



COPPE/UFRJ

ALGORITMOS HEURÍSTICOS E HÍBRIDOS PARA O PROBLEMA DA
ALOCAÇÃO DINÂMICA DE ESPAÇOS

Geiza Cristina da Silva

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Produção.

Orientadores: Paulo Oswaldo Boaventura
Netto
Laura Silvia Bahiense da Silva
Leite

Rio de Janeiro
Março de 2010

ALGORITMOS HEURÍSTICOS E HÍBRIDOS PARA O PROBLEMA DA
ALOCAÇÃO DINÂMICA DE ESPAÇOS

Geiza Cristina da Silva

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Examinada por:

Prof. Paulo Oswaldo Boaventura Netto, Ph.D.

Profa. Laura Silvia Bahiense da Silva Leite, D.Sc.

Prof. Luiz Satoru Ochi, D.Sc.

Profa. Simone de Lima Martins, D.Sc.

Prof. Samuel Jurkiewicz, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2010

Silva, Geiza Cristina da

Algoritmos heurísticos e híbridos para o Problema da Alocação Dinâmica de Espaços/Geiza Cristina da Silva. – Rio de Janeiro: UFRJ/COPPE, 2010.

XXI, 213 p.: il.; 29,7cm.

Orientadores: Paulo Oswaldo Boaventura Netto

Laura Silvia Bahiense da Silva Leite

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Produção, 2010.

Referências Bibliográficas: p. 104 – 108.

1. Problema Quadrático de Alocação. 2. Metaheurísticas. 3. Otimização Combinatória. I. Boaventura Netto, Paulo Oswaldo *et al.*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Produção. III. Título.

Aos meus pais queridos.

Agradecimentos

Aos meus orientadores, Prof. Paulo Oswaldo Boaventura Netto, Prof. Luiz Satoru Ochi e Profa. Laura Bahiense, pelas contribuições e pela paciência durante este trabalho.

À Prof. Simone de Lima Martins e ao Prof. Samuel Jurkiewicz pelos comentários valiosos.

Ao Prof. Tarcísio de Souza Lima, uma pessoa muito querida, que em 2001 no final da minha graduação, apresentou-me o interessante mundo da pesquisa.

Pelas contribuições na implementação, ao Prof. Marcone Jamilson Freitas de Souza e, por me salvar em vários momentos de luta com o CPLEX, ao amigo Haroldo Gambini.

Aos meus pais, meu porto seguro, meus maiores admiradores, pela vida, pela esperança no meu futuro, pela orientação, pela preocupação e tudo mais. Eu poderia escrever páginas e páginas e não seriam suficientes para agradecer por tudo que fizeram por mim. Muito menos eu seria capaz de descrever o que eles são para mim.

Aos amigos que, de perto ou de longe, acompanharam todo esse processo: Priscila, Tati (Rio), Luciana, Sidney, Gleide, Idalmis, Jacques, Stênio, Cris Norbiato e Taiane.

À Tati (Minas), companheira nos últimos três anos, nos quais não teve um dia que não tenha escutado um “comentário” à respeito da tese e, outros mais, de toda a complexidade que envolve a vida.

À Rita, pela grande amizade, cumplicidade e pelos momentos em que rimos e choramos juntas, pelos mais diversos motivos.

Ao Leo, pelo incentivo em concluir esse trabalho.

À todos os funcionários do PEP, sempre prestrativos, em especial: Andreia, Pedro e Roberta.

À Capes, pela ajuda financeira e àqueles colegas do DECEA/UFOP que puderam entender algumas ausências devido a este trabalho.

À Deus, “porque dEle, por Ele e para Ele são todas as coisas”.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

ALGORITMOS HEURÍSTICOS E HÍBRIDOS PARA O PROBLEMA DA ALOCAÇÃO DINÂMICA DE ESPAÇOS

Geiza Cristina da Silva

Março/2010

Orientadores: Paulo Oswaldo Boaventura Netto
Laura Silvia Bahiense da Silva Leite

Programa: Engenharia de Produção

O Problema da Alocação Dinâmica de Espaços (PADE) foi recentemente formalizado na literatura. Foi inspirado na necessidade de otimização da distância percorrida por recursos requeridos para realização de atividades na ocorrência de desligamentos planejados em centrais de energia elétrica. Em geral, o problema apresenta aplicações potenciais em projetos nos quais movimentar um recurso é uma tarefa difícil ou cara, ou nos quais o congestionamento desses recursos é indesejável, como em construção de pontes e atividades de mineração.

Este problema é considerado difícil para se resolver computacionalmente e, portanto, a dificuldade em encontrar uma solução ótima para instâncias de elevadas dimensões justifica o uso de métodos aproximados. Este trabalho propõe um estudo detalhado do PADE e das principais técnicas referenciadas na literatura para resolução de instâncias do mesmo. Além disso, são propostos novos métodos de construção e busca local e algoritmos baseados nas metaheurísticas VNS, ILS, GRASP e Busca Tabu. Este estudo inclui uma proposta de abordagem por meio de algoritmos exatos a fim de se conseguir soluções ótimas, viáveis ou bons limites inferiores através do CPLEX, e ainda, a combinação de dois ou mais métodos, em uma metodologia híbrida.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

HEURISTIC AND HYBRID ALGORITHMS TO THE DYNAMIC SPACE
ALLOCATION PROBLEM

Geiza Cristina da Silva

March/2010

Advisors: Paulo Oswaldo Boaventura Netto
Laura Silvia Bahiense da Silva Leite

Department: Production Engineering

The Dynamic Space Allocation Problem (DSAP) was recently formalized in the literature. It was inspired by the need of optimizing the distance traveled by resources required to activities during planned outages at electrical power plants. However, DSAP has potential applications in managing resources in projects when the movement of resources is difficult, expensive, or where the congestion is undesirable, such as construction of bridges and mining activities. This work presents new construction and local search methods and algorithms based on the metaheuristic VNS, ILS, GRASP and Tabu Search. This study includes a proposed approach for exact algorithms to achieve optimal or feasible solutions or good lower bounds by CPLEX, and even the combination of two or more methods in a hybrid versions.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	2
2 O Problema da Alocação Dinâmica de Espaços	5
3 Estado-da-arte	11
3.1 Revisão Bibliográfica	11
3.1.1 Modelos matemáticos	12
3.2 Algoritmos da Literatura	15
3.2.1 Algoritmos de Construção	15
3.2.2 Movimentos	16
3.2.3 Metaheurísticas	19
3.3 Instâncias para o PADE	26
4 Heurísticas de Construção e Busca Local Propostas para o PADE	29
4.1 Representação de uma solução	29
4.2 Heurísticas de Construção	31
4.2.1 Heurística Randomizada de Construção Adaptada (RCA) . . .	31
4.2.2 Heurística de Construção Randomizada FA1 (RFA1)	33
4.2.3 Heurística de Construção Randomizada FA2 (RFA2)	33
4.2.4 Heurística de Construção de Atividades Adaptativa (CAA) . . .	34
4.2.5 Heurística RSP Adaptada (RSPA)	35
4.2.6 Heurística de Construção de Ociosos Adaptativa (CAO)	36
4.2.7 Algoritmos de Construção Propostos	37
4.3 Heurísticas de Refinamento	37
4.3.1 Busca Local <i>Best Improvement</i> ou Melhor Vizinho	37
4.3.2 Busca Local <i>First Improvement</i> ou Primeiro de Melhor	38
4.3.3 Algoritmos de busca local propostos	38
4.4 Análises Experimentais Preliminares	39

5	Metaheurísticas Propostas para o PADE	45
5.1	A metaheurística <i>Variable Neighborhood Search</i>	45
5.1.1	VNS proposto para o PADE	46
5.2	A metaheurística <i>Iterated Local Search</i>	48
5.2.1	ILS proposto para o PADE	50
5.3	A Metaheurística GRASP	56
5.3.1	Heurísticas GRASP propostas para o PADE	57
5.4	Busca Tabu proposta para o PADE	62
6	Heurísticas Híbridas	69
6.1	Heurística ILS + VNS Propostas para o PADE	69
6.2	Heurísticas ILS + Busca Tabu Propostas para o PADE	70
6.3	Heurísticas VNS + Busca Tabu Propostas para o PADE	71
6.4	Heurísticas GRASP + Busca Tabu Proposta para o PADE	73
6.5	Considerações sobre Algoritmos Híbridos	75
7	Experimentos e resultados computacionais	77
7.1	Comparação dos resultados entre algoritmos exato, da literatura e propostos	78
7.2	Comparação de resultados entre os algoritmos propostos	88
7.2.1	Análise com tempos estabelecidos	88
7.2.2	Análise Probabilística	91
8	Conclusões e Direções Futuras	102
	Referências Bibliográficas	104
A	Resultados Computacionais dos Algoritmos VNS Propostos	110
B	Resultados Computacionais dos Algoritmos ILS Propostos	132
C	Resultados Computacionais dos Algoritmos GRASP Propostos	140
D	Resultados Computacionais dos Algoritmos Tabu Propostos	172
E	Resultados Computacionais dos Algoritmos ILS+Tabu Propostos	180
F	Resultados Computacionais dos Algoritmos VNS+Tabu Propostos	188
G	Resultados Computacionais dos Algoritmos GRASP+Tabu Pro- postos	196

H	Comparação de Desvios Percentuais entre Algoritmos da Literatura e Propostos	204
I	Instâncias	207

Lista de Figuras

3.1	Algoritmo SA1	21
3.2	Algoritmo BT1	23
4.1	Algoritmo RCA	32
4.2	Algoritmo RFA1	33
4.3	Algoritmo RFA2	34
4.4	Algoritmo CAA	35
4.5	Algoritmo RSPA	36
4.6	Algoritmo CAO	36
4.7	Pseudocódigo do algoritmo de busca pelo melhor vizinho	38
5.1	Pseudocódigo da metaheurística VNS	46
5.2	Pseudocódigo da metaheurística ILS	49
5.3	Pseudocódigo do algoritmo ILS Proposto	54
5.4	Pseudocódigo da metaheurística GRASP	56
5.5	Pseudocódigo do algoritmo GRASP Reativo	60
5.6	Algoritmo BTP	63
6.1	Algoritmo VNS+BTP	72
7.1	Comparação entre algoritmos propostos para a instância 33 com alvo 57	94
7.2	Comparação entre algoritmos propostos para a instância 33 com alvo 55	94
7.3	Comparação entre algoritmos propostos para a instância 34 com alvo 76	95
7.4	Comparação entre algoritmos propostos para a instância 34 com alvo 74	95
7.5	Comparação entre algoritmos propostos para a instância 37 com alvo 50	96
7.6	Comparação entre algoritmos propostos para a instância 37 com alvo 49	96

7.7	Comparação entre algoritmos propostos para a instância 52 com alvo 98	97
7.8	Comparação entre algoritmos propostos para a instância 52 com alvo 92	97
7.9	Comparação entre algoritmos propostos para a instância 54 com alvo 67	98
7.10	Comparação entre algoritmos propostos para a instância 54 com alvo 65	98
7.11	Comparação entre algoritmos propostos para a instância 56 com alvo 97	99
7.12	Comparação entre algoritmos propostos para a instância 56 com alvo 93	99
A.1	Comparativo dos tempos médios do algoritmo VNS1 para execuções com 1000, 1200 e 1500 iterações	130

Lista de Tabelas

2.1	Agenda do projeto	7
2.2	<i>Layout</i> com 6 espaços	7
2.3	Matriz de distâncias entre locações	8
2.4	Solução - alocação de atividades	8
2.5	Solução - alocação de recursos ociosos	8
2.6	Resumo da movimentação de recursos da agenda-exemplo	10
3.1	Principais diferenças entre os trabalhos da literatura	12
3.2	Exemplo do movimento \mathcal{M}_1	17
3.3	Exemplo do movimento \mathcal{M}_2	17
3.4	Exemplo do movimento \mathcal{M}_3	17
3.5	Exemplo do movimento \mathcal{M}_4	18
3.6	Exemplo do movimento \mathcal{M}_5	18
3.7	Exemplo do movimento \mathcal{M}_6	18
3.8	Exemplo do movimento \mathcal{M}_7	19
3.9	Exemplo do movimento \mathcal{M}_8	19
3.10	<i>Layout</i> 2×3 para problemas com 6 espaços	26
3.11	<i>Layout</i> 4×5 para problemas com 20 espaços	27
4.1	Exemplo da matriz \mathcal{L}	30
4.2	Exemplo de matriz \mathcal{D}	30
4.3	Algoritmos de construção propostos	37
4.4	Algoritmos de busca local propostos	39
4.5	Sumário dos resultados das construções	40
4.6	Sumário dos resultados das construções com busca MV1	40
4.7	Sumário dos resultados das construções com busca PM1	40
4.8	Sumário dos resultados das construções com busca MV2	41
4.9	Sumário dos resultados das construções com busca PM2	41
4.10	Sumário dos resultados das construções com busca MV4	41
4.11	Sumário dos resultados das construções com busca MV5	42
4.12	Sumário dos resultados das construções com busca MV6	42

5.1	Algoritmos VNS Propostos	47
5.2	Sumário dos resultados do algoritmos VNS para 500 e 700 iterações .	47
5.3	Sumário dos resultados do algoritmo VNS1 para 1000, 1200 e 1500 iterações	47
5.4	Melhorias de custo obtidas com os algoritmos VNS	48
5.5	Melhorias de custo obtidas com o algoritmo VNS1	48
5.6	Exemplo da perturbação de nível 1	50
5.7	Exemplo da perturbação de nível 2	51
5.8	Exemplo da perturbação de nível 2 com inviabilidade	51
5.9	Exemplo da perturbação de nível 3	52
5.10	Exemplo da perturbação de nível 5	52
5.11	Exemplo da perturbação de nível 6	53
5.12	Algoritmos ILS Propostos	54
5.13	Sumário dos resultados do algoritmos ILS	55
5.14	Comparação dos algoritmos ILS com literatura	55
5.15	Melhoria de custos obtidos com implementação ILS com perturbação aleatória com 700, 1000, 1500 e 2000 iterações	56
5.16	Algoritmos GRASP Propostos	57
5.17	Sumário dos resultados da variação de α para G1 e G2	57
5.18	Sumário dos resultados da variação de α para G3 e G4	58
5.19	Sumário dos resultados da variação de α para G5 e G6	58
5.20	Sumário dos resultados da variação de α para G7 e G8	58
5.21	Sumário dos resultados da variação de α para G9 e G10	58
5.22	Sumário dos resultados obtidos em relação à literatura para algumas versões de GRASP	59
5.23	Algoritmos GRASP Reativo	61
5.24	Sumário dos resultados dos algoritmos GRASP Reativo Propostos . .	61
5.25	Sumário dos resultados obtidos em relação a literatura para algumas versões de GRASP Reativo	61
5.26	Limites inferiores e superiores para lista tabu	64
5.27	Variação do tamanho de lista tabu para o algoritmo de Construção C1	64
5.28	Variação do tamanho de lista tabu para o algoritmo de Construção C2	64
5.29	Variação do tamanho de lista tabu para o algoritmo de Construção C3	65
5.30	Variação do tamanho de lista tabu para o algoritmo de Construção C4	65
5.31	Variação do tamanho de lista tabu para o algoritmo de Construção C5	65
5.32	Variação do tamanho de lista tabu para o algoritmo de Construção C6	65
5.33	Variação do tamanho de lista tabu para o algoritmo de Construção C7	65
5.34	Variação do tamanho de lista tabu para o algoritmo de Construção C8	66
5.35	Variação do tamanho de lista tabu para o algoritmo de Construção C9	66

5.36	Melhores combinações de algoritmo de construção com tamanho de lista tabu	66
5.37	Comparação dos melhores algoritmos busca tabu com literatura . . .	67
6.1	Comparativo de resultados com diferentes números de iterações VNS no método ILS+VNS	70
6.2	Algoritmos híbridos de ILS e BTP Propostos	70
6.3	Comparação dos algoritmos ILS+BTP com a literatura	71
6.4	Sumário da comparação dos algoritmos ILS+BTP com literatura . . .	71
6.5	Algoritmos híbridos de VNS e BTP Propostos	72
6.6	Melhores combinações de algoritmos VNS+BTP com tamanho de lista tabu	73
6.7	Comparação dos melhores algoritmos VNS+BTP com literatura	73
6.8	Sumário dos resultados do algoritmos GRASP+BTP	74
6.9	Sumário da comparação dos algoritmos GRASP+BTP com literatura .	75
7.1	Metaheurísticas e heurísticas propostas	77
7.2	Custos dos algoritmos CPLEX, literatura e propostos	79
7.3	Desvio percentual entre custo exato, melhor solução da literatura e dos algoritmos propostos	80
7.4	Tempos do algoritmo exato, e da melhor solução da literatura e dos algoritmos propostos	81
7.5	Comparação dos algoritmos da literatura e propostos com exato . . .	82
7.6	Custos dos algoritmos da literatura e propostos para instâncias com 12 locações	82
7.7	Custos dos algoritmos da literatura e propostos para instâncias com 20 locações	83
7.8	Custos dos algoritmos da literatura e propostos para instâncias com 32 locações	84
7.9	Tempos de processamento da literatura e dos algoritmos propostos para instâncias com 12 locações	85
7.10	Tempos de processamento da literatura e dos algoritmos propostos para instâncias com 20 locações	85
7.11	Tempos de processamento da literatura e dos algoritmos propostos para instâncias com 32 locações	86
7.12	Comparação dos algoritmos da literatura e propostos para instâncias com 12, 20 e 32 locações	87
7.13	Sumário de desempenho dos algoritmos propostos	87
7.14	Tempo de execução (em segundos) mínimo e máximo de acordo com grupos de instâncias	88

7.15	Melhoria de custos dos algoritmos para tempos <i>min</i> e <i>max</i>	89
7.16	Sumário do desempenho dos algoritmos para tempo <i>min</i> e <i>max</i> e 39 segundos em comparação com a literatura	90
7.17	Média de desvio e de número de <i>Best</i> obtidos no tempo <i>min</i> e <i>max</i> .	91
7.18	Alvos usados no teste probabilístico	92
7.19	Tempo (em segundos) para que 50%, 75% e 90% das execuções atinjam o valor alvo	100
A.1	Melhor solução (obtida de 10 execuções) com 500 iterações do VNS para instâncias de 6 e 12 locações	112
A.2	Melhor solução (obtida de 10 execuções) com 700 iterações do VNS para instâncias de 6 e 12 locações	113
A.3	Melhor solução (obtida de 10 execuções) com 1000, 1200 e 1500 iterações do algoritmo VNS1 para instâncias de 6 e 12 locações	114
A.4	Melhor solução (obtida de 10 execuções) com 500 iterações do VNS para instâncias de 20 e 32 locações	115
A.5	Melhor solução (obtida de 10 execuções) com 700 iterações do VNS para instâncias de 20 e 32 locações	116
A.6	Melhor solução (obtida de 10 execuções) com 1000, 1200 e 1500 iterações do algoritmo VNS1 para instâncias de 20 e 32 locações	117
A.7	Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 500 iterações para instâncias de 6 e 12 locações	118
A.8	Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 700 iterações para instâncias de 6 e 12 locações	119
A.9	Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) do algoritmo VNS1 com 1000, 1200 e 1500 iterações para instâncias de 6 e 12 locações	120
A.10	Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 500 iterações para instâncias de 20 e 32 locações	121
A.11	Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 700 iterações para instâncias de 20 e 32 locações	122
A.12	Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) do algoritmo VNS1 com 1000, 1200 e 1500 iterações para instâncias de 20 e 32 locações	123

A.13	Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 500 iterações para instâncias de 6 e 12 locações	124
A.14	Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 700 iterações para instâncias de 6 e 12 locações	125
A.15	Tempo médio (em segundos) de 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para instâncias de 6 e 12 locações	126
A.16	Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 100 iterações para instâncias de 20 e 32 locações	127
A.17	Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 100 iterações para instâncias de 20 e 32 locações	128
A.18	Tempo médio (em segundos) de 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para instâncias de 20 e 32 locações	129
B.1	Melhor solução (obtida de 10 execuções) do ILS para instâncias de 6 e 12 locações	133
B.2	Melhor solução (obtida de 10 execuções) do ILS para instâncias de 20 e 32 locações	134
B.3	Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos ILS para instâncias de 6 e 12 locações	135
B.4	Desvio percentual entre a melhor solução da literatura e melhor solução dos algoritmos ILS para instâncias de 20 e 32 locações	136
B.5	Tempo Médio das solução ILS para instâncias de 6 e 12 locações em 10 execuções de cada algoritmo	137
B.6	Tempo Médio das solução ILS para instâncias de 20 e 32 locações em 10 execuções de cada algoritmo	138
C.1	Melhor solução dos algoritmos GRASP G1 e G2 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações	141
C.2	Melhor solução dos algoritmos GRASP G3 e G4 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações	142
C.3	Melhor solução dos algoritmos GRASP G5 e G6 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações	143
C.4	Melhor solução dos algoritmos GRASP G7 e G8 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações	144
C.5	Melhor solução dos algoritmos GRASP G9 e G10 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações	145
C.6	Melhor solução dos algoritmos GRASP G1 e G2 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações	146
C.7	Melhor solução dos algoritmos GRASP G3 e G4 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações	147

C.8	Melhor solução dos algoritmos GRASP G5 e G6 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações	148
C.9	Melhor solução dos algoritmos GRASP G7 e G8 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações	149
C.10	Melhor solução dos algoritmos GRASP G9 e G10 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações . . .	150
C.11	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G1 e G2 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações . .	151
C.12	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G3 e G4 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações . .	152
C.13	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G5 e G6 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações . .	153
C.14	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G7 e G8 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações . .	154
C.15	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G9 e G10 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações . .	155
C.16	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G1 e G2 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações .	156
C.17	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G3 e G4 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações .	157
C.18	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G5 e G6 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações .	158
C.19	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G7 e G8 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações .	159
C.20	Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G9 e G10 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações .	160
C.21	Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP G1 e G2 de acordo com variação de alfa para instâncias de 6 e 12 locações	161

C.22	Tempo médio (em segundos) de 10 execuções do G3 e G4 de acordo com variação de alfa para instâncias de 6 e 12 locações	162
C.23	Tempo médio (em segundos) de 10 execuções do G5 e G6 de acordo com variação de alfa para instâncias de 6 e 12 locações	163
C.24	Tempo médio (em segundos) de 10 execuções do G7 e G8 de acordo com variação de alfa para instâncias de 6 e 12 locações	164
C.25	Tempo médio (em segundos) de 10 execuções do G9 e G10 de acordo com variação de alfa para instâncias de 6 e 12 locações	165
C.26	Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP G1 e G2 de acordo com variação de alfa para instâncias de 20 e 32 locações	166
C.27	Tempo médio (em segundos) de 10 execuções do G3 e G4 de acordo com variação de alfa para instâncias de 20 e 32 locações	167
C.28	Tempo médio (em segundos) de 10 execuções do G5 e G6 de acordo com variação de alfa para instâncias de 20 e 32 locações	168
C.29	Tempo médio (em segundos) de 10 execuções do G7 e G8 de acordo com variação de alfa para instâncias de 20 e 32 locações	169
C.30	Tempo médio (em segundos) de 10 execuções do G9 e G10 de acordo com variação de alfa para instâncias de 20 e 32 locações	170
D.1	Melhor solução (obtida de 10 execuções) da combinação de construção e tamanho de lista tabu para instâncias de 6 e 12 locações	173
D.2	Melhor solução (obtida de 10 execuções) da combinação de construção e tamanho de lista tabu para instâncias de 20 e 32 locações	174
D.3	Desvio percentual entre a melhor solução da literatura e da combinação de construção e tamanho de lista tabu para instâncias de 6 e 12 locações	175
D.4	Desvio percentual entre a melhor solução da literatura e da combinação de construção e tamanho de lista tabu para instâncias de 20 e 32 locações	176
D.5	Tempo médio (em segundos) de 10 execuções da combinação de construção e tamanho de lista tabu para instâncias de 6 e 12 locações	177
D.6	Tempo médio (em segundos) de 10 execuções da combinação de construção e tamanho de lista tabu para instâncias de 20 e 32 locações	178
E.1	Melhor solução (obtida de 10 execuções) da heurística ILS+BTP para instâncias de 6 e 12 locações	181
E.2	Melhor solução (obtida de 10 execuções) da heurística ILS+BTP para instâncias de 20 e 32 locações	182

E.3	Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos ILS+BTP para instâncias de 6 e 12 locações	183
E.4	Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos ILS+BTP para instâncias de 20 e 32 locações	184
E.5	Tempo Médio das solução ILS+BTP para instâncias de 6 e 12 locações em 10 execuções de cada algoritmo	185
E.6	Tempo Médio das solução ILS+BTP para instâncias de 20 e 32 locações em 10 execuções de cada algoritmo	186
F.1	Melhor solução (obtida de 10 execuções) da heurística VNS+BTP com 20 iterações VNS e 100 iterações BTP para instâncias de 6 e 12 locações	189
F.2	Melhor solução (obtida de 10 execuções) da heurística VNS+BTP com 20 iterações VNS e 100 iterações BTP para instâncias de 20 e 32 locações	190
F.3	Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos VNS+BTP para instâncias de 6 e 12 locações	191
F.4	Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos VNS+BTP para instâncias de 20 e 32 locações	192
F.5	Tempo Médio das solução VNS+BTP para instâncias de 6 e 12 locações em 10 execuções de cada algoritmo	193
F.6	Tempo Médio das solução VNS+BTP para instâncias de 20 e 32 locações em 10 execuções de cada algoritmo	194
G.1	Melhor solução (obtida de 10 execuções) do GRASP+BTP para instâncias de 6 e 12 locações	197
G.2	Melhor solução (obtida de 10 execuções) do GRASP+BTP para instâncias de 20 e 32 locações	198
G.3	Desvio percentual entre a melhor solução conhecida e melhor solução de GRASP+BTP para instâncias de 6 e 12 locações	199
G.4	Desvio percentual entre a melhor solução conhecida e melhor solução de GRASP+BTP para instâncias de 20 e 32 locações	200
G.5	Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP+BTP para instâncias de 6 e 12 locações	201
G.6	Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP+BTP para instâncias de 20 e 32 locações	202
H.1	Desvio percentual entre custo da melhor solução da literatura e dos algoritmos propostos	204
H.2	Desvio percentual entre custo da melhor solução da literatura e dos algoritmos propostos	205

H.3	Desvio percentual entre custo da melhor solução da literatura e dos algoritmos propostos	205
I.1	Características das instâncias PADE de 6 e 12 locações	208
I.2	Características das instâncias PADE de 20 e 32 locações	209
I.3	Atividades, período de início e fim e recursos das atividades na instância 25	210
I.4	Solução CPLEX - alocação de atividades para instância 25	210
I.5	Solução CPLEX - alocação de recursos ociosos para instância 25	210
I.6	Solução CPLEX para a instância 25	210
I.7	Atividades, período de início e fim e recursos das atividades na instância 26	211
I.8	Solução CPLEX - alocação de atividades para instância 26	211
I.9	Solução CPLEX - alocação de recursos ociosos para instância 26	211
I.10	Solução CPLEX para a instância 26	211
I.11	Atividades, período de início e fim e recursos das atividades na instância 28	212
I.12	Solução CPLEX - alocação de atividades para instância 28	212
I.13	Solução CPLEX - alocação de recursos ociosos para instância 28	212
I.14	Solução CPLEX para a instância 28	212
I.15	Atividades, período de início e fim e recursos das atividades na instância 32	213
I.16	Solução CPLEX - alocação de atividades para instância 32	213
I.17	Solução CPLEX - alocação de recursos ociosos para instância 32	213
I.18	Solução CPLEX para a instância 32	213

Capítulo 1

Introdução

O Problema da Alocação Dinâmica de Espaços (PADE) é relativamente novo na literatura e foi inspirado na necessidade de otimização da distância percorrida por recursos requeridos para a realização de atividades em projetos. Um projeto é dividido por um número de períodos consecutivos e, em cada um deles, uma quantidade de atividades é realizada. Os recursos necessários para as atividades devem ser associados a espaços de trabalho e, os recursos ociosos no período devem ser guardados em depósitos. O objetivo do problema é minimizar a distância total percorrida pelos recursos entre os espaços (de trabalho ou depósitos).

Este problema é considerado difícil para se resolver computacionalmente e, portanto, a dificuldade em encontrar uma solução ótima para instâncias de elevadas dimensões justifica o uso de métodos aproximados. Os métodos *simulated annealing* e busca tabu foram aplicados em [17, 18, 19].

O presente trabalho apresenta, primeiramente, algoritmos de construção de solução inicial e de busca local a partir dos quais são obtidos métodos heurísticos baseados nas metaheurísticas ILS, VNS, GRASP e Busca Tabu a fim de que sejam encontradas boas soluções em tempo viável. Em seguida, é apresentada uma proposta de versões híbridas de algoritmos a partir de dois ou mais métodos heurísticos. O conjunto de instâncias e um modelo matemático disponível na literatura são utilizados junto ao CPLEX para obtenção de soluções exatas. Nesta etapa do trabalho, foi possível chegar à soluções ótimas para instâncias com valor exato em aberto, comprovar a eficiência das heurísticas propostas e impugnar alguns resultados da literatura.

O Capítulo seguinte descreve o PADE, apresenta problemas correlacionados e um exemplo de instância para o problema. O Capítulo 3 faz a apresentação do estado-da-arte e portanto é feita uma revisão bibliográfica, reproduzidos os modelos matemáticos disponíveis, descritos os algoritmos da literatura e o conjunto de problemas-testes. No Capítulo 4 são propostas heurísticas de construção e de busca

local e alguns resultados preliminares são apresentados. Heurísticas baseadas nas metaheurísticas ILS, VNS, GRASP e Busca Tabu são tema do Capítulo 5. O Capítulo 6 apresenta a implementação de algumas heurísticas híbridas. No Capítulo 7 os resultados computacionais são discutidos, incluindo entre eles os custos ótimos em instâncias onde a solução exata é conhecida. E no encerramento deste trabalho, são feitas algumas conclusões e direções futuras para esta pesquisa.

Capítulo 2

O Problema da Alocação Dinâmica de Espaços

O Problema da Alocação Dinâmica de Espaços (PADE) foi inspirado na necessidade de otimização da distância percorrida por recursos necessários para a realização de atividades em projetos. Originalmente foi proposto tendo como aplicação a manutenção de redes de energia nuclear. Posteriormente, verificou-se que o mesmo tem aplicações potenciais em projetos onde movimentar um recurso é uma tarefa difícil ou cara, ou nos quais o congestionamento desses recursos é indesejável, como em construção de pontes e atividades de mineração.

Em todos esses casos, o espaço é um recurso escasso importante para a redução dos custos de manipulação de materiais, através da alocação eficiente das atividades e de equipamentos e materiais de construção ativos ou ociosos, sendo o seu uso eficiente fundamental para a redução das ocorrências de congestionamento.

A despeito de ser um problema de definição relativamente recente, sua importância pode ser justificada pelas aplicações já citadas e ainda do ponto de vista teórico, já que o PADE apresenta relações muito próximas com outros problemas importantes na área de Otimização Combinatória: o Problema Quadrático de Alocação (PQA) [1, 2, 3, 4, 5, 6], o Problema Dinâmico de *Layout* de Facilidades (PDLF) [7, 8, 9, 10, 11, 12] e o Problema Quadrático de Alocação Generalizado (PQAG) [13, 14, 15, 16], referenciados na literatura, respectivamente, como *Quadratic Assignment Problem* (QAP), *Dynamic Facility Layout Problem* (DFLP) e *Generalized Quadratic Assignment Problem* (GQAP).

O PQA foi introduzido por Koopmans e Beckmann em 1957 [1] e ainda hoje é um dos problemas mais difíceis da classe NP-Difícil [6]. O PQA pode ser definido como o problema da alocação de n facilidades a n locações. O objetivo do problema é, sendo conhecidas as distâncias entre os pares de locações e os fluxos de algum tipo de demanda entre os pares de facilidades, encontrar uma alocação de custo mínimo.

Resumidamente, o PDFL consiste em encontrar a melhor disposição de facilidades em locações durante uma construção. Definido por Rosenblatt em 1986 [7], tem por meta minimizar o custo de fluxo de materiais e o custo de realocação de cada facilidade a uma locação específica em múltiplos períodos. O problema pode ser visto, então, como uma sucessão de PQA's. Isto é, a cada período um PQA deve ser resolvido.

O PADE pode ser dividido em dois sub-problemas: o sub-problema de associar atividades a espaços de trabalho e o sub-problema de alocar recursos ociosos a depósitos. O primeiro sub-problema comporta-se como o PDFL devendo-se notar que no PADE, o custo é dado pela distância total percorrida pelos recursos durante o planejamento e no PDLF consiste da soma do custo do fluxo do material mais o custo de realocação de cada facilidade. Similar ao segundo sub-problema tem-se o PQAG. O problema, proposto por Lee e Ma em 2005 [13], considera alocar múltiplas facilidades a uma única locação de acordo com sua capacidade. Para cada par de facilidades é conhecido seu fluxo e para cada par de locações, a sua distância. O objetivo é encontrar uma alocação de custo de transporte mínimo respeitando a capacidade das locações.

Para definir o PADE, considere uma agenda de atividades de um projeto dividida em *períodos* de tempo. A cada período um conjunto de *atividades* deve ser realizado e cada atividade necessita de um conjunto de *recursos* para ser executada. Um *recurso* pode ser considerado *necessário* quando for utilizado por alguma atividade do período, ou *ocioso*, caso contrário, isto é, quando nenhuma das atividades realizadas no período o requer. Uma *solução* do problema consiste na associação dos recursos a espaços próprios, a saber: atividades e seus recursos a *espaços de trabalho* e recursos ociosos a *depósitos*. O objetivo do problema é minimizar o deslocamento dos recursos entre os espaços.

São dados de entrada do problema:

- o *layout* (espaços de trabalho e depósitos) do projeto, as distâncias e a capacidade de cada espaço, isto é, a quantidade de recursos que cada um pode suportar;
- a *agenda* do projeto, na qual são determinados itens como o número e o conjunto de atividades a ser realizado por período, e os recursos necessários para a execução de cada atividade.

Na definição do problema, considera-se que somente uma atividade pode ser realizada em cada espaço de trabalho em um determinado período e que a capacidade do espaço de trabalho associado a uma atividade é suficiente para guardar seus respectivos recursos. Além disso, cada atividade requer somente um espaço de trabalho e pelo menos um recurso.

A fim de evitar a movimentação desnecessária de recursos e, assim, minimizar a distância percorrida por eles, considera-se também que, no caso de uma atividade ser realizada por mais de um período consecutivo, esta deve permanecer sempre no mesmo espaço de trabalho em cada um destes períodos.

As Tabelas 2.1 a 2.3 apresentam um exemplo dos dados de entrada para uma instância do problema.

Na Tabela 2.1 são dados os períodos, as atividades (e seus recursos) a serem realizadas e os recursos ociosos em cada período de tempo. Vê-se, portanto, que a duração deste projeto é de 10 períodos. Ao longo deste, 6 atividades devem ser realizadas utilizando no total 9 recursos. A atividade A1, por exemplo, necessita os recursos 6 e 7 e a atividade A2 os recursos 1 e 5. Como estas são as duas atividades a serem realizadas no segundo período, os recursos 2, 3, 4, 8 e 9 são ociosos neste período.

Tabela 2.1: Agenda do projeto

Períodos	Atividades(Recursos)	Recursos Ociosos
1	A1(6,7)	1,2,3,4,5,8,9
2	A1(6,7), A2(1,5)	2,3,4,8,9
3	A2(1,5)	2,3,4,6,7,8,9
4	A2(1,5)	2,3,4,6,7,8,9
5	A2(1,5), A3(4), A4(2,8)	3,6,7,9
6	A3(4), A4(2,8)	1,3,5,6,7,9
7	A4(2,8), A5(3,4)	1,5,6,7,9
8	A5(3,4)	1,2,5,6,7,8,9
9	A5(3,4), A6(9)	1,2,5,6,7,8
10	A6(9)	1,2,3,4,5,6,7,8

A Tabela 2.2 mostra um exemplo de *layout* contendo 6 espaços. Destes, na primeira fileira temos 3 espaços de trabalho, *ES1*, *ES2* e *ES3* e, na segunda, 3 depósitos, *D1*, *D2*, *D3*.

Tabela 2.2: *Layout* com 6 espaços

<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
<i>D1</i>	<i>D2</i>	<i>D3</i>

A Tabela 2.3 apresenta as distâncias entre os espaços, considerando-se a distância de Manhattan. As colunas/linhas 1 a 3 representam os espaços de trabalho e as colunas/linhas de 4 a 6, os depósitos.

Tabela 2.3: Matriz de distâncias entre locações

	1	2	3	4	5	6
1	0	1	2	1	2	3
2	1	0	1	2	1	2
3	2	1	0	3	2	1
4	1	2	3	0	1	2
5	2	1	2	1	0	1
6	3	2	1	2	1	0

Uma solução para o PADE constitui-se de: *i*) uma matriz de atividades que relaciona períodos, espaços de trabalho e atividades; *ii*) uma matriz de recursos ociosos, descrita por períodos, depósitos e recursos ociosos; e *iii*) a distância total percorrida pelos recursos.

Este exemplo considera que a capacidade máxima de cada espaço é de três recursos. Uma solução para o problema é dada nas Tabelas 2.4 e 2.5, que contêm, respectivamente, a alocação das atividades nos espaços de trabalho *ES1*, *ES2* e *ES3* e a associação de recursos ociosos aos depósitos *D1*, *D2* e *D3*.

Tabela 2.4: Solução - alocação de atividades

Período	Espaço de trabalho		
	<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
1	A1 (6,7)		
2	A1 (6,7)	A2 (1,5)	
3		A2 (1,5)	
4		A2 (1,5)	
5	A3 (4)	A2 (1,5)	A4 (2,8)
6	A3 (4)		A4 (2,8)
7	A5 (3,4)		A4 (2,8)
8	A5 (3,4)		
9	A5 (3,4)		A6 (9)
10			A6 (9)

Tabela 2.5: Solução - alocação de recursos ociosos

Período	Depósito								
	<i>D1</i>		<i>D2</i>		<i>D3</i>				
1	4	3	1	5	2	8	9		
2	4	3			2	8	9		
3	4	3	6	7		2	8	9	
4	4	3	6	7		2	8	9	
5		3	6	7			9		
6		3	6	7	1	5	9		
7			6	7	1	5	9		
8			6	7	1	5	2	8	9
9			6	7	1	5	2	8	
10	4	3	6	7	1	5	2	8	

A alocação de recursos pode ser entendida da seguinte maneira: no período 1, a atividade $A1$ entra em execução e seus recursos são alocados no espaço de trabalho $ES1$. Os recursos ociosos 3 e 4, 1 e 5 e 2, 8 e 9 são associados aos depósitos $D1$, $D2$ e $D3$, respectivamente. No período 2 os recursos da atividade $A2(1,5)$ são transportados do depósito $D2$ para o espaço de trabalho $ES2$. No período 3 os recursos 6 e 7 da atividade $A1$ tornam-se ociosos. O recurso 6 é associado ao depósito $D1$ e o recurso 7 ao depósito $D2$. No período 5 as atividades $A3$ e $A4$ entram em execução e, por isso, o recurso 4 ($A3$) é levado do depósito $D1$ para o espaço de trabalho $ES1$ e os recursos 2 e 8 ($A4$) são movimentados de $D3$ para $ES3$. No período 6 os recursos da atividade $A2$ são levados do espaço de trabalho $ES2$ para o depósito $D2$. No período 7 começa a realização da atividade $A5$. O recurso 3 é levado do depósito $D1$ para $ES1$ e o recurso 4 que já estava sendo utilizado não é movimentado. No período 8 os recursos 2 e 8 da atividade $A4$ que deixa de ser executada são movidos do espaço $ES3$ para $D3$. No período 9 a atividade $A6$ é iniciada e o recurso 9 é transportado do depósito $D3$ para o espaço de trabalho $ES3$ e, finalmente, no período 10 a atividade $A5$ termina e seus recursos saem do espaço de trabalho $ES1$. Os recursos 3 e 4 são levados para o depósito $D1$.

A movimentação dos recursos detalhada no parágrafo anterior, pode ser sumariada na Tabela 2.6. A distância total percorrida pelos recursos é de 16 unidades.

Tabela 2.6: Resumo da movimentação de recursos da agenda-exemplo

Rec.	Períodos									
	1	2	3	4	5	6	7	8	9	10
1		$D2 \rightarrow ES2$				$ES2 \rightarrow D2$	$D1 \rightarrow ES1$	$ES3 \rightarrow D3$		
2					$D3 \rightarrow ES3$					
3					$D1 \rightarrow ES1$	$ES2 \rightarrow D2$				
4										
5		$D2 \rightarrow ES2$								
6			$ES1 \rightarrow D1$							
7			$ES1 \rightarrow D2$							
8					$D3 \rightarrow ES3$			$ES3 \rightarrow D3$		
9									$D3 \rightarrow ES3$	
Dist.	0	2	3	0	3	2	1	2	1	2

Capítulo 3

Estado-da-arte

Este capítulo faz um apanhado geral dos avanços do problema até o momento atual. Nele, é apresentada uma revisão bibliográfica, são descritos os algoritmos e as instâncias propostas na literatura para o problema.

3.1 Revisão Bibliográfica

A literatura apresenta ainda poucos trabalhos dedicados ao Problema da Alocação Dinâmica de Espaços. De fato, encontramos apenas três artigos, do mesmo grupo de pesquisa, tratando diretamente instâncias do problema assim como propondo algoritmos que gerem soluções.

O PADE foi definido em 2005 por McKendall Jr. *et al.* [17]. Como descrito anteriormente, ele foi suscitado pela tarefa de minimizar a distância percorrida por recursos, tais como ferramentas, equipamentos, peças de reposição, entre outros, na ocorrência de desligamentos planejados em centrais de energia elétrica. Este trabalho apresenta, além da própria definição do problema, uma formulação matemática. Os autores propõem um conjunto de instâncias para o problema, parte das quais é resolvida utilizando o CPLEX. É citado que para as instâncias maiores os resultados obtidos pelo CPLEX em 72h de execução são de qualidade baixa, mas os *gaps* para estas instâncias não são relatados. Duas heurísticas baseadas na metaheurística *simulated annealing* (seção 3.2.3) são propostas e os resultados obtidos comparados. Os autores concluem que as heurísticas obtiveram bons resultados tanto em relação à qualidade das soluções quanto pelo tempo computacional.

Em 2006, McKendall Jr. e Jaramillo [18] propuseram novas soluções para as instâncias do problema através de cinco algoritmos de construção e uma busca tabu. Este trabalho adicionou ao conjunto de considerações do problema, definidos no trabalho anterior, uma nova restrição: se um recurso é ocioso por períodos de tempo consecutivos ele deve ser associado ao mesmo depósito. São citadas duas razões para a adição desta restrição: Primeiro, a redução do número de movimentos dos recursos.

Isto porque, em uma situação real, o tempo de preparação de um recurso para sua movimentação pode ser alto. Os autores citam que este custo deve ser considerado na função objetivo ou então esta restrição deve ser obedecida. A outra razão é a diminuição do tamanho do espaço de soluções, o que facilitaria a resolução. Testes mostram que a heurística busca tabu proposta supera os resultados anteriores.

O terceiro trabalho endereçando o PADE foi publicado em 2007 [19]. O autor generaliza o problema fazendo com que o objetivo seja a minimização da realocação dos recursos ao invés da otimização somente da distância. Como custos de realocação são considerados o custo de transporte e o de preparação dos recursos. Neste trabalho, a imposição de não haver movimentos de recursos ociosos que ocorrem em períodos consecutivos feita em [18], não é mantida. Um modelo matemático modificado (seção 3.1.1) a fim de incorporar o custo de realocação é apresentado. Além disso, o trabalho apresenta três diferentes heurísticas baseadas em busca tabu: a primeira uma busca tabu simples; a segunda considera estratégias de intensificação e diversificação e a terceira trata de uma busca tabu probabilística. A metodologia apresentada é comparada com os algoritmos da literatura e os resultados reportados apontam a heurística que implementa intensificação e diversificação como o melhor método. Apesar de terem proposto uma nova formulação para o problema, esta não é utilizada já que os algoritmos são testados nas mesmas instâncias dos trabalhos anteriores, que não previam os itens acrescentados. Além disso, é possível comprovar, através do uso do CPLEX, que o trabalho apresenta alguns resultados inconsistentes, conforme pode ser visto no Capítulo 7.

A Tabela 3.1 apresenta as principais diferenças apresentadas nos trabalhos da literatura para o PADE.

Tabela 3.1: Principais diferenças entre os trabalhos da literatura

Trabalho 1	Minimiza custo de transporte de recursos Heurística <i>Simulated Annealing</i>
Trabalho 2	Minimiza custo de transporte de recursos Heurística Busca Tabu Recursos ociosos em períodos consecutivos são alocados sempre ao mesmo depósito
Trabalho 3	Minimiza custo de realocação e custo de transporte de recursos Heurística Busca Tabu com lista dinâmica e mecanismos de memória mais avançados

3.1.1 Modelos matemáticos

Nesta seção são apresentados dois modelos matemáticos da literatura para o PADE. A diferença principal entre os dois está no objetivo do problema. O PADE necessita os seguintes dados para sua definição formal:

- J = conjunto de atividades ($j = 1, 2, \dots, |J|$);
- R = conjunto de recursos ($r = 1, 2, \dots, |R|$);
- P = conjunto de períodos ($p = 1, 2, \dots, |P|$);
- N = número total de espaços do *layout* (espaços de trabalho e depósitos);
- L = conjunto de espaços, $|L| = N$ ($l = 1, 2, \dots, N$);
- W = conjunto de espaços de trabalho, onde $W \subset L$ ($w \in W$);
- S = conjunto de depósitos, onde $S \subset L$ ($s \in S$) e $W \cup S = L$;
- R_j = conjunto de recursos necessários para realizar a atividade j ;
- I_p = conjunto de recursos ociosos no período p ;
- A_p = conjunto de atividades no período p ;
- d_{kl} = a distância entre os espaços k e l ;
- C_s = capacidade do depósito s ;

Formulação $\mathcal{F}1$ [17]

Nesta abordagem o objetivo é *minimizar a soma das distâncias percorridas pelos recursos durante todo o projeto*. Considerando a variável de decisão x_{prk} que se torna 1 se no período p , o recurso r está associado ao espaço k e 0 caso contrário; e a variável de decisão y_{jw} que assume valor 1 se a atividade j é realizada no espaço de trabalho w e 0 caso contrário, pode-se definir formalmente o PADE como:

Minimizar:

$$\sum_{r=1}^R \sum_{k=1}^N \sum_{l=1}^N \sum_{p=1}^{|P|-1} d_{kl} x_{prk} x_{p+1rl} \quad (3.1)$$

Sujeito a:

$$\sum_{s \in S} x_{prs} = 1, \quad \forall p \in P, \forall r \in I_p, \quad (3.2)$$

$$\sum_{r \in I_p} x_{prs} \leq C_s, \quad \forall s \in S, \forall p \in P, \quad (3.3)$$

$$\sum_{w \in W} y_{jw} = 1, \quad \forall j \in J, \quad (3.4)$$

$$\sum_{j \in A_p} y_{jw} \leq 1, \quad \forall w \in W, \forall p \in P, \quad (3.5)$$

$$\sum_{r \in R_j} x_{prk} = |R_j| y_{jw}, \quad \forall p \in P, \forall j \in A_p, \forall w \in W, \quad (3.6)$$

$$x_{prk} = 0 \text{ ou } 1, \quad \forall p \in P, \forall r \in R, \forall k \in L, \quad (3.7)$$

$$y_{jw} = 0 \text{ ou } 1, \quad \forall j \in J, \forall w \in W. \quad (3.8)$$

A função objetivo (3.1) minimiza a distância percorrida pelos recursos ao longo dos períodos em que o projeto é realizado. As restrições 3.2 garantem que cada um dos recursos ociosos, em cada período, seja associado a somente um depósito e as 3.3 asseguram que a capacidade dos depósitos seja respeitada em cada período de tempo. As restrições 3.4 e 3.5 garantem, respectivamente, que toda atividade seja alocada a um único espaço de trabalho e que cada espaço de trabalho tenha no máximo uma atividade associada. As restrições 3.6 garantem que todos os recursos necessários à realização de uma atividade sejam associados ao mesmo espaço de trabalho ao qual a atividade tenha sido associada. Finalmente, as restrições 3.7 e 3.8 determinam que as variáveis de decisão são binárias.

Formulação $\mathcal{F}2$ [19]

Em seguida é apresentada uma formulação modificada a fim de incorporar o custo de realocação dos recursos durante o horizonte de planejamento do projeto que, além dos dados apresentados no início desta seção, considera os seguintes índices:

- a_{prkl} = a soma dos custos de carregar o recurso r no espaço k e descarregá-lo na espaço l no período p .
- c_{prkl} = o custo unitário (unidade de distância) de mover o recurso r do espaço k para o espaço l no período p .
- q_r = o espaço total necessário para acomodar o recurso r ;
- C_{ps} = a capacidade do espaço s no período p .

Minimizar:

$$\sum_{r=1}^R \sum_{k=1}^N \sum_{l=1}^N \sum_{p=1}^{P-1} a_{prkl} x_{prk} x_{p+1rl} + \sum_{r=1}^R \sum_{k=1}^N \sum_{l=1}^N \sum_{p=1}^{P-1} c_{prkl} d_{kl} x_{prk} x_{p+1rl} \quad (3.9)$$

Sujeito a:

Equações 3.2, 3.4 – 3.8 e

$$\sum_{r \in I_p} q_r x_{prs} \leq C_{ps}, \quad \forall s \in S, \forall p \in P, \quad (3.10)$$

Esta formulação merece alguns comentários:

- $\mathcal{F}2$ traz melhoria em relação a $\mathcal{F}1$ quando inclui a variável q_r que corresponde ao espaço total necessário para acomodar o recurso r . Isto torna o modelo mais próximo de uma situação real, já que um recurso pode ter um volume diferente de outro;
- os custos de preparação e de movimentação de recursos e a capacidade de depósitos foram definidos de forma que possam variar de acordo com o período, o que não parece fazer muito sentido em uma aplicação de fábrica. Como o problema foi originalmente formulado para manutenção de redes elétricas, esta variabilidade pode ter significado naquele caso. Além disso, traz uma modificação grande nas instâncias do problema e não aparecem no trabalho instâncias que possam aproveitar a maior latitude do modelo, já que o único conjunto de instâncias disponível continua a ser o original.

3.2 Algoritmos da Literatura

Os algoritmos propostos nos trabalhos citados na seção 3.1 são apresentados aqui. Primeiramente, são descritos os algoritmos de construção e, em seguida, definidos os movimentos e algoritmos baseados em metaheurísticas que, aplicados a uma solução, tem por objetivo atingir um ótimo local.

3.2.1 Algoritmos de Construção

Algoritmo de Construção *First Assignment 1* (FA1) [17]

Este algoritmo constrói uma solução para o PADE em duas fases: a primeira alocando atividades e a segunda, os recursos ociosos. Na primeira fase, a primeira atividade do período é associada ao primeiro espaço de trabalho disponível, ou seja, que não é ocupado por nenhuma atividade, a segunda atividade ao segundo espaço disponível e assim por diante, concluindo-se esta fase quando cada atividade tiver sido alocada a um espaço de trabalho. Se uma atividade é realizada em períodos consecutivos, ela deve ser associada ao mesmo espaço de trabalho em cada um destes períodos. Na segunda fase do algoritmo, para cada período, um subconjunto do conjunto de recursos ociosos é associado ao primeiro depósito. O número de recursos deste subconjunto é determinado pela capacidade do depósito. O segundo depósito recebe os recursos do segundo subconjunto de recursos ociosos e assim por diante até que todos eles tenham sido alocados.

Algoritmo de Construção *First Assignment 2* (FA2) [18]

Trata-se de uma adaptação de FA1, com modificações na fase de associação de recursos ociosos. Desta forma, a primeira fase permanece inalterada. Na segunda fase, a associação dos recursos ociosos é modificada somente para aqueles que permanecem ociosos por períodos consecutivos. Neste caso, o recurso é associado ao mesmo depósito em todos os períodos que permanecer ocioso.

Algoritmo de Construção *Randomized Clustering* (RC) [18]

Este algoritmo constrói uma solução parcial para o PADE em duas fases: A primeira segue a idéia do algoritmo de clusterização de Chan e Milner [20], utilizado no problema de células de manufatura, para alocação de atividades. Em seguida é utilizada *Randomized Storage Policy* (RSP) para associar os recursos ociosos a depósitos, similar à proposta de Tompkins *et al.* [21] para associação de produtos em almoxarifados.

RC aloca atividades a espaços de trabalho da seguinte maneira: são criadas duas matrizes, uma chamada matriz Atividades-Recursos que associa cada recurso às atividades nas quais ele é utilizado e, a outra, denominada matriz de Escalonamento que associa atividades a períodos. O procedimento agrupa em *clusters* atividades que ocorrem em períodos diferentes, respeitando o limite de capacidade do espaço de trabalho. Após o processo de clusterização, em cada grupo, tem-se as atividades associadas a um espaço de trabalho.

A solução determinando a alocação dos recursos ociosos a depósitos, obtida por RSP, é feita da seguinte maneira: após se obter a completa alocação das atividades, para alocar cada recurso ocioso de um período é verificado o depósito (observada sua capacidade) mais próximo do espaço de trabalho ao qual está associada a atividade que necessita o recurso ocioso, a este depósito o recurso é alocado. Os recursos ociosos em um período que são requeridos mais cedo por atividades são associados primeiro. Além disso, é necessário alocar os recursos que não serão mais utilizados por atividades no restante dos períodos, neste caso, o recurso é alocado ao depósito mais próximo do espaço de trabalho ao qual está associada a última atividade que o utilizou.

3.2.2 Movimentos

Um *movimento* pode ser definido como uma modificação em uma solução s que a transforma em outra solução s' vizinha de s . Os movimentos definidos para o problema em [18] e [17] podem ser de atividades ou de recursos ociosos, como descritos e exemplificados a seguir, tendo como base as Tabelas 2.4 e 2.5.

Movimentos de atividades

Movimento \mathcal{M}_1 : Troca dos espaços de trabalho de duas ou mais atividades nos períodos consecutivos aos quais elas estejam alocadas (Tabela 3.2).

Tabela 3.2: Exemplo do movimento \mathcal{M}_1

Período	Espaço de trabalho		
	<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
1		1 (6,7)	
2	2 (1,5)	1 (6,7)	
3	2 (1,5)		
4	2 (1,5)		
5	2 (1,5)	3 (4)	4 (2,8)
6		3 (4)	4 (2,8)
7	5 (3,4)		4 (2,8)
8	5 (3,4)		
9	5 (3,4)		6 (9)
10			6 (9)

Movimento \mathcal{M}_2 : Remoção de uma atividade de um espaço de trabalho para outro que esteja disponível em um ou mais períodos consecutivos aos quais ela esteja alocada (Tabela 3.3).

Tabela 3.3: Exemplo do movimento \mathcal{M}_2

Período	Espaço de trabalho		
	<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
1			1 (6,7)
2		2 (1,5)	1 (6,7)
3		2 (1,5)	
4		2 (1,5)	
5	3 (4)	2 (1,5)	4 (2,8)
6	3 (4)		4 (2,8)
7	5 (3,4)		4 (2,8)
8	5 (3,4)		
9	5 (3,4)		6 (9)
10			6 (9)

Movimento \mathcal{M}_3 : Combinação de \mathcal{M}_1 e \mathcal{M}_2 (Tabela 3.4).

Tabela 3.4: Exemplo do movimento \mathcal{M}_3

Período	Espaço de trabalho		
	<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
1	1 (6,7)		
2	1 (6,7)	2 (1,5)	
3		2 (1,5)	
4		2 (1,5)	
5	4 (2,8)	2 (1,5)	3 (4)
6	4 (2,8)		3 (4)
7	4 (2,8)		5 (3,4)
8			5 (3,4)
9	6 (9)		5 (3,4)
10	6 (9)		

Movimentos de Recursos Ociosos

Movimento \mathcal{M}_4 : Troca dos depósitos de dois recursos (Tabela 3.5).

Tabela 3.5: Exemplo do movimento \mathcal{M}_4

Período	Depósito					
	D1		D2		D3	
1	1	3		4	5	2 8 9
2	4	3				2 8 9
3	4	3	6	7		2 8 9
4	4	3	6	7		2 8 9
5		3	6	7		9
6		3	6	7	1 5	9
7			6	7	1 5	9
8			6	7	1 5	2 8 9
9			6	7	1 5	2 8
10	4	3	6	7	1 5	2 8

Movimento \mathcal{M}_5 : Remoção de um recurso de um depósito para outro no qual a capacidade não seja ultrapassada (Tabela 3.6).

Tabela 3.6: Exemplo do movimento \mathcal{M}_5

Período	Depósito					
	D1		D2		D3	
1	4	3	1		5	2 8 9
2	4	3				2 8 9
3	4	3	6	7		2 8 9
4	4	3	6	7		2 8 9
5		3	6	7		9
6		3	6	7	1 5	9
7			6	7	1 5	9
8			6	7	1 5	2 8 9
9			6	7	1 5	2 8
10	4	3	6	7	1 5	2 8

Movimento \mathcal{M}_6 : Troca dos depósitos de dois ou mais recursos nos períodos consecutivos aos quais estes estejam alocados (Tabela 3.7).

Tabela 3.7: Exemplo do movimento \mathcal{M}_6

Período	Depósito					
	D1		D2		D3	
1	2	3		1	5	4 8 9
2	2	3				4 8 9
3	2	3	6	7		4 8 9
4	2	3	6	7		4 8 9
5		3	6	7		9
6		3	6	7	1 5	9
7			6	7	1 5	9
8			6	7	1 5	2 8 9
9			6	7	1 5	2 8
10	4	3	6	7	1 5	2 8

Movimento \mathcal{M}_7 : Remoção de um recurso de um depósito para outro nos períodos consecutivos aos quais este esteja alocado (Tabela 3.8).

Tabela 3.8: Exemplo do movimento \mathcal{M}_7

Período	Depósito					
	D1		D2		D3	
1	3		1	5	4	2 8 9
2	3				4	2 8 9
3	3	6	7		4	2 8 9
4	3	6	7		4	2 8 9
5	3	6	7			9
6	3	6	7	1	5	9
7		6	7	1	5	9
8		6	7	1	5	2 8 9
9		6	7	1	5	2 8
10	4	3	6	7	1	5 2 8

Movimento \mathcal{M}_8 : Combinação de \mathcal{M}_6 e \mathcal{M}_7 (Tabela 3.9).

Tabela 3.9: Exemplo do movimento \mathcal{M}_8

Período	Depósito					
	D1		D2		D3	
1	9	3	1	5		2 8 4
2	9	3				2 8 4
3	9	3	6	7		2 8 4
4	9	3	6	7		2 8 4
5	9	3	6	7		
6	9	3	6	7	1	5
7	9		6	7	1	5
8	9		6	7	1	5 2 8
9			6	7	1	5 2 8
10	4	3	6	7	1	5 2 8

Os movimentos de \mathcal{M}_1 a \mathcal{M}_5 foram propostos em [17], enquanto, \mathcal{M}_6 a \mathcal{M}_8 em [18] obedecendo a restrição imposta no trabalho, conforme já comentado na seção 3.1.

3.2.3 Metaheurísticas

Uma metaheurística pode ser definida como um processo iterativo que guia uma heurística subordinada pela combinação de conceitos capazes de explorar e valorizar o espaço de busca, utilizando o aprendizado com a estrutura do problema para que se possa chegar, de forma eficiente, a soluções próximas do ótimo [22]. As heurísticas apresentadas mais à frente utilizam os movimentos apresentados na seção anterior e os conceitos das metaheurísticas *simulated annealing* e busca tabu para o melhoramento de soluções iniciais construídas para o PADE. A seguir é feita uma breve descrição das metaheurísticas *simulated annealing* e busca tabu.

A metaheurística *Simulated Annealing*

A metaheurística *simulated annealing* é uma técnica de busca local probabilística com ajuste automático de parâmetro. Foi proposta por Kirkpatrick *et al.* [23] inspirados em um processo da metalurgia, conhecido como “recozimento”, que é o

aquecimento seguido de um resfriamento lento de um metal. Nesta metaheurística, a cada passo o método gera uma solução vizinha a partir da solução corrente. Se a solução vizinha for melhor do que a corrente, ela é aceita. Caso contrário, ela pode ser aceita com uma probabilidade que depende da diferença entre os custos destas soluções e um parâmetro T , chamado de temperatura. Analogamente ao processo físico, o valor da temperatura decresce durante o processo. Conforme a função utilizada, quando T é grande, a solução corrente muda quase aleatoriamente, mas à medida que T se aproxima de zero, a probabilidade de aceitar movimentos de piora diminui. Desta forma, o método tende a escapar de ótimos locais no início apenas.

Mais detalhadamente, a busca por um ótimo local a partir de uma solução inicial qualquer ocorre da seguinte maneira: o laço principal do procedimento executa um movimento. O vizinho y' obtido através do movimento aplicado à solução corrente y^c é avaliado. É calculada a diferença Δ entre os custos das soluções, como: $\Delta = f(y') - f(y^c)$. Considerando um problema de minimização, como é o caso do PADE, um movimento é dito aceito se $\Delta < 0$. Desta forma, a solução y' passa a ser a solução corrente e o custo da solução corrente $TC(y^c)$ passa a ser $TC(y')$. Se $\Delta \geq 0$ o movimento *pode* ser aceito com uma probabilidade $e^{-\Delta/T^c}$, sendo T^c um parâmetro, chamado temperatura corrente. No começo do procedimento, T^c corresponde à temperatura inicial T_0 , um valor bastante elevado, forma de evitar que o método fique preso a ótimos locais, estabelecendo uma probabilidade alta de que soluções de piora sejam aceitas. Após um número dado de iterações nas quais uma temperatura permanece a mesma, a temperatura corrente é recalculada pela equação $T^c = T_0\alpha_{r-1}$, onde: α é a taxa de resfriamento e $r = 1, 2, \dots$ valor unitário de diminuição da temperatura. Com isto, a chance de que soluções de piora sejam aceitas se reduz gradativamente em função da queda de T^c . O procedimento termina quando um critério de parada é alcançado, que é normalmente quando a temperatura atinge um valor próximo a 0.

Heurística *Simulated Annealing SA1* [17]

A Figura 3.1 apresenta o pseudocódigo de SA1. Inicialmente, o algoritmo recebe os seguintes parâmetros: T_0 , a temperatura inicial; α , a taxa de resfriamento; $SAmax$, o número de iterações para cada temperatura; Max_Iter , o número máximo de iterações consecutivas sem melhora; p , a probabilidade de realizar um movimento de recursos ociosos e a solução inicial y^0 , obtida através do algoritmo de construção FA1.

Figura 3.1: Algoritmo SA1

```

Heurística SA1( $T_0, \alpha, S_{Amax}, Max\_Iter, p, y^0$ )
1:  $r \leftarrow 1; i \leftarrow 0;$ 
2:  $y^c \leftarrow y^0; TC(y^c) \leftarrow TC(y^0);$ 
3:  $Melhor\_Sol \leftarrow y^c; Melhor\_Custo \leftarrow TC(y^c);$ 
4: enquanto  $i < Max\_Iter$  faça
5:    $j \leftarrow 0;$  /*Contador do número de movimentos de cada temperatura*/
6:    $T_c \leftarrow T_0 \alpha^{r-1};$ 
7:   enquanto  $j < S_{Amax}$  faça
8:     a) Escolher aleatoriamente  $t$  e movimento, probabilidade  $p$  para recursos ociosos;
9:     b) Gerar um vizinho  $y'$  de  $y^c$ ;
10:     $j \leftarrow j + 1;$ 
11:    Calcular  $f(y')$ ;
12:    Calcular  $\Delta \leftarrow f(y') - TC(y^c);$ 
13:    se  $\Delta < 0$  então
14:       $y^c \leftarrow y';$ 
15:       $TC(y^c) \leftarrow f(y');$ 
16:      se  $Melhor\_Custo > TC(y^c)$  então
17:         $Melhor\_Sol \leftarrow y^c;$ 
18:         $Melhor\_Custo \leftarrow TC(y^c);$ 
19:      fim se
20:    se não
21:       $x \leftarrow \text{random}(0,1);$ 
22:      se  $x < e^{-\Delta TC/T_c}$  então
23:         $TC(y^c) \leftarrow f(y');$ 
24:         $i \leftarrow i + 1;$ 
25:      fim se
26:    fim se
27:  fim enquanto
28:   $r \leftarrow r + 1;$ 
29: fim enquanto
30: Retornar  $Melhor\_Sol$  e  $Melhor\_Custo$ ;
31: end

```

Heurística *Simulated Annealing* SA2 [17]

A segunda implementação de *simulated annealing* utiliza a mesma metodologia de SA1, diferenciando-se dela no método para construção da solução inicial, que passa a ser FA2 (ver seção 3.2.1) e nos movimentos de recursos ociosos. Em SA1 os movimentos considerados são \mathcal{M}_1 - \mathcal{M}_3 e \mathcal{M}_4 e \mathcal{M}_5 , enquanto em SA2 são escolhidos entre \mathcal{M}_1 - \mathcal{M}_3 e \mathcal{M}_6 - \mathcal{M}_8 .

Considerações sobre SA1 e SA2

Os dois algoritmos são implementações simples de *simulated annealing*. Os parâmetros do método, segundo os autores, foram definidos teórica e experimentalmente.

A escolha do movimento a ser aplicado pode ser considerada a única proposta original para o método. Um movimento de recursos ociosos (\mathcal{M}_6 - \mathcal{M}_8) tem a probabilidade p de ser escolhido. Durante os testes foram utilizados três valores para p : 0,9, 0,7 e 0,6. Os autores discutiram uma variação no comportamento das duas

versões propostas. Em SA1 as melhores soluções obtidas utilizaram o valor de 90%, enquanto em SA2, os melhores resultados foram obtidos com 70% dos movimentos feitos com recursos ociosos. Esta variação é relacionada à diferença de tamanho do espaço de busca. Em SA1, o espaço de busca é maior do que em SA2.

A comparação reportada entre as duas versões mostra que SA2, em média, pode obter soluções melhores que SA1.

A metaheurística Busca Tabu

A busca tabu foi proposta, individualmente, por Fred Glover [24] e Pierre Hansen [25]. Ela foi imaginada para escape de ótimos locais. Normalmente, a cada iteração, uma busca local percorre a vizinhança de uma solução corrente até obter a melhor solução vizinha, mas isto pode fazer com que ela se fixe em um ótimo local. Para que isso não ocorra, soluções de piora podem ser aceitas. A busca tabu provê ainda uma maneira de evitar o fenômeno conhecido como ciclagem utilizando o conceito de *lista tabu*, que é uma lista onde são armazenados os movimentos reversos dos últimos movimentos realizados. O tamanho T da lista tabu é crítico para o método, uma vez que os movimentos que nela se encontram são proibidos de serem executados por um tempo dado, o que é conhecido por *tabu tenure*. Um movimento tabu pode ser permitido, dentro da faixa de proibição, através de um *critério de aspiração* que absorve o *status* de tabu de um movimento de acordo com a ocorrência de alguma circunstância. Normalmente, aceita-se um movimento tabu se este produzir uma solução melhor que a encontrada até o momento. A solução obtida no fim de cada iteração é considerada a solução corrente. Quando um critério de parada é alcançado o método é encerrado e a melhor solução encontrada durante a busca é retornada.

Busca Tabu BT1 [18]

BT1 utiliza os movimentos \mathcal{M}_1 - \mathcal{M}_3 e \mathcal{M}_6 - \mathcal{M}_8 e mantém duas listas tabu: uma de atividades e outra de recursos ociosos. Os movimentos recentes são guardados nestas listas.

A cada iteração da busca tabu são feitos todos os movimentos possíveis na solução corrente. O movimento de melhor custo é executado se ocorrer uma das duas situações: o movimento é não tabu, ou seja, não foi recentemente executado; ou, se tabu, for autorizado pelo critério de aspiração. Se o movimento de menor custo não for possível, o próximo melhor movimento é considerado. A solução resultante da execução do movimento passa a ser a solução corrente na próxima iteração da busca.

O algoritmo termina quando um número de iterações sem melhora é alcançado.

Este número foi, segundo os autores, definido experimentalmente como o produto do número de recursos, número de atividades e número de períodos. Além disso, foi definido o tamanho de três vezes o número total de atividades, para a lista tabu de atividades e de três vezes o número de conjuntos de recursos ociosos, para a lista tabu de recursos ociosos. A definição de um conjunto de recursos ociosos vem da restrição imposta neste trabalho ([18]) de que recursos que são ociosos em períodos consecutivos devem permanecer no mesmo depósito em todos os períodos nos quais são ociosos. A ideia é de que um recurso ocioso que ocorre em períodos consecutivos seja considerado um só já que gera somente um movimento.

A Figura 3.2 apresenta o pseudocódigo de BT1. Os parâmetros do método são *Iter_Sem_Melhora*, o número de iterações sem melhora; *Sol_Inicial*, a solução inicial obtida por meio de um dos algoritmos de construção apresentados no mesmo trabalho e *Custo_Sol_Inicial*, o custo da solução inicial.

Figura 3.2: Algoritmo BT1

```

Heurística BT1(Iter_Sem_Melhora, Sol_Inicial, Custo_Sol_Inicial)
1:  $i \leftarrow 0$ ;
2:  $Melhor\_Sol \leftarrow Sol\_Inicial$ ;
3:  $Melhor\_Custo \leftarrow Custo\_Sol\_Inicial$ ;
4:  $Sol\_Corrente \leftarrow Sol\_Inicial$ ;
5:  $Custo\_Sol\_Corrente \leftarrow Custo\_Sol\_Inicial$ ;
6: Inicializar a lista tabu de atividades e lista tabu de recursos ociosos;
7: enquanto  $i < Iter\_Sem\_Melhora$  faça
8:   Avaliar todos os possíveis movimentos em  $Sol\_corrente$ ;
9:   Identificar o melhor movimento;
10:  se Movimento é não tabu ou é permitido pelo critério de aspiração então
11:    Executar o movimento e obtém  $Sol$ ;
12:     $Sol\_Corrente \leftarrow Sol$ ;
13:     $Custo\_Sol\_Corrente \leftarrow f(Sol)$ ;
14:  se não
15:    Selecionar e executar o próximo melhor movimento não tabu;
16:  fim se
17:  Atualizar lista tabu de acordo com o tipo do movimento;
18:  se  $Custo\_Sol\_Corrente < Melhor\_Custo$  então
19:     $Melhor\_Sol \leftarrow Sol\_Corrente$ ;
20:     $Melhor\_Custo \leftarrow Custo\_Sol\_Corrente$ ;
21:     $i \leftarrow 0$ ;
22:  se não
23:     $i \leftarrow i + 1$ ;
24:  fim se
25: fim enquanto
26: Retornar  $Melhor\_Sol$  e  $Melhor\_Custo$ ;

end

```

Busca Tabu BT2 [19]

Este algoritmo é descrito pelo autor como um melhoramento de BT1 uma vez que não fixa os recursos que são ociosos em períodos consecutivos ao mesmo depósito.

BT2 utiliza portanto os mesmos movimentos de atividades de BT1 e, diferentemente desta, os movimentos \mathcal{M}_4 e \mathcal{M}_5 de recursos ociosos. Nesta heurística o custo de preparação é explicitamente considerado na função objetivo. Apesar desta afirmação, as instâncias utilizadas nos testes são as mesmas dos demais trabalhos e não consta no artigo nenhuma informação a respeito de custo de preparação, como pode ser visto mais detalhadamente na Seção 3.3.

Busca Tabu BT3 [19]

Esta heurística adiciona as listas tabu tamanho variável e estratégias de diversificação e intensificação aos outros dois métodos baseados em busca tabu.

Mais especificamente, são usadas listas tabu com variação dinâmica de tamanho. Sejam sol_prox , a solução obtida do movimento aplicado à solução corrente e sol_cor , a solução corrente. A cada iteração da busca o tamanho da lista tabu varia entre um limite inferior (LI) e limite superior (LS) de acordo com um percentual de redução (PR) da solução. Considerando TL o tamanho da lista tabu, a variação ocorre de acordo com os seguintes valores, adaptados do trabalho de Chiang e Kouvelis [26] para o PQA:

- Se $PR \leq 0\%$ então sol_prox é pior que sol_cor , a lista é mantida com o mesmo tamanho da iteração anterior.
- Se $0\% \leq PR \leq \alpha\%$ então ou sol_prox é a mesma que sol_cor ou a redução não é significativa. Neste caso, $TL = LI + (LS - LI)PR/\alpha$.
- Se $\alpha\% \leq PR \leq \beta\%$ então sol_prox apresenta uma redução significativa de sol_cor . $TL = LS$.
- Se $\alpha\% \leq \beta\%$ e $PR > \alpha\%$ então sol_prox representa uma grande redução em relação a sol_cor . $TL = VA$, onde VA corresponde a um valor alto.

O mecanismo de memória baseada em frequência é originário do utilizado em [26, 27] para o PQA e modificado para o PADE. As listas tabu são usadas para armazenar a frequência dos movimentos e uma função de penalidade é aplicada a movimentos de atividades e recursos ociosos de piora. Considerando $LTA[i][j]$ e $LTR[i][j][p]$ para $i < j$, respectivamente, as listas tabu de atividades e de recursos ociosos, a função de penalidade é definida como:

$$f_A(i, j) = \begin{cases} 0 & \text{se } sol_prox < sol_cor \text{ ou movimento } \in M_4, M_5; \\ p_A LTA[i][j] & \text{caso contrário.} \end{cases}$$

$$f_R(i, j, p) = \begin{cases} 0 & \text{se } sol_prox < sol_cor \text{ ou movimento } \in M_1, M_2, M_3; \\ p_R LTR[i][j][p] & \text{caso contrário.} \end{cases}$$

onde p_A é o valor de penalidade para um movimento de atividade de piora e p_R é o valor de penalidade para um movimento de recursos ociosos de piora. A função objetivo é modificada para avaliar cada solução sol na vizinhança de uma solução obtida através de um movimento de piora: $F = f(sol) + f_A(i, j) + f_R(i, j, p)$.

A estratégia de intensificação é similar à apresentada em [28]. A idéia é que explorar soluções “pobres” no início do procedimento é um desperdício de tempo, então estes movimentos são proibidos até que o percentual de redução PR alcance um certo valor.

Busca Tabu BT4 [19]

Este método consiste de uma busca tabu “probabilística”. É similar à BT2 mas, em vez de selecionar o melhor movimento após explorar toda a vizinhança da solução corrente, em BT4 todos os possíveis movimentos são avaliados e ordenados, e os melhores \mathcal{M} movimentos considerados movimentos candidatos. A partir daí é considerado o primeiro movimento da lista de movimentos candidatos (lmc). Este é aceito com uma probabilidade p . Se o movimento for aceito, a solução resultante passa a ser corrente para a próxima iteração. Caso contrário, o próximo movimento é considerado e aceito com uma probabilidade p , sendo este passo repetido até que um movimento seja selecionado. Se nenhum dos movimento de lmc for aceito, o melhor movimento da lista é escolhido.

Considerações sobre BT1, BT2, BT3 e BT4

A heurística BT3 possui uma série de parâmetros que não são apresentados pelo autor. Ao invés disto, aparece a observação que, pelo fato de BT3 exigir muito mais parâmetros que as demais heurísticas, foi aplicado um grande esforço para atribuir valores a eles. Outro comentário a se fazer é que nem mesmo é apresentada a maneira como é calculado o valor de PR . Uma suposição advinda deste trabalho é que a fórmula para seu cálculo envolva os valores da solução corrente (sol_cor) e solução obtida com a movimentação (sol_prox).

As heurísticas BT1 e BT2 têm por diferença única a maneira como os recursos ociosos são explorados. Na primeira, como já dito, os recursos ociosos que ocorrem em períodos consecutivos são considerados conjuntos de recursos e são movimentados conjuntamente. Na segunda versão, cada recurso é movimentado individualmente.

Observam-se diferenças, para algumas instâncias, na apresentação dos resultados de BT1 mostrados no artigo [18] e os que aparecem na coluna OldTS em [19]. Como

conseqüência aparecem dúvidas sobre a influência da mudança do espaço de busca nestes casos. Por outro lado, para a coluna OldTS não é citada a heurística de construção usada com BT1 para cada caso.

A seção que trata os experimentos computacionais de [19] reporta que BT3, BT4, BT2, BT1 obtiveram as melhores soluções em 79, 54, 42, 35 e 30 das 96 instâncias, respectivamente. Existem, no entanto, todos os problemas citados no parágrafo anterior.

3.3 Instâncias para o PADE

As instâncias propostas para o problema foram apresentadas por McKendall Jr. *et al.*, em [17]. Agradecemos a gentileza dos autores em cedê-las.

O conjunto de testes foi desenvolvido baseado nos seguintes parâmetros:

- N : número de espaços do *layout*, $N = \{6, 12, 20, 32\}$. Metade são espaços de trabalho e a outra metade, depósitos.
- P : número de períodos, com $P = \{10, 15, 20\}$.
- A : número de atividades, $A = \{6, 9, 10, 12, 13, 14, 15, 16, 18, 20, 21, 23, 24, 25, 31, 33, 34, 37, 38, 40, 41, 51, 52, 55, 57, 60, 78, 87\}$.

Quanto à organização dos espaços tem-se:

- problemas com 6 espaços têm *layout* 2x3, sendo, a primeira linha composta de espaços de trabalho e a segunda linha, de depósitos;
- problemas com 12 espaços têm *layout* 2x6 considerando que a primeira linha são espaços de trabalho e a segunda linha, os depósitos;
- problemas com 20 espaços têm *layout* 4x5. As linhas 1 e 4 são depósitos e as linhas 2 e 3, espaços de trabalho;
- problemas com 32 espaços têm *layout* 4x8. Nestes, a primeira e última linhas são depósitos e as duas linhas do meio, espaços de trabalho.

As Tabelas 3.10 e 3.11 exemplificam o *layout* dos problemas com 6 e 20 espaços. No esquema, T e D representam, respectivamente, espaços de trabalho e depósitos.

Tabela 3.10: *Layout* 2x3 para problemas com 6 espaços

T1	T2	T3
D1	D2	D3

Tabela 3.11: *Layout* 4×5 para problemas com 20 espaços

D1	D2	D3	D4	D5
T1	T2	T3	T4	T5
T6	T7	T8	T9	T10
D6	D7	D8	D9	D10

As distâncias entre os espaços foram calculadas utilizando métrica de Manhattan. Além disso, deve-se destacar que:

- a capacidade máxima de cada espaço é de 3 recursos;
- à cada atividade é atribuído no mínimo 1 e no máximo 3 recursos;
- o número total de recursos para cada problema depende do número de espaços. Isto é, problemas com 6, 12, 20 e 32 espaços têm, respectivamente, 9, 18, 30 e 48 recursos.

Baseados nestes fatores os autores geraram 96 instâncias, sendo 24 instâncias para cada valor de N .

É interessante notar, neste conjunto de instâncias, particularidades tais como: o número de recursos por locação é fixo em *três*. Cada recurso, por diferente rótulo que tenha, ocupa o mesmo espaço, ou seja, tem o mesmo volume. O custo de realocação no problema foi definido em um trabalho posterior à proposta das instâncias e não foi considerado.

O Apêndice I apresenta os parâmetros considerados para cada instância.

Capítulo 4

Heurísticas de Construção e Busca Local Propostas para o PADE

Neste capítulo, inicialmente, é apresentada a forma de representar uma solução para o problema, usada neste trabalho. Em seguida, são descritos os algoritmos para a geração de soluções iniciais e os algoritmos de busca local desenvolvidos para o PADE. Finalmente, os resultados computacionais obtidos com a combinação destes algoritmos são apresentados.

4.1 Representação de uma solução

Neste trabalho uma solução para o problema é representada em 3 matrizes:

- uma matriz $\mathcal{A}_{|P|\times|W|}$, onde a_{pw} armazena o número da atividade alocada ao espaço de trabalho w no período p ou 0, caso não haja associação de atividade a w no período p ;
- uma matriz $\mathcal{R}_{|P|\times(\sum_{s=1}^{|S|} C_s)}$, onde r_{pc_s} armazena o número do recurso ocioso alocado a uma posição entre 1 e c_s do depósito s no período p ou 0, caso não haja associação de recurso em c_s de s no período p ;
- uma matriz $\mathcal{L}_{|P|\times|R|}$, onde l_{pr} armazena o número do espaço em que o recurso r está alocado no período p .

As Tabelas 2.4 e 2.5 são exemplos das matrizes \mathcal{A} e \mathcal{R} . A matriz \mathcal{L} foi implementada a fim de facilitar o cálculo da função objetivo, com exemplo apresentado na Tabela 4.1. O cálculo do custo é obtido computando-se, para cada recurso, a distância entre os espaços de dois períodos consecutivos da matriz \mathcal{L} , ou seja, $custo = \sum_{i=1}^R \sum_{j=1}^{P-1} D[\mathcal{L}[i][j]][\mathcal{L}[i][j+1]]$.

Tabela 4.1: Exemplo da matriz \mathcal{L}

Períodos	Recursos								
	1	2	3	4	5	6	7	8	9
1	4	6	4	4	5	1	1	6	6
2	2	6	4	4	2	1	1	6	6
3	2	6	4	4	2	4	5	6	6
4	2	6	4	4	2	4	5	6	6
5	2	3	4	1	2	4	5	3	6
6	4	3	4	1	5	4	5	3	6
7	4	3	1	1	5	4	5	3	6
8	4	6	1	1	5	4	5	6	6
9	4	6	1	1	5	4	5	6	3
10	4	6	4	4	5	4	5	6	3

Utilizamos ainda uma matriz auxiliar $\mathcal{D}_{|P| \times (\sum_{s=1}^{|S|} C_s)}$, proposta por [18], que é uma transformação da matriz \mathcal{R} , obtida através dos seguintes passos:

Iniciando no primeiro período, a cada recurso ocioso é dado um número fictício em todos os períodos em que este recurso ocorrer consecutivamente. A contagem dos números começa em 1 e é incrementada a cada mudança de recurso. No período 2 e nos demais é repetido o mesmo processo para os recursos que ainda não tiverem um número fictício associado.

O objetivo da matriz \mathcal{D} é facilitar o movimento dos recursos ociosos a partir de uma solução para atingir uma solução vizinha quando utilizamos movimentos do tipo $\mathcal{M}6$, $\mathcal{M}7$ e $\mathcal{M}8$. Assim os recursos ociosos que ocorrem em períodos consecutivos podem ser mais facilmente explorados como se fosse um bloco de um mesmo recurso. Um exemplo do resultado da transformação da matriz apresentada na Tabela 2.5 em matriz \mathcal{D} é mostrado na Tabela 4.2. O recurso ocioso 4 é associado ao recurso fictício 1 nos períodos 1–4 e ao recurso fictício 14 no período 10.

Tabela 4.2: Exemplo de matriz \mathcal{D}

Período	Depósito								
	D1			D2			D3		
1	1	2		3	4		5	6	7
2	1	2					5	6	7
3	1	2	8	9			5	6	7
4	1	2	8	9			5	6	7
5		2	8	9					7
6		2	8	9	10	11			7
7			8	9	10	11			7
8			8	9	10	11	12	13	7
9			8	9	10	11	12	13	
10	14	15	8	9	10	11	12	13	

4.2 Heurísticas de Construção

Os algoritmos construtivos aqui apresentados consistem basicamente de métodos onde se mistura gulosidade e aleatoriedade. Dos seis algoritmos aqui discutidos, os quatro primeiros computam uma solução parcial onde atividades são associadas a espaços de trabalhos e os dois últimos métodos geram soluções parciais com alocação de recursos ociosos a depósitos. Com exceção das heurísticas apresentadas nas Seções 4.2.1 e 4.2.5, as demais são projetadas com base na estrutura da metaheurística GRASP a ser discutida mais adiante, para a geração da solução parcial de atividades.

4.2.1 Heurística Randomizada de Construção Adaptada (RCA)

O algoritmo RCA é uma adaptação da heurística construtiva randomizada proposta em [29] para o Problema de Células de Manufatura.

O método (Figura 4.1) é baseado na geração de *clusters*. O número de espaços de trabalho, um dos parâmetros do método ($|W|$), equivale ao número de *clusters* que devem ser formados. O outro parâmetro é naturalmente o número de atividades (J).

As atividades pertencentes a um *cluster* são as atividades que ao fim do procedimento são alocadas a um espaço de trabalho correspondente àquele *cluster*. Por este motivo, o algoritmo verifica antes de associar uma atividade a um *cluster* se, eventualmente, esta tem algum conflito de períodos com alguma atividade que já pertença àquele *cluster*.

Figura 4.1: Algoritmo RCA

```

Heurística RCA( $W, J$ )
1: Inicializar  $cluster_1, cluster_2$  até  $cluster_{|W|}$ ;
2: para  $i \leftarrow 1$  até  $J$  faça
3:   para  $j \leftarrow 1$  até  $J$  faça
4:     se  $i \neq j$  então
5:       Calcular similaridade entre atividades  $i$  e  $j$ ;
6:     fim se
7:   fim para
8: fim para
9: /*Associar uma atividade a cada um dos dois primeiros clusters */
10:  $Par\_Escolhido \leftarrow$  Escolher aleatoriamente 1 entre os 3 pares menos similares;
11:  $cluster_1 \leftarrow atividade_1 \in Par\_Escolhido$ ;
12:  $cluster_2 \leftarrow atividade_2 \in Par\_Escolhido$ ;
13: /*Associar uma atividade a cada um dos demais clusters */
14:  $k \leftarrow 3$ ;
15: enquanto  $k \leq |W|$  faça
16:   para  $j \leftarrow 1$  até  $J$  faça
17:     se  $j$  ainda não é associada a cluster então
18:       Calcular a similaridade entre  $j$  e as atividades pertencentes aos demais clusters;
19:     fim se
20:   fim para
21:    $cluster_k \leftarrow$  atividade de menor similaridade;
22: fim enquanto
23: /*Aloca as atividades que permanecem fora de clusters */
24: para  $j \leftarrow 1$  até  $J$  faça
25:   para  $k \leftarrow 1$  até  $|W|$  faça
26:     Calcular a similaridade entre  $j$  e  $cluster_k$  e verifica conflito de períodos;
27:   fim para
28:    $cluster_k \leftarrow$  atividade de maior similaridade sem conflito;
29: fim para
30: para  $j \leftarrow 1$  até  $|W|$  faça
31:    $W_j \leftarrow$  atividades  $\in cluster_j$ ;
32: fim para
33: Retornar associação de atividades;

end

```

O primeiro passo do algoritmo consiste em calcular e guardar os três pares menos similares de atividades. Um dentre eles é escolhido aleatoriamente para compor dois *clusters*. A medida de similaridade entre duas atividades i e j consiste do número de recursos que elas compartilham.

Em seguida, para a formação do terceiro *cluster* em diante, até que se atinja o número de espaços de trabalho, é calculada a medida de similaridade de cada atividade ainda não atribuída a um *cluster* com os *clusters* já formados. Como medida de similaridade, neste caso, considera-se o número de recursos utilizados em comum pela atividade em questão e os *clusters* já formados. A atividade menos similar é atribuída ao *cluster*. Em caso de empate, a atividade de menor rótulo é escolhida. O método passa então a fazer associações das atividades restantes aos *clusters* já formados. Para cada uma destas atividades, é calculada a medida de similaridade e a atividade é atribuída ao *cluster* com o qual possui maior similaridade

desde que não haja conflito de períodos com alguma atividade já pertencente ao *cluster*.

Por fim, as atividades associadas a cada *cluster* durante o algoritmo são alocadas aos espaços de trabalho correspondentes àquele *cluster*.

4.2.2 Heurística de Construção Randomizada FA1 (RFA1)

Esta heurística é baseada no algoritmo construtivo utilizado em [17], proposta a fim de que incorpore aleatoriedade.

Através de RFA1 uma solução parcial de atividades é construída da seguinte forma: Para cada atividade a , em ordem lexicográfica, é criada uma lista restrita de candidatos (*LRC*) contendo os espaços de trabalho disponíveis do período de início ao de término da atividade. Um espaço de trabalho é escolhido aleatoriamente a partir da *LRC*. A quantidade de elementos da *LRC* é definida pelo número de espaços disponíveis para a atividade. O pseudocódigo do algoritmo é mostrado na Figura 4.2.

Figura 4.2: Algoritmo RFA1

```

Heurística RFA1( $J, W$ )
1: para  $a \leftarrow 1$  até  $J$  faça
2:   Inicializar  $LRC(a)$ ;
3:    $inicio \leftarrow$  primeiro período de ocorrência de  $a$ ;
4:    $fim \leftarrow$  último período de ocorrência de  $a$ ;
5:   para cada  $w \in W$  faça
6:     se  $w$  é disponível em todos os períodos de  $inicio$  a  $fim$  então
7:       Incluir  $w$  em  $LRC(a)$ ;
8:     fim se
9:   fim para
10:  Escolher aleatoriamente um elemento em  $LRC$ ;
11:  Alocar a atividade  $a$  ao espaço de trabalho  $w$ ;
12:  Liberar  $LRC(a)$ ;
13: fim para
14: Retornar associação de atividades;

end

```

4.2.3 Heurística de Construção Randomizada FA2 (RFA2)

Difere de RFA1 na ordenação e tamanho da *LRC*. Nesta heurística, para cada atividade a são examinados os espaços de trabalho disponíveis. Uma *lista inicial de candidatos* (*LIC*) para cada atividade é criada, com os espaços de trabalho disponíveis ordenados do de melhor para o de pior localização em relação aos outros espaços de trabalho. O parâmetro de qualidade de uma localização aqui considerado é a distância para os demais espaços. Os primeiros $\alpha \times \min\{disp, |W|\}$, onde *disp* corresponde à quantidade de espaços de trabalho disponíveis, $|W|$, à cardinalidade

do conjunto de espaços de trabalhos e α é um parâmetro do algoritmo definido no intervalo $[0, 1]$, são escolhidos para compor a $LRC(a)$. Um espaço de trabalho é escolhido aleatoriamente a partir da LRC .

Figura 4.3: Algoritmo RFA2

```

Heurística RFA1( $J, W$ )
1: para  $a \leftarrow 1$  até  $J$  faça
2:   Inicializar  $LIC(a), LRC(a)$ ;
3:    $inicio \leftarrow$  primeiro período de ocorrência de  $a$ ;
4:    $fim \leftarrow$  último período de ocorrência de  $a$ ;
5:   para cada  $w \in W$  faça
6:     se  $w$  é disponível em todos os períodos de  $inicio$  a  $fim$  então
7:       Incluir  $w$  em  $LIC(a)$ ;
8:     fim se
9:     Ordenar  $LIC(a)$  da melhor para a pior localização;
10:     $LRC(a) \leftarrow$  os primeiros  $\alpha \times \min\{disp, |W|\}$  elementos em  $LIC(a)$ ;
11:  fim para
12:  Escolher aleatoriamente um elemento em  $LRC$ ;
13:  Alocar a atividade  $a$  ao espaço de trabalho  $w$ ;
14:  Liberar  $LRC(a)$ ;
15: fim para
16: Retornar associação de atividades;

end

```

4.2.4 Heurística de Construção de Atividades Adaptativa (CAA)

Esta heurística foi projetada para ser ainda mais adaptativa. A primeira associação de atividade a espaço de trabalho é feita aleatoriamente. A partir daí, a solução é construída elemento a elemento associando uma atividade a um espaço de trabalho de acordo com o custo da inserção na solução parcial. Para cada par (*atividade não associada, espaço de trabalho disponível*) é calculado o custo da inserção na solução parcial, formando uma LIC . Este custo é calculado utilizando a matriz \mathcal{L} parcialmente preenchida. Em seguida, LIC é ordenada em ordem não crescente de custo e os primeiros $\alpha \times \min\{disp, |W|\}$ são inseridos em uma LRC . Um elemento é aleatoriamente escolhido da LRC . O pseudocódigo do método é mostrado na Figura 4.4.

Figura 4.4: Algoritmo CAA

```

Heurística CAA( $J, W, \alpha, max\_tentativas$ )
1: Escolher uma atividade  $a \in J$  e um espaço de trabalho  $w \in W$  aleatoriamente;
2: Fazer a associação da atividade  $a$  ao espaço  $w$ ;
3: enquanto existir atividade não associada faça
4:   para cada atividade  $a_n \in J$  não associada faça
5:     para cada espaço de trabalho  $w_d \in W$  disponível faça
6:        $custo \leftarrow$  Calcular o custo de alocar  $a_n$  em  $w_d$ ;
7:       Incluir  $a_n, w_d, custo$  em  $LIC$ ;
8:     fim para
9:   fim para
10:  se  $|LIC| = 0$  então
11:     $tentativas \leftarrow tentativas + 1$ ;
12:    se  $tentativas > max\_tentativas$  então
13:      Chamar RFA2; /* Solução inviável */
14:    fim se
15:  se não
16:    Ordenar  $LIC$  por custo (ordem não crescente);
17:     $LRC \leftarrow$  os primeiros  $\alpha \times \min\{disp, |W|\}$  elementos de  $LIC$ ;
18:    Escolher aleatoriamente um elemento em  $LRC$  para associar a solução parcial;
19:  fim se
20: fim enquanto
21: Retornar associação de atividades;

end

```

É importante ressaltar que pelo fato da associação das atividades neste método não observar, necessariamente, a relação de prioridade entre estas, uma execução deste algoritmo pode gerar uma solução não viável, ou seja, uma solução em que nem todas as atividades tenham sido associadas a espaços de trabalhos. Neste caso, se após um número de tentativas de execução do método, nenhuma solução parcial viável for gerada, uma solução construída por meio de RFA2 é considerada.

4.2.5 Heurística RSP Adaptada (RSPA)

A partir de uma solução parcial de atividades, este procedimento associa recursos ociosos a depósitos. Consiste de uma adaptação da heurística RSP [18] por considerar a associação de recursos feita no período anterior. Desta forma, a partir do segundo período os recursos que permanecem ociosos de um período para o outro, são alocados com prioridade ao mesmo depósito do período anterior. Os demais recursos ociosos no período são alocados, como em RSP ao depósito mais próximo ao espaço de trabalho que este tiver sido associado mais recentemente. O pseudocódigo da heurística RSPA é apresentado na Figura 4.5.

Figura 4.5: Algoritmo RSPA

```

Heurística RSPA( $P, I_p, mA$ )
  No primeiro período, alocar cada recurso ocioso ao depósito mais próximo ao espaço de trabalho onde
  este é mais recentemente utilizado;
  para cada  $p \leftarrow 2$  até  $|P|$  faça
    Alocar os recursos ociosos que ocorrem em  $p - 1$ , ao mesmo depósito;
    Alocar o restante dos recursos ao depósito mais próximo ao espaço de trabalho onde este é mais
    recentemente associado;
  fim para
  Retornar associação de recursos ociosos;
end

```

4.2.6 Heurística de Construção de Ociosos Adaptativa (CAO)

A partir de uma solução parcial de atividades, este método (Figura 4.6) constrói uma solução parcial de recursos ociosos a partir da associação de um recurso ocioso a um depósito por vez, levando-se em consideração critérios de gulosidade e aleatoriedade. A cada período uma *LIC* contendo os recursos ociosos ainda não alocados e os depósitos que ainda possuem capacidade para armazenar recursos é ordenada do menor para maior custo de inserção na solução parcial é obtida. Os primeiros $\alpha \times \min(\text{disp}, |S| * C_s)$, onde *disp* é o número de depósitos disponíveis, $|S|$ é a cardinalidade do conjunto de depósitos e C_s é a capacidade do depósito s elementos da *LIC* são selecionados para compor a *LRC* e a partir desta um elemento é selecionado aleatoriamente. Além disso, são parâmetros de entrada do algoritmo: P , o conjunto de períodos; R o conjunto de recursos, I_p , o conjunto de recursos ociosos por período e α , um parâmetro do método.

Figura 4.6: Algoritmo CAO

```

Heurística CAO( $P, R, I_p, \alpha$ )
1: para cada  $p \leftarrow 1$  até  $P$  faça
2:   Inicializar LIC, LRC;
3:   para cada  $r \in I_p$  ( $I_p \subset R$ ) ainda não associado a depósito no período faça
4:     para cada  $s \in S$   $|s| < C_s$  faça
5:       custo  $\leftarrow$  Calcular o custo de alocar  $r$  em  $s$ ;
6:       Incluir  $r, s, \text{custo}$  em LIC;
7:     fim para
8:   fim para
9:   Ordenar LIC por custo (ordem não crescente);
10:  LRC  $\leftarrow$  os primeiros  $\alpha \times \min\{\text{disp}, |S| * C_s\}$  elementos de LIC;
11:  Escolher aleatoriamente um elemento em LRC para associar a solução parcial;
12: fim para
13: Retornar associação de atividades;
end

```

4.2.7 Algoritmos de Construção Propostos

Os algoritmos de construção relacionados a associação de atividades RCA, RFA1, RFA2 e CAA combinados, cada um, com os algoritmos para alocação de recursos ociosos RSP [18] e RSPA geram as seguintes heurísticas de construção, apresentados na Tabela 4.3 que ainda relaciona mais uma versão, formada pelo método CAA combinado com a heurística CAO.

Tabela 4.3: Algoritmos de construção propostos

Construção Proposta	Alocação de Atividades	Alocação de Ociosos
C1	RFA1	RSPA
C2	RFA1	RSP
C3	RFA2	RSPA
C4	RFA2	RSP
C5	RCA	RSPA
C6	RCA	RSP
C7	CAA	RSPA
C8	CAA	RSP
C9	CAA	CAO

4.3 Heurísticas de Refinamento

A busca local é uma fase de tentativa de melhoria da solução obtida na etapa de construção. Nesta fase, percorre-se a vizinhança da solução corrente buscando uma solução de melhor qualidade. Em um problema onde o objetivo é a minimização de uma função f , entende-se que uma solução t é melhor que a solução corrente s se $f(t) < f(s)$. Quando não existir uma melhor solução na vizinhança, a solução corrente é considerada um ótimo local.

Neste trabalho são implementadas duas versões clássicas da literatura de buscas locais que são descritas a seguir [30].

4.3.1 Busca Local *Best Improvement* ou Melhor Vizinho

O método consiste em, a partir de uma solução corrente s obtida pelo método de construção, percorrer toda a vizinhança da solução $V(s)$ obtendo um melhor vizinho que executado corresponde a solução corrente para a próxima iteração. A busca prossegue até que nenhum vizinho seja capaz de melhorar a solução corrente.

O pseudocódigo da busca pelo Melhor Vizinho é mostrado na Figura 4.7. Os parâmetros do algoritmo são: s , a solução inicial vinda da construção; seu custo, $f(s)$; e a vizinhança $V(s)$.

Figura 4.7: Pseudocódigo do algoritmo de busca pelo melhor vizinho

```

Heurística MV ( $s, f(s), V(s)$ )
1: enquanto  $\exists s' \in V(s)$  tal que  $s' = \arg \min\{s' | f(s') < f(s)\}$  faça
2:    $s \leftarrow s'$ ;
3:    $f(s) \leftarrow f(s')$ ;
4: fim enquanto
5: Retornar  $s$ ;
end

```

4.3.2 Busca Local *First Improvement* ou Primeiro de Melhora

Este método de busca local tem por característica ser não exaustivo. A busca, neste caso, aplica o movimento ao primeiro vizinho de melhora na solução corrente s , isto é, a primeira solução $s' \in N(s)$ para o qual $f(s') < f(s)$ é feita solução corrente da próxima iteração. A busca termina quando em uma iteração toda a vizinhança da solução corrente é explorada e nenhum vizinho de melhora é encontrado.

4.3.3 Algoritmos de busca local propostos

Na implementação dos algoritmos de refinamento, foram utilizados os movimentos descritos na Seção 3.2.2. De acordo com a escolha dos movimentos, foram definidas as seguintes vizinhanças:

- \mathcal{V}_1 : vizinhança explorada pelos movimentos \mathcal{M}_1 a \mathcal{M}_3 , isto é, são feitos todos os possíveis movimentos na solução parcial de atividades.
- \mathcal{V}_2 : vizinhança explorada pelos movimentos \mathcal{M}_4 e \mathcal{M}_5 , isto é, são feitos todos os possíveis movimentos individuais de recursos ociosos na solução parcial de recursos ociosos.
- \mathcal{V}_3 : vizinhança explorada pelos movimentos \mathcal{M}_6 a \mathcal{M}_8 , isto é, são feitos todos os possíveis movimentos de recursos ociosos que aparecem na solução parcial de recursos ociosos em períodos consecutivos.
- \mathcal{V}_4 : vizinhança explorada pela combinação das vizinhanças \mathcal{V}_1 e \mathcal{V}_2 . Isto é, a cada vizinho gerado em \mathcal{V}_1 , são investigados todos os vizinhos em \mathcal{V}_2 .

Desta maneira, com as vizinhanças disponíveis e as formas de implementação de busca local acima descritas, são propostas as seguintes buscas locais, dadas na Tabela 4.4.

Tabela 4.4: Algoritmos de busca local propostos

Busca Local Proposta	Estratégia de Busca	Estrutura de Vizinhança
MV1	Melhor Vizinho	\mathcal{V}_1 e \mathcal{V}_2
PM1	Primeiro Melhora	\mathcal{V}_1 e \mathcal{V}_2
MV2	Melhor Vizinho	\mathcal{V}_1 e \mathcal{V}_3
PM2	Primeiro Melhora	\mathcal{V}_1 e \mathcal{V}_3
MV4	Melhor Vizinho	\mathcal{V}_1
MV5	Melhor Vizinho	\mathcal{V}_2
MV6	Melhor Vizinho	\mathcal{V}_3

4.4 Análises Experimentais Preliminares

O objetivo aqui é de se fazer uma análise preliminar dos métodos propostos neste capítulo. Para isto, foram testadas todas as combinações de algoritmos apresentadas. Estes experimentos computacionais, e todos os outros apresentados neste trabalho, foram executados em um computador com processador Dual Intel Pentium 4, 3.192 GHz com 4Gb de memória RAM usando sistema operacional Ubuntu Linux 6.0.4, com exceção dos testes feitos com CPLEX, apresentados no Capítulo 7, que foram executados sobre o sistema operacional Windows 64 bits. O valor de α utilizado nos algoritmos C5, C9 é igual a 0. Para estas heurísticas são feitas 10 execuções para cada instância testada. Assim, todos os algoritmos têm sua execução baseada em gulosidade. Os valores calculados foram: os desvios percentuais médios dos algoritmos (*Desvio*), o número de melhores resultados obtidos (*#Best*) nas 96 instâncias e o tempo para obtenção dos melhores resultados (*Tempo*). O valor dado em *Desvio* é calculado como: $Desvio = (\sum_{i=1}^{96} (custo[i] - menor[i]) / menor[i] * 100) / 96$, onde $custo[i]$ é o custo obtido pelo algoritmo para instância i e $menor[i]$ é o menor custo para a instância i , alcançado pelos algoritmos comparados.

A Tabela 4.5 mostra o desempenho das heurísticas de construção sem que seja usado nenhum método de refinamento. Vê-se que os algoritmos C1, C5 e C7 têm os menores desvios da melhor solução e também os maiores números de melhores resultados, com destaque para os algoritmos C1 e C7. Observa-se que o algoritmo C3 não mantém o mesmo desempenho dos já citados para $\alpha = 0$. No entanto, C3 demonstra bons resultados quando são utilizados valores de $\alpha \neq 0$ juntamente com as heurísticas baseadas nas metaheurísticas GRASP, como pode ser visto nos Capítulos 5 e 6. O algoritmo C4 não foi capaz de encontrar nenhuma vez a melhor solução, assim como C9, que além disso, esteve na média a aproximadamente 334% da melhor solução.

Podemos atribuir o bom desempenho dos algoritmos C1, C5 e C7 em detrimento de C2, C6 e C8 à heurística utilizada para alocação dos recursos ociosos, como já dito anteriormente, a heurística RSPA mantém posicionados os recursos que apa-

recem em períodos consecutivos no mesmo depósito, evitando assim, movimentos desnecessários.

Tabela 4.5: Sumário dos resultados das construções

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	6,03	25,09	22,16	46,21	6,09	25,58	6,14	26,70	334,23
<i>#Best</i>	40	9	2	0	22	3	50	12	0
Tempo	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02

Nas combinações das construções e busca local, vê-se que MV1 e PM1 (Tabelas 4.6 e 4.7) têm desempenho similar sobre os algoritmos de construção. Em termos de tempo de computação, PM1 é ligeiramente mais rápida, com a ocorrência de casos em que o tempo de PM1 é em média 50% menor que MV1. A introdução das buscas faz com que o resultado médio de todos os algoritmos melhore. A combinação de C9 com MV1 e/ou PM1 tem redução média de custos de aproximadamente 5 vezes do custo encontrado pela construção.

Tabela 4.6: Sumário dos resultados das construções com busca MV1

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	3,47	14,06	16,14	30,16	4,96	14,07	5,84	15,49	65,52
<i>#Best</i>	39	19	3	0	24	11	37	12	0
Tempo	0,39	0,79	0,43	0,86	0,28	0,79	0,27	0,79	6,65

Tabela 4.7: Sumário dos resultados das construções com busca PM1

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	3,58	14,01	16,19	31,14	5,01	14,41	5,86	15,72	64,36
<i>#Best</i>	36	20	3	0	25	11	37	12	0
Tempo	0,19	0,51	0,19	0,58	0,16	0,55	0,17	0,54	6,72

Quando o mesmo teste é feito com as buscas MV2 e PM2 (Tabelas 4.8 e 4.9) vê-se que o tempo de processamento, em média, é mais alto. O aumento do tempo pode se dar pelo motivo das buscas serem capazes de fazer mais melhorias na solução, iterando mais vezes durante a execução dos métodos. Este fato justifica também que

os resultados sejam melhores em comparação as duas primeiras. Elas são capazes de reduzir os custos em todas as combinações com as construções. Isto pode ser visto mais claramente no caso de **C9+MV1** e **C9+PM1** no qual foi possível encontrar uma melhor solução para 15 e 6 das 96 instâncias, respectivamente, enquanto nos testes anteriores as combinações envolvendo **C9** não foi vista nenhuma melhoria. Os custos médios das soluções, em relação às buscas **MV2** e **PM2** também tiveram uma redução de aproximadamente 25% em relação à **MV1** e **PM1**.

Tabela 4.8: Sumário dos resultados das construções com busca **MV2**

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	2,65	11,45	14,85	26,56	4,99	11,66	5,79	14,00	25,56
<i>#Best</i>	48	21	4	0	23	11	36	11	15
Tempo	0,68	2,48	1,00	3,16	0,60	1,94	0,69	2,06	50,19

Tabela 4.9: Sumário dos resultados das construções com busca **PM2**

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	2,35	10,57	13,72	25,94	4,46	10,73	5,20	12,82	27,37
<i>#Best</i>	47	20	3	0	29	11	36	12	6
Tempo	0,33	0,90	0,36	1,01	0,38	0,76	0,38	1,19	8,85

A busca **MV4** aplicada aos algoritmos de construção mostrou-se leve, isto é, com baixo tempo de processamento e isto pode ser justificado pelo tamanho do espaço da busca. A vizinhança explorada nesta busca é menor do que todas as outras já testadas. Imagina-se que também por este motivo os resultados alcançados por ela (Tabela 4.10) não sejam melhores do que os já reportados, exceto para o caso onde é executada somente a construção.

Tabela 4.10: Sumário dos resultados das construções com busca **MV4**

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	3,56	25,16	16,37	46,09	4,87	25,84	5,51	28,30	311,21
<i>#Best</i>	41	10	3	0	26	4	42	11	0
Tempo	0,02	0,02	0,03	0,02	0,02	0,02	0,04	0,04	0,18

A Tabela 4.11 mostra que a busca MV5 é na média mais eficiente em tempo de computação do que MV1 e PM1 e mantém seus custos um pouco superiores aos destas. Uma característica interessante a ser observada neste caso é a combinação de C8 com MV5 onde o número de melhores soluções subiu para 17 contra a média de 11 mantidas com as outras buscas e o *gap* para a melhor solução só não foi melhor do que quando usada a busca MV2.

Tabela 4.11: Sumário dos resultados das construções com busca MV5

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	6,08	14,24	22,04	31,95	6,10	14,66	6,27	14,20	75,21
# <i>Best</i>	36	14	1	0	20	10	50	17	2
Tempo	0,07	0,71	0,08	0,73	0,07	0,74	0,08	0,72	6,67

Resultados interessantes podem ser vistos também na Tabela 4.12. A combinação dos algoritmos de construção com a busca MV6 mostra um desempenho médio superior ao das buscas MV1 e PM1 em termos de custo e principalmente quanto ao tempo de processamento. Particularmente avaliando a construção C9 é possível chegar 7 vezes ao melhor resultado com um desvio percentual médio aproximadamente 50% abaixo dos resultados alcançados pelas buscas MV1 e PM1.

Tabela 4.12: Sumário dos resultados das construções com busca MV6

Algoritmo	C1	C2	C3	C4	C5	C6	C7	C8	C9
Desvio(%)	5,05	10,51	20,90	27,03	5,91	11,42	5,70	11,23	31,65
# <i>Best</i>	33	18	2	0	27	10	49	17	7
Tempo	0,13	0,82	0,13	1,03	0,11	0,75	0,12	0,77	22,68

Pode-se nesse momento inferir a respeito de algumas características da busca baseando-se na descrição das mesmas e nos resultados apresentados.

- As buscas MV4 e MV5 podem ser consideradas subconjuntos das buscas MV1. Enquanto em MV1 são feitas (1) todas as tentativas de trocas e realocação de atividades e (2) todas as tentativas de trocas e realocação de recursos ociosos, em MV4 é feito somente (1) e em MV5 é feito somente (2). Em vista disto, os tempos de computação de MV4 e MV5 tendem a serem menores que os de MV1 e que, como já mostrado, por ser mais completa MV1 pode levar a custos mais baixos.

- A busca MV6 pode ser considerada subconjunto da busca MV2, uma vez que em MV2 são feitas (1) todas as tentativas de trocas e realocação de atividades e (2) todas as tentativas de trocas e realocação de recursos ociosos em blocos. Já em MV6 é feita somente a segunda parte de MV2. Desta forma, como dito a respeito de MV4 e MV5, seu tempo de processamento tende a ser menor, mas observa-se, que apesar disto, as médias dos custos obtidos são bastante próximas, obtendo na combinação com algumas construções custos mais altos e em outras, custos mais baixos que obtidos por MV2.

As buscas locais, de maneira geral, se mostraram eficientes para o refinamento das soluções obtidas pelos algoritmos de construção, como pôde ser mostrado nos resultados acima. Ainda sim, no próximo Capítulo são apresentados algoritmos heurísticos baseados nas metaheurísticas VNS, ILS, GRASP e Busca Tabu e alguns métodos híbridos derivados destas, com a proposta de melhorar os resultados aqui obtidos mostrando que as metaheurísticas são mais hábeis a escapar de ótimos locais que as buscas locais.

Capítulo 5

Metaheurísticas Propostas para o PADE

Neste capítulo são propostos e preliminarmente avaliados algoritmos para o PADE baseados nas metaheurísticas VNS (*Variable Neighborhood Search*), ILS (*Iterated Local Search*), GRASP (*Greedy Randomized Adaptive Procedure*) e sua variação, GRASP Reativo e Busca Tabu.

5.1 A metaheurística *Variable Neighborhood Search*

Variable Neighborhood Search (VNS) é uma metaheurística proposta por Hansen e Mladenović [31]. Trata-se de um método que se utiliza da exploração de mais de uma vizinhança. Ele pode combinar mudanças estocásticas e determinísticas de vizinhanças [32]. Um pseudocódigo simples, para o caso determinístico, aparece na Figura 5.1. Inicialmente, um conjunto de estruturas de vizinhança é definido. Estas vizinhanças podem ser arbitrariamente escolhidas, mas frequentemente uma sequência $|\mathcal{N}_1| < |\mathcal{N}_2| < \dots < |\mathcal{N}_{kmax}|$ com cardinalidade crescente é definida [22]. Uma solução inicial é gerada e o índice das vizinhanças é inicializado. O *loop* interno (linhas 5–14) gera uma solução s' vizinha da solução corrente s escolhida aleatoriamente na vizinhança $\mathcal{N}_k(s)$. Em seguida, é executada uma busca local sobre s' . Se uma solução s'' melhor que a corrente é encontrada, s'' passa a ser a solução corrente e o procedimento é reiniciado com $k = 1$. Se nenhuma melhora é encontrada k é incrementado e uma nova busca recomeça. O *loop* é finalizado quando não há solução de melhora em algum vizinho $\mathcal{N}_k(s)$. O *loop* mais externo, iniciado na linha 3, faz com que sejam feitas tentativas até que um critério de parada estabelecido seja alcançado.

Figura 5.1: Pseudocódigo da metaheurística VNS

```

Heurística VNS()
1: Selecione o conjunto de estruturas de vizinhanças  $\mathcal{N}_k$ , com  $k = 1, \dots, k_{max}$ ;
2:  $s \leftarrow GeraSolucaoInicial()$ ;
3: enquanto critério de parada não satisfeito faça
4:    $k \leftarrow 1$ ;
5:   enquanto  $k \leq k_{max}$  faça
6:      $s' \leftarrow$  Vizinho qualquer em  $\mathcal{N}_k(s)$ 
7:      $s'' \leftarrow BuscaLocal(s')$ ;
8:     se  $f(s'') < f(s)$  então
9:        $s \leftarrow s''$ ;
10:     $k \leftarrow 1$ ;
11:    se não
12:       $k \leftarrow k + 1$ ;
13:    fim se
14:  fim enquanto
15: fim enquanto
end

```

5.1.1 VNS proposto para o PADE

Neste trabalho são utilizadas as estruturas de vizinhanças definidas na Seção 4.3.3. Para cada valor de $k = \{1, 2, 3\}$ uma solução s' , vizinha da solução corrente s , é gerada aleatoriamente da seguinte maneira:

- $k = 1$: É escolhido aleatoriamente um tipo de movimento: de atividades ou de recursos ociosos. Caso o movimento escolhido seja de atividades, é gerado um vizinho na vizinhança \mathcal{V}_1 . Se, ao contrário, o movimento escolhido for de recursos ociosos, é gerado um vizinho na vizinhança \mathcal{V}_2 .
- $k = 2$: Gera um vizinho aleatório na vizinhança \mathcal{V}_4 .
- $k = 3$: É escolhido aleatoriamente um tipo de movimento: de atividades ou de recursos ociosos. Caso o movimento escolhido seja de atividades, é gerado um vizinho na vizinhança \mathcal{V}_1 . Se, ao contrário, o movimento escolhido for de recursos ociosos, é gerado um vizinho na vizinhança \mathcal{V}_3 .

Além disso, no passo da busca local, são feitas buscas de três tipos:

- MV4: explora toda vizinhança da solução corrente em \mathcal{V}_1 , fazendo com que o melhor movimento seja executado e a solução gerada seja a solução corrente para próxima iteração. A busca prossegue até que nenhum melhor vizinho seja encontrado.
- MV5: investiga toda vizinhança da solução corrente em \mathcal{V}_2 . O melhor movimento é executado e a solução gerada é a solução corrente para próxima iteração. A busca prossegue até que nenhum melhor vizinho seja encontrado.

- **MV6**: explora toda vizinhança da solução corrente na vizinhança \mathcal{V}_3 . O melhor movimento é executado e a solução gerada é a solução corrente para próxima iteração. A busca prossegue até que nenhum melhor vizinho seja encontrado.

Em nossa implementação, a solução inicial é gerada através dos procedimentos **C1**, **C3**, **C5**, **C7** e **C9**, procurando gerar boas soluções iniciais. Considera-se o valor de $\alpha = 0$ para **C3**, **C7** e **C9**. A partir dos algoritmos de construção são propostas 5 versões independentes de VNS apresentadas na Tabela 5.1.

Tabela 5.1: Algoritmos VNS Propostos

VNS	Construção
VNS1	C1
VNS2	C3
VNS3	C5
VNS4	C7
VNS5	C9

O critério de parada utilizado é o do número de iterações sem melhora. A escolha adequada deste número para o método foi crucial na qualidade das soluções. Neste sentido, são testados os seguintes valores: 100, 500 e 700, 1000, 1200 e 1500.

O Apêndice A apresenta os resultados obtidos pelas versões de VNS em função da variação do número de iterações sem melhora utilizado.

Apresentamos nas Tabelas 5.2 e 5.3 um sumário dos resultados alcançados nas quais são mostrados o desvio percentual médio da melhor solução (*Desvio*), o número de melhores soluções obtidas dentro do conjunto das 96 instâncias (*#Best*) e o tempo médio de processamento (*Tempo*) do algoritmo.

Tabela 5.2: Sumário dos resultados do algoritmos VNS para 500 e 700 iterações

Alg.	500					700				
	VNS1	VNS2	VNS3	VNS4	VNS5	VNS1	VNS2	VNS3	VNS4	VNS5
Desvio (%)	2,34	3,10	2,72	2,67	14,29	1,96	2,42	2,22	2,50	13,27
<i>#Best</i>	43	36	41	43	13	47	40	41	43	12
Tempo	272,6	133,8	116,8	116,3	112,8	152,1	168,6	141,0	151,3	105,4

Tabela 5.3: Sumário dos resultados do algoritmo VNS1 para 1000, 1200 e 1500 iterações

	1000	1200	1500
Desvio(%)	1,46	1,40	1,29
<i>#Best</i>	50	50	49
Tempo	184,6	177,4	319,2

A primeira linha da tabela mostra o número de iterações com as quais o algoritmo VNS foi executado. Foram reservados desta comparação os resultados de execuções com 100 iterações já que não atingiram valores promissores. As tabelas mostram que, com o aumento do número de iterações pode-se obter valores menores de desvio. Pode-se observar também que a versão VNS1 se mostra mais promissora que as demais até 700 iterações. Por este motivo, os testes com 1000, 1200 e 1500 iterações, são realizados somente com esta versão. O tempo de processamento cresce com o número de iterações, como é de se esperar, e por isso, os testes foram encerrados com 1500 iterações.

As Tabelas 5.4 e 5.5 apresentam os resultados comparados com os melhores custos conhecidos para o conjunto de instâncias. São dados o desvio percentual médio dos algoritmos propostos quando comparados ao melhor custo (*Desvio*) e o número de melhores soluções encontrado por cada algoritmo no conjunto de 96 instâncias (*#Best*).

Tabela 5.4: Melhorias de custo obtidas com os algoritmos VNS

Alg.	500					700				
	VNS1	VNS2	VNS3	VNS4	VNS5	VNS1	VNS2	VNS3	VNS4	VNS5
Desvio (%)	0,71	1,40	1,07	1,00	10,26	0,35	0,79	0,61	0,84	9,45
<i>#Best</i>	56	48	56	55	18	61	55	58	59	21

Tabela 5.5: Melhorias de custo obtidas com o algoritmo VNS1

	1000	1200	1500
Desvio(%)	-0,12	-0,18	-0,28
<i>#Best</i>	65	66	66

Neste aspecto, pode ser visto que, ao atingir o limite de 1500 iterações, chega-se a 0,03% do melhor custo conhecido. Apesar de não parecer um resultado muito promissor, acredita-se que deve-se considerar VNS1 nas comparações posteriores. Espera-se, ainda, que versões híbridas utilizando a metaheurística VNS sejam propostas mais interessantes.

5.2 A metaheurística *Iterated Local Search*

A metaheurística *Iterated Local Search* (ILS) é um procedimento iterativo no qual buscas locais são aplicadas a novas soluções de partida, obtidas através de perturbações em soluções de ótimos locais.

O método funciona da seguinte maneira [33]: Dada uma solução corrente s^* , uma perturbação é aplicada levando a uma solução s' . Em seguida, s' é explorada através de um procedimento de busca local resultando na solução $s^{*'}$, que corresponde a um ótimo local. Um critério de aceitação é utilizado para decidir qual das duas soluções, entre s^* e $s^{*'}$ tornar-se-á corrente na próxima iteração. A Figura 5.2 apresenta o pseudocódigo do algoritmo ILS.

Figura 5.2: Pseudocódigo da metaheurística ILS

```

Heurística ILS()
1:  $s_0 \leftarrow GeraSolucaoInicial();$ 
2:  $s^* \leftarrow BuscaLocal(s_0);$ 
3: repita
4:    $s' \leftarrow Perturbacao(s^*, historia);$ 
5:    $s^{*'} \leftarrow BuscaLocal(s');$ 
6:    $s^* \leftarrow CriterioAceitacao(s^*, s^{*'}, historia);$ 
7: até Critério de parada satisfeito

end

```

Existem quatro componentes fundamentais do método a serem considerados, representados no pseudocódigo por meio das funções:

- *GeraSolucaoInicial* (linha 1): retorna uma solução inicial que pode ser gerada aleatoriamente ou através de algum procedimento guloso;
- *BuscaLocal* (linha 2): a partir de uma dada solução inicial, retorna um ótimo local;
- *Perturbacao* (linha 3): perturba uma solução através de um ou mais movimentos;
- *CriterioAceitacao* (linha 4): decide qual solução deverá ser considerada na próxima iteração do algoritmo.

Para o bom funcionamento do método, é necessário que o nível da perturbação seja cuidadosamente definido. Uma perturbação não deve ser demasiadamente pequena para que o procedimento não retorne frequentemente à solução corrente e poucas novas soluções sejam exploradas, assim como, não deve ser grande demais, pois neste caso o método recairia em um algoritmo de múltiplos reinícios aleatórios.

O mecanismo de perturbação, junto com o de busca local, define a possível transição entre a solução corrente s^* para a solução vizinha $s^{*'}$. O critério de aceitação determina se $s^{*'}$ é aceita ou não como a nova solução corrente. Pode servir como um procedimento de intensificação se for definido de forma que somente melhores soluções sejam aceitas, isto é, para um problema onde o objetivo é a minimização tem-se que *CriterioAceitacao* retorna $s^{*'}$ se $f(s^{*'}) < f(s^*)$ ou s^* em caso contrário.

5.2.1 ILS proposto para o PADE

Neste trabalho, as quatro componentes do ILS foram definidas da seguinte maneira:

Construção da solução inicial: Uma solução é construída através dos procedimentos C1, C3, C5, C7 e C9, descritos em 4.2.

Mecanismo de perturbação: Foram utilizados seis níveis de perturbação definidos a seguir. Cada tipo de perturbação é exemplificado considerando como solução corrente a associação de atividades e espaços de trabalho dada nas Tabelas 2.4 e 2.5.

- No primeiro nível dois diferentes espaços de trabalho escolhidos aleatoriamente têm seus elementos trocados período a período. No exemplo mostrado na Tabela 5.6 são permutados os elementos dos espaços de trabalho *ES1* e *ES2*.

Tabela 5.6: Exemplo da perturbação de nível 1

Período	Espaço de trabalho		
	<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
1		A1 (6,7)	
2	A2 (1,5)	A1 (6,7)	
3	A2 (1,5)		
4	A2 (1,5)		
5	A2 (1,5)	A3 (4)	A4 (2,8)
6		A3 (4)	A4 (2,8)
7		A5 (3,4)	A4 (2,8)
8		A5 (3,4)	
9		A5 (3,4)	A6 (9)
10			A6 (9)

- No nível 2 são escolhidos aleatoriamente duas atividades em diferentes espaços de trabalho, e estas são trocadas por t vezes. A Tabela 5.7 mostra um exemplo onde a perturbação é executada, considerando as atividades A2 e A4, os espaços de trabalho *ES2* e *ES3* e $t = 1$.

Tabela 5.7: Exemplo da perturbação de nível 2

Período	Espaço de trabalho		
	<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
1	A1 (6,7)		
2	A1 (6,7)		A2 (1,5)
3			A2 (1,5)
4			A2 (1,5)
5	A3 (4)	A4 (2,8)	A2 (1,5)
6	A3 (4)	A4 (2,8)	
7	A5 (3,4)	A4 (2,8)	
8	A5 (3,4)		
9	A5 (3,4)		A6 (9)
10			A6 (9)

Deve-se notar que uma nova tentativa de perturbação deveria ser feita se por exemplo tivessem sido selecionadas as atividades A2 e A3 já que seria gerada uma solução inviável, uma vez que a atividade A1 seria realizada em diferentes espaços de trabalho em períodos consecutivos, conforme mostra a Tabela 5.8.

Tabela 5.8: Exemplo da perturbação de nível 2 com inviabilidade

Período	Espaço de trabalho		
	<i>ES1</i>	<i>ES2</i>	<i>ES3</i>
1	A1 (6,7)		
2	A2 (1,5)	A1 (6,7)	
3	A2 (1,5)		
4	A2 (1,5)		
5	A2 (1,5)	A3 (4)	A4 (2,8)
6		A3 (4)	A4 (2,8)
7	A5 (3,4)		A4 (2,8)
8	A5 (3,4)		
9	A5 (3,4)		A6 (9)
10			A6 (9)

- A perturbação de nível 3 consiste em, por t vezes, trocar todos os elementos de dois pares diferentes de espaços de armazenamento em um período, escolhidos aleatoriamente. Pelo menos um dos depósitos deve ser não vazio. A Tabela 5.9 demonstra o resultado desta perturbação com $t = 2$, aplicada primeiramente no período 8, depósitos $D2$ e $D3$ e, em sequência, no período 10, depósitos $D1$ e $D2$.

Tabela 5.9: Exemplo da perturbação de nível 3

Período	Depósito								
	D1		D2		D3				
1	4	3	1	5	2	8	9		
2	4	3			2	8	9		
3	4	3	6	7	2	8	9		
4	4	3	6	7	2	8	9		
5		3	6	7			9		
6		3	6	7	1	5	9		
7			6	7	1	5	9		
8			6	2	8	9	7	1	5
9			6	7	1	5	2	8	0
10	7	1	5	4	3	6	2	8	0

- Uma perturbação de nível 4 executa uma perturbação de nível 2 e em seguida uma perturbação de nível 3 por t vezes.
- No nível 5 são escolhidos aleatoriamente um período e dois diferentes depósitos e um dos recursos dos respectivos depósitos é trocado com um recurso do outro depósito. O procedimento é repetido t vezes. Quando há espaço em um dos depósitos pode ocorrer um movimento de realocação. Na Tabela 5.10 ocorre a troca de dois recursos dos depósitos $D1$ e $D2$ no período 1 e no período 2, a realocação do recurso 2 do depósito $D3$ para o $D2$, para $t = 2$.

Tabela 5.10: Exemplo da perturbação de nível 5

Período	Depósito								
	D1		D2		D3				
1	1	3	4	5	2	8	9		
2	4	3	2			8	9		
3	4	3	6	7	2	8	9		
4	4	3	6	7	2	8	9		
5		3	6	7			9		
6		3	6	7	1	5	9		
7			6	7	1	5	9		
8			6	7	1	5	2	8	9
9			6	7	1	5	2	8	0
10	4	3	6	7	1	5	2	8	0

- Uma perturbação de nível 6 escolhe dois recursos em diferentes depósitos e faz a troca destes em todos os períodos consecutivos nos quais ocorram. O processo é repetido por t vezes. Uma nova tentativa é feita caso a troca dos dois recursos faça com que um (ou mais) dos recursos ociosos fiquem em depósitos diferentes nos períodos consecutivos nos quais ocorram. A Tabela 5.11 mostra a troca dos recursos 4 e 9. Nota-se que seria impossível, no entanto, realizar a perturbação se os recursos escolhidos fossem 1 e 4.

Tabela 5.11: Exemplo da perturbação de nível 6

Período	Depósito								
	D1		D2		D3				
1	9	3	1	5	2	8	4		
2	9	3			2	8	4		
3	9	3	6	7	2	8	4		
4	9	3	6	7	2	8	4		
5	9	3	6	7					
6	9	3	6	7	1	5			
7	9		6	7	1	5			
8	9		6	7	1	5	2	8	
9			6	7	1	5	2	8	
10		3	6	7	1	5	2	8	4

Algoritmo de busca local: Uma solução é refinada através dos procedimentos MV2 e PM2 descritos em 4.3.

Critério de aceitação: De acordo com o critério de aceitação adotado, uma solução s'^* passa a ser corrente caso melhore o valor da função objetivo da solução s_0 , ou seja, se $f(s'^*) < f(s_0)$.

O pseudocódigo do ILS proposto é apresentado na Figura 5.3. Cada nível de perturbação é aplicado a uma mesma solução um número de vezes, limitado pelo parâmetro $vezes_{max}$, conforme pode ser visto no *loop* mais interno. O processo é repetido até que um número de iterações sem melhora seja alcançado, dado por $iter_{max}$. A cada vez que uma solução passa por todos os níveis de perturbação, t , o número de trocas em uma perturbação é aumentado em uma unidade.

Figura 5.3: Pseudocódigo do algoritmo ILS Proposto

```

Heurística ILS( $iter_{max}$ ,  $vezes_{max}$ )
1:  $s_0 \leftarrow GeraSolucaoInicial()$ ;
2:  $s_0 \leftarrow BuscaLocal(s_0)$ ;
3:  $iter \leftarrow 0$ ;  $nivel \leftarrow 1$ ;  $t \leftarrow 2$ ;
4: enquanto  $iter < iter_{max}$  faça
5:    $vezes \leftarrow 0$ ;
6:   enquanto  $vezes < vezes_{max}$  faça
7:      $s' \leftarrow Perturbacao(s_0, nivel, t)$ ;
8:      $s'^* \leftarrow BuscaLocal(s')$ ;
9:     se  $f(s'^*) < f(s_0)$  então
10:       $f(s_0) \leftarrow f(s'^*)$ ;
11:       $iter \leftarrow 0$ ;
12:       $nivel \leftarrow 1$ ;
13:       $vezes \leftarrow 0$ ;
14:     se não
15:        $vezes \leftarrow vezes + 1$ ;
16:        $iter \leftarrow iter + 1$ ;
17:     fim se
18:   fim enquanto
19:    $nivel \leftarrow nivel + 1$ ;
20:   se  $nivel > 6$  então
21:      $nivel \leftarrow 1$ ;
22:      $t \leftarrow t + 1$ ;
23:   fim se
24: fim enquanto

end

```

A partir da combinação de algoritmos de construção e busca local são propostas 10 versões independentes de ILS, apresentadas na Tabela 5.12.

Tabela 5.12: Algoritmos ILS Propostos

ILS	Construção	Busca Local	ILS	Construção	Busca Local
ILS1	C1	MV2	ILS6	C1	PM2
ILS2	C3	MV2	ILS7	C3	PM2
ILS3	C5	MV2	ILS8	C5	PM2
ILS4	C7	MV2	ILS9	C7	PM2
ILS5	C9	MV2	ILS10	C9	PM2

Como já observado na metaheurística VNS, a escolha adequada do número de iterações do método influencia a qualidade das soluções. Nesta implementação de ILS há ainda uma particularidade uma vez que é necessário definir o número de vezes que um nível de perturbação é aplicado a uma solução ($vezes_{max}$). Vale ressaltar que a combinação destes dois fatores está diretamente ligada ao tempo de execução do algoritmo. Para definir estes dois parâmetros, testes foram feitos até que se chegou a um número de iterações sem melhora ($iter_{max}$) igual a 700 e o número de vezes de cada perturbação ($vezes_{max}$) igual a 20.

O Apêndice B apresenta os resultados obtidos pelas versões de ILS com os parâmetros definidos.

A Tabela 5.13 apresenta um sumário dos resultados alcançados. Dentre os algoritmos ILS propostos, as versões ILS1, ILS3 e ILS6 têm menor desvio percentual médio e maior número de melhores solução. Além disso, pode-se observar uma grande diferença de tempo de processamento das versões ILS de ILS1 a ILS5, que utilizam o método de busca local MV, quando comparadas as demais.

Tabela 5.13: Sumário dos resultados do algoritmos ILS

Alg.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
Desvio(%)	1,5	2,1	1,5	2,0	5,1	1,5	2,2	1,8	2,0	4,8
#Best	50	44	54	46	32	48	40	44	42	26
Tempo	794,2	911,9	772,4	829,6	1027,8	155,0	231,2	238,9	223,1	406,8

Quando os algoritmos ILS propostos são confrontados com o melhor custo conhecido, os resultados apresentados na Tabela 5.14 confirmam um melhor desempenho da versão ILS6 diante das outras. Esta versão, além de reduzir os melhores custos conhecidos em 0,57% e ter obtido 71 soluções com custo igual ou menor que o da literatura, tem o menor tempo de processamento médio.

Tabela 5.14: Comparação dos algoritmos ILS com literatura

Alg.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
Desvio(%)	-0,73	-0,10	-0,72	-0,28	2,75	-0,74	-0,01	-0,46	-0,28	2,46
#Best	70	62	73	66	48	73	59	71	67	46

Uma outra tentativa de implementação da metaheurística ILS foi feita, na qual cada perturbação, escolhida aleatoriamente, é executada uma única vez. A tentativa é justificada pela diminuição de parâmetros no método. No entanto, os resultados computacionais obtidos em alguns testes mostram que o desempenho é menos promissor que o da proposta anterior. Como exemplo disto, apresenta-se na Tabela 5.15 um sumário dos resultados obtidos em comparação ao melhor custo conhecido, respectivamente, com o uso de 700, 1000, 1500 e 2000 iterações sem melhora do método, com solução inicial a partir da heurística de construção C1 e busca local PM2, que deve ser comparado com a implementação do algoritmo ILS1. Percebe-se que ao atingir 2000 iterações, os resultados alcançados são ainda inferiores se comparados com a proposta anterior e além disso, com este número de iterações, o tempo de processamento já supera a versão ILS6.

Tabela 5.15: Melhoria de custos obtidos com implementação ILS com perturbação aleatória com 700, 1000, 1500 e 2000 iterações

	700	1000	1500	2000
Desvio(%)	0,16	0,02	-0,23	-0,28
#Best	60	61	63	64
Tempo	152,7	155,7	222,4	357,0

5.3 A Metaheurística GRASP

A metaheurística GRASP foi proposta em 1995 por Feo e Resende [34] e, desde então, tem mostrado bons resultados para problemas de Otimização Combinatória [35, 36, 37, 38, 39]. GRASP é um método iterativo onde cada iteração é composta de duas etapas: construção de uma solução e busca local. As iterações do GRASP são totalmente independentes, e portanto ela é uma heurística do tipo *multistart*.

A fase de construção do GRASP é iterativa, gulosa, randômica e adaptativa. Ela é iterativa porque constrói uma solução elemento a elemento e é adaptativa pois a escolha do próximo elemento da solução parcial é influenciada pelas escolhas anteriores. Para a seleção do próximo elemento da solução parcial, a princípio todos os candidatos são considerados, contudo o número de candidatos pode em muitos casos ser extremamente elevado, por isso, normalmente, é considerada apenas uma lista restrita de candidatos (*LRC*). O tamanho da *LRC* é $p = 1 + \alpha(a - 1)$, onde a é o número total de elementos candidatos e α um parâmetro de entrada. A partir de uma *LRC*, seleciona-se um elemento deste conjunto aleatoriamente e não necessariamente o melhor. Esta escolha aleatória permite que este procedimento possa ser usado várias vezes para obter soluções distintas.

Pela forma como são construídas as soluções na etapa inicial, estas não representam na maioria dos casos um ótimo local, daí ser habitualmente usada uma etapa de busca local. O critério de parada do GRASP é normalmente o número máximo de iterações e a solução final GRASP é a melhor solução obtida ao final de sua execução. O pseudocódigo da metaheurística é apresentado na Figura 5.4.

Figura 5.4: Pseudocódigo da metaheurística GRASP

```

Metaheurística GRASP(MAX_ITER,  $\alpha$ )
1: para  $k \leftarrow 1$  até MAX_ITER faça
2:   Solucao  $\leftarrow$  Constroir_Solucao( $\alpha$ );
3:   Solucao  $\leftarrow$  Busca_Local(Solucao);
4:   Atualizar_Solucao(Solucao, Melhor_Solucao);
5: fim para
6: retornar (Melhor_Solucao);

end

```

5.3.1 Heurísticas GRASP propostas para o PADE

A partir das heurísticas propostas no Capítulo 4, são selecionados os algoritmos de construção C3, C4, C7, C8 e C9 por serem heurísticas formuladas de acordo com os requisitos da fase de construção do GRASP e as versões de busca local MV2 e PM2 por mostrarem melhores resultados médios nos testes apresentados na Seção 4.4 e alguns testes preliminares. Os algoritmos GRASP propostos são apresentados na Tabela 5.16.

Tabela 5.16: Algoritmos GRASP Propostos

GRASP	Construção	Busca Local	GRASP	Construção	Busca Local
G1	C3	MV2	G6	C3	PM2
G2	C4	MV2	G7	C4	PM2
G3	C7	MV2	G8	C7	PM2
G4	C8	MV2	G9	C8	PM2
G5	C9	MV2	G10	C9	PM2

Como pode ser visto, um longo estudo deve ser feito para determinar quais destas versões de algoritmos GRASP são mais eficientes, já que todas diferem entre si de alguma maneira. Além disso é necessário determinar o único parâmetro do método, o valor de α a ser usado.

Resende [40] mostrou através de experimentos que, além da influência direta do valor do parâmetro α para o bom desempenho do GRASP em termos de custo de solução, este parâmetro é responsável também pela variação do tempo de processamento.

Por todos estes motivos apresentados, fica claro que a escolha apropriada do valor do parâmetro α é crítica e relevante para a obtenção de um bom balanço entre qualidade da solução e tempo de processamento [40].

No Anexo C são apresentados os resultados obtidos pelos algoritmos GRASP propostos em um conjunto de 6 valores de α , a saber: $\alpha = 0,0, 0,2, 0,4, 0,6, 0,8$ e $1,0$. Nestes testes, o número de iterações GRASP utilizado é de 100 e cada combinação de algoritmo-parâmetro GRASP foi executada 10 vezes.

As Tabelas 5.17 a 5.21 mostram um sumário destes resultados para cada valor de α .

Tabela 5.17: Sumário dos resultados da variação de α para G1 e G2

Alg.	G1						G2					
	α	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8
Desvio	7,3	3,4	1,5	1,1	0,9	0,9	7,3	3,3	1,4	1,1	0,9	0,9
# Best	17	22	40	53	59	63	17	22	40	53	59	63
Tempo	141,1	209,8	87,6	85,1	81,9	195,0	90,7	121,8	87,8	84,5	81,5	161,9

Tabela 5.18: Sumário dos resultados da variação de α para G3 e G4

Alg.	G3						G4					
α	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
Desvio	3,2	1,7	1,0	0,9	0,8	0,9	3,2	1,7	1,0	0,9	0,8	0,9
# <i>Best</i>	34	39	58	58	62	63	34	39	58	58	62	63
Tempo	74,6	76,8	77,6	79,2	79,2	79,9	74,7	76,2	77,6	79,8	80,2	80,3

Tabela 5.19: Sumário dos resultados da variação de α para G5 e G6

Alg.	G5						G6					
α	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
Desvio	3,2	1,6	0,9	0,8	0,8	0,9	31,7	23,0	15,9	12,7	10,0	8,9
# <i>Best</i>	34	39	59	62	58	59	3	3	4	7	31	31
Tempo	84,2	85,9	87,7	90,1	89,9	90,4	28,7	26,3	26,0	24,3	22,6	22,0

Tabela 5.20: Sumário dos resultados da variação de α para G7 e G8

Alg.	G7						G8					
α	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
Desvio	45,7	35,5	26,9	22,7	17,9	16,7	15,7	12,5	10,6	9,4	9,0	8,6
# <i>Best</i>	0	0	0	1	20	25	16	17	21	27	29	32
Tempo	89,8	82,6	78,7	72,8	67,6	66,2	22,8	24,2	24,4	24,1	23,7	23,5

Tabela 5.21: Sumário dos resultados da variação de α para G9 e G10

Alg.	G9						G10					
α	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
Desvio	24,0	19,3	17,8	16,4	16,6	16,3	32,6	27,7	29,9	32,6	34,0	34,2
# <i>Best</i>	11	12	14	21	22	24	1	8	11	13	15	12
Tempo	68,6	72,1	69,6	69,9	68,8	68,4	40,3	132,4	256,8	394,0	512,7	634,4

Pode-se resumir os resultados alcançados da seguinte maneira:

- a busca local MV2, quando utilizada na heurística GRASP, é capaz de escapar de ótimos locais mais facilmente que PM2, em vista disso, os algoritmos G1, G2, G3, G4 e G5 que a utilizam tiveram resultados melhores do que os demais (versões G6–G9).

- a influência do parâmetro α sobre **G1**, **G2**, **G3**, **G4** e **G5** segue em geral a mesma tendência: melhores custos são alcançados quando o valor de α é variado de 0,0 até 1,0. Pode-se inferir com base neste resultado que quanto mais gulosa a solução maior é a tendência de que a busca fique presa a um ótimo local, isto talvez justifique também o ligeiro aumento de tempo de processamento no sentido contrário, isto é, quando se está mais próximo da aleatoriedade. Há, no entanto, uma exceção em **G5** que tem $\alpha=0,6$ obtendo os melhores resultados. Este valor de α poderia representar um equilíbrio entre os fatores gulosidade e aleatoriedade.
- em se tratando de tempo de processamento, não contrariamos o que dissemos acima ao afirmarmos que, com poucas exceções, os algoritmos mantêm o tempo médio similar. Difere deste comportamento o caso **G10** que tem em média tempos de computação mais altos.

A Tabela 5.22 mostra o desvio percentual médio dos algoritmos GRASP propostos quando comparados ao melhor custo conhecido (*Desvio*) e o número de melhores soluções encontrado por cada algoritmo no conjunto de 96 instâncias. Selecionamos os algoritmos **G1** e **G2** com $\alpha = 1,0$, **G3** e **G4** com $\alpha = 0,8$ e **G5** com $\alpha = 0,6$ que são os melhores em desvio e número de *bests*.

Tabela 5.22: Sumário dos resultados obtidos em relação à literatura para algumas versões de GRASP

Alg.	α	<i>#Best</i>	Desvio
G1	1,0	78	-1,6
G2	1,0	78	-1,6
G3	0,8	79	-1,6
G4	0,8	79	-1,6
G5	0,6	80	-1,7

De acordo com os resultados preliminares discutidos, o algoritmo **G5** com $\alpha=0,6$, tem em média os melhores custos e portanto ele será utilizado em comparações posteriores.

Prais e Ribeiro [41] propuseram uma variação no método GRASP chamada GRASP Reativo cuja principal característica é o auto-ajuste do valor de α de acordo com a qualidade das soluções previamente encontradas. Uma proposta baseada neste procedimento é apresentada na próxima seção.

A ideia do método é definir um conjunto $\Psi = \{\alpha_1, \dots, \alpha_m\}$ de m possíveis valores de α . A cada elemento $\alpha_i \in \Psi$ é associada uma probabilidade $p_i = 1/m, i = 1, \dots, m$,

inicialmente uniforme. Periodicamente, ou seja, a cada bloco de iterações, a distribuição de probabilidade é atualizada, utilizando-se as informações coletadas das soluções obtidas anteriormente. Na aquisição destas informações várias estratégias podem ser usadas. Em uma delas, o valor médio obtido com cada α_i serve para determinar as probabilidades de cada α , isto é, no próximo período de iterações, α_i com maior valor de p_i poderá ser utilizado mais frequentemente.

Em nossa proposta, o conjunto Ψ de valores de α pode ter no máximo 10 valores. O número de valores de α é definido durante o algoritmo de acordo com o seu desempenho, isto é, um valor de $alpha$ é utilizado por no máximo 5 iterações consecutivas sem melhora. A Figura 5.5 apresenta o pseudocódigo do algoritmo implementado.

Figura 5.5: Pseudocódigo do algoritmo GRASP Reativo

```

Heurística GRASPReativo(MAX_ITER)
1: iter_Alfa ← 0;
2: mudanca ← 0;
3: primeira_mudanca ← 1;
4: alfa ← 0,1;
5: repita
6:   se mudanca = 1 então
7:     se primeira_mudanca = 1 então
8:       alfa ← 0,5;
9:       primeira_mudanca ← 0;
10:    se não
11:      se alfa > 0,9 então
12:        alfa ← 0,1;
13:      se não
14:        alfa ← alfa + 0,1;
15:      fim se
16:    fim se
17:    iter_alfa ← 0;
18:  fim se
19:  Solucao ← Constroi_Solucao(alfa);
20:  Solucao ← Busca_Local(Solucao);
21:  se Solucao tem o menor custo até o momento então
22:    Atualizar_Solucao(Solucao, Melhor_Solucao);
23:    iter_alfa ← 0;
24:  se não
25:    iter_alfa ← iter_alfa + 1;
26:  fim se
27:  se iter_alfa >= 5 então
28:    mudanca ← 1;
29:  fim se
30: até alcançar MAX_ITER
31: retornar (Melhor_Solucao);
end

```

A diferença entre este e o método GRASP proposto está nas alterações dos valores de α durante as iterações do algoritmo. O método funciona da seguinte maneira: inicialmente, α é igual a 0,1. Um mesmo valor de α é utilizado até que 5 iterações consecutivas sem que haja melhora no custo da melhor solução sejam realizadas. A cada 5 iterações consecutivas sem melhora o valor de α é incrementado em 0,1 até que alcance o valor de 0,9. Depois disto, α retorna a 0,1. Este processo é repetido sempre da mesma maneira, com exceção da primeira mudança de α que varia de 0,1 diretamente para 0,5.

Como pode ser visto no pseudocódigo apresentado, a melhora da solução é investigada após a etapa de construção e busca local. Desta maneira, prioriza-se o uso de valores de α capazes de obter as melhores soluções.

Seguindo os mesmos algoritmos de construção (C3, C4, C7, C8 e C9) e buscas local (MV2 e PM2) utilizados na formação das heurísticas GRASP, as seguintes versões de GRASP Reativo são propostas (Tabela 5.23).

Tabela 5.23: Algoritmos GRASP Reativo

Reativo	Construção	Busca Local	Reativo	Construção	Busca Local
G1R	C3	MV2	G6R	C3	PM2
G2R	C4	MV2	G7R	C4	PM2
G3R	C7	MV2	G8R	C7	PM2
G4R	C8	MV2	G9R	C8	PM2
G5R	C9	MV2	G10R	C9	PM2

As Tabelas 5.24 e 5.25 mostram um sumário dos resultados encontrados. Para cada algoritmo proposto, são apresentados: o desvio percentual médio da melhor solução encontrada nos algoritmos GRASP Reativo propostos; o número de melhores soluções encontradas e o tempo médio de processamento nas 96 instâncias testadas.

Tabela 5.24: Sumário dos resultados dos algoritmos GRASP Reativo Propostos

Alg.	G1R	G2R	G3R	G4R	G5R	G6R	G7R	G8R	G9R	G10R
Desvio (%)	0,5	0,5	0,7	0,7	0,4	8,0	13,6	7,7	11,8	17,8
#Best	71	71	59	59	73	31	22	32	23	13
Tempo	85,8	85,5	78,4	79,0	88,2	24,5	74,6	23,8	69,7	346,0

A Tabela 5.25 mostra o desvio percentual dos algoritmos GRASP propostos quando comparados ao melhor da literatura e o número de melhores soluções obtidas nas 96 instâncias testadas. Selecionamos os algoritmos G1R, G2R, G3R, G4R e G5R que são os melhores em desvio e número de *bests*.

Tabela 5.25: Sumário dos resultados obtidos em relação a literatura para algumas versões de GRASP Reativo

Alg.	#Best	Desvio
G1R	74	-1,3
G2R	74	-1,3
G3R	75	-1,2
G4R	75	-1,2
G5R	77	-1,5

De acordo com os resultados preliminares discutidos, constata-se que a melhoria de custos alcançada por esta proposta de GRASP Reativo não supera a anterior. Algumas outras tentativas de implementações foram feitas, entre elas a proposta clássica do GRASP Reativo apresentado em [41], nas quais resultados semelhantes foram encontrados. Acredita-se que o motivo deste está nos exaustivos testes feitos para calibragem do parâmetro α que fez com que se pudesse chegar a uma implementação de algoritmo GRASP promissora. Faz-se, portanto, a opção pelo GRASP tradicional por se tratar de uma implementação mais simples.

5.4 Busca Tabu proposta para o PADE

A partir de uma solução inicial, o método BTP é executado até que um dado número de iterações sem melhora seja alcançado.

A cada iteração, o melhor movimento é identificado. Um movimento pode ser de melhora ou não. Com a execução do movimento, a solução corrente e a lista tabu são atualizadas para a próxima iteração.

O método avalia todos os movimentos de atividades definidos sobre a vizinhança de $V1$. A cada movimento, a heurística RSPA é executada a fim de realocar os recursos ociosos de acordo com a nova matriz de atividades decorrente deste movimento. Um movimento é classificado tabu se ocorrer pelo menos uma de duas situações: se é um movimento que já consta na lista, e/ou, se o movimento é reverso a algum outro da lista tabu.

Uma lista tabu dinâmica é utilizada para armazenar movimentos recentes de atividades. A lista tabu é implementada pela política FIFO e guarda as seguintes informações: os números dos dois espaços de trabalho nos quais ocorre a troca (relocação) e os números das atividades movimentadas. O tamanho da lista é limitado inferiormente por um valor (\mathcal{L}_i) e superiormente por um valor (\mathcal{L}_s). No início do método e a cada N iterações sem melhora, é escolhido um valor l aleatoriamente entre \mathcal{L}_s e \mathcal{L}_i , que passa ser o tamanho da lista tabu na próxima iteração. Se l é maior do que o tamanho de lista atual, os movimentos do início da lista são retirados até que a lista fique com tamanho l .

Se um movimento é tabu, este pode ser ainda considerado se atender ao critério de aspiração, que permite que um movimento tabu seja aceito desde que o custo da solução gerada pelo movimento seguido da heurística RSPA seja melhor do que a melhor solução até o momento.

O pseudocódigo de BTP pode ser visto na Figura 5.6.

A fase mais crítica do método é definir o tamanho da lista tabu de forma a evitar que ocorra ciclagem e um tempo excessivo de processamento. Vários trabalhos têm

Figura 5.6: Algoritmo BTP

```

Heurística BTP(Iter_Sem_Melhora, Sol_Inicial, Custo_Sol_Inicial,  $\mathcal{L}_i$ ,  $\mathcal{L}_s$ ,  $\mathcal{N}$ )
1:  $i \leftarrow 0$ ;
2:  $i\_lista \leftarrow 0$ ;
3:  $Melhor\_Sol \leftarrow Sol\_Inicial$ ;
4:  $Melhor\_Custo \leftarrow Custo\_Sol\_Inicial$ ;
5:  $Sol\_Corrente \leftarrow Sol\_Inicial$ ;
6:  $Custo\_Sol\_Corrente \leftarrow Custo\_Sol\_Inicial$ ;
7:  $l \leftarrow$  Escolher aleatoriamente um valor entre  $\mathcal{L}_i$  e  $\mathcal{L}_s$ ;
8: Inicializar a lista tabu vazia;
9: enquanto  $i < Iter\_Sem\_Melhora$  faça
10:   enquanto  $\exists$  movimento sem investigar na vizinhança  $\mathcal{V}1(Sol\_Corrente)$  faça
11:     Avaliar movimento em  $Sol\_corrente$ ;
12:     Executar a heurística RSPA em  $Sol\_corrente$ ;
13:   fim enquanto
14:   Identificar o melhor movimento seguido de RSPA;
15:   se Movimento é não tabu ou é permitido pelo critério de aspiração então
16:     Executar o movimento e obtém  $Sol$ ;
17:     Executar a heurística RSPA em  $Sol$ ;
18:      $Sol\_Corrente \leftarrow Sol$ ;
19:      $Custo\_Sol\_Corrente \leftarrow f(Sol)$ ;
20:   se não
21:     Selecionar e executar o próximo melhor movimento não tabu;
22:   fim se
23:   Atualizar lista tabu;
24:   se  $Custo\_Sol\_Corrente < Melhor\_Custo$  então
25:      $Melhor\_Sol \leftarrow Sol\_Corrente$ ;
26:      $Melhor\_Custo \leftarrow Custo\_Sol\_Corrente$ ;
27:      $i \leftarrow 0$ ;
28:      $i\_lista \leftarrow 0$ ;
29:   se não
30:      $i \leftarrow i + 1$ ;
31:      $i\_lista \leftarrow i\_lista + 1$ ;
32:   fim se
33:   se  $i\_lista \geq \mathcal{N}$  então
34:     Escolher aleatoriamente um valor entre  $\mathcal{L}_i$  e  $\mathcal{L}_s$ ;
35:     Adaptar lista tabu;
36:      $i\_lista \leftarrow 0$ ;
37:   fim se
38: fim enquanto
39: Retornar  $Melhor\_Sol$  e  $Melhor\_Custo$ ;
end

```

sido publicados no sentido de definir o melhor tamanho de lista para determinadas classes de problemas, tais como problemas de *scheduling* [42, 43] e do Caixeiro Viajante [44, 45, 46].

Neste trabalho também buscamos determinar o tamanho da lista tabu que gere melhores resultados gerais para o conjunto de instâncias considerado. Para isto, é proposto um conjunto de 10 tamanhos de lista, estabelecendo diferentes limite inferiores e superiores, identificados como promissores a partir do trabalho de . O conjunto de tamanhos de lista tabu é apresentado na Tabela 5.26. Considere n_ati o número de atividades do problema.

Tabela 5.26: Limites inferiores e superiores para lista tabu

Tamanho	li	ls
1	$0,7 * \sqrt{n_ati}$	$1,1 * \sqrt{n_ati}$
2	$0,7 * \sqrt{n_ati}$	$0,7 * n_ati$
3	$0,7 * \sqrt{n_ati}$	n_ati
4	$0,7 * \sqrt{n_ati}$	$1,1 * n_ati$
5	$1,1 * \sqrt{n_ati}$	$0,7 * n_ati$
6	$1,1 * \sqrt{n_ati}$	n_ati
7	$1,1 * \sqrt{n_ati}$	$1,1 * n_ati$
8	$1,1 * \sqrt{n_ati}$	$(n_ws - 1) * n_ati$
9	$0,7 * n_ati$	n_ati
10	$0,7 * n_ati$	$1,1 * n_ati$

O estudo é realizado da seguinte maneira: para cada tamanho de lista tabu são feitas 10 execuções de cada combinação do algoritmo de construção (Seção 4.2) com BTP. A partir daí, são analisados os resultados considerando o número de melhores custos encontrados no conjunto de instâncias, o desvio percentual médio da melhor solução e o tempo médio de processamento. Um sumário destes resultados são apresentados nas Tabelas 5.27 a 5.32.

Tabela 5.27: Variação do tamanho de lista tabu para o algoritmo de Construção C1

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	1,2	0,6	0,4	0,4	0,4	18,0	1,6	18,0	18,0	18,0
$\# Best$	55	68	71	73	71	16	67	16	16	16
Tempo	6,7	10,2	10,6	10,9	9,5	0,1	8,5	0,1	0,1	0,0

Tabela 5.28: Variação do tamanho de lista tabu para o algoritmo de Construção C2

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	27,8	27,8	27,8	27,8	0,3	27,8	27,8	0,3	27,8	27,8
$\# Best$	10	10	10	10	78	10	10	79	10	10
Tempo	0,2	0,2	0,2	0,2	9,6	0,2	0,2	10,9	0,2	0,2

Tabela 5.29: Variação do tamanho de lista tabu para o algoritmo de Construção C3

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	17,1	17,1	17,1	17,1	0,3	17,1	17,1	0,4	17,1	17,1
# <i>Best</i>	18	18	18	18	83	18	18	77	18	18
Tempo	0,1	0,1	0,1	0,1	11,7	0,1	0,1	10,2	0,1	0,1

Tabela 5.30: Variação do tamanho de lista tabu para o algoritmo de Construção C4

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	27,1	27,1	27,1	27,1	0,2	27,1	27,1	0,3	27,1	27,1
# <i>Best</i>	11	11	11	11	85	11	11	79	11	11
Tempo	0,2	0,2	0,2	0,2	11,7	0,2	0,2	10,2	0,2	0,2

Tabela 5.31: Variação do tamanho de lista tabu para o algoritmo de Construção C5

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	20,0	20,0	20,0	20,0	0,2	20,0	20,0	0,2	20,0	20,0
# <i>Best</i>	12	12	12	12	86	12	12	77	12	12
Tempo	0,0	0,0	0,0	0,0	12,4	0,0	0,0	10,3	0,0	0,0

Tabela 5.32: Variação do tamanho de lista tabu para o algoritmo de Construção C6

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	27,5	27,5	27,5	27,5	0,2	27,5	27,5	0,2	27,5	27,5
# <i>Best</i>	5	5	5	5	87	5	5	78	5	5
Tempo	0,2	0,2	0,2	0,2	12,4	0,2	0,2	10,8	0,2	0,2

Tabela 5.33: Variação do tamanho de lista tabu para o algoritmo de Construção C7

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	1,5	0,5	0,3	0,4	0,4	0,5	0,3	0,7	0,4	0,4
# <i>Best</i>	53	70	80	74	75	71	77	67	75	73
Tempo	6,0	10,8	10,5	9,8	11,3	9,5	10,7	10,1	9,7	10,1

Tabela 5.34: Variação do tamanho de lista tabu para o algoritmo de Construção C8

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	1,5	0,5	0,3	0,4	0,4	0,5	0,3	0,7	0,4	0,4
<i>#Best</i>	53	70	80	74	75	71	77	67	75	73
Tempo	6,0	10,7	10,4	9,6	11,3	9,4	10,8	10,0	9,7	10,4

Tabela 5.35: Variação do tamanho de lista tabu para o algoritmo de Construção C9

Tamanho	1	2	3	4	5	6	7	8	9	10
Desvio(%)	1,5	0,4	0,4	0,3	0,3	0,3	0,3	0,6	0,4	0,5
<i>#Best</i>	57	72	73	75	74	73	76	67	74	71
Tempo	6,4	9,9	11,5	10,4	9,1	10,6	10,6	8,2	10,5	9,9

As tabelas demonstram que para a construção C1 o tamanho 4 para limites da lista tabu rendeu 73 melhores resultados para as 96 instâncias testadas e o desvio percentual médio foi de 0,4%. A mesma análise feita para as demais construções resulta nos valores apresentados na Tabela 5.36.

Tabela 5.36: Melhores combinações de algoritmo de construção com tamanho de lista tabu

Const.	Tam (t)	Desvio(%)	<i>#Best</i>
C1	4	0,4	73
C2	8	0,3	79
C3	5	0,3	83
C4	5	0,2	85
C5	5	0,2	86
C6	5	0,2	87
C7	3	0,3	80
C8	3	0,3	80
C9	7	0,3	76

Além disso, nota-se que os tamanhos de lista que proporcionam os melhores custos, são aqueles que têm a maior média de tempo de processamento, resultado este que é habitualmente encontrado ao se trabalhar com heurísticas.

Para definir a melhor combinação de construção e BTP é analisados o desempenho de cada uma, com seu melhor tamanho de lista diante dos melhores custos da literatura. A Tabela 5.37 reporta estes resultados.

Tabela 5.37: Comparação dos melhores algoritmos busca tabu com literatura

	C1t4	C2t8	C3t5	C4t5	C5t5	C6t5	C7t3	C8t3	C9t7
Desvio (%)	-2,31	-2,28	-2,33	-2,33	-2,20	-2,20	-2,36	-2,36	-2,32
# <i>Best</i>	80	81	81	81	80	80	82	82	82

Pode-se observar que todas as versões propostas são capazes de diminuir, na média, o melhor custo conhecido. Dentre os algoritmos testados, BTP utilizando soluções iniciais geradas por C7 e C8 apresenta o maior desvio percentual médio e o maior número de resultados iguais ou melhores que a literatura. A primeira, portanto, será considerada para comparações posteriores. No Anexo D são mostrados os menores custos, desvio percentual médio da melhor solução conhecida e tempo médio de processamento alcançados pelas melhores versões de busca tabu.

Capítulo 6

Heurísticas Híbridas

São propostas neste capítulo, heurísticas híbridas construídas a partir da junção de duas metaheurísticas em um mesmo procedimento. A intenção teórica desta proposta está em tentar escapar mais facilmente de ótimos locais o que, na prática, indicaria uma melhora nos resultados obtidos com os dois métodos isoladamente. São discutidas heurísticas a partir da combinação de ILS com outras: VNS e Busca Tabu. Em seguida, é proposta uma heurística baseada nos métodos VNS e Busca Tabu e, por fim, um híbrido de GRASP e Busca Tabu.

6.1 Heurística ILS + VNS Propostas para o PADE

Nesta implementação é feita uma tentativa de juntar as heurísticas ILS e VNS, substituindo-se a fase de busca local do método ILS pelo algoritmo VNS1, do qual o passo de construção da solução foi retirado.

Esta proposta exige a definição de basicamente três parâmetros: o número de iterações sem melhora do ILS e do VNS e o número de vezes que cada perturbação é repetida. Durante testes identificou-se que o aumento do número de iterações do ILS não faz tanta diferença nos resultados obtidos quanto o número de iterações do VNS, ao invés disto, ainda traz pioras ao tempo de processamento. Nestes fase, chegou-se a 15, o número de iterações sem melhora do ILS e 20, o número de vezes que cada perturbação é executada. Quanto ao número de iterações do VNS chegamos a ter reduções de custo quando usamos valores a partir de 20. A Tabela 6.1 faz um comparativo com o uso de 10 e 30 iterações do VNS.

Tabela 6.1: Comparativo de resultados com diferentes números de iterações VNS no método ILS+VNS

Iterações	10	30
Desvio (%)	0,7	-0,6
# <i>Best</i>	62	67
Tempo	120,5	335,7

O desvio percentual entre os custos obtidos em comparação com os melhores valores conhecidos chega a 0,6% com 30 iterações entretanto, podemos observar o aumento do tempo médio de processamento. Por considerarmos este tempo alto para a melhoria de custos alcançadas, em vista dos resultados anteriormente relatados, resolvemos abandonar esta abordagem. Se observarmos os tempos médios de computação destas duas heurísticas implementadas individualmente, apresentadas no Capítulo 5 podemos justificar o crescimento deste tempo.

6.2 Heurísticas ILS + Busca Tabu Propostas para o PADE

Uma nova tentativa de obter uma heurística híbrida baseada na metaheurística ILS é apresentada aqui. Desta vez, a ideia é substituir a fase de busca local do ILS clássico pela Busca Tabu Proposta (BTP), apresentada no Capítulo 5, Seção 5.4. Nesta proposta é preciso definir os seguintes parâmetros: número de iterações sem melhora do ILS e da BTP e número de vezes em que uma perturbação é repetida do ILS. Alguns testes preliminares foram feitos até que se chegou a: 15 iterações ILS, 50 iterações BTP e 20 repetições de perturbações. Além disso, conforme a implementação da BTP, a cada 20% de iterações sem melhora, um novo tamanho de lista tabu é escolhido aleatoriamente. São propostas as seguintes versões de ILS de acordo com a variação do algoritmo de construção, todos executados com parâmetro $\alpha = 0$, apresentadas na Tabela 6.2.

Tabela 6.2: Algoritmos híbridos de ILS e BTP Propostos

Alg.	Construção
ILS1+BTP	C1
ILS2+BTP	C3
ILS3+BTP	C5
ILS4+BTP	C7
ILS5+BTP	C9

No Apêndice E são apresentados os melhores custos, os desvios percentuais da

melhor solução conhecida e os tempos médios de processamento, obtidos de 10 execuções da heurística ILS+BTP.

A Tabela 6.3 faz um resumo dos resultados obtidos pelos algoritmos, sendo mostrados o desvio percentual, o número de melhores soluções e o tempo de processamento médio dos algoritmos, em 10 execuções independentes. A Tabela 6.4 faz uma análise parecida considerando a melhor solução conhecida.

Tabela 6.3: Comparação dos algoritmos ILS+BTP com a literatura

Alg.	Desvio (%)	# <i>Best</i>	Tempo
ILS1+BTP	0,3	77	61,6
ILS2+BTP	0,3	74	65,5
ILS3+BTP	0,3	78	67,7
ILS4+BTP	0,3	79	67,6
ILS5+BTP	0,4	74	75,1

Tabela 6.4: Sumário da comparação dos algoritmos ILS+BTP com literatura

Alg.	Desvio (%)	# <i>Best</i>
ILS1+BTP	-2,64	81
ILS2+BTP	-2,60	82
ILS3+BTP	-2,65	82
ILS4+BTP	-2,67	82
ILS5+BTP	-2,59	81

Pelos resultados apresentados é possível ver que o algoritmo ILS4+BTP tem melhores números. Esta, portanto, é a versão escolhida para os próximos testes.

6.3 Heurísticas VNS + Busca Tabu Propostas para o PADE

Esta proposta substitui a fase de busca local do método VNS clássico pela Busca Tabu, conforme apresentado no trabalho de Sevaux e Sörensen [47]. Para tanto, utilizamos BTP. O pseudocódigo do algoritmo proposto é mostrado na Figura 6.1.

Figura 6.1: Algoritmo VNS+BTP

```

Heurística VNS+BTP()
1: Selecione o conjunto de estruturas de vizinhanças  $\mathcal{N}_k$ , com  $k = 1, \dots, k_{max}$ ;
2:  $s \leftarrow GeraSoluc\tilde{a}oInicial()$ ;
3: enquanto critério de parada não satisfeito faça
4:    $k \leftarrow 1$ ;
5:   enquanto  $k \leq k_{max}$  faça
6:      $s' \leftarrow$  Vizinho qualquer em  $\mathcal{N}_k(s)$ 
7:      $s'' \leftarrow BTP(s')$ ;
8:     se  $Z(s'') < Z(s)$  então
9:        $s \leftarrow s''$ ;
10:     $k \leftarrow 1$ ;
11:    se não
12:       $k \leftarrow k + 1$ ;
13:    fim se
14:  fim enquanto
15: fim enquanto
end

```

Combinando algoritmos de construções para a geração de solução inicial do VNS, são obtidas as seguintes propostas de algoritmos híbridos com busca tabu, apresentadas na Tabela 6.5. Nos testes é utilizado valor de α igual a 0 em todos os algoritmos de construção.

Tabela 6.5: Algoritmos híbridos de VNS e BTP Propostos

Alg.	Construção
VNS1+BTP	C1
VNS2+BTP	C3
VNS3+BTP	C5
VNS4+BTP	C7
VNS5+BTP	C9

O primeiro desafio desta proposta é o de encontrar uma boa combinação entre o número de iterações sem melhora dos dois métodos. Uma extensa bateria de testes foi realizada até que essa combinação trouxesse um bom desempenho levando-se em consideração os dois fatores mais desejados na implementação de métodos heurísticos: bom custo de soluções e tempo computacional razoável. Neste sentido, são feitas 20 iterações sem melhora do VNS e, em cada uma delas, 100 iterações sem melhora de BTP. A cada 20% de iterações sem melhora de BTP, um novo tamanho de lista tabu é escolhido. Além disso, foi necessária a realização de uma outra extensa bateria de testes para definir os limites para o tamanho da lista tabu.

No Apêndice F são mostrados os menores custos, desvio da melhor solução conhecida e tempo médio de processamento alcançados pelas versões de VNS+BTP para cada instância testada. A Tabela 6.6 apresenta um sumário dos resultados obtidos com os algoritmos VNS combinados com BTP. Para cada algoritmo são apresentados o tamanho da lista tabu utilizado (Tam), o desvio percentual médio da melhor solução ($Desvio$) obtida em 10 execuções independentes, o número de melhores soluções

obtidas dentro do conjunto das 96 instâncias ($\#Best$) e o tempo médio de processamento ($Tempo$). É interessante notar que os tamanhos de lista tabu que apresentam melhores resultados são os mesmos considerados melhores na implementação pura de BTP.

Tabela 6.6: Melhores combinações de algoritmos VNS+BTP com tamanho de lista tabu

Alg.	Tam	Desvio(%)	$\#Best$	Tempo
VNS1+BTP	4	0,2	82	256,0
VNS2+BTP	5	0,2	83	246,1
VNS3+BTP	5	0,1	83	315,9
VNS4+BTP	3	0,2	83	265,9
VNS5+BTP	7	0,1	81	267,3

A Tabela 6.7 mostra o desvio percentual destes algoritmos quando comparados ao melhor custo conhecido e o número de melhores soluções obtidas nas 96 instâncias testadas.

Tabela 6.7: Comparação dos melhores algoritmos VNS+BTP com literatura

Alg.	Desvio (%)	$\# Best$
VNS1+BTP	-3,02	82
VNS2+BTP	-3,01	82
VNS3+BTP	-3,06	82
VNS4+BTP	-3,02	81
VNS5+BTP	-3,05	82

Os resultados obtidos mostram que a heurística híbrida VNS3+BTP se destaca das demais apresentando o maior número de melhores soluções e menor desvio percentual médio da melhor solução encontrada pelos algoritmos. Além disso, este algoritmo teve um desvio médio da melhor solução conhecida de 3,1%. Por estes motivos, esta versão de algoritmo híbrido de VNS e Busca Tabu será considerada para as próximas comparações.

6.4 Heurísticas GRASP + Busca Tabu Proposta para o PADE

A heurística GRASP híbrida aqui implementada utiliza a heurística Busca Tabu Proposta como uma fase de intensificação na busca por melhores soluções. O método

GRASP utilizado é o mesmo proposto na Seção 5.3.1 do Capítulo 5, que a partir das combinações de construção e busca local foram denominadas: G1, G2, G3, G4, G5, G6, G7, G8, G9 e G10. Ao fim de cada iteração destes métodos a heurística BTP é chamada.

A parametrização deste método envolve o número de iterações e valor de α do método GRASP e, com relação à busca tabu, o número de iterações sem melhora, os limites de variação do tamanho da lista tabu, além da quantidade de iterações sem melhora para mudança do tamanho da lista tabu. Quanto ao valor de α inerente da fase de construção do GRASP foi utilizado o estudo feito na Seção 5.3.1, Capítulo 5. Desta maneira, o valor de α utilizado para cada versão de heurística GRASP+BTP é o mesmo escolhido como melhor na versão pura de GRASP. Para definir uma combinação adequada do número de iterações de cada método, testes preliminares foram feitos até que se chegou a 20 iterações GRASP e 50 iterações sem melhora de BTP. Os limites para tamanho da lista tabu também foi definido com base nas experiências anteriores. Utilizamos um dos limites que foi bem aproveitado pelas versões de BTP: $t=8$ (vide Tabela 5.26). Por fim, o número de iterações sem melhora para que ocorra a escolha de um novo tamanho de lista tabu utilizado é o mesmo da implementação de BTP: a cada 20% do número de iterações sem melhora.

No Apêndice G são apresentadas as tabelas com melhor custo, desvio da melhor solução conhecida e tempo médio de processamento de cada uma das 96 instâncias testadas. A Tabela 6.8 apresenta o desvio médio, o número de melhores soluções e o tempo de processamento das heurísticas GRASP+BTP, a partir da qual se destaca a versão G6+BTP que, das 96 instâncias testadas, pode obter 81 melhores custos, com desvio percentual de 0,2%.

Tabela 6.8: Sumário dos resultados do algoritmos GRASP+BTP

Alg.	Desvio (%)	# <i>Best</i>	Tempo
G1+BTP	0,3	76	102,2
G2+BTP	0,4	74	113,9
G3+BTP	0,2	78	106,6
G4+BTP	0,4	73	114,2
G5+BTP	0,3	79	199,4
G6+BTP	0,2	81	145,3
G7+BTP	0,4	73	108,6
G8+BTP	0,2	78	95,1
G9+BTP	0,4	74	106,9
G10+BTP	0,4	74	107,2

A Tabela 6.9 compara as heurísticas propostas com a literatura. A melhoria de custos obtida também com a versão G6+BTP é de 2,9%, chegando a encontrar 87 soluções de custo igual ou melhor que o melhor conhecido.

Tabela 6.9: Sumário da comparação dos algoritmos GRASP+BTP com literatura

Alg.	Desvio (%)	# <i>Best</i>
G1+BTP	-2,8	86
G2+BTP	-2,7	83
G3+BTP	-2,8	86
G4+BTP	-2,7	84
G5+BTP	-2,8	85
G6+BTP	-2,9	87
G7+BTP	-2,7	83
G8+BTP	-2,8	86
G9+BTP	-2,7	83
G10+BTP	-2,7	83

6.5 Considerações sobre Algoritmos Híbridos

Os algoritmos híbridos aqui apresentados são todos combinação de duas heurísticas já discutidas no Capítulo 5. É interessante ver que, ao contrário do que era esperado, cada algoritmo híbrido teve melhor desempenho utilizando um algoritmo de construção independente dos que produziram melhores resultados obtidos pela heurística guia (ILS, VNS, GRASP) e guiada (Busca Tabu).

Pela necessidade de um número menor de iterações até chegar a resultados promissores, as heurísticas híbridas propostas ILS+BTP e VNS+BTP, apresentaram um tempo de processamento mais baixo que os das heurísticas ILS e VNS clássicas propostas neste trabalho.

Capítulo 7

Experimentos e resultados computacionais

Este capítulo apresenta resultados de comparações e novos experimentos computacionais aplicados àquelas heurísticas propostas nos Capítulos 5 e 6 que obtiveram melhor desempenho nos teste preliminares. Nesta oportunidade, por questões de simplicidade, os algoritmos investigados daqui por diante são renomeados. Os nomes dados aos algoritmos e as heurísticas propostas a serem comparadas nas próximas seções são apresentadas na Tabela 7.1. Além disso, relembramos as configurações e parâmetros utilizados por cada método, onde: *Const.* é a construção utilizada; α o valor de α da construção; *Iter.* o número de iterações de cada método, para os híbridos, na sequência o número do iterações método principal (ILS, VNS ou GRASP)/número de iterações da BTP; *Vezez* o número de repetições da perturbação do ILS e $Tam(t)$, o tamanho da lista tabu. Vale lembrar que na BTP, a cada 20% do número de iterações sem melhora, é feita a troca de tamanho de lista.

Tabela 7.1: Metaheurísticas e heurísticas propostas

Alg.	Heurística Proposta	Configurações				
		Const.	α	Iter.	Vezez	$Tam(t)$
VNS	VNS1	C1	-	1500	-	-
ILS	ILS6	C1	-	700	20	-
GRASP	G5	C3	0,6	100	-	-
BTP	BTP	C7	-	1500	-	3
hILS	ILS4+BTP	C7	0,0	15/20	20	8
hVNS	VNS3+BTP	C5	0,0	20/100/20	-	5
hGRASP	G6+BTP	C3	0,8	20/50	-	8

7.1 Comparação dos resultados entre algoritmos exato, da literatura e propostos

A primeira comparação de resultados a ser feita é a partir das instâncias nas quais o ótimo é conhecido. A literatura reporta os custos de soluções exatas para 25 das 96 instâncias: de 1 a 24 e 27. Neste trabalho, a formulação matemática proposta em [17], apresentada na Seção 3.1.1, Capítulo 3, foi resolvida usando o CPLEX 11 e, além dessas instâncias, foi possível obter as soluções ótimas das instâncias 25, 26, 28 a 32, 35, 36, 39 e 40. A abordagem exata foi abandonada para as demais instâncias devido ao grande tempo de processamento. Esta análise de resultados é dividida em três partes: custos, desvios percentuais e tempos de processamento.

A Tabela 7.2 apresenta os custos obtidos pelo CPLEX, os melhores custos reportados na literatura e os melhores custos obtidos por cada uma das heurísticas propostas (Tabela 7.1) em 10 execuções usando sementes aleatórias independentes. Custos ótimos são destacados em negrito. Algumas constatações interessantes podem ser feitas na leitura desta tabela.

- Inicialmente, ao se observar a coluna *Lit.* que reproduz os custos reportados na literatura como melhores, aparentemente há algum problema no que diz respeito as instâncias 25, 26, 28, 32, 35, 36, 39 e 40, destacados na tabela em itálico e sublinhado. Estes custos são menores do que os obtidos pelo CPLEX. O modelo dado como entrada no CPLEX foi repetidas vezes avaliado para que se pudesse atestar sua correção e as soluções exatas obtidas para estas instâncias, que como já dissemos contradiz a literatura, estão expostas no Apêndice I. Para as demais instâncias, os resultados da literatura coincidem com o valor da solução ótima do CPLEX, exceto para a instância 30.
- Em relação aos algoritmos propostos, as heurísticas ILS e hGRASP são as que chegaram mais vezes a soluções de custo ótimo: em 33 e em 34, respectivamente. Enquanto os algoritmos GRASP, BTP, hILS e hVNS obtiveram o ótimo em 28 instâncias. Há de se observar um comportamento similar nestes algoritmos já que todos atingem o mesmo custo em cada uma das 7 instâncias nas quais eles não chegaram ao ótimo. O algoritmo VNS não segue esta tendência: chegou ao ótimo 31 vezes.

Tabela 7.2: Custos dos algoritmos CPLEX, literatura e propostos

Inst.	CPLEX	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
1	16	16	16	16	16	16	16	16	16
2	25	25	25	25	26	26	26	26	25
3	18	18	18	18	18	18	18	18	18
4	25	25	25	25	26	26	26	26	25
5	16	16	16	16	16	16	16	16	16
6	27	27	27	27	27	27	27	27	27
7	16	16	16	16	16	16	16	16	16
8	31	31	31	31	31	31	31	31	31
9	25	25	25	25	25	25	25	25	25
10	46	46	46	46	46	46	46	46	46
11	32	32	32	32	32	32	32	32	32
12	41	41	41	41	43	43	43	43	41
13	28	28	28	28	28	28	28	28	28
14	45	45	45	45	45	45	45	45	45
15	35	35	35	35	35	35	35	35	35
16	49	49	49	49	49	49	49	49	49
17	35	35	35	35	35	35	35	35	35
18	60	60	61	60	62	62	62	62	62
19	46	46	46	46	46	46	46	46	46
20	60	60	60	60	63	63	63	63	60
21	46	46	47	46	48	48	48	48	46
22	67	67	67	67	67	67	67	67	67
23	55	55	55	55	56	56	56	56	55
24	74	74	74	74	74	74	74	74	74
25	31	<u>30</u>	31	31	31	31	31	31	31
26	43	<u>42</u>	43	43	43	43	43	43	43
27	43	43	43	43	43	43	43	43	43
28	55	<u>54</u>	55	55	55	55	55	55	55
29	29	29	29	29	29	29	29	29	29
30	49	50	49	49	49	49	49	49	49
31	42	42	43	43	42	42	42	42	42
32	69	<u>66</u>	69	69	69	69	69	69	69
35	73	<u>68</u>	74	76	73	73	73	73	73
36	95	<u>90</u>	96	95	95	95	95	95	95
39	68	<u>67</u>	68	71	69	69	69	69	69
40	108	<u>104</u>	108	108	108	108	108	108	108

A Tabela 7.3 reporta os desvios percentuais entre o custo exato e os demais algoritmos comparados.

Tabela 7.3: Desvio percentual entre custo exato, melhor solução da literatura e dos algoritmos propostos

Inst.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
1	-	-	-	-	-	-	-	-
2	-	-	-	4,0	4,0	4,0	4,0	-
3	-	-	-	-	-	-	-	-
4	-	-	-	4,0	4,0	4,0	4,0	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-
12	-	-	-	4,9	4,9	4,9	4,9	-
13	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-
18	-	1,7	-	3,3	3,3	3,3	3,3	3,3
19	-	-	-	-	-	-	-	-
20	-	-	-	5,0	5,0	5,0	5,0	-
21	-	2,2	-	4,3	4,3	4,3	4,3	-
22	-	-	-	-	-	-	-	-
23	-	-	-	1,8	1,8	1,8	1,8	-
24	-	-	-	-	-	-	-	-
25	<u>-3,2</u>	-	-	-	-	-	-	-
26	<u>-2,3</u>	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-
28	<u>-1,8</u>	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-
30	2,0	-	-	-	-	-	-	-
31	-	2,4	2,4	-	-	-	-	-
32	<u>-4,3</u>	-	-	-	-	-	-	-
35	<u>-6,8</u>	1,4	4,1	-	-	-	-	-
36	<u>-5,3</u>	1,1	-	-	-	-	-	-
39	<u>-1,5</u>	-	4,4	1,5	1,5	1,5	1,5	1,5
40	<u>-3,7</u>	-	-	-	-	-	-	-

Nesta tabela, desvio percentual nulo é indicada pelo símbolo (-). A coluna *Lit.*, em função dos custos menores que os exatos obtidos, apresenta uma discrepância: desvios percentuais negativos, quando o menor desvio percentual que pode se chegar do custo exato é zero. Quanto aos algoritmos propostos, o maior desvio percentual apresentada é de 5,0%, dada pelos algoritmos GRASP, BTP, hVNS e hGRASP na instância 20.

A Tabela 7.4 faz uma comparação de tempos (em segundos). São dados os tempos de execução do CPLEX, os tempos médios de processamento do algoritmo que obteve melhor custo da literatura e os tempos para obtenção da melhor solução, em 10 execuções, dos algoritmos propostos. A máquina reportada em [19], um Pentium IV 2,4GHz, tem seu poder de computação estimado em 4595 MFLOPS

(<http://www.activewin.com/reviews/hardware/processors/intel/p424ghz/benchs.shtml>) enquanto a utilizada para os testes neste trabalho, em 44300 MFLOPS (http://techgauge.com/print/intel_core_2_quad_q6600). Para ajustar os tempos computacionais da literatura foi utilizada a fração 4595/44300 MFLOPS. Além disso, vale acrescentar que são considerados os tempos referentes ao melhor custo obtido em [19] para cada instância, onde é reportado que os tempos são médios de 5 ou de 18 execuções, de acordo com o algoritmo que obteve o melhor custo.

Nesta tabela, é importante enfatizar o crescimento de tempo do algoritmo exato em função do aumento do problema (vide Apêndice I).

Tabela 7.4: Tempos do algoritmo exato, e da melhor solução da literatura e dos algoritmos propostos

Inst.	CPLEX	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
1	0,3	0,8	1,9	0,9	0,1	0,0	0,1	0,8	0,2
2	0,3	1,0	1,6	0,5	0,1	0,0	0,1	0,7	0,2
3	0,3	0,8	1,9	0,4	0,1	0,0	0,1	1,5	0,2
4	3,5	0,8	1,3	0,3	0,0	0,0	0,1	1,0	0,2
5	1,3	1,6	1,8	0,5	0,1	0,0	0,1	0,8	0,4
6	5,5	2,1	2,0	0,5	0,1	0,0	0,1	1,1	0,2
7	3,5	2,3	1,8	0,4	0,1	0,0	0,1	3,3	0,2
8	0,9	1,6	1,1	0,3	0,1	0,0	0,1	1,1	0,2
9	6,8	1,8	3,6	1,1	0,2	0,0	0,2	2,4	0,4
10	19,0	2,1	3,4	1,1	0,1	0,0	0,2	3,8	0,4
11	7,7	1,8	3,1	0,9	0,2	0,0	0,2	3,1	0,5
12	15,0	2,1	2,3	0,7	0,1	0,0	0,1	2,2	0,4
13	11,0	2,6	3,6	0,9	0,2	0,0	0,2	5,3	0,5
14	18,9	1,8	3,0	1,4	0,1	0,0	0,2	1,5	0,4
15	17,8	2,1	3,6	0,7	0,2	0,0	0,2	7,5	0,5
16	4,5	1,8	2,8	1,1	0,1	0,0	0,2	2,1	0,5
17	16,2	7,5	5,9	2,3	0,4	0,1	0,3	4,1	0,8
18	62,3	4,9	5,5	2,7	0,3	0,0	0,4	6,9	0,8
19	26,6	5,2	5,1	1,5	0,4	0,1	0,4	4,5	1,0
20	57,8	4,7	3,9	1,1	0,2	0,0	0,2	8,1	0,6
21	46,5	4,7	6,1	3,9	0,4	0,1	0,3	3,0	1,0
22	103,2	4,7	5,3	1,7	0,3	0,0	0,3	6,2	0,7
23	24,6	4,2	6,0	1,4	0,3	0,1	0,4	5,3	1,1
24	32,6	3,1	4,6	0,9	0,2	0,0	0,3	4,3	0,8
25	59551,0	7,5	13,7	5,4	1,1	0,2	1,0	16,4	3,3
26	20663,1	8,3	13,8	11,3	1,0	0,1	1,1	18,3	3,4
27	1008,4	4,9	14,3	11,0	1,0	0,2	1,1	15,6	2,4
28	582,3	4,4	9,3	6,3	0,7	0,1	0,7	8,1	1,8
29	94397,6	10,2	13,9	5,1	0,9	0,1	0,9	23,5	3,7
30	27609,2	8,6	14,5	9,7	0,9	0,1	0,8	11,9	3,0
31	1950,3	7,8	14,7	6,4	1,0	0,2	1,1	13,9	4,3
32	3716,4	7,3	10,3	4,3	0,7	0,1	0,6	8,8	2,2
35	790670,6	12,8	30,4	17,7	2,7	0,4	2,3	32,3	16,8
36	57012,0	117,9	19,8	7,5	1,9	0,2	1,4	17,6	10,5
39	177400,5	13,0	28,0	19,1	3,1	0,5	3,1	32,5	14,9
40	211216,3	15,1	18,6	13,1	2,2	0,2	1,6	19,5	4,4

A Tabela 7.5 sumariza os resultados obtidos pela literatura e pelos algoritmos

propostos em comparação com o exato. São apresentados o desvio percentual médio do custo exato, o número de soluções exatas obtidas e o tempo médio de processamento. O desvio percentual do algoritmo da literatura é desconsiderado por ter sido descaracterizado nas instâncias 25, 26, 28 e 32.

Tabela 7.5: Comparação dos algoritmos da literatura e propostos com exato

Alg.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
Desvio(%)	?	0,2	0,3	0,8	0,8	0,8	0,8	0,1
<i>#Best</i>	27	31	33	28	28	28	28	34
Tempo	7,9	7,2	4,0	0,3	0,0	0,6	8,3	1,8

Seguindo este relatório de resultados, o próximo passo é avaliar os algoritmos diante das instâncias nas quais o custo exato não foi processado através do CPLEX. Dividimos as instâncias pelo número de locações. Assim, são apresentados nas Tabelas 7.6, 7.7 e 7.8 os melhores custos reportados na literatura e os melhores custos obtidos com os algoritmos propostos em 10 execuções independentes, para as instâncias com 12, 20 e 32 locações.

Nas instâncias com 12 locações (Tabela 7.6), nota-se que das 12 instâncias nas quais o ótimo não é conhecido, o algoritmo da literatura obteve custo melhor em 10 instâncias. É neste conjunto de instâncias que os algoritmos propostos encontram mais dificuldade em alcançar melhorias de custos. O algoritmo proposto VNS não é capaz de encontrar nenhuma solução de custo melhor que a literatura, contrariando o desempenho que teve no primeiro grupo de instâncias estudado, assim como o algoritmo ILS que conseguiu empatar com a literatura em 2 instâncias. Os algoritmos GRASP, BTP e os híbridos obtiveram todos soluções de mesmo custo e nas instâncias 33 e 42, obtiveram custos melhores do que a literatura.

Tabela 7.6: Custos dos algoritmos da literatura e propostos para instâncias com 12 locações

Inst.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
33	53	57	55	52	52	52	52	52
34	72	73	74	72	72	72	72	72
37	47	50	50	48	48	48	48	48
38	77	82	82	83	83	83	83	83
41	78	80	78	78	78	78	78	78
42	104	106	104	102	102	102	102	102
43	110	112	110	110	110	110	110	110
44	137	143	140	140	140	140	140	140
45	66	68	71	66	66	66	66	66
46	111	116	115	116	116	116	116	116
47	111	118	116	115	115	115	115	115
48	169	171	171	171	171	171	171	171

Já nas instâncias com 20 locações (Tabela 7.7) é possível ver melhorias mais efetivas de custos pelos algoritmos propostos. Em todas as instâncias, exceto na 69 na qual houve empate com a literatura, pelo menos um dos algoritmos propostos foi capaz de encontrar soluções de melhores custos. Os algoritmos VNS e ILS mantêm seu desempenho abaixo dos demais propostos, mas são agora capazes de encontrar soluções melhores do que a literatura e deve-se destacar a instância 58 e 72 nas quais o ILS chega a ótimos locais, o que os demais algoritmos não conseguiram. Até a instância 58 GRASP, BTP e os híbridos obtêm soluções de mesmo custo.

Tabela 7.7: Custos dos algoritmos da literatura e propostos para instâncias com 20 locações

Inst.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
49	45	44	44	44	44	44	44	44
50	63	60	59	60	60	60	60	60
51	55	55	56	53	53	53	53	53
52	98	89	89	89	89	89	88	89
53	49	47	47	47	47	47	47	47
54	67	63	63	62	62	62	62	62
55	63	60	63	60	60	60	60	60
56	97	90	89	89	89	89	89	89
57	67	67	69	66	66	66	66	66
58	106	102	96	100	100	100	100	100
59	101	96	97	94	93	93	93	93
60	159	154	155	153	153	153	153	153
61	82	75	75	74	74	74	74	74
62	129	117	117	113	112	112	112	112
63	121	121	120	119	116	116	116	116
64	190	180	176	176	175	175	175	174
65	105	105	101	101	101	98	98	99
66	156	152	150	153	151	151	151	150
67	157	155	150	147	147	147	144	145
68	234	227	224	218	220	220	218	218
69	112	119	118	116	112	112	112	113
70	178	176	173	169	166	166	166	166
71	170	173	169	167	165	162	162	162
72	265	252	247	248	248	248	248	248

Por fim, no último conjunto de instâncias de 32 locações (Tabela 7.8), os algoritmos mantêm o mesmo desempenho, excetuando-se a heurística VNS que não consegue soluções melhores que a literatura em 11 instâncias. Além disso, hGRASP perde para hVNS e hILS (melhores resultados em 12 instâncias, contra 20 e 17 das outras duas).

Tabela 7.8: Custos dos algoritmos da literatura e propostos para instâncias com 32 locações

Inst.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
73	74	71	71	71	71	71	71	71
74	97	92	96	91	90	89	89	90
75	110	105	109	104	101	101	101	101
76	155	155	150	145	144	144	144	144
77	73	70	70	70	70	70	70	70
78	101	97	98	93	92	92	92	92
79	110	109	110	100	99	99	99	99
80	175	171	164	164	165	166	163	163
81	119	117	118	114	109	109	109	110
82	176	181	168	167	166	161	161	161
83	192	191	197	192	188	184	183	183
84	282	291	287	271	269	269	264	266
85	125	126	125	122	117	117	117	117
86	192	193	196	188	182	177	179	178
87	193	199	191	185	182	178	178	179
88	302	292	292	290	275	274	274	275
89	171	171	174	171	161	157	157	158
90	262	275	256	256	243	244	239	246
91	284	295	287	281	272	264	262	262
92	395	395	374	387	388	380	374	379
93	189	199	185	186	178	171	174	177
94	281	288	282	284	271	268	270	268
95	318	337	328	318	307	302	295	300
96	464	482	479	454	439	435	438	438

As Tabelas H.1, H.2 e H.3 apresentam os desvios percentuais de custo entre a literatura e os algoritmos propostos. Estas tabelas só vêm reforçar o que foi dito a respeito dos custos, por isto são apresentadas no Apêndice H.

As Tabelas 7.9, 7.10 e 7.11 apresentam os tempos de processamentos (em segundos) dos algoritmos da literatura e propostos. Os tempos seguem um comportamento similar em todos os conjuntos de instâncias: o algoritmo mais rápido é o BTP. Este, GRASP e hILS são, em média, mais rápidos que o da literatura. O algoritmo mais lento é o VNS.

Tabela 7.9: Tempos de processamento da literatura e dos algoritmos propostos para instâncias com 12 locações

Inst.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
33	17,2	29	16,9	3,2	0,5	3,47	46	6,68
34	20,6	26,2	21,3	2,97	0,52	3,16	28,4	15,22
37	12,0	29	17,7	3,18	0,38	3,3	27,7	7,13
38	26,6	23,5	33,3	2,72	0,3	2,03	24,1	6,26
41	25,0	50,1	50	6,81	1	5,24	51	13,11
42	20,0	44,3	22,4	5,63	0,87	4,58	47	27,57
43	267,9	48,8	50,1	6,1	0,74	4,36	65,5	14,46
44	20,8	31,1	32,8	4,13	0,42	2,54	38	13,29
45	47,6	49,8	42,4	7,02	1,2	5,24	51,9	14,19
46	39,6	48,3	53,1	5,83	0,69	4,48	62,7	25,1
47	33,8	43,1	26,7	6,31	1,51	7,83	54,5	15,72
48	23,7	31,8	19,9	4,28	0,48	2,91	46	18,86

Tabela 7.10: Tempos de processamento da literatura e dos algoritmos propostos para instâncias com 20 locações

Inst.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
49	25,3	59,9	23,2	9,7	1,1	7,5	117,4	43,5
50	25,5	54,5	53,7	8,9	1,6	7,0	72,3	50,9
51	24,0	56,2	19,9	9,8	2,1	8,4	114,7	53,8
52	20,8	37,6	17,6	5,9	0,7	4,3	63,6	23,1
53	36,2	69,8	28,2	10,4	1,2	9,2	102,5	22,3
54	24,2	67,0	86,7	10,0	1,9	10,8	110,3	54,9
55	22,6	93,3	28,5	10,6	2,0	7,9	98,3	34,4
56	27,6	67,1	34,2	7,9	1,0	5,4	61,8	17,7
57	81,2	202,8	73,9	29,5	5,3	22,3	209,8	71,9
58	106,2	167,3	90,7	24,7	3,1	36,6	181,5	72,8
59	96,6	169,3	45,7	27,1	3,4	17,2	191,7	57,2
60	70,8	121,0	124,7	16,3	3,5	10,6	124,4	83,0
61	120,5	209,6	73,5	36,8	4,9	47,6	273,5	162,9
62	100,2	233,1	119,8	32,6	4,8	27,4	232,3	152,0
63	68,2	189,8	95,3	36,3	4,4	48,2	238,9	122,1
64	68,5	149,7	85,5	27,2	4,2	13,3	166,3	78,5
65	106,5	370,7	192,9	64,8	12,1	69,9	428,4	162,4
66	203,1	309,6	117,7	53,3	8,4	34,8	294,7	136,9
67	119,5	296,0	114,3	59,9	8,7	44,9	349,7	173,8
68	84,6	356,8	112,8	37,0	6,3	17,3	196,0	106,4
69	112,5	407,0	170,0	78,3	8,2	102,8	468,3	152,6
70	262,9	336,7	103,0	72,2	11,3	73,6	368,0	319,0
71	176,2	324,7	96,3	88,3	10,5	102,4	433,8	357,5
72	128,1	318,7	212,1	57,8	11,0	43,3	268,4	104,2

Tabela 7.11: Tempos de processamento da literatura e dos algoritmos propostos para instâncias com 32 locações

Inst.	<i>Lit.</i>	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
73	130,4	402,9	135,0	72,6	6,1	41,7	548,8	134,8
74	106,2	383,6	193,6	70,1	9,0	56,4	488,0	326,4
75	116,4	406,3	300,7	90,3	22,2	44,7	1137,3	230,9
76	80,4	307,1	144,2	61,8	9,7	30,3	733,5	119,2
77	97,4	445,3	241,7	75,7	6,4	44,7	1251,0	140,3
78	163,2	458,5	668,9	79,1	13,2	73,3	571,7	420,5
79	117,7	443,9	142,0	109,3	17,7	101,2	946,8	244,4
80	87,7	334,7	133,6	81,8	9,1	29,2	884,3	129,0
81	266,8	1114,8	472,5	267,1	37,5	451,4	1353,0	573,7
82	240,8	817,7	738,4	231,4	43,4	278,2	758,8	436,7
83	357,4	1356,2	783,8	295,1	38,0	198,3	799,1	562,7
84	247,8	885,0	339,4	192,6	13,1	162,4	441,0	790,3
85	343,9	1131,8	516,1	286,9	37,4	260,4	857,0	513,6
86	540,5	1010,3	528,9	251,1	31,5	248,7	823,4	604,8
87	461,8	890,1	234,4	337,5	48,1	451,0	828,7	802,8
88	342,3	910,0	626,0	246,3	29,3	162,9	527,8	235,1
89	447,5	1679,3	1303,0	649,5	115,2	505,1	1788,6	706,9
90	696,7	1782,5	1214,5	602,0	25,1	540,7	1470,7	499,7
91	400,7	1632,3	901,4	686,0	69,2	302,7	1544,9	636,4
92	1458,4	2316,7	728,0	423,3	41,8	207,4	692,4	350,7
93	770,3	2003,1	514,3	714,4	84,6	649,5	1843,7	821,1
94	887,0	1464,6	428,6	612,9	39,1	265,9	1347,8	627,4
95	717,2	1522,8	568,5	846,9	85,9	350,7	1632,7	709,1
96	663,6	1588,7	368,1	535,2	63,6	186,2	1049,0	459,3

Faz-se agora um resumo das comparações de resultados para as instâncias que têm ótimo desconhecido, na Tabela 7.12 que mostra o desvio médio da melhor solução reportada na literatura, o número de melhores soluções e os tempos de processamento dos algoritmos da literatura e propostos. Para efeito de comparação de tempos é importante informar que o algoritmo da literatura, em média, computa as instâncias no qual o ótimo não é conhecido de 12 locações em 44,6 segundos, as instâncias de 20 locações, em 88,0 segundos e as de 32 locações, em 405,9 segundos. Outra constatação a ser reforçada através desta tabela é que os algoritmos BTP, hILS e hVNS são capazes de obter melhores custos em todas as instâncias com 20 locações e hGRASP inclui-se nesta lista para as instâncias com 32 locações.

Tabela 7.12: Comparação dos algoritmos da literatura e propostos para instâncias com 12, 20 e 32 locações

12 Locações			
Alg.	Desvio(%)	<i>#Best</i>	Tempo
VNS	4,0	0	37,5
ILS	3,2	3	32,2
GRASP	1,5	6	4,4
BTP	1,5	6	0,2
hILS	1,5	6	4,1
hVNS	1,5	6	45,2
hGRASP	1,5	6	14,4
20 Locações			
Alg.	Desvio(%)	<i>#Best</i>	Tempo
VNS	-3,3	22	193,2
ILS	-3,9	21	88,3
GRASP	-5,1	23	33,5
BTP	-5,6	24	4,7
hILS	-5,8	24	32,2
hVNS	-5,9	24	215,3
hGRASP	-5,8	23	108,4
32 Locações			
Alg.	Desvio(%)	<i>#Best</i>	Tempo
VNS	0,3	13	1053,2
ILS	-1,1	16	509,4
GRASP	-3,5	23	325,4
BTP	-5,8	24	37,0
hILS	-6,8	24	235,1
hVNS	-7,1	24	1013,3
hGRASP	-6,7	24	461,0

Para encerrar esta seção é apresentado na Tabela 7.13 um sumário incluindo todas as instâncias, da qual fazem parte os desvios médios da melhor solução, números de custos ótimos quando este é conhecido e número de soluções com custo igual ou menor que a literatura, para aquelas instâncias nas quais o ótimo não foi obtido e o tempo de processamento médio dos algoritmos.

Tabela 7.13: Sumário de desempenho dos algoritmos propostos

Alg.	Desvio(%)	<i>#Best</i>	Tempo
VNS	-0,2	66	319,2
ILS	-0,7	73	155,0
GRASP	-1,7	80	90,4
BTP	-2,4	82	10,5
hILS	-2,7	82	67,6
hVNS	-2,8	82	315,9
hGRASP	-2,9	87	144,8

7.2 Comparação de resultados entre os algoritmos propostos

Nesta seção são realizadas duas baterias de testes para a avaliação e comparação dos algoritmos propostos partindo do princípio que as metaheurísticas utilizadas, assim como, as heurísticas propostas baseadas nestas, são de naturezas diferentes e, assim sendo, elas trabalham normalmente com critérios de paradas diferentes entre si e, mesmo durante uma iteração, cada algoritmo tem seus passos próprios e um número diferente deles. A fim de dar a cada algoritmo a mesma chance para mostrar seu desempenho a Seção 7.2.1 considera tempos fixos para execução dos algoritmos e a Seção 7.2.2, os tempos que os algoritmos necessitam até atingir uma solução de custo estabelecido.

7.2.1 Análise com tempos estabelecidos

Nos testes apresentados a seguir são utilizados dois tempos, em segundos, um considerado tempo mínimo e o outro, tempo máximo de execução. Os custos obtidos em 10 execuções com diferentes sementes randômicas são computados e são avaliados os seguintes itens: custo obtido com o tempo mínimo (máximo), desvio da solução obtida com o tempo mínimo (máximo) para a melhor solução conhecida e número de melhores soluções obtidas com o tempo mínimo (máximo), por conjuntos de 24 instâncias com 6, 12, 20 e 32 locações. A definição dos tempos foi feita da seguinte maneira: como tempo mínimo, o tempo médio de execução dos algoritmos considerados e, como tempo máximo, o dobro do tempo mínimo. Estes, são apresentados na Tabela 7.14.

Tabela 7.14: Tempo de execução (em segundos) mínimo e máximo de acordo com grupos de instâncias

# Loc.	Mínimo (<i>min</i>)	Máximo (<i>max</i>)
6	1	2
12	13	26
20	94	188
32	300	600

A Tabela 7.15 apresenta um sumário do desempenho dos algoritmos considerando tempo mínimo (*min*) e máximo (*max*) em comparação à melhor solução conhecida.

Tabela 7.15: Melhoria de custos dos algoritmos para tempos *min* e *max*

6 Locações				
Alg.	<i>Min</i>		<i>Max</i>	
	Desvio(%)	<i>#Best</i>	Desvio(%)	<i>#Best</i>
VNS	1,00	16	1,00	16
ILS	2,52	13	0,52	19
GRASP	1,14	17	1,14	17
BTP	1,14	17	1,14	17
hILS	0,81	19	0,81	19
hVNS	1,14	17	1,14	17
hGRASP	0,14	23	0,14	23
12 Locações				
Alg.	<i>Min</i>		<i>Max</i>	
	Desvio(%)	<i>#Best</i>	Desvio(%)	<i>#Best</i>
VNS	11,60	3	11,60	3
ILS	15,32	0	8,96	1
GRASP	0,79	17	0,79	17
BTP	0,79	17	0,79	17
hILS	0,79	17	0,79	17
hVNS	0,79	17	0,79	17
hGRASP	0,79	17	0,79	17
20 Locações				
Alg.	<i>Min</i>		<i>Max</i>	
	Desvio(%)	<i>#Best</i>	Desvio(%)	<i>#Best</i>
VNS	8,04	2	8,00	2
ILS	8,71	2	5,07	4
GRASP	-4,70	22	-5,00	22
BTP	-5,90	24	-5,90	24
hILS	-5,76	24	-5,76	24
hVNS	-5,88	23	-5,92	24
hGRASP	-5,84	23	-6,00	24
32 Locações				
Alg.	<i>Min</i>		<i>Max</i>	
	Desvio(%)	<i>#Best</i>	Desvio(%)	<i>#Best</i>
VNS	17,82	0	16,89	0
ILS	18,61	0	16,61	0
GRASP	-2,26	16	-2,82	17
BTP	-6,70	24	-6,98	24
hILS	-6,65	24	-6,91	24
hVNS	-6,65	24	-6,87	24
hGRASP	-6,29	24	-6,47	24

Para as instâncias com 6 locações, no tempo mínimo, o melhor desempenho é atribuído ao algoritmo BTP. Este algoritmo no estudo anterior mostrou que com pouco tempo de processamento é capaz de alcançar algumas soluções de boa qualidade, mas o que constata-se agora é que o aumento de tempo, pelo menos para este tempo máximo estabelecido, não necessariamente indica maior ganho. Os resultados obtidos pelos algoritmos ILS, GRASP, hVNS e hGRASP comparando-se os resultados obtidos com os tempos mínimo e máximo, apontam para melhorias. Este último é capaz de obter quase a totalidade de custos melhores, no tempo máximo.

Nas instâncias com 12 locações, o tempo mínimo estabelecido parece pouco para os algoritmos VNS, ILS, GRASP e BTP. Já no tempo máximo, os resultados destes equiparam-se aos demais. Para os métodos híbridos, a mudança de tempos não surtiu efeito. Estes resultados fizeram com que uma dúvida surgisse: será que o tempo de processamento estabelecido como máximo para estas instâncias não foi suficiente para gerar soluções de melhor qualidade? A resposta foi encontrada após a realização de mais uma bateria de testes com tempo máximo estabelecido em 39 segundos e obtemos os seguintes resultados, apresentados na Tabela 7.16.

Tabela 7.16: Sumário do desempenho dos algoritmos para tempo *min* e *max* e 39 segundos em comparação com a literatura

12 Locações						
Alg.	<i>Min</i>		<i>Max</i>		39seg.	
	Desvio(%)	# <i>Best</i>	Desvio(%)	# <i>Best</i>	Desvio(%)	# <i>Best</i>
VNS	11,60	3	11,60	3	11,48	3
ILS	15,32	0	8,96	1	0,39	17
GRASP	0,79	17	0,79	17	0,79	17
BTP	0,79	17	0,79	17	0,70	17
hILS	0,79	17	0,79	17	0,72	17
hVNS	0,79	17	0,79	17	0,79	17
hGRASP	0,79	17	0,79	17	0,79	17

Pode-se observar que o aumento de tempo, a não ser no caso do ILS, não ocasiona uma diminuição relevante nos desvios. Então, o que parece é que os algoritmos, neste conjunto de instâncias, ficam presos à ótimos locais, em média, a aproximadamente 0,8% de uma solução de melhor custo ou exato, quando este é conhecido.

Nas instâncias com 20 locações é que se começa a ter melhorias de custos em relação à melhor solução conhecida. Já no tempo mínimo os algoritmos BTP e hILS podem alcançar custos menores ou iguais do que a literatura em todo o conjunto de instâncias. E no tempo máximo, hVNS e hGRASP são capazes de igualar este desempenho.

Em se tratando das instâncias com 32 instâncias, com o tempo mínimo e máximo, VNS e ILS, são incapazes de conseguir soluções de custo menor ou igual ao melhor conhecido em todo conjunto. Excetuando-se estes e GRASP, os algoritmos podem obter melhoria de custos em todas as 24 instâncias.

A Tabela 7.17 apresenta, considerando as 96 instâncias, as médias de desvios ($D_Min(\%)$ e $D_Max(\%)$) e o número de melhores soluções (B_Min e B_Max) dos algoritmos testados no tempo mínimo e máximo estabelecidos em relação ao melhor custo conhecido.

Tabela 7.17: Média de desvio e de número de *Best* obtidos no tempo *min* e *max*

Alg.	<i>D_Min</i> (%)	<i>B_Min</i>	<i>D_Max</i> (%)	<i>B_Max</i>
VNS	9,62	21	9,37	21
ILS	11,29	15	7,79	24
GRASP	-1,25	72	-1,47	73
BTP	-2,67	82	-2,75	82
hILS	-2,70	84	-2,79	84
hVNS	-2,65	81	-2,71	82
hGRASP	-2,80	87	-2,88	88

No desempenho geral, ganham os algoritmos BTP e os híbridos hILS, hVNS e hGRASP para os tempos estabelecidos. Das 96 instâncias testadas, estes algoritmos são capazes de obter solução de custo melhor ou igual ao melhor conhecido em 82, 84, 81 e 87, quando executados com o tempo mínimo e, em 82, 84, 82 e 88, respectivamente, executados até o limite do tempo máximo. Já foi dito que, para as instâncias de 6, 20 e 32 locações, estes algoritmos encontram soluções de melhor custo ou igual ao melhor conhecido em quase sua totalidade, mas não são capazes de manter este desempenho para as instâncias de 12 locações. Em se tratando deste grupo de instâncias em particular, os demais algoritmos tiveram também uma performance prejudicada.

Das 96 instâncias testadas com o tempo mínimo, VNS, ILS e GRASP são melhores ou iguais à literatura em 21, 15 e 72 instâncias, respectivamente. Quando a mesma análise é feita com o tempo máximo de execução, GRASP e ILS podem igualar-se aos melhores custos ou mantê-los em mais uma instância cada um.

7.2.2 Análise Probabilística

Neste tipo de experimento, para cada instância os algoritmos propostos são executados 100 vezes usando sementes distintas para a geração de números aleatórios. O critério de parada neste caso é um valor alvo, ou seja, o algoritmo é encerrado quando encontra uma solução de valor igual ou menor do que um alvo estabelecido.

Alguns algoritmos tiveram que sofrer adaptações para este novo critério de parada, como as heurísticas híbridas hILS, hVNS e hGRASP. Nestes três casos, o algoritmo guia (ILS, VNS e GRASP) têm seu critério de parada alterado para alvo e, o método BTP que é chamado dentro desses demanda um critério de parada. O número de iterações da BTP é o mesmo que já vinha sendo usado nos testes anteriores. Portanto, o a cada iteração de hILS, hVNS e hGRASP são feitas 50, 100 e 50 iterações sem melhora de BTP. Todos os outros parâmetros de cada método foram preservados.

Em cada execução, o tempo de processamento até que o valor alvo seja alcançado é armazenado. Os tempos são dispostos em ordem crescente e uma probabilidade

$p_i = (i - 0,5)/100$ é associada a cada i -ésimo tempo t_i . Com isto, os pontos $z_i = (t_i, p_i)$ são plotados, estabelecendo uma distribuição de probabilidade empírica de tempo para que um algoritmo possa alcançar um valor alvo determinado, como proposto em [48].

São escolhidas 6 instâncias para participar desta análise, seguindo o seguinte critério: que os custos encontrados pelos algoritmos para a instância não fossem muito parecidos, a fim de não prejudicar a análise. Para cada uma delas são estabelecidos dois alvos baseados nas soluções obtidas nos testes anteriores (Tabelas 7.6 e 7.7): Alvo_1, que correspondente ao valor da pior solução encontrada na execução de todos os algoritmos, incluindo os resultados da literatura e Alvo_2, que é o maior valor entre a média dos custos de soluções encontrados e 5% da melhor solução para a instância em questão. Para cada alvo tem-se um mais difícil do que o outro. Os alvos são apresentados na Tabela 7.18 e os resultados, nas Figuras 7.1 a 7.12. É apresentado, para cada instância testada, primeiramente o alvo mais fácil e, em seguida, o mais difícil.

Tabela 7.18: Alvos usados no teste probabilístico

Inst.	Alvo_1	Alvo_2
33	57	55
34	76	74
37	50	49
52	98	92
54	67	65
56	97	93

A probabilidade de que um algoritmo encontre uma solução de valor menor ou igual a determinado valor alvo em um dado tempo de processamento aumenta da esquerda para a direita, ou seja, nestas figuras quanto mais à esquerda estiver a curva associada a um algoritmo, melhor será seu desempenho.

De acordo com os experimentos realizados, os algoritmos propostos se mostram na seguinte ordem de desempenho: **BTP**, **hILS** e **hVNS**, com probabilidade cumulativa muito próxima. Em seguida, **hGRASP**. Os métodos **GRASP** e **ILS** nem sempre são capazes de encontrar soluções com os alvos estabelecidos, por isto em alguns gráficos, suas curvas não aparecem e **VNS**, nos testes realizados neste presente momento, é incapaz de obter os 100 alvos estabelecidos para as instâncias escolhidas, mesmo após diversas tentativas com limite de tempo estabelecido, primeiramente em 3600 segundos e depois estendido para 10000 segundos. Por este motivo, a curva do **VNS** não aparece nos gráficos.

Nas Figuras 7.1 e 7.2 o primeiro gráfico tem o alvo mais fácil que o segundo. Para o Alvo_1, entre 0,01 e 0,1 segundos, os algoritmos **BTP**, **hILS** e **hVNS** alcançam probabilidades altas (próximas de 100%) de chegar ao alvo. Depois de 1 segundo de execução os algoritmos **GRASP**, **ILS** e **hGRASP**, têm aproximadamente 40%, 10% e 4% de chance de chegar ao alvo determinado, respectivamente. A 10 segundos de

processamento, essa probabilidade cresce para 100%, 63% e 58%, respectivamente. O algoritmo mais lento neste caso é o ILS que necessita aproximadamente 100 segundos para atingir 80% de chance de alcançar o valor alvo. No Alvo_2, além do VNS, o algoritmo ILS também não pôde chegar ao alvo por diversas vezes, por isso sua curva não foi plotada. Quanto aos demais algoritmos, é necessário mais tempo até que se atinja o alvo. Ainda sim, com pouco mais de 0,1 segundo, os algoritmos BTP, hILS e hVNS podem convergir totalmente ao valor alvo. Aos 10 segundos de processamento, os algoritmos GRASP e hGRASP alcançam, aproximadamente, 39% e 10% de chances de atingir o alvo.

Considerando a instância 34, do alvo 76 para o 74 (Figuras 7.3 e 7.4), apreciamos um comportamento bastante similar. Os algoritmos BTP, hILS e hVNS mantêm o mesmo desempenho do caso anterior podendo convergir mais rapidamente para o alvo. Para o alvo 76, o algoritmo GRASP, durante quase toda a curva, se sustenta mais a esquerda de hGRASP e os dois, em aproximadamente 1 segundo, são capazes de alcançar 100% de chance de obter o alvo. Para o segundo alvo, o algoritmo hGRASP é capaz de manter probabilidades de chegar ao alvo com menos tempo que o algoritmo GRASP, este por sinal, é capaz de alcançar uma solução alvo somente depois de passado aproximadamente 1 segundo de processamento. Os dois algoritmos, no entanto, convergem a 100% em momentos bem próximos, em aproximadamente 100 segundos de processamento.

A diferença de comportamento é um pouco acentuada no caso da instância 52 para os alvos 98 e 92. No primeiro alvo, os algoritmos BTP, hILS e hVNS continuam tendo desempenho superior. O interessante é ver que é preciso menos de 10 segundos para que o algoritmo hGRASP alcance 100% de chance de chegar ao valor alvo, mantendo-se na maior parte mais à esquerda de hVNS. Ao passar para o segundo alvo, o algoritmo GRASP não teve sua curva plotada em função de não ter chegado ao alvo dentro do tempo limite e o algoritmo ILS atinge aproximadamente 90% de chance de chegar ao alvo aos 1000 segundos, antes, no alvo 98, este pôde chegar à mesma probabilidade em aproximadamente 10 segundos. Uma tendência parecida pode ser vista nos dois alvos estabelecidos nas instâncias 54 e 56.

Figura 7.1: Comparação entre algoritmos propostos para a instância 33 com alvo 57

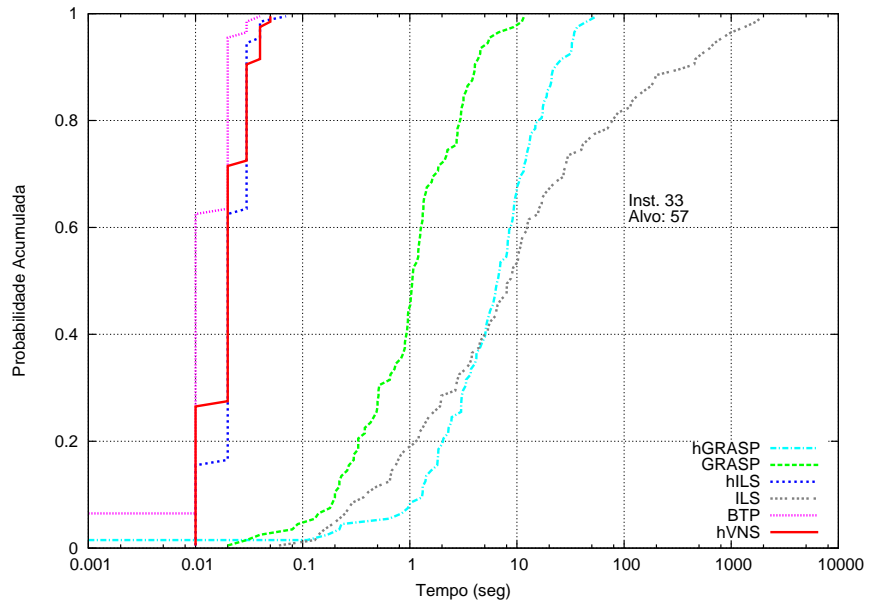


Figura 7.2: Comparação entre algoritmos propostos para a instância 33 com alvo 55

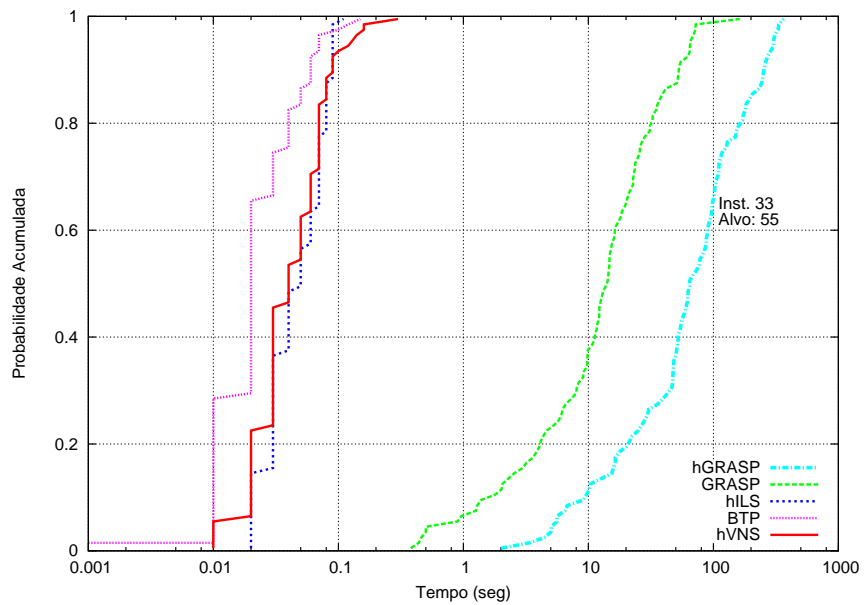


Figura 7.3: Comparação entre algoritmos propostos para a instância 34 com alvo 76

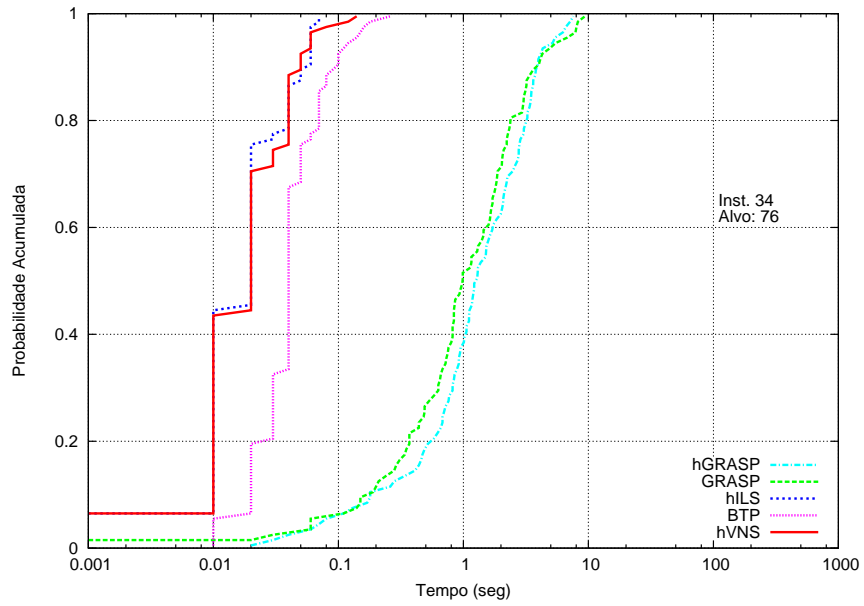


Figura 7.4: Comparação entre algoritmos propostos para a instância 34 com alvo 74

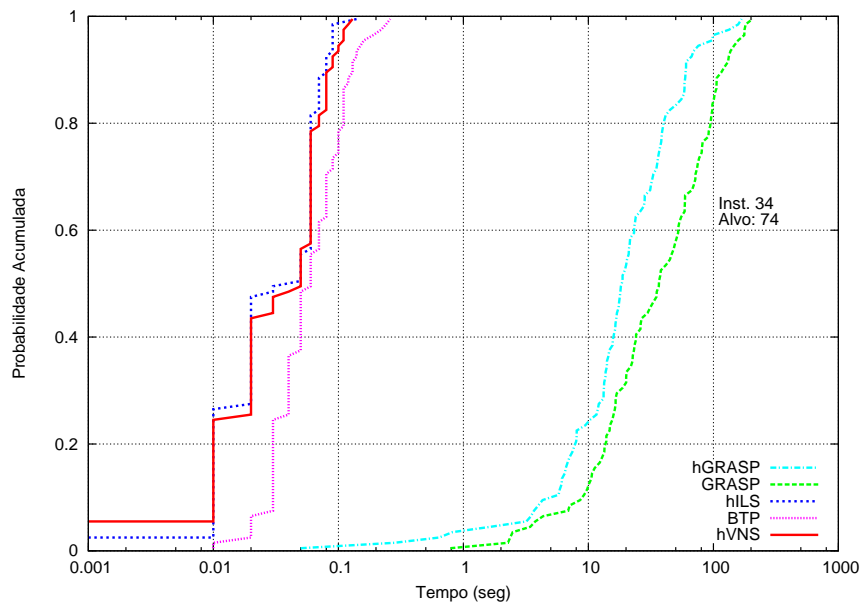


Figura 7.5: Comparação entre algoritmos propostos para a instância 37 com alvo 50

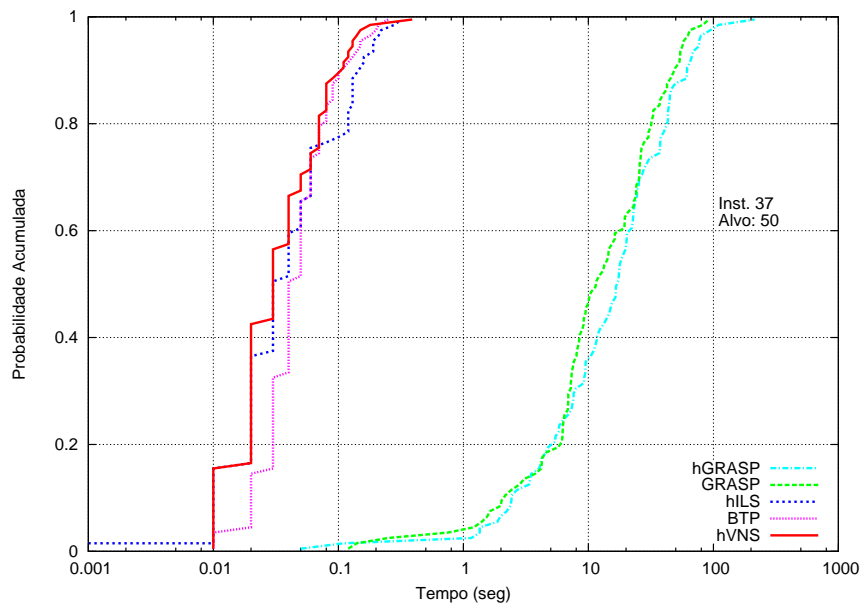


Figura 7.6: Comparação entre algoritmos propostos para a instância 37 com alvo 49

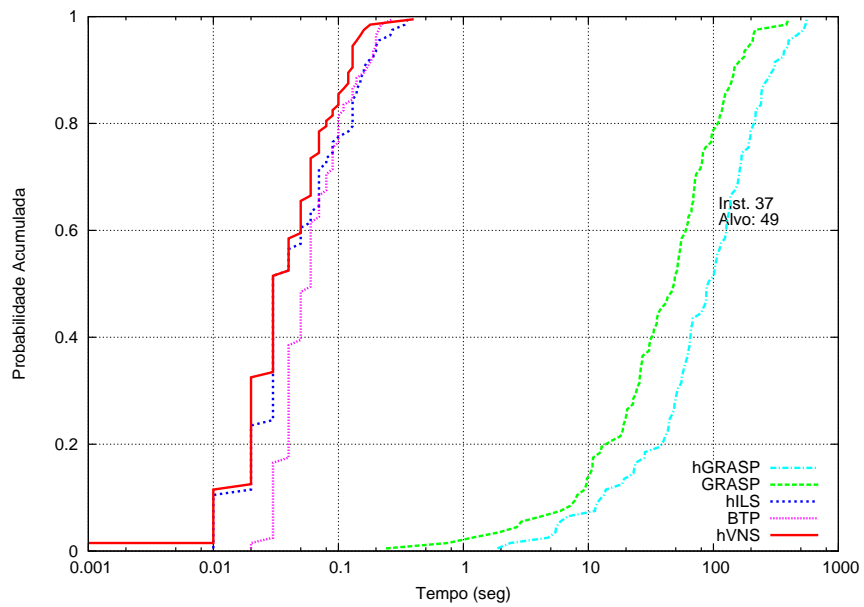


Figura 7.7: Comparação entre algoritmos propostos para a instância 52 com alvo 98

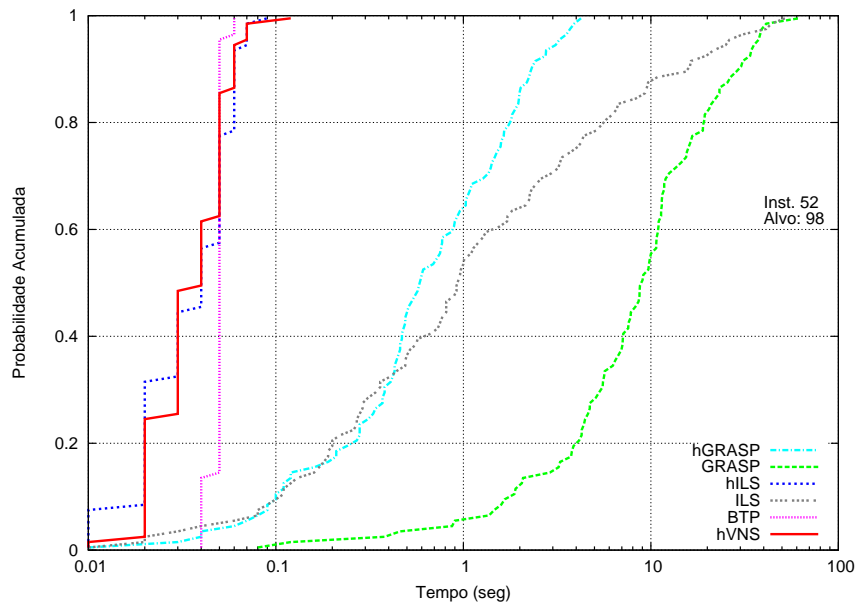


Figura 7.8: Comparação entre algoritmos propostos para a instância 52 com alvo 92

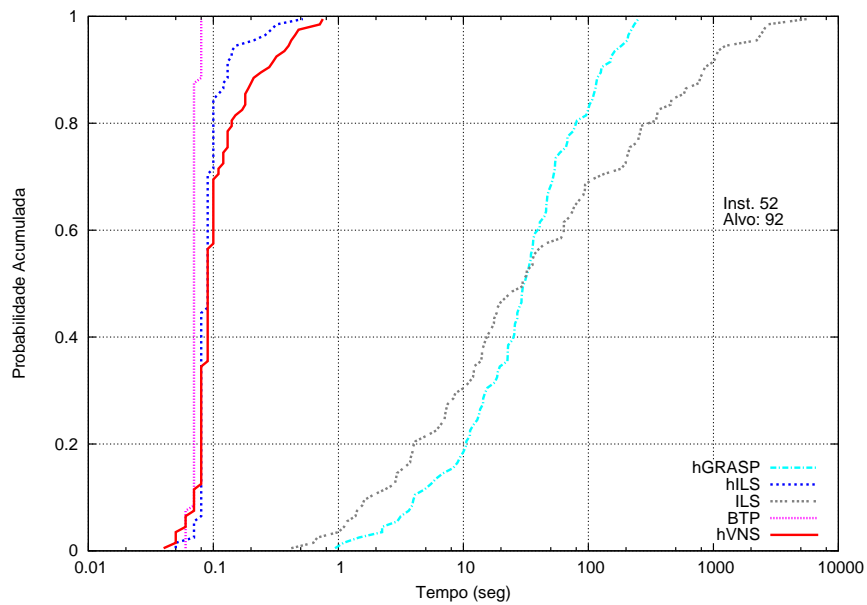


Figura 7.9: Comparação entre algoritmos propostos para a instância 54 com alvo 67

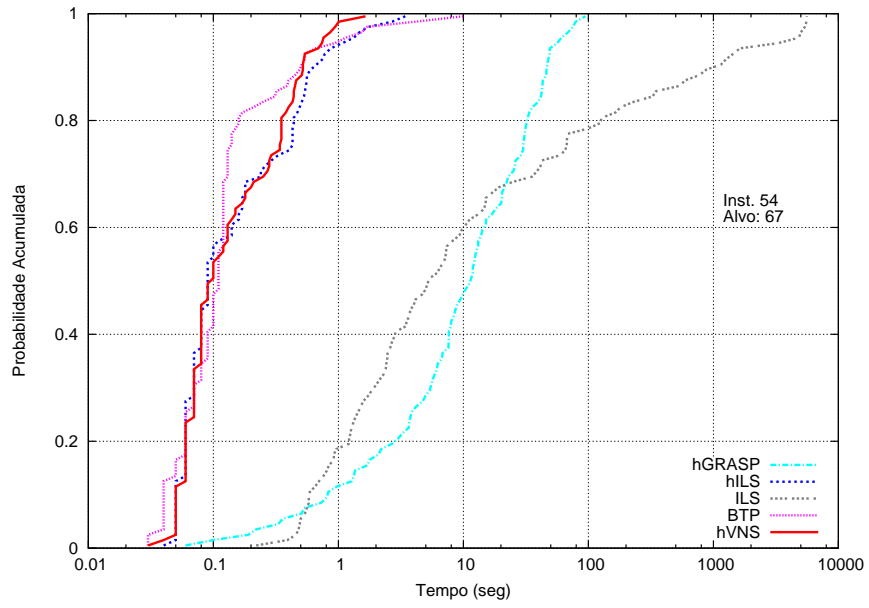


Figura 7.10: Comparação entre algoritmos propostos para a instância 54 com alvo 65

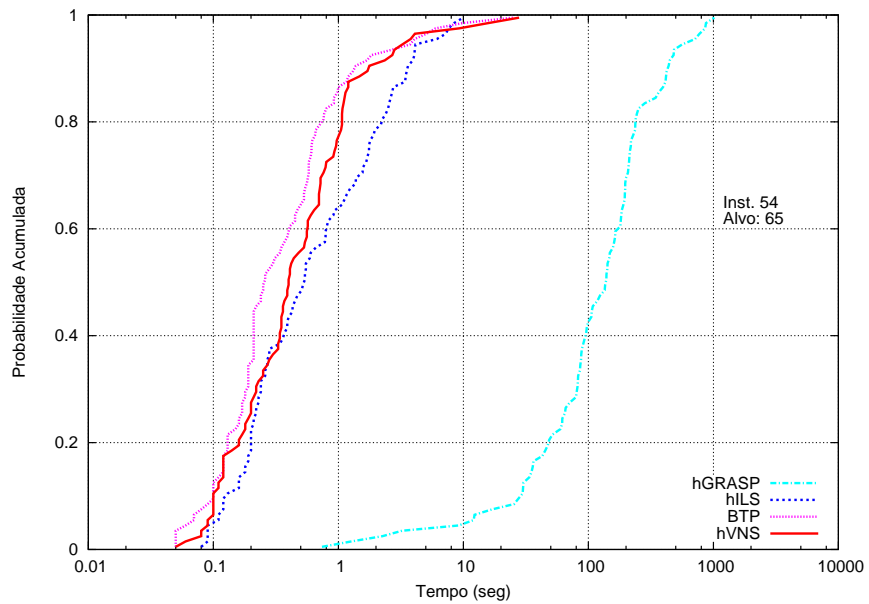


Figura 7.11: Comparação entre algoritmos propostos para a instância 56 com alvo 97

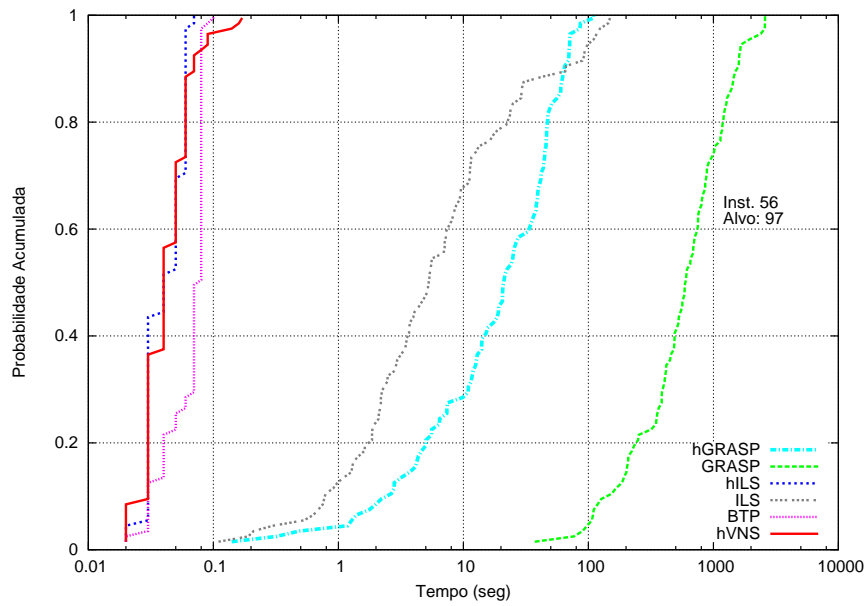
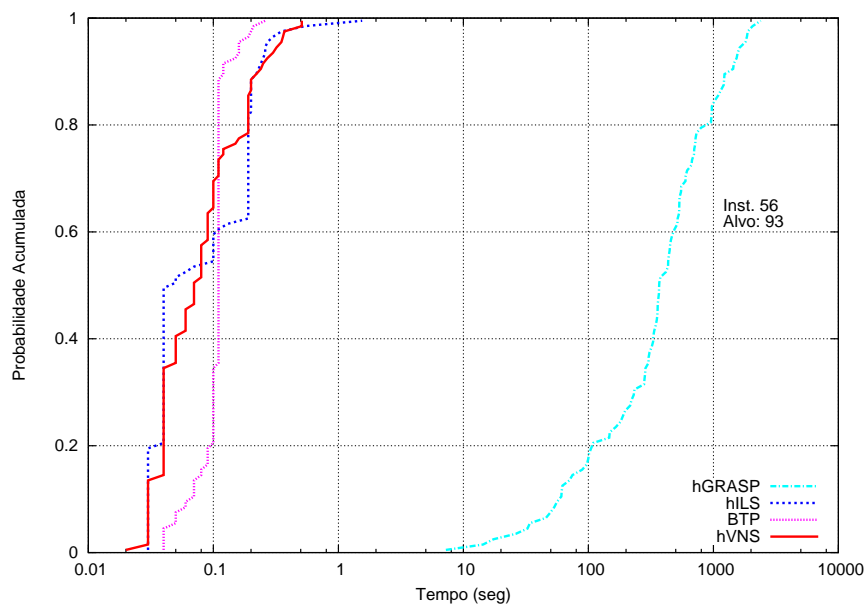


Figura 7.12: Comparação entre algoritmos propostos para a instância 56 com alvo 93



A Tabela 7.19 apresenta o tempo de execução (segundos) necessário para que os

algoritmos propostos atinjam diferentes probabilidades de alcançar os valores alvo nestes experimentos. Os tempos são baseados no alvo 2.

Tabela 7.19: Tempo (em segundos) para que 50%, 75% e 90% das execuções atinjam o valor alvo

Alg.	%	Instância/Alvo					
		33/55	34/74	37/49	52/92	54/65	56/93
ILS	50	?	?	?	29,77	?	?
	75	?	?	?	106,97	?	?
	90	?	?	?	823,06	?	?
GRASP	50	13,78	36,59	47,66	?	?	?
	75	25,73	80,63	82,36	?	?	?
	90	52,53	115,54	145,99	?	?	?
BTP	50	0,02	0,06	0,06	0,07	0,25	0,11
	75	0,03	0,1	0,09	0,07	0,61	0,11
	90	0,06	0,13	0,16	0,08	1,31	0,12
hILS	50	0,05	0,03	0,03	0,09	0,52	0,04
	75	0,07	0,06	0,09	0,09	1,76	0,19
	90	0,09	0,08	0,16	0,13	3,52	0,22
hVNS	50	0,04	0,05	0,03	0,09	0,40	0,07
	75	0,03	0,06	0,07	0,1	0,93	0,12
	90	0,09	0,08	0,12	0,24	1,71	0,22
hGRASP	50	64,27	18,19	88,41	29,95	136,26	367,43
	75	117,25	36,7	168,59	51,76	214,94	700,17
	90	251,78	59,96	290,6	125,71	423,65	1226,04

Na tabela, o símbolo ? indica que o alvo poderia ter sido alcançado em um valor acima de 10000 segundos ou de fato não seria alcançado. Devemos ressaltar que BTP, hILS e hVNS atingem valores altos de probabilidade com tempo baixo de processamento. Considera-se ainda que hGRASP apesar de ser mais lento em sua execução teve a sua robustez demonstrada, atingindo todos os valores alvo propostos para cada instância.

Capítulo 8

Conclusões e Direções Futuras

Neste trabalho foi estudado o Problema da Alocação Dinâmica de Espaços, um problema recentemente definido na literatura e pouco estudado. No decorrer deste, foram descritas suas principais características, o que já nos faz perceber se tratar de um problema difícil, quanto mais quando se mostra correlacionado ao Problema Quadrático de Alocação (PQA), ao Problema Dinâmico de *Layout* de Facilidades (PDLF) e ao Problema Quadrático de Alocação Generalizado. Foi apresentado o estado-da-arte do tema e detalhados os principais algoritmos existentes na literatura.

Acredita-se que os resultados apresentados durante o trabalho mostram que os objetivos deste foram em parte alcançados: obter bons limites superiores para o problema. Foram apresentadas em três capítulos metodologias para buscar soluções viáveis para o problema. Trabalhando-se com o CPLEX foi possível obter soluções exatas para seis problemas em aberto na literatura e baixos *gaps* para duas instâncias. Como ponto alto do trabalho, foi possível corrigir os limites inferiores das instâncias 25, 26, 28 e 32, apresentados em [19].

As propostas de heurísticas baseadas em nas metaheurísticas VNS, ILS, GRASP e Busca Tabu originaram, através das extensas baterias de testes, um estudo sistemático, que serviu para que cada método chegasse a bons resultados para o conjunto de instâncias testado. Além disso, foi mostrado que, para estas instâncias, a combinação destes métodos entre si constitui-se uma proposta promissora. A partir destes resultados, acreditamos ter chegado a métodos eficientes para o tratamento do problema, o que pôde ser provado em diversas comparações. Dentre estes métodos podemos escolher de acordo com um determinado interesse, que poderia versar sobre tempo e/ou qualidade de soluções. Se a escolha for sobre qualidade de soluções seria indicado os métodos híbridos hVNS ou hGRASP. Já se a questão é tempo de processamento aliado à qualidade de soluções seria apropriado o uso dos métodos GRASP, BTP e hILS.

A análise probabilística foi muito útil para avaliar a robustez dos métodos. Pode-

se inferir que os algoritmos propostos BTP, hILS, hVNS e hILS se portaram de maneira semelhante mesmo com a variação de instâncias e do valor alvo. Como ocorreu que, nos testes realizados, estas heurísticas propostas chegaram sempre aos valores alvo estabelecidos, pode-se supor que elas possuem uma característica desejável em heurísticas: robustez. O mesmo, no entanto, de acordo com as instâncias testadas e os alvos estabelecidos, não se pode dizer dos algoritmos VNS, ILS e GRASP.

Como proposta para trabalhos futuros consideramos um caminho promissor a interação entre as boas metaheurísticas aqui propostas e o CPLEX, em um processo similar ao *local branching*. Acreditamos também que é necessário tornar o modelo matemático da literatura mais próximo do real no contexto da Engenharia de Produção e, como consequência disto, apresentação de novas instâncias e algoritmos para o problema.

Referências Bibliográficas

- [1] KOOPMANS, T. C., BECKMANN, M. “Assignment problems and the location of economic activities”, *Econometrica*, v. 25, pp. 53–76, 1957.
- [2] TAILLARD, E. “Robust taboo search for the quadratic assignment problem”, *Parallel Computing*, v. 17, pp. 443–455, 1991.
- [3] LI, Y., PARDALOS, P. M., RESENDE, M. G. C. “A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem”. In: Pardalos, P. M., Wolkowicz, H. (Eds.), *Quadratic assignment and related problems*, v. 16, *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society, pp. 237–261, Providence, USA, 1994. Disponível em: <<http://www.research.att.com/~mgcr/doc/grpqap.ps.Z>>.
- [4] MANIEZZO, V., COLORNI, A. “The ant system applied to the quadratic assignment problem”, *Knowledge and Data Engineering, IEEE Transactions on*, v. 11, pp. 769–778, 1999. ISSN: 1041-4347. doi: 10.1109/69.806935.
- [5] OLIVEIRA, C., PARDALOS, P., RESENDE, M. “GRASP with path-relinking for the quadratic assignment problem”. In: Ribeiro, C., Martins, S. (Eds.), *Proceedings of III Workshop on Efficient and Experimental Algorithms (WEA2004)*, v. 3059, Springer, pp. 356–368, Angra dos Reis, Brasil, 2004.
- [6] LOIOLA, E. M., ABREU, N. M. M. A., BOAVENTURA-NETTO, P. O., et al. “A survey for the quadratic assignment problem”, *European Journal of Operational Research*, v. 176, pp. 657–690, 2007.
- [7] ROSENBLATT, M. J. “The dynamics of plant layout”, *Management Science*, v. 32, pp. 76–86, 1986.
- [8] URBAN, T. “Solution procedures for the dynamic facility layout problem”, *Annals of Operations Research*, pp. 323–342, 1998.

- [9] ZOUEIN, P. P., TOMMELEIN, I. D. “Dynamic layout planning using a hybrid incremental solution method”, *Journal of Construction Engineering and Management*, pp. 400–408, 1999.
- [10] LACKSONEN, T. A., ENSCORE, E. “Quadratic assignment algorithms for the dynamic layout problem”, *International Journal of Production Research*, v. 31, pp. 503–517, 1993.
- [11] DUNKER, T., RADONS, G., WESTKAMPER, E. “Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem”, *European Journal of Operational Research*, pp. 55–69, 2005.
- [12] MCKENDALL JR., A. R., SHANG, J., KUPPUSAMY, S. “Simulated annealing heuristics for the dynamic facility layout problem”, *Comput. Oper. Res.*, v. 33, n. 8, pp. 2431–2444, 2006. ISSN: 0305-0548. doi: <http://dx.doi.org/10.1016/j.cor.2005.02.021>.
- [13] LEE, C.-G., MA, Z. *The generalized quadratic assignment problem*. Relatório técnico, Department of Mechanical and Industrial Engineering, University of Toronto, Ontario, M5S 3G8, Canada, 2004.
- [14] CORDEAU, J.-F., GAUDIOSO, M., LAPORTE, G., et al. “A Memetic Heuristic for the Generalized Quadratic Assignment Problem”, *INFORMS J. on Computing*, v. 18, n. 4, pp. 433–443, 2006. ISSN: 1526-5528. doi: <http://dx.doi.org/10.1287/ijoc.1040.0128>.
- [15] HAHN, P. M., KIM, B.-J., GUIGNARD, M., et al. “An algorithm for the generalized quadratic assignment problem”, *Computational Optimization and Applications Publisher*, pp. doi:10.1007/s10589-007-9093-1, novembro 2007.
- [16] MATEUS, G. R., SILVA, R. M. A., RESENDE, M. *GRASP with path-relinking for the generalized quadratic assignment problem*. Relatório Técnico 00.7.2, AT & T Labs Research, 2008.
- [17] MCKENDALL JR., A. R., NOBLE, J. S., KLEIN, C. M. “Simulated annealing heuristics for managing resources during planned outages at electric power plants”, *Computers & Operations Research*, v. 32, pp. 107–125, 2005.
- [18] MCKENDALL JR., A. R., JARAMILLO, J. “A tabu search heuristic for the dynamic space allocation problem”, *Computers & Operations Research*, v. 33, pp. 768–789, 2006.

- [19] MCKENDALL JR., A. R. “Improved tabu search heuristics for the dynamic space allocation problem”, *Computers & Operations Research*, p. doi:10.1016/j.cor.2007.03.002, 2007.
- [20] CHAN, H. M., MILNER, D. A. “Direct clustering algorithm for group formation in cellular manufacture”, *Journal of Manufacturing Systems*, v. 1, pp. 165–175, 1982.
- [21] TOMPKINS, J. A., WHITE, J. A., BOZER, Y. A., et al. *Facilities Planning*. New Jersey, John Wiley, 1996.
- [22] BLUM, C., ROLI, A. “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”, *ACM Comput. Surv.*, v. 35, n. 3, pp. 268–308, 2003. ISSN: 0360-0300. doi: <http://doi.acm.org/10.1145/937503.937505>.
- [23] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. “Optimization by Simulated Annealing”, *Science*, v. 220, pp. 671–680, 1983. Disponível em: <http://citeseer.ist.psu.edu/kirkpatrick83optimization.html>.
- [24] GLOVER, F. “Tabu Search: A Tutorial”, *Interfaces*, v. 20, pp. 74–94, 1990.
- [25] MLADENOVIC, N., HANSEN, P. “Variable neighborhood search”, *Comput. Oper. Res.*, v. 24, n. 11, pp. 1097–1100, 1997. ISSN: 0305-0548. doi: [http://dx.doi.org/10.1016/S0305-0548\(97\)00031-2](http://dx.doi.org/10.1016/S0305-0548(97)00031-2).
- [26] CHIANG, W.-C., KOUVELIS, P. “An improved tabu search heuristic for solving facility layout design problems”, *International Journal of Production*, v. 34, pp. 2566–2585, 1996.
- [27] SHORIN-KAPOV, J. “Tabu search applied to the quadratic assignment problem formulation”, *ORSA Journal on Computing*, pp. 443–455, 1990.
- [28] CHIANG, W.-C., KOUVELIS, P. “An improved tabu search strategies for solving facility layout problem with quadratic assignment problem formulation”, *European Journal of Operational Research*, pp. 457–488, 1998.
- [29] TRINDADE, A., OCHI, L. “Um Algoritmo Evolutivo Híbrido para a Formação de Células de Manufatura em Sistemas de Produção”, *Pesquisa Operacional*, v. 26, pp. 255–294, 2006.
- [30] HOOS, H., STTZLE, T. *Stochastic Local Search: Foundations & Applications*. San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 2004. ISBN: 1558608729.

- [31] HANSEN, P., MLADENOVIĆ, N. *A Tutorial on Variable Neighborhood Search*. Relatório Técnico G-2003-46, GERAD, 2003.
- [32] HANSEN, P., MLADENOVIĆ, N. “Variable Neighborhood Search”. In: Glover, F., Kochenberger, G. (Eds.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, pp. 145–184, Boston, 2003.
- [33] LOURENÇO, H. R., MARTIN, O., STÜTZLE, T. “Iterated Local Search”. In: Glover, F., Kochenberger, G. (Eds.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, pp. 321–353, Boston, 2003.
- [34] FEO, T., RESENDE, M. “Greedy randomized adaptive search procedures”, *J. of Global Optimization*, v. 6, pp. 109–133, 1995. Disponível em: <<http://www.research.att.com/~mgcr/doc/gtut.ps.Z>>.
- [35] OLIVEIRA, C., PARDALOS, P., RESENDE, M. “GRASP with path-relinking for the QAP”. In: Ibaraki, T., Yoshitomi, Y. (Eds.), *Proceedings of the Fifth Metaheuristics International Conference (MIC2003)*, pp. 57–1 – 57–6, Kyoto, Japan, 2003.
- [36] SILVA, G. C., OCHI, L. S., MARTINS, S. L. “Proposta e Avaliação de Heurísticas GRASP para o Problema da Diversidade Máxima”, *Pesquisa Operacional*, v. 26, pp. 321–360, 2006.
- [37] SILVA, G. C., QUINET, M., OCHI, L. S., et al. “New heuristics for the maximum diversity problem”, *Journal of Heuristics*, v. 13, pp. 315–336, 2007. doi: 10.1007/s10732-007-9010-x.
- [38] FESTA, P., RESENDE, M. “An annotated bibliography of GRASP, Part I: Algorithms”, *International Transactions in Operational Research*, v. 16, pp. 1–24, 2009.
- [39] FESTA, P., RESENDE, M. “An annotated bibliography of GRASP, Part II: Applications”, *International Transactions in Operational Research*, v. 16, pp. 131–172, 2009.
- [40] RESENDE, M. “Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures”. In: Chen, Z.-L., Raghavan, S. (Eds.), *TutORials in Operations Research*, INFORMS, pp. 295–319, Hanover, 2008.
- [41] PRAIS, M., RIBEIRO, C. C. “Reactive GRASP: An Application to a Matrix Decomposition Problem in TDMA Traffic Assignment”, *INFORMS Journal on Computing*, v. 12, n. 3, pp. 164–176, 2000. ISSN: 1526-5528. doi: <http://dx.doi.org/10.1287/ijoc.12.3.164.12639>.

- [42] GLOVER, F. “Future paths for integer programming and links to artificial intelligence”, *Comput. Oper. Res.*, v. 13, n. 5, pp. 533–549, May 1986. ISSN: 0305-0548. doi: 10.1016/0305-0548(86)90048-1. Disponível em: [http://dx.doi.org/10.1016/0305-0548\(86\)90048-1](http://dx.doi.org/10.1016/0305-0548(86)90048-1).
- [43] SALHI, S. “Defining tabu list size and aspiration criterion within tabu search methods”, *Computers & OR*, v. 29, n. 1, pp. 67–86, 2002.
- [44] GLOVER, F. “Tabu Search, Part I”, *ORSA Journal on Computing*, v. 1, n. 3, pp. 190–206, Summer 1989.
- [45] KNOX, J., GLOVER, F. *Comparative Testing of Traveling Salesman Heuristics Derived from Tabu Search, Genetic Algorithms and Simulated Annealing*. Relatório técnico, California State Polytechnic University, September 1989.
- [46] TSUBAKITANI, S., EVANS, J. R. “Optimizing tabu list size for the traveling salesman problem”, *Computers & OR*, v. 25, n. 2, pp. 91–97, 1998.
- [47] SEVAUX, M., SÖRENSEN, K. “VNS/TS for a parallel machine scheduling problem”. In: *Proceedings of the 18th Mini EURO Conference on Variable Neighborhood Search (VNS)*, 2005. Disponível em: <http://hal.archives-ouvertes.fr/hal-00019096/en/>.
- [48] AIEX, R., RESENDE, M., RIBEIRO, C. “Probability distribution of solution time in GRASP: An experimental investigation”, *J. of Heuristics*, v. 8, pp. 343–373, 2002.

Apêndice A

Resultados Computacionais dos Algoritmos VNS Propostos

Os algoritmos VNS propostos no Capítulo 5, Seção 5.1.1, foram testados e os resultados obtidos podem ser vistos nas tabelas a seguir.

As Tabelas A.1 e A.2 apresentam, respectivamente, os melhores custos obtidos com 10 execuções dos algoritmos VNS1, VNS2, VNS3, VNS4 e VNS5 com 500 e 700 iterações para as instâncias com 6 e 12 locações. Em negrito encontra-se destacado o melhor valor encontrado para cada instância.

A Tabela A.3 mostra os melhores custos obtidos com 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para as instâncias com 6 e 12 locações. Em negrito encontra-se destacado o melhor valor encontrado para cada instância.

As Tabelas A.4 e A.5 apresentam, respectivamente, os melhores custos obtidos com 10 execuções dos algoritmos VNS1, VNS2, VNS3, VNS4 e VNS5 com 500 e 700 iterações para as instâncias com 20 e 32 locações. Em negrito encontra-se destacado o melhor valor encontrado para cada instância.

A Tabela A.6 mostra os melhores custos obtidos com 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para as instâncias com 20 e 32 locações. Em negrito encontra-se destacado o melhor valor encontrado para cada instância.

As Tabelas A.7 e A.8 apresentam, respectivamente, os desvios percentuais entre o melhor custo conhecido e os melhores custos obtidos com 10 execuções dos algoritmos VNS1, VNS2, VNS3, VNS4 e VNS5 com 500 e 700 iterações para as instâncias com 6 e 12 locações. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo VNS1 é menor do que o da literatura.

A Tabela A.9 mostra os desvios percentuais entre o melhor custo da literatura e os melhores custos obtidos com 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para as instâncias com 6 e 12 locações. Em negrito encontra-se

destacado o melhor valor encontrado para cada instância. Na tabela o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo VNS1 é menor do que o da literatura.

As Tabelas A.10 e A.11 apresentam, respectivamente, os desvios percentuais entre o melhor custo da literatura e os melhores custos obtidos com 10 execuções dos algoritmos VNS1, VNS2, VNS3, VNS4 e VNS5 com 500 e 700 iterações para as instâncias com 20 e 32 locações. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo VNS1 é menor do que o da literatura.

A Tabela A.12 mostra os desvios percentuais entre o melhor custo da literatura e os melhores custos obtidos com 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para as instâncias com 20 e 32 locações. Em negrito encontra-se destacado o melhor valor encontrado para cada instância. Na tabela o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo VNS1 é menor do que o da literatura.

As Tabelas A.13 e A.14 apresentam, respectivamente, o tempo médio em segundos de 10 execuções dos algoritmos VNS1, VNS2, VNS3, VNS4 e VNS5 com 500 e 700 iterações para as instâncias com 6 e 12 locações.

A Tabela A.15 mostra o tempo médio em segundos de 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para as instâncias com 6 e 12 locações.

As Tabelas A.16 e A.17 apresentam, respectivamente, o tempo médio em segundos de 10 execuções dos algoritmos VNS1, VNS2, VNS3, VNS4 e VNS5 com 500 e 700 iterações para as instâncias com 20 e 32 locações.

A Tabela A.18 mostra o tempo médio em segundos de 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para as instâncias com 20 e 32 locações.

Tabela A.1: Melhor solução (obtida de 10 execuções) com 500 iterações do VNS para instâncias de 6 e 12 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
1	16	16	16	16	16
2	25	25	25	25	25
3	18	18	18	18	20
4	25	25	25	25	25
5	16	16	16	16	20
6	27	27	27	27	31
7	16	16	16	16	19
8	31	31	31	31	31
9	25	25	25	25	28
10	46	46	46	46	46
11	32	33	32	32	36
12	41	41	41	41	41
13	28	29	29	28	34
14	45	47	45	45	48
15	35	36	35	35	38
16	49	49	49	49	49
17	36	38	36	35	38
18	60	62	63	62	68
19	46	47	46	47	50
20	60	60	60	60	60
21	47	46	47	47	57
22	67	67	67	67	70
23	55	55	55	55	60
24	74	74	74	74	74
25	31	31	31	31	31
26	44	44	44	44	48
27	43	43	43	43	43
28	55	55	55	55	55
29	29	29	29	29	34
30	49	49	49	49	50
31	42	43	42	42	46
32	69	69	69	69	69
33	54	56	55	55	61
34	76	74	72	74	84
35	75	81	73	73	78
36	95	96	95	95	98
37	50	49	48	50	69
38	84	83	83	84	90
39	72	73	72	72	78
40	108	108	108	108	110
41	80	80	80	83	100
42	107	110	108	105	120
43	112	114	111	111	118
44	140	140	140	140	146
45	71	67	71	68	96
46	118	117	119	117	126
47	118	118	119	119	126
48	171	171	173	173	174

Tabela A.2: Melhor solução (obtida de 10 execuções) com 700 iterações do VNS para instâncias de 6 e 12 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
1	16	16	16	16	16
2	25	25	25	25	25
3	18	18	18	18	20
4	25	25	25	25	25
5	16	16	16	16	20
6	27	27	27	27	30
7	16	16	16	16	18
8	31	31	31	31	31
9	25	25	25	25	28
10	46	46	46	46	46
11	32	33	32	33	34
12	41	41	41	41	41
13	28	29	28	28	35
14	45	45	45	45	48
15	35	36	35	35	38
16	49	49	49	49	49
17	35	36	35	36	38
18	61	60	61	61	68
19	46	47	46	46	50
20	60	60	60	60	60
21	47	46	47	47	53
22	67	67	67	67	70
23	55	55	55	55	58
24	74	74	74	74	74
25	31	31	31	31	34
26	43	44	44	43	50
27	43	43	43	43	43
28	55	55	56	55	55
29	29	29	29	29	37
30	49	50	49	49	50
31	42	42	42	42	45
32	69	69	69	69	69
33	55	55	55	54	63
34	75	74	74	76	84
35	76	79	73	75	74
36	95	95	96	95	97
37	48	48	49	51	70
38	84	85	84	85	85
39	69	68	71	74	76
40	109	109	108	108	109
41	78	82	77	83	98
42	103	108	105	103	114
43	111	114	110	113	123
44	140	141	140	142	142
45	69	67	69	67	94
46	116	118	119	118	124
47	118	119	119	118	126
48	171	172	171	171	174

Tabela A.3: Melhor solução (obtida de 10 execuções) com 1000, 1200 e 1500 iterações do algoritmo VNS1 para instâncias de 6 e 12 locações

Inst.	1000	1200	1500
1	16	16	16
2	25	25	25
3	18	18	18
4	25	25	25
5	16	16	16
6	27	27	27
7	16	16	16
8	31	31	31
9	25	25	25
10	46	46	46
11	33	32	32
12	41	41	41
13	28	28	28
14	45	45	45
15	35	35	35
16	49	49	49
17	35	35	35
18	60	60	61
19	46	46	46
20	60	60	60
21	46	47	47
22	67	67	67
23	55	55	55
24	74	74	74
25	31	31	31
26	44	43	43
27	43	43	43
28	55	55	55
29	29	29	29
30	49	49	49
31	42	43	43
32	69	69	69
33	56	52	57
34	73	73	73
35	73	75	74
36	95	95	96
37	50	49	50
38	84	84	82
39	69	68	68
40	109	108	108
41	78	81	80
42	110	108	106
43	112	111	112
44	140	141	143
45	67	70	68
46	117	114	116
47	118	116	118
48	171	171	171

Tabela A.4: Melhor solução (obtida de 10 execuções) com 500 iterações do VNS para instâncias de 20 e 32 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
49	44	44	44	44	47
50	60	61	59	61	73
51	56	56	56	55	59
52	87	91	94	89	91
53	47	47	47	47	61
54	64	65	65	63	72
55	61	60	62	60	61
56	92	89	90	91	91
57	66	68	66	67	85
58	107	101	103	102	119
59	92	92	98	99	106
60	158	159	160	156	160
61	78	78	77	76	105
62	115	123	120	120	131
63	122	118	122	121	126
64	180	176	184	174	179
65	108	109	105	102	131
66	153	157	153	159	182
67	151	153	156	149	175
68	227	233	233	238	238
69	118	124	119	117	159
70	177	175	171	176	199
71	174	183	172	173	194
72	256	261	263	268	263
73	71	71	71	71	83
74	95	95	96	95	104
75	108	108	105	108	121
76	158	152	163	154	165
77	70	70	70	70	86
78	100	95	98	102	121
79	109	114	109	113	127
80	171	180	176	180	179
81	124	123	124	124	155
82	189	180	179	188	197
83	216	211	204	204	242
84	296	305	303	298	325
85	129	129	131	132	179
86	196	201	196	197	247
87	205	206	212	219	242
88	316	314	314	313	319
89	178	184	192	192	252
90	272	281	286	264	290
91	306	321	308	322	357
92	423	448	437	442	449
93	193	200	203	199	279
94	286	296	301	298	348
95	358	361	340	355	403
96	515	492	503	506	516

Tabela A.5: Melhor solução (obtida de 10 execuções) com 700 iterações do VNS para instâncias de 20 e 32 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
49	44	44	44	44	45
50	60	59	60	61	71
51	56	55	56	56	62
52	89	89	91	89	89
53	47	47	47	47	60
54	64	65	64	63	66
55	60	60	61	60	62
56	89	93	91	93	92
57	66	67	68	68	84
58	100	100	103	100	122
59	98	98	100	99	110
60	154	157	159	154	156
61	75	80	78	77	108
62	118	126	118	114	128
63	123	118	118	119	128
64	179	176	180	177	179
65	106	104	105	105	124
66	155	147	150	155	177
67	150	154	151	152	164
68	227	232	228	229	236
69	120	122	122	120	159
70	177	176	171	179	193
71	174	179	169	173	193
72	263	262	250	262	263
73	71	71	71	71	82
74	95	93	93	95	102
75	109	107	108	107	118
76	154	166	162	153	165
77	70	70	70	70	82
78	99	96	100	101	121
79	109	108	111	114	127
80	172	180	178	173	176
81	121	121	123	122	155
82	181	183	180	178	192
83	206	202	206	192	238
84	299	296	297	302	325
85	126	129	130	125	174
86	190	195	199	195	242
87	209	206	208	214	241
88	309	311	306	321	308
89	187	184	185	183	252
90	276	264	279	274	283
91	300	313	309	321	350
92	418	431	413	439	436
93	204	201	201	203	279
94	300	290	296	297	348
95	365	336	349	359	388
96	475	490	490	514	514

Tabela A.6: Melhor solução (obtida de 10 execuções) com 1000, 1200 e 1500 iterações do algoritmo VNS1 para instâncias de 20 e 32 locações

Inst.	1000	1200	1500
49	44	44	44
50	61	60	60
51	55	57	55
52	89	87	89
53	47	47	47
54	64	65	63
55	61	61	60
56	89	90	90
57	67	67	67
58	101	102	102
59	96	97	96
60	154	155	154
61	75	75	75
62	116	119	117
63	118	121	121
64	177	178	180
65	106	105	105
66	152	152	152
67	152	151	155
68	226	228	227
69	121	119	119
70	176	177	176
71	167	173	173
72	261	260	252
73	71	71	71
74	93	90	92
75	106	106	105
76	156	149	155
77	70	70	70
78	98	98	97
79	108	101	109
80	171	171	171
81	119	118	117
82	173	173	181
83	202	201	191
84	285	299	291
85	130	130	126
86	195	199	193
87	201	192	199
88	303	303	292
89	183	182	171
90	261	272	275
91	294	287	295
92	405	418	395
93	198	196	199
94	296	296	288
95	342	341	337
96	481	487	482

Tabela A.7: Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 500 iterações para instâncias de 6 e 12 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
1	-	-	-	-	-
2	-	-	-	-	-
3	-	-	-	-	10,0
4	-	-	-	-	-
5	-	-	-	-	20,0
6	-	-	-	-	12,9
7	-	-	-	-	15,8
8	-	-	-	-	-
9	-	-	-	-	10,7
10	-	-	-	-	-
11	-	3,0	-	-	11,1
12	-	-	-	-	-
13	-	3,4	3,4	-	17,6
14	-	4,3	-	-	6,3
15	-	2,8	-	-	7,9
16	-	-	-	-	-
17	2,8	7,9	2,8	-	7,9
18	-	3,2	4,8	3,2	11,8
19	-	2,1	-	2,1	8,0
20	-	-	-	-	-
21	2,1	-	2,1	2,1	19,3
22	-	-	-	-	4,3
23	-	-	-	-	8,3
24	-	-	-	-	-
25	-	-	-	-	-
26	2,3	2,3	2,3	2,3	10,4
27	-	-	-	-	-
28	-	-	-	-	-
29	-	-	-	-	14,7
30	-2,0	-2,0	-2,0	-2,0	-
31	-	2,3	-	-	8,7
32	-	-	-	-	-
33	1,9	5,4	3,6	3,6	13,1
34	5,3	2,7	-	2,7	14,3
35	9,3	16,0	6,8	6,8	12,8
36	5,3	6,3	5,3	5,3	8,2
37	6,0	4,1	2,1	6,0	31,9
38	8,3	7,2	7,2	8,3	14,4
39	6,9	8,2	6,9	6,9	14,1
40	3,7	3,7	3,7	3,7	5,5
41	2,5	2,5	2,5	6,0	22,0
42	2,8	5,5	3,7	1,0	13,3
43	1,8	3,5	0,9	0,9	6,8
44	2,1	2,1	2,1	2,1	6,2
45	7,0	1,5	7,0	2,9	31,3
46	5,9	5,1	6,7	5,1	11,9
47	5,9	5,9	6,7	6,7	11,9
48	1,2	1,2	2,3	2,3	2,9

Tabela A.8: Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 700 iterações para instâncias de 6 e 12 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
1	-	-	-	-	-
2	-	-	-	-	-
3	-	-	-	-	10,0
4	-	-	-	-	-
5	-	-	-	-	20,0
6	-	-	-	-	10,0
7	-	-	-	-	11,1
8	-	-	-	-	-
9	-	-	-	-	10,7
10	-	-	-	-	-
11	-	3,0	-	3,0	5,9
12	-	-	-	-	-
13	-	3,4	-	-	20,0
14	-	-	-	-	6,3
15	-	2,8	-	-	7,9
16	-	-	-	-	-
17	-	2,8	-	2,8	7,9
18	1,6	-	1,6	1,6	11,8
19	-	2,1	-	-	8,0
20	-	-	-	-	-
21	2,1	-	2,1	2,1	13,2
22	-	-	-	-	4,3
23	-	-	-	-	5,2
24	-	-	-	-	-
25	-	-	-	-	8,8
26	-	2,3	2,3	-	14,0
27	-	-	-	-	-
28	-	-	1,8	-	-
29	-	-	-	-	21,6
30	-2,0	-	-2,0	-2,0	-
31	-	-	-	-	6,7
32	-	-	-	-	-
33	3,6	3,6	3,6	1,9	15,9
34	4,0	2,7	2,7	5,3	14,3
35	10,5	13,9	6,8	9,3	8,1
36	5,3	5,3	6,3	5,3	7,2
37	2,1	2,1	4,1	7,8	32,9
38	8,3	9,4	8,3	9,4	9,4
39	2,9	1,5	5,6	9,5	11,8
40	4,6	4,6	3,7	3,7	4,6
41	-	4,9	-1,3	6,0	20,4
42	-1,0	3,7	1,0	-1,0	8,8
43	0,9	3,5	-	2,7	10,6
44	2,1	2,8	2,1	3,5	3,5
45	4,3	1,5	4,3	1,5	29,8
46	4,3	5,9	6,7	5,9	10,5
47	5,9	6,7	6,7	5,9	11,9
48	1,2	1,7	1,2	1,2	2,9

Tabela A.9: Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) do algoritmo VNS1 com 1000, 1200 e 1500 iterações para instâncias de 6 e 12 locações

Inst.	1000	1200	1500
1	-	-	-
2	-	-	-
3	-	-	-
4	-	-	-
5	-	-	-
6	-	-	-
7	-	-	-
8	-	-	-
9	-	-	-
10	-	-	-
11	3,0	-	-
12	-	-	-
13	-	-	-
14	-	-	-
15	-	-	-
16	-	-	-
17	-	-	-
18	-	-	1,6
19	-	-	-
20	-	-	-
21	-	2,1	2,1
22	-	-	-
23	-	-	-
24	-	-	-
25	-	-	-
26	2,3	-	-
27	-	-	-
28	-	-	-
29	-	-	-
30	-2,0	-2,0	-2,0
31	-	2,3	2,3
32	-	-	-
33	5,4	-1,9	7,0
34	1,4	1,4	1,4
35	6,8	9,3	8,1
36	5,3	5,3	6,3
37	6,0	4,1	6,0
38	8,3	8,3	6,1
39	2,9	1,5	1,5
40	4,6	3,7	3,7
41	-	3,7	2,5
42	5,5	3,7	1,9
43	1,8	0,9	1,8
44	2,1	2,8	4,2
45	1,5	5,7	2,9
46	5,1	2,6	4,3
47	5,9	4,3	5,9
48	1,2	1,2	1,2

Tabela A.10: Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 500 iterações para instâncias de 20 e 32 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
49	-2,3	-2,3	-2,3	-2,3	4,3
50	-5,0	-3,3	-6,8	-3,3	13,7
51	1,8	1,8	1,8	-	6,8
52	-12,6	-7,7	-4,3	-10,1	-7,7
53	-4,3	-4,3	-4,3	-4,3	19,7
54	-4,7	-3,1	-3,1	-6,3	6,9
55	-3,3	-5,0	-1,6	-5,0	-3,3
56	-5,4	-9,0	-7,8	-6,6	-6,6
57	-1,5	1,5	-1,5	-	21,2
58	0,9	-5,0	-2,9	-3,9	10,9
59	-9,8	-9,8	-3,1	-2,0	4,7
60	-0,6	-	0,6	-1,9	0,6
61	-5,1	-5,1	-6,5	-7,9	21,9
62	-12,2	-4,9	-7,5	-7,5	1,5
63	0,8	-2,5	0,8	-	4,0
64	-5,6	-8,0	-3,3	-9,2	-6,1
65	2,8	3,7	-	-2,9	19,8
66	-2,0	0,6	-2,0	1,9	14,3
67	-4,0	-2,6	-0,6	-5,4	10,3
68	-3,1	-0,4	-0,4	1,7	1,7
69	5,1	9,7	5,9	4,3	29,6
70	-0,6	-1,7	-4,1	-1,1	10,6
71	2,3	7,1	1,2	1,7	12,4
72	-3,5	-1,5	-0,8	1,1	-0,8
73	-4,2	-4,2	-4,2	-4,2	10,8
74	-2,1	-2,1	-1,0	-2,1	6,7
75	-1,9	-1,9	-4,8	-1,9	9,1
76	1,9	-2,0	4,9	-0,6	6,1
77	-4,3	-4,3	-4,3	-4,3	15,1
78	-1,0	-6,3	-3,1	1,0	16,5
79	-0,9	3,5	-0,9	2,7	13,4
80	-2,3	2,8	0,6	2,8	2,2
81	4,0	3,3	4,0	4,0	23,2
82	6,9	2,2	1,7	6,4	10,7
83	11,1	9,0	5,9	5,9	20,7
84	4,7	7,5	6,9	5,4	13,2
85	3,1	3,1	4,6	5,3	30,2
86	2,0	4,5	2,0	2,5	22,3
87	5,9	6,3	9,0	11,9	20,2
88	4,4	3,8	3,8	3,5	5,3
89	3,9	7,1	10,9	10,9	32,1
90	3,7	6,8	8,4	0,8	9,7
91	7,2	11,5	7,8	11,8	20,4
92	6,6	11,8	9,6	10,6	12,0
93	2,1	5,5	6,9	5,0	32,3
94	1,7	5,1	6,6	5,7	19,3
95	11,2	11,9	6,5	10,4	21,1
96	9,9	5,7	7,8	8,3	10,1

Tabela A.11: Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) dos algoritmos VNS com 700 iterações para instâncias de 20 e 32 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
49	-2,3	-2,3	-2,3	-2,3	-
50	-5,0	-6,8	-5,0	-3,3	11,3
51	1,8	-	1,8	1,8	11,3
52	-10,1	-10,1	-7,7	-10,1	-10,1
53	-4,3	-4,3	-4,3	-4,3	18,3
54	-4,7	-3,1	-4,7	-6,3	-1,5
55	-5,0	-5,0	-3,3	-5,0	-1,6
56	-9,0	-4,3	-6,6	-4,3	-5,4
57	-1,5	-	1,5	1,5	20,2
58	-6,0	-6,0	-2,9	-6,0	13,1
59	-3,1	-3,1	-1,0	-2,0	8,2
60	-3,2	-1,3	-	-3,2	-1,9
61	-9,3	-2,5	-5,1	-6,5	24,1
62	-9,3	-2,4	-9,3	-13,2	-0,8
63	1,6	-2,5	-2,5	-1,7	5,5
64	-6,1	-8,0	-5,6	-7,3	-6,1
65	0,9	-1,0	-	-	15,3
66	-0,6	-6,1	-4,0	-0,6	11,9
67	-4,7	-1,9	-4,0	-3,3	4,3
68	-3,1	-0,9	-2,6	-2,2	0,8
69	6,7	8,2	8,2	6,7	29,6
70	-0,6	-1,1	-4,1	0,6	7,8
71	2,3	5,0	-0,6	1,7	11,9
72	-0,8	-1,1	-6,0	-1,1	-0,8
73	-4,2	-4,2	-4,2	-4,2	9,8
74	-2,1	-4,3	-4,3	-2,1	4,9
75	-0,9	-2,8	-1,9	-2,8	6,8
76	-0,6	6,6	4,3	-1,3	6,1
77	-4,3	-4,3	-4,3	-4,3	11,0
78	-2,0	-5,2	-1,0	-	16,5
79	-0,9	-1,9	0,9	3,5	13,4
80	-1,7	2,8	1,7	-1,2	0,6
81	1,7	1,7	3,3	2,5	23,2
82	2,8	3,8	2,2	1,1	8,3
83	6,8	5,0	6,8	-	19,3
84	5,7	4,7	5,1	6,6	13,2
85	0,8	3,1	3,8	-	28,2
86	-1,1	1,5	3,5	1,5	20,7
87	7,7	6,3	7,2	9,8	19,9
88	2,3	2,9	1,3	5,9	1,9
89	8,6	7,1	7,6	6,6	32,1
90	5,1	0,8	6,1	4,4	7,4
91	5,3	9,3	8,1	11,5	18,9
92	5,5	8,4	4,4	10,0	9,4
93	7,4	6,0	6,0	6,9	32,3
94	6,3	3,1	5,1	5,4	19,3
95	12,9	5,4	8,9	11,4	18,0
96	2,3	5,3	5,3	9,7	9,7

Tabela A.12: Desvio percentual entre a melhor solução da literatura e melhor solução (obtida de 10 execuções) do algoritmo VMS1 com 1000, 1200 e 1500 iterações para instâncias de 20 e 32 locações

Inst.	1000	1200	1500
49	-2,3	-2,3	-2,3
50	-3,3	-5,0	-5,0
51	-	3,5	-
52	-10,1	-12,6	-10,1
53	-4,3	-4,3	-4,3
54	-4,7	-3,1	-6,3
55	-3,3	-3,3	-5,0
56	-9,0	-7,8	-7,8
57	-	-	-
58	-5,0	-3,9	-3,9
59	-5,2	-4,1	-5,2
60	-3,2	-2,6	-3,2
61	-9,3	-9,3	-9,3
62	-11,2	-8,4	-10,3
63	-2,5	-	-
64	-7,3	-6,7	-5,6
65	0,9	-	-
66	-2,6	-2,6	-2,6
67	-3,3	-4,0	-1,3
68	-3,5	-2,6	-3,1
69	7,4	5,9	5,9
70	-1,1	-0,6	-1,1
71	-1,8	1,7	1,7
72	-1,5	-1,9	-5,2
73	-4,2	-4,2	-4,2
74	-4,3	-7,8	-5,4
75	-3,8	-3,8	-4,8
76	0,6	-4,0	-
77	-4,3	-4,3	-4,3
78	-3,1	-3,1	-4,1
79	-1,9	-8,9	-0,9
80	-2,3	-2,3	-2,3
81	-	-0,8	-1,7
82	-1,7	-1,7	2,8
83	5,0	4,5	-0,5
84	1,1	5,7	3,1
85	3,8	3,8	0,8
86	1,5	3,5	0,5
87	4,0	-0,5	3,0
88	0,3	0,3	-3,4
89	6,6	6,0	-
90	-0,4	3,7	4,7
91	3,4	1,0	3,7
92	2,5	5,5	-
93	4,5	3,6	5,0
94	5,1	5,1	2,4
95	7,0	6,7	5,6
96	3,5	4,7	3,7

Tabela A.13: Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 500 iterações para instâncias de 6 e 12 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
1	1,6	0,8	0,8	0,8	0,7
2	1,6	0,7	0,7	0,8	0,7
3	1,7	0,8	0,8	0,8	0,7
4	1,8	0,5	0,6	0,6	0,6
5	3,0	0,8	0,9	0,9	0,7
6	2,6	0,8	0,7	0,8	0,7
7	2,6	0,7	0,8	0,7	0,7
8	1,8	0,5	0,5	0,5	0,5
9	5,5	1,6	1,5	1,6	1,5
10	5,0	1,4	1,4	1,5	1,4
11	5,0	1,4	1,4	1,4	1,3
12	2,7	1,0	0,9	1,0	0,9
13	5,4	1,6	1,6	1,6	1,4
14	5,0	1,5	1,4	1,5	1,3
15	5,5	1,6	1,5	1,6	1,5
16	4,4	1,1	1,3	1,3	1,2
17	9,5	2,7	2,8	2,8	2,5
18	5,6	2,3	2,4	2,4	2,3
19	5,5	2,4	2,4	2,5	2,3
20	3,9	1,7	1,7	1,9	1,6
21	6,2	2,7	2,7	2,8	2,4
22	5,3	2,4	2,4	2,5	2,1
23	6,5	2,8	2,8	2,9	2,7
24	5,1	1,9	2,2	2,1	2,4
25	15,4	6,9	6,7	7,1	6,5
26	13,8	6,5	6,3	6,4	6,1
27	14,5	7,1	6,9	7,2	6,6
28	10,8	5,4	5,1	5,3	5,2
29	14,9	7,2	6,9	7,0	6,3
30	14,4	6,7	6,4	6,6	6,1
31	13,4	5,7	6,1	5,7	5,6
32	10,1	4,5	4,7	4,6	5,2
33	32,1	14,5	14,4	14,7	13,5
34	28,0	12,3	12,7	12,8	12,2
35	33,5	13,7	13,6	13,5	13,6
36	14,4	8,8	9,1	9,4	9,3
37	30,5	14,3	14,6	14,6	13,3
38	29,2	13,0	13,7	13,0	12,8
39	34,1	12,5	13,5	12,4	12,5
40	18,7	9,3	9,4	9,2	9,3
41	68,5	25,9	25,3	25,6	22,7
42	65,5	22,3	21,8	21,8	20,8
43	61,3	22,9	23,7	23,3	22,6
44	28,2	15,6	15,5	15,4	16,4
45	67,9	27,9	25,8	25,7	22,5
46	62,8	23,4	22,8	22,7	21,1
47	54,4	21,9	22,5	21,2	20,6
48	43,9	15,8	15,9	16,6	15,9

Tabela A.14: Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 700 iterações para instâncias de 6 e 12 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
1	0,9	1,1	0,9	1,1	0,8
2	0,8	0,9	0,8	1,1	0,8
3	0,9	1,1	0,9	1,1	0,8
4	0,7	0,8	0,6	0,8	0,6
5	0,9	1,1	0,9	1,2	0,7
6	0,8	1,2	0,9	1,0	0,7
7	0,8	1,0	0,8	1,0	0,7
8	0,6	0,7	0,6	0,7	0,6
9	1,6	2,2	1,7	2,2	1,5
10	1,5	1,9	1,5	2,0	1,5
11	1,6	2,0	1,4	2,0	1,4
12	1,0	1,4	1,0	1,3	1,0
13	1,8	2,2	1,6	2,2	1,5
14	1,6	2,0	1,5	2,0	1,4
15	1,6	2,2	1,7	2,2	1,6
16	1,4	1,6	1,4	1,7	1,4
17	2,8	3,7	2,9	3,7	2,7
18	2,5	3,2	2,5	3,3	2,4
19	2,5	3,4	2,5	3,3	2,5
20	1,8	2,4	1,7	2,5	1,8
21	2,9	3,7	2,9	3,6	2,5
22	2,5	3,3	2,4	3,2	2,3
23	2,9	3,8	3,0	4,1	3,3
24	2,3	2,7	2,5	2,8	2,7
25	6,9	9,7	6,5	9,6	6,3
26	6,3	8,9	6,3	9,0	6,1
27	6,8	9,9	6,8	10,0	6,7
28	4,8	7,5	4,9	7,5	5,2
29	7,0	10,4	6,8	10,0	6,3
30	6,4	8,9	6,4	9,4	6,1
31	6,0	8,0	5,8	8,2	5,5
32	4,6	6,5	4,6	6,6	5,1
33	13,8	20,2	13,7	20,3	12,9
34	12,1	17,5	12,2	17,6	11,6
35	13,4	18,0	13,3	19,0	12,9
36	8,7	12,8	8,8	12,6	9,4
37	13,8	20,1	14,3	20,7	12,3
38	12,6	17,8	12,7	18,8	12,3
39	12,6	17,4	12,7	18,3	12,0
40	8,9	12,8	8,7	12,9	9,3
41	24,5	36,1	24,4	34,2	22,7
42	21,5	29,7	20,5	29,6	20,0
43	22,2	31,9	22,9	33,5	21,5
44	14,7	21,5	15,1	21,5	15,3
45	24,9	35,3	25,3	35,5	22,5
46	20,5	31,1	22,3	30,5	20,9
47	20,4	31,0	20,1	28,9	20,3
48	15,1	21,4	14,5	22,4	15,2

Tabela A.15: Tempo médio (em segundos) de 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para instâncias de 6 e 12 locações

Inst.	1000	1200	1500
1	1,6	1,5	1,9
2	1,5	1,5	1,7
3	1,6	1,6	2,0
4	1,1	1,1	1,3
5	1,7	1,7	1,9
6	1,6	1,4	1,7
7	1,5	1,4	1,8
8	1,1	1,0	1,2
9	3,1	2,9	3,7
10	2,8	2,6	3,3
11	2,9	2,7	3,2
12	1,9	1,9	2,3
13	3,1	2,9	3,7
14	2,7	2,7	3,2
15	3,1	2,9	3,7
16	2,5	2,4	2,8
17	5,2	4,7	5,9
18	4,7	4,3	5,3
19	4,7	4,4	5,5
20	3,4	3,2	3,8
21	5,4	5,0	6,1
22	4,6	4,2	5,2
23	5,5	4,9	6,4
24	4,3	4,0	5,2
25	13,6	11,5	14,1
26	12,4	10,4	13,6
27	13,2	11,4	14,5
28	9,7	8,1	10,5
29	13,3	11,7	14,4
30	12,2	10,9	13,3
31	11,7	10,1	13,0
32	9,2	7,7	10,0
33	28,3	24,0	29,2
34	24,2	20,3	25,2
35	26,4	23,0	27,7
36	17,0	14,6	18,8
37	28,4	23,7	29,9
38	24,7	21,3	25,6
39	25,4	20,8	26,7
40	18,0	15,1	19,2
41	48,7	40,7	51,6
42	40,7	34,6	44,0
43	47,2	37,4	47,9
44	31,4	26,0	33,1
45	48,5	42,5	51,7
46	42,0	36,4	46,0
47	40,6	34,8	44,1
48	30,3	25,7	32,0

Tabela A.16: Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 100 iterações para instâncias de 20 e 32 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
49	96,9	33,6	31,1	34,1	30,2
50	81,5	27,2	27,2	28,6	26,1
51	71,7	29,8	27,2	28,1	25,9
52	41,5	18,7	19,5	18,2	19,4
53	98,3	35,5	33,5	34,9	32,5
54	65,1	34,5	31,1	33,9	30,2
55	50,2	33,4	30,9	33,3	31,7
56	35,9	25,5	24,3	24,6	26,3
57	110,2	72,9	70,8	72,1	67,8
58	104,0	65,2	60,4	62,0	59,1
59	153,0	57,6	60,1	60,2	55,3
60	68,4	43,2	41,9	44,3	48,6
61	173,2	79,9	75,2	80,1	70,0
62	122,3	80,9	74,1	72,7	67,9
63	139,4	69,0	68,1	67,6	66,6
64	137,0	52,5	51,9	52,9	60,3
65	324,7	131,5	129,6	128,7	126,3
66	209,6	113,3	112,4	110,0	110,2
67	199,8	117,2	111,0	110,2	116,3
68	203,7	78,9	84,4	81,9	87,3
69	318,7	137,2	135,8	130,6	124,3
70	303,8	136,3	120,0	128,2	122,3
71	264,9	121,8	122,5	126,4	112,7
72	226,4	104,4	90,0	95,5	104,0
73	308,4	156,8	148,7	154,4	152,1
74	308,1	150,8	133,5	156,5	139,9
75	361,0	167,2	149,7	144,6	157,0
76	207,4	136,9	112,7	111,2	108,0
77	393,9	159,7	150,4	158,3	145,0
78	402,9	174,6	159,0	166,9	152,7
79	405,4	154,7	158,7	163,6	154,2
80	197,5	157,2	127,0	120,8	145,0
81	605,7	431,4	349,4	366,6	340,6
82	619,8	380,7	298,4	305,9	319,2
83	820,1	454,3	350,1	360,0	330,0
84	742,0	336,3	275,7	287,3	266,6
85	770,7	380,8	377,7	355,1	325,6
86	680,3	355,1	319,3	355,0	342,9
87	733,3	430,5	383,1	339,9	335,2
88	634,2	315,6	290,4	305,9	311,6
89	1596,5	902,7	681,4	625,9	604,7
90	1303,2	610,0	555,9	590,2	536,5
91	1382,1	794,7	588,8	576,6	665,4
92	842,9	428,4	434,4	428,1	406,7
93	1015,4	653,0	673,4	648,5	588,1
94	1187,9	753,1	607,4	614,8	644,0
95	1114,2	611,8	658,7	665,1	741,9
96	1040,5	677,1	538,9	510,7	578,8

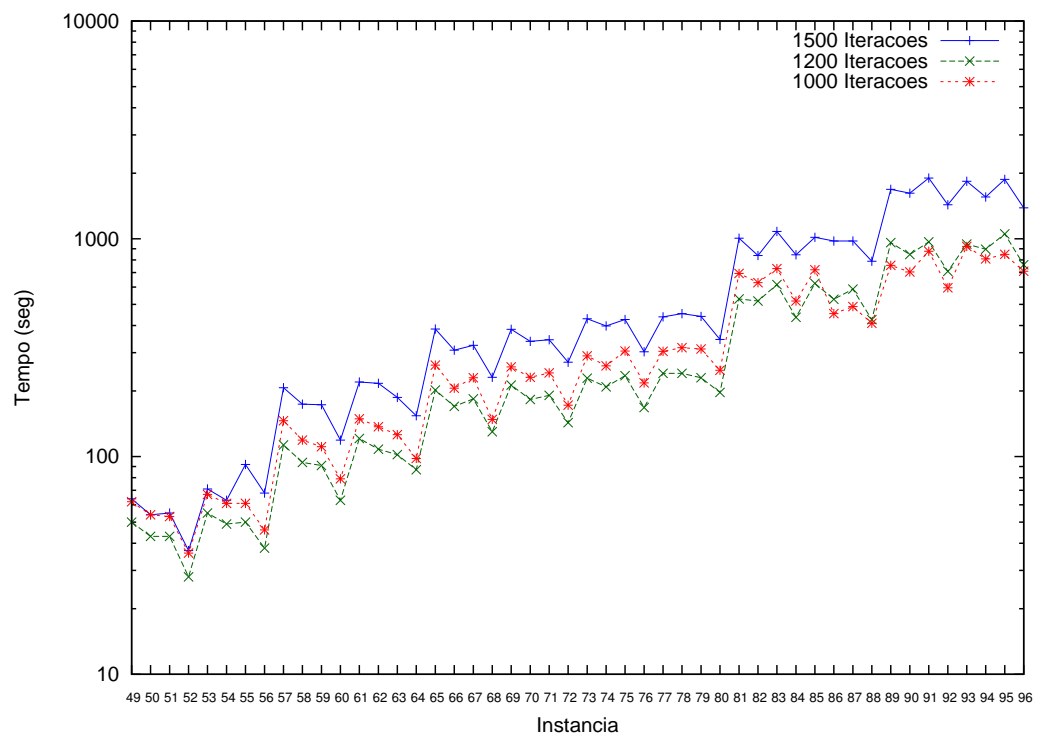
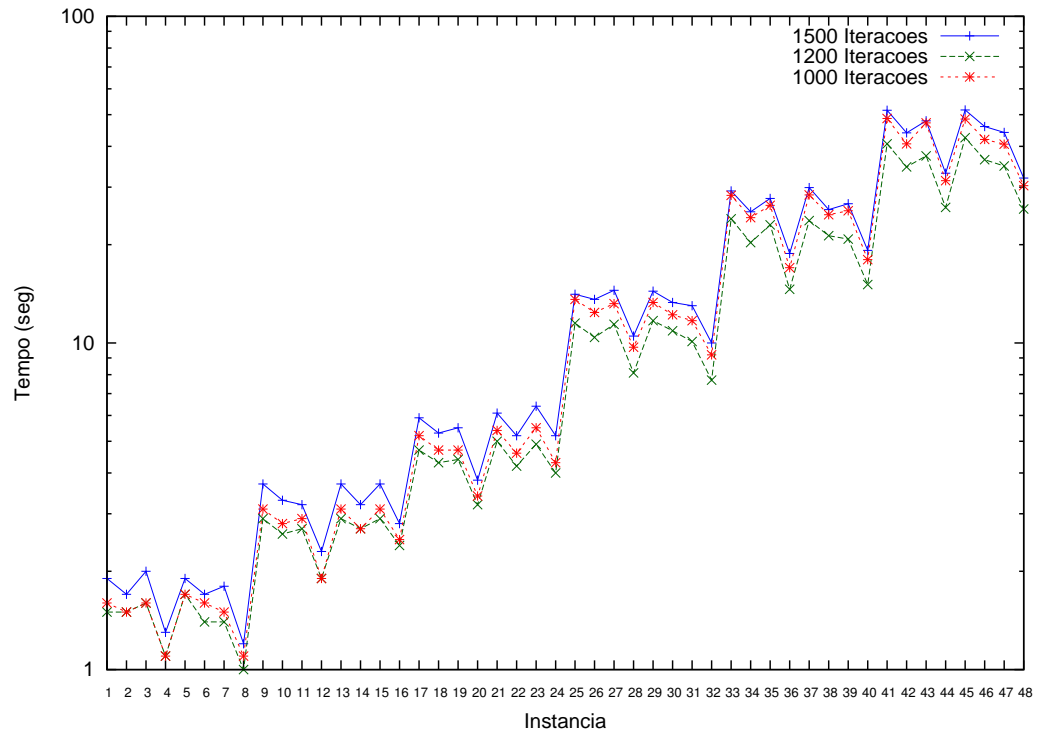
Tabela A.17: Tempo médio (em segundos) de 10 execuções dos algoritmos VNS com 100 iterações para instâncias de 20 e 32 locações

Inst.	VNS1	VNS2	VNS3	VNS4	VNS5
49	29,6	46,2	29,9	48,3	30,0
50	25,7	38,0	26,3	38,5	24,1
51	25,6	37,4	25,5	38,4	23,8
52	17,3	26,6	16,8	26,8	18,3
53	32,2	47,6	32,2	48,5	29,7
54	29,1	44,9	28,8	43,2	29,2
55	30,8	44,2	28,8	43,5	28,8
56	22,6	34,6	22,8	33,3	23,0
57	67,3	98,5	68,4	100,4	61,9
58	59,3	88,4	58,1	83,6	54,6
59	54,0	81,4	54,4	82,6	53,6
60	38,2	60,2	40,8	60,2	43,2
61	73,7	106,0	70,4	106,4	64,8
62	68,5	104,1	65,6	99,0	64,6
63	61,4	90,8	62,0	89,7	60,0
64	47,5	78,6	47,8	74,1	52,8
65	128,5	182,3	120,8	202,2	121,8
66	105,0	157,3	112,9	164,7	103,8
67	112,3	166,9	103,9	155,9	103,5
68	80,6	119,1	78,3	126,8	84,0
69	130,5	194,3	132,8	190,3	122,1
70	115,8	170,0	167,8	175,9	119,4
71	160,5	176,0	159,7	176,1	114,3
72	126,2	141,4	129,0	138,9	92,9
73	207,9	207,8	202,2	213,1	140,4
74	189,6	226,4	187,8	208,9	127,0
75	209,2	218,4	208,5	219,9	147,6
76	159,5	148,0	158,0	153,4	98,1
77	209,8	211,6	219,8	217,5	139,0
78	223,8	225,5	208,1	214,3	133,2
79	218,1	222,8	217,0	267,8	130,5
80	170,4	191,5	176,0	189,2	130,2
81	459,2	549,9	484,6	508,8	302,6
82	416,3	499,7	438,0	469,0	290,2
83	514,7	552,5	540,7	677,2	304,0
84	373,8	394,2	377,0	431,3	236,6
85	530,7	470,0	494,8	469,7	318,4
86	469,4	491,0	492,7	487,9	292,9
87	484,1	535,5	506,1	543,4	333,7
88	399,1	470,8	388,2	402,1	281,7
89	894,4	1020,0	907,5	841,5	541,5
90	766,9	783,6	841,5	795,6	519,2
91	1029,2	549,5	635,5	554,6	611,4
92	644,9	438,4	576,5	383,1	430,2
93	893,1	609,2	878,9	569,4	582,7
94	901,5	577,6	791,3	577,7	555,5
95	896,6	631,2	934,9	602,9	618,0
96	647,8	523,3	821,0	461,2	531,0

Tabela A.18: Tempo médio (em segundos) de 10 execuções do algoritmo VNS1 com 1000, 1200 e 1500 iterações para instâncias de 20 e 32 locações

Inst.	1000	1200	1500
49	62,2	50,1	64,7
50	54,9	43,8	54,6
51	53,0	43,8	55,4
52	36,2	28,9	37,3
53	67,8	55,1	71,1
54	61,2	49,5	63,7
55	61,9	50,8	92,8
56	46,5	38,3	68,9
57	146,3	113,3	207,6
58	119,7	94,2	174,6
59	111,5	91,1	173,9
60	79,3	63,5	119,9
61	149,5	121,8	220,8
62	137,5	108,4	217,0
63	126,7	102,8	187,2
64	98,1	87,7	154,8
65	263,1	202,3	385,6
66	206,3	170,3	308,9
67	230,8	184,1	324,4
68	148,8	130,7	231,2
69	258,2	213,2	384,9
70	231,4	183,6	338,1
71	242,5	191,4	344,7
72	172,6	143,1	271,1
73	290,6	229,8	429,8
74	261,6	209,0	398,2
75	305,9	235,1	426,6
76	218,7	168,4	303,2
77	304,8	241,3	438,8
78	316,4	241,5	453,0
79	312,2	230,2	440,9
80	249,1	197,6	345,9
81	693,1	529,4	1006,7
82	629,0	518,0	838,6
83	730,8	616,6	1081,2
84	518,5	436,8	844,9
85	721,0	624,4	1015,6
86	453,4	528,2	977,0
87	488,3	587,3	977,6
88	409,8	425,2	789,2
89	755,0	960,9	1687,0
90	704,4	847,6	1619,6
91	877,9	968,6	1901,0
92	596,7	709,3	1434,5
93	922,0	945,1	1838,1
94	809,1	895,9	1555,7
95	849,4	1053,5	1875,5
96	710,6	760,7	1388,2

Figura A.1: Comparativo dos tempos médios do algoritmo VNS1 para execuções com 1000, 1200 e 1500 iterações



Apêndice B

Resultados Computacionais dos Algoritmos ILS Propostos

Os algoritmos ILS propostos no Capítulo 5, Seção 5.2, foram testados e os resultados obtidos podem ser vistos nas tabelas a seguir.

As Tabelas B.1 e B.2 apresentam os melhores custos obtidos em 10 execuções de cada algoritmo ILS. Em negrito encontra-se destacados o melhor valor encontrado pelos algoritmos VNS para cada instância.

As Tabelas B.3 e B.4 apresentam os desvios percentuais entre o melhor custo conhecido e os melhores custos encontrados pelos algoritmos ILS. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo ILS é menor do que o da literatura.

As Tabelas B.5 e B.6 apresentam as médias dos tempos em segundos em 10 execuções independentes de cada algoritmo.

Tabela B.1: Melhor solução (obtida de 10 execuções) do ILS para instâncias de 6 e 12 locações

Inst.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
1	16	16	16	16	16	16	16	16	16	16
2	25	25	25	25	25	25	25	25	25	25
3	18	18	18	18	18	18	18	18	18	18
4	25	25	25	25	25	25	25	25	25	25
5	16	16	16	16	16	16	16	16	16	16
6	27	27	27	27	27	27	27	27	27	27
7	16	16	16	16	16	16	16	16	16	16
8	31	31	31	31	31	31	31	31	31	31
9	25	25	25	25	25	25	25	25	25	25
10	46	46	46	46	46	46	46	46	46	46
11	32	33	32	32	32	32	33	32	32	32
12	41	41	41	41	41	41	41	41	41	41
13	28	29	28	28	29	28	29	28	28	29
14	45	45	45	45	45	45	45	45	45	45
15	35	35	35	35	35	35	35	35	35	35
16	49	49	49	49	49	49	49	49	49	49
17	35	35	35	35	35	35	35	35	35	35
18	60	61	60	60	62	60	61	60	60	62
19	46	47	46	47	46	46	47	46	47	46
20	60	60	60	60	60	60	60	60	60	60
21	46	46	47	46	48	46	46	46	46	46
22	67	67	67	67	67	67	67	67	67	67
23	55	56	55	55	55	55	56	55	55	55
24	74	74	74	74	74	74	74	74	74	74
25	31	31	31	31	32	31	31	31	31	33
26	43	44	43	44	44	43	45	43	43	44
27	43	43	43	43	43	43	43	43	43	43
28	55	55	55	55	55	55	55	55	55	55
29	29	29	29	29	29	29	29	29	29	30
30	49	49	49	49	49	49	49	49	49	50
31	42	42	42	44	44	43	42	42	42	45
32	69	69	69	69	69	69	69	69	69	69
33	54	54	54	54	55	55	54	55	54	56
34	74	73	73	74	79	74	73	75	75	76
35	77	77	73	77	80	76	77	74	77	80
36	95	95	95	95	95	95	95	95	95	102
37	48	48	48	48	52	50	49	48	50	57
38	82	82	83	80	86	82	84	81	85	84
39	68	73	70	74	74	71	74	70	74	74
40	108	110	109	110	108	108	109	109	110	110
41	80	78	77	82	86	78	78	80	80	84
42	105	102	104	108	114	104	107	108	108	110
43	115	114	112	111	118	110	116	111	115	116
44	142	140	141	140	143	140	140	141	140	141
45	67	68	66	66	83	71	67	71	70	75
46	116	115	115	114	118	115	115	114	117	116
47	118	121	118	117	118	116	120	117	120	119
48	171	173	170	174	173	171	171	174	175	174

Tabela B.2: Melhor solução (obtida de 10 execuções) do ILS para instâncias de 20 e 32 locações

Inst.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
49	44	44	44	44	44	44	44	44	44	44
50	59	60	60	61	64	59	61	61	61	60
51	56	57	54	55	57	56	57	54	57	56
52	87	90	90	92	91	89	91	90	91	89
53	47	47	47	47	48	47	47	47	47	47
54	62	63	64	63	65	63	63	66	63	63
55	62	63	62	62	61	63	64	61	61	68
56	89	92	88	89	92	89	92	90	89	93
57	66	65	66	65	71	69	66	67	67	72
58	101	98	98	99	104	96	99	102	98	100
59	100	100	101	95	105	97	98	96	100	101
60	154	157	160	160	156	155	162	154	163	159
61	74	77	78	75	80	75	76	78	76	82
62	114	118	114	114	118	117	115	114	115	115
63	123	121	121	116	127	120	117	119	121	128
64	176	177	180	177	173	176	173	179	173	176
65	100	100	100	102	112	101	104	103	103	109
66	150	147	151	149	156	150	152	151	149	157
67	153	155	156	150	163	150	147	149	150	156
68	223	225	228	225	229	224	221	224	225	225
69	117	113	115	115	127	118	119	118	121	125
70	173	169	164	170	180	173	171	170	173	183
71	175	181	169	180	191	169	162	167	172	179
72	249	256	254	250	256	247	258	247	249	250
73	71	71	71	71	71	71	71	71	71	73
74	94	88	92	90	96	96	95	94	93	100
75	107	112	110	106	108	109	111	107	106	110
76	152	153	148	148	153	150	157	153	150	154
77	70	70	70	70	73	70	71	71	71	73
78	99	98	98	99	108	98	105	101	99	106
79	107	111	110	116	114	110	117	112	107	117
80	167	166	170	168	182	164	168	171	171	177
81	116	121	115	117	120	118	121	115	118	126
82	167	168	171	175	180	168	174	175	177	192
83	209	201	199	189	216	197	195	198	193	208
84	283	288	290	284	293	287	272	278	275	292
85	127	123	120	121	133	125	124	126	127	126
86	197	191	195	195	203	196	192	195	191	205
87	197	209	202	211	215	191	201	191	198	216
88	295	311	300	299	299	292	296	291	280	286
89	170	170	167	177	178	174	180	178	177	192
90	253	250	257	256	271	256	259	262	252	267
91	291	309	296	301	316	287	290	292	292	309
92	393	410	407	409	412	374	396	392	382	377
93	192	190	192	193	212	185	199	189	190	212
94	268	268	274	280	295	282	268	282	275	293
95	327	349	322	359	374	328	327	337	334	361
96	474	477	483	481	483	479	465	475	471	475

Tabela B.3: Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos ILS para instâncias de 6 e 12 locações

Inst.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
11	-	3,0	-	-	-	-	3,0	-	-	-
12	-	-	-	-	-	-	-	-	-	-
13	-	3,4	-	-	3,4	-	3,4	-	-	3,4
14	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-
18	-	1,6	-	-	3,2	-	1,6	-	-	3,2
19	-	2,1	-	2,1	-	-	2,1	-	2,1	-
20	-	-	-	-	-	-	-	-	-	-
21	-	-	2,1	-	4,2	-	-2,2	-	-	-
22	-	-	-	-	-	-	-	-	-	-
23	-	1,8	-	-	-	-	1,8	-	-	-
24	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	3,1	-	-	-	-	6,1
26	-	2,3	-	2,3	2,3	-	4,4	-	-	2,3
27	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	3,3
30	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-
31	-	-	-	4,5	4,5	2,3	-	-	-	6,7
32	-	-	-	-	-	-	-	-	-	-
33	1,9	1,9	1,9	1,9	3,6	3,6	1,9	3,6	1,9	5,4
34	2,7	1,4	1,4	2,7	8,9	2,7	1,4	4,0	4,0	5,3
35	11,7	11,7	6,8	11,7	15,0	10,5	11,7	8,1	11,7	15,0
36	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	11,8
37	2,1	2,1	2,1	2,1	9,6	6,0	4,1	2,1	6,0	17,5
38	6,1	6,1	7,2	3,8	10,5	6,1	8,3	4,9	9,4	8,3
39	1,5	8,2	4,3	9,5	9,5	5,6	9,5	4,3	9,5	9,5
40	3,7	5,5	4,6	5,5	3,7	3,7	4,6	4,6	5,5	5,5
41	2,5	-	-1,3	4,9	9,3	-	-	2,5	2,5	7,1
42	1,0	-2,0	-	3,7	8,8	-	2,8	3,7	3,7	5,5
43	4,3	3,5	1,8	0,9	6,8	-	5,2	0,9	4,3	5,2
44	3,5	2,1	2,8	2,1	4,2	2,1	2,1	2,8	2,1	2,8
45	1,5	2,9	-	-	20,5	7,0	1,5	7,0	5,7	12,0
46	4,3	3,5	3,5	2,6	5,9	3,5	3,5	2,6	5,1	4,3
47	5,9	8,3	5,9	5,1	5,9	4,3	7,5	5,1	7,5	6,7
48	1,2	2,3	0,6	2,9	2,3	1,2	1,2	2,9	3,4	2,9

Tabela B.4: Desvio percentual entre a melhor solução da literatura e melhor solução dos algoritmos ILS para instâncias de 20 e 32 locações

Inst.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
49	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3
50	-6,8	-5,0	-5,0	-3,3	1,6	-6,8	-3,3	-3,3	-3,3	-5,0
51	1,8	3,5	-1,9	-	3,5	1,8	3,5	-1,9	3,5	1,8
52	-12,6	-8,9	-8,9	-6,5	-7,7	-10,1	-7,7	-8,9	-7,7	-10,1
53	-4,3	-4,3	-4,3	-4,3	-2,1	-4,3	-4,3	-4,3	-4,3	-4,3
54	-8,1	-6,3	-4,7	-6,3	-3,1	-6,3	-6,3	-1,5	-6,3	-6,3
55	-1,6	-	-1,6	-1,6	-3,3	-	1,6	-3,3	-3,3	7,4
56	-9,0	-5,4	-10,2	-9,0	-5,4	-9,0	-5,4	-7,8	-9,0	-4,3
57	-1,5	-3,1	-1,5	-3,1	5,6	2,9	-1,5	-	-	6,9
58	-5,0	-8,2	-8,2	-7,1	-1,9	-10,4	-7,1	-3,9	-8,2	-6,0
59	-1,0	-1,0	-	-6,3	3,8	-4,1	-3,1	-5,2	-1,0	-
60	-3,2	-1,3	0,6	0,6	-1,9	-2,6	1,9	-3,2	2,5	-
61	-10,8	-6,5	-5,1	-9,3	-2,5	-9,3	-7,9	-5,1	-7,9	-
62	-13,2	-9,3	-13,2	-13,2	-9,3	-10,3	-12,2	-13,2	-12,2	-12,2
63	1,6	-	-	-4,3	4,7	-0,8	-3,4	-1,7	-	5,5
64	-8,0	-7,3	-5,6	-7,3	-9,8	-8,0	-9,8	-6,1	-9,8	-8,0
65	-5,0	-5,0	-5,0	-2,9	6,3	-4,0	-1,0	-1,9	-1,9	3,7
66	-4,0	-6,1	-3,3	-4,7	-	-4,0	-2,6	-3,3	-4,7	0,6
67	-2,6	-1,3	-0,6	-4,7	3,7	-4,7	-6,8	-5,4	-4,7	-0,6
68	-4,9	-4,0	-2,6	-4,0	-2,2	-4,5	-5,9	-4,5	-4,0	-4,0
69	4,3	0,9	2,6	2,6	11,8	5,1	5,9	5,1	7,4	10,4
70	-2,9	-5,3	-8,5	-4,7	1,1	-2,9	-4,1	-4,7	-2,9	2,7
71	2,9	6,1	-0,6	5,6	11,0	-0,6	-4,9	-1,8	1,2	5,0
72	-6,4	-3,5	-4,3	-6,0	-3,5	-7,3	-2,7	-7,3	-6,4	-6,0
73	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-1,4
74	-3,2	-10,2	-5,4	-7,8	-1,0	-1,0	-2,1	-3,2	-4,3	3,0
75	-2,8	1,8	-	-3,8	-1,9	-0,9	0,9	-2,8	-3,8	-
76	-2,0	-1,3	-4,7	-4,7	-1,3	-3,3	1,3	-1,3	-3,3	-0,6
77	-4,3	-4,3	-4,3	-4,3	-	-4,3	-2,8	-2,8	-2,8	-
78	-2,0	-3,1	-3,1	-2,0	6,5	-3,1	3,8	-	-2,0	4,7
79	-2,8	0,9	-	5,2	3,5	-	6,0	1,8	-2,8	6,0
80	-4,8	-5,4	-2,9	-4,2	3,8	-6,7	-4,2	-2,3	-2,3	1,1
81	-2,6	1,7	-3,5	-1,7	0,8	-0,8	1,7	-3,5	-0,8	5,6
82	-5,4	-4,8	-2,9	-0,6	2,2	-4,8	-1,1	-0,6	0,6	8,3
83	8,1	4,5	3,5	-1,6	11,1	2,5	1,5	3,0	0,5	7,7
84	0,4	2,1	2,8	0,7	3,8	1,7	-3,7	-1,4	-2,5	3,4
85	1,6	-1,6	-4,2	-3,3	6,0	-	-0,8	0,8	1,6	0,8
86	2,5	-0,5	1,5	1,5	5,4	2,0	-	1,5	-0,5	6,3
87	2,0	7,7	4,5	8,5	10,2	-1,0	4,0	-1,0	2,5	10,6
88	-2,4	2,9	-0,7	-1,0	-1,0	-3,4	-2,0	-3,8	-7,9	-5,6
89	-0,6	-0,6	-2,4	3,4	3,9	1,7	5,0	3,9	3,4	10,9
90	-3,6	-4,8	-1,9	-2,3	3,3	-2,3	-1,2	-	-4,0	1,9
91	2,4	8,1	4,1	5,6	10,1	1,0	2,1	2,7	2,7	8,1
92	-0,5	3,7	2,9	3,4	4,1	-5,6	0,3	-0,8	-3,4	-4,8
93	1,6	0,5	1,6	2,1	10,8	-2,2	5,0	-	0,5	10,8
94	-4,9	-4,9	-2,6	-0,4	4,7	0,4	-4,9	0,4	-2,2	4,1
95	2,8	8,9	1,2	11,4	15,0	3,0	2,8	5,6	4,8	11,9
96	2,1	2,7	3,9	3,5	3,9	3,1	0,2	2,3	1,5	2,3

Tabela B.5: Tempo Médio das solução ILS para instâncias de 6 e 12 locações em 10 execuções de cada algoritmo

Inst.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
1	1,4	1,4	1,3	1,3	2,0	0,7	0,3	0,4	0,5	0,5
2	1,2	0,9	1,5	1,4	1,2	0,6	0,3	0,5	0,5	0,5
3	1,1	0,9	1,1	1,2	1,2	0,4	0,3	0,3	0,4	0,5
4	0,4	0,4	0,4	0,5	0,5	0,3	0,2	0,2	0,2	0,2
5	1,6	1,5	1,8	1,8	1,9	0,5	0,4	0,5	0,4	0,8
6	1,1	1,4	0,9	1,1	1,6	0,5	0,3	0,4	0,4	0,6
7	1,1	1,3	1,2	1,3	1,3	0,4	0,3	0,3	0,4	0,5
8	0,4	0,4	0,5	0,4	0,5	0,3	0,2	0,2	0,2	0,3
9	3,1	3,9	2,9	2,7	4,4	1,3	1,0	0,9	1,0	1,1
10	2,6	2,3	2,7	2,6	3,5	1,1	0,8	1,0	1,0	1,3
11	1,9	2,3	2,0	2,3	3,4	0,8	0,7	0,7	0,7	1,0
12	1,0	1,4	1,1	1,4	1,0	0,6	0,5	0,4	0,4	0,4
13	3,3	2,6	3,7	3,2	4,0	1,2	1,2	1,1	0,9	1,4
14	2,5	2,1	2,7	3,1	3,6	1,2	1,0	1,0	1,0	1,5
15	3,0	2,7	2,5	2,5	2,8	1,1	0,9	0,7	1,0	1,1
16	1,4	1,3	1,2	1,2	1,6	0,8	0,7	0,6	0,6	0,8
17	4,7	5,0	4,3	4,8	7,9	2,5	1,8	1,7	1,5	2,4
18	5,2	7,2	5,1	4,9	5,3	2,3	1,6	2,4	2,0	2,0
19	3,6	3,6	4,0	3,2	5,6	1,6	1,1	1,3	1,0	2,6
20	2,1	1,7	2,0	1,7	2,1	1,2	0,9	0,7	0,8	1,0
21	6,1	6,6	5,3	5,4	8,6	2,5	1,3	1,6	1,6	3,2
22	4,6	3,3	4,2	5,1	6,3	2,2	1,3	1,8	1,8	2,4
23	4,3	2,3	4,9	2,7	4,4	1,7	1,1	1,4	1,2	1,7
24	2,0	2,4	2,2	2,3	2,1	1,2	1,2	1,0	1,1	1,1
25	16,5	15,4	22,3	25,0	25,7	7,0	6,2	5,3	5,8	6,7
26	23,6	24,6	27,3	32,5	35,7	9,2	6,6	6,5	7,0	7,8
27	13,4	19,8	18,3	14,1	15,7	7,4	4,9	5,4	4,6	5,1
28	10,2	10,0	12,7	11,7	12,2	5,1	4,2	4,8	4,5	4,1
29	17,6	15,5	19,2	18,3	23,6	5,6	4,4	4,6	4,1	10,2
30	22,0	18,8	17,3	18,9	22,0	7,1	5,8	6,3	4,3	6,0
31	12,8	13,0	12,8	11,3	14,6	6,3	4,6	5,1	4,5	4,7
32	7,7	8,3	8,0	8,4	10,6	4,7	4,3	2,9	4,7	4,0
33	50,0	48,6	72,3	39,6	68,5	14,8	12,3	15,0	12,0	20,1
34	49,6	51,3	51,2	45,1	44,0	22,4	13,3	12,4	19,0	15,6
35	29,8	43,1	53,1	47,3	29,0	16,4	10,4	13,9	11,8	11,1
36	21,7	23,5	19,7	31,7	31,3	10,1	10,9	7,3	8,5	6,4
37	65,8	43,7	53,9	49,5	53,4	15,9	13,7	11,6	13,8	15,5
38	51,2	38,3	43,3	42,5	66,0	23,7	14,8	15,2	12,3	18,8
39	36,7	24,3	32,1	30,8	42,0	13,0	8,6	9,6	8,2	15,4
40	21,4	28,3	24,9	27,6	37,2	11,8	7,9	12,5	9,0	13,2
41	81,1	88,4	78,3	90,2	147,3	31,0	22,4	27,9	21,5	29,9
42	71,2	104,9	108,3	87,7	76,4	28,3	33,3	29,6	29,9	31,0
43	53,7	44,6	64,7	70,6	82,9	28,9	20,6	15,7	24,0	30,9
44	30,9	43,1	54,2	51,0	51,7	20,5	13,7	13,7	18,3	15,4
45	86,6	54,8	81,0	84,5	93,4	26,2	20,7	24,1	24,4	34,7
46	73,4	86,2	71,2	80,7	107,9	29,5	24,7	23,8	18,2	28,7
47	52,1	63,2	64,0	50,8	91,8	24,8	20,0	17,6	21,2	22,7
48	42,7	44,1	40,7	58,7	53,8	16,3	18,1	14,9	13,8	20,7

Tabela B.6: Tempo Médio das solução ILS para instâncias de 20 e 32 locações em 10 execuções de cada algoritmo

Inst.	ILS1	ILS2	ILS3	ILS4	ILS5	ILS6	ILS7	ILS8	ILS9	ILS10
49	110,4	87,6	128,5	94,7	164,7	26,6	24,1	22,4	39,6	34,1
50	113,6	124,2	108,5	120,3	117,2	51,9	25,5	34,1	28,6	35,7
51	77,8	65,4	105,6	98,3	135,1	21,8	17,1	18,4	22,5	29,8
52	64,2	66,2	56,7	69,0	68,6	28,5	19,1	18,9	27,6	27,8
53	125,3	112,8	131,4	154,0	164,9	31,3	29,8	22,9	24,7	46,2
54	152,2	144,4	168,9	136,1	179,1	52,6	35,0	27,6	30,8	43,8
55	102,0	132,3	112,4	116,4	98,8	32,4	25,7	22,1	29,4	31,1
56	80,1	97,7	85,9	84,3	144,8	34,7	32,3	24,4	31,2	25,9
57	342,4	323,7	196,2	246,6	390,6	79,2	51,9	54,6	50,5	80,5
58	217,0	335,4	335,4	237,0	303,5	119,5	58,4	67,1	64,6	86,7
59	201,9	178,1	156,0	206,9	254,5	84,9	40,5	46,4	57,7	62,4
60	133,7	161,8	97,7	155,8	214,1	66,2	43,7	50,4	39,3	73,9
61	293,5	336,1	299,8	319,7	503,7	107,9	63,4	63,7	73,7	126,8
62	288,9	565,8	273,2	319,5	514,4	107,7	85,0	74,1	77,0	94,4
63	245,4	183,4	188,0	301,5	287,2	75,3	53,4	58,4	52,6	76,3
64	219,1	376,8	188,6	264,0	364,8	79,9	100,2	78,6	74,7	73,6
65	529,0	540,5	453,1	717,4	673,5	151,2	136,1	127,8	91,9	249,5
66	467,0	600,5	436,3	599,3	865,9	142,2	110,6	120,8	132,3	296,1
67	283,1	336,6	371,2	395,5	466,6	132,2	87,7	76,8	122,5	182,7
68	277,5	279,7	188,9	284,6	320,7	115,2	92,6	95,7	85,0	208,5
69	523,4	655,1	506,9	603,0	817,6	188,8	144,3	114,3	131,0	296,9
70	655,4	775,2	630,7	640,9	911,9	193,0	173,4	116,7	176,9	334,2
71	582,4	616,9	391,0	656,1	430,7	148,6	121,7	165,7	136,5	317,6
72	397,9	605,5	410,4	461,4	462,3	150,0	104,4	123,9	125,4	242,1
73	654,8	981,1	568,6	652,5	916,5	192,1	141,0	145,7	142,8	336,2
74	897,2	1008,8	855,5	1050,3	975,2	199,1	275,6	215,5	198,3	328,6
75	768,6	962,0	961,8	691,8	1213,0	191,4	233,1	178,1	232,8	277,0
76	618,9	469,8	459,1	515,1	569,1	144,9	124,8	127,9	161,1	262,4
77	1128,1	678,1	747,2	697,8	1251,0	170,1	142,9	146,4	122,7	547,3
78	911,6	1160,9	1024,2	792,0	1657,2	283,3	172,0	154,4	187,5	494,6
79	890,3	989,9	787,4	658,2	1186,6	131,7	165,5	194,6	206,8	354,0
80	903,1	612,3	661,1	577,8	629,0	152,7	172,4	194,0	136,1	290,0
81	1938,7	2524,0	2593,6	2196,2	4000,5	467,7	491,7	545,7	373,5	864,0
82	2241,7	2391,1	2039,6	2117,7	2377,6	495,6	559,2	637,0	401,8	869,4
83	2486,1	2634,5	2119,2	2002,0	2834,6	599,6	866,9	616,5	406,4	1336,4
84	1182,0	2131,1	1496,4	1283,9	1396,9	392,4	496,5	393,5	511,9	728,1
85	1717,0	1966,2	2354,6	2006,4	2723,4	429,3	500,7	467,7	336,5	1258,9
86	2227,7	1888,7	2360,1	2185,8	2814,4	478,6	448,7	478,8	502,8	1372,6
87	2514,0	2883,5	2023,6	2725,7	1550,6	491,6	601,8	428,3	494,7	1234,2
88	1610,3	1408,9	1522,1	1683,3	1168,3	454,3	401,1	411,8	429,9	929,0
89	3920,0	4596,2	3691,9	3785,0	4137,3	851,2	923,3	871,2	677,1	1709,6
90	2185,5	3834,5	3856,9	2851,8	3501,8	789,3	971,2	1123,6	884,3	1869,6
91	4361,6	3853,2	3798,9	3856,4	2766,7	778,0	1271,8	1235,2	988,5	2444,5
92	3515,6	2156,3	1793,6	2814,0	1947,2	864,1	762,8	1121,3	853,7	1818,4
93	3715,1	4069,1	3856,3	4208,5	2986,0	952,6	912,7	1232,8	925,8	1850,1
94	7079,8	4990,2	5015,0	3896,7	7527,5	746,9	922,6	1142,8	1533,0	1778,7
95	3388,9	3711,5	3295,8	3323,0	2658,6	1053,0	1270,8	1085,2	1278,4	2119,5
96	2622,0	3532,0	2768,1	3542,8	3017,0	611,2	1314,9	1131,2	1195,4	1340,7

Apêndice C

Resultados Computacionais dos Algoritmos GRASP Propostos

Os algoritmos GRASP propostos no Capítulo 5, Seção 5.3.1, foram testados utilizando os valores de α iguais a 0,2, 0,4, 0,6, 0,8 e 1,0.

As Tabelas C.1, C.2, C.3, C.4 e C.5 apresentam, respectivamente, os melhores custos obtidos em 10 execuções dos algoritmos **G1** e **G2**, **G3** e **G4**, **G5** e **G6**, **G7** e **G8** e, finalmente, **G9** e **G10**, para as instâncias de 6 e 12 locações. Em negrito encontra-se destacado o melhor valor encontrado pelos algoritmos GRASP para cada instância.

As Tabelas C.6, C.7, C.8, C.9 e C.10 apresentam, respectivamente, os melhores custos obtidos em 10 execuções dos algoritmos **G1** e **G2**, **G3** e **G4**, **G5** e **G6**, **G7** e **G8** e, finalmente, **G9** e **G10**, para as instâncias de 20 e 32 locações. Em negrito encontra-se destacado o melhor valor encontrado pelos algoritmos GRASP para cada instância.

As Tabelas C.11, C.12, C.13, C.14 e C.15 apresentam, respectivamente, os desvios percentuais entre o melhor custo conhecido e o melhor custo obtido em 10 execuções dos algoritmos **G1** e **G2**, **G3** e **G4**, **G5** e **G6**, **G7** e **G8** e, finalmente, **G9** e **G10**, para as instâncias de 6 e 12 locações. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo GRASP é menor do que o da literatura.

As Tabelas C.16, C.17, C.18, C.19 e C.20 apresentam, respectivamente, os desvios percentuais entre o melhor custo reportado na literatura e o melhor custo obtido em 10 execuções dos algoritmos **G1** e **G2**, **G3** e **G4**, **G5** e **G6**, **G7** e **G8** e, finalmente, **G9** e **G10**, para as instâncias de 6 e 12 locações. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo GRASP é menor do que o da literatura.

As Tabelas C.21, C.22, C.23, C.24 e C.25 apresentam, respectivamente, o tempo médio (em segundos) de 10 execuções dos algoritmos **G1** e **G2**, **G3** e **G4**, **G5** e **G6**, **G7** e **G8** e, finalmente, **G9** e **G10**, para as instâncias de 20 e 32 locações.

As Tabelas C.26, C.27, C.28, C.29 e C.30 apresentam, respectivamente, o tempo médio (em segundos) de 10 execuções dos algoritmos G1 e G2, G3 e G4, G5 e G6, G7 e G8 e, finalmente, G9 e G10, para as instâncias de 20 e 32 locações.

Tabela C.1: Melhor solução dos algoritmos GRASP G1 e G2 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações

Inst.	G1						G2					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	16	16	16	16	16	16	16	16	16	16	16	16
2	27	27	27	27	26	26	26	26	26	26	26	26
3	18	18	18	18	18	18	18	18	18	18	18	18
4	26	26	26	26	26	26	26	26	26	26	26	26
5	16	16	16	16	16	16	16	16	16	16	16	16
6	27	27	27	27	27	27	27	27	27	27	27	27
7	16	16	16	16	16	16	16	16	16	16	16	16
8	31	31	31	31	31	31	31	31	31	31	31	31
9	26	26	26	26	25	25	26	26	26	26	25	25
10	49	49	49	46	46	46	49	49	49	46	46	46
11	33	33	33	32	32	32	33	33	33	32	32	32
12	43	43	43	43	43	43	43	43	43	43	43	43
13	29	29	29	29	28	28	29	29	29	29	28	28
14	45	45	45	45	45	45	45	45	45	45	45	45
15	36	36	36	36	35	35	36	36	36	36	35	35
16	51	51	51	49	49	49	51	51	51	49	49	49
17	35	35	35	35	35	35	35	35	35	35	35	35
18	64	64	64	62	62	62	64	64	64	62	62	62
19	47	47	47	47	46	46	47	47	47	47	46	46
20	63	63	63	63	63	63	63	63	63	63	63	63
21	48	48	48	48	48	48	48	48	48	48	48	48
22	67	67	67	67	67	67	67	67	67	67	67	67
23	56	56	56	56	56	56	56	56	56	56	56	56
24	74	74	74	74	74	74	74	74	74	74	74	74
25	31	31	31	31	31	31	31	31	31	31	31	31
26	44	44	43	43	43	43	44	44	43	43	43	43
27	43	43	43	43	43	43	43	43	43	43	43	43
28	55	55	55	55	55	55	55	55	55	55	55	55
29	29	29	29	29	29	29	29	29	29	29	29	29
30	52	52	50	49	49	49	52	52	50	49	49	49
31	44	44	44	42	42	42	44	44	44	42	42	42
32	69	69	69	69	69	69	69	69	69	69	69	69
33	57	57	54	53	52	52	57	57	54	53	52	52
34	77	77	73	72	72	72	77	77	73	72	72	72
35	74	74	74	73	73	73	74	74	74	73	73	73
36	100	100	96	95	95	95	100	100	96	95	95	95
37	53	53	48	48	48	48	53	53	48	48	48	48
38	84	84	84	83	83	83	84	84	84	83	83	83
39	72	72	72	69	69	69	72	72	72	69	69	69
40	111	111	108	108	108	108	111	111	108	108	108	108
41	88	88	78	78	78	78	88	88	78	78	78	78
42	112	112	103	102	102	102	112	112	103	102	102	102
43	114	114	110	110	110	110	114	114	110	110	110	110
44	140	140	140	140	140	140	140	140	140	140	140	140
45	73	73	66	66	66	66	73	73	66	66	66	66
46	124	124	118	116	116	116	124	124	118	116	116	116
47	121	121	119	115	115	115	121	121	119	115	115	115
48	176	176	171	171	171	171	176	176	171	171	171	171

Tabela C.2: Melhor solução dos algoritmos GRASP G3 e G4 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações

Inst.	G3						G4					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	16	16	16	16	16	16	16	16	16	16	16	16
2	27	27	27	26	26	26	27	27	27	26	26	26
3	18	18	18	18	18	18	18	18	18	18	18	18
4	26	26	26	26	26	26	26	26	26	26	26	26
5	16	16	16	16	16	16	16	16	16	16	16	16
6	27	27	27	27	27	27	27	27	27	27	27	27
7	16	16	16	16	16	16	16	16	16	16	16	16
8	31	31	31	31	31	31	31	31	31	31	31	31
9	25	25	25	25	25	25	25	25	25	25	25	25
10	46	46	46	46	46	46	46	46	46	46	46	46
11	32	32	32	32	32	32	32	32	32	32	32	32
12	43	43	43	43	43	43	43	43	43	43	43	43
13	28	28	28	28	28	28	28	28	28	28	28	28
14	45	45	45	45	45	45	45	45	45	45	45	45
15	35	35	35	35	35	35	35	35	35	35	35	35
16	49	49	49	49	49	49	49	49	49	49	49	49
17	35	35	35	35	35	35	35	35	35	35	35	35
18	63	63	63	62	62	62	63	63	63	62	62	62
19	46	46	46	46	46	46	46	46	46	46	46	46
20	63	63	63	63	63	63	63	63	63	63	63	63
21	48	48	48	48	48	48	48	48	48	48	48	48
22	67	67	67	67	67	67	67	67	67	67	67	67
23	56	56	56	56	56	56	56	56	56	56	56	56
24	74	74	74	74	74	74	74	74	74	74	74	74
25	31	31	31	31	31	31	31	31	31	31	31	31
26	43	43	43	43	43	43	43	43	43	43	43	43
27	43	43	43	43	43	43	43	43	43	43	43	43
28	55	55	55	55	55	55	55	55	55	55	55	55
29	29	29	29	29	29	29	29	29	29	29	29	29
30	49	49	49	49	49	49	49	49	49	49	49	49
31	43	43	42	42	42	42	43	43	42	42	42	42
32	69	69	69	69	69	69	69	69	69	69	69	69
33	54	54	52	52	52	52	54	54	52	52	52	52
34	74	74	72	72	72	72	74	74	72	72	72	72
35	73	73	73	73	73	73	73	73	73	73	73	73
36	95	95	95	95	95	95	95	95	95	95	95	95
37	49	49	48	48	48	48	49	49	48	48	48	48
38	86	86	83	83	83	83	86	86	83	83	83	83
39	72	72	72	69	69	69	72	72	72	69	69	69
40	110	110	108	108	108	108	110	110	108	108	108	108
41	78	78	78	78	78	78	78	78	78	78	78	78
42	104	104	102	102	102	102	104	104	102	102	102	102
43	110	110	110	110	110	110	110	110	110	110	110	110
44	140	140	140	140	140	140	140	140	140	140	140	140
45	67	67	66	66	66	66	67	67	66	66	66	66
46	119	119	116	116	116	116	119	119	116	116	116	116
47	118	118	119	115	115	115	118	118	119	115	115	115
48	172	172	171	171	171	171	172	172	171	171	171	171

Tabela C.3: Melhor solução dos algoritmos GRASP G5 e G6 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações

Inst.	G5						G6					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	16	16	16	16	16	16	16	16	16	16	16	16
2	27	26	26	26	26	26	25	25	25	25	25	25
3	18	18	18	18	18	18	19	19	19	19	18	18
4	26	26	25	26	26	26	27	27	27	27	25	25
5	16	16	16	16	16	16	17	17	17	17	16	16
6	27	27	27	27	27	27	30	30	30	30	27	27
7	16	16	16	16	16	16	17	17	17	17	16	16
8	31	31	31	31	31	31	31	31	31	31	31	31
9	25	25	25	25	25	25	35	35	35	27	25	25
10	46	46	46	46	46	46	51	51	51	48	46	46
11	32	32	32	32	32	32	38	38	38	38	32	32
12	43	42	41	43	43	43	47	47	47	47	43	41
13	28	28	28	28	28	28	31	31	31	31	28	28
14	45	45	45	45	45	45	49	49	49	49	45	45
15	35	35	35	35	35	35	39	39	39	39	35	35
16	49	49	49	49	49	49	54	54	54	54	49	49
17	35	35	35	35	35	35	50	50	50	39	35	35
18	63	63	63	62	62	62	69	69	69	62	60	62
19	46	46	46	46	46	46	56	56	56	56	46	46
20	62	61	62	63	63	63	69	69	69	69	60	60
21	48	48	48	48	48	48	49	49	49	49	46	47
22	67	67	67	67	67	67	75	75	75	75	67	67
23	56	56	56	56	56	56	60	60	60	60	55	55
24	74	74	74	74	74	74	79	79	79	79	74	74
25	31	31	31	31	31	31	36	36	35	31	31	31
26	43	43	43	43	43	43	63	63	54	47	43	43
27	43	43	43	43	43	43	47	47	47	45	43	43
28	55	55	55	55	55	55	75	75	65	58	55	55
29	29	29	29	29	29	29	35	35	31	29	29	29
30	49	49	49	49	49	49	60	60	56	49	49	50
31	43	43	42	42	42	42	56	56	47	44	43	42
32	69	69	69	69	69	69	81	81	74	74	71	69
33	54	54	53	52	52	52	71	71	60	58	58	55
34	74	74	72	72	72	72	100	100	81	79	80	79
35	73	73	73	73	73	73	90	90	89	82	79	78
36	95	95	95	95	95	95	123	123	109	104	98	97
37	49	49	48	48	48	48	76	76	55	52	53	52
38	86	86	83	83	83	83	111	111	96	87	88	86
39	72	72	69	69	69	69	91	91	85	77	75	73
40	110	110	108	108	108	108	134	134	118	117	112	111
41	78	78	78	78	78	78	108	108	90	89	87	88
42	104	104	102	102	102	102	146	146	118	115	113	114
43	110	110	110	110	110	110	136	136	135	121	114	114
44	140	140	140	140	140	140	190	190	165	160	147	146
45	67	67	66	66	66	66	110	110	82	79	75	75
46	119	119	116	116	116	116	147	147	130	122	122	124
47	118	118	117	115	115	115	156	156	135	124	126	124
48	172	172	171	171	171	171	217	217	201	187	181	179

Tabela C.4: Melhor solução dos algoritmos GRASP G7 e G8 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações

Inst.	G7						G8					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	17	17	17	17	16	16	16	16	16	16	16	16
2	28	28	28	28	26	26	25	25	25	25	25	25
3	19	19	19	19	18	18	18	18	18	18	18	18
4	27	27	27	27	25	25	25	25	25	25	25	25
5	17	17	17	17	16	16	16	16	16	16	16	16
6	33	33	33	31	27	27	29	29	29	27	27	27
7	17	17	17	17	16	16	16	16	16	16	16	16
8	33	33	33	31	31	31	31	31	31	31	31	31
9	37	37	37	28	25	25	25	25	25	25	25	25
10	52	52	52	51	46	46	49	49	49	46	46	46
11	38	38	38	38	32	32	32	32	32	32	32	32
12	47	47	47	47	43	41	41	41	41	41	41	41
13	32	32	32	32	28	28	28	28	28	28	28	28
14	50	50	50	50	45	45	48	48	48	45	45	45
15	39	39	39	39	35	35	35	35	35	35	35	35
16	56	56	56	54	49	49	49	49	49	49	49	49
17	49	49	49	41	37	35	41	41	41	35	35	35
18	74	74	74	68	63	62	67	67	67	61	61	60
19	56	56	56	56	46	46	48	48	48	47	46	46
20	70	70	70	70	61	61	60	60	60	60	60	60
21	55	55	55	55	49	49	49	49	49	47	46	46
22	74	74	74	74	67	67	72	72	72	67	67	67
23	62	62	62	62	57	57	55	55	55	55	55	55
24	81	81	81	81	74	74	74	74	74	74	74	74
25	43	43	40	35	32	31	35	35	33	32	31	31
26	64	64	56	51	46	45	50	50	45	45	44	45
27	47	47	47	45	43	43	44	44	43	43	43	43
28	75	75	65	58	55	55	60	60	55	55	55	55
29	34	34	32	30	29	29	30	30	29	29	29	29
30	61	61	57	52	50	50	52	52	50	49	50	50
31	56	56	47	46	44	42	48	48	45	43	43	42
32	81	81	74	74	71	69	74	74	71	69	72	69
33	77	77	70	62	59	59	56	56	55	57	56	56
34	108	108	90	86	79	81	84	84	75	75	76	75
35	90	90	89	82	79	78	87	87	82	80	77	78
36	123	123	109	104	98	97	106	106	100	104	97	98
37	82	82	64	59	56	55	62	62	53	50	50	50
38	115	115	101	92	89	86	91	91	87	87	85	85
39	90	90	84	78	77	74	76	76	75	76	73	73
40	135	135	118	118	115	111	115	115	115	111	111	110
41	117	117	105	96	91	89	91	91	88	87	87	88
42	151	151	127	124	124	118	120	120	112	111	113	111
43	136	136	135	121	114	119	133	133	126	121	118	119
44	190	190	165	160	147	146	160	160	148	158	149	146
45	123	123	96	85	82	80	88	88	80	79	74	78
46	167	167	143	128	128	126	122	122	120	123	121	125
47	157	157	137	125	126	122	140	140	134	124	124	117
48	217	217	201	187	181	180	186	186	189	181	182	177

Tabela C.5: Melhor solução dos algoritmos GRASP G9 e G10 (obtida de 10 execuções) com variação de alfa para instâncias de 6 e 12 locações

Inst.	G9						G10					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	16	16	16	16	16	16	19	17	17	17	17	17
2	27	27	27	27	26	26	27	25	25	25	25	25
3	18	18	18	18	18	18	20	19	18	18	18	18
4	25	25	25	25	25	25	26	25	25	25	25	25
5	16	16	16	16	16	16	22	17	17	17	16	17
6	29	29	29	27	27	27	31	29	29	27	28	28
7	16	16	16	16	16	16	19	16	16	16	16	16
8	31	31	31	31	31	31	32	31	31	31	31	31
9	28	28	28	25	25	25	30	29	29	31	32	31
10	49	49	49	46	46	46	55	50	49	48	46	47
11	32	32	32	32	32	32	34	34	33	33	32	33
12	42	42	42	41	41	41	43	41	41	41	41	41
13	28	28	28	28	28	28	34	30	32	33	33	32
14	48	48	48	45	45	45	51	49	49	47	47	46
15	35	35	35	35	35	35	38	36	35	35	37	36
16	49	49	49	49	49	49	50	49	49	49	49	49
17	42	42	42	37	37	35	46	43	43	46	44	46
18	69	69	69	64	64	62	71	67	67	65	66	66
19	48	48	48	47	46	46	51	49	49	51	48	47
20	63	63	63	61	61	61	62	60	60	60	60	60
21	49	49	49	49	49	49	53	49	49	53	53	50
22	71	71	71	67	67	67	73	71	73	70	68	69
23	57	57	57	57	57	57	58	56	56	60	57	55
24	74	74	74	74	74	74	74	74	74	74	74	74
25	39	39	34	32	31	31	41	45	44	41	46	45
26	51	51	47	44	45	46	54	56	50	50	52	53
27	44	44	43	43	43	43	52	44	48	46	45	45
28	60	60	55	55	55	55	65	60	55	55	55	55
29	30	30	29	29	29	29	40	36	35	39	35	42
30	54	54	51	50	50	51	54	57	56	55	56	57
31	48	48	46	45	43	43	50	52	48	44	44	45
32	74	74	71	69	72	69	75	73	72	69	69	69
33	62	62	58	57	57	60	72	72	77	79	80	81
34	85	85	77	78	79	77	89	89	87	91	90	90
35	87	87	82	80	77	78	88	88	85	84	86	83
36	106	106	100	104	97	98	111	106	98	97	95	97
37	66	66	57	52	53	52	73	73	71	76	76	77
38	93	93	89	89	88	86	94	95	100	98	99	101
39	78	78	79	78	74	75	84	89	88	87	79	78
40	115	115	115	111	114	110	117	117	115	111	110	111
41	97	97	90	87	91	92	102	105	110	115	118	119
42	123	123	115	114	117	114	128	127	128	125	136	133
43	134	134	126	121	118	119	136	133	130	125	129	129
44	160	160	148	158	149	146	165	160	151	144	143	142
45	96	96	84	84	82	79	104	105	107	114	119	117
46	134	134	123	127	127	128	139	139	140	144	141	137
47	140	140	137	125	126	119	149	147	145	134	131	130
48	188	188	191	182	184	180	188	188	184	178	177	174

Tabela C.6: Melhor solução dos algoritmos GRASP G1 e G2 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações

Inst.	G1						G2					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	46	44	44	44	44	44	46	44	44	44	44	44
50	66	63	60	60	60	60	66	63	60	60	60	60
51	58	57	53	53	54	53	58	57	53	53	54	53
52	93	89	89	89	89	89	93	89	89	89	89	89
53	50	47	47	47	47	47	50	47	47	47	47	47
54	66	66	63	63	62	62	66	66	63	63	62	62
55	64	63	61	61	60	60	64	63	61	61	60	60
56	95	92	89	89	89	89	95	92	89	89	89	89
57	71	68	66	66	66	66	71	68	66	66	66	66
58	110	106	101	101	101	100	110	106	101	101	101	100
59	112	106	95	93	95	93	112	106	95	93	95	93
60	174	161	153	153	153	153	174	161	153	153	153	153
61	86	78	74	74	74	74	86	78	74	74	74	74
62	121	114	113	113	113	113	121	114	113	113	113	113
63	126	124	118	118	118	117	126	124	118	118	118	117
64	192	180	178	175	177	176	192	180	178	175	177	176
65	114	107	101	103	103	102	114	107	101	103	103	102
66	164	160	153	152	153	152	164	160	153	152	153	152
67	163	156	148	150	147	148	163	156	148	150	147	148
68	249	229	220	218	218	218	249	229	220	218	218	218
69	128	119	115	116	117	117	128	119	115	116	117	117
70	179	175	170	171	169	167	179	175	170	171	169	167
71	179	178	167	166	165	169	179	178	167	166	165	169
72	277	259	253	248	252	255	277	259	253	248	252	255
73	72	71	71	71	71	71	72	71	71	71	71	71
74	96	91	92	91	91	91	96	91	92	91	91	91
75	126	104	104	104	103	104	126	104	104	104	103	104
76	154	145	145	146	146	144	154	145	145	146	146	144
77	70	70	70	70	70	70	70	70	70	70	70	70
78	99	94	93	93	93	93	99	94	93	93	93	93
79	115	102	101	101	101	100	115	102	101	101	101	100
80	185	165	163	165	164	165	185	165	163	165	164	165
81	122	113	111	112	113	114	122	113	111	112	113	114
82	189	168	169	166	170	168	189	168	169	166	170	168
83	219	195	189	191	192	194	219	195	189	191	192	194
84	297	272	273	271	277	273	297	272	273	271	277	273
85	134	121	122	123	122	124	134	121	122	123	122	124
86	214	186	187	185	190	185	214	186	187	185	190	185
87	211	191	190	188	186	191	211	191	190	188	186	191
88	328	280	285	286	283	288	328	280	285	286	283	288
89	205	168	164	174	169	168	205	168	164	174	169	168
90	281	254	261	257	256	257	281	254	261	257	256	257
91	314	282	283	286	279	285	314	282	283	286	279	285
92	425	392	391	394	396	391	425	392	391	394	396	391
93	214	186	187	189	188	186	214	186	187	189	188	186
94	307	282	281	282	283	284	307	282	281	282	283	284
95	362	320	321	321	319	318	362	320	321	321	319	318
96	501	458	450	465	459	460	501	458	450	465	459	460

Tabela C.7: Melhor solução dos algoritmos GRASP G3 e G4 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações

Inst.	G3						G4					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	44	44	44	44	44	44	44	44	44	44	44	44
50	61	60	60	60	60	60	61	60	60	60	60	60
51	55	55	53	53	53	53	55	55	53	53	53	53
52	93	89	89	89	89	89	93	89	89	89	89	89
53	47	47	47	47	47	47	47	47	47	47	47	47
54	63	63	63	62	63	62	63	63	63	62	63	62
55	61	61	60	61	60	60	61	61	60	61	60	60
56	91	91	89	89	89	89	91	91	89	89	89	89
57	67	66	66	66	66	66	67	66	66	66	66	66
58	101	101	100	101	100	101	101	101	100	101	100	101
59	99	105	96	95	94	95	99	105	96	95	94	95
60	174	158	153	153	153	153	174	158	153	153	153	153
61	75	75	74	74	74	74	75	75	74	74	74	74
62	116	113	113	113	113	113	116	113	113	113	113	113
63	126	119	118	117	119	117	126	119	118	117	119	117
64	183	180	175	176	176	177	183	180	175	176	176	177
65	101	101	102	101	101	103	101	101	102	101	101	103
66	156	153	155	153	154	152	156	153	155	153	154	152
67	163	156	147	148	147	149	163	156	147	148	147	149
68	249	229	222	220	220	218	249	229	222	220	220	218
69	116	116	117	116	118	117	116	116	117	116	118	117
70	172	170	171	171	169	170	172	170	171	171	169	170
71	174	173	168	169	169	166	174	173	168	169	169	166
72	264	254	253	249	248	248	264	254	253	249	248	248
73	71	71	71	71	71	71	71	71	71	71	71	71
74	92	91	91	92	91	90	92	91	91	92	91	90
75	105	103	104	103	104	103	105	103	104	103	104	103
76	147	147	146	145	147	145	147	147	146	145	147	145
77	70	70	70	70	70	70	70	70	70	70	70	70
78	97	94	95	94	93	94	97	94	95	94	93	94
79	115	101	100	100	101	101	115	101	100	100	101	101
80	185	163	166	164	165	165	185	163	166	164	165	165
81	116	112	110	114	113	114	116	112	110	114	113	114
82	171	169	165	166	170	169	171	169	165	166	170	169
83	192	195	194	194	187	193	192	195	194	194	187	193
84	281	273	272	274	270	273	281	273	272	274	270	273
85	123	123	122	124	122	124	123	123	122	124	122	124
86	188	185	186	185	189	185	188	185	186	185	189	185
87	211	192	186	193	194	193	211	192	186	193	194	193
88	328	280	286	287	284	279	328	280	286	287	284	279
89	177	171	172	172	171	170	177	171	172	172	171	170
90	253	252	254	251	253	257	253	252	254	251	253	257
91	291	287	280	284	281	279	291	287	280	284	281	279
92	399	390	389	385	385	394	399	390	389	385	385	394
93	192	189	185	188	183	186	192	189	185	188	183	186
94	280	286	281	284	283	272	280	286	281	284	283	272
95	362	317	313	318	313	322	362	317	313	318	313	322
96	501	464	458	461	454	463	501	464	458	461	454	463

Tabela C.8: Melhor solução dos algoritmos GRASP G5 e G6 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações

	G5						G6					
49	44	44	44	44	44	44	48	46	44	44	44	44
50	61	60	60	60	60	60	73	72	64	63	63	62
51	55	55	53	53	53	53	76	66	58	58	60	56
52	93	89	89	89	89	89	117	114	100	96	95	92
53	47	47	47	47	47	47	56	51	48	48	47	47
54	63	63	63	62	62	63	87	78	67	67	68	66
55	61	61	61	60	60	60	85	73	64	67	64	65
56	91	91	89	89	89	89	115	106	104	101	97	96
57	67	66	66	66	66	66	81	77	71	72	74	70
58	101	101	101	100	101	101	119	116	109	109	108	107
59	99	106	94	94	94	93	138	120	105	99	107	103
60	174	155	153	153	153	153	181	172	166	161	166	163
61	75	74	74	74	75	74	108	93	87	85	86	84
62	116	113	113	113	112	113	163	140	124	131	129	128
63	126	122	118	119	118	117	153	153	135	132	133	133
64	183	177	175	176	175	176	239	213	204	192	198	193
65	101	102	103	101	102	102	137	119	115	113	115	108
66	156	153	153	153	153	154	177	177	171	167	164	169
67	163	156	148	147	149	149	194	194	166	163	166	163
68	249	229	220	218	219	219	279	265	247	243	237	236
69	116	117	116	116	116	117	163	141	135	129	132	132
70	172	172	168	169	167	170	230	210	199	192	192	187
71	174	178	168	167	169	169	229	204	194	191	196	196
72	264	253	256	248	252	252	326	298	285	290	287	279
73	71	71	71	71	71	71	93	77	74	75	75	73
74	92	91	91	91	92	91	133	109	110	108	108	107
75	104	103	103	104	104	103	154	128	128	121	122	124
76	147	147	146	145	147	144	213	190	177	166	172	171
77	70	70	70	70	70	70	97	76	74	73	72	73
78	97	94	93	93	94	93	136	117	114	113	114	112
79	115	102	100	100	101	101	169	136	125	129	132	126
80	185	165	163	164	164	163	266	204	205	206	197	196
81	115	113	114	114	113	113	179	141	145	138	143	141
82	170	170	165	167	167	168	246	210	196	210	210	204
83	192	195	192	192	193	197	295	247	220	236	245	242
84	281	271	275	271	267	271	416	330	333	335	326	329
85	123	124	124	122	125	125	174	156	153	153	149	150
86	187	185	184	188	184	190	256	226	230	225	234	222
87	211	192	192	185	189	189	297	248	241	243	242	235
88	328	283	283	290	280	283	397	351	364	348	351	340
89	177	173	173	171	169	171	280	220	208	220	211	214
90	257	244	253	256	251	256	408	319	319	315	305	314
91	291	281	284	281	283	280	388	341	356	350	354	342
92	399	391	391	387	389	392	546	469	459	469	464	454
93	191	190	189	186	187	186	278	245	240	233	234	231
94	285	280	283	284	284	283	372	336	330	336	337	325
95	362	316	323	318	320	321	473	412	409	410	392	400
96	501	462	455	454	460	457	647	579	572	565	564	547

Tabela C.9: Melhor solução dos algoritmos GRASP G7 e G8 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações

Inst.	G7						G8					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	65	61	52	52	50	47	45	44	44	44	44	44
50	87	78	74	71	71	73	63	65	63	63	64	64
51	78	77	65	63	67	65	63	61	59	60	60	59
52	119	113	101	100	96	95	117	97	100	96	95	93
53	77	69	59	60	54	52	49	48	47	48	47	47
54	109	90	81	78	73	76	67	66	67	66	67	66
55	95	77	69	70	71	69	69	66	67	66	66	65
56	115	107	105	100	104	101	100	98	96	101	102	95
57	120	120	89	94	89	90	74	70	71	71	71	69
58	137	137	127	123	121	124	105	105	107	105	107	102
59	143	131	122	115	111	112	107	108	105	103	101	104
60	194	184	171	168	168	167	181	172	164	164	168	163
61	147	129	108	107	104	107	88	83	84	83	83	85
62	201	175	147	153	150	148	132	125	129	127	125	131
63	176	172	159	150	154	145	143	136	135	134	130	133
64	232	213	212	205	195	195	187	198	204	196	197	190
65	171	170	143	143	142	142	114	108	110	110	115	112
66	222	219	198	194	184	187	172	163	165	162	162	167
67	208	194	175	179	172	175	194	194	162	163	160	157
68	294	277	242	244	243	238	279	243	244	236	242	234
69	206	201	166	166	163	159	137	132	131	127	132	131
70	302	258	225	228	223	218	187	185	191	187	190	190
71	269	243	226	221	218	203	207	202	199	198	195	192
72	333	298	303	286	284	287	279	283	294	283	280	282
73	108	92	88	88	86	87	78	74	73	74	73	75
74	155	133	125	127	126	124	109	98	99	101	103	104
75	170	152	144	134	134	137	120	119	125	127	127	118
76	209	188	175	176	180	160	170	175	180	173	175	176
77	139	103	95	83	86	87	78	74	73	74	72	73
78	178	140	140	136	133	134	113	110	106	110	110	112
79	189	155	145	149	142	137	169	134	132	126	128	118
80	269	205	214	208	203	199	266	204	200	201	204	196
81	234	194	187	176	164	177	151	142	136	139	135	134
82	310	255	247	243	243	240	213	200	198	196	197	200
83	318	271	271	261	276	270	234	249	246	248	244	239
84	390	348	331	342	346	337	341	335	323	335	334	333
85	247	204	203	198	194	197	153	145	144	148	152	150
86	328	272	287	267	268	260	209	216	223	226	223	226
87	316	273	268	268	268	274	297	249	247	244	244	234
88	412	363	362	355	362	348	397	357	346	348	337	336
89	349	280	274	273	261	263	226	206	208	206	203	208
90	433	361	364	354	358	340	301	289	289	300	296	303
91	451	390	386	390	393	386	349	342	348	350	346	345
92	567	471	462	476	458	461	486	471	471	463	452	456
93	401	318	306	305	299	292	237	228	230	228	227	228
94	484	411	421	403	394	395	327	325	328	322	322	333
95	512	443	443	440	439	426	473	416	394	381	391	396
96	686	606	585	580	578	576	647	587	554	546	548	541

Tabela C.10: Melhor solução dos algoritmos GRASP G9 e G10 (obtida de 10 execuções) com variação de alfa para instâncias de 20 e 32 locações

Inst.	G9						G10					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	47	44	51	51	52	48	59	62	67	73	77	77
50	71	70	68	68	72	73	84	78	80	78	79	78
51	68	66	65	64	64	66	78	70	69	72	73	68
52	119	95	101	100	96	95	119	102	99	94	94	93
53	58	48	53	51	53	57	68	66	68	75	83	79
54	83	74	73	74	75	77	80	85	86	90	87	88
55	76	73	72	71	73	70	76	76	70	82	77	81
56	99	102	103	105	103	98	104	101	103	98	101	99
57	85	81	86	86	89	90	90	96	116	127	134	133
58	118	115	113	117	120	121	133	128	138	143	135	137
59	124	125	122	116	117	110	131	131	123	133	138	131
60	194	179	174	166	173	168	211	174	170	170	163	164
61	97	98	92	101	103	107	116	116	138	147	150	151
62	157	146	141	147	146	139	144	153	160	168	169	164
63	155	148	149	151	148	148	154	155	164	152	154	156
64	198	201	209	201	202	195	206	200	208	196	194	191
65	140	128	137	134	141	139	139	155	182	199	208	211
66	190	178	185	183	183	185	199	197	205	216	220	215
67	208	194	178	177	176	180	211	198	196	199	205	202
68	294	250	247	238	246	238	293	256	248	238	237	238
69	160	149	159	157	158	158	169	178	204	211	231	235
70	225	212	212	207	219	217	225	221	232	246	243	243
71	234	225	222	220	219	214	227	228	235	233	232	230
72	283	293	307	294	291	285	286	294	295	289	271	278
73	96	88	84	85	82	86	106	110	118	127	136	138
74	122	112	111	116	119	114	122	130	130	139	134	130
75	137	135	144	140	136	139	160	150	154	150	154	154
76	170	176	182	178	180	177	187	180	173	173	169	172
77	104	80	80	82	85	87	114	95	110	121	137	145
78	127	121	127	122	120	126	151	137	134	146	149	157
79	189	152	149	143	143	145	235	146	157	145	155	153
80	269	205	208	208	207	196	272	211	197	197	196	195
81	194	167	162	163	171	170	205	207	223	249	247	260
82	238	214	213	219	217	223	232	259	263	269	272	280
83	250	271	278	262	267	260	267	298	301	294	307	293
84	349	348	334	342	344	345	360	353	343	327	325	322
85	180	176	177	186	184	188	199	213	240	261	275	278
86	250	244	243	246	261	259	268	262	291	297	290	306
87	316	273	276	257	259	266	355	297	298	284	300	287
88	412	364	355	364	350	360	463	365	340	351	339	333
89	267	246	251	251	246	245	279	290	322	370	394	398
90	325	319	322	324	326	336	325	363	371	375	398	388
91	369	388	386	388	392	389	420	438	427	436	417	420
92	493	483	467	477	465	470	587	479	467	456	453	463
93	297	279	282	284	289	295	299	319	362	400	419	432
94	371	371	371	376	375	380	382	392	413	438	455	454
95	512	448	434	446	429	434	553	481	476	472	475	441
96	686	614	598	553	550	566	686	587	535	548	539	537

Tabela C.11: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G1 e G2 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações

Inst.	G1						G2					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	-	-	-	-	-	-	-	-	-	-	-	-
2	7,4	7,4	7,4	7,4	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8
3	-	-	-	-	-	-	-	-	-	-	-	-
4	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8
5	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	3,8	3,8	3,8	3,8	-	-	3,8	3,8	3,8	3,8	-	-
10	6,1	6,1	6,1	-	-	-	6,1	6,1	6,1	-	-	-
11	3,0	3,0	3,0	-	-	-	3,0	3,0	3,0	-	-	-
12	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7
13	3,4	3,4	3,4	3,4	-	-	3,4	3,4	3,4	3,4	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-
15	2,8	2,8	2,8	2,8	-	-	2,8	2,8	2,8	2,8	-	-
16	3,9	3,9	3,9	-	-	-	3,9	3,9	3,9	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-
18	6,3	6,3	6,3	3,2	3,2	3,2	6,3	6,3	6,3	3,2	3,2	3,2
19	2,1	2,1	2,1	2,1	-	-	2,1	2,1	2,1	2,1	-	-
20	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8
21	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2
22	-	-	-	-	-	-	-	-	-	-	-	-
23	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8
24	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-
26	2,3	2,3	-	-	-	-	2,3	2,3	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-
30	3,8	3,8	-	-2,0	-2,0	-2,0	3,8	3,8	-	-2,0	-2,0	-2,0
31	4,5	4,5	4,5	-	-	-	4,5	4,5	4,5	-	-	-
32	-	-	-	-	-	-	-	-	-	-	-	-
33	7,0	7,0	1,9	-	-1,9	-1,9	7,0	7,0	1,9	-	-1,9	-1,9
34	6,5	6,5	1,4	-	-	-	6,5	6,5	1,4	-	-	-
35	8,1	8,1	8,1	6,8	6,8	6,8	8,1	8,1	8,1	6,8	6,8	6,8
36	10,0	10,0	6,3	5,3	5,3	5,3	10,0	10,0	6,3	5,3	5,3	5,3
37	11,3	11,3	2,1	2,1	2,1	2,1	11,3	11,3	2,1	2,1	2,1	2,1
38	8,3	8,3	8,3	7,2	7,2	7,2	8,3	8,3	8,3	7,2	7,2	7,2
39	6,9	6,9	6,9	2,9	2,9	2,9	6,9	6,9	6,9	2,9	2,9	2,9
40	6,3	6,3	3,7	3,7	3,7	3,7	6,3	6,3	3,7	3,7	3,7	3,7
41	11,4	11,4	-	-	-	-	11,4	11,4	-	-	-	-
42	7,1	7,1	-1,0	-2,0	-2,0	-2,0	7,1	7,1	-1,0	-2,0	-2,0	-2,0
43	3,5	3,5	-	-	-	-	3,5	3,5	-	-	-	-
44	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1
45	9,6	9,6	-	-	-	-	9,6	9,6	-	-	-	-
46	10,5	10,5	5,9	4,3	4,3	4,3	10,5	10,5	5,9	4,3	4,3	4,3
47	8,3	8,3	6,7	3,5	3,5	3,5	8,3	8,3	6,7	3,5	3,5	3,5
48	4,0	4,0	1,2	1,2	1,2	1,2	4,0	4,0	1,2	1,2	1,2	1,2

Tabela C.12: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G3 e G4 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações

Inst.	G3						G4					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	-	-	-	-	-	-	-	-	-	-	-	-
2	7,4	7,4	7,4	3,8	3,8	3,8	7,4	7,4	7,4	3,8	3,8	3,8
3	-	-	-	-	-	-	-	-	-	-	-	-
4	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8
5	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-
12	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7
13	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-
18	4,8	4,8	4,8	3,2	3,2	3,2	4,8	4,8	4,8	3,2	3,2	3,2
19	-	-	-	-	-	-	-	-	-	-	-	-
20	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8
21	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2
22	-	-	-	-	-	-	-	-	-	-	-	-
23	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8
24	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-
30	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0
31	2,3	2,3	-	-	-	-	2,3	2,3	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-	-	-
33	1,9	1,9	-1,9	-1,9	-1,9	-1,9	1,9	1,9	-1,9	-1,9	-1,9	-1,9
34	2,7	2,7	-	-	-	-	2,7	2,7	-	-	-	-
35	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8
36	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3
37	4,1	4,1	2,1	2,1	2,1	2,1	4,1	4,1	2,1	2,1	2,1	2,1
38	10,5	10,5	7,2	7,2	7,2	7,2	10,5	10,5	7,2	7,2	7,2	7,2
39	6,9	6,9	6,9	2,9	2,9	2,9	6,9	6,9	6,9	2,9	2,9	2,9
40	5,5	5,5	3,7	3,7	3,7	3,7	5,5	5,5	3,7	3,7	3,7	3,7
41	-	-	-	-	-	-	-	-	-	-	-	-
42	-	-	-2,0	-2,0	-2,0	-2,0	-	-	-2,0	-2,0	-2,0	-2,0
43	-	-	-	-	-	-	-	-	-	-	-	-
44	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1
45	1,5	1,5	-	-	-	-	1,5	1,5	-	-	-	-
46	6,7	6,7	4,3	4,3	4,3	4,3	6,7	6,7	4,3	4,3	4,3	4,3
47	5,9	5,9	6,7	3,5	3,5	3,5	5,9	5,9	6,7	3,5	3,5	3,5
48	1,7	1,7	1,2	1,2	1,2	1,2	1,7	1,7	1,2	1,2	1,2	1,2

Tabela C.13: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G5 e G6 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações

Inst.	G5						G6					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	-	-	-	-	-	-	-	-	-	-	-	-
2	7,4	3,8	3,8	3,8	3,8	3,8	-	-	-	-	-	-
3	-	-	-	-	-	-	5,3	5,3	5,3	5,3	-	-
4	3,8	3,8	-	3,8	3,8	3,8	7,4	7,4	7,4	7,4	-	-
5	-	-	-	-	-	-	5,9	5,9	5,9	5,9	-	-
6	-	-	-	-	-	-	10,0	10,0	10,0	10,0	-	-
7	-	-	-	-	-	-	5,9	5,9	5,9	5,9	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	28,6	28,6	28,6	7,4	-	-
10	-	-	-	-	-	-	9,8	9,8	9,8	4,2	-	-
11	-	-	-	-	-	-	15,8	15,8	15,8	15,8	-	-
12	4,7	2,4	-	4,7	4,7	4,7	12,8	12,8	12,8	12,8	4,7	-
13	-	-	-	-	-	-	9,7	9,7	9,7	9,7	-	-
14	-	-	-	-	-	-	8,2	8,2	8,2	8,2	-	-
15	-	-	-	-	-	-	10,3	10,3	10,3	10,3	-	-
16	-	-	-	-	-	-	9,3	9,3	9,3	9,3	-	-
17	-	-	-	-	-	-	30,0	30,0	30,0	10,3	-	-
18	4,8	4,8	4,8	3,2	3,2	3,2	13,0	13,0	13,0	3,2	-	3,2
19	-	-	-	-	-	-	17,9	17,9	17,9	17,9	-	-
20	3,2	1,6	3,2	4,8	4,8	4,8	13,0	13,0	13,0	13,0	-	-
21	4,2	4,2	4,2	4,2	4,2	4,2	6,1	6,1	6,1	6,1	-	2,1
22	-	-	-	-	-	-	10,7	10,7	10,7	10,7	-	-
23	1,8	1,8	1,8	1,8	1,8	1,8	8,3	8,3	8,3	8,3	-	-
24	-	-	-	-	-	-	6,3	6,3	6,3	6,3	-	-
25	-	-	-	-	-	-	13,9	13,9	11,4	-	-	-
26	-	-	-	-	-	-	31,7	31,7	20,4	8,5	-	-
27	-	-	-	-	-	-	8,5	8,5	8,5	4,4	-	-
28	-	-	-	-	-	-	26,7	26,7	15,4	5,2	-	-
29	-	-	-	-	-	-	17,1	17,1	6,5	-	-	-
30	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	16,7	16,7	10,7	-2,0	-2,0	-
31	2,3	2,3	-	-	-	-	25,0	25,0	10,6	4,5	2,3	-
32	-	-	-	-	-	-	14,8	14,8	6,8	6,8	2,8	-
33	1,9	1,9	-	-1,9	-1,9	-1,9	25,4	25,4	11,7	8,6	8,6	3,6
34	2,7	2,7	-	-	-	-	28,0	28,0	11,1	8,9	10,0	8,9
35	6,8	6,8	6,8	6,8	6,8	6,8	24,4	24,4	23,6	17,1	13,9	12,8
36	5,3	5,3	5,3	5,3	5,3	5,3	26,8	26,8	17,4	13,5	8,2	7,2
37	4,1	4,1	2,1	2,1	2,1	2,1	38,2	38,2	14,5	9,6	11,3	9,6
38	10,5	10,5	7,2	7,2	7,2	7,2	30,6	30,6	19,8	11,5	12,5	10,5
39	6,9	6,9	2,9	2,9	2,9	2,9	26,4	26,4	21,2	13,0	10,7	8,2
40	5,5	5,5	3,7	3,7	3,7	3,7	22,4	22,4	11,9	11,1	7,1	6,3
41	-	-	-	-	-	-	27,8	27,8	13,3	12,4	10,3	11,4
42	-	-	-2,0	-2,0	-2,0	-2,0	28,8	28,8	11,9	9,6	8,0	8,8
43	-	-	-	-	-	-	19,1	19,1	18,5	9,1	3,5	3,5
44	2,1	2,1	2,1	2,1	2,1	2,1	27,9	27,9	17,0	14,4	6,8	6,2
45	1,5	1,5	-	-	-	-	40,0	40,0	19,5	16,5	12,0	12,0
46	6,7	6,7	4,3	4,3	4,3	4,3	24,5	24,5	14,6	9,0	9,0	10,5
47	5,9	5,9	5,1	3,5	3,5	3,5	28,8	28,8	17,8	10,5	11,9	10,5
48	1,7	1,7	1,2	1,2	1,2	1,2	22,1	22,1	15,9	9,6	6,6	5,6

Tabela C.14: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G7 e G8 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações

Inst.	G7						G8					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	5,9	5,9	5,9	5,9	-	-	-	-	-	-	-	-
2	10,7	10,7	10,7	10,7	3,8	3,8	-	-	-	-	-	-
3	5,3	5,3	5,3	5,3	-	-	-	-	-	-	-	-
4	7,4	7,4	7,4	7,4	-	-	-	-	-	-	-	-
5	5,9	5,9	5,9	5,9	-	-	-	-	-	-	-	-
6	18,2	18,2	18,2	12,9	-	-	6,9	6,9	6,9	-	-	-
7	5,9	5,9	5,9	5,9	-	-	-	-	-	-	-	-
8	6,1	6,1	6,1	-	-	-	-	-	-	-	-	-
9	32,4	32,4	32,4	10,7	-	-	-	-	-	-	-	-
10	11,5	11,5	11,5	9,8	-	-	6,1	6,1	6,1	-	-	-
11	15,8	15,8	15,8	15,8	-	-	-	-	-	-	-	-
12	12,8	12,8	12,8	12,8	4,7	-	-	-	-	-	-	-
13	12,5	12,5	12,5	12,5	-	-	-	-	-	-	-	-
14	10,0	10,0	10,0	10,0	-	-	6,3	6,3	6,3	-	-	-
15	10,3	10,3	10,3	10,3	-	-	-	-	-	-	-	-
16	12,5	12,5	12,5	9,3	-	-	-	-	-	-	-	-
17	28,6	28,6	28,6	14,6	5,4	-	14,6	14,6	14,6	-	-	-
18	18,9	18,9	18,9	11,8	4,8	3,2	10,4	10,4	10,4	1,6	1,6	-
19	17,9	17,9	17,9	17,9	-	-	4,2	4,2	4,2	2,1	-	-
20	14,3	14,3	14,3	14,3	1,6	1,6	-	-	-	-	-	-
21	16,4	16,4	16,4	16,4	6,1	6,1	6,1	6,1	6,1	2,1	-	-
22	9,5	9,5	9,5	9,5	-	-	6,9	6,9	6,9	-	-	-
23	11,3	11,3	11,3	11,3	3,5	3,5	-	-	-	-	-	-
24	8,6	8,6	8,6	8,6	-	-	-	-	-	-	-	-
25	27,9	27,9	22,5	11,4	3,1	-	11,4	11,4	6,1	3,1	-	-
26	32,8	32,8	23,2	15,7	6,5	4,4	14,0	14,0	4,4	4,4	2,3	4,4
27	8,5	8,5	8,5	4,4	-	-	2,3	2,3	-	-	-	-
28	26,7	26,7	15,4	5,2	-	-	8,3	8,3	-	-	-	-
29	14,7	14,7	9,4	3,3	-	-	3,3	3,3	-	-	-	-
30	18,0	18,0	12,3	3,8	-	-	3,8	3,8	-	-2,0	-	-
31	25,0	25,0	10,6	8,7	4,5	-	12,5	12,5	6,7	2,3	2,3	-
32	14,8	14,8	6,8	6,8	2,8	-	6,8	6,8	2,8	-	4,2	-
33	31,2	31,2	24,3	14,5	10,2	10,2	5,4	5,4	3,6	7,0	5,4	5,4
34	33,3	33,3	20,0	16,3	8,9	11,1	14,3	14,3	4,0	4,0	5,3	4,0
35	24,4	24,4	23,6	17,1	13,9	12,8	21,8	21,8	17,1	15,0	11,7	12,8
36	26,8	26,8	17,4	13,5	8,2	7,2	15,1	15,1	10,0	13,5	7,2	8,2
37	42,7	42,7	26,6	20,3	16,1	14,5	24,2	24,2	11,3	6,0	6,0	6,0
38	33,0	33,0	23,8	16,3	13,5	10,5	15,4	15,4	11,5	11,5	9,4	9,4
39	25,6	25,6	20,2	14,1	13,0	9,5	11,8	11,8	10,7	11,8	8,2	8,2
40	23,0	23,0	11,9	11,9	9,6	6,3	9,6	9,6	9,6	6,3	6,3	5,5
41	33,3	33,3	25,7	18,8	14,3	12,4	14,3	14,3	11,4	10,3	10,3	11,4
42	31,1	31,1	18,1	16,1	16,1	11,9	13,3	13,3	7,1	6,3	8,0	6,3
43	19,1	19,1	18,5	9,1	3,5	7,6	17,3	17,3	12,7	9,1	6,8	7,6
44	27,9	27,9	17,0	14,4	6,8	6,2	14,4	14,4	7,4	13,3	8,1	6,2
45	46,3	46,3	31,3	22,4	19,5	17,5	25,0	25,0	17,5	16,5	10,8	15,4
46	33,5	33,5	22,4	13,3	13,3	11,9	9,0	9,0	7,5	9,8	8,3	11,2
47	29,3	29,3	19,0	11,2	11,9	9,0	20,7	20,7	17,2	10,5	10,5	5,1
48	22,1	22,1	15,9	9,6	6,6	6,1	9,1	9,1	10,6	6,6	7,1	4,5

Tabela C.15: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G9 e G10 (em 10 execuções) com variação de alfa do GRASP para instâncias de 6 e 12 locações

Inst.	G9						G10					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	-	-	-	-	-	-	15,8	5,9	5,9	5,9	5,9	5,9
2	7,4	7,4	7,4	7,4	3,8	3,8	7,4	-	-	-	-	-
3	-	-	-	-	-	-	10,0	5,3	-	-	-	-
4	-	-	-	-	-	-	3,8	-	-	-	-	-
5	-	-	-	-	-	-	27,3	5,9	5,9	5,9	-	5,9
6	6,9	6,9	6,9	-	-	-	12,9	6,9	6,9	-	3,6	3,6
7	-	-	-	-	-	-	15,8	-	-	-	-	-
8	-	-	-	-	-	-	3,1	-	-	-	-	-
9	10,7	10,7	10,7	-	-	-	16,7	13,8	13,8	19,4	21,9	19,4
10	6,1	6,1	6,1	-	-	-	16,4	8,0	6,1	4,2	-	2,1
11	-	-	-	-	-	-	5,9	5,9	3,0	3,0	-	3,0
12	2,4	2,4	2,4	-	-	-	4,7	-	-	-	-	-
13	-	-	-	-	-	-	17,6	6,7	12,5	15,2	15,2	12,5
14	6,3	6,3	6,3	-	-	-	11,8	8,2	8,2	4,3	4,3	2,2
15	-	-	-	-	-	-	7,9	2,8	-	-	5,4	2,8
16	-	-	-	-	-	-	2,0	-	-	-	-	-
17	16,7	16,7	16,7	5,4	5,4	-	23,9	18,6	18,6	23,9	20,5	23,9
18	13,0	13,0	13,0	6,3	6,3	3,2	15,5	10,4	10,4	7,7	9,1	9,1
19	4,2	4,2	4,2	2,1	-	-	9,8	6,1	6,1	9,8	4,2	2,1
20	4,8	4,8	4,8	1,6	1,6	1,6	3,2	-	-	-	-	-
21	6,1	6,1	6,1	6,1	6,1	6,1	13,2	6,1	6,1	13,2	13,2	8,0
22	5,6	5,6	5,6	-	-	-	8,2	5,6	8,2	4,3	1,5	2,9
23	3,5	3,5	3,5	3,5	3,5	3,5	5,2	1,8	1,8	8,3	3,5	-
24	-	-	-	-	-	-	-	-	-	-	-	-
25	20,5	20,5	8,8	3,1	-	-	24,4	31,1	29,5	24,4	32,6	31,1
26	15,7	15,7	8,5	2,3	4,4	6,5	20,4	23,2	14,0	14,0	17,3	18,9
27	2,3	2,3	-	-	-	-	17,3	2,3	10,4	6,5	4,4	4,4
28	8,3	8,3	-	-	-	-	15,4	8,3	-	-	-	-
29	3,3	3,3	-	-	-	-	27,5	19,4	17,1	25,6	17,1	31,0
30	7,4	7,4	2,0	-	-	2,0	7,4	12,3	10,7	9,1	10,7	12,3
31	12,5	12,5	8,7	6,7	2,3	2,3	16,0	19,2	12,5	4,5	4,5	6,7
32	6,8	6,8	2,8	-	4,2	-	8,0	5,5	4,2	-	-	-
33	14,5	14,5	8,6	7,0	7,0	11,7	26,4	26,4	31,2	32,9	33,8	34,6
34	15,3	15,3	6,5	7,7	8,9	6,5	19,1	19,1	17,2	20,9	20,0	20,0
35	21,8	21,8	17,1	15,0	11,7	12,8	22,7	22,7	20,0	19,0	20,9	18,1
36	15,1	15,1	10,0	13,5	7,2	8,2	18,9	15,1	8,2	7,2	5,3	7,2
37	28,8	28,8	17,5	9,6	11,3	9,6	35,6	35,6	33,8	38,2	38,2	39,0
38	17,2	17,2	13,5	13,5	12,5	10,5	18,1	18,9	23,0	21,4	22,2	23,8
39	14,1	14,1	15,2	14,1	9,5	10,7	20,2	24,7	23,9	23,0	15,2	14,1
40	9,6	9,6	9,6	6,3	8,8	5,5	11,1	11,1	9,6	6,3	5,5	6,3
41	19,6	19,6	13,3	10,3	14,3	15,2	23,5	25,7	29,1	32,2	33,9	34,5
42	15,4	15,4	9,6	8,8	11,1	8,8	18,8	18,1	18,8	16,8	23,5	21,8
43	17,9	17,9	12,7	9,1	6,8	7,6	19,1	17,3	15,4	12,0	14,7	14,7
44	14,4	14,4	7,4	13,3	8,1	6,2	17,0	14,4	9,3	4,9	4,2	3,5
45	31,3	31,3	21,4	21,4	19,5	16,5	36,5	37,1	38,3	42,1	44,5	43,6
46	17,2	17,2	9,8	12,6	12,6	13,3	20,1	20,1	20,7	22,9	21,3	19,0
47	20,7	20,7	19,0	11,2	11,9	6,7	25,5	24,5	23,4	17,2	15,3	14,6
48	10,1	10,1	11,5	7,1	8,2	6,1	10,1	10,1	8,2	5,1	4,5	2,9

Tabela C.16: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G1 e G2 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações

Inst.	G1						G2					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	2,2	-2,3	-2,3	-2,3	-2,3	-2,3	2,2	-2,3	-2,3	-2,3	-2,3	-2,3
50	4,5	-	-5,0	-5,0	-5,0	-5,0	4,5	-	-5,0	-5,0	-5,0	-5,0
51	5,2	3,5	-3,8	-3,8	-1,9	-3,8	5,2	3,5	-3,8	-3,8	-1,9	-3,8
52	-5,4	-10,1	-10,1	-10,1	-10,1	-10,1	-5,4	-10,1	-10,1	-10,1	-10,1	-10,1
53	2,0	-4,3	-4,3	-4,3	-4,3	-4,3	2,0	-4,3	-4,3	-4,3	-4,3	-4,3
54	-1,5	-1,5	-6,3	-6,3	-8,1	-8,1	-1,5	-1,5	-6,3	-6,3	-8,1	-8,1
55	1,6	-	-3,3	-3,3	-5,0	-5,0	1,6	-	-3,3	-3,3	-5,0	-5,0
56	-2,1	-5,4	-9,0	-9,0	-9,0	-9,0	-2,1	-5,4	-9,0	-9,0	-9,0	-9,0
57	5,6	1,5	-1,5	-1,5	-1,5	-1,5	5,6	1,5	-1,5	-1,5	-1,5	-1,5
58	3,6	-	-5,0	-5,0	-5,0	-6,0	3,6	-	-5,0	-5,0	-5,0	-6,0
59	9,8	4,7	-6,3	-8,6	-6,3	-8,6	9,8	4,7	-6,3	-8,6	-6,3	-8,6
60	8,6	1,2	-3,9	-3,9	-3,9	-3,9	8,6	1,2	-3,9	-3,9	-3,9	-3,9
61	4,7	-5,1	-10,8	-10,8	-10,8	-10,8	4,7	-5,1	-10,8	-10,8	-10,8	-10,8
62	-6,6	-13,2	-14,2	-14,2	-14,2	-14,2	-6,6	-13,2	-14,2	-14,2	-14,2	-14,2
63	4,0	2,4	-2,5	-2,5	-2,5	-3,4	4,0	2,4	-2,5	-2,5	-2,5	-3,4
64	1,0	-5,6	-6,7	-8,6	-7,3	-8,0	1,0	-5,6	-6,7	-8,6	-7,3	-8,0
65	7,9	1,9	-4,0	-1,9	-1,9	-2,9	7,9	1,9	-4,0	-1,9	-1,9	-2,9
66	4,9	2,5	-2,0	-2,6	-2,0	-2,6	4,9	2,5	-2,0	-2,6	-2,0	-2,6
67	3,7	-0,6	-6,1	-4,7	-6,8	-6,1	3,7	-0,6	-6,1	-4,7	-6,8	-6,1
68	6,0	-2,2	-6,4	-7,3	-7,3	-7,3	6,0	-2,2	-6,4	-7,3	-7,3	-7,3
69	12,5	5,9	2,6	3,4	4,3	4,3	12,5	5,9	2,6	3,4	4,3	4,3
70	0,6	-1,7	-4,7	-4,1	-5,3	-6,6	0,6	-1,7	-4,7	-4,1	-5,3	-6,6
71	5,0	4,5	-1,8	-2,4	-3,0	-0,6	5,0	4,5	-1,8	-2,4	-3,0	-0,6
72	4,3	-2,3	-4,7	-6,9	-5,2	-3,9	4,3	-2,3	-4,7	-6,9	-5,2	-3,9
73	-2,8	-4,2	-4,2	-4,2	-4,2	-4,2	-2,8	-4,2	-4,2	-4,2	-4,2	-4,2
74	-1,0	-6,6	-5,4	-6,6	-6,6	-6,6	-1,0	-6,6	-5,4	-6,6	-6,6	-6,6
75	12,7	-5,8	-5,8	-5,8	-6,8	-5,8	12,7	-5,8	-5,8	-5,8	-6,8	-5,8
76	-0,6	-6,9	-6,9	-6,2	-6,2	-7,6	-0,6	-6,9	-6,9	-6,2	-6,2	-7,6
77	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
78	-2,0	-7,4	-8,6	-8,6	-8,6	-8,6	-2,0	-7,4	-8,6	-8,6	-8,6	-8,6
79	4,3	-7,8	-8,9	-8,9	-8,9	-10,0	4,3	-7,8	-8,9	-8,9	-8,9	-10,0
80	5,4	-6,1	-7,4	-6,1	-6,7	-6,1	5,4	-6,1	-7,4	-6,1	-6,7	-6,1
81	2,5	-5,3	-7,2	-6,3	-5,3	-4,4	2,5	-5,3	-7,2	-6,3	-5,3	-4,4
82	6,9	-4,8	-4,1	-6,0	-3,5	-4,8	6,9	-4,8	-4,1	-6,0	-3,5	-4,8
83	12,3	1,5	-1,6	-0,5	-	1,0	12,3	1,5	-1,6	-0,5	-	1,0
84	5,1	-3,7	-3,3	-4,1	-1,8	-3,3	5,1	-3,7	-3,3	-4,1	-1,8	-3,3
85	6,7	-3,3	-2,5	-1,6	-2,5	-0,8	6,7	-3,3	-2,5	-1,6	-2,5	-0,8
86	10,3	-3,2	-2,7	-3,8	-1,1	-3,8	10,3	-3,2	-2,7	-3,8	-1,1	-3,8
87	8,5	-1,0	-1,6	-2,7	-3,8	-1,0	8,5	-1,0	-1,6	-2,7	-3,8	-1,0
88	7,9	-7,9	-6,0	-5,6	-6,7	-4,9	7,9	-7,9	-6,0	-5,6	-6,7	-4,9
89	16,6	-1,8	-4,3	1,7	-1,2	-1,8	16,6	-1,8	-4,3	1,7	-1,2	-1,8
90	6,8	-3,1	-0,4	-1,9	-2,3	-1,9	6,8	-3,1	-0,4	-1,9	-2,3	-1,9
91	9,6	-0,7	-0,4	0,7	-1,8	0,4	9,6	-0,7	-0,4	0,7	-1,8	0,4
92	7,1	-0,8	-1,0	-0,3	0,3	-1,0	7,1	-0,8	-1,0	-0,3	0,3	-1,0
93	11,7	-1,6	-1,1	-	-0,5	-1,6	11,7	-1,6	-1,1	-	-0,5	-1,6
94	8,5	0,4	-	0,4	0,7	1,1	8,5	0,4	-	0,4	0,7	1,1
95	12,2	0,6	0,9	0,9	0,3	-	12,2	0,6	0,9	0,9	0,3	-
96	7,4	-1,3	-3,1	0,2	-1,1	-0,9	7,4	-1,3	-3,1	0,2	-1,1	-0,9

Tabela C.17: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G3 e G4 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações

Inst.	G3						G4					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3
50	-3,3	-5,0	-5,0	-5,0	-5,0	-5,0	-3,3	-5,0	-5,0	-5,0	-5,0	-5,0
51	-	-	-3,8	-3,8	-3,8	-3,8	-	-	-3,8	-3,8	-3,8	-3,8
52	-5,4	-10,1	-10,1	-10,1	-10,1	-10,1	-5,4	-10,1	-10,1	-10,1	-10,1	-10,1
53	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
54	-6,3	-6,3	-6,3	-8,1	-6,3	-8,1	-6,3	-6,3	-6,3	-8,1	-6,3	-8,1
55	-3,3	-3,3	-5,0	-3,3	-5,0	-5,0	-3,3	-3,3	-5,0	-3,3	-5,0	-5,0
56	-6,6	-6,6	-9,0	-9,0	-9,0	-9,0	-6,6	-6,6	-9,0	-9,0	-9,0	-9,0
57	-	-1,5	-1,5	-1,5	-1,5	-1,5	-	-1,5	-1,5	-1,5	-1,5	-1,5
58	-5,0	-5,0	-6,0	-5,0	-6,0	-5,0	-5,0	-5,0	-6,0	-5,0	-6,0	-5,0
59	-2,0	3,8	-5,2	-6,3	-7,4	-6,3	-2,0	3,8	-5,2	-6,3	-7,4	-6,3
60	8,6	-0,6	-3,9	-3,9	-3,9	-3,9	8,6	-0,6	-3,9	-3,9	-3,9	-3,9
61	-9,3	-9,3	-10,8	-10,8	-10,8	-10,8	-9,3	-9,3	-10,8	-10,8	-10,8	-10,8
62	-11,2	-14,2	-14,2	-14,2	-14,2	-14,2	-11,2	-14,2	-14,2	-14,2	-14,2	-14,2
63	4,0	-1,7	-2,5	-3,4	-1,7	-3,4	4,0	-1,7	-2,5	-3,4	-1,7	-3,4
64	-3,8	-5,6	-8,6	-8,0	-8,0	-7,3	-3,8	-5,6	-8,6	-8,0	-8,0	-7,3
65	-4,0	-4,0	-2,9	-4,0	-4,0	-1,9	-4,0	-4,0	-2,9	-4,0	-4,0	-1,9
66	-	-2,0	-0,6	-2,0	-1,3	-2,6	-	-2,0	-0,6	-2,0	-1,3	-2,6
67	3,7	-0,6	-6,8	-6,1	-6,8	-5,4	3,7	-0,6	-6,8	-6,1	-6,8	-5,4
68	6,0	-2,2	-5,4	-6,4	-6,4	-7,3	6,0	-2,2	-5,4	-6,4	-6,4	-7,3
69	3,4	3,4	4,3	3,4	5,1	4,3	3,4	3,4	4,3	3,4	5,1	4,3
70	-3,5	-4,7	-4,1	-4,1	-5,3	-4,7	-3,5	-4,7	-4,1	-4,1	-5,3	-4,7
71	2,3	1,7	-1,2	-0,6	-0,6	-2,4	2,3	1,7	-1,2	-0,6	-0,6	-2,4
72	-0,4	-4,3	-4,7	-6,4	-6,9	-6,9	-0,4	-4,3	-4,7	-6,4	-6,9	-6,9
73	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2
74	-5,4	-6,6	-6,6	-5,4	-6,6	-7,8	-5,4	-6,6	-6,6	-5,4	-6,6	-7,8
75	-4,8	-6,8	-5,8	-6,8	-5,8	-6,8	-4,8	-6,8	-5,8	-6,8	-5,8	-6,8
76	-5,4	-5,4	-6,2	-6,9	-5,4	-6,9	-5,4	-5,4	-6,2	-6,9	-5,4	-6,9
77	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
78	-4,1	-7,4	-6,3	-7,4	-8,6	-7,4	-4,1	-7,4	-6,3	-7,4	-8,6	-7,4
79	4,3	-8,9	-10,0	-10,0	-8,9	-8,9	4,3	-8,9	-10,0	-10,0	-8,9	-8,9
80	5,4	-7,4	-5,4	-6,7	-6,1	-6,1	5,4	-7,4	-5,4	-6,7	-6,1	-6,1
81	-2,6	-6,3	-8,2	-4,4	-5,3	-4,4	-2,6	-6,3	-8,2	-4,4	-5,3	-4,4
82	-2,9	-4,1	-6,7	-6,0	-3,5	-4,1	-2,9	-4,1	-6,7	-6,0	-3,5	-4,1
83	-	1,5	1,0	1,0	-2,7	0,5	-	1,5	1,0	1,0	-2,7	0,5
84	-0,4	-3,3	-3,7	-2,9	-4,4	-3,3	-0,4	-3,3	-3,7	-2,9	-4,4	-3,3
85	-1,6	-1,6	-2,5	-0,8	-2,5	-0,8	-1,6	-1,6	-2,5	-0,8	-2,5	-0,8
86	-2,1	-3,8	-3,2	-3,8	-1,6	-3,8	-2,1	-3,8	-3,2	-3,8	-1,6	-3,8
87	8,5	-0,5	-3,8	-	0,5	-	8,5	-0,5	-3,8	-	0,5	-
88	7,9	-7,9	-5,6	-5,2	-6,3	-8,2	7,9	-7,9	-5,6	-5,2	-6,3	-8,2
89	3,4	-	0,6	0,6	-	-0,6	3,4	-	0,6	0,6	-	-0,6
90	-3,6	-4,0	-3,1	-4,4	-3,6	-1,9	-3,6	-4,0	-3,1	-4,4	-3,6	-1,9
91	2,4	1,0	-1,4	-	-1,1	-1,8	2,4	1,0	-1,4	-	-1,1	-1,8
92	1,0	-1,3	-1,5	-2,6	-2,6	-0,3	1,0	-1,3	-1,5	-2,6	-2,6	-0,3
93	1,6	-	-2,2	-0,5	-3,3	-1,6	1,6	-	-2,2	-0,5	-3,3	-1,6
94	-0,4	1,7	-	1,1	0,7	-3,3	-0,4	1,7	-	1,1	0,7	-3,3
95	12,2	-0,3	-1,6	-	-1,6	1,2	12,2	-0,3	-1,6	-	-1,6	1,2
96	7,4	-	-1,3	-0,7	-2,2	-0,2	7,4	-	-1,3	-0,7	-2,2	-0,2

Tabela C.18: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G5 e G6 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações

Inst.	G5						G6					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	6,3	2,2	-2,3	-2,3	-2,3	-2,3
50	-3,3	-5,0	-5,0	-5,0	-5,0	-5,0	13,7	12,5	1,6	-	-	-1,6
51	-	-	-3,8	-3,8	-3,8	-3,8	27,6	16,7	5,2	5,2	8,3	1,8
52	-5,4	-10,1	-10,1	-10,1	-10,1	-10,1	16,2	14,0	2,0	-2,1	-3,2	-6,5
53	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	12,5	3,9	-2,1	-2,1	-4,3	-4,3
54	-6,3	-6,3	-6,3	-8,1	-8,1	-6,3	23,0	14,1	-	-	1,5	-1,5
55	-3,3	-3,3	-3,3	-5,0	-5,0	-5,0	25,9	13,7	1,6	6,0	1,6	3,1
56	-6,6	-6,6	-9,0	-9,0	-9,0	-9,0	15,7	8,5	6,7	4,0	-	-1,0
57	-	-1,5	-1,5	-1,5	-1,5	-1,5	17,3	13,0	5,6	6,9	9,5	4,3
58	-5,0	-5,0	-5,0	-6,0	-5,0	-5,0	10,9	8,6	2,8	2,8	1,9	0,9
59	-2,0	4,7	-7,4	-7,4	-7,4	-8,6	26,8	15,8	3,8	-2,0	5,6	1,9
60	8,6	-2,6	-3,9	-3,9	-3,9	-3,9	12,2	7,6	4,2	1,2	4,2	2,5
61	-9,3	-10,8	-10,8	-10,8	-9,3	-10,8	24,1	11,8	5,7	3,5	4,7	2,4
62	-11,2	-14,2	-14,2	-14,2	-15,2	-14,2	20,9	7,9	-4,0	1,5	-	-0,8
63	4,0	0,8	-2,5	-1,7	-2,5	-3,4	20,9	20,9	10,4	8,3	9,0	9,0
64	-3,8	-7,3	-8,6	-8,0	-8,6	-8,0	20,5	10,8	6,9	1,0	4,0	1,6
65	-4,0	-2,9	-1,9	-4,0	-2,9	-2,9	23,4	11,8	8,7	7,1	8,7	2,8
66	-	-2,0	-2,0	-2,0	-2,0	-1,3	11,9	11,9	8,8	6,6	4,9	7,7
67	3,7	-0,6	-6,1	-6,8	-5,4	-5,4	19,1	19,1	5,4	3,7	5,4	3,7
68	6,0	-2,2	-6,4	-7,3	-6,8	-6,8	16,1	11,7	5,3	3,7	1,3	0,8
69	3,4	4,3	3,4	3,4	3,4	4,3	31,3	20,6	17,0	13,2	15,2	15,2
70	-3,5	-3,5	-6,0	-5,3	-6,6	-4,7	22,6	15,2	10,6	7,3	7,3	4,8
71	2,3	4,5	-1,2	-1,8	-0,6	-0,6	25,8	16,7	12,4	11,0	13,3	13,3
72	-0,4	-4,7	-3,5	-6,9	-5,2	-5,2	18,7	11,1	7,0	8,6	7,7	5,0
73	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	20,4	3,9	-	1,3	1,3	-1,4
74	-5,4	-6,6	-6,6	-6,6	-5,4	-6,6	27,1	11,0	11,8	10,2	10,2	9,3
75	-5,8	-6,8	-6,8	-5,8	-5,8	-6,8	28,6	14,1	14,1	9,1	9,8	11,3
76	-5,4	-5,4	-6,2	-6,9	-5,4	-7,6	27,2	18,4	12,4	6,6	9,9	9,4
77	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	24,7	3,9	1,4	-	-1,4	-
78	-4,1	-7,4	-8,6	-8,6	-7,4	-8,6	25,7	13,7	11,4	10,6	11,4	9,8
79	4,3	-7,8	-10,0	-10,0	-8,9	-8,9	34,9	19,1	12,0	14,7	16,7	12,7
80	5,4	-6,1	-7,4	-6,7	-6,7	-7,4	34,2	14,2	14,6	15,0	11,2	10,7
81	-3,5	-5,3	-4,4	-4,4	-5,3	-5,3	33,5	15,6	17,9	13,8	16,8	15,6
82	-3,5	-3,5	-6,7	-5,4	-5,4	-4,8	28,5	16,2	10,2	16,2	16,2	13,7
83	-	1,5	-	-	0,5	2,5	34,9	22,3	12,7	18,6	21,6	20,7
84	-0,4	-4,1	-2,5	-4,1	-5,6	-4,1	32,2	14,5	15,3	15,8	13,5	14,3
85	-1,6	-0,8	-0,8	-2,5	-	-	28,2	19,9	18,3	18,3	16,1	16,7
86	-2,7	-3,8	-4,3	-2,1	-4,3	-1,1	25,0	15,0	16,5	14,7	17,9	13,5
87	8,5	-0,5	-0,5	-4,3	-2,1	-2,1	35,0	22,2	19,9	20,6	20,2	17,9
88	7,9	-6,7	-6,7	-4,1	-7,9	-6,7	23,9	14,0	17,0	13,2	14,0	11,2
89	3,4	1,2	1,2	-	-1,2	-	38,9	22,3	17,8	22,3	19,0	20,1
90	-1,9	-7,4	-3,6	-2,3	-4,4	-2,3	35,8	17,9	17,9	16,8	14,1	16,6
91	2,4	-1,1	-	-1,1	-0,4	-1,4	26,8	16,7	20,2	18,9	19,8	17,0
92	1,0	-1,0	-1,0	-2,1	-1,5	-0,8	27,7	15,8	13,9	15,8	14,9	13,0
93	1,0	0,5	-	-1,6	-1,1	-1,6	32,0	22,9	21,3	18,9	19,2	18,2
94	1,4	-0,4	0,7	1,1	1,1	0,7	24,5	16,4	14,8	16,4	16,6	13,5
95	12,2	-0,6	1,5	-	0,6	0,9	32,8	22,8	22,2	22,4	18,9	20,5
96	7,4	-0,4	-2,0	-2,2	-0,9	-1,5	28,3	19,9	18,9	17,9	17,7	15,2

Tabela C.19: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G7 e G8 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações

Inst.	G7						G8					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	30,8	26,2	13,5	13,5	10,0	4,3	-	-2,3	-2,3	-2,3	-2,3	-2,3
50	27,6	19,2	14,9	11,3	11,3	13,7	-	3,1	-	-	1,6	1,6
51	29,5	28,6	15,4	12,7	17,9	15,4	12,7	9,8	6,8	8,3	8,3	6,8
52	17,6	13,3	3,0	2,0	-2,1	-3,2	16,2	-1,0	2,0	-2,1	-3,2	-5,4
53	36,4	29,0	16,9	18,3	9,3	5,8	-	-2,1	-4,3	-2,1	-4,3	-4,3
54	38,5	25,6	17,3	14,1	8,2	11,8	-	-1,5	-	-1,5	-	-1,5
55	33,7	18,2	8,7	10,0	11,3	8,7	8,7	4,5	6,0	4,5	4,5	3,1
56	15,7	9,3	7,6	3,0	6,7	4,0	3,0	1,0	-1,0	4,0	4,9	-2,1
57	44,2	44,2	24,7	28,7	24,7	25,6	9,5	4,3	5,6	5,6	5,6	2,9
58	22,6	22,6	16,5	13,8	12,4	14,5	-1,0	-1,0	0,9	-1,0	0,9	-3,9
59	29,4	22,9	17,2	12,2	9,0	9,8	5,6	6,5	3,8	1,9	-	2,9
60	18,0	13,6	7,0	5,4	5,4	4,8	12,2	7,6	3,0	3,0	5,4	2,5
61	44,2	36,4	24,1	23,4	21,2	23,4	6,8	1,2	2,4	1,2	1,2	3,5
62	35,8	26,3	12,2	15,7	14,0	12,8	2,3	-3,2	-	-1,6	-3,2	1,5
63	31,3	29,7	23,9	19,3	21,4	16,6	15,4	11,0	10,4	9,7	6,9	9,0
64	18,1	10,8	10,4	7,3	2,6	2,6	-1,6	4,0	6,9	3,1	3,6	-
65	38,6	38,2	26,6	26,6	26,1	26,1	7,9	2,8	4,5	4,5	8,7	6,3
66	29,7	28,8	21,2	19,6	15,2	16,6	9,3	4,3	5,5	3,7	3,7	6,6
67	24,5	19,1	10,3	12,3	8,7	10,3	19,1	19,1	3,1	3,7	1,9	-
68	20,4	15,5	3,3	4,1	3,7	1,7	16,1	3,7	4,1	0,8	3,3	-
69	45,6	44,3	32,5	32,5	31,3	29,6	18,2	15,2	14,5	11,8	15,2	14,5
70	41,1	31,0	20,9	21,9	20,2	18,3	4,8	3,8	6,8	4,8	6,3	6,3
71	36,8	30,0	24,8	23,1	22,0	16,3	17,9	15,8	14,6	14,1	12,8	11,5
72	20,4	11,1	12,5	7,3	6,7	7,7	5,0	6,4	9,9	6,4	5,4	6,0
73	31,5	19,6	15,9	15,9	14,0	14,9	5,1	-	-1,4	-	-1,4	1,3
74	37,4	27,1	22,4	23,6	23,0	21,8	11,0	1,0	2,0	4,0	5,8	6,7
75	35,3	27,6	23,6	17,9	17,9	19,7	8,3	7,6	12,0	13,4	13,4	6,8
76	25,8	17,6	11,4	11,9	13,9	3,1	8,8	11,4	13,9	10,4	11,4	11,9
77	47,5	29,1	23,2	12,0	15,1	16,1	6,4	1,4	-	1,4	-1,4	-
78	43,3	27,9	27,9	25,7	24,1	24,6	10,6	8,2	4,7	8,2	8,2	9,8
79	41,8	29,0	24,1	26,2	22,5	19,7	34,9	17,9	16,7	12,7	14,1	6,8
80	34,9	14,6	18,2	15,9	13,8	12,1	34,2	14,2	12,5	12,9	14,2	10,7
81	49,1	38,7	36,4	32,4	27,4	32,8	21,2	16,2	12,5	14,4	11,9	11,2
82	43,2	31,0	28,7	27,6	27,6	26,7	17,4	12,0	11,1	10,2	10,7	12,0
83	39,6	29,2	29,2	26,4	30,4	28,9	17,9	22,9	22,0	22,6	21,3	19,7
84	27,7	19,0	14,8	17,5	18,5	16,3	17,3	15,8	12,7	15,8	15,6	15,3
85	49,4	38,7	38,4	36,9	35,6	36,5	18,3	13,8	13,2	15,5	17,8	16,7
86	41,5	29,4	33,1	28,1	28,4	26,2	8,1	11,1	13,9	15,0	13,9	15,0
87	38,9	29,3	28,0	28,0	28,0	29,6	35,0	22,5	21,9	20,9	20,9	17,5
88	26,7	16,8	16,6	14,9	16,6	13,2	23,9	15,4	12,7	13,2	10,4	10,1
89	51,0	38,9	37,6	37,4	34,5	35,0	24,3	17,0	17,8	17,0	15,8	17,8
90	39,5	27,4	28,0	26,0	26,8	22,9	13,0	9,3	9,3	12,7	11,5	13,5
91	37,0	27,2	26,4	27,2	27,7	26,4	18,6	17,0	18,4	18,9	17,9	17,7
92	30,3	16,1	14,5	17,0	13,8	14,3	18,7	16,1	16,1	14,7	12,6	13,4
93	52,9	40,6	38,2	38,0	36,8	35,3	20,3	17,1	17,8	17,1	16,7	17,1
94	41,9	31,6	33,3	30,3	28,7	28,9	14,1	13,5	14,3	12,7	12,7	15,6
95	37,9	28,2	28,2	27,7	27,6	25,4	32,8	23,6	19,3	16,5	18,7	19,7
96	32,4	23,4	20,7	20,0	19,7	19,4	28,3	21,0	16,2	15,0	15,3	14,2

Tabela C.20: Desvio percentual entre a melhor solução da literatura e a melhor solução obtida pelos algoritmos GRASP G9 e G10 (em 10 execuções) com variação de alfa do GRASP para instâncias de 20 e 32 locações

Inst.	G9						G10					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	4,3	-2,3	11,8	11,8	13,5	6,3	23,7	27,4	32,8	38,4	41,6	41,6
50	11,3	10,0	7,4	7,4	12,5	13,7	25,0	19,2	21,3	19,2	20,3	19,2
51	19,1	16,7	15,4	14,1	14,1	16,7	29,5	21,4	20,3	23,6	24,7	19,1
52	17,6	-3,2	3,0	2,0	-2,1	-3,2	17,6	3,9	1,0	-4,3	-4,3	-5,4
53	15,5	-2,1	7,5	3,9	7,5	14,0	27,9	25,8	27,9	34,7	41,0	38,0
54	19,3	9,5	8,2	9,5	10,7	13,0	16,3	21,2	22,1	25,6	23,0	23,9
55	17,1	13,7	12,5	11,3	13,7	10,0	17,1	17,1	10,0	23,2	18,2	22,2
56	2,0	4,9	5,8	7,6	5,8	1,0	6,7	4,0	5,8	1,0	4,0	2,0
57	21,2	17,3	22,1	22,1	24,7	25,6	25,6	30,2	42,2	47,2	50,0	49,6
58	10,2	7,8	6,2	9,4	11,7	12,4	20,3	17,2	23,2	25,9	21,5	22,6
59	18,5	19,2	17,2	12,9	13,7	8,2	22,9	22,9	17,9	24,1	26,8	22,9
60	18,0	11,2	8,6	4,2	8,1	5,4	24,6	8,6	6,5	6,5	2,5	3,0
61	15,5	16,3	10,9	18,8	20,4	23,4	29,3	29,3	40,6	44,2	45,3	45,7
62	17,8	11,6	8,5	12,2	11,6	7,2	10,4	15,7	19,4	23,2	23,7	21,3
63	21,9	18,2	18,8	19,9	18,2	18,2	21,4	21,9	26,2	20,4	21,4	22,4
64	4,0	5,5	9,1	5,5	5,9	2,6	7,8	5,0	8,7	3,1	2,1	0,5
65	25,0	18,0	23,4	21,6	25,5	24,5	24,5	32,3	42,3	47,2	49,5	50,2
66	17,9	12,4	15,7	14,8	14,8	15,7	21,6	20,8	23,9	27,8	29,1	27,4
67	24,5	19,1	11,8	11,3	10,8	12,8	25,6	20,7	19,9	21,1	23,4	22,3
68	20,4	6,4	5,3	1,7	4,9	1,7	20,1	8,6	5,6	1,7	1,3	1,7
69	30,0	24,8	29,6	28,7	29,1	29,1	33,7	37,1	45,1	46,9	51,5	52,3
70	20,9	16,0	16,0	14,0	18,7	18,0	20,9	19,5	23,3	27,6	26,7	26,7
71	27,4	24,4	23,4	22,7	22,4	20,6	25,1	25,4	27,7	27,0	26,7	26,1
72	6,4	9,6	13,7	9,9	8,9	7,0	7,3	9,9	10,2	8,3	2,2	4,7
73	22,9	15,9	11,9	12,9	9,8	14,0	30,2	32,7	37,3	41,7	45,6	46,4
74	20,5	13,4	12,6	16,4	18,5	14,9	20,5	25,4	25,4	30,2	27,6	25,4
75	19,7	18,5	23,6	21,4	19,1	20,9	31,3	26,7	28,6	26,7	28,6	28,6
76	8,8	11,9	14,8	12,9	13,9	12,4	17,1	13,9	10,4	10,4	8,3	9,9
77	29,8	8,8	8,8	11,0	14,1	16,1	36,0	23,2	33,6	39,7	46,7	49,7
78	20,5	16,5	20,5	17,2	15,8	19,8	33,1	26,3	24,6	30,8	32,2	35,7
79	41,8	27,6	26,2	23,1	23,1	24,1	53,2	24,7	29,9	24,1	29,0	28,1
80	34,9	14,6	15,9	15,9	15,5	10,7	35,7	17,1	11,2	11,2	10,7	10,3
81	38,7	28,7	26,5	27,0	30,4	30,0	42,0	42,5	46,6	52,2	51,8	54,2
82	26,1	17,8	17,4	19,6	18,9	21,1	24,1	32,0	33,1	34,6	35,3	37,1
83	23,2	29,2	30,9	26,7	28,1	26,2	28,1	35,6	36,2	34,7	37,5	34,5
84	19,2	19,0	15,6	17,5	18,0	18,3	21,7	20,1	17,8	13,8	13,2	12,4
85	30,6	29,0	29,4	32,8	32,1	33,5	37,2	41,3	47,9	52,1	54,5	55,0
86	23,2	21,3	21,0	22,0	26,4	25,9	28,4	26,7	34,0	35,4	33,8	37,3
87	38,9	29,3	30,1	24,9	25,5	27,4	45,6	35,0	35,2	32,0	35,7	32,8
88	26,7	17,0	14,9	17,0	13,7	16,1	34,8	17,3	11,2	14,0	10,9	9,3
89	36,0	30,5	31,9	31,9	30,5	30,2	38,7	41,0	46,9	53,8	56,6	57,0
90	19,4	17,9	18,6	19,1	19,6	22,0	19,4	27,8	29,4	30,1	34,2	32,5
91	23,0	26,8	26,4	26,8	27,6	27,0	32,4	35,2	33,5	34,9	31,9	32,4
92	19,9	18,2	15,4	17,2	15,1	16,0	32,7	17,5	15,4	13,4	12,8	14,7
93	36,4	32,3	33,0	33,5	34,6	35,9	36,8	40,8	47,8	52,8	54,9	56,3
94	24,3	24,3	24,3	25,3	25,1	26,1	26,4	28,3	32,0	35,8	38,2	38,1
95	37,9	29,0	26,7	28,7	25,9	26,7	42,5	33,9	33,2	32,6	33,1	27,9
96	32,4	24,4	22,4	16,1	15,6	18,0	32,4	21,0	13,3	15,3	13,9	13,6

Tabela C.21: Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP G1 e G2 de acordo com variação de alfa para instâncias de 6 e 12 locações

Inst.	G1						G2					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	0,1	0,1	0,0	0,0	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1
2	0,1	0,1	0,0	0,0	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1
3	0,1	0,1	0,0	0,0	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1
4	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,1	0,0	0,0	0,0	0,1
5	0,2	0,1	0,0	0,0	0,0	0,1	0,0	0,2	0,0	0,0	0,0	0,1
6	0,1	0,1	0,0	0,0	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1
7	0,1	0,1	0,0	0,0	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1
8	0,1	0,1	0,0	0,0	0,0	0,1	0,0	0,1	0,0	0,0	0,0	0,1
9	0,3	0,3	0,1	0,1	0,1	0,2	0,1	0,3	0,1	0,1	0,1	0,3
10	0,3	0,2	0,1	0,1	0,1	0,3	0,1	0,1	0,1	0,1	0,1	0,3
11	0,3	0,3	0,1	0,1	0,1	0,3	0,1	0,2	0,1	0,1	0,1	0,3
12	0,2	0,2	0,1	0,1	0,1	0,2	0,1	0,1	0,1	0,1	0,1	0,2
13	0,2	0,2	0,1	0,1	0,1	0,3	0,1	0,1	0,1	0,1	0,1	0,3
14	0,3	0,2	0,1	0,1	0,1	0,3	0,1	0,1	0,1	0,1	0,1	0,3
15	0,2	0,2	0,1	0,1	0,1	0,3	0,1	0,1	0,1	0,1	0,1	0,3
16	0,4	0,3	0,1	0,1	0,1	0,3	0,1	0,4	0,1	0,1	0,1	0,3
17	0,8	0,7	0,2	0,2	0,2	0,7	0,2	0,8	0,2	0,2	0,2	0,7
18	0,5	0,3	0,1	0,2	0,2	0,7	0,1	0,5	0,1	0,2	0,2	0,3
19	0,7	0,7	0,2	0,2	0,2	0,6	0,2	0,7	0,2	0,2	0,2	0,3
20	0,4	0,4	0,1	0,1	0,1	0,4	0,1	0,4	0,1	0,1	0,1	0,2
21	0,7	0,6	0,2	0,2	0,2	0,6	0,2	0,7	0,2	0,2	0,2	0,3
22	0,8	0,7	0,2	0,2	0,2	0,5	0,2	0,8	0,2	0,2	0,2	0,6
23	0,4	0,3	0,1	0,1	0,2	0,3	0,1	0,2	0,1	0,1	0,2	0,6
24	0,5	0,4	0,1	0,1	0,1	0,3	0,1	0,2	0,1	0,1	0,1	0,5
25	3,4	3,3	0,7	0,5	0,5	1,5	0,9	1,6	0,7	0,5	0,5	1,6
26	2,4	2,4	0,7	0,7	0,6	1,9	0,7	1,1	0,7	0,7	0,6	1,4
27	1,7	1,7	0,6	0,6	0,6	1,2	0,5	0,8	0,6	0,6	0,6	1,0
28	1,8	1,5	0,5	0,5	0,5	1,0	0,5	0,8	0,5	0,5	0,5	0,9
29	1,0	1,0	0,4	0,4	0,4	1,4	0,3	1,2	0,4	0,4	0,4	0,7
30	2,0	1,9	0,6	0,6	0,6	1,9	0,6	2,2	0,6	0,6	0,6	0,9
31	2,1	2,2	0,6	0,6	0,6	1,9	0,7	2,5	0,6	0,6	0,6	1,0
32	0,8	1,5	0,5	0,5	0,4	1,0	0,5	0,8	0,5	0,5	0,4	0,7
33	7,2	6,2	2,1	2,0	1,9	5,0	2,0	3,4	2,1	2,0	1,9	4,6
34	8,6	7,4	2,3	2,2	2,1	4,4	2,4	4,0	2,3	2,2	2,1	4,7
35	5,8	4,8	1,6	1,8	2,0	5,4	1,6	2,7	1,7	1,8	2,0	3,4
36	4,8	4,0	1,4	1,4	1,4	3,2	1,3	2,2	1,4	1,4	1,4	2,3
37	9,5	8,5	2,2	2,1	2,0	3,8	2,6	4,4	2,2	2,1	2,0	3,9
38	10,2	11,9	2,2	2,0	1,9	4,3	3,3	5,6	2,3	2,0	1,9	3,1
39	4,4	4,7	1,8	2,2	2,0	4,8	1,7	2,9	1,8	2,1	2,0	3,6
40	6,5	6,3	1,6	1,6	1,5	2,8	1,9	3,2	1,6	1,6	1,4	2,5
41	17,4	11,1	5,2	4,7	4,6	12,0	5,0	8,2	5,2	4,7	4,5	7,3
42	19,4	11,0	5,0	5,0	4,6	10,7	5,8	9,5	5,0	4,9	4,4	7,6
43	10,6	7,0	3,9	4,0	4,4	11,8	3,6	6,0	3,9	3,9	4,3	7,4
44	10,7	6,5	2,9	3,0	3,1	8,4	3,3	5,4	3,0	3,0	3,1	5,0
45	15,7	9,7	5,6	5,2	5,1	10,0	4,6	7,7	5,7	5,2	5,2	13,0
46	19,6	11,5	5,4	4,6	4,4	9,8	6,0	9,9	5,4	4,6	4,5	14,6
47	6,8	6,5	4,6	4,1	4,3	12,8	3,3	5,5	4,6	4,1	4,4	7,8
48	12,7	7,4	3,0	3,3	3,2	7,7	3,9	6,5	3,0	3,2	3,2	7,2

Tabela C.22: Tempo médio (em segundos) de 10 execuções do G3 e G4 de acordo com variação de alfa para instâncias de 6 e 12 locações

Inst.	G3						G4					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
9	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
10	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
11	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
12	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
13	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
14	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
15	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
16	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
17	0,2	0,2	0,2	0,2	0,2	0,2	0,3	0,3	0,3	0,2	0,2	0,2
18	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
19	0,2	0,2	0,2	0,3	0,2	0,2	0,2	0,2	0,2	0,3	0,2	0,2
20	0,1	0,1	0,1	0,2	0,1	0,2	0,1	0,1	0,1	0,2	0,2	0,2
21	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
22	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
23	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
24	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
25	0,6	0,7	0,5	0,5	0,5	0,5	0,7	0,6	0,5	0,5	0,5	0,5
26	0,7	0,7	0,6	0,6	0,6	0,6	0,7	0,7	0,6	0,6	0,6	0,6
27	0,5	0,6	0,7	0,6	0,6	0,7	0,5	0,5	0,7	0,6	0,7	0,7
28	0,5	0,6	0,6	0,5	0,6	0,6	0,5	0,6	0,6	0,5	0,6	0,6
29	0,5	0,5	0,5	0,4	0,4	0,4	0,5	0,5	0,5	0,4	0,4	0,4
30	0,5	0,5	0,5	0,5	0,5	0,6	0,5	0,5	0,5	0,5	0,6	0,6
31	0,8	0,8	0,7	0,7	0,6	0,7	0,8	0,8	0,7	0,7	0,7	0,7
32	0,5	0,5	0,6	0,5	0,5	0,5	0,5	0,5	0,6	0,6	0,5	0,5
33	2,1	2,1	1,9	1,8	1,9	1,9	2,1	2,1	2,0	1,9	1,9	1,9
34	1,8	1,8	1,8	1,8	1,9	2,0	1,8	1,8	1,8	1,9	2,0	2,0
35	2,2	2,2	1,9	2,0	2,2	2,2	2,2	2,2	1,9	2,0	2,3	2,2
36	1,6	1,6	1,6	1,6	1,7	1,6	1,6	1,6	1,6	1,6	1,7	1,6
37	2,2	2,3	2,0	1,9	2,0	2,0	2,3	2,3	2,0	2,0	2,0	2,0
38	1,6	1,7	1,7	1,8	1,8	1,8	1,6	1,7	1,7	1,8	1,8	1,9
39	1,9	2,0	2,0	2,4	2,2	2,3	1,9	1,9	2,1	2,4	2,2	2,3
40	2,2	2,2	1,8	1,8	1,7	1,7	2,2	2,2	1,9	1,9	1,7	1,6
41	4,2	4,3	4,6	4,3	4,8	4,4	4,2	4,2	4,6	4,5	4,6	4,5
42	4,0	3,9	4,0	4,0	4,3	4,3	3,9	4,0	4,0	4,1	4,4	4,3
43	4,5	4,5	4,4	4,5	4,9	4,9	4,6	4,6	4,4	4,5	5,1	4,9
44	3,7	3,7	3,5	3,5	3,7	3,6	3,7	3,7	3,5	3,6	3,7	3,6
45	5,5	5,5	5,0	4,8	4,9	5,0	5,6	5,6	5,0	4,9	5,0	5,0
46	3,4	3,4	3,8	4,0	4,1	4,2	3,5	3,6	3,8	4,0	4,2	4,2
47	3,8	3,8	5,1	4,5	4,6	4,8	3,8	3,8	5,1	4,6	4,9	4,8
48	4,4	4,4	3,6	3,8	3,6	3,7	4,4	4,4	3,6	3,9	3,8	3,6

Tabela C.23: Tempo médio (em segundos) de 10 execuções do G5 e G6 de acordo com variação de alfa para instâncias de 6 e 12 locações

Inst.	G5						G6					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0
2	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0
3	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
5	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0
6	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,1	0,0	0,0	0,0
7	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0
8	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0
9	0,2	0,2	0,2	0,2	0,2	0,2	0,0	0,0	0,0	0,0	0,0	0,0
10	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
11	0,2	0,2	0,2	0,2	0,2	0,2	0,0	0,0	0,0	0,0	0,0	0,0
12	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,1	0,0	0,1
13	0,2	0,2	0,2	0,2	0,2	0,2	0,1	0,1	0,1	0,1	0,0	0,0
14	0,1	0,1	0,1	0,2	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
15	0,2	0,2	0,2	0,2	0,2	0,2	0,0	0,0	0,0	0,0	0,0	0,0
16	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
17	0,4	0,4	0,4	0,4	0,4	0,4	0,0	0,0	0,0	0,1	0,1	0,1
18	0,3	0,3	0,3	0,3	0,3	0,3	0,1	0,1	0,1	0,1	0,1	0,1
19	0,3	0,3	0,3	0,4	0,3	0,3	0,0	0,0	0,0	0,1	0,1	0,1
20	0,2	0,2	0,2	0,2	0,2	0,2	0,1	0,1	0,1	0,1	0,1	0,1
21	0,3	0,3	0,3	0,4	0,3	0,3	0,1	0,1	0,1	0,2	0,1	0,1
22	0,3	0,3	0,3	0,3	0,3	0,3	0,1	0,1	0,1	0,1	0,1	0,1
23	0,3	0,3	0,3	0,3	0,3	0,3	0,0	0,0	0,0	0,0	0,1	0,1
24	0,2	0,2	0,2	0,2	0,2	0,3	0,1	0,1	0,1	0,1	0,1	0,1
25	1,1	1,1	1,1	1,1	1,0	1,0	0,2	0,2	0,2	0,2	0,2	0,2
26	1,0	1,0	1,0	1,0	1,0	1,0	0,1	0,1	0,2	0,2	0,2	0,2
27	0,8	0,8	1,0	1,0	1,0	0,9	0,1	0,1	0,1	0,1	0,1	0,1
28	0,7	0,7	0,7	0,7	0,7	0,7	0,1	0,1	0,4	0,3	0,4	0,3
29	1,0	0,9	1,0	1,0	0,9	0,9	0,1	0,1	0,1	0,2	0,2	0,1
30	0,7	0,8	0,8	0,9	0,8	0,8	0,2	0,2	0,2	0,2	0,2	0,2
31	1,0	1,0	0,9	1,0	0,9	0,9	0,1	0,1	0,1	0,2	0,1	0,2
32	0,6	0,6	0,7	0,7	0,6	0,6	0,4	0,4	0,3	0,4	0,2	0,2
33	3,2	3,1	3,1	3,2	3,0	3,0	0,5	0,5	0,5	0,5	0,5	0,4
34	2,6	2,7	2,6	2,9	2,7	2,8	0,6	0,6	0,6	0,7	0,7	0,7
35	2,8	2,8	2,5	2,8	2,8	2,8	0,2	0,2	0,2	0,2	0,3	0,3
36	1,7	1,8	1,8	1,9	1,9	1,8	0,2	0,2	0,7	0,5	0,8	0,8
37	3,3	3,3	3,1	3,3	3,1	3,1	0,6	0,6	0,6	0,6	0,6	0,5
38	2,4	2,3	2,4	2,7	2,5	2,6	0,5	0,5	0,9	1,0	0,9	0,9
39	2,6	2,6	2,7	3,2	2,9	2,8	0,5	0,5	0,3	0,5	0,5	0,5
40	2,4	2,4	2,1	2,2	1,9	1,8	0,9	0,9	0,7	0,9	0,7	0,6
41	6,2	6,1	6,6	6,8	6,5	6,5	1,1	1,1	1,2	1,3	1,1	1,0
42	5,3	5,3	5,3	6,0	5,7	5,8	0,9	0,9	1,4	1,7	1,6	1,6
43	5,6	5,6	5,7	6,0	6,1	5,9	0,3	0,3	0,4	0,5	0,9	0,9
44	4,1	4,0	3,8	4,2	4,0	3,9	0,3	0,3	1,7	1,3	1,7	1,6
45	7,5	7,6	7,1	7,3	6,9	6,9	1,7	1,6	1,6	1,5	1,4	1,2
46	4,9	4,8	5,1	6,0	5,5	5,5	3,9	3,9	2,4	2,3	2,1	1,9
47	4,9	4,9	6,2	6,0	5,9	5,9	0,9	0,9	1,4	0,9	1,2	1,1
48	4,8	4,8	4,0	4,3	4,1	4,0	1,7	1,6	1,4	1,7	1,5	1,4

Tabela C.24: Tempo médio (em segundos) de 10 execuções do G7 e G8 de acordo com variação de alfa para instâncias de 6 e 12 locações

Inst.	G7						G8					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
5	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0
6	0,1	0,1	0,1	0,1	0,1	0,0	0,1	0,0	0,0	0,0	0,0	0,0
7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
9	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
10	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
11	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,1	0,1	0,1
12	0,0	0,0	0,0	0,1	0,0	0,0	0,1	0,1	0,1	0,1	0,1	0,1
13	0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,1
14	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
15	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
16	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
17	0,3	0,3	0,3	0,3	0,2	0,2	0,1	0,1	0,1	0,1	0,1	0,1
18	0,2	0,2	0,2	0,3	0,2	0,2	0,1	0,1	0,1	0,2	0,1	0,1
19	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
20	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,2
21	0,2	0,2	0,2	0,2	0,2	0,2	0,1	0,1	0,1	0,1	0,1	0,1
22	0,3	0,3	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
23	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
24	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,2	0,2
25	0,7	0,7	0,7	0,6	0,4	0,4	0,3	0,3	0,2	0,2	0,2	0,2
26	0,6	0,6	0,6	0,5	0,4	0,3	0,2	0,2	0,3	0,2	0,2	0,2
27	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,2
28	0,1	0,1	0,3	0,3	0,3	0,3	0,2	0,2	0,4	0,4	0,4	0,4
29	1,2	1,2	0,7	0,5	0,5	0,4	0,2	0,2	0,2	0,2	0,2	0,2
30	0,7	0,7	0,5	0,5	0,5	0,4	0,3	0,3	0,3	0,2	0,2	0,2
31	0,3	0,3	0,3	0,2	0,2	0,2	0,1	0,1	0,2	0,2	0,2	0,2
32	0,4	0,4	0,3	0,4	0,2	0,2	0,4	0,4	0,4	0,4	0,3	0,3
33	0,8	0,8	1,2	1,2	0,9	0,8	0,8	0,8	0,6	0,5	0,5	0,5
34	0,7	0,7	0,7	1,0	0,8	0,9	0,5	0,5	0,6	0,6	0,6	0,6
35	0,2	0,2	0,2	0,2	0,4	0,4	0,4	0,4	0,4	0,4	0,5	0,5
36	0,2	0,2	0,7	0,5	0,9	0,8	0,4	0,4	0,9	0,8	1,1	1,0
37	2,1	2,1	2,1	1,6	1,5	1,0	0,7	0,7	0,6	0,7	0,6	0,6
38	2,7	2,7	2,9	2,3	2,1	1,6	1,2	1,2	1,2	1,0	1,0	1,0
39	1,1	1,1	0,7	0,7	0,6	0,7	0,7	0,7	0,6	0,7	0,7	0,7
40	1,1	1,1	0,8	1,0	0,7	0,7	1,2	1,1	1,0	1,1	0,9	0,9
41	4,0	4,0	3,6	3,1	2,5	2,1	1,4	1,3	1,5	1,3	1,3	1,3
42	4,1	4,0	2,7	2,9	2,7	2,4	1,4	1,4	1,7	1,5	1,5	1,6
43	0,3	0,3	0,5	0,5	0,9	0,8	1,0	0,9	0,9	1,0	1,4	1,4
44	0,3	0,3	1,7	1,3	1,7	1,6	0,8	0,8	2,3	1,9	2,3	2,2
45	5,4	5,3	6,7	4,5	4,4	3,1	2,2	2,2	1,9	1,6	1,4	1,5
46	7,0	6,9	8,7	5,9	5,9	4,5	3,3	3,2	2,8	2,3	2,2	2,3
47	1,2	1,2	1,8	1,4	1,9	1,8	1,5	1,4	2,0	1,4	1,7	1,7
48	2,0	2,0	1,7	2,0	2,2	2,1	2,2	2,3	1,9	2,3	2,1	2,0

Tabela C.25: Tempo médio (em segundos) de 10 execuções do G9 e G10 de acordo com variação de alfa para instâncias de 6 e 12 locações

Inst.	G9						G10					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
1	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,2	0,4	0,6
2	0,0	0,0	0,0	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,4	0,5
3	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,2	0,4	0,5
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,2	0,2	0,3
5	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,2	0,4	0,6
6	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,3	0,5
7	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,2	0,4	0,5
8	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,1	0,2	0,2
9	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,5	1,0	1,5
10	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,3	0,5	0,9	1,3
11	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,3	0,6	1,0	1,5
12	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,4	0,5	0,7
13	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,3	0,5	1,1	1,5
14	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,3	0,6	0,9	1,3
15	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,3	0,6	1,0	1,6
16	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,1	0,2	0,4	0,6	0,8
17	0,4	0,4	0,4	0,2	0,2	0,2	0,3	0,3	0,4	0,9	2,1	3,4
18	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,5	1,1	1,8	2,9
19	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,5	1,3	2,1	3,3
20	0,1	0,1	0,1	0,2	0,2	0,2	0,1	0,2	0,4	0,9	1,0	1,5
21	0,2	0,2	0,2	0,2	0,2	0,2	0,3	0,3	0,5	1,0	2,1	3,4
22	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,5	1,1	1,8	2,8
23	0,2	0,2	0,2	0,1	0,2	0,2	0,2	0,2	0,5	1,2	2,2	3,4
24	0,2	0,2	0,2	0,2	0,2	0,2	0,3	0,3	0,4	0,9	1,1	1,7
25	1,1	1,0	0,5	0,4	0,4	0,4	1,2	1,4	3,0	5,8	8,7	11,8
26	0,7	0,6	0,5	0,4	0,4	0,4	1,1	1,4	3,3	5,7	7,9	10,7
27	0,2	0,2	0,2	0,2	0,2	0,2	0,9	1,5	3,9	6,3	8,2	11,1
28	0,2	0,2	0,4	0,3	0,4	0,4	0,8	1,5	2,6	3,5	4,4	5,0
29	1,0	1,0	0,7	0,5	0,5	0,5	0,7	1,1	2,8	5,5	8,1	11,7
30	0,6	0,5	0,5	0,4	0,4	0,4	0,9	1,3	3,1	5,3	7,2	9,4
31	0,3	0,3	0,3	0,3	0,3	0,3	1,1	1,2	3,5	5,8	7,4	10,2
32	0,4	0,4	0,4	0,4	0,3	0,3	0,7	0,9	1,5	2,4	2,9	3,9
33	2,0	1,8	1,2	0,9	0,9	0,9	2,5	3,1	7,8	16,1	24,8	35,3
34	1,2	1,1	1,0	0,8	0,8	0,9	2,4	3,3	9,1	16,8	22,7	31,9
35	0,5	0,5	0,4	0,4	0,6	0,6	2,0	3,6	10,3	17,4	23,9	31,9
36	0,4	0,4	0,9	0,8	1,0	1,0	1,8	2,9	6,3	8,9	11,1	13,8
37	2,8	2,7	1,8	1,3	1,2	1,2	1,6	2,7	7,3	16,0	24,0	34,7
38	3,0	2,9	2,4	2,0	1,9	2,0	2,0	3,1	8,7	16,1	22,1	30,3
39	1,4	1,3	0,9	1,0	0,8	0,8	2,6	2,7	8,8	16,5	22,2	30,0
40	1,4	1,4	1,1	1,2	0,9	0,9	1,5	2,2	4,6	7,8	9,8	12,7
41	4,5	4,1	3,1	2,5	2,4	2,5	4,6	5,9	16,2	36,4	54,2	79,9
42	3,9	3,7	3,4	2,6	2,5	2,6	4,8	7,1	20,2	35,3	49,6	71,3
43	1,0	0,9	0,9	1,0	1,4	1,4	3,9	7,9	24,2	38,3	51,8	70,7
44	0,8	0,8	2,2	1,9	2,3	2,3	3,5	6,5	13,6	18,5	23,1	29,4
45	10,5	9,9	6,2	3,7	3,6	3,4	3,1	5,6	15,8	34,2	53,1	81,2
46	9,0	8,8	7,2	5,5	5,3	5,0	3,7	6,7	18,8	35,1	50,2	70,9
47	1,8	1,8	2,5	1,8	2,5	2,4	4,8	7,7	22,4	39,1	50,3	68,7
48	2,6	2,6	2,4	2,6	2,8	2,7	3,5	7,6	12,3	18,1	23,1	29,5

Tabela C.26: Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP G1 e G2 de acordo com variação de alfa para instâncias de 20 e 32 locações

Inst.	G1						G2					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	14,6	11,6	6,9	7,0	6,4	15,2	4,5	10,0	6,9	6,8	6,6	19,4
50	15,3	14,6	7,6	7,5	7,2	19,5	5,4	10,7	7,6	7,3	7,2	15,6
51	20,5	17,2	8,1	7,2	7,5	18,0	8,9	14,8	8,0	7,0	7,4	12,0
52	15,2	11,1	5,7	4,9	4,8	10,9	5,9	9,5	5,7	4,8	4,7	8,0
53	15,6	16,4	7,6	7,2	6,5	14,1	7,0	28,4	7,6	7,0	6,4	15,3
54	24,9	23,3	8,5	8,0	7,7	20,5	11,2	31,8	8,5	7,8	7,5	15,5
55	21,3	23,0	7,4	8,1	8,0	18,7	8,3	24,1	7,5	7,9	7,9	17,3
56	21,0	21,2	6,9	6,7	6,6	15,3	9,3	25,0	6,9	6,6	6,6	20,6
57	57,8	67,3	24,5	25,1	23,5	64,2	24,4	51,6	24,6	24,5	23,9	60,6
58	58,9	69,8	23,8	22,8	22,3	55,6	20,2	45,5	23,7	22,4	22,3	52,3
59	37,1	55,0	23,0	21,5	21,9	54,6	14,3	71,3	22,8	21,3	21,6	36,6
60	24,3	39,3	14,7	13,8	14,0	31,0	10,6	48,6	14,6	13,7	14,0	47,8
61	86,3	80,6	31,1	30,0	28,5	74,6	30,0	76,0	31,1	30,0	28,6	76,4
62	84,1	88,5	31,6	29,4	28,1	70,8	32,9	71,5	31,3	30,1	27,6	72,0
63	82,7	68,6	27,3	28,5	27,5	57,3	33,0	49,0	27,1	28,7	27,5	62,7
64	52,3	60,4	23,7	23,2	21,4	58,0	19,4	56,9	23,5	23,2	21,2	48,9
65	164,6	160,5	56,8	55,3	51,7	139,7	66,3	141,4	56,4	55,7	51,3	110,5
66	140,5	139,4	55,0	52,8	49,4	122,7	56,8	126,4	55,2	52,7	49,6	96,8
67	149,4	137,1	50,9	49,3	47,3	119,8	54,2	101,5	51,0	48,6	47,6	107,1
68	65,6	93,0	33,7	32,1	31,0	72,0	23,0	78,2	33,4	31,6	30,8	90,5
69	262,9	206,8	73,8	70,1	66,5	163,1	105,3	171,0	74,0	70,0	65,4	159,3
70	227,9	191,0	72,4	70,3	66,2	166,1	92,7	197,5	72,4	68,6	66,0	120,0
71	206,9	157,3	69,6	70,9	67,9	169,2	81,5	173,0	69,7	69,7	67,4	150,9
72	117,7	108,9	53,0	52,9	49,1	125,5	46,0	126,7	52,4	52,0	48,6	115,9
73	152,1	149,1	64,4	55,8	53,0	140,9	60,1	134,4	64,6	54,9	52,3	120,2
74	178,7	160,2	71,0	66,8	64,3	165,6	75,0	133,3	71,3	66,2	63,2	133,9
75	190,4	200,3	76,6	74,5	72,9	191,9	75,3	207,5	76,9	73,6	72,1	169,8
76	143,5	125,5	52,0	52,0	51,1	127,6	61,7	94,3	52,0	51,3	52,0	101,0
77	199,2	165,4	60,2	55,7	52,5	132,2	80,0	109,5	59,7	56,0	52,2	137,7
78	194,4	199,0	75,4	71,7	68,9	169,2	80,1	166,5	76,0	70,7	68,8	161,1
79	303,7	245,5	94,3	93,8	93,7	214,3	114,7	166,4	95,3	92,3	92,4	193,1
80	156,5	172,5	75,1	71,3	70,3	165,6	59,0	199,2	74,8	69,1	71,6	146,9
81	776,5	621,8	274,1	259,1	242,1	609,0	317,5	610,2	274,2	249,2	240,1	613,0
82	597,5	621,8	252,7	238,5	230,5	555,1	244,3	574,0	252,0	237,9	229,0	617,4
83	586,6	595,3	255,9	255,7	252,3	596,2	235,8	553,8	255,9	254,1	249,0	504,8
84	489,3	408,6	170,4	173,3	174,4	428,2	194,9	469,3	170,3	170,4	172,7	426,4
85	647,9	697,9	276,6	263,5	248,2	596,1	265,7	566,2	274,7	264,6	246,6	577,5
86	569,5	673,7	263,0	256,2	241,7	564,1	234,1	267,9	267,3	251,9	238,6	544,2
87	750,3	751,4	295,8	301,3	291,2	667,8	320,9	304,3	296,5	295,9	285,5	715,2
88	237,7	509,2	224,0	216,3	216,3	481,1	234,9	232,2	224,1	215,6	213,4	538,9
89	696,1	1565,0	713,1	675,6	638,8	1524,2	698,3	714,8	715,7	667,6	638,2	636,3
90	731,9	1458,3	633,5	610,0	585,2	1408,0	717,7	627,5	637,9	606,8	576,9	1039,7
91	616,3	1446,1	610,3	618,4	609,8	1514,0	617,7	642,4	618,5	619,3	604,5	2067,7
92	416,2	780,1	380,7	389,8	384,0	922,7	415,0	369,8	378,8	389,2	380,1	1334,9
93	794,1	1843,5	717,7	692,5	650,8	1522,3	795,3	736,1	726,7	686,1	647,1	1269,3
94	857,1	1796,6	705,4	664,2	628,8	1438,1	871,6	703,0	705,1	662,3	628,8	630,0
95	585,4	1851,1	777,4	752,4	727,5	1643,6	590,6	765,1	762,8	750,8	731,9	718,9
96	518,0	1097,3	510,8	498,1	486,8	1135,2	519,5	502,3	513,7	498,2	494,0	494,9

Tabela C.27: Tempo médio (em segundos) de 10 execuções do G3 e G4 de acordo com variação de alfa para instâncias de 20 e 32 locações

Inst.	G3						G4					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	4,8	5,2	5,8	6,1	6,3	6,7	4,8	5,4	5,9	6,2	6,5	6,7
50	4,8	5,2	6,0	6,3	6,8	7,2	4,9	5,2	6,0	6,5	7,1	7,2
51	8,9	9,2	8,5	7,6	7,8	7,7	9,0	9,2	8,6	7,8	8,0	7,8
52	6,4	6,1	6,2	5,3	5,1	5,1	6,4	6,2	6,2	5,5	5,3	5,1
53	5,3	5,8	6,2	6,0	6,5	6,9	5,3	5,8	6,2	6,1	6,7	6,9
54	6,8	6,4	6,7	7,0	7,0	7,3	6,9	6,5	6,7	6,9	7,4	7,4
55	8,4	7,3	7,9	8,3	8,4	8,3	8,5	7,5	8,0	8,3	8,8	8,5
56	8,5	7,4	7,2	7,2	7,0	6,8	8,6	7,5	7,2	7,3	7,2	7,0
57	13,3	18,1	20,4	21,1	22,3	23,3	13,4	18,5	20,4	21,4	23,4	23,5
58	16,0	16,3	17,5	19,1	20,0	20,7	16,4	16,1	17,6	19,2	20,7	21,0
59	16,2	22,7	24,4	22,8	22,8	22,9	16,1	22,9	24,5	23,1	23,8	23,3
60	11,7	14,8	16,1	15,0	15,4	15,3	11,8	14,9	16,1	15,2	15,5	15,2
61	26,6	25,9	27,7	28,5	27,9	29,4	27,5	25,6	27,8	28,2	28,6	29,1
62	28,2	24,9	25,7	26,3	27,3	28,2	28,1	24,8	25,6	26,8	27,0	28,2
63	34,3	28,3	28,9	30,0	29,6	29,6	34,8	28,0	29,0	30,5	29,7	29,4
64	20,6	26,3	25,6	24,7	24,0	23,9	20,4	26,7	25,7	25,2	23,5	23,5
65	38,0	42,4	45,8	48,2	49,5	50,2	37,2	42,9	46,4	48,4	49,4	50,3
66	41,9	37,0	39,1	41,8	43,0	46,5	42,4	36,8	39,0	42,6	44,1	45,8
67	57,4	57,3	53,8	51,3	49,4	50,5	57,7	57,9	53,7	51,5	51,9	50,1
68	26,2	42,5	37,1	34,7	33,9	33,7	26,1	42,2	37,0	35,0	34,8	33,4
69	60,1	56,7	59,2	61,2	62,8	64,4	61,0	54,9	59,6	62,0	64,4	64,0
70	55,6	53,3	56,2	58,2	59,3	62,2	55,2	53,0	55,5	58,9	61,1	63,0
71	85,0	67,6	72,7	75,8	71,9	70,3	84,9	68,2	73,2	75,2	73,3	70,1
72	46,6	52,6	56,0	55,3	52,8	51,9	46,4	52,4	56,1	56,1	53,0	52,9
73	43,3	47,4	49,0	50,8	52,9	55,3	43,6	50,1	49,2	50,5	52,8	55,4
74	53,3	49,1	52,3	55,8	57,8	60,2	53,5	49,7	53,8	57,6	58,8	59,9
75	72,0	87,4	78,2	78,4	74,6	73,7	71,4	88,0	78,9	77,4	76,6	74,7
76	46,8	56,1	54,8	54,0	54,4	53,1	47,6	56,4	55,3	54,4	54,4	53,2
77	43,2	44,0	47,1	50,4	52,1	52,6	42,8	44,8	47,1	50,8	52,3	53,5
78	62,5	55,4	60,5	63,1	64,7	66,8	63,6	56,2	61,1	63,8	64,3	68,2
79	119,0	98,4	96,6	94,3	92,3	91,6	118,8	99,1	96,3	95,4	93,4	93,9
80	60,7	75,8	77,1	75,5	71,8	72,1	60,2	76,7	77,3	75,3	72,7	71,9
81	188,7	195,0	206,4	217,4	225,9	230,7	189,6	195,2	206,3	221,7	228,9	234,6
82	190,8	179,3	185,0	196,0	198,8	207,3	197,8	179,4	184,8	199,3	199,5	208,3
83	230,5	266,4	268,3	263,1	260,6	258,7	236,5	259,2	266,9	268,2	264,9	256,2
84	203,8	178,5	179,7	185,5	185,2	181,6	212,6	177,0	181,1	182,5	190,5	182,0
85	209,2	216,1	221,8	232,4	235,9	245,7	211,4	216,3	221,7	233,9	242,2	243,5
86	213,8	192,6	206,4	221,5	219,8	227,1	214,0	198,9	206,2	213,0	230,2	228,2
87	329,9	313,3	308,7	314,7	298,7	293,5	334,7	326,5	304,6	312,0	311,7	298,0
88	247,0	237,8	233,8	227,0	230,0	219,1	247,7	251,8	232,8	224,8	228,0	220,9
89	466,5	492,8	521,2	548,4	575,1	588,1	469,4	496,1	520,5	549,2	569,3	593,4
90	431,6	452,1	448,6	466,7	476,2	496,2	420,4	436,2	448,8	484,4	485,3	508,1
91	629,1	678,8	640,5	625,2	621,1	607,7	614,5	648,1	641,4	640,1	641,8	607,6
92	441,9	396,4	397,4	404,1	399,8	404,0	437,2	380,4	400,2	408,1	400,3	444,4
93	549,8	555,6	570,5	609,3	606,1	641,4	556,2	552,2	573,1	613,6	606,3	605,3
94	490,6	485,7	523,4	549,1	564,0	567,9	485,4	486,5	524,8	547,0	567,7	573,4
95	609,6	781,1	775,6	775,7	748,3	731,7	611,9	781,6	772,1	780,0	750,2	724,2
96	548,3	538,7	531,5	521,3	512,9	509,3	542,5	514,2	529,6	527,2	512,9	513,0

Tabela C.28: Tempo médio (em segundos) de 10 execuções do G5 e G6 de acordo com variação de alfa para instâncias de 20 e 32 locações

Inst.	G5						G6					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	8,4	8,8	9,6	10,3	10,2	10,1	1,1	2,2	1,9	2,2	1,7	1,9
50	6,9	7,3	8,1	8,7	9,2	9,2	3,0	2,5	2,8	3,0	2,8	3,0
51	11,0	11,2	10,6	9,8	10,0	9,8	2,7	2,3	2,5	2,2	2,4	2,3
52	7,0	6,6	6,7	6,0	5,8	5,6	2,9	2,8	3,5	2,8	2,5	2,4
53	9,0	9,7	10,1	10,3	10,4	10,7	2,2	1,8	2,5	2,0	1,8	1,9
54	9,7	9,5	9,6	10,0	10,2	10,3	1,6	2,2	3,3	2,9	2,6	2,6
55	10,6	9,6	10,1	10,5	10,6	10,4	1,2	2,1	2,0	2,0	2,1	2,1
56	9,4	8,2	8,2	8,1	8,0	7,7	3,1	3,4	3,0	2,6	2,6	2,7
57	22,0	26,4	29,1	30,1	31,0	31,4	5,4	6,7	5,9	5,5	5,1	5,4
58	21,6	20,8	23,2	24,6	25,3	26,2	9,0	7,8	7,3	6,9	6,9	7,3
59	20,7	27,0	28,6	27,3	28,6	27,1	6,1	5,4	6,8	6,8	6,8	6,7
60	13,2	16,2	17,6	16,7	17,9	16,3	11,8	13,9	10,8	8,5	7,7	7,4
61	35,3	35,0	36,6	38,2	40,6	37,9	10,3	6,9	8,2	7,0	6,5	6,7
62	34,5	31,1	32,0	33,3	35,8	34,8	16,5	9,7	12,4	11,7	10,4	10,9
63	39,7	31,6	33,3	35,3	35,3	33,3	7,2	6,5	8,7	9,2	8,5	8,1
64	22,0	28,3	27,6	27,0	27,6	25,0	10,0	11,9	12,5	11,8	11,3	10,9
65	52,3	57,8	62,0	64,4	68,7	65,9	15,3	17,7	13,2	12,7	12,0	12,2
66	50,3	45,7	48,7	53,0	56,2	55,6	31,8	19,9	17,2	17,0	16,4	17,0
67	65,4	65,1	61,9	59,3	61,3	57,3	15,1	15,5	15,6	16,2	14,4	14,8
68	28,3	43,7	38,9	37,1	36,7	34,8	19,4	22,1	22,8	20,3	16,4	15,2
69	74,4	72,0	75,3	77,1	83,0	79,6	18,6	18,0	17,8	15,9	14,9	15,6
70	67,2	65,9	68,6	71,0	76,0	73,1	42,4	25,7	27,6	25,2	23,3	22,9
71	94,6	78,1	81,4	84,3	85,9	78,4	17,4	22,1	19,8	21,4	20,0	20,0
72	50,1	56,0	59,1	59,2	59,1	56,3	39,1	32,0	29,5	27,8	27,6	26,4
73	65,7	68,9	72,1	72,4	74,7	77,4	11,2	12,0	11,5	9,8	9,4	10,0
74	66,3	63,1	67,1	70,3	70,6	75,1	9,3	16,0	17,7	15,7	15,0	15,0
75	82,9	99,7	90,7	90,2	87,8	86,9	20,0	20,2	15,9	16,8	16,1	15,7
76	50,3	58,6	59,8	60,3	57,6	57,9	17,1	24,7	22,0	20,1	19,9	18,5
77	65,2	66,4	70,2	75,9	74,2	76,2	18,6	12,8	11,8	10,5	9,8	10,0
78	79,9	74,0	77,1	80,5	83,1	88,0	32,4	21,4	20,3	17,7	17,0	16,6
79	130,7	111,1	109,7	109,1	107,7	105,4	13,2	18,2	20,7	18,7	16,7	16,1
80	65,0	80,1	82,4	80,1	81,9	76,4	16,2	28,0	32,2	30,3	26,9	24,7
81	236,8	241,8	254,4	268,5	285,6	286,5	65,2	51,8	49,4	42,0	37,6	37,6
82	222,9	212,5	222,4	228,8	234,2	242,6	86,7	73,3	76,9	69,0	58,0	60,1
83	259,6	288,7	299,4	298,3	291,3	290,1	67,0	79,8	74,2	68,0	65,2	61,9
84	215,7	187,4	192,0	193,6	193,5	196,1	38,5	89,9	92,9	83,1	85,0	80,0
85	256,6	270,8	273,6	286,5	289,8	291,1	80,1	58,5	53,3	46,6	41,3	41,0
86	250,5	229,1	243,2	252,6	259,6	260,5	99,3	86,9	81,7	76,3	69,8	69,8
87	359,5	343,6	336,0	337,3	329,1	327,4	57,9	61,5	66,0	67,7	64,8	62,3
88	256,9	247,6	243,9	238,1	232,5	233,7	142,8	102,5	109,4	106,2	96,6	91,2
89	558,0	582,6	610,6	655,7	671,9	678,2	172,5	116,9	120,0	106,1	95,8	97,4
90	483,1	486,3	519,0	569,3	537,6	583,5	144,9	178,6	180,7	165,2	150,6	148,0
91	660,0	699,2	681,4	681,1	665,2	662,6	264,3	203,0	171,3	165,1	157,6	143,3
92	449,9	401,7	413,5	428,2	414,9	417,2	109,6	182,9	174,5	163,9	164,7	157,3
93	645,0	652,9	672,1	714,2	710,9	714,5	227,7	146,2	140,3	124,2	109,5	110,7
94	561,6	564,1	587,9	616,9	637,0	656,9	253,8	197,6	204,0	189,0	175,7	178,9
95	662,2	839,7	826,6	819,1	790,7	790,8	169,9	194,3	193,8	196,5	180,3	176,7
96	559,1	531,0	545,7	534,6	525,8	517,8	323,8	273,2	275,5	265,1	244,4	220,0

Tabela C.29: Tempo médio (em segundos) de 10 execuções do G7 e G8 de acordo com variação de alfa para instâncias de 20 e 32 locações

Inst.	G7						G8					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	9,7	12,3	12,7	12,7	12,9	12,8	1,3	1,5	1,8	2,0	2,0	2,1
50	10,4	9,6	12,4	12,4	11,5	11,4	1,3	1,6	2,0	2,4	2,7	2,9
51	12,7	9,3	7,2	7,3	6,5	6,6	3,1	2,7	2,9	2,7	2,7	2,8
52	3,6	4,4	4,1	3,4	3,0	2,9	3,5	3,3	4,0	3,3	3,0	2,9
53	12,6	12,1	17,2	16,4	16,2	16,7	2,4	2,4	2,0	2,0	1,9	2,1
54	13,4	16,3	16,7	16,0	15,1	15,4	2,5	2,9	2,7	2,5	2,5	2,8
55	5,3	7,9	7,8	7,8	7,4	7,4	1,8	2,5	2,4	2,5	2,6	2,5
56	5,9	5,2	4,6	4,4	4,3	4,4	3,1	3,8	3,4	3,2	3,0	3,2
57	60,7	42,7	33,5	36,4	34,7	35,9	3,2	4,3	5,3	5,4	5,6	5,7
58	37,6	36,6	32,9	33,1	31,3	31,5	4,8	4,8	5,7	6,0	6,4	6,8
59	29,9	21,7	23,9	25,8	21,3	22,2	7,8	6,9	8,6	8,6	8,7	8,6
60	10,7	12,2	13,1	12,1	10,1	10,3	14,5	16,4	13,2	10,9	9,6	9,4
61	58,4	45,9	53,3	54,0	50,4	52,5	7,2	6,8	7,5	7,2	7,0	7,5
62	81,5	64,7	66,7	65,0	61,7	61,4	7,2	9,4	10,2	10,5	10,2	10,7
63	15,5	18,5	26,6	28,0	25,7	25,8	9,8	8,8	11,2	12,3	11,2	10,8
64	21,5	20,3	18,0	19,1	18,7	17,7	11,2	14,5	15,9	14,9	14,1	14,3
65	89,0	87,5	78,3	81,9	78,2	83,2	9,6	10,0	11,3	12,1	12,7	12,6
66	80,9	79,5	77,8	77,7	76,9	76,6	16,2	12,9	13,8	14,1	14,8	15,7
67	52,4	51,3	47,8	48,7	38,0	40,1	18,3	18,9	19,8	19,9	17,3	18,6
68	22,2	28,9	25,4	24,1	20,0	18,8	24,5	26,2	26,7	24,0	20,6	19,2
69	114,2	89,2	109,7	111,5	108,4	108,9	12,1	13,4	15,2	15,6	15,8	16,1
70	104,6	123,2	133,7	129,0	121,3	121,9	15,2	17,3	21,0	21,0	21,3	22,0
71	36,3	57,0	62,7	62,9	62,2	65,1	21,6	27,7	24,9	26,2	25,5	24,6
72	63,6	51,0	48,6	46,2	44,8	43,9	32,6	37,7	35,6	33,1	31,0	30,7
73	86,6	91,3	82,5	79,3	75,7	79,1	5,8	8,0	8,8	9,9	10,6	10,6
74	70,7	90,2	84,4	72,3	64,7	63,1	11,7	11,5	11,3	12,8	13,5	13,9
75	59,3	50,6	46,7	47,4	47,8	49,1	19,7	23,5	19,5	20,0	19,4	19,0
76	38,8	30,4	26,2	24,1	22,4	21,6	15,6	27,9	26,2	24,1	23,4	21,9
77	134,2	110,2	99,4	91,0	80,6	82,4	9,5	8,8	8,6	10,0	10,2	10,4
78	186,5	133,9	115,4	109,5	97,6	99,6	12,5	13,1	12,9	14,1	15,0	16,0
79	62,1	52,8	56,0	51,4	48,6	49,1	17,2	22,6	25,2	22,7	20,4	20,0
80	28,9	49,9	48,8	45,5	40,4	37,5	19,9	31,9	38,6	36,2	31,0	29,3
81	258,0	282,5	271,6	246,6	233,1	231,0	41,1	31,3	31,8	34,7	37,4	37,6
82	226,0	283,1	279,4	236,5	209,6	204,3	40,1	40,8	44,1	46,5	50,0	52,7
83	194,4	204,4	172,5	173,3	164,7	166,4	66,6	92,4	88,0	79,6	76,8	74,1
84	149,3	118,6	119,9	108,9	109,4	107,4	48,3	104,5	109,8	97,4	100,3	95,3
85	415,6	382,2	341,5	297,4	281,0	270,2	39,5	40,4	36,9	41,3	41,4	43,1
86	489,3	399,6	360,1	328,4	287,7	293,7	59,4	53,4	54,1	57,0	59,4	62,3
87	165,7	170,5	172,2	173,6	175,3	164,2	71,1	78,9	79,2	83,2	78,2	77,8
88	173,3	144,5	149,7	145,0	132,5	120,1	162,3	121,3	128,8	122,2	116,3	105,4
89	675,7	660,6	563,3	514,8	482,6	489,3	83,1	71,4	78,4	85,6	89,3	91,6
90	761,2	651,0	591,8	503,4	432,6	422,3	80,4	94,2	95,6	99,5	106,9	112,8
91	282,8	415,1	366,7	368,5	340,5	331,0	242,5	229,9	194,3	182,7	177,2	167,0
92	148,2	204,8	199,0	184,0	184,9	181,0	133,6	203,3	197,2	190,0	183,6	188,7
93	1073,4	812,0	785,2	683,0	628,7	622,1	99,5	96,3	99,6	105,9	108,4	109,9
94	1371,7	872,9	867,0	788,3	740,9	712,3	107,5	121,2	135,0	140,0	145,1	151,9
95	357,7	449,9	453,4	434,6	396,6	378,2	191,5	221,8	220,6	219,2	212,6	205,5
96	223,0	323,8	326,7	308,9	294,5	259,4	354,0	296,1	306,3	294,4	277,2	258,6

Tabela C.30: Tempo médio (em segundos) de 10 execuções do G9 e G10 de acordo com variação de alfa para instâncias de 20 e 32 locações

Inst.	G9						G10					
	0,0	0,2	0,4	0,6	0,8	1,0	0,0	0,2	0,4	0,6	0,8	1,0
49	11,8	11,8	12,5	12,9	13,5	13,5	7,7	13,3	32,8	55,2	78,6	102,1
50	11,4	11,2	11,5	10,9	11,7	11,9	4,3	14,0	34,2	51,5	66,7	82,2
51	13,6	10,1	8,1	7,9	7,3	7,3	7,0	17,4	37,7	53,6	66,7	82,2
52	4,3	5,1	4,7	3,9	3,5	3,4	3,1	11,0	18,6	21,9	26,0	30,4
53	21,4	17,1	14,6	16,0	16,5	17,0	6,0	14,0	35,6	58,9	86,9	110,4
54	17,7	17,1	16,1	15,4	15,2	15,3	6,9	15,3	38,8	57,8	78,5	101,4
55	6,4	8,4	8,3	8,5	8,0	8,2	5,5	18,5	41,9	61,3	79,4	95,2
56	6,3	5,5	5,3	5,1	4,8	5,0	5,9	17,0	27,8	34,7	44,0	49,8
57	35,4	33,1	34,2	35,3	36,1	36,3	15,0	33,8	89,6	164,3	243,9	324,0
58	32,5	31,7	30,7	30,2	30,6	31,9	11,5	38,3	99,4	156,7	213,7	269,2
59	34,0	24,7	27,8	28,5	25,5	26,0	18,7	45,2	109,1	160,8	213,8	261,5
60	13,1	14,4	15,7	14,4	12,4	12,6	10,2	33,4	55,5	68,3	84,1	97,7
61	57,3	52,3	49,5	48,9	49,6	52,3	14,4	35,0	92,8	172,2	260,9	345,9
62	52,3	64,7	62,0	59,7	62,0	63,7	16,6	45,4	112,4	184,8	254,8	318,9
63	19,9	23,2	32,3	32,2	31,2	30,9	25,3	65,2	140,1	198,1	255,6	312,5
64	22,7	25,2	22,3	22,9	22,1	22,2	23,0	62,3	95,7	118,5	142,8	168,5
65	65,0	70,5	74,7	79,4	80,7	82,8	29,1	72,7	203,0	379,6	574,1	760,8
66	83,2	81,1	80,3	76,9	75,6	77,4	23,1	85,9	228,6	378,1	500,0	627,6
67	56,8	53,9	52,5	51,7	44,6	43,4	35,3	107,1	259,7	381,1	506,7	602,9
68	26,8	33,9	30,1	28,4	24,1	21,8	28,7	80,6	128,9	158,7	186,8	218,6
69	133,9	104,8	98,3	106,0	106,2	108,3	27,7	73,9	205,9	381,2	583,3	777,2
70	111,8	126,6	124,9	121,7	119,8	119,1	30,4	95,3	255,1	421,0	562,7	710,1
71	43,5	64,3	70,0	70,9	69,9	67,1	49,2	130,5	279,0	425,0	541,2	662,1
72	59,2	56,5	53,1	51,9	50,0	48,5	35,0	122,9	190,1	242,8	285,6	337,7
73	72,3	63,1	59,6	72,3	77,8	76,3	43,1	117,2	265,2	446,1	599,8	775,8
74	63,0	66,4	63,9	61,9	63,3	62,0	41,8	128,8	276,0	409,9	510,4	633,4
75	59,4	57,2	52,6	52,7	55,2	55,8	49,2	178,3	340,6	463,7	564,0	669,7
76	29,1	33,1	31,0	27,8	26,3	25,7	18,5	121,8	173,9	208,6	240,2	274,1
77	94,0	81,1	68,9	78,2	83,1	84,1	41,8	122,0	249,1	415,6	567,8	748,2
78	83,7	95,7	92,3	96,2	96,6	97,1	37,0	132,3	286,3	431,5	573,5	704,4
79	74,1	61,1	63,4	61,8	57,1	56,8	22,4	186,3	339,1	466,1	577,8	697,4
80	34,4	56,1	57,8	52,7	45,9	43,6	57,7	156,3	228,0	275,8	321,4	385,5
81	275,4	215,0	196,7	218,9	231,0	236,8	100,3	299,7	687,0	1237,6	1733,5	2238,2
82	195,9	205,9	206,6	204,8	202,8	202,7	117,5	328,8	752,3	1140,8	1497,8	1860,2
83	193,8	228,7	192,5	197,0	191,0	190,5	151,3	474,4	915,5	1353,8	1698,9	2088,9
84	170,8	135,4	134,7	119,6	125,0	121,7	48,3	361,9	506,3	641,1	762,5	890,8
85	301,2	256,6	235,9	256,1	270,0	274,6	101,4	332,5	708,7	1254,7	1782,5	2323,2
86	329,2	304,4	287,6	292,4	293,0	291,5	104,3	391,4	817,3	1315,9	1712,7	2155,4
87	183,8	196,3	194,8	201,7	193,9	190,7	125,5	579,8	1032,6	1472,6	1827,5	2186,9
88	199,6	171,6	168,2	166,9	151,1	139,8	143,1	453,8	655,3	808,8	953,5	1086,4
89	396,8	436,1	441,5	457,8	476,7	482,5	172,8	622,4	1478,9	2799,9	4038,3	5200,5
90	408,5	414,5	409,9	396,2	392,8	405,7	233,8	755,8	1728,3	2786,7	3649,4	4447,8
91	321,0	452,2	414,7	387,9	363,5	363,5	315,4	991,4	1974,1	2806,5	3555,1	4264,2
92	170,9	225,4	218,9	209,0	214,7	207,7	231,3	740,0	1027,3	1269,8	1508,7	1730,1
93	639,4	573,7	552,9	602,0	609,3	609,6	186,3	719,2	1612,8	2967,8	4239,3	5470,1
94	653,3	745,6	704,7	712,9	691,5	693,1	181,9	857,8	1870,6	3105,1	4053,1	5001,8
95	385,8	493,5	495,4	465,0	424,9	407,4	304,2	1269,2	2324,9	3268,4	3901,3	4699,6
96	252,5	357,6	356,4	347,0	309,8	287,9	541,9	1045,2	1358,1	1666,1	1818,1	2088,9

Apêndice D

Resultados Computacionais dos Algoritmos Tabu Propostos

Os algoritmos Tabu propostos no Capítulo 5, Seção 5.4, foram testados e os resultados obtidos são apresentados a seguir.

As Tabelas D.1 e D.2 apresentam, respectivamente, os melhores custos obtidos com 10 execuções dos algoritmos Tabu, construções e tamanhos de lista tabu **C1t4**, **C2t8**, **C3t5**, **C4t5**, **C5t5**, **C6t5**, **C7t3**, **C8t3** e **C9t7**. Em negrito encontra-se destacado o melhor valor encontrado para cada instância.

As Tabelas D.3 e D.4 apresentam, respectivamente, os desvios percentuais entre o melhor custo conhecido e os melhores custos obtidos com 10 execuções dos algoritmos Tabu, construções e tamanhos de lista tabu **C1t4**, **C2t8**, **C3t5**, **C4t5**, **C5t5**, **C6t5**, **C7t3**, **C8t3** e **C9t7**. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo GRASP Reativo é menor do que o da literatura.

As Tabelas D.5 e D.6 apresentam, respectivamente, o tempo médio em segundos de 10 execuções dos algoritmos Tabu, construções e tamanhos de lista tabu **C1t4**, **C2t8**, **C3t5**, **C4t5**, **C5t5**, **C6t5**, **C7t3**, **C8t3** e **C9t7**.

Tabela D.1: Melhor solução (obtida de 10 execuções) da combinação de construção e tamanho de lista tabu para instâncias de 6 e 12 locações

Inst.	C1t4	C2t8	C3t5	C4t5	C5t5	C6t5	C7t3	C8t3	C9t7
1	16	16	16	16	16	16	16	16	16
2	26	26	26	26	26	26	26	26	26
3	18	18	18	18	18	18	18	18	18
4	26	26	26	26	26	26	26	26	26
5	16	16	16	16	16	16	16	16	16
6	27	27	27	27	27	27	27	27	27
7	16	16	16	16	16	16	16	16	16
8	31	31	31	31	31	31	31	31	31
9	25	25	25	25	25	25	25	25	25
10	46	46	46	46	46	46	46	46	46
11	32	32	32	32	32	32	32	32	32
12	43	43	43	43	43	43	43	43	43
13	28	28	28	28	28	28	28	28	28
14	45	45	45	45	45	45	45	45	45
15	35	35	35	35	35	35	35	35	35
16	49	49	49	49	49	49	49	49	49
17	35	35	35	35	35	35	35	35	35
18	62	62	62	62	62	62	62	62	62
19	46	46	46	46	46	46	46	46	46
20	63	63	63	63	63	63	63	63	62
21	48	48	48	48	48	48	48	48	48
22	67	67	67	67	67	67	67	67	67
23	56	56	56	56	56	56	56	56	56
24	74	74	74	74	74	74	74	74	74
25	31	31	31	31	31	31	31	31	31
26	43	43	43	43	43	43	43	43	43
27	43	43	43	43	43	43	43	43	43
28	55	55	55	55	55	55	55	55	55
29	29	29	29	29	29	29	29	29	29
30	49	49	49	49	49	49	49	49	49
31	42	42	42	42	42	42	42	42	42
32	69	69	69	69	69	69	69	69	69
33	52	52	53	53	54	54	52	52	52
34	72	72	72	72	72	72	72	72	72
35	73	73	73	73	73	73	73	73	73
36	95	95	95	95	95	95	95	95	95
37	48	48	48	48	48	48	48	48	48
38	83	83	83	83	83	83	83	83	83
39	69	69	69	69	69	69	69	69	69
40	108	108	108	108	108	108	108	108	108
41	78	78	78	78	78	78	78	78	78
42	102	102	102	102	102	102	102	102	102
43	110	110	110	110	110	110	110	110	110
44	140	140	140	140	140	140	140	140	140
45	67	66	66	66	66	66	66	66	66
46	116	116	116	116	116	116	116	116	116
47	115	115	115	115	115	115	115	115	115
48	171	171	171	171	171	171	171	171	171

Tabela D.2: Melhor solução (obtida de 10 execuções) da combinação de construção e tamanho de lista tabu para instâncias de 20 e 32 locações

Inst.	C1t4	C2t8	C3t5	C4t5	C5t5	C6t5	C7t3	C8t3	C9t7
49	44	44	44	44	44	44	44	44	44
50	60	60	60	60	60	60	60	60	60
51	53	53	53	53	53	53	53	53	53
52	89	89	89	89	89	89	89	89	89
53	47	47	47	47	47	47	47	47	47
54	62	62	62	62	62	62	62	62	62
55	60	60	60	60	60	60	60	60	60
56	90	89	89	89	90	90	89	89	89
57	66	66	66	66	66	66	66	66	66
58	101	100	101	101	101	101	100	100	101
59	93	93	93	93	93	93	93	93	93
60	153	153	153	153	153	153	153	153	153
61	75	74	74	74	74	74	74	74	74
62	113	112	112	112	112	112	112	112	112
63	116	118	116	116	116	116	116	116	116
64	175	176	176	176	176	176	175	175	175
65	99	100	98	98	98	98	101	101	98
66	152	152	153	153	152	152	151	151	152
67	145	147	144	144	147	147	147	147	147
68	221	220	220	220	220	220	220	220	220
69	115	113	114	114	116	116	112	112	112
70	166	166	167	167	166	166	166	166	166
71	162	162	162	162	164	164	165	165	166
72	248	252	248	248	248	248	248	248	248
73	71	71	71	71	71	71	71	71	71
74	90	90	91	91	91	91	90	90	91
75	102	102	101	101	103	103	101	101	102
76	144	144	144	144	144	144	144	144	144
77	70	70	70	70	70	70	70	70	70
78	93	92	93	93	93	93	92	92	93
79	99	99	99	99	99	99	99	99	100
80	163	163	163	163	163	163	165	165	166
81	109	111	111	111	110	110	109	109	109
82	162	161	162	162	167	167	166	166	164
83	181	185	183	183	189	189	188	188	186
84	270	270	268	268	267	267	269	269	272
85	117	119	119	119	119	119	117	117	119
86	183	180	177	177	181	181	182	182	182
87	183	186	181	181	182	182	182	182	180
88	276	277	276	276	275	275	275	275	274
89	167	163	162	162	162	162	161	161	163
90	248	245	251	251	251	251	243	243	241
91	274	274	274	274	269	269	272	272	278
92	380	383	388	388	382	382	388	388	387
93	175	181	178	178	180	180	178	178	181
94	270	279	277	277	269	269	271	271	271
95	308	297	308	308	307	307	307	307	306
96	449	450	439	439	449	449	439	439	436

Tabela D.3: Desvio percentual entre a melhor solução da literatura e da combinação de construção e tamanho de lista tabu para instâncias de 6 e 12 locações

Inst.	C1t4	C2t8	C3t5	C4t5	C5t5	C6t5	C7t3	C8t3	C9t7
1	-	-	-	-	-	-	-	-	-
2	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8
3	-	-	-	-	-	-	-	-	-
4	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8	3,8
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-
12	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7
13	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-
18	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2
19	-	-	-	-	-	-	-	-	-
20	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	3,2
21	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2
22	-	-	-	-	-	-	-	-	-
23	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8
24	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-
30	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0
31	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-
33	-1,9	-1,9	-	-	1,9	1,9	-1,9	-1,9	-1,9
34	-	-	-	-	-	-	-	-	-
35	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8
36	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3
37	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1
38	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2
39	2,9	2,9	2,9	2,9	2,9	2,9	2,9	2,9	2,9
40	3,7	3,7	3,7	3,7	3,7	3,7	3,7	3,7	3,7
41	-	-	-	-	-	-	-	-	-
42	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0
43	-	-	-	-	-	-	-	-	-
44	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1
45	1,5	-	-	-	-	-	-	-	-
46	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3
47	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5
48	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2

Tabela D.4: Desvio percentual entre a melhor solução da literatura e da combinação de construção e tamanho de lista tabu para instâncias de 20 e 32 locações

Inst.	C1t4	C2t8	C3t5	C4t5	C5t5	C6t5	C7t3	C8t3	C9t7
49	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3
50	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0
51	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8
52	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1
53	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
54	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1
55	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0
56	-7,8	-9,0	-9,0	-9,0	-7,8	-7,8	-9,0	-9,0	-9,0
57	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5
58	-5,0	-6,0	-5,0	-5,0	-5,0	-5,0	-6,0	-6,0	-5,0
59	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6
60	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9
61	-9,3	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8
62	-14,2	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2
63	-4,3	-2,5	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
64	-8,6	-8,0	-8,0	-8,0	-8,0	-8,0	-8,6	-8,6	-8,6
65	-6,1	-5,0	-7,1	-7,1	-7,1	-7,1	-4,0	-4,0	-7,1
66	-2,6	-2,6	-2,0	-2,0	-2,6	-2,6	-3,3	-3,3	-2,6
67	-8,3	-6,8	-9,0	-9,0	-6,8	-6,8	-6,8	-6,8	-6,8
68	-5,9	-6,4	-6,4	-6,4	-6,4	-6,4	-6,4	-6,4	-6,4
69	2,6	0,9	1,8	1,8	3,4	3,4	0,0	0,0	0,0
70	-7,2	-7,2	-6,6	-6,6	-7,2	-7,2	-7,2	-7,2	-7,2
71	-4,9	-4,9	-4,9	-4,9	-3,7	-3,7	-3,0	-3,0	-2,4
72	-6,9	-5,2	-6,9	-6,9	-6,9	-6,9	-6,9	-6,9	-6,9
73	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2
74	-7,8	-7,8	-6,6	-6,6	-6,6	-6,6	-7,8	-7,8	-6,6
75	-7,8	-7,8	-8,9	-8,9	-6,8	-6,8	-8,9	-8,9	-7,8
76	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6
77	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
78	-8,6	-9,8	-8,6	-8,6	-8,6	-8,6	-9,8	-9,8	-8,6
79	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1	-10,0
80	-7,4	-7,4	-7,4	-7,4	-7,4	-7,4	-6,1	-6,1	-5,4
81	-9,2	-7,2	-7,2	-7,2	-8,2	-8,2	-9,2	-9,2	-9,2
82	-8,6	-9,3	-8,6	-8,6	-5,4	-5,4	-6,0	-6,0	-7,3
83	-6,1	-3,8	-4,9	-4,9	-1,6	-1,6	-2,1	-2,1	-3,2
84	-4,4	-4,4	-5,2	-5,2	-5,6	-5,6	-4,8	-4,8	-3,7
85	-6,8	-5,0	-5,0	-5,0	-5,0	-5,0	-6,8	-6,8	-5,0
86	-4,9	-6,7	-8,5	-8,5	-6,1	-6,1	-5,5	-5,5	-5,5
87	-5,5	-3,8	-6,6	-6,6	-6,0	-6,0	-6,0	-6,0	-7,2
88	-9,4	-9,0	-9,4	-9,4	-9,8	-9,8	-9,8	-9,8	-10,2
89	-2,4	-4,9	-5,6	-5,6	-5,6	-5,6	-6,2	-6,2	-4,9
90	-5,6	-6,9	-4,4	-4,4	-4,4	-4,4	-7,8	-7,8	-8,7
91	-3,6	-3,6	-3,6	-3,6	-5,6	-5,6	-4,4	-4,4	-2,2
92	-3,9	-3,1	-1,8	-1,8	-3,4	-3,4	-1,8	-1,8	-2,1
93	-8,0	-4,4	-6,2	-6,2	-5,0	-5,0	-6,2	-6,2	-4,4
94	-4,1	-0,7	-1,4	-1,4	-4,5	-4,5	-3,7	-3,7	-3,7
95	-3,2	-7,1	-3,2	-3,2	-3,6	-3,6	-3,6	-3,6	-3,9
96	-3,3	-3,1	-5,7	-5,7	-3,3	-3,3	-5,7	-5,7	-6,4

Tabela D.5: Tempo médio (em segundos) de 10 execuções da combinação de construção e tamanho de lista tabu para instâncias de 6 e 12 locações

Inst.	C1t4	C2t8	C3t5	C4t5	C5t5	C6t5	C7t3	C8t3	C9t7
1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
8	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
9	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
10	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
11	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
12	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
13	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
14	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
15	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
16	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
17	0,1	0,1	0,0	0,0	0,0	0,1	0,1	0,1	0,1
18	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
19	0,1	0,0	0,1	0,1	0,1	0,1	0,1	0,1	0,1
20	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
21	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1
22	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1
23	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
24	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
25	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,2	0,2
26	0,2	0,2	0,2	0,2	0,1	0,2	0,2	0,2	0,2
27	0,1	0,2	0,2	0,1	0,2	0,2	0,1	0,2	0,2
28	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
29	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
30	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
31	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
32	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
33	0,5	0,5	0,4	0,4	0,4	0,4	0,4	0,4	0,5
34	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4
35	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4
36	0,2	0,3	0,3	0,3	0,2	0,2	0,3	0,3	0,3
37	0,4	0,4	0,5	0,5	0,5	0,5	0,4	0,4	0,4
38	0,3	0,4	0,3	0,3	0,4	0,4	0,4	0,4	0,4
39	0,4	0,4	0,5	0,4	0,4	0,4	0,5	0,5	0,5
40	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
41	0,8	0,9	0,9	0,9	0,8	0,8	0,9	0,9	0,9
42	0,7	0,8	0,7	0,7	0,8	0,8	0,7	0,7	0,7
43	0,8	0,9	0,7	0,7	0,9	1,0	0,9	1,0	0,8
44	0,4	0,5	0,5	0,4	0,5	0,5	0,4	0,4	0,4
45	0,8	1,0	0,9	0,9	0,8	0,8	0,9	0,9	0,9
46	0,7	0,7	0,6	0,6	0,7	0,7	0,7	0,7	0,7
47	0,9	0,9	0,9	0,9	0,8	0,8	1,1	1,1	1,0
48	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5

Tabela D.6: Tempo médio (em segundos) de 10 execuções da combinação de construção e tamanho de lista tabu para instâncias de 20 e 32 locações

Inst.	C1t4	C2t8	C3t5	C4t5	C5t5	C6t5	C7t3	C8t3	C9t7
49	1,1	1,2	1,3	1,3	1,1	1,1	1,1	1,1	1,2
50	1,3	1,2	1,3	1,2	1,4	1,4	1,2	1,2	1,3
51	1,4	1,5	1,3	1,2	1,3	1,3	1,3	1,4	1,5
52	0,8	0,8	0,9	0,9	0,8	0,8	0,6	0,7	0,7
53	1,4	1,5	1,5	1,5	1,3	1,3	1,3	1,4	1,4
54	1,5	1,5	1,4	1,4	1,3	1,3	1,4	1,4	1,5
55	1,5	1,5	1,3	1,3	1,3	1,3	1,5	1,7	1,4
56	1,3	1,1	1,1	1,1	0,9	0,9	1,1	1,1	1,0
57	4,4	4,0	4,3	4,2	3,7	3,7	4,2	4,3	4,5
58	3,1	3,5	3,2	3,3	3,3	3,3	2,5	2,6	3,3
59	3,8	4,2	4,1	4,0	4,1	4,1	3,3	3,3	3,7
60	2,1	2,2	1,8	1,8	2,1	2,1	2,2	2,2	2,2
61	4,7	5,4	4,5	4,5	4,2	4,2	3,7	3,6	4,7
62	4,6	4,9	4,3	4,3	4,9	4,9	3,7	3,7	4,9
63	4,4	3,9	4,1	4,1	4,2	4,2	5,0	5,1	4,4
64	3,4	2,5	2,8	2,8	2,8	2,8	2,6	2,6	2,9
65	7,9	7,0	8,6	8,5	8,6	8,7	7,5	7,4	7,2
66	6,3	6,7	6,3	6,3	5,8	5,9	6,3	6,2	6,1
67	8,5	6,6	7,0	7,2	6,2	6,2	7,3	7,2	7,6
68	3,7	3,3	4,1	4,2	3,7	3,7	4,3	4,3	4,1
69	10,0	10,4	11,6	11,6	8,9	8,8	8,5	8,6	9,3
70	8,5	8,8	8,1	8,2	8,6	8,7	7,8	7,7	7,9
71	10,0	9,8	10,1	10,0	10,1	10,1	7,5	7,4	5,8
72	5,6	4,6	7,0	6,9	6,0	6,0	7,8	7,9	5,8
73	7,6	7,3	7,1	7,1	7,0	7,2	6,2	6,4	6,9
74	9,9	9,4	8,4	8,4	8,5	8,5	8,1	10,5	6,9
75	10,8	11,7	11,8	11,7	10,9	10,9	11,6	11,3	8,7
76	7,0	9,4	7,5	7,5	7,9	8,0	7,3	7,2	5,0
77	7,7	7,8	7,7	7,7	7,6	7,6	7,8	7,9	7,8
78	11,6	12,2	10,3	10,4	10,6	11,3	9,7	9,8	10,7
79	11,3	14,0	10,1	10,1	13,3	13,3	12,3	12,6	12,0
80	7,6	8,1	8,0	7,9	7,3	7,4	5,7	5,6	5,4
81	30,2	31,8	27,7	27,9	29,6	30,0	30,2	30,1	26,9
82	29,9	26,6	26,7	26,3	25,5	25,6	23,6	23,3	28,3
83	30,6	28,0	32,7	32,6	30,1	28,3	31,0	31,1	30,5
84	16,9	18,4	20,6	20,7	15,9	16,2	11,8	12,3	12,6
85	39,5	37,3	33,8	34,0	32,3	32,1	28,0	28,4	27,2
86	30,5	32,8	30,0	30,2	27,7	28,0	23,9	23,5	25,9
87	38,6	34,7	38,2	38,2	30,3	30,5	43,3	42,2	39,2
88	20,9	20,6	23,1	23,1	25,9	26,2	19,8	19,9	20,8
89	74,7	69,8	84,8	82,2	73,6	74,2	72,8	73,4	81,1
90	47,6	43,7	54,4	53,8	61,4	60,7	43,0	42,4	48,9
91	69,2	58,9	64,8	64,6	65,0	65,2	50,6	51,2	54,9
92	33,5	27,4	31,5	31,7	31,0	31,0	38,4	38,7	34,0
93	81,5	70,2	97,7	100,6	64,8	64,1	60,5	61,0	69,6
94	83,7	60,2	58,3	58,7	74,1	74,3	51,3	51,9	58,6
95	74,9	69,1	75,7	75,6	98,9	98,1	74,6	73,9	70,0
96	44,5	33,9	47,0	47,5	37,5	37,5	47,6	48,5	47,5

Apêndice E

Resultados Computacionais dos Algoritmos ILS+Tabu Propostos

Os algoritmos híbridos ILS+BTP propostos no Capítulo 6, Seção 6.2, foram testados e os resultados obtidos podem ser vistos nas tabelas a seguir.

As Tabelas E.1 e E.2 apresentam os melhores custos obtidos em 10 execuções de cada algoritmo ILS+BTP. Em negrito encontra-se destacados o melhor valor encontrado pelos algoritmos para cada instância.

As Tabelas E.3 e E.4 apresentam os desvios percentuais entre o melhor custo conhecido e os melhores custos encontrados pelos algoritmos VNS+BTP. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo VNS+BTP é menor do que o da literatura.

As Tabelas B.5 e B.6 apresentam as médias dos tempos em segundos em 10 execuções independentes de cada algoritmo.

Tabela E.1: Melhor solução (obtida de 10 execuções) da heurística ILS+BTP para instâncias de 6 e 12 locações

Inst.	hILS1	hILS2	hILS3	hILS4	hILS5
1	16	16	16	16	16
2	26	26	26	26	26
3	18	18	18	18	18
4	26	26	26	26	26
5	16	16	16	16	16
6	27	27	27	27	27
7	16	16	16	16	16
8	31	31	31	31	31
9	25	25	25	25	25
10	46	46	46	46	46
11	32	32	32	32	32
12	43	43	43	43	43
13	28	28	28	28	28
14	45	45	45	45	45
15	35	35	35	35	35
16	49	49	49	49	49
17	35	35	35	35	35
18	62	62	62	62	62
19	46	46	46	46	46
20	63	63	63	63	62
21	48	48	48	48	48
22	67	67	67	67	67
23	56	56	56	56	56
24	74	74	74	74	74
25	31	31	31	31	31
26	43	43	43	43	43
27	43	43	43	43	43
28	55	55	55	55	55
29	29	29	29	29	29
30	49	49	49	49	49
31	42	42	42	42	42
32	69	69	69	69	69
33	52	52	52	52	52
34	72	72	72	72	72
35	73	73	73	73	73
36	95	95	95	95	95
37	48	48	48	48	48
38	83	83	83	83	83
39	69	69	69	69	69
40	108	108	108	108	108
41	78	78	78	78	78
42	102	102	102	102	102
43	110	110	110	110	110
44	140	140	140	140	140
45	66	66	66	66	66
46	116	116	116	116	116
47	115	115	115	115	115
48	171	171	171	171	171

Tabela E.2: Melhor solução (obtida de 10 execuções) da heurística ILS+BTP para instâncias de 20 e 32 locações

Inst.	hILS1	hILS2	hILS3	hILS4	hILS5
49	44	44	44	44	44
50	60	60	60	60	60
51	53	53	53	53	53
52	89	89	89	89	89
53	47	47	47	47	47
54	62	62	62	62	62
55	60	60	60	60	60
56	89	89	89	89	89
57	66	66	66	66	66
58	100	100	100	100	101
59	93	93	93	93	93
60	153	153	153	153	153
61	74	74	74	74	74
62	112	112	112	112	112
63	116	116	116	116	116
64	175	176	176	175	175
65	98	98	98	98	99
66	151	151	150	151	150
67	144	145	145	147	147
68	218	220	218	220	218
69	113	112	112	112	113
70	166	166	166	166	166
71	164	162	164	162	162
72	248	248	248	248	248
73	71	71	71	71	71
74	90	89	89	89	90
75	101	102	101	101	102
76	144	144	145	144	144
77	70	70	70	70	70
78	92	92	92	92	92
79	99	99	99	99	99
80	163	163	163	166	163
81	109	109	109	109	109
82	161	163	161	161	162
83	184	185	182	184	185
84	271	270	266	269	269
85	117	117	116	117	117
86	177	178	179	177	179
87	180	180	179	178	179
88	275	274	276	274	274
89	160	158	156	157	157
90	245	238	242	244	244
91	265	273	267	264	268
92	376	379	382	380	384
93	170	174	170	171	176
94	270	272	272	268	268
95	298	298	305	302	302
96	439	441	444	435	435

Tabela E.3: Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos ILS+BTP para instâncias de 6 e 12 locações

Inst.	hILS1	hILS2	hILS3	hILS4	hILS5
1	-	-	-	-	-
2	4,0	4,0	4,0	4,0	4,0
3	-	-	-	-	-
4	4,0	4,0	4,0	4,0	4,0
5	-	-	-	-	-
6	-	-	-	-	-
7	-	-	-	-	-
8	-	-	-	-	-
9	-	-	-	-	-
10	-	-	-	-	-
11	-	-	-	-	-
12	4,9	4,9	4,9	4,9	4,9
13	-	-	-	-	-
14	-	-	-	-	-
15	-	-	-	-	-
16	-	-	-	-	-
17	-	-	-	-	-
18	3,3	3,3	3,3	3,3	3,3
19	-	-	-	-	-
20	5,0	5,0	5,0	5,0	3,3
21	4,3	4,3	4,3	4,3	4,3
22	-	-	-	-	-
23	1,8	1,8	1,8	1,8	1,8
24	-	-	-	-	-
25	-	-	-	-	-
26	-	-	-	-	-
27	-	-	-	-	-
28	-	-	-	-	-
29	-	-	-	-	-
30	-2,0	-2,0	-2,0	-2,0	-2,0
31	-	-	-	-	-
32	-	-	-	-	-
33	-1,9	-1,9	-1,9	-1,9	-1,9
34	-	-	-	-	-
35	7,4	7,4	7,4	7,4	7,4
36	5,6	5,6	5,6	5,6	5,6
37	2,1	2,1	2,1	2,1	2,1
38	7,8	7,8	7,8	7,8	7,8
39	3,0	3,0	3,0	3,0	3,0
40	3,8	3,8	3,8	3,8	3,8
41	-	-	-	-	-
42	-1,9	-1,9	-1,9	-1,9	-1,9
43	-	-	-	-	-
44	2,2	2,2	2,2	2,2	2,2
45	-	-	-	-	-
46	4,5	4,5	4,5	4,5	4,5
47	3,6	3,6	3,6	3,6	3,6
48	1,2	1,2	1,2	1,2	1,2

Tabela E.4: Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos ILS+BTP para instâncias de 20 e 32 locações

Inst.	hILS1	hILS2	hILS3	hILS4	hILS5
49	-2,2	-2,2	-2,2	-2,2	-2,2
50	-4,8	-4,8	-4,8	-4,8	-4,8
51	-3,6	-3,6	-3,6	-3,6	-3,6
52	-9,2	-9,2	-9,2	-9,2	-9,2
53	-4,1	-4,1	-4,1	-4,1	-4,1
54	-7,5	-7,5	-7,5	-7,5	-7,5
55	-4,8	-4,8	-4,8	-4,8	-4,8
56	-8,2	-8,2	-8,2	-8,2	-8,2
57	-1,5	-1,5	-1,5	-1,5	-1,5
58	-5,7	-5,7	-5,7	-5,7	-4,7
59	-7,9	-7,9	-7,9	-7,9	-7,9
60	-3,8	-3,8	-3,8	-3,8	-3,8
61	-9,8	-9,8	-9,8	-9,8	-9,8
62	-13,2	-13,2	-13,2	-13,2	-13,2
63	-4,1	-4,1	-4,1	-4,1	-4,1
64	-7,9	-7,4	-7,4	-7,9	-7,9
65	-6,7	-6,7	-6,7	-6,7	-5,7
66	-3,2	-3,2	-3,8	-3,2	-3,8
67	-8,3	-7,6	-7,6	-6,4	-6,4
68	-6,8	-6,0	-6,8	-6,0	-6,8
69	0,9	-	-	-	0,9
70	-6,7	-6,7	-6,7	-6,7	-6,7
71	-3,5	-4,7	-3,5	-4,7	-4,7
72	-6,4	-6,4	-6,4	-6,4	-6,4
73	-4,1	-4,1	-4,1	-4,1	-4,1
74	-7,2	-8,2	-8,2	-8,2	-7,2
75	-8,2	-7,3	-8,2	-8,2	-7,3
76	-7,1	-7,1	-6,5	-7,1	-7,1
77	-4,1	-4,1	-4,1	-4,1	-4,1
78	-8,9	-8,9	-8,9	-8,9	-8,9
79	-10,0	-10,0	-10,0	-10,0	-10,0
80	-6,9	-6,9	-6,9	-5,1	-6,9
81	-8,4	-8,4	-8,4	-8,4	-8,4
82	-8,5	-7,4	-8,5	-8,5	-8,0
83	-4,2	-3,6	-5,2	-4,2	-3,6
84	-3,9	-4,3	-5,7	-4,6	-4,6
85	-6,4	-6,4	-7,2	-6,4	-6,4
86	-7,8	-7,3	-6,8	-7,8	-6,8
87	-6,7	-6,7	-7,3	-7,8	-7,3
88	-8,9	-9,3	-8,6	-9,3	-9,3
89	-6,4	-7,6	-8,8	-8,2	-8,2
90	-6,5	-9,2	-7,6	-6,9	-6,9
91	-6,7	-3,9	-6,0	-7,0	-5,6
92	-4,8	-4,1	-3,3	-3,8	-2,8
93	-10,1	-7,9	-10,1	-9,5	-6,9
94	-3,9	-3,2	-3,2	-4,6	-4,6
95	-6,3	-6,3	-4,1	-5,0	-5,0
96	-5,4	-5,0	-4,3	-6,3	-6,3

Tabela E.5: Tempo Médio das solução ILS+BTP para instâncias de 6 e 12 locações em 10 execuções de cada algoritmo

Inst.	hILS1	hILS2	hILS3	hILS4	hILS5
1	0,1	0,1	0,1	0,1	0,1
2	0,1	0,1	0,1	0,1	0,1
3	0,1	0,1	0,1	0,1	0,1
4	0,1	0,1	0,1	0,1	0,1
5	0,1	0,1	0,1	0,1	0,1
6	0,1	0,1	0,1	0,1	0,1
7	0,1	0,1	0,1	0,1	0,1
8	0,1	0,1	0,1	0,1	0,1
9	0,2	0,2	0,2	0,2	0,2
10	0,2	0,2	0,2	0,2	0,2
11	0,2	0,2	0,2	0,2	0,2
12	0,1	0,1	0,1	0,1	0,1
13	0,2	0,2	0,2	0,2	0,2
14	0,2	0,2	0,2	0,2	0,2
15	0,2	0,2	0,2	0,2	0,2
16	0,2	0,2	0,2	0,2	0,2
17	0,3	0,4	0,3	0,3	0,3
18	0,3	0,3	0,3	0,3	0,3
19	0,4	0,4	0,4	0,4	0,4
20	0,2	0,2	0,2	0,2	0,2
21	0,4	0,4	0,3	0,4	0,4
22	0,4	0,3	0,3	0,4	0,3
23	0,4	0,4	0,4	0,4	0,4
24	0,3	0,3	0,3	0,3	0,3
25	1,0	1,0	0,9	1,0	1,0
26	1,1	1,1	0,9	1,0	1,1
27	1,0	1,0	1,0	1,0	1,0
28	0,6	0,7	0,7	0,7	0,7
29	0,9	1,0	0,9	0,9	0,9
30	0,9	0,8	0,8	0,9	0,9
31	1,3	1,1	1,2	1,1	1,2
32	0,6	0,6	0,6	0,6	0,6
33	3,7	3,4	3,3	2,9	3,4
34	3,0	3,1	3,3	3,0	3,3
35	2,3	2,4	2,4	2,4	2,5
36	1,6	1,5	1,4	1,5	1,5
37	2,8	2,9	2,7	3,0	2,7
38	2,3	2,4	2,2	2,2	2,4
39	2,6	2,7	2,5	2,9	2,9
40	1,5	1,7	1,6	1,7	1,6
41	5,7	6,5	5,8	6,0	5,9
42	5,7	5,1	4,9	5,4	5,2
43	4,7	4,8	4,8	4,9	4,9
44	2,9	2,5	3,1	2,6	2,5
45	5,8	6,1	5,6	6,8	6,0
46	4,4	4,9	4,6	4,7	4,8
47	5,0	6,0	5,0	6,5	6,8
48	3,1	2,9	3,3	3,0	2,9

Tabela E.6: Tempo Médio das solução ILS+BTP para instâncias de 20 e 32 locações em 10 execuções de cada algoritmo

Inst.	hILS1	hILS2	hILS3	hILS4	hILS5
49	7,5	7,4	7,6	7,6	7,7
50	8,0	7,2	7,4	7,6	7,8
51	6,7	8,0	7,8	8,0	8,4
52	4,4	4,1	4,8	4,1	4,0
53	8,5	8,6	8,0	8,2	8,5
54	9,5	7,7	9,4	10,0	9,6
55	8,6	10,0	7,5	8,3	9,7
56	5,1	7,0	6,5	6,5	6,4
57	24,7	28,4	27,5	25,3	25,2
58	26,7	22,4	26,0	25,4	19,7
59	19,6	17,4	20,9	18,6	17,3
60	10,9	11,4	9,9	11,0	11,4
61	30,5	33,2	29,8	30,7	32,8
62	25,7	22,9	22,9	21,6	23,2
63	25,5	36,1	25,5	30,4	29,4
64	13,0	17,1	13,9	17,2	19,0
65	66,0	55,3	49,4	64,2	64,8
66	43,2	32,2	52,1	47,1	49,5
67	37,7	40,5	43,3	39,5	38,9
68	19,8	19,2	23,8	20,9	20,5
69	59,6	57,4	72,8	74,9	64,6
70	59,2	57,4	53,0	56,0	51,4
71	55,0	64,9	45,8	55,7	59,5
72	32,2	38,6	26,1	44,3	39,7
73	46,3	47,6	42,6	43,4	43,9
74	59,5	52,3	58,7	51,5	54,1
75	54,6	64,2	55,5	55,1	52,0
76	36,8	27,0	34,8	32,4	29,7
77	45,3	46,7	47,5	44,8	45,5
78	64,4	61,1	68,1	55,2	55,5
79	69,2	58,4	62,8	57,5	55,8
80	39,1	30,1	38,5	29,8	32,9
81	160,3	191,2	206,5	207,8	243,7
82	170,9	163,5	140,8	161,9	176,5
83	147,3	165,0	136,4	164,4	154,7
84	104,3	78,5	92,3	78,9	72,1
85	229,4	218,7	207,6	190,0	222,5
86	155,5	161,4	178,4	177,3	189,9
87	201,4	179,1	179,7	213,2	203,3
88	125,7	95,7	95,8	126,3	105,4
89	423,4	632,7	591,2	484,8	562,3
90	305,6	425,4	350,5	298,2	321,9
91	391,2	365,2	261,9	319,8	331,1
92	157,2	178,5	125,4	172,0	181,5
93	463,9	606,6	459,6	456,2	434,9
94	373,8	410,9	396,4	345,6	382,7
95	419,9	363,2	342,6	305,3	359,7
96	196,8	185,3	187,1	183,8	168,3

Apêndice F

Resultados Computacionais dos Algoritmos VNS+Tabu Propostos

Os algoritmos híbridos VNS+BTP propostos no Capítulo 6, Seção 6.3, foram testados e os resultados obtidos podem ser vistos nas tabelas a seguir.

As Tabelas F.1 e F.2 apresentam os melhores custos obtidos em 10 execuções de cada algoritmo VNS+BTP. Em negrito encontra-se destacados o melhor valor encontrado pelos algoritmos para cada instância.

As Tabelas F.3 e F.4 apresentam os desvios percentuais entre o melhor custo conhecido e os melhores custos encontrados pelos algoritmos VNS+BTP. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo VNS+BTP é menor do que o da literatura.

As Tabelas F.5 e F.6 apresentam as médias dos tempos em segundos em 10 execuções independentes de cada algoritmo.

Tabela F.1: Melhor solução (obtida de 10 execuções) da heurística VNS+BTP com 20 iterações VNS e 100 iterações BTP para instâncias de 6 e 12 locações

Inst.	VNS1+BTP	VNS2+BTP	VNS3+BTP	VNS4+BTP	VNS5+BTP
1	16	16	16	16	16
2	26	26	26	26	26
3	18	18	18	18	18
4	26	26	26	26	26
5	16	16	16	16	16
6	27	27	27	27	27
7	16	16	16	16	16
8	31	31	31	31	31
9	25	25	25	25	25
10	46	46	46	46	46
11	32	32	32	32	32
12	43	43	43	43	43
13	28	28	28	28	28
14	45	45	45	45	45
15	35	35	35	35	35
16	49	49	49	49	49
17	35	35	35	35	35
18	62	62	62	62	62
19	46	46	46	46	46
20	63	63	63	63	62
21	48	48	48	48	48
22	67	67	67	67	67
23	56	56	56	56	56
24	74	74	74	74	74
25	31	31	31	31	31
26	43	43	43	43	43
27	43	43	43	43	43
28	55	55	55	55	55
29	29	29	29	29	29
30	49	49	49	49	49
31	42	42	42	42	42
32	69	69	69	69	69
33	52	52	52	52	52
34	72	72	72	72	72
35	73	73	73	73	73
36	95	95	95	95	95
37	48	48	48	48	48
38	83	83	83	83	83
39	69	69	69	69	69
40	108	108	108	108	108
41	78	78	78	78	78
42	102	102	102	102	102
43	110	110	110	110	110
44	140	140	140	140	140
45	66	66	66	66	66
46	116	116	116	116	116
47	115	115	115	115	115
48	171	171	171	171	171

Tabela F.2: Melhor solução (obtida de 10 execuções) da heurística VNS+BTP com 20 iterações VNS e 100 iterações BTP para instâncias de 20 e 32 locações

Inst.	VNS1+BTP	VNS2+BTP	VNS3+BTP	VNS4+BTP	VNS5+BTP
49	44	44	44	44	44
50	60	60	60	60	60
51	53	53	53	53	53
52	89	89	88	89	89
53	47	47	47	47	47
54	62	62	62	62	62
55	60	60	60	60	60
56	89	89	89	89	89
57	66	66	66	66	66
58	100	100	100	100	100
59	93	93	93	93	93
60	153	153	153	153	153
61	74	74	74	74	74
62	112	112	112	111	112
63	116	116	116	116	116
64	175	175	175	175	175
65	98	98	98	99	98
66	151	152	151	151	151
67	144	147	144	147	145
68	218	218	218	218	219
69	112	112	112	113	111
70	166	166	166	166	166
71	162	162	162	162	162
72	248	248	248	248	248
73	71	71	71	71	71
74	90	90	89	89	89
75	101	101	101	101	101
76	144	144	144	144	144
77	70	70	70	70	70
78	92	92	92	92	92
79	99	99	99	99	99
80	163	163	163	163	163
81	109	109	109	109	109
82	161	162	161	161	161
83	181	185	183	182	182
84	267	264	264	264	264
85	117	116	117	117	117
86	179	177	179	177	178
87	177	177	178	177	178
88	275	274	274	274	274
89	158	155	157	159	158
90	240	238	239	240	240
91	261	267	262	264	262
92	381	374	374	379	376
93	174	177	174	174	176
94	267	268	270	266	268
95	296	293	295	294	295
96	435	436	438	437	435

Tabela F.3: Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos VNS+BTP para instâncias de 6 e 12 locações

Inst.	VNS1+BTP	VNS2+BTP	VNS3+BTP	VNS4+BTP	VNS5+BTP
1	-	-	-	-	-
2	3,8	3,8	3,8	3,8	3,8
3	-	-	-	-	-
4	3,8	3,8	3,8	3,8	3,8
5	-	-	-	-	-
6	-	-	-	-	-
7	-	-	-	-	-
8	-	-	-	-	-
9	-	-	-	-	-
10	-	-	-	-	-
11	-	-	-	-	-
12	4,7	4,7	4,7	4,7	4,7
13	-	-	-	-	-
14	-	-	-	-	-
15	-	-	-	-	-
16	-	-	-	-	-
17	-	-	-	-	-
18	3,2	3,2	3,2	3,2	3,2
19	-	-	-	-	-
20	4,8	4,8	4,8	4,8	3,2
21	4,2	4,2	4,2	4,2	4,2
22	-	-	-	-	-
23	1,8	1,8	1,8	1,8	1,8
24	-	-	-	-	-
25	-	-	-	-	-
26	-	-	-	-	-
27	-	-	-	-	-
28	-	-	-	-	-
29	-	-	-	-	-
30	-2,0	-2,0	-2,0	-2,0	-2,0
31	-	-	-	-	-
32	-	-	-	-	-
33	-1,9	-1,9	-1,9	-1,9	-1,9
34	-	-	-	-	-
35	6,8	6,8	6,8	6,8	6,8
36	5,3	5,3	5,3	5,3	5,3
37	2,1	2,1	2,1	2,1	2,1
38	7,2	7,2	7,2	7,2	7,2
39	2,9	2,9	2,9	2,9	2,9
40	3,7	3,7	3,7	3,7	3,7
41	-	-	-	-	-
42	-2,0	-2,0	-2,0	-2,0	-2,0
43	-	-	-	-	-
44	2,1	2,1	2,1	2,1	2,1
45	-	-	-	-	-
46	4,3	4,3	4,3	4,3	4,3
47	3,5	3,5	3,5	3,5	3,5
48	1,2	1,2	1,2	1,2	1,2

Tabela F.4: Desvio percentual entre a melhor solução conhecida e melhor solução dos algoritmos VNS+BTP para instâncias de 20 e 32 locações

Inst.	VNS1+BTP	VNS2+BTP	VNS3+BTP	VNS4+BTP	VNS5+BTP
49	-2,3	-2,3	-2,3	-2,3	-2,3
50	-5,0	-5,0	-5,0	-5,0	-5,0
51	-3,8	-3,8	-3,8	-3,8	-3,8
52	-10,1	-10,1	-11,4	-10,1	-10,1
53	-4,3	-4,3	-4,3	-4,3	-4,3
54	-8,1	-8,1	-8,1	-8,1	-8,1
55	-5,0	-5,0	-5,0	-5,0	-5,0
56	-9,0	-9,0	-9,0	-9,0	-9,0
57	-1,5	-1,5	-1,5	-1,5	-1,5
58	-6,0	-6,0	-6,0	-6,0	-6,0
59	-8,6	-8,6	-8,6	-8,6	-8,6
60	-3,9	-3,9	-3,9	-3,9	-3,9
61	-10,8	-10,8	-10,8	-10,8	-10,8
62	-15,2	-15,2	-15,2	-16,2	-15,2
63	-4,3	-4,3	-4,3	-4,3	-4,3
64	-8,6	-8,6	-8,6	-8,6	-8,6
65	-7,1	-7,1	-7,1	-6,1	-7,1
66	-3,3	-2,6	-3,3	-3,3	-3,3
67	-9,0	-6,8	-9,0	-6,8	-8,3
68	-7,3	-7,3	-7,3	-7,3	-6,8
69	-	-	-	0,9	-0,9
70	-7,2	-7,2	-7,2	-7,2	-7,2
71	-4,9	-4,9	-4,9	-4,9	-4,9
72	-6,9	-6,9	-6,9	-6,9	-6,9
73	-4,2	-4,2	-4,2	-4,2	-4,2
74	-7,8	-7,8	-9,0	-9,0	-9,0
75	-8,9	-8,9	-8,9	-8,9	-8,9
76	-7,6	-7,6	-7,6	-7,6	-7,6
77	-4,3	-4,3	-4,3	-4,3	-4,3
78	-9,8	-9,8	-9,8	-9,8	-9,8
79	-11,1	-11,1	-11,1	-11,1	-11,1
80	-7,4	-7,4	-7,4	-7,4	-7,4
81	-9,2	-9,2	-9,2	-9,2	-9,2
82	-9,3	-8,6	-9,3	-9,3	-9,3
83	-6,1	-3,8	-4,9	-5,5	-5,5
84	-5,6	-6,8	-6,8	-6,8	-6,8
85	-6,8	-7,8	-6,8	-6,8	-6,8
86	-7,3	-8,5	-7,3	-8,5	-7,9
87	-9,0	-9,0	-8,4	-9,0	-8,4
88	-9,8	-10,2	-10,2	-10,2	-10,2
89	-8,2	-10,3	-8,9	-7,5	-8,2
90	-9,2	-10,1	-9,6	-9,2	-9,2
91	-8,8	-6,4	-8,4	-7,6	-8,4
92	-3,7	-5,6	-5,6	-4,2	-5,1
93	-8,6	-6,8	-8,6	-8,6	-7,4
94	-5,2	-4,9	-4,1	-5,6	-4,9
95	-7,4	-8,5	-7,8	-8,2	-7,8
96	-6,7	-6,4	-5,9	-6,2	-6,7

Tabela F.5: Tempo Médio das solução VNS+BTP para instâncias de 6 e 12 locações em 10 execuções de cada algoritmo

Inst.	VNS1+BTP	VNS2+BTP	VNS3+BTP	VNS4+BTP	VNS5+BTP
1	0,8	0,8	0,8	0,9	0,8
2	0,7	0,7	0,7	0,7	0,7
3	0,8	0,8	1,3	0,9	0,8
4	0,7	0,6	0,9	0,6	0,6
5	0,8	0,8	0,9	0,8	0,7
6	0,7	0,7	1,1	0,7	0,7
7	0,9	0,9	1,9	0,9	0,9
8	0,7	0,8	1,1	0,8	0,7
9	1,7	1,6	2,5	1,7	1,6
10	1,5	1,5	2,7	1,5	1,4
11	1,9	1,9	3,2	2,0	1,9
12	1,4	1,4	2,3	1,4	1,3
13	1,6	1,6	3,4	1,6	1,6
14	1,4	1,6	2,0	1,5	1,4
15	2,1	2,1	4,4	2,2	2,1
16	1,7	1,7	2,6	1,8	1,7
17	3,1	3,1	4,5	3,2	3,0
18	2,7	2,8	5,0	2,9	2,6
19	3,2	3,2	4,8	3,2	3,1
20	2,3	2,3	5,0	2,4	2,1
21	3,0	3,0	4,3	3,1	2,9
22	2,7	2,6	4,7	2,7	2,6
23	3,6	3,7	5,5	3,9	3,6
24	3,0	2,9	4,4	3,0	2,8
25	8,8	8,7	14,6	9,0	9,1
26	7,7	7,7	13,9	8,0	8,1
27	9,0	9,3	15,2	9,8	9,8
28	6,2	6,3	9,7	6,6	6,3
29	8,6	8,6	15,8	8,9	9,1
30	7,4	7,2	11,8	7,5	7,6
31	9,4	9,2	15,0	9,7	9,7
32	6,1	5,9	9,0	6,2	6,0
33	21,2	20,8	35,4	21,2	21,8
34	17,2	17,4	28,2	18,1	17,9
35	19,2	19,0	32,2	20,1	19,8
36	13,0	12,9	19,9	13,3	13,0
37	19,1	19,3	28,9	20,1	20,5
38	16,0	16,1	25,0	16,7	16,7
39	20,5	20,4	33,8	21,6	21,2
40	13,9	14,1	21,2	15,0	14,4
41	34,5	35,2	59,0	36,2	36,8
42	30,1	30,6	46,2	31,6	31,1
43	34,8	35,0	60,2	36,4	37,1
44	22,9	22,6	37,7	23,6	22,3
45	33,5	33,6	52,1	35,6	36,2
46	29,1	29,0	50,2	30,0	29,9
47	37,8	38,1	56,7	40,9	38,8
48	25,8	25,6	42,7	26,1	25,4

Tabela F.6: Tempo Médio das solução VNS+BTP para instâncias de 20 e 32 locações em 10 execuções de cada algoritmo

Inst.	VNS1+BTP	VNS2+BTP	VNS3+BTP	VNS4+BTP	VNS5+BTP
49	63,1	63,9	104,1	65,5	71,9
50	50,4	50,2	76,2	53,9	54,3
51	58,6	59,1	95,9	61,9	61,6
52	36,0	35,6	57,7	37,8	35,8
53	68,4	68,4	103,5	69,4	75,5
54	60,0	58,0	96,3	60,8	65,4
55	62,6	59,7	95,7	61,7	64,9
56	41,2	41,4	65,3	44,3	46,1
57	146,3	147,7	227,3	141,6	158,6
58	117,9	122,4	178,8	118,5	123,5
59	124,7	122,8	195,4	122,3	129,8
60	76,9	78,9	123,4	77,7	78,2
61	163,1	161,4	248,9	165,9	176,4
62	144,9	141,2	223,2	141,9	151,5
63	148,1	148,2	223,5	151,7	162,1
64	103,6	100,2	157,0	104,9	106,2
65	258,1	253,1	399,6	257,7	288,7
66	203,1	197,2	320,6	211,3	222,1
67	235,6	229,1	355,1	230,2	250,0
68	131,8	131,7	202,5	132,4	134,9
69	293,1	280,2	468,1	293,7	319,1
70	244,7	245,2	375,9	246,8	265,3
71	286,0	278,3	435,1	287,7	300,9
72	178,6	182,4	278,3	188,1	187,7
73	350,6	364,7	570,6	356,5	396,2
74	294,2	283,0	616,3	294,3	331,3
75	340,0	329,7	807,6	357,0	358,7
76	213,5	209,7	700,3	208,6	212,1
77	361,0	382,6	998,6	376,4	401,7
78	338,1	353,3	882,1	334,2	362,0
79	356,9	354,9	1262,2	365,1	390,3
80	256,1	258,8	878,2	258,6	281,4
81	835,6	825,9	2033,8	845,5	940,4
82	676,4	660,2	684,8	680,2	746,9
83	815,3	748,2	795,9	756,2	838,8
84	477,3	451,7	467,0	474,9	492,7
85	866,8	893,2	909,0	940,1	976,3
86	768,7	791,7	787,5	750,5	868,2
87	860,3	884,3	862,4	855,8	995,1
88	527,5	541,1	565,1	537,8	564,2
89	1571,8	1599,1	1580,9	1750,1	1879,8
90	1381,4	1363,4	1247,4	1313,2	1359,5
91	1431,3	1537,7	1374,3	1536,9	1494,1
92	762,0	770,9	773,4	759,7	816,0
93	1763,8	1871,8	1801,9	1747,1	2024,4
94	1473,9	1481,8	1432,5	1616,9	1540,3
95	1783,1	1619,8	1795,8	1712,4	1697,6
96	924,1	919,8	946,3	924,2	944,2

Apêndice G

Resultados Computacionais dos Algoritmos GRASP+Tabu Propostos

Os algoritmos híbridos GRASP+BTP propostos no Capítulo 6, Seção 6.4, foram testados e os resultados obtidos podem ser vistos nas tabelas a seguir.

As Tabelas G.1 e G.2 apresentam os melhores custos obtidos em 10 execuções de cada algoritmo GRASP+BTP. Em negrito encontra-se destacados o melhor valor encontrado pelos algoritmos para cada instância.

As Tabelas G.3 e G.4 apresentam as diferenças percentuais entre o melhor custo conhecido e os melhores custos encontrados pelos algoritmos GRASP+BTP. Nas tabelas o símbolo (-) indica que os custos são os mesmos, ou seja, que não há diferença e valores negativos indicam que o custo obtido pelo algoritmo GRASP+BTP é menor do que o da literatura.

As Tabelas G.5 e G.6 apresentam as médias dos tempos em segundos em 10 execuções independentes de cada algoritmo GRASP+BTP.

Tabela G.1: Melhor solução (obtida de 10 execuções) do GRASP+BTP para instâncias de 6 e 12 locações

Inst.	G1+BTP	G2+BTP	G3+BTP	G4+BTP	G5+BTP	G6+BTP	G7+BTP	G8+BTP	G9+BTP	G10+BTP
1	16	16	16	16	16	16	16	16	16	16
2	25	26	25	26	26	25	26	25	26	26
3	18	18	18	18	18	18	18	18	18	18
4	25	25	25	25	25	25	25	25	25	25
5	16	16	16	16	16	16	16	16	16	16
6	27	27	27	27	27	27	27	27	27	27
7	16	16	16	16	16	16	16	16	16	16
8	31	31	31	31	31	31	31	31	31	31
9	25	25	25	25	25	25	25	25	25	25
10	46	46	46	46	46	46	46	46	46	46
11	32	32	32	32	32	32	32	32	32	32
12	41	41	41	41	41	41	41	41	42	42
13	28	28	28	28	28	28	28	28	28	28
14	45	45	45	45	45	45	45	45	45	45
15	35	35	35	35	35	35	35	35	35	35
16	49	49	49	49	49	49	49	49	49	49
17	35	35	35	35	35	35	35	35	35	35
18	62	62	62	62	62	62	62	62	62	62
19	46	46	46	46	46	46	46	46	46	46
20	60	63	60	61	60	60	61	60	61	61
21	47	48	48	48	48	46	48	48	48	48
22	67	67	67	67	67	67	67	67	67	67
23	55	56	55	56	56	55	56	55	56	56
24	74	74	74	74	74	74	74	74	74	74
25	31	31	31	31	31	31	31	31	31	31
26	43	43	43	43	43	43	43	43	43	43
27	43	43	43	43	43	43	43	43	43	43
28	55	55	55	55	55	55	55	55	55	55
29	29	29	29	29	29	29	29	29	29	29
30	49	49	49	49	49	49	49	49	49	49
31	42	42	42	42	42	42	42	42	42	42
32	69	69	69	69	69	69	69	69	69	69
33	52	52	52	52	52	52	52	52	52	52
34	72	72	72	72	72	72	72	72	72	72
35	73	73	73	73	73	73	73	73	73	73
36	95	95	95	95	95	95	95	95	95	95
37	48	48	48	48	48	48	48	48	48	48
38	83	83	83	83	83	83	83	83	83	83
39	69	69	69	69	69	69	69	69	69	69
40	108	108	108	108	108	108	108	108	108	108
41	78	78	78	78	78	78	78	78	78	78
42	102	102	102	102	102	102	102	102	102	102
43	110	110	110	110	110	110	110	110	110	110
44	140	140	140	140	140	140	140	140	140	140
45	66	66	66	66	66	66	66	66	66	66
46	116	116	116	116	116	116	116	116	116	116
47	115	115	115	115	115	115	115	115	115	115
48	171	171	171	171	171	171	171	171	171	171

Tabela G.2: Melhor solução (obtida de 10 execuções) do GRASP+BTP para instâncias de 20 e 32 locações

Inst.	G1+BTP	G2+BTP	G3+BTP	G4+BTP	G5+BTP	G6+BTP	G7+BTP	G8+BTP	G9+BTP	G10+BTP
49	44	44	44	44	44	44	44	44	44	44
50	60	60	60	60	60	60	60	60	60	60
51	53	53	53	53	53	53	53	53	53	53
52	89	89	89	89	89	89	89	89	89	89
53	47	47	47	47	47	47	47	47	47	47
54	62	62	62	62	62	62	62	62	62	62
55	60	60	60	60	60	60	60	60	60	60
56	89	89	89	89	89	89	89	89	89	89
57	66	66	66	66	66	66	66	66	66	66
58	100	100	100	100	100	100	100	100	100	100
59	93	93	93	93	93	93	93	93	93	93
60	153	153	153	153	153	153	153	153	153	153
61	74	74	74	74	74	74	74	74	74	74
62	112	112	112	112	112	112	112	112	112	112
63	116	116	116	116	116	116	116	116	116	116
64	175	175	175	175	175	174	175	175	175	175
65	98	98	98	98	98	99	98	98	99	99
66	150	151	151	151	150	150	151	151	150	150
67	145	145	145	145	144	145	144	145	144	144
68	218	218	218	218	218	218	218	218	218	218
69	113	113	113	112	111	113	113	113	112	112
70	166	166	166	166	166	166	166	166	166	166
71	162	162	163	162	162	162	162	162	164	164
72	248	248	248	248	248	248	248	248	248	248
73	71	71	71	71	71	71	71	71	71	71
74	90	90	89	90	89	90	90	90	89	89
75	101	101	101	101	101	101	101	101	101	101
76	144	144	144	144	144	144	144	144	144	144
77	70	70	70	70	70	70	70	70	70	70
78	92	92	92	92	92	92	92	92	92	92
79	99	99	99	99	99	99	99	99	99	99
80	163	163	163	163	163	163	163	163	163	163
81	109	110	109	110	110	110	109	109	109	109
82	162	164	162	162	161	161	163	163	162	162
83	183	182	181	181	184	183	184	183	183	183
84	266	264	266	266	266	266	266	266	264	264
85	118	118	118	118	117	117	118	117	117	117
86	179	182	180	180	180	178	178	177	179	179
87	180	178	180	181	182	179	179	181	181	181
88	275	275	276	274	275	275	275	276	276	276
89	161	159	160	162	160	158	162	159	162	162
90	246	245	244	247	246	246	246	245	245	245
91	267	267	264	268	264	262	266	269	264	264
92	380	374	375	376	381	379	375	371	379	379
93	177	175	179	177	177	177	178	176	176	176
94	273	270	271	268	266	268	272	271	271	271
95	300	303	297	303	303	300	301	301	301	301
96	439	444	440	441	441	438	444	441	441	441

Tabela G.3: Desvio percentual entre a melhor solução conhecida e melhor solução de GRASP+BTP para instâncias de 6 e 12 locações

Inst.	G1+BTP	G2+BTP	G3+BTP	G4+BTP	G5+BTP	G6+BTP	G7+BTP	G8+BTP	G9+BTP	G10+BTP
1	-	-	-	-	-	-	-	-	-	-
2	-	3,8	-	3,8	3,8	-	3,8	-	3,8	3,8
3	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	2,4	2,4
13	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-
18	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2
19	-	-	-	-	-	-	-	-	-	-
20	-	4,8	-	1,6	-	-	1,6	-	1,6	1,6
21	2,1	4,2	4,2	4,2	4,2	-	4,2	4,2	4,2	4,2
22	-	-	-	-	-	-	-	-	-	-
23	-	1,8	-	1,8	1,8	-	1,8	-	1,8	1,8
24	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-
30	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0
31	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-
33	-1,9	-1,9	-1,9	-1,9	-1,9	-1,9	-1,9	-1,9	-1,9	-1,9
34	-	-	-	-	-	-	-	-	-	-
35	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8	6,8
36	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3
37	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1
38	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2
39	2,9	2,9	2,9	2,9	2,9	2,9	2,9	2,9	2,9	2,9
40	3,7	3,7	3,7	3,7	3,7	3,7	3,7	3,7	3,7	3,7
41	-	-	-	-	-	-	-	-	-	-
42	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0	-2,0
43	-	-	-	-	-	-	-	-	-	-
44	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1	2,1
45	-	-	-	-	-	-	-	-	-	-
46	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3
47	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5	3,5
48	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,2

Tabela G.4: Desvio percentual entre a melhor solução conhecida e melhor solução de GRASP+BTP para instâncias de 20 e 32 locações

Inst.	G1+BTP	G2+BTP	G3+BTP	G4+BTP	G5+BTP	G6+BTP	G7+BTP	G8+BTP	G9+BTP	G10+BTP
49	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3	-2,3
50	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0
51	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8	-3,8
52	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1	-10,1
53	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
54	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1	-8,1
55	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0	-5,0
56	-9,0	-9,0	-9,0	-9,0	-9,0	-9,0	-9,0	-9,0	-9,0	-9,0
57	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5	-1,5
58	-6,0	-6,0	-6,0	-6,0	-6,0	-6,0	-6,0	-6,0	-6,0	-6,0
59	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6	-8,6
60	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9	-3,9
61	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8	-10,8
62	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2	-15,2
63	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
64	-8,6	-8,6	-8,6	-8,6	-8,6	-9,2	-8,6	-8,6	-8,6	-8,6
65	-7,1	-7,1	-7,1	-7,1	-7,1	-6,1	-7,1	-7,1	-6,1	-6,1
66	-4,0	-3,3	-3,3	-3,3	-4,0	-4,0	-3,3	-3,3	-4,0	-4,0
67	-8,3	-8,3	-8,3	-8,3	-9,0	-8,3	-9,0	-8,3	-9,0	-9,0
68	-7,3	-7,3	-7,3	-7,3	-7,3	-7,3	-7,3	-7,3	-7,3	-7,3
69	0,9	0,9	0,9	-	-0,9	0,9	0,9	0,9	-	-
70	-7,2	-7,2	-7,2	-7,2	-7,2	-7,2	-7,2	-7,2	-7,2	-7,2
71	-4,9	-4,9	-4,3	-4,9	-4,9	-4,9	-4,9	-4,9	-3,7	-3,7
72	-6,9	-6,9	-6,9	-6,9	-6,9	-6,9	-6,9	-6,9	-6,9	-6,9
73	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2	-4,2
74	-7,8	-7,8	-9,0	-7,8	-9,0	-7,8	-7,8	-7,8	-9,0	-9,0
75	-8,9	-8,9	-8,9	-8,9	-8,9	-8,9	-8,9	-8,9	-8,9	-8,9
76	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6	-7,6
77	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3	-4,3
78	-9,8	-9,8	-9,8	-9,8	-9,8	-9,8	-9,8	-9,8	-9,8	-9,8
79	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1	-11,1
80	-7,4	-7,4	-7,4	-7,4	-7,4	-7,4	-7,4	-7,4	-7,4	-7,4
81	-9,2	-8,2	-9,2	-8,2	-8,2	-8,2	-9,2	-9,2	-9,2	-9,2
82	-8,6	-7,3	-8,6	-8,6	-9,3	-9,3	-8,0	-8,0	-8,6	-8,6
83	-4,9	-5,5	-6,1	-6,1	-4,3	-4,9	-4,3	-4,9	-4,9	-4,9
84	-6,0	-6,8	-6,0	-6,0	-6,0	-6,0	-6,0	-6,0	-6,8	-6,8
85	-5,9	-5,9	-5,9	-5,9	-6,8	-6,8	-5,9	-6,8	-6,8	-6,8
86	-7,3	-5,5	-6,7	-6,7	-6,7	-7,9	-7,9	-8,5	-7,3	-7,3
87	-7,2	-8,4	-7,2	-6,6	-6,0	-7,8	-7,8	-6,6	-6,6	-6,6
88	-9,8	-9,8	-9,4	-10,2	-9,8	-9,8	-9,8	-9,4	-9,4	-9,4
89	-6,2	-7,5	-6,9	-5,6	-6,9	-8,2	-5,6	-7,5	-5,6	-5,6
90	-6,5	-6,9	-7,4	-6,1	-6,5	-6,5	-6,5	-6,9	-6,9	-6,9
91	-6,4	-6,4	-7,6	-6,0	-7,6	-8,4	-6,8	-5,6	-7,6	-7,6
92	-3,9	-5,6	-5,3	-5,1	-3,7	-4,2	-5,3	-6,5	-4,2	-4,2
93	-6,8	-8,0	-5,6	-6,8	-6,8	-6,8	-6,2	-7,4	-7,4	-7,4
94	-2,9	-4,1	-3,7	-4,9	-5,6	-4,9	-3,3	-3,7	-3,7	-3,7
95	-6,0	-5,0	-7,1	-5,0	-5,0	-6,0	-5,6	-5,6	-5,6	-5,6
96	-5,7	-4,5	-5,5	-5,2	-5,2	-5,9	-4,5	-5,2	-5,2	-5,2

Tabela G.5: Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP+BTP para instâncias de 6 e 12 locações

Inst.	G1+BTP	G2+BTP	G3+BTP	G4+BTP	G5+BTP	G6+BTP	G7+BTP	G8+BTP	G9+BTP	G10+BTP
1	0,1	0,1	0,1	0,1	0,2	0,2	0,1	0,2	0,1	0,1
2	0,1	0,1	0,1	0,1	0,2	0,2	0,1	0,2	0,1	0,1
3	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,1	0,1
4	0,1	0,1	0,1	0,1	0,1	0,2	0,1	0,2	0,1	0,1
5	0,1	0,1	0,1	0,1	0,2	0,2	0,1	0,2	0,1	0,1
6	0,1	0,1	0,1	0,1	0,2	0,2	0,1	0,2	0,1	0,1
7	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
8	0,1	0,1	0,1	0,1	0,2	0,2	0,1	0,2	0,1	0,1
9	0,3	0,3	0,3	0,3	0,5	0,4	0,3	0,4	0,3	0,3
10	0,3	0,3	0,3	0,3	0,4	0,4	0,3	0,4	0,3	0,3
11	0,3	0,3	0,3	0,4	0,5	0,5	0,3	0,5	0,3	0,3
12	0,2	0,2	0,2	0,2	0,3	0,4	0,2	0,4	0,2	0,2
13	0,3	0,3	0,3	0,3	0,5	0,4	0,3	0,5	0,3	0,3
14	0,3	0,3	0,3	0,3	0,4	0,4	0,3	0,4	0,3	0,3
15	0,4	0,4	0,4	0,4	0,6	0,6	0,4	0,6	0,4	0,4
16	0,3	0,3	0,3	0,3	0,4	0,5	0,3	0,5	0,3	0,3
17	0,6	0,6	0,6	0,6	1,0	0,9	0,6	0,9	0,6	0,6
18	0,5	0,5	0,5	0,5	0,8	0,8	0,5	0,5	0,5	0,5
19	0,6	0,6	0,6	0,6	1,0	0,9	0,6	0,6	0,6	0,6
20	0,4	0,4	0,4	0,4	0,6	0,6	0,4	0,5	0,4	0,4
21	0,6	0,6	0,6	0,6	1,0	1,0	0,6	0,6	0,6	0,6
22	0,5	0,5	0,5	0,6	0,9	0,8	0,5	0,6	0,5	0,5
23	0,6	0,7	0,7	0,7	1,1	1,2	0,6	0,7	0,7	0,7
24	0,5	0,5	0,5	0,5	0,7	0,8	0,5	0,5	0,5	0,5
25	1,8	1,8	1,8	1,9	3,4	3,2	1,9	1,7	1,8	1,9
26	1,9	1,9	1,9	1,9	3,4	3,5	1,9	1,8	1,9	2,0
27	1,7	1,6	1,7	1,7	3,3	2,4	1,7	1,7	1,7	1,7
28	1,2	1,1	1,2	1,2	1,9	1,8	1,1	1,2	1,2	1,2
29	1,6	1,6	1,6	1,6	3,2	3,1	1,6	1,5	1,6	1,7
30	1,6	1,6	1,6	1,6	3,0	3,1	1,7	1,6	1,6	1,7
31	2,2	2,2	2,2	2,2	3,7	4,2	2,2	2,2	2,3	2,3
32	1,1	1,1	1,2	1,2	1,7	2,1	1,1	1,2	1,2	1,2
33	4,9	5,2	5,0	5,1	9,8	7,1	5,1	4,8	5,2	5,2
34	4,4	4,5	4,3	4,3	8,6	14,8	4,5	4,1	4,3	4,4
35	4,7	4,8	4,8	4,8	9,4	16,4	4,8	4,6	4,9	4,9
36	2,9	3,0	3,1	3,1	4,9	11,1	3,0	3,1	3,1	3,2
37	5,1	5,2	5,0	5,1	9,7	7,3	5,1	4,9	5,0	5,1
38	4,2	4,5	4,0	4,3	8,3	6,1	4,3	4,2	4,4	4,3
39	4,7	4,9	5,0	5,1	9,1	11,1	4,7	4,9	4,9	4,9
40	3,0	3,1	3,0	3,2	4,9	4,5	3,0	3,2	3,1	3,1
41	9,2	10,1	9,8	10,1	20,6	14,5	9,9	9,6	10,1	10,2
42	8,7	8,8	8,7	8,6	18,9	29,8	8,9	8,6	8,9	9,0
43	9,2	9,3	9,5	9,6	19,8	13,9	9,7	9,4	9,8	9,9
44	5,9	5,8	6,0	6,1	10,3	9,6	5,6	6,1	5,9	5,9
45	9,8	10,4	10,1	10,2	20,9	16,5	10,1	9,5	10,2	10,2
46	8,0	8,7	7,9	8,7	17,8	18,8	8,4	8,2	8,6	8,6
47	9,7	9,6	9,5	10,1	19,9	14,8	9,8	9,7	10,4	10,5
48	6,1	6,2	5,9	6,1	10,0	20,0	6,0	6,2	6,1	6,1

Tabela G.6: Tempo médio (em segundos) de 10 execuções dos algoritmos GRASP+BTP para instâncias de 20 e 32 locações

Inst.	G1+BTP	G2+BTP	G3+BTP	G4+BTP	G5+BTP	G6+BTP	G7+BTP	G8+BTP	G9+BTP	G10+BTP
49	13,9	15,8	13,2	15,5	29,3	43,6	14,7	12,9	14,7	14,7
50	15,5	17,1	15,2	17,0	28,0	51,0	16,1	14,2	15,6	15,6
51	16,3	17,2	16,4	17,6	28,2	53,5	16,6	16,0	17,4	17,3
52	9,7	9,9	9,7	10,4	14,3	20,7	9,7	9,7	10,2	10,2
53	16,6	20,3	16,4	20,5	33,1	23,6	18,5	15,0	17,5	17,7
54	19,0	22,8	19,6	22,8	33,5	41,3	21,6	18,6	19,9	19,8
55	16,7	17,7	16,5	18,0	31,6	35,0	16,9	16,4	17,5	17,6
56	12,6	12,7	12,5	13,6	20,1	19,3	12,4	12,3	12,7	12,7
57	45,0	52,1	46,3	52,7	92,3	73,9	49,2	42,0	48,9	49,3
58	37,3	42,4	35,5	42,7	79,6	80,5	41,4	35,1	37,9	37,9
59	43,3	43,5	41,9	47,1	80,6	65,8	43,8	42,4	43,8	44,1
60	24,3	24,8	24,7	25,6	37,7	82,5	24,6	24,9	24,0	24,0
61	51,7	60,3	50,6	64,6	104,1	166,7	55,8	47,0	56,1	56,6
62	50,4	65,6	51,0	68,0	97,8	173,0	55,9	48,3	56,9	56,5
63	52,0	56,3	50,9	57,0	96,0	98,2	53,3	49,2	51,9	52,1
64	34,0	35,3	34,2	36,6	57,9	75,7	34,9	32,1	34,2	34,4
65	92,5	111,2	98,1	111,1	206,7	169,4	98,6	88,8	98,7	98,3
66	76,0	84,3	69,8	86,6	169,3	125,3	76,3	65,5	79,1	78,6
67	78,3	83,3	76,3	82,4	172,1	141,0	79,2	75,0	80,1	80,2
68	42,1	44,3	45,6	46,1	76,0	115,0	42,8	43,8	43,3	43,2
69	111,4	129,7	108,3	134,4	232,0	174,1	122,6	98,0	122,3	122,7
70	97,6	116,4	93,2	117,0	202,3	303,2	104,5	89,4	105,1	104,8
71	102,7	111,7	103,9	112,2	208,4	262,9	104,7	99,3	107,0	107,7
72	66,0	68,6	68,9	70,8	121,3	115,0	67,8	65,1	67,7	67,7
73	85,9	99,1	89,1	99,8	204,2	126,3	91,3	75,9	89,1	89,1
74	111,1	114,5	102,1	116,0	204,6	280,2	111,2	92,0	107,0	107,9
75	119,1	127,0	125,7	128,7	225,2	234,4	126,9	112,1	124,6	125,2
76	81,6	79,9	79,3	84,2	121,0	129,1	78,4	79,0	79,4	79,2
77	101,2	118,2	99,3	117,0	212,0	167,7	107,7	90,1	108,0	108,3
78	124,6	140,6	127,1	144,5	229,9	252,1	134,5	117,8	127,7	127,3
79	142,8	149,0	144,6	153,9	247,9	258,7	146,1	143,2	144,9	145,7
80	92,3	100,7	97,6	95,5	140,4	169,3	89,0	88,0	97,5	95,6
81	337,1	399,8	334,9	395,1	690,3	502,5	377,2	326,3	353,5	357,1
82	296,5	312,1	281,2	308,2	571,0	467,6	300,0	262,1	288,2	286,8
83	353,3	373,2	354,5	387,4	663,4	595,6	361,8	342,9	366,1	368,2
84	204,2	226,5	220,0	214,3	336,8	719,8	210,5	205,1	214,4	214,5
85	381,6	415,9	376,3	423,4	705,0	562,0	390,2	348,6	403,4	404,0
86	345,1	404,3	343,3	391,6	665,7	684,3	365,5	312,2	376,5	381,5
87	398,7	423,9	388,9	416,4	713,3	704,1	397,5	368,2	400,8	404,6
88	236,5	240,4	242,6	244,9	390,9	227,5	224,1	237,3	245,6	243,9
89	711,5	814,3	744,9	856,0	1520,4	646,6	801,4	688,7	773,1	776,9
90	553,8	620,5	573,3	603,7	1238,7	528,7	586,5	521,2	575,7	579,9
91	714,5	722,9	690,1	782,1	1357,3	627,1	723,9	640,5	713,8	719,1
92	372,6	368,3	371,1	392,3	635,4	349,4	359,9	345,6	350,7	353,3
93	818,9	927,3	789,6	893,4	1649,4	729,6	849,4	720,0	845,9	849,7
94	658,5	801,2	679,0	799,8	1505,8	628,9	701,4	593,7	711,0	719,4
95	825,2	885,7	810,4	871,9	1549,7	713,9	836,2	737,8	854,6	847,9
96	472,9	475,1	468,4	489,5	791,2	437,6	456,8	438,4	452,7	452,3

Apêndice H

Comparação de Desvios Percentuais entre Algoritmos da Literatura e Propostos

Neste apêndice são apresentadas os desvios percentuais de custos do melhor algoritmo da literatura e os algoritmos propostos, discutidos no Capítulo 7, Seção 7.1.

Tabela H.1: Desvio percentual entre custo da melhor solução da literatura e dos algoritmos propostos

Inst.	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
33	7,5	3,8	-1,9	-1,9	-1,9	-1,9	-1,9
34	1,4	2,8	-	-	-	-	-
37	6,4	6,4	2,1	2,1	2,1	2,1	2,1
38	6,5	6,5	7,8	7,8	7,8	7,8	7,8
41	2,6	-	-	-	-	-	-
42	1,9	-	-1,9	-1,9	-1,9	-1,9	-1,9
43	1,8	-	-	-	-	-	-
44	4,4	2,2	2,2	2,2	2,2	2,2	2,2
45	3,0	7,6	-	-	-	-	-
46	4,5	3,6	4,5	4,5	4,5	4,5	4,5
47	6,3	4,5	3,6	3,6	3,6	3,6	3,6
48	1,2	1,2	1,2	1,2	1,2	1,2	1,2

Tabela H.2: Desvio percentual entre custo da melhor solução da literatura e dos algoritmos propostos

Inst.	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
49	-2,2	-2,2	-2,2	-2,2	-2,2	-2,2	-2,2
50	-4,8	-6,3	-4,8	-4,8	-4,8	-4,8	-4,8
51	-	1,8	-3,6	-3,6	-3,6	-3,6	-3,6
52	-9,2	-9,2	-9,2	-9,2	-9,2	-10,2	-9,2
53	-4,1	-4,1	-4,1	-4,1	-4,1	-4,1	-4,1
54	-6,0	-6,0	-7,5	-7,5	-7,5	-7,5	-7,5
55	-4,8	-	-4,8	-4,8	-4,8	-4,8	-4,8
56	-7,2	-8,2	-8,2	-8,2	-8,2	-8,2	-8,2
57	-	3,0	-1,5	-1,5	-1,5	-1,5	-1,5
58	-3,8	-9,4	-5,7	-5,7	-5,7	-5,7	-5,7
59	-5,0	-4,0	-6,9	-7,9	-7,9	-7,9	-7,9
60	-3,1	-2,5	-3,8	-3,8	-3,8	-3,8	-3,8
61	-8,5	-8,5	-9,8	-9,8	-9,8	-9,8	-9,8
62	-9,3	-9,3	-12,4	-13,2	-13,2	-13,2	-13,2
63	-	-0,8	-1,7	-4,1	-4,1	-4,1	-4,1
64	-5,3	-7,4	-7,4	-7,9	-7,9	-7,9	-8,4
65	-	-3,8	-3,8	-3,8	-6,7	-6,7	-5,7
66	-2,6	-3,8	-1,9	-3,2	-3,2	-3,2	-3,8
67	-1,3	-4,5	-6,4	-6,4	-6,4	-8,3	-7,6
68	-3,0	-4,3	-6,8	-6,0	-6,0	-6,8	-6,8
69	6,3	5,4	3,6	-	-	-	0,9
70	-1,1	-2,8	-5,1	-6,7	-6,7	-6,7	-6,7
71	1,8	-0,6	-1,8	-2,9	-4,7	-4,7	-4,7
72	-4,9	-6,8	-6,4	-6,4	-6,4	-6,4	-6,4

Tabela H.3: Desvio percentual entre custo da melhor solução da literatura e dos algoritmos propostos

Inst.	VNS	ILS	GRASP	BTP	hILS	hVNS	hGRASP
73	-4,1	-4,1	-4,1	-4,1	-4,1	-4,1	-4,1
74	-5,2	-1,0	-6,2	-7,2	-8,2	-8,2	-7,2
75	-4,5	-0,9	-5,5	-8,2	-8,2	-8,2	-8,2
76	-	-3,2	-6,5	-7,1	-7,1	-7,1	-7,1
77	-4,1	-4,1	-4,1	-4,1	-4,1	-4,1	-4,1
78	-4,0	-3,0	-7,9	-8,9	-8,9	-8,9	-8,9
79	-0,9	-	-9,1	-10,0	-10,0	-10,0	-10,0
80	-2,3	-6,3	-6,3	-5,7	-5,1	-6,9	-6,9
81	-1,7	-0,8	-4,2	-8,4	-8,4	-8,4	-7,6
82	2,8	-4,5	-5,1	-5,7	-8,5	-8,5	-8,5
83	-0,5	2,6	-	-2,1	-4,2	-4,7	-4,7
84	3,2	1,8	-3,9	-4,6	-4,6	-6,4	-5,7
85	0,8	-	-2,4	-6,4	-6,4	-6,4	-6,4
86	0,5	2,1	-2,1	-5,2	-7,8	-6,8	-7,3
87	3,1	-1,0	-4,1	-5,7	-7,8	-7,8	-7,3
88	-3,3	-3,3	-4,0	-8,9	-9,3	-9,3	-8,9
89	-	1,8	-	-5,8	-8,2	-8,2	-7,6
90	5,0	-2,3	-2,3	-7,3	-6,9	-8,8	-6,1
91	3,9	1,1	-1,1	-4,2	-7,0	-7,7	-7,7
92	-	-5,3	-2,0	-1,8	-3,8	-5,3	-4,1
93	5,3	-2,1	-1,6	-5,8	-9,5	-7,9	-6,3
94	2,5	0,4	1,1	-3,6	-4,6	-3,9	-4,6
95	6,0	3,1	-	-3,5	-5,0	-7,2	-5,7
96	3,9	3,2	-2,2	-5,4	-6,3	-5,6	-5,6

Apêndice I

Instâncias

Este Apêndice reporta dados sobre as instâncias do PADE. As Tabelas I.1 e I.1 mostram as principais características de cada instância disponível para o PADE. As colunas R, C, L, A, P apresentam, respectivamente, o número de recursos, a capacidade de cada depósito, o número de espaços disponíveis sendo metade destes espaços de trabalho e a outra metade, depósitos e, por fim, o número de períodos. As instâncias citadas no Capítulo 7 (25, 26, 28 e 32) cujas soluções encontradas pelo CPLEX é de custo maior do que os reportados em [18] são apresentadas nas Tabelas I.3 a I.18.

A Tabela I.3 contém a agenda de atividades da instância 25. Nas Tabelas I.4 e I.5 é dada a solução obtida pelo CPLEX, sendo apresentadas a alocação de atividades e de recursos ociosos, respectivamente. O relatório final reportado pelo CPLEX no final do processamento é dado na Tabela I.6.

As Tabelas I.7, I.8, I.9, I.10, I.11, I.12, I.13, I.14, I.15, I.16, I.17 e I.18 contêm, respectivamente, a agenda de atividades, a alocação de atividades e de recursos ociosos e o relatório final do CPLEX para as instâncias 26, 28 e 32.

Tabela I.1: Características das instâncias PADE de 6 e 12 locações

Inst.	R	C	L	A	P
1	9	3	6	6	10
2	9	3	6	6	10
3	9	3	6	9	10
4	9	3	6	9	10
5	9	3	6	6	10
6	9	3	6	6	10
7	9	3	6	9	10
8	9	3	6	9	10
9	9	3	6	9	15
10	9	3	6	9	15
11	9	3	6	14	15
12	9	3	6	14	15
13	9	3	6	9	15
14	9	3	6	9	15
15	9	3	6	14	15
16	9	3	6	14	15
17	9	3	6	12	20
18	9	3	6	12	20
19	9	3	6	18	20
20	9	3	6	18	20
21	9	3	6	13	20
22	9	3	6	13	20
23	9	3	6	20	20
24	9	3	6	20	20
25	18	3	12	10	10
26	18	3	12	10	10
27	18	3	12	16	10
28	18	3	12	16	10
29	18	3	12	10	10
30	18	3	12	10	10
31	18	3	12	16	10
32	18	3	12	16	10
33	18	3	12	16	15
34	18	3	12	16	15
35	18	3	12	24	15
36	18	3	12	24	15
37	18	3	12	15	15
38	18	3	12	15	15
39	18	3	12	24	15
40	18	3	12	24	15
41	18	3	12	21	20
42	18	3	12	21	20
43	18	3	12	33	20
44	18	3	12	33	20
45	18	3	12	20	20
46	18	3	12	20	20
47	18	3	12	34	20
48	18	3	12	34	20

Tabela I.2: Características das instâncias PADE de 20 e 32 locações

Inst.	R	C	L	A	P
49	30	3	20	16	10
50	30	3	20	16	10
51	30	3	20	25	10
52	30	3	20	25	10
53	30	3	20	16	10
54	30	3	20	16	10
55	30	3	20	25	10
56	30	3	20	25	10
57	30	3	20	23	15
58	30	3	20	23	15
59	30	3	20	37	15
60	30	3	20	37	15
61	30	3	20	25	15
62	30	3	20	25	15
63	30	3	20	41	15
64	30	3	20	41	15
65	30	3	20	31	20
66	30	3	20	31	20
67	30	3	20	51	20
68	30	3	20	51	20
69	30	3	20	34	20
70	30	3	20	34	20
71	30	3	20	55	20
72	30	3	20	55	20
73	48	3	32	25	10
74	48	3	32	25	10
75	48	3	32	38	10
76	48	3	32	38	10
77	48	3	32	25	10
78	48	3	32	25	10
79	48	3	32	40	10
80	48	3	32	40	10
81	48	3	32	37	15
82	48	3	32	37	15
83	48	3	32	57	15
84	48	3	32	57	15
85	48	3	32	38	15
86	48	3	32	38	15
87	48	3	32	60	15
88	48	3	32	60	15
89	48	3	32	51	20
90	48	3	32	51	20
91	48	3	32	78	20
92	48	3	32	78	20
93	48	3	32	52	20
94	48	3	32	52	20
95	48	3	32	87	20
96	48	3	32	87	20

Tabela I.3: Atividades, período de início e fim e recursos das atividades na instância 25

Atividade	Início	Fim	Recursos
A1	1	3	2,6
A2	2	4	15
A3	2	6	10,16
A4	3	5	4,17
A5	3	6	12
A6	4	6	1,3
A7	6	9	7,11,18
A8	7	9	14
A9	8	10	5,13
A10	9	10	8,9

Tabela I.4: Solução CPLEX - alocação de atividades para instância 25

Período	Espaço de Trabalho					
	ES1	ES2	ES3	ES4	ES5	ES6
1			A1			
2	A2		A1		A3	
3	A2	A4	A1	A5	A3	
4	A2	A4	A6	A5	A3	
5		A4	A6	A5	A3	
6			A6	A5	A3	A7
7					A8	A7
8				A9	A8	A7
9	A10			A9	A8	
10	A10			A9		

Tabela I.5: Solução CPLEX - alocação de recursos ociosos para instância 25

Período	Depósitos																	
	D1			D2			D3			D4			D5			D6		
1	8	9	15	4	17	1	3	5	12	13	10	14	16	7	11	18		
2	8	9	4	17	1	3	5	12	13	14	7	11	18					
3	8	9	1	3	5	13	14	7	11	18								
4	8	9	2	6	5	13	14	7	11	18								
5	8	9	15	2	6	5	13	14	7	11	18							
6	8	9	15	4	17	2	6	5	13	14								
7	8	9	15	1	4	17	2	3	6	5	12	13	10	16				
8	8	9	15	1	4	17	2	3	6	12	10	16						
9	15	1	4	17	2	3	6	12	10	16								
10	12	15	1	4	17	2	3	6	12	10	14	16	11	18				

Tabela I.6: Solução CPLEX para a instância 25

MIP - Integer optimal solution: Objective = 3.1000000000e+001
Solution time = 57247.27 sec. Iterations = 89805578 Nodes = 2310500

Tabela I.7: Atividades, período de início e fim e recursos das atividades na instância 26

Atividade	Início	Fim	Recursos
A1	1	3	2,5
A2	2	4	9,15
A3	2	6	10,16
A4	3	5	4,14,17
A5	3	6	12
A6	4	6	1,3,5
A7	6	9	7,11,18
A8	7	9	4,14,16
A9	8	10	5,13
A10	9	10	6,8

Tabela I.8: Solução CPLEX - alocação de atividades para instância 26

Período	Espaço de Trabalho					
	ES1	ES2	ES3	ES4	ES5	ES6
1				A1		
2		A3	A2	A1		
3	A4	A3	A2	A1	A5	
4	A4	A3	A2	A6	A5	
5	A4	A3		A6	A5	
6		A3		A6	A5	A7
7	A8					A7
8	A8				A9	A7
9	A8		A10		A9	A7
10			A10		A9	

Tabela I.9: Solução CPLEX - alocação de recursos ociosos para instância 26

Período	Depósitos																
	D1			D2			D3		D4			D5		D6			
1	4	14	17	8	10	16	9	15	1	3	12	13	7	11	18		
2	4	14	17	8					1	3		13	7	11	18		
3				8					1	3		13	7	11	18		
4				8					2	6		13	7	11	18		
5				8			6	9	15	2		13	7	11	18		
6	4	14	17	8			6	9	15	2		13					
7			17	8	10		6	9	15	1	2	3	5	13	12		
8			17	8	10		6	9	15	1	2	3			12		
9			17		10			15		1	2	3			12		
10	4	14	16	17	10			15		1	2	3		12	7	11	18

Tabela I.10: Solução CPLEX para a instância 26

MIP - Integer optimal solution: Objective = 4.3000000000e+001
Solution time = 17595.88 sec. Iterations = 22534864 Nodes = 649940

Tabela I.11: Atividades, período de início e fim e recursos das atividades na instância 28

Atividade	Início	Fim	Recursos
A1	1	2	7,17,18
A2	1	3	2,5,6
A3	1	4	8,11,13
A4	2	4	9,2
A5	2	6	10
A6	3	5	4,14,17
A7	3	6	12
A8	4	6	1,3,5
A9	5	7	8,9,13
A10	5	8	2,6,15
A11	6	9	7,11,18
A12	7	9	4,14,16
A13	7	10	1,10,17
A14	7	10	12
A15	8	10	5,1
A16	9	10	6,8,9

Tabela I.12: Solução CPLEX - alocação de atividades para instância 28

Período	Espaço de Trabalho					
	ES1	ES2	ES3	ES4	ES5	ES6
1			A1	A2		A3
2		A5	A1	A2	A4	A3
3	A7	A5	A6	A2	A4	A3
4	A7	A5	A6	A8	A4	A3
5	A7	A5	A6	A8	A10	A9
6	A7	A5	A11	A8	A10	A9
7	A14	A12	A11	A13	A10	A9
8	A14	A12	A11	A13	A10	A15
9	A14	A12	A11	A13	A16	A15
10	A14			A13	A16	A15

Tabela I.13: Solução CPLEX - alocação de recursos ociosos para instância 28

Período	Depósitos											
	D1		D2		D3		D4		D5		D6	
1	12		10	16	4	14	1	3	9	15		
2	12				4	14	1	3				
3					7	18	1	3				
4					7	18	6		2			
5					7	18			11			
6			4		14		17					
7							3				5	
8							3		8	9		
9							3		15			
10			4	14	16	7	11	18	3	15	2	

Tabela I.14: Solução CPLEX para a instância 28

MIP - Integer optimal solution: Objective = 5.5000000000e+001
SSolution time = 598.84 sec. Iterations = 288730 Nodes = 3325

Tabela I.15: Atividades, período de início e fim e recursos das atividades na instância 32

Atividade	Início	Fim	Recursos
A1	1	1	2,4,14
A2	1	2	3,5,10
A3	1	5	8,13,16
A4	1	6	11
A5	2	3	7,14,18
A6	2	3	6,9,17
A7	2	7	1,2
A8	3	5	2,4,10
A9	4	8	18
A10	5	10	3,7,17
A11	6	6	13,14,16
A12	6	10	5,8,9
A13	7	9	6,10,11
A14	7	10	4,12,16
A15	8	10	2,1
A16	9	9	13,15,18

Tabela I.16: Solução CPLEX - alocação de atividades para instância 32

Período	Espaço de Trabalho					
	ES1	ES2	ES3	ES4	ES5	ES6
1			A1	A2		A3
2		A5	A1	A2	A4	A3
3	A7	A5	A6	A2	A4	A3
4	A7	A5	A6	A8	A4	A3
5	A7	A5	A6	A8	A10	A9
6	A7	A5	A11	A8	A10	A9
7	A14	A12	A11	A13	A10	A9
8	A14	A12	A11	A13	A10	A15
9	A14	A12	A11	A13	A16	A15
10	A14			A13	A16	A15

Tabela I.17: Solução CPLEX - alocação de recursos ociosos para instância 32

Período	Depósitos										
	D1		D2		D3	D4		D5		D6	
1	1	15	7	18				6	9	17	
2			2		4						
3						3	5				
4			14		7	3	5	6	9	17	
5					14		5	6	9		
6			2		4						6 10
7			2	13	14						
8	1		15	13							
9	1										
10	1		13	15	18						6 10 11

Tabela I.18: Solução CPLEX para a instância 32

MIP - Integer optimal solution: Objective = 6.9000000000e+001
Solution time = 3897.77 sec. Iterations = 2774638 Nodes = 40833