0023	0,0025	0,0086	0,0084
40	0,0048	0,0050	0,0159	0,0162
50	0,0077	0,0088	0,0279	0,0291
60	0,0130	0,0143	0,0456	0,0475
70	0,0187	0,0214	0,0696	0,0722
80	0,0266	0,0303	0,1012	0,1061

Tabela 86: Tempo Médio de Computação (s) para 5 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	0,0325	0,0331	0,1319	0,1341
100	0,0442	0,0447	0,1755	0,1799
110	0,0572	0,0591	0,2353	0,2369
120	0,0738	0,0747	0,3017	0,3044
130	0,0862	0,0942	0,3802	0,3833

Tabela 87: Tempo Médio de Computação (s) para 10 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	0,0351	0,0383	0,1397	0,1406
100	0,0470	0,0509	0,1851	0,1895
110	0,0615	0,0662	0,2430	0,2490
120	0,0781	0,0843	0,3111	0,3184
130	0,0984	0,1064	0,3925	0,4025

Tabela 88: Tempo Médio de Computação (s) para 15 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	0,0367	0,0433	0,1433	0,1500
100	0,0489	0,0594	0,1900	0,2004
110	0,0650	0,0783	0,2498	0,2652
120	0,0817	0,0990	0,3595	0,3586
130	0,1019	0,1256	0,4408	0,4655

Tabela 89: Tempo Médio de Computação (s) para 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	0,0392	0,0522	0,1470	0,1586
100	0,0520	0,0698	0,1966	0,2127
110	0,0669	0,0903	0,2559	0,2778
120	0,0850	0,1161	0,3740	0,4153
130	0,1061	0,1462	0,4287	0,5086

Tabela 90: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	Raj2Mod2	Raj2Mod3	PH1(p)	PH1(p)Mod
90	0,0361	0,0416	0,1410	0,1453
100	0,0482	0,0561	0,1879	0,1945
110	0,0631	0,0730	0,2464	0,2568
120	0,0799	0,0933	0,3373	0,3485
130	0,1001	0,1161	0,4113	0,4392

Experimentação Computacional – Parte Final

Tabela 91: Porcentagem de Sucesso para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	95,00	92,00	92,00	94,00
6	88,00	88,00	88,00	93,00
7	88,00	80,00	80,00	83,00
8	68,00	72,00	70,00	72,00
9	68,00	66,00	63,00	72,00

Tabela 92: Porcentagem de Sucesso para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	91,00	91,00	89,00	87,00
6	87,00	84,00	84,00	88,00
7	85,00	84,00	83,00	81,00
8	67,00	66,00	76,00	80,00
9	56,00	63,00	65,00	64,00

Tabela 93: Porcentagem de Sucesso para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	92,00	93,00	95,00	91,00
6	90,00	81,00	81,00	80,00
7	82,00	80,00	72,00	80,00
8	73,00	76,00	72,00	77,00
9	61,00	63,00	61,00	69,00

Tabela 94: Porcentagem de Sucesso para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	91,00	92,00	93,00	90,00
6	88,00	87,00	87,00	88,00
7	85,00	76,00	76,00	79,00
8	72,00	65,00	65,00	75,00
9	60,00	61,00	66,00	59,00

Tabela 95: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	92,25	92,00	92,25	90,50
6	88,25	85,00	85,00	87,25
7	85,00	80,00	77,75	80,75
8	70,00	69,75	70,75	76,00
9	61,25	63,25	63,75	66,00

Tabela 96: Porcentagem de Sucesso para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	70,00	59,00	53,00	61,00
20	18,00	22,00	26,00	47,00
30	21,00	17,00	26,00	37,00
40	3,00	14,00	26,00	58,00
50	3,00	11,00	32,00	54,00
60	3,00	9,00	28,00	60,00
70	1,00	5,00	25,00	69,00
80	1,00	2,00	21,00	76,00

Tabela 97: Porcentagem de Sucesso para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	53,00	65,00	53,00	53,00
20	27,00	30,00	24,00	30,00
30	16,00	27,00	28,00	34,00
40	15,00	33,00	24,00	28,00
50	10,00	27,00	25,00	39,00
60	12,00	27,00	24,00	37,00
70	6,00	22,00	34,00	38,00
80	6,00	29,00	26,00	39,00

Tabela 98: Porcentagem de Sucesso para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	60,00	54,00	47,00	56,00
20	30,00	28,00	27,00	36,00
30	17,00	33,00	21,00	33,00
40	16,00	25,00	21,00	38,00
50	18,00	29,00	16,00	37,00
60	18,00	24,00	18,00	40,00
70	15,00	23,00	17,00	45,00
80	12,00	23,00	21,00	44,00

Tabela 99: Porcentagem de Sucesso para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	55,00	49,00	53,00	64,00
20	25,00	27,00	19,00	38,00
30	28,00	30,00	14,00	28,00
40	19,00	32,00	17,00	32,00
50	22,00	23,00	15,00	40,00
60	20,00	29,00	16,00	35,00
70	15,00	32,00	16,00	38,00
80	17,00	28,00	19,00	36,00

Tabela 100: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	59,50	56,75	51,50	58,50
20	25,00	26,75	24,00	37,75
30	20,50	26,75	22,25	33,00
40	13,25	26,00	22,00	39,00
50	13,25	22,50	22,00	42,50
60	13,25	22,25	21,50	43,00
70	9,25	20,50	23,00	47,50
80	9,00	20,50	21,75	48,75

Tabela 101: Porcentagem de Sucesso para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	0,00	3,00	22,00	75,00
100	0,00	1,00	26,00	73,00
110	0,00	0,00	17,00	83,00
120	0,00	0,00	24,00	76,00
130	0,00	0,00	23,00	77,00

Tabela 102: Porcentagem de Sucesso para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	6,00	24,00	21,00	49,00
100	3,00	19,00	24,00	55,00
110	3,00	12,00	25,00	60,00
120	0,00	15,00	24,00	61,00
130	0,00	10,00	32,00	58,00

Tabela 103: Porcentagem de Sucesso para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	10,00	30,00	14,00	46,00
100	5,00	19,00	28,00	48,00
110	5,00	25,00	16,00	54,00
120	6,00	29,00	21,00	44,00
130	3,00	26,00	27,00	44,00

Tabela 104: Porcentagem de Sucesso para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	13,00	31,00	11,00	45,00
100	8,00	35,00	18,00	39,00
110	5,00	29,00	17,00	49,00
120	8,00	19,00	24,00	49,00
130	7,00	33,00	10,00	50,00

Tabela 105: Porcentagem de Sucesso para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	7,25	22,00	17,00	53,75
100	4,00	18,50	24,00	53,75
110	3,25	16,50	18,75	61,50
120	3,50	15,75	23,25	57,50
130	2,50	17,25	23,00	57,25

Tabela 106: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,06	0,19	0,20	0,11
6	0,27	0,12	0,12	0,10
7	0,12	0,26	0,20	0,11
8	0,49	0,26	0,29	0,34
9	0,39	0,43	0,39	0,30

Tabela 107: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,11	0,17	0,15	0,19
6	0,20	0,33	0,33	0,24
7	0,23	0,23	0,21	0,22
8	0,32	0,44	0,24	0,22
9	0,49	0,37	0,20	0,26

Tabela 108: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,06	0,08	0,05	0,08
6	0,11	0,15	0,15	0,12
7	0,15	0,17	0,20	0,14
8	0,32	0,20	0,16	0,16
9	0,42	0,32	0,39	0,25

Tabela 109: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,05	0,03	0,04	0,09
6	0,14	0,09	0,09	0,07
7	0,19	0,35	0,29	0,27
8	0,28	0,32	0,29	0,23
9	0,44	0,38	0,28	0,35

Tabela 110: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,07	0,12	0,11	0,12
6	0,18	0,17	0,17	0,13
7	0,17	0,25	0,22	0,19
8	0,35	0,31	0,25	0,24
9	0,44	0,38	0,31	0,29

Tabela 111: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,38	0,55	0,45	0,36
20	1,30	0,68	0,59	0,47
30	1,24	1,06	0,72	0,50
40	2,09	1,13	0,72	0,28
50	1,91	1,20	0,55	0,26
60	2,32	1,25	0,62	0,20
70	2,62	1,41	0,73	0,18
80	2,77	1,52	0,69	0,13

Tabela 112: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,48	0,42	0,44	0,48
20	0,79	0,69	0,56	0,73
30	1,02	0,59	0,60	0,56
40	0,99	0,58	0,54	0,60
50	1,42	0,65	0,58	0,45
60	1,34	0,61	0,52	0,40
70	1,40	0,68	0,48	0,45
80	1,37	0,61	0,56	0,34

Tabela 113: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,44	0,52	0,49	0,48
20	0,78	0,53	0,50	0,46
30	0,96	0,65	0,59	0,59
40	1,22	0,75	0,68	0,53
50	1,02	0,70	0,78	0,55
60	1,19	0,64	0,66	0,38
70	1,02	0,60	0,71	0,42
80	1,12	0,62	0,56	0,31

Tabela 114: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,40	0,47	0,42	0,36
20	0,87	0,72	0,64	0,50
30	0,85	0,57	0,72	0,48
40	0,98	0,68	0,86	0,58
50	0,97	0,54	0,75	0,42
60	0,87	0,53	0,72	0,46
70	1,12	0,58	0,66	0,44
80	1,03	0,53	0,68	0,41

Tabela 115: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,42	0,49	0,45	0,42
20	0,93	0,66	0,57	0,54
30	1,02	0,72	0,66	0,53
40	1,32	0,78	0,70	0,50
50	1,33	0,77	0,66	0,42
60	1,43	0,76	0,63	0,36
70	1,54	0,82	0,65	0,37
80	1,58	0,82	0,62	0,30

Tabela 116: Porcentagem de Desvio Médio Relativo para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	2,74	1,74	0,72	0,10
100	2,75	1,77	0,51	0,12
110	3,12	1,82	0,66	0,08
120	3,25	1,94	0,65	0,14
130	3,40	1,82	0,62	0,06

Tabela 117: Porcentagem de Desvio Médio Relativo para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	1,40	0,66	0,61	0,28
100	1,56	0,65	0,58	0,21
110	1,84	0,79	0,57	0,20
120	1,85	0,66	0,51	0,18
130	1,76	0,79	0,49	0,15

Tabela 118: Porcentagem de Desvio Médio Relativo para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	1,36	0,55	0,62	0,28
100	1,28	0,57	0,56	0,33
110	1,37	0,56	0,65	0,25
120	1,37	0,54	0,56	0,26
130	1,36	0,51	0,55	0,29

Tabela 119: Porcentagem de Desvio Médio Relativo para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	1,10	0,49	0,65	0,41
100	1,23	0,48	0,63	0,30
110	1,19	0,52	0,66	0,30
120	1,21	0,61	0,57	0,27
130	1,21	0,51	0,74	0,32

Tabela 120: Porcentagem de Desvio Médio Relativo para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	1,65	0,86	0,65	0,27
100	1,71	0,86	0,57	0,24
110	1,88	0,92	0,64	0,21
120	1,92	0,94	0,57	0,21
130	1,93	0,91	0,60	0,21

Tabela 121: Tempo Médio de Computação (s) para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,0003	0,0003	0,0002	0,0000
6	0,0003	0,0000	0,0000	0,0001
7	0,0003	0,0001	0,0000	0,0000
8	0,0012	0,0006	0,0003	0,0006
9	0,0011	0,0008	0,0006	0,0002

Tabela 122: Tempo Médio de Computação (s) para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,0005	0,0002	0,0003	0,0006
6	0,0006	0,0003	0,0003	0,0003
7	0,0008	0,0005	0,0000	0,0003
8	0,0008	0,0005	0,0000	0,0005
9	0,0008	0,0006	0,0005	0,0005

Tabela 123: Tempo Médio de Computação (s) para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,0006	0,0005	0,0003	0,0006
6	0,0005	0,0003	0,0003	0,0003
7	0,0006	0,0009	0,0003	0,0002
8	0,0008	0,0011	0,0008	0,0005
9	0,0009	0,0019	0,0006	0,0008

Tabela 124: Tempo Médio de Computação (s) para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,0005	0,0006	0,0005	0,0006
6	0,0008	0,0006	0,0006	0,0006
7	0,0009	0,0008	0,0006	0,0005
8	0,0016	0,0008	0,0005	0,0006
9	0,0008	0,0022	0,0006	0,0005

Tabela 125: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
5	0,0005	0,0004	0,0003	0,0005
6	0,0005	0,0003	0,0003	0,0003
7	0,0007	0,0006	0,0002	0,0002
8	0,0011	0,0007	0,0004	0,0005
9	0,0009	0,0014	0,0006	0,0005

Tabela 126: Tempo Médio de Computação (s) para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,0008	0,0002	0,0006	0,0005
20	0,0027	0,0031	0,0019	0,0014
30	0,0072	0,0095	0,0071	0,0075
40	0,0144	0,0206	0,0205	0,0205
50	0,0256	0,0459	0,0467	0,0473
60	0,0419	0,0880	0,0942	0,0948
70	0,0655	0,1581	0,1732	0,1728
80	0,0956	0,2625	0,2910	0,2908

Tabela 127: Tempo Médio de Computação (s) para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,0008	0,0008	0,0005	0,0003
20	0,0025	0,0028	0,0019	0,0022
30	0,0074	0,0094	0,0077	0,0084
40	0,0149	0,0211	0,0226	0,0206
50	0,0272	0,0464	0,0489	0,0494
60	0,0447	0,0903	0,0972	0,0967
70	0,0686	0,1607	0,1758	0,1761
80	0,1003	0,2664	0,2958	0,2953

Tabela 128: Tempo Médio de Computação (s) para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,0011	0,0011	0,0006	0,0006
20	0,0031	0,0034	0,0024	0,0028
30	0,0092	0,0115	0,0092	0,0096
40	0,0169	0,0222	0,0222	0,0227
50	0,0289	0,0480	0,0498	0,0493
60	0,0469	0,0922	0,0987	0,0986
70	0,0704	0,1649	0,1781	0,1791
80	0,1025	0,2703	0,2986	0,2997

Tabela 129: Tempo Médio de Computação (s) para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,0008	0,0011	0,0008	0,0006
20	0,0027	0,0031	0,0027	0,0030
30	0,0105	0,0125	0,0105	0,0114
40	0,0176	0,0236	0,0240	0,0236
50	0,0297	0,0498	0,0512	0,0526
60	0,0491	0,0942	0,1005	0,1009
70	0,0739	0,1672	0,1811	0,1817
80	0,1064	0,2730	0,3028	0,3025

Tabela 130: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
10	0,0009	0,0008	0,0006	0,0005
20	0,0027	0,0031	0,0022	0,0023
30	0,0086	0,0107	0,0086	0,0092
40	0,0159	0,0219	0,0223	0,0218
50	0,0279	0,0475	0,0492	0,0497
60	0,0456	0,0912	0,0977	0,0978
70	0,0696	0,1627	0,1770	0,1774
80	0,1012	0,2681	0,2970	0,2971

Tabela 131: Tempo Médio de Computação (s) para 5 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	0,1341	0,4131	0,4606	0,4620
100	0,1799	0,6253	0,7038	0,7041
110	0,2369	0,9033	1,0178	1,0183
120	0,3044	1,4417	1,6282	1,6288
130	0,3833	1,9634	2,2297	2,2300

Tabela 132: Tempo Médio de Computação (s) para 10 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	0,1397	0,4180	0,5057	0,4893
100	0,1851	0,6313	0,7075	0,7107
110	0,2430	0,9300	1,0839	1,0840
120	0,3111	1,4504	1,6369	1,6391
130	0,3925	1,9731	2,2394	2,2394

Tabela 133: Tempo Médio de Computação (s) para 15 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	0,1433	0,4425	0,5299	0,5664
100	0,1900	0,6553	0,7142	0,7152
110	0,2498	0,9784	1,1302	1,1114
120	0,3595	1,4538	1,6438	1,6450
130	0,4408	1,9848	2,2475	2,2487

Tabela 134: Tempo Médio de Computação (s) para 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	0,1470	0,4669	0,5160	0,5346
100	0,1966	0,6608	0,7202	0,7200
110	0,2559	0,9822	1,1378	1,1347
120	0,3740	1,4641	1,6541	1,6545
130	0,4287	1,9930	2,2586	2,2589

Tabela 135: Tempo Médio de Computação (s) para 5, 10, 15 e 20 máquinas

n	PH1(p)Mod	C_SN(2F)	C1_SN	C2_SN
90	0,1410	0,4351	0,5030	0,5131
100	0,1879	0,6432	0,7114	0,7125
110	0,2464	0,9485	1,0924	1,0871
120	0,3373	1,4525	1,6407	1,6418
130	0,4113	1,9786	2,2438	2,2443