5
Resultados

Os algoritmos propostos neste trabalho foram implementados na linguagem C. Foi utilizado o compilador do ambiente integrado de desenvolvimento Microsoft Visual C++ ® 6.0.

Cada instância do problema, composta pelos grafos G_M e G_I e pelas matrizes de similaridades entre vértices e arestas, foi representada por uma estrutura com as matrizes e listas de adjacências dos dois grafos e com as matrizes de similaridades. Cada solução foi representada por um vetor com as associações de cada vértice de V_I e um vetor de listas com os conjuntos $A_S(i)$, $\forall i \in V_M$.

Os experimentos computacionais foram realizados com os mesmos 12 problemas teste utilizados em [14] em um microcomputador Pentium IV 2.0 GHz, com 256 Mb de memória principal, sob o sistema operacional Windows 2000. O resolvedor de problemas de programação linear e inteira utilizado foi o ILOG CPLEX versão 9.0. O gerador de números aleatórios utilizado foi uma implementação em linguagem C da versão original proposta por Schrage [62].

5.1
Problemas teste

Os dados dos 12 problemas teste utilizados nos experimentos realizados neste trabalho foram fornecidos por Boeres [14].

Os problemas I_6 e I_7 foram construídos a partir da descrição de imagens reais. O primeiro a partir de imagens do cérebro humano e o segundo a partir de imagens 2D de um corte de um músculo. Todos os demais foram construídos artificialmente, com valores de similaridades entre

vértices e arestas gerados aleatoriamente no intervalo $[0, 1]$. O grafo G_I do problema I_5 e o grafo G_M do problema I_8 apresentam vértices isolados. Os problemas I_{5a} e I_{8a} foram criados a partir de I_5 e I_8, respectivamente, com grafos sem vértices isolados.

A Tabela 5.1 apresenta os números de vértices e arestas para os grafos de cada problema teste.

Problema teste	$\|V_M\|$	$\|E_M\|$	$\|V_I\|$	$\|E_I\|$
I_0	3	2	7	10
I_1	4	5	9	15
I_2	5	8	12	23
I_3	6	9	13	25
I_4	7	11	15	29
I_5	10	15	30	39
I_{5a}	10	15	30	41
I_6	12	42	95	1434
I_7	14	27	28	63
I_8	30	39	100	297
I_{8a}	30	42	100	297
I_9	50	88	250	1681

Tabela 5.1: Problemas teste.

5.2 Resolução do modelo de programação inteira

A Tabela 5.2 apresenta os resultados da resolução do problema de programação inteira, cuja formulação foi apresentada no Capítulo 3, e de sua respectiva relaxação linear. Tal formulação permitiu a resolução exata de nove dos 12 problemas teste utilizados, além do cálculo de limites superiores para 11 das 12 instâncias. São apresentados os valores da função objetivo baseada exclusivamente na contribuição de vértices para a solução ótima do problema inteiro (f_{PI}) e para sua relaxação linear (f_{RL}), sendo que esta última representa um limite superior para o problema. São mostrados também os tempos de execução para cada problema teste.

As colunas T_1 a T_4 desta tabela indicam tempos de processamento em segundos. A coluna T_1 representa o tempo de resolução da relaxação linear. A coluna T_2 representa o tempo de resolução do problema de programação inteira com a utilização de um limite inferior obtido pelo algoritmo GRASP com reconexão por caminhos modificada. Conforme

Problema teste	f_{RL}	T_1	f_{PI}	T_2	T_3	T_4
I_0	0,9771	0,01	0,9771	0,44	0,58	0,02
I_1	0,8372	0,02	0,8322	0,02	0,28	0,02
I_2	0,8092	0,01	0,8062	0,08	0,56	0,06
I_3	0,7340	0,02	0,7340	0,03	0,52	0,05
I_4	0,8018	0,02	0,8018	0,09	1,09	0,11
I_5	0,5863	0,12	0,5676	6701,39	6710,09	7113,34
I_{5a}	0,5863	0,11	0,5690	2894,19	2902,81	2559,45
I_6	0,4295	1,38	0,4294	277,31	455,80	23668,17
I_7	0,7037	0,11	0,6999	10,58	16,52	113,84
I_8	0,5331	4,27	—	—	—	—
I_{8a}	0,5331	4,12	—	—	—	—
I_9	—	—	—	—	—	—

Tabela 5.2: Resultados da resolução do modelo de programação inteira. f_{RL} é o valor da solução ótima da relaxação linear; T_1 é o tempo da resolução da relaxação linear; f_{PI} é o valor da solução ótima do problema inteiro; T_2 é o tempo da resolução do modelo dado um limite inferior como parâmetro; T_3 é o tempo total de execução (cálculo do limite e resolução do modelo) e T_4 é o tempo da resolução do modelo sem a utilização de um limite inferior. Todos os tempos são fornecidos em segundos.

posteriormente discutido na Seção 5.4, o GRASP foi executado com o parâmetro $MaxIter$ igual a 200 e utilizou-se do algoritmo `Construtivo_2`, que retorna a melhor entre 50 soluções construídas, obtidas em no máximo 100 tentativas. A coluna T_3 mostra o tempo total de resolução do modelo com a utilização do limite inferior, ou seja, representa a soma de T_2 com o tempo de execução do algoritmo GRASP. Finalmente, a coluna T_4 representa o tempo de resolução do problema inteiro sem a utilização do limite inferior.

Em virtude do grande número de variáveis e restrições, o resolvedor não conseguiu encontrar uma solução inteira para os problemas I_8 e I_{8a} antes que se esgotasse um tempo limite de 8 horas. Para o problema I_9, nem mesmo a relaxação linear pode ser concluída, devida à falta de memória.

Pela inspeção da Tabela 5.2 pode-se notar que para os problemas I_0 a I_4, I_6 e I_7 o valor da relaxação linear foi igual ou bem próximo do valor da solução ótima. Somente para os problemas I_5 e I_{5a} estes limites não se aproximaram da otimalidade. Uma explicação plausível para este resultado é o fato de que os problemas I_5 e I_{5a} apresentam valores de similaridades entre vértices e arestas muito próximos, de forma que não há uma distinção clara entre os vértices, dificultando-se assim o processo de correspondência. Além de I_5 e I_{5a}, os problemas I_6, I_8 e I_{8a} também apresentam tal característica.

Entretanto, o problema I_6 apresenta valores de similaridade mais altos que a média para alguns vértices ou arestas, que auxiliam no processo de correspondência. Como as instâncias I_8 e I_{8a} não foram resolvidas de forma ótima, não se pode garantir a distância da relaxação linear para o valor ótimo, entretanto as características das instâncias sugerem que essa distância não seja desprezível.

Em relação aos tempos de processamento, exceto para os problemas teste menores, a utilização do limite inferior obtido pelo algoritmo aproximado mostrou-se bastante útil, pois reduziu o tempo total de execução em relação à resolução modelo quando o limite superior não foi utilizado. Apenas a instância I_{5a} apresentou um tempo inferior sem a a utilização de um limite, mesmo em comparação à resolução do modelo com utilização do limite descontando-se o tempo de execução da heurística. Para as instâncias I_6 e I_7 o limite superior proporcionou notáveis melhoras dos tempos de processamento.

5.3
Métodos construtivos

Esta seção mostra os resultados obtidos pelos algoritmos construtivos apresentados no Capítulo 4, aplicados aos problemas teste citados na Seção 5.1.

Estes resultados referem-se a 20 execuções de cada algoritmo com diferentes sementes para a geração dos números pseudo-aleatórios. Os algoritmos foram executados com os parâmetros $MaxTentativas = 500$ e $MaxSolucoes = 100$. Foram selecionados valores de α iguais a $0,5$ e $1,0$. O valor 0,5 foi utilizado com o objetivo de se comparar os resultados dos algoritmos propostos com aqueles obtidos por Boeres [14]. Por sua vez, se $\alpha = 1,0$ a contribuição de arestas é eliminada da função f, que passa a considerar exclusivamente a contribuição de vértices. Desta forma, permite-se a comparação do resultado dos algoritmos construtivos com os da resolução exata.

As Tabelas 5.3 e 5.4 mostram os resultados para $\alpha = 1,0$. Pode-se notar que para os problemas I_0 a I_2 todos os algoritmos obtiveram soluções muito próximas ou iguais à solução ótima. Dentre os quatro algoritmos propostos, `Construtivo_2` apresentou os melhores resultados para todas

Problema teste	Construtivo_	T_m	f^{PI}	f^*
I_0	1	0,001	0,9771*	**0,9771**
	2	0,001		**0,9771**
	2a	0,001		**0,9771**
	3	0,001		**0,9771**
I_1	1	0,001	0,8322*	**0,8322**
	2	0,001		**0,8322**
	2a	0,002		**0,8322**
	3	0,002		**0,8322**
I_2	1	0,002	0,8062*	**0,8062**
	2	0,002		**0,8062**
	2a	0,002		**0,8062**
	3	0,004		**0,8062**
I_3	1	0,002	0,7340*	0,7315
	2	0,002		**0,7340**
	2a	0,005		**0,7340**
	3	0,005		0,7289
I_4	1	0,005	0,8018*	0,8000
	2	0,005		**0,8018**
	2a	0,014		0,8000
	3	0,034		0,7868

Tabela 5.3: Resultados dos métodos construtivos com $\alpha = 1,0$ para as instâncias I_0 a I_4. Os valores em negrito representam o melhor resultado para cada instância, dentre os quatro algoritmos. T_m é a média dos tempos de processamento para as 20 execuções; f^{PI} é o valor da solução obtida pela resolução do modelo de programação linear inteira; se disponível, é exibido o valor da solução ótima (indicada por um *), caso contrário apresenta-se o valor da relaxação linear; f^* é o valor da melhor solução encontrada por cada algoritmo construtivo.

as instâncias exceto I_5, tendo encontrado soluções ótimas também para as instâncias I_3, I_4 e I_7, e uma muito próxima da ótima para I_6. Para as instâncias I_5 e I_{5a} o valor ótimo não foi alcançado por algoritmo algum.

Dentre os demais algoritmos, Construtivo_3 obteve melhores soluções exceto para as instâncias I_3 e I_4, para as quais Construtivo_1 e Construtivo_2a foram superiores. Para a instância I_5, Construtivo_3 foi superior até mesmo a Construtivo_2. Os algoritmos Construtivo_1 e Construtivo_2a apresentaram resultados mais fracos para a maioria dos problemas teste.

As Tabelas 5.5 e 5.6 apresentam os resultados obtidos para $\alpha = 0,5$. Novamente, todos os algoritmos alcançaram resultados muito bons e semelhantes para as instâncias menores, com os algoritmos Construtivo_2 e Construtivo_3 obtendo melhores resultados para os problemas maiores.

Problema teste	Construtivo_	T_m	f^{PI}	f^*
I_5	1	0,018	0,5676*	0,5255
	2	0,039		0,5548
	2a	0,042		0,5257
	3	0,025		**0,5567**
I_{5a}	1	0,019	0,5690*	0,5324
	2	0,038		**0,5617**
	2a	0,044		0,5302
	3	0,024		0,5539
I_6	1	0,045	0,4294*	0,4209
	2	0,582		**0,4289**
	2a	0,127		0,4214
	3	0,787		0,4252
I_7	1	0,017	0,6999*	0,6941
	2	0,028		**0,6999**
	2a	0,045		0,6938
	3	0,048		0,6995
I_8	1	0,138	0,5331	0,5015
	2	0,405		**0,5242**
	2a	0,555		0,5054
	3	0,329		0,5178
I_{8a}	1	0,138	0,5331	0,5015
	2	0,412		**0,5242**
	2a	0,561		0,5054
	3	0,327		0,5178
I_9	1	0,379	—	0,5036
	2	1,052		**0,5213**
	2a	1,266		0,5077
	3	1,205		0,5147

Tabela 5.4: Resultados dos métodos construtivos com $\alpha = 1,0$ para as instâncias I_5 a I_9, I_{5a} e I_{8a}. Os valores em negrito representam o melhor resultado para cada instância, dentre os quatro algoritmos. T_m é a média dos tempos de processamento para as 20 execuções; f^{PI} é o valor da solução obtida pela resolução do modelo de programação linear inteira; se disponível, é exibido o valor da solução ótima (indicada por um *), caso contrário apresenta-se o valor da relaxação linear; f^* é o valor da melhor solução encontrada por cada algoritmo construtivo.

Estas duas tabelas permitem uma comparação destes algoritmos com os resultados obtidos por Boeres [14]. O algoritmo Construtivo_2 obteve ótimos ou melhores soluções que aquelas apresentas em [14] para todas as instâncias teste. Construtivo_3 também foi superior na maioria dos casos, perdendo apenas na instâncias I_3 e I_4.

Analisando-se os resultados apresentados nesta seção, percebe-se que o algoritmo Construtivo_2a, na maior parte dos casos, obteve melhores re-

Problema teste	Construtivo_	T_m	f^*	f_m	f_1
I_0	1	0,001	**0,6142**	**0,6142**	0,6142
	2	0,000	**0,6142**	**0,6142**	
	2a	0,001	**0,6142**	**0,6142**	
	3	0,002	**0,6142**	**0,6142**	
I_1	1	0,002	**0,5202**	0,5181	0,5135
	2	0,002	**0,5202**	**0,5202**	
	2a	0,002	**0,5202**	0,5157	
	3	0,002	**0,5202**	**0,5202**	
I_2	1	0,002	**0,5098**	0,5073	0,5090
	2	0,002	**0,5098**	**0,5098**	
	2a	0,003	0,5078	0,5043	
	3	0,004	**0,5098**	**0,5098**	
I_3	1	0,002	0,6309	0,6163	0,6020
	2	0,002	**0,6375**	**0,6375**	
	2a	0,005	0,6222	0,6121	
	3	0,010	0,6309	0,6294	
I_4	1	0,003	0,6738	0,6635	0,6559
	2	0,005	**0,6754**	**0,6754**	
	2a	0,007	0,6668	0,6528	
	3	0,015	0,6627	0,6620	

Tabela 5.5: Resultados dos métodos construtivos com $\alpha = 0,5$ para as instâncias I_0 a I_4. Os valores em negrito representam o melhor resultado para cada instância, dentre os quatro algoritmos. T_m é a média dos tempos de processamento para as 20 execuções; f^* é o valor da melhor solução encontrada por cada algoritmo construtivo; f_m é a média dos valores das soluções obtidas nas 20 execuções; f_1 é o valor da solução obtida por Boeres [14].

sultados que Construtivo_1, completamente aleatório, mas não conseguiu melhorar os resultados do algoritmo guloso Construtivo_2. Este fato sugere que a aleatorização da escolha gulosa feita de forma local, ou seja, para um dado vértice de V_I escolher um vértice de V_M diferente daquele com melhores contribuições, não resultou em boas soluções. Entretanto, a aleatorização introduzida juntamente com a lista de candidatos do algoritmo Construtivo_3 obteve resultados quase tão bons quanto Construtivo_2, sendo melhor para o problema I_5. É importante notar, contudo, que Construtivo_3 apresentou um certo grau de instabilidade, visto que obteve os piores resultados dentre os quatro algoritmos para duas das instâncias.

Em relação aos tempos de processamento, o algoritmo Construtivo_1 foi o mais rápido, pois não necessita realizar o cálculo de uma função gulosa. Em geral, o mais lento foi Construtivo_3 seguido de Construtivo_2a. Todas as instâncias consumiram menos de um segundo para cada algoritmo,

Problema teste	`Construtivo_`	T_m	f^*	f_m	f_1
I_5	1	0,018	0,5096	0,5027	0,5038
	2	0,041	0,5257	0,5210	
	2a	0,045	0,5130	0,5035	
	3	0,027	**0,5268**	**0,5252**	
I_{5a}	1	0,008	0,5217	0,5189	0,5223
	2	0,016	**0,5416**	**0,5367**	
	2a	0,019	0,5258	0,5192	
	3	0,026	0,5394	0,5364	
I_6	1	0,045	0,4052	0,4046	0,4045
	2	0,177	**0,4085**	**0,4077**	
	2a	0,161	0,4057	0,4049	
	3	0,396	0,4048	0,4043	
I_7	1	0,009	0,3791	0,3768	0,3757
	2	0,023	**0,3871**	**0,3847**	
	2a	0,027	0,3775	0,3755	
	3	0,045	0,3843	0,3828	
I_8	1	0,055	0,4986	0,4976	0,5023
	2	0,173	**0,5108**	**0,5099**	
	2a	0,239	0,5010	0,4986	
	3	0,348	0,5069	0,5057	
I_{8a}	1	0,055	0,5164	0,5154	0,5200
	2	0,177	**0,5287**	**0,5277**	
	2a	0,239	0,5190	0,5164	
	3	0,366	0,5251	0,5235	
I_9	1	0,365	0,4993	0,4989	0,5031
	2	1,173	**0,5075**	**0,5068**	
	2a	1,284	0,5002	0,4997	
	3	1,409	0,5045	0,5042	

Tabela 5.6: Resultados dos métodos construtivos com $\alpha = 0,5$ para as instâncias I_5 a I_9, I_{5a} e I_{8a}. Os valores em negrito representam o melhor resultado para cada instância, dentre os quatro algoritmos. T_m é a média dos tempos de processamento para as 20 execuções; f^* é o valor da melhor solução encontrada por cada algoritmo construtivo; f_m é a média dos valores das soluções obtidas nas 20 execuções; f_1 é o valor da solução obtida por Boeres [14].

exceto o problema teste I_9 que custou pouco mais de um segundo aos algoritmos `Construtivo_2`, `Construtivo_2a` e `Construtivo_3`. Neste caso, estes algoritmos, em especial `Construtivo_2` e `Construtivo_3`, pela qualidade das soluções construídas, constituem uma boa opção para a composição de um algoritmo GRASP.

5.4 GRASP

As tabelas desta seção apresentam os resultados computacionais das três variantes de GRASP propostas no Capítulo 4. Estes resultados são referentes a 10 execuções com diferentes sementes para geração de números aleatórios. Em todos os casos, o GRASP teve como parâmetro $MaxIter = 200$ e o método construtivo utilizado na fase de construção utilizou $MaxTentativas = 100$ e $MaxSolucoes = 50$. Os algoritmos atingiram a solução ótima para praticamente todos os testes com as instâncias menores, com pequenas distâncias nos poucos casos em que o ótimo não foi encontrado. Portanto, nesta seção são apresentados apenas os resultados para as instâncias I_5 a I_9, I_{5a} e I_{8a}.

As Tabelas 5.7 e 5.8 têm como finalidade a comparação dos resultados das três variantes implementadas do GRASP. Todas utilizam o mesmo algoritmo `Construtivo_2` na fase de construção, que foi aquele que obteve os melhores resultados dentre os métodos construtivos. Os valores de α utilizados foram novamente 0,5 e 1,0.

Assim como para os métodos construtivos, os resultados obtidos com $\alpha = 1,0$ são comparados aos da resolução exata e aqueles obtidos com $\alpha = 0,5$ são comparados com os resultados apresentados em [14]. Como esperado, o GRASP com reconexão por caminhos modificada foi o que obteve os melhores resultados, uma vez que apresenta elementos de otimização adicionais em relação aos outros dois algoritmos. Para as instâncias I_5 a I_7 e I_{5a} todos os três algoritmos alcançaram soluções ótimas ou muito próximas da ótima para $\alpha = 1,0$. Para $\alpha = 0,5$, os algoritmos propostos neste trabalho superaram os resultados de Boeres [14], exceto para a instância I_9 na qual o algoritmo de Boeres se igualou ao valor médio das soluções encontradas pelo GRASP com reconexão por caminhos modificada tendo sido superior aos outros dois propostos.

Os tempos de processamento para os algoritmos apresentaram variações de até 50,003% (instância I_9 com $\alpha = 0,5$). A variante básica do GRASP foi a mais rápida para todos os testes. Este resultado era esperado, pois as demais variantes apresentam computações adicionais em relação à básica. Entre as variações com reconexão por caminhos, a versão que permite inviabilidade no caminho de busca obteve melhores soluções sem um grande aumento no tempo de processamento.

Problema teste	GRASP	T_m	f^{PI}	f^*
I_5	Básico	3,252	0,5677*	0,5642
	Básico + RC	3,366		0,5644
	Básico + RC$^+$	3,575		**0,5666**
I_{5a}	Básico	1,864	0,5691*	0,5669
	Básico + RC	1,988		0,5669
	Básico + RC$^+$	2,141		**0,5683**
I_6	Básico	83,642	0,4295*	**0,4295**
	Básico + RC	86,855		**0,4295**
	Básico + RC$^+$	86,445		**0,4295**
I_7	Básico	2,236	0,6999*	**0,6999**
	Básico + RC	2,284		**0,6999**
	Básico + RC$^+$	2,423		**0,6999**
I_8	Básico	25,906	0,5331	0,5281
	Básico + RC	31,391		0,5286
	Básico + RC$^+$	33,039		**0,5289**
I_{8a}	Básico	26,480	0,5331	0,5281
	Básico + RC	30,773		0,5286
	Básico + RC$^+$	33,330		**0,5289**
I_9	Básico	210,459	—	0,5245
	Básico + RC	311,303		0,5258
	Básico + RC$^+$	328,948		**0,5259**

Tabela 5.7: Resultados dos algoritmos GRASP, utilizando o algoritmo `Construtivo_2` com $\alpha = 1,0$. Os valores em negrito representam o melhor resultado para cada instância, dentre as três variantes. A segunda coluna indica cada variante da heurística GRASP: básico, básico com reconexão por caminhos (Básico + RC) ou básico com reconexão por caminhos modificada (Básico + RC$^+$); T_m é a média dos tempos de processamento para as 10 execuções; f^{PI} é o valor da solução obtida pela resolução do modelo de programação linear inteira. Se disponível, é exibido o valor da solução ótima (indicada por *), caso contrário apresenta-se o valor da relaxação linear; f^* é o valor da melhor solução encontrada por cada variante de GRASP.

As Tabelas 5.9 e 5.10 permitem a comparação do desempenho da variante GRASP com reconexão por caminhos modificada utilizando, na fase de construção, cada um dos algoritmos construtivos. Este tipo de comparação se mostra interessante, uma vez que a qualidade das soluções obtidas por um método de construção não é o único fator que influencia na eficiência do GRASP. Por constituir um método iterativo de construção e busca local, um algoritmo GRASP é sensível à diversidade das soluções construídas. Como comentado no Capítulo 4, quanto maior essa diversidade, mantendo-se um nível de qualidade, maior a chance de que o GRASP consiga uma boa solução final.

Assim como nos testes dos métodos construtivos individualmente, os

Problema teste	GRASP	T_m	f^*	f_m	f_1
I_5	Básico	3,422	0,5302	0,5293	0,5271
	Básico + RC	3,547	0,5460	0,5382	
	Básico + RC$^+$	3,759	**0,5481**	**0,5395**	
I_{5a}	Básico	1,936	0,5439	0,5429	0,5389
	Básico + RC	2,069	0,5595	0,5524	
	Básico + RC$^+$	2,247	**0,5633**	**0,5587**	
I_6	Básico	84,516	0,4103	0,4103	0,4102
	Básico + RC	95,950	**0,4117**	**0,4115**	
	Básico + RC$^+$	97,606	**0,4117**	**0,4115**	
I_7	Básico	2,455	0,3887	0,3885	0,3855
	Básico + RC	2,539	0,4065	0,4009	
	Básico + RC$^+$	2,712	**0,4247**	**0,4177**	
I_8	Básico	28,361	0,5135	0,5132	0,5124
	Básico + RC	33,855	**0,5178**	**0,5155**	
	Básico + RC$^+$	36,061	**0,5178**	0,5149	
I_{8a}	Básico	28,809	0,5314	0,5311	0,5302
	Básico + RC	33,583	**0,5338**	0,5323	
	Básico + RC$^+$	36,341	**0,5338**	**0,5325**	
I_9	Básico	268,991	0,5099	0,5097	0,5106
	Básico + RC	364,809	0,5106	0,5105	
	Básico + RC$^+$	403,495	**0,5112**	**0,5106**	

Tabela 5.8: Resultados dos algoritmos GRASP, utilizando o algoritmo `Construtivo_2` com $\alpha = 0,5$. Os valores em negrito representam o melhor resultado para cada instância, dentre as três variantes. A segunda coluna indica cada variante da heurística GRASP: Básico, Básico com reconexão por caminhos (Básico + RC) ou Básico com reconexão por caminhos modificada (Básico + RC$^+$); T_m é a média dos tempos de processamento para as 10 execuções; f^* é o valor da melhor solução encontrada por cada variante de GRASP; f_m é a média dos valores das soluções das 10 execuções. f_1 é o valor da solução obtida por Boeres [14].

algoritmos `Construtivo_2` e `Construtivo_3` foram os que proporcionaram os melhores resultados ao GRASP. Nos 14 testes realizados (sete instâncias para dois valores de α), o algoritmo `Construtivo_2` obteve a melhor solução encontrada em nove casos. Para $\alpha = 0,5$, obteve a melhor média em cinco dos sete testes realizados. Por sua vez, o algoritmo `Construtivo_3` alcançou a melhor solução encontrada em cinco casos e, para $\alpha = 0,5$, obteve a melhor média em dois (I_5 e I_7). Na maioria dos demais testes obteve soluções muito próximas às do `Construtivo_2`. Apesar do algoritmo `Construtivo_2` ter obtido as melhores soluções, tais resultados reafirmam o fato de que a maior diversidade de soluções proporcionada pelo maior grau de aleatorização do algoritmo `Construtivo_3` também pode ser útil para utilização num algoritmo GRASP.

Problema teste	`Construtivo_`	T_m	f^*
I_5	1	2,484	0,5632
	2	3,575	**0,5666**
	2a	4,009	**0,5666**
	3	3,436	0,5656
I_{5a}	1	1,809	0,5649
	2	2,141	**0,5683**
	2a	2,813	0,5676
	3	3,339	0,5671
I_6	1	87,545	**0,4295**
	2	86,445	**0,4295**
	2a	94,569	**0,4295**
	3	111,978	**0,4295**
I_7	1	1,544	**0,6999**
	2	2,423	**0,6999**
	2a	3,133	**0,6999**
	3	4,975	**0,6999**
I_8	1	47,813	0,5265
	2	33,039	0,5283
	2a	59,958	0,5261
	3	67,880	**0,5284**
I_{8a}	1	47,791	0,5265
	2	33,330	0,5283
	2a	60,178	0,5261
	3	67,567	**0,5284**
I_9	1	612,566	0,5238
	2	328,948	**0,5248**
	2a	641,464	0,5240
	3	421,903	0,5241

Tabela 5.9: Resultados do algoritmo GRASP com reconexão por caminhos modificada com $\alpha = 1,0$. Os valores em negrito representam o melhor resultado para cada instância, dentre as quatro variantes. T_m é a média dos tempos de processamento para as 10 execuções; f^* é o valor da melhor solução encontrada por cada variante.

5.5 Discussão

Este capítulo apresentou os resultados computacionais dos algoritmos aproximados e exato para o PCIG. A Seção 5.2 mostrou que o modelo de programação inteira possibilitou a resolução exata de nove das 12 instâncias teste utilizadas neste trabalho. Além disso, foram calculados limites superiores, dados pela relaxação linear do modelo, para 11 dos 12 problemas. Somente para a instância I_9 não foi possível a resolução do

Problema teste	Construtivo_	T_m	f^*	f_m
I_5	1	2,495	**0,5531**	0,5436
	2	3,759	0,5481	0,5395
	2a	4,039	0,5486	0,5382
	3	3,444	0,5502	**0,5452**
I_{5a}	1	1,797	0,5582	0,5518
	2	2,247	0,5633	**0,5587**
	2a	2,914	0,5553	0,5509
	3	3,277	**0,5684**	0,5571
I_6	1	116,559	0,4116	0,4114
	2	97,606	**0,4117**	**0,4115**
	2a	128,008	0,4116	0,4114
	3	153,633	0,4115	0,4114
I_7	1	1,789	0,4154	0,4050
	2	2,712	0,4247	0,4177
	2a	3,530	0,4131	0,4032
	3	5,444	**0,4452**	**0,4264**
I_8	1	48,828	0,5143	0,5132
	2	36,061	**0,5178**	**0,5149**
	2a	66,959	0,5155	0,5139
	3	66,195	0,5155	0,5143
I_{8a}	1	49,169	0,5322	0,5315
	2	36,341	**0,5338**	**0,5325**
	2a	67,569	0,5321	0,5314
	3	65,509	0,5323	0,5319
I_9	1	617,987	0,5108	0,5103
	2	403,495	**0,5112**	**0,5106**
	2a	739,594	0,5109	0,5103
	3	459,497	0,5105	0,5101

Tabela 5.10: Resultados do algoritmo GRASP com reconexão por caminhos modificada com $\alpha = 0,5$. Os valores em negrito representam o melhor resultado para cada instância, dentre as quatro variantes. T_m é a média dos tempos de processamento para as 10 execuções; f^* é o valor da melhor solução encontrada por cada variante; f_m é a média dos valores das soluções das 10 execuções.

modelo, nem mesmo sua relaxação linear.

Comparações feitas com os resultados apresentados em [14, 15] mostraram que os métodos construtivos propostos neste trabalho alcançaram soluções melhores para todos os problemas teste. Além disso, os baixos tempos de processamento necessários a estes algoritmos possibilitaram sua combinação à busca local $\texttt{BL}_{a+b}$ e aos procedimentos de reconexão por caminhos, o que resultou em algoritmos GRASP cujos resultados foram também mostrados neste capítulo. Estes algoritmos permitiram que soluções ótimas, ou muito próximas da ótima, fossem encontradas para praticamente

todos os problemas teste utilizados.

Outra aplicação desses algoritmos aproximados foi o cálculo de limites inferiores que foram adicionados ao modelo de programação inteira. Estes limites se mostraram muito úteis, uma vez que proporcionaram uma significativa diminuição nos tempos de processamento da resolução exata modelo pelo resolvedor utilizado.