6
Experimentos e Resultados

Neste capítulo apresentamos os experimentos realizados e os resultados obtidos pelas heurísticas apresentadas. A primeira parte do capítulo aborda os experimentos e resultados obtidos pela busca tabu reativa que estão apresentados em [26]. O objetivo destes experimentos foi o de estabelecer os primeiros limites para as instâncias do problema de códigos curtos de cobertura, onde a qualidade dos limites obtidos foram atestados pelos resultados apresentados pela busca tabu reativa para o problema clássico de códigos de cobertura. A segunda parte do capítulo aborda os experimentos realizados para comparação de desempenho entre a busca tabu reativa, a busca tabu sem o mecanismo de reação e a HMGC apresentada. O principal objetivo destes experimentos foi o de testar a validade e a qualidade da HMGC, avaliando se a HMGC contribui para a melhoria dos resultados da busca tabu reativa e da busca tabu clássica aplicada aos problemas de códigos de cobertura.

Os algoritmos foram implementados utilizando a linguagem de programação C++ e os experimentos foram realizados em um computador com processador Intel(R) Core(TM)2 Duo 2.2 GHz e 3GB de memória DDR2.

6.1
Limites Iniciais para Coberturas Curtas via Busca Tabu Reativa

Foram realizados experimentos tanto para o problema de códigos curtos de cobertura quanto para o problema clássico de códigos de cobertura. Os experimentos no problema clássico tiveram como objetivo demonstrar a qualidade da busca tabu reativa, de forma a nos dar razão para acreditar que as soluções e, por consequência, os limites, apresentadas para as coberturas curtas são soluções próximas da otimalidade.

Os parâmetros do algoritmo foram determinados empiricamente. Eles foram ajustados para cada problema e tamanho de instância. Os parâmetros principais são **Max_Visits**, **Chaos**, **Cycle_Min**, **Min_Tenure**, **Max_Tenure**, **Increase** e **Decrease**. Para os primeiros três parâmetros foram utilizados os valores de 10, 5 e 8 respectivamente para ambos os problemas. Os valores do tabu tenure ficaram na faixa entre 20 e 400 para o problema clássico, enquanto que para o de coberturas curtas ele variou de 5 a 200. Os valores dos parâmetros **increase** e **decrease** foram setados para 1.2 e 0.9 respectivamente.

As configurações de parâmetros utilizados tiveram como objetivo deixar a busca tabu ter uma mudança flexível do tabu tenure e escapar de regiões não-promissoras sem demorar um número excessivo de iterações. Desta forma se espera que a busca obtenha soluções próximas da otimalidade.

O algoritmo implementado se inicia com uma solução inicial gerada através de duas estratégias distintas, sendo estas: (i) geração aleatória: onde palavras são adicionadas aleatoriamente ao código atual até que o mesmo se torne uma solução viável; (ii) geração gulosa: a partir do conjunto de palavras não cobertas se adiciona ao código atual a palavra que cubra o maior número de palavras, repetindo este procedimento até que o código atual se torne uma solução viável. O algoritmo foi executado dez vezes para cada instância e para cada uma das estratégias de geração inicial de solução.

6.1.1
Resultados para o Problema Clássico de Códigos de Cobertura

Encontrar soluções próximas da otimalidade para R-coberturas é muitas vezes uma tarefa difícil. A tabela 6.1.1 apresentada abaixo apresenta os limites superiores obtidos nos experimentos para $K_q(n, R)$. Nesta tabela **NI** representa o número de iterações executadas, **TF** indica o tempo de execução em segundos para encontrar a melhor solução da busca, **TT** indica o tempo total de execução do algoritmo. A coluna **RTS** indica o limite superior obtido para cada instância utilizando a busca tabu reativa proposta, enquanto **BestKnown** denota o melhor valor de solução presente na literatura para cada instância, de acordo com Kéri [18]. A escolha das instâncias de R-coberturas testadas foi feita de acordo com a relevância das mesmas nos trabalhos relacionados.

A apresentação desta tabela para o problema clássico de códigos de cobertura é importante também por permitir a comparação das cardinalidades apresentadas com as cardinalidades correspondentes do problema de coberturas curtas. A questão a ser respondida é de quanto é a redução de tamanho dos códigos quando a versão curta de cobertura é considerada.

n	q	R	NI	TF(s)	TT(s)	RTS	BestKnown
5	3	1	5000	0.008	0.148	27	27
5	3	2	5000	0.156	0.201	8	8
6	3	1	10000	0.127	0.576	73	73
6	3	2	10000	0.284	1.260	17	17
6	3	3	10000	0.006	3.380	6	6
7	3	1	100000	4.496	9.400	186	186
7	3	2	100000	20.141	33.694	34	34
7	3	3	100000	11.696	94.737	12	12
5	4	1	10000	0.168	0.880	64	64
5	4	2	10000	2.024	3.068	16	16
5	4	3	10000	0.016	6.788	4	4
6	4	1	100000	2.928	15.365	256	256
6	4	2	100000	82.371	90.709	63	52
6	4	3	100000	72.996	147.849	16	14

Tabela 6.1: Limites para $K_q(n, R)$, obtidos pela busca tabu reativa

Analisando a tabela acima, podemos notar que o algoritmo atingiu o melhor valor conhecido para 12 das 14 instâncias testadas, gastando um tempo computacional relativamente pequeno. Com isto, devido aos bons resultados apresentados se espera que a performance do algoritmo seja similar para as coberturas curtas, ou seja, que os testes para as mesmas levem a soluções próximas da otimalidade, gerando bons limites superiores iniciais para o problema.

6.1.2

Resultados para o Problema de Códigos Curtos de Cobertura

A tabela 6.1.2 apresenta os valores de limites superiores obtidos para $c_q(n, R)$ pela busca tabu reativa. Nesta tabela, **UB** indica o limite superior obtido para a instância em questão, isto é, o custo da melhor solução encontrada pelos experimentos.

n	q	R	NI	TF(s)	TT(s)	UB
5	3	1	5000	0.004	0.192	13
6	3	1	100000	9.788	10.100	37
6	3	2	100000	0.036	25.593	8
7	3	1	100000	10.104	16.645	93
7	3	2	100000	54.467	66.148	17
7	3	3	100000	6.132	172.827	6
4	4	1	10000	0.004	0.384	10
5	4	1	10000	0.148	1.512	21
5	4	2	10000	2.328	5.932	5
6	4	1	100000	23.077	24.345	85
6	4	2	100000	39.874	115.419	21
6	4	3	100000	0.156	406.645	5
7	4	1	10000000	4932.12	6117.92	341
7	4	2	1000000	1073.673	2720.241	63
7	4	3	1000000	2192.81	9522.332	14

Tabela 6.2: Limites para $c_q(n, R)$, obtidos pela busca tabu reativa

Para nos ajudar a analisar o resultado, as tabelas abaixo apresentam ambos limite inferior e superior para $c_q(n, R)$, para $q = 3$, 4, onde $2 \leq n \leq 7$ e $1 \leq R \leq 3$.

n	R=1	R=2	R=3
2	1	1	1
3	3^a	1	1
4	4^b	1	1
5	$^e13^f$	$^e4^d$	1
6	$^e35\text{-}37^f$	$^e7\text{-}8^f$	3^c
7	$^e78\text{-}93^f$	$^e13\text{-}17^f$	$^e5\text{-}6^f$

Tabela 6.3: Limites para $c_3(n, R)$

Legendas para as tabelas: As instâncias não marcadas seguem o teorema 2.5. Os símbolos abaixo são usados para as instâncias restantes:

a : ref. [29].

b : Teorema 2.4.

c : Teorema 2.7.

d : $c_q(n+1, R+1) \leq c_q(n, R)$, ref. [29].

e : limite inferior do teorema 2.3, utilizando as tabelas em ref. [18].

f : limite superior da tabela 2 (ver códigos no Apêndice A).

g : Teorema 2.6.

n	R=1	R=2	R=3
2	1	1	1
3	3^a	1	1
4	$^e8\text{-}10^f$	2^g	1
5	21^b	$^e5^f$	1
6	$^e76\text{-}85^f$	$^e11\text{-}21^d$	$^e3\text{-}5^d$
7	$^e254\text{-}341^f$	$^e27\text{-}63\ ^f$	$^e5\text{-}14^f$

Tabela 6.4: Limites para $c_4(n, R)$

Vamos analisar os resultados computacionais para as coberturas curtas utilizando como base as tabelas acima. Além de melhorar diversos limites superiores dados pela proposição 2.1 e teorema 2.3, limites próximos da otimalidade foram atingidos em diversas instâncias, e o GAP entre os limites inferior e superior foi quase fechado para algumas delas. Além disto, também foram obtidas soluções exatas em algumas instâncias, cujos valores coincidem com os teoremas apresentados, teoremas 2.4 e 2.7. Dessa forma, estas instâncias nos dão evidências sobre a corretude e qualidade da abordagem proposta.

Como dito anteriormente, fazemos uma análise comparativa entre o códigos clássicos e os curtos de cobertura. A proposição 2.1 nos permite notar que para as coberturas curtas é necessário menos memória que para as coberturas clássicas. Além disto, a cardinalidade das soluções dos códigos curtos são menores que as do problema clássico. De fato, os valores e gaps nas tabelas 3 e 4 são significativamente menores que os apresentados em [18]. Como existe uma relação direta entre $c_q(n, R)$ e $K_q(n, R)$ através do teorema 2.3, a obtenção de bons limites para $c_q(n, R)$ pode implicar na melhoria de limites para $K_q(n, R)$, justificando o estudo dos códigos curtos, já que dependendo da estratégia de busca utilizada, o tamanho menor dos códigos curtos pode favorecer o desempenho da busca por códigos curtos sobre a busca pelos códigos clássicos.

6.2
Comparação entre as Heurísticas para Códigos de Cobertura

Os experimentos descritos nesta seção tiveram como principal objetivo testar a validade e a qualidade da HMGC, avaliando se a HMGC contribui para a melhoria dos resultados da busca tabu reativa e da busca tabu clássica aplicada aos problemas de códigos de cobertura, além de verificar qual a melhor estratégia, entre as propostas, para obtenção de limites para códigos de cobertura.

Nos experimentos realizados foram escolhidas sete tuplas de valores para (n, q, R), para cada uma delas os algoritmos foram testados tanto para $K_q(n, R)$ quanto para $c_q(n, R)$. Estas instâncias foram escolhidas com o objetivo de formar um conjunto representativo dos problemas abordados. Os experimentos foram realizados para os seguintes valores de (n, q, R): (6, 3, 1), (6, 4, 1), (6, 4, 2), (7, 3, 1), (7, 3, 2), (7, 4, 1) e (7, 4, 2).

Os testes foram realizados tomando por base o algoritmo da HMGC proposta (ver 5.2), utilizando para a mesma tanto a busca tabu reativa quanto a busca tabu sem o mecanismo de reação. Estes testes foram baseados no número de vezes em que a busca local (busca tabu reativa ou busca tabu) é executada pela HMGC proposta, lembrando que o algoritmo utilizado para a mesma foi o algoritmo de sucessivas transformações, o qual está descrito em 5.1.2. Pelo algoritmo notamos que a primeira execução da busca local ocorre no espaço original do problema, enquanto as demais execuções ocorrem em espaços que foram transformados sucessivas vezes dependendo da iteração corrente.

Para cada teste foi estabelecido um tempo limite total para execução da HMGC (**T-Total**) e o número de vezes que a busca local seria executada (n), sendo o tempo de execução de cada busca local $T = \frac{T-Total}{n}$. Como a primeira busca local é executada no espaço original do problema, os testes para $n = 1$ equivalem a testar a busca tabu reativa e a busca tabu de forma isolada, para os demais testes $n > 1$ o papel da HMGC passa a atuar, já que as buscas passam a serem feitas em espaços transformados pela HMGC.

A seguir analisamos os resultados apresentados para os testes com cada valor (n, q, R), apresentando para cada um uma tabela para $K_q(n, R)$ e outra para $c_q(n, R)$. Nas tabelas a seguir mostramos os resultados da HMGC utilizando a busca tabu reativa **BT-R** e a busca tabu sem o mecanismo de reação **BT**, sendo que para cada teste mostramos o custo da melhor solução encontrada (Melhor) pelo algoritmo e o tempo em que a mesma foi encontrada (Tempo), as demais convenções foram explicadas no parágrafo anterior. Os algoritmos foram executados para cada um dos seguintes valores de $n = \{1, 2, 5, 20, 50, 100\}$, sendo a solução inicial da busca gerada através da estratégia gulosa descrita em 6.1.

Tabela 6.5: Tabela de resultados dos algoritmos para $K_3(6,1)$

$K_3(6,1) \leq 73$						
T-Total(s) = 100	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 100	1	73	7.95	1	73	0.16
T(s) = 50	2	73	0.41	2	73	1.13
T(s) = 20	5	73	2.59	5	73	3.34
T(s) = 5	20	73	1.23	20	73	0.47
T(s) = 2	50	73	1.33	50	78	1.92
T(s) = 1	100	73	0.17	100	73	2.20

Tabela 6.6: Tabela de resultados dos algoritmos para $c_3(6,1)$

$c_3(6,1) \leq 37$						
T-Total(s) = 100	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 100	1	37	0.73	1	37	1.13
T(s) = 50	2	37	0.05	2	37	1.50
T(s) = 20	5	37	0.13	5	37	4.64
T(s) = 5	20	37	0.20	20	37	0.28
T(s) = 2	50	37	0.09	50	37	0.73
T(s) = 1	100	37	0.20	100	40	1.31

Os resultados acima mostram que tanto para $K_3(6,1)$ quanto para $c_3(6,1)$ as buscas não tiveram dificuldades para chegar no limite superior conhecido, o que serve para atestar a qualidade da abordagem usando a busca tabu proposta. Em praticamente todos os casos o valor do limite superior foi atingido na primeira execução da busca (seja a busca tabu reativa ou a busca tabu), ou seja, a busca no espaço original do problema. Este fato pode ser notado avaliando o tempo em que a melhor solução foi encontrada, se o mesmo for menor que o valor de T(tempo limite de cada busca), então a melhor solução foi encontrada na busca no espaço original. Devido a isto não é possível avaliar a contribuição da HMGC utilizando os resultados acima.

Tabela 6.7: Tabela de resultados dos algoritmos para $K_4(6,1)$

$K_4(6,1) \leq 256$						
T-Total(s) = 500	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 500	1	256	20.06	1	256	17.55
T(s) = 250	2	256	54.91	2	256	28.88
T(s) = 100	5	256	56.82	5	256	60.15
T(s) = 25	20	256	22.56	20	329	85.31
T(s) = 10	50	325	243.81	50	325	15.49
T(s) = 5	100	326	86.42	100	329	33.67

Tabela 6.8: Tabela de resultados dos algoritmos para $c_4(6,1)$

$c_4(6,1) \leq 85$						
T-Total(s) = 500	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 500	1	85	1.45	1	85	0.89
T(s) = 250	2	85	0.84	2	85	0.75
T(s) = 100	5	85	1.17	5	85	0.39
T(s) = 25	20	85	0.67	20	85	1.61
T(s) = 10	50	85	0.67	50	85	0.64
T(s) = 5	100	85	2.36	100	85	1.61

Os resultados para o $c_4(6,1)$ seguem a análise feita anteriormente para $(n,q,R) = (6,3,1)$. Já para $K_4(6,1)$ percebemos que a HMGC acaba atrapalhando os resultados das buscas para a partir de $n = 20$, ou seja, quando na primeira execução da busca não se consegue chegar ao valor do limite superior. As sucessivas melhorias da solução nas buscas nos espaços transformados não são suficientes para atingir o valor do limite superior.

Tabela 6.9: Tabela de resultados dos algoritmos para $K_4(6,2)$

$K_4(6,2) \leq 52$						
T-Total(s) = 500	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 500	1	65	9.38	1	67	2.31
T(s) = 250	2	65	19.39	2	67	59.11
T(s) = 100	5	64	82.74	5	66	107.89
T(s) = 25	20	66	5.07	20	67	464.69
T(s) = 10	50	65	13.71	50	66	167.05
T(s) = 5	100	64	20.31	100	67	30.00

Tabela 6.10: Tabela de resultados dos algoritmos para $c_4(6,2)$

$c_4(6,2) \leq 21$						
T-Total(s) = 500	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 500	1	21	79.12	1	21	217.01
T(s) = 250	2	19	385.19	2	21	14.88
T(s) = 100	5	**17**	410.59	5	21	29.31
T(s) = 25	20	18	84.11	20	21	15.02
T(s) = 10	50	21	0.87	50	21	29.52
T(s) = 5	100	21	0.25	100	21	18.34

Os resultados acima mostram que a busca tabu reativa possui um desempenho superior a busca tabu sem o mecanismo de reação, configurando uma importante melhoria ao esquema básico da busca tabu.

Os resultados para o $K_4(6,2)$ mostram uma dificuldade da busca em atingir o valor do limite superior. Uma das possíveis explicações para o baixo desempenho da busca seria a configuração de parâmetros, que pode não ter sido feita de forma adequada, além da possível grande dificuldade da instância.

Analisando a tabela para o $c_4(6,2)$ percebemos claramente que a HMGC contribuiu para os resultados da busca tabu reativa, o limite superior anterior foi baixado de 21 para 17 no teste da HMGC com a busca tabu reativa para $n = 5$. O papel de livrar a busca local de mínimos locais, provavelmente as soluções de tamanho 21, foi desempenhado com sucesso pela HMGC com a busca tabu reativa, provando então o conceito da HMGC.

Tabela 6.11: Tabela de resultados dos algoritmos para $K_3(7,1)$

$K_3(7,1) \leq 186$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	186	234.71	1	194	846.09
T(s) = 500	2	192	891.36	2	194	997.25
T(s) = 200	5	194	44.52	5	195	181.06
T(s) = 50	20	194	67.46	20	210	15.29
T(s) = 20	50	210	166.79	50	208	16.83
T(s) = 10	100	209	117.57	100	211	114.26

Tabela 6.12: Tabela de resultados dos algoritmos para $c_3(7,1)$

$c_3(7,1) \leq 93$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	93	5.66	1	100	137.75
T(s) = 500	2	101	52.47	2	101	1.97
T(s) = 200	5	101	37.88	5	100	226.91
T(s) = 50	20	97	48.19	20	101	8.81
T(s) = 20	50	97	2.11	50	101	17.61
T(s) = 10	100	102	2.53	100	103	15.95

Os resultados para $K_3(7,1)$ e para $c_3(7,1)$ mostram que apenas a busca tabu reativa executada de forma isolada foi capaz de atingir o valor do limite superior, e além disto a estratégia de escape de mínimos locais da HMGC não conseguiu contribuir para a melhoria dos resultados. Também podemos notar que no geral a estratégia utilizando a busca tabu reativa obteve melhores resultados.

Tabela 6.13: Tabela de resultados dos algoritmos para $K_3(7,2)$

$K_3(7,2) \leq 34$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	47	22.01	1	48	61.20
T(s) = 500	2	47	163.71	2	48	87.92
T(s) = 200	5	47	4.12	5	34	299.36
T(s) = 50	20	41	395.12	20	47	305.88
T(s) = 20	50	48	0.30	50	47	235.83
T(s) = 10	100	49	15.93	100	47	52.84

Tabela 6.14: Tabela de resultados dos algoritmos para $c_3(7,2)$

$c_3(7,2) \leq 17$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	17	4.32	1	18	6.55
T(s) = 500	2	17	0.95	2	17	549.00
T(s) = 200	5	17	0.69	5	17	105.33
T(s) = 50	20	17	0.34	20	18	83.88
T(s) = 20	50	17	6.16	50	17	40.00
T(s) = 10	100	18	0.59	100	18	13.29

Os resultados para o $K_3(7,2)$ mostram que provavelmente existe um mínimo local nas soluções de tamanho 47, e que as buscas tiveram dificuldades em superar este mínimo local, onde apenas em 2 casos foi possível atingir soluções menores, e em ambos notamos que as soluções foram encontradas graças a ajuda da HMGC, pois foram encontradas em espaços transformados pela mesma. Um fato interessante a se notar é que o melhor resultado foi obtido pela HMGC usando a busca tabu sem o mecanismo de reação.

Analisando os resultados para $c_3(7,2)$ percebemos que, na maioria dos casos, o limite superior foi atingido já na primeira execução da busca local (reativa ou tabu), novamente atestando a qualidade dos algoritmos propostos, sendo que os tempos de encontrar o limite superior foram bem menores na busca tabu reativa.

Tabela 6.15: Tabela de resultados dos algoritmos para $K_4(7,1)$

$K_4(7,1) \leq 992$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	1227	873.26	1	1235	829.01
T(s) = 500	2	1229	966.21	2	1232	851.39
T(s) = 200	5	1229	567.64	5	1237	411.73
T(s) = 50	20	1243	140.82	20	1264	43.24
T(s) = 20	50	1239	900.23	50	1247	471.19
T(s) = 10	100	1241	135.66	100	1254	170.28

Tabela 6.16: Tabela de resultados dos algoritmos para $c_4(7,1)$

$c_4(7,1) \leq 341$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	389	724.08	1	400	258.01
T(s) = 500	2	398	254.23	2	400	980.59
T(s) = 200	5	403	403.74	5	405	140.87
T(s) = 50	20	401	237.86	20	408	62.79
T(s) = 20	50	403	203.26	50	405	709.81
T(s) = 10	100	409	200.17	100	408	19.06

Os resultados para $K_4(7,1)$ e $c_4(7,1)$ mostram que o desempenho da busca tabu reativa foi superior à busca tabu, além disto a HMGC não conseguiu contribuir para a melhoria dos resultados, pelo contrário na maioria dos casos a medida em que o valor de n aumenta as soluções pioram. Podemos notar também que as buscas ficaram longe do valor do limite superior, especialmente para $K_4(7,1)$. Provavelmente teríamos melhores resultados se o tempo computacional para os algoritmos fosse maior, já que em muitos casos a melhor solução foi encontrada quase no limite de tempo de 1000 segundos e pelo fato do tamanho das instâncias em questão ser bem maior que as instâncias anteriores.

Tabela 6.17: Tabela de resultados dos algoritmos para $K_4(7,2)$

$K_4(7,2) \leq 128$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	211	476.83	1	222	604.45
T(s) = 500	2	211	260.13	2	221	0.00
T(s) = 200	5	211	230.66	5	221	200.44
T(s) = 50	20	212	150.39	20	222	325.53
T(s) = 20	50	213	127.63	50	220	420.36
T(s) = 10	100	211	830.45	100	220	90.36

Tabela 6.18: Tabela de resultados dos algoritmos para $c_4(7,2)$

$c_4(7,2) \leq 63$						
T-Total(s) = 1000	**BT-R**			**BT**		
-	n	Melhor	Tempo(s)	n	Melhor	Tempo(s)
T(s) = 1000	1	67	105.44	1	73	571.28
T(s) = 500	2	67	30.20	2	73	58.63
T(s) = 200	5	67	138.84	5	73	130.55
T(s) = 50	20	67	388.55	20	73	99.20
T(s) = 20	50	65	243.80	50	70	203.41
T(s) = 10	100	67	26.06	100	74	68.25

Os resultados para $K_4(7,2)$ e $c_4(7,2)$ mostram novamente uma superioridade de desempenho da busca tabu reativa, além disto, as buscas ficaram provavelmente presas em mínimos locais, pois não foi possível atingir os valores de limite superior e alguns resultados se repetiram várias vezes, como o 211 para o $K_4(7,2)$ e o 67 para o $c_4(7,2)$. No caso dos resultados para o $c_4(7,2)$ mais uma vez podemos perceber a contribuição da HMGC para os resultados, já que os melhores valores encontrados tanto pela busca tabu reativa quanto pela busca tabu ocorreram para $n = 50$ durante a busca em um espaço transformado pela HMGC.

Analisando os resultados de uma forma geral, podemos concluir que a abordagem utilizando o mecanismo de reação é superior ao esquema básico da busca tabu, mostrando que a busca tabu reativa é uma importante variação da busca tabu. Além disto, na maioria dos casos os melhores resultados foram obtidos pela busca tabu reativa utilizada de forma isolada. Porém, em alguns casos pudemos perceber a contribuição da HMGC, principalmente no caso do $c_4(6,2)$, onde o limite superior foi diminuído de 21 para 17 graças a mesma.

Com isto, pudemos provar o conceito da nova heurística apresentada neste trabalho, a HMGC (Heurística de Melhoria via Geração de Colunas).